
KS22/KS20 子系列参考手册

支持: MKS22FN256VLL12, MKS22FN256VLH12,
MKS22FN256VFT12; MKS22FN128VLL12, MKS22FN128VLH12,
MKS22FN128VFT12; MKS20FN256VLL12, MKS20FN256VLH12,
MKS20FN256VFT12; MKS20FN128VLL12, MKS20FN128VLH12,
MKS20FN128VFT12.

Document Number: KS22P100M120SF0RM

Rev 3, May 2016





内容

小节编号

标题

页

第 1 章 关于本手册

1.1	读者.....	51
1.2	组织.....	51
1.3	模块说明.....	51
1.3.1	示例：用于阐明相同章节中的内容的芯片特性信息.....	52
1.3.2	示例：参考不同章节的芯片特性信息.....	53
1.4	寄存器说明.....	54
1.5	约定.....	55
1.5.1	编码系统.....	55
1.5.2	印刷符号.....	55
1.5.3	特殊术语.....	56

第 2 章 简介

2.1	概述.....	57
2.2	Kinetis KS 系列特性汇总.....	57
2.3	结构框图.....	60
2.4	模块功能类别.....	61
2.4.1	ARM® Cortex®-M4 内核模块.....	62
2.4.2	系统模块.....	63
2.4.3	存储器和存储器接口.....	64
2.4.4	时钟.....	64
2.4.5	安全性和完整性模块.....	64
2.4.6	模拟模块.....	65
2.4.7	定时器模块.....	65
2.4.8	通信接口.....	66
2.4.9	人机接口.....	66
2.5	可订购部件编号.....	66

第 3 章 内核概述

3.1	ARM Cortex-M4 内核配置.....	69
3.1.1	总线、互连和接口.....	70
3.1.2	系统节拍定时器.....	70
3.1.3	调试设施.....	70
3.1.4	内核权限级别.....	70
3.2	嵌套向量中断控制器 (NVIC) 配置.....	71
3.2.1	中断优先级.....	71
3.2.2	不可屏蔽中断.....	71
3.2.3	中断通道分配.....	72
3.3	异步唤醒中断控制器 (AWIC) 配置.....	76
3.3.1	唤醒源.....	77
3.4	FPU 配置.....	78
3.5	JTAG 控制器配置.....	78

第 4 章 存储器和存储器接口

4.1	Flash 存储器配置.....	81
4.1.1	Flash 存储器类型.....	81
4.1.2	Flash 存储器大小.....	82
4.1.3	Flash 加密.....	82
4.1.4	Flash 程序限制.....	82
4.1.5	Flash 模式.....	82
4.1.6	擦除所有 Flash 内容.....	82
4.1.7	FTF_FOFT 寄存器.....	82
4.2	Flash 存储器控制器配置.....	83
4.2.1	主机数.....	83
4.3	SRAM 配置.....	83
4.3.1	SRAM 大小.....	84

小节编号	标题	页
4.3.2	低功耗模式下的 SRAM 保留.....	84
4.4	系统寄存器文件配置.....	85
4.4.1	系统寄存器文件.....	85
4.5	VBAT 寄存器文件配置.....	85
4.5.1	VBAT 寄存器文件.....	86

第 5 章 存储器映射

5.1	简介.....	87
5.2	系统存储器映射.....	87
5.2.1	位带别名区.....	88
5.2.2	Flash 访问控制简介.....	89
5.3	Flash 存储器映射.....	90
5.3.1	可选的非易失性 IRC 用户调整说明.....	90
5.4	SRAM 存储器映射.....	91
5.5	外设桥 (AIPS-Lite) 存储器映射.....	91
5.5.1	先写后读顺序和存储器操作所需的串行化.....	91
5.5.2	外设桥 0 (AIPS-Lite 0) 存储器映射.....	92
5.6	私有外设总线 (PPB) 存储器映射.....	95

第 6 章 时钟分布

6.1	简介.....	97
6.2	编程模型.....	97
6.3	高级设备时钟图.....	97
6.4	时钟定义.....	98
6.4.1	器件时钟汇总.....	99
6.5	内部时钟要求.....	102
6.5.1	复位后的时钟分频器值.....	103
6.5.2	VLPR 模式时钟.....	103
6.6	时钟门控.....	104

小节编号	标题	页
6.7	模块时钟.....	104
6.7.1	PMC 1-kHz LPO 时钟.....	105
6.7.2	IRC 48MHz 时钟.....	106
6.7.3	WDOG 时钟.....	106
6.7.4	调试跟踪时钟.....	107
6.7.5	PORT 数字滤波器时钟.....	107
6.7.6	LPTMR 时钟.....	108
6.7.7	RTC_CLKOUT 和 CLKOUT32K 时钟.....	108
6.7.8	USB FS OTG 控制器时钟.....	109
6.7.9	UART 时钟.....	110
6.7.10	LPUART0 时钟.....	110
6.7.11	I2S/SAI 时钟.....	111
6.7.12	FlexIO 时钟.....	112
6.7.13	LPI2C 时钟.....	112
6.7.14	TPM 时钟.....	113
6.7.15	FlexCAN 时钟.....	113

第 7 章 复位与引导

7.1	简介.....	115
7.2	复位.....	115
7.2.1	上电复位 (POR)	115
7.2.2	系统复位源.....	116
7.2.3	MCU 复位.....	119
7.2.4	复位引脚	120
7.2.5	调试复位.....	121
7.3	启动.....	122
7.3.1	启动源.....	122
7.3.2	启动选项.....	122
7.3.3	FOPT 启动选项.....	122

小节编号	标题	页
7.3.4	启动序列.....	123
第 8 章 电源管理		
8.1	简介.....	125
8.2	时钟模式.....	125
8.2.1	部分停止.....	125
8.2.2	DMA 唤醒.....	126
8.2.3	运算操作.....	126
8.2.4	外设休眠.....	127
8.2.5	时钟门控.....	128
8.3	功耗模式描述.....	128
8.4	进入和退出功耗模式.....	130
8.5	功耗模式转换.....	131
8.6	功耗模式关闭序列.....	132
8.7	Flash 程序限制.....	132
8.8	低功率模式中的模块操作.....	132
第 9 章 加密		
9.1	简介.....	137
9.2	Flash 加密.....	137
9.3	与其它模块之间的安全性交互.....	137
9.3.1	与调试器之间的安全性.....	137
第 10 章 调试		
10.1	简介.....	139
10.1.1	参考资料.....	140
10.2	调试端口.....	140
10.2.1	JTAG 至 SWD 更改顺序.....	141
10.2.2	JTAG 至 cJTAG 更改顺序.....	141
10.3	调试端口引脚说明.....	142

小节编号	标题	页
10.4	系统 TAP 连接.....	142
10.4.1	IR 代码.....	142
10.5	JTAG 状态和控制寄存器.....	143
10.5.1	MDM-AP 控制寄存器.....	144
10.5.2	MDM-AP 状态寄存器.....	145
10.6	调试复位.....	146
10.7	AHB-AP.....	147
10.8	ITM.....	147
10.9	内核跟踪连接性.....	148
10.10	TPIU.....	148
10.11	DWT.....	148
10.12	低功耗模式中的调试.....	148
10.12.1	低功耗模式中的调试模块状态.....	149
10.13	调试和加密.....	150

第 11 章 信号复用及信号说明

11.1	简介.....	151
11.2	引脚分配.....	151
11.2.1	信号复用和引脚分配.....	151
11.2.2	引脚配置.....	155
11.3	模块信号说明表.....	158
11.3.1	内核模块.....	158
11.3.2	系统模块.....	159
11.3.3	时钟模块.....	159
11.3.4	模拟.....	160
11.3.5	定时器模块.....	160
11.3.6	通信接口.....	162
11.3.7	人机接口 (HMI)	165

第 12 章

小节编号	标题	页
端口控制和中断 (PORT)		
12.1	此模块的芯片实现细节.....	167
12.1.1	信号复用集成.....	167
12.2	简介.....	169
12.3	概述.....	169
12.3.1	特性.....	169
12.3.2	工作模式.....	170
12.4	外部信号说明.....	171
12.5	详细信号描述.....	171
12.6	存储器映射和寄存器定义.....	171
12.6.1	引脚控制寄存器 n (PORTx_PCRn).....	177
12.6.2	全局引脚控制低位寄存器 (PORTx_GPCLR).....	180
12.6.3	全局引脚控制高位寄存器 (PORTx_GPCHR).....	180
12.6.4	中断状态标志寄存器 (PORTx_ISFR).....	181
12.6.5	数字滤波器使能寄存器 (PORTx_DFER).....	181
12.6.6	数字滤波器时钟寄存器 (PORTx_DFRCR).....	182
12.6.7	数字滤波器宽度寄存器 (PORTx_DFWR).....	182
12.7	功能说明.....	183
12.7.1	引脚控制.....	183
12.7.2	全局引脚控制.....	184
12.7.3	外部中断.....	184
12.7.4	数字滤波器.....	185
第 13 章		
系统集成模块 (SIM)		
13.1	简介.....	187
13.1.1	特性.....	187
13.2	存储器映射和寄存器定义.....	188
13.2.1	系统选项寄存器 1 (SIM_SOPT1).....	189
13.2.2	系统选项寄存器 2 (SIM_SOPT2).....	190

小节编号	标题	页
13.2.3	系统选项寄存器 5 (SIM_SOPT5).....	192
13.2.4	系统选项寄存器 7 (SIM_SOPT7).....	194
13.2.5	系统选项寄存器 9 (SIM_SOPT9).....	195
13.2.6	系统能够设备标识寄存器 (SIM_SDID).....	197
13.2.7	系统时钟门控寄存器 4 (SIM_SCGC4).....	199
13.2.8	系统时钟门控寄存器 5 (SIM_SCGC5).....	201
13.2.9	系统时钟门控寄存器 6 (SIM_SCGC6).....	202
13.2.10	系统时钟门控寄存器 7 (SIM_SCGC7).....	206
13.2.11	系统时钟分频器寄存器 1 (SIM_CLKDIV1).....	206
13.2.12	系统时钟分频器寄存器 2 (SIM_CLKDIV2).....	208
13.2.13	Flash 配置寄存器 1 (SIM_FCFG1).....	209
13.2.14	Flash 配置寄存器 2 (SIM_FCFG2).....	211
13.2.15	高电平寄存器唯一标识 (SIM_UIDH).....	211
13.2.16	中高电平寄存器唯一标识 (SIM_UIDMH).....	212
13.2.17	中低电平寄存器唯一标识 (SIM_UIDML).....	212
13.2.18	低电平寄存器唯一标识 (SIM_UIDL).....	213
13.2.19	系统时钟分频器寄存器 3 (SIM_CLKDIV3).....	213
13.2.20	其他控制寄存器 (SIM_MISCCTL).....	214
13.3	功能说明.....	215

第 14 章 Kinetis Flashloader

14.1	此模块的芯片实现细节.....	217
14.1.1	Kinetis Flashloader.....	217
14.2	简介.....	218
14.3	功能说明.....	219
14.3.1	存储器映射.....	219
14.3.2	启动流程.....	219
14.3.3	时钟配置.....	221
14.3.4	Flashloader 协议.....	221

小节编号	标题	页
14.3.5	Flashloader 包类型.....	225
14.3.6	Flashloader 命令 API.....	231
14.4	支持的外设.....	251
14.4.1	I2C 外设.....	251
14.4.2	SPI 外设.....	253
14.4.3	UART 外设.....	255
14.4.4	USB 外设.....	257
14.4.5	CAN (或 FlexCAN) 外设.....	259
14.5	Get/SetProperty 命令属性.....	261
14.5.1	属性定义.....	262
14.6	Kinetis Flashloader 状态错误代码.....	263

第 15 章 复位控制模块 (RCM)

15.1	简介.....	267
15.2	复位存储器映射和寄存器说明.....	267
15.2.1	系统复位状态寄存器 0 (RCM_SRS0).....	268
15.2.2	系统复位状态寄存器 1 (RCM_SRS1).....	269
15.2.3	复位引脚滤波器控制寄存器 (RCM_RPFC).....	270
15.2.4	复位引脚滤波器带宽寄存器 (RCM_RPFW).....	271
15.2.5	粘滞系统复位状态寄存器 0 (RCM_SRS0).....	273
15.2.6	粘滞系统复位状态寄存器 1 (RCM_SRS1).....	274

第 16 章 系统模式控制器 (SMC)

16.1	简介.....	277
16.2	操作模式.....	277
16.3	存储器映射和寄存器说明.....	279
16.3.1	功耗模式保护寄存器 (SMC_PMPROT).....	279
16.3.2	功耗模式控制寄存器 (SMC_PMCTRL).....	281
16.3.3	停止控制寄存器 (SMC_STOPCTRL).....	282

小节编号	标题	页
16.3.4	功耗模式状态寄存器 (SMC_PMSTAT).....	283
16.4	功能说明.....	284
16.4.1	功耗模式转换.....	284
16.4.2	功耗模式进入/退出顺序.....	287
16.4.3	运行模式.....	288
16.4.4	等待模式.....	290
16.4.5	停止模式.....	291
16.4.6	低功耗模式中的调试.....	293

第 17 章 电源管理控制器 (PMC)

17.1	简介.....	295
17.2	特性.....	295
17.3	低压检测 (LVD) 系统.....	295
17.3.1	LVD 复位操作.....	296
17.3.2	LVD 中断操作.....	296
17.3.3	低压警告 (LVW) 中断操作.....	296
17.4	高压检测 (HVD) 系统.....	297
17.4.1	HVD 复位操作.....	297
17.4.2	HVD 中断操作.....	297
17.5	I/O 保留.....	297
17.6	存储器映射和寄存器说明.....	298
17.6.1	低压检测状态和控制 1 寄存器 (PMC_LVDSC1).....	298
17.6.2	低压检测状态和控制 2 寄存器 (PMC_LVDSC2).....	299
17.6.3	稳压器状态和控制寄存器 (PMC_REGSC).....	301
17.6.4	高压检测状态和控制 1 寄存器 (PMC_HVDSC1).....	302

第 18 章 低漏电唤醒单元 (LLWU)

18.1	此模块的芯片实现细节.....	305
18.1.1	唤醒源.....	305

小节编号	标题	页
18.2	简介.....	306
18.2.1	特性.....	307
18.2.2	操作模式.....	307
18.2.3	结构框图.....	308
18.3	LLWU 信号说明.....	309
18.4	存储器映射/寄存器定义.....	310
18.4.1	LLWU 引脚使能 1 寄存器 (LLWU_PE1).....	311
18.4.2	LLWU 引脚使能 2 寄存器 (LLWU_PE2).....	312
18.4.3	LLWU 引脚使能 3 寄存器 (LLWU_PE3).....	313
18.4.4	LLWU 引脚使能 4 寄存器 (LLWU_PE4).....	314
18.4.5	LLWU 引脚使能 5 寄存器 (LLWU_PE5).....	315
18.4.6	LLWU 引脚使能 6 寄存器 (LLWU_PE6).....	317
18.4.7	LLWU 引脚使能 7 寄存器 (LLWU_PE7).....	318
18.4.8	LLWU 引脚使能 8 寄存器 (LLWU_PE8).....	319
18.4.9	LLWU 模块使能寄存器 (LLWU_ME).....	320
18.4.10	LLWU 引脚标志 1 寄存器 (LLWU_PF1).....	322
18.4.11	LLWU 引脚标志 2 寄存器 (LLWU_PF2).....	323
18.4.12	LLWU 引脚标志 3 寄存器 (LLWU_PF3).....	325
18.4.13	LLWU 引脚标志 4 寄存器 (LLWU_PF4).....	326
18.4.14	LLWU 模块标志 5 寄存器 (LLWU_MF5).....	328
18.4.15	LLWU 引脚滤波器 1 寄存器 (LLWU_FILT1).....	330
18.4.16	LLWU 引脚滤波器 2 寄存器 (LLWU_FILT2).....	331
18.5	功能说明.....	332
18.5.1	LLS 模式.....	332
18.5.2	VLLS 模式.....	332
18.5.3	初始化.....	332

第 19 章 杂项控制模块(MCM)

19.1	简介.....	335
------	---------	-----

小节编号	标题	页
19.1.1	特性.....	335
19.2	存储器映射/寄存器说明.....	335
19.2.1	交叉开关(AXBS)从机配置 (MCM_PLASC).....	336
19.2.2	交叉开关(AXBS)主机配置 (MCM_PLAMC).....	336
19.2.3	交叉开关 (AXBS) 控制寄存器 (MCM_PLACR).....	337
19.2.4	中断状态和控制寄存器 (MCM_ISCR).....	337
19.2.5	计算操作控制寄存器 (MCM_CPO).....	340
19.3	功能说明.....	341
19.3.1	中断.....	341

第 20 章 交叉开关-精简版 (AXBS-Lite)

20.1	此模块的芯片实现细节.....	343
20.1.1	简化交叉开关主机分配.....	343
20.1.2	简化交叉开关从机分配.....	343
20.2	简介.....	343
20.2.1	特性.....	344
20.3	存储器映射/寄存器定义.....	344
20.4	功能说明.....	344
20.4.1	常规操作.....	344
20.4.2	仲裁.....	345
20.5	初始化/应用信息.....	346

第 21 章 外设桥 (AIPS-Lite)

21.1	此模块的芯片实现细节.....	347
21.1.1	外设桥的数量.....	347
21.1.2	存储器映射.....	347
21.2	简介.....	347
21.2.1	特性.....	347
21.2.2	一般操作.....	348

小节编号	标题	页
21.3	存储器映射/寄存器定义.....	348
21.4	功能说明.....	348
21.4.1	访问支持.....	348

第 22 章 直接存储器访问多路复用器 (DMAMUX)

22.1	此模块的芯片实现细节.....	349
22.1.1	DMA MUX 请求源.....	349
22.1.2	DMA 通过 PIT 触发器传输.....	351
22.2	简介.....	351
22.2.1	概述.....	351
22.2.2	特性.....	352
22.2.3	操作模式.....	352
22.3	外部信号说明.....	353
22.4	存储器映射/寄存器定义.....	353
22.4.1	端字节序.....	353
22.4.2	通道配置寄存器 (DMAMUX_CHCFGn).....	354
22.5	功能说明.....	355
22.5.1	带定期触发功能的 DMA 通道.....	355
22.5.2	无触发功能的 DMA 通道.....	357
22.5.3	始终使能的 DMA 源.....	358
22.6	初始化/应用信息.....	359
22.6.1	复位.....	359
22.6.2	使能和配置源.....	359

第 23 章 增强型直接存储器访问(eDMA)

23.1	简介.....	363
23.1.1	eDMA 系统结构框图.....	363
23.1.2	结构块.....	364
23.1.3	特性.....	365

小节编号	标题	页
23.2	操作模式.....	366
23.3	存储器映射/寄存器定义.....	367
23.3.1	TCD 存储器.....	367
23.3.2	TCD 初始化.....	367
23.3.3	TCD 结构.....	367
23.3.4	保留存储器和位字段.....	368
23.3.5	控制寄存器 (DMA_CR).....	378
23.3.6	错误状态寄存器 (DMA_ES).....	380
23.3.7	使能请求寄存器 (DMA_ERQ).....	382
23.3.8	使能错误中断寄存器 (DMA_EEI).....	384
23.3.9	清除使能错误中断寄存器 (DMA_CEEI).....	386
23.3.10	置位使能错误中断寄存器 (DMA_SEEI).....	387
23.3.11	清除使能请求寄存器 (DMA_CERQ).....	388
23.3.12	置位使能请求寄存器 (DMA_SERQ).....	389
23.3.13	清除 DONE 状态位寄存器 (DMA_CDNE).....	390
23.3.14	置位 START 位寄存器 (DMA_SSRT).....	391
23.3.15	清除错误寄存器 (DMA_CERR).....	392
23.3.16	清除中断请求寄存器 (DMA_CINT).....	393
23.3.17	中断请求寄存器 (DMA_INT).....	393
23.3.18	错误寄存器 (DMA_ERR).....	396
23.3.19	硬件请求状态寄存器 (DMA_HRS).....	398
23.3.20	使能在停止状态下的异步请求寄存器 (DMA_EARS).....	401
23.3.21	通道 n 优先级寄存器 (DMA_DCHPRI _n).....	403
23.3.22	TCD 源地址 (DMA_TCD _n _SADDR).....	404
23.3.23	TCD 有符号的源地址偏移 (DMA_TCD _n _SOFF).....	404
23.3.24	TCD 传输属性 (DMA_TCD _n _ATTR).....	405
23.3.25	TCD 次字节数目 (次循环映射禁用) (DMA_TCD _n _NBYTES_MLNO).....	406
23.3.26	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD _n _NBYTES_MLOFFNO).....	406
23.3.27	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD _n _NBYTES_MLOFFYES).....	408

小节编号	标题	页
23.3.28	TCD 最后源地址调整 (DMA_TCDn_SLAST).....	409
23.3.29	TCD 目标地址 (DMA_TCDn_DADDR).....	409
23.3.30	TCD 有符号的目标地址偏移 (DMA_TCDn_DOFF).....	410
23.3.31	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCDn_CITER_ELINKYES).....	410
23.3.32	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCDn_CITER_ELINKNO).....	411
23.3.33	TCD 最后的目标地址调整/分散集中地址 (DMA_TCDn_DLASTSGA).....	412
23.3.34	TCD 控制和状态 (DMA_TCDn_CSR).....	413
23.3.35	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCDn_BITER_ELINKYES).....	415
23.3.36	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCDn_BITER_ELINKNO).....	416
23.4	功能说明.....	416
23.4.1	eDMA 基础数据流.....	416
23.4.2	故障报告和处理.....	419
23.4.3	通道抢占.....	421
23.4.4	性能.....	421
23.5	初始化/应用信息.....	425
23.5.1	eDMA 初始化.....	425
23.5.2	编程错误.....	427
23.5.3	仲裁模式注意事项.....	427
23.5.4	执行 DMA 传输.....	428
23.5.5	监控传输描述符状态.....	431
23.5.6	通道链接.....	433
23.5.7	动态编程.....	434

第 24 章 外部看门狗监控器 (EWM)

24.1	此模块的芯片实现细节.....	439
24.1.1	EWM 时钟.....	439
24.1.2	EWM 低功耗模式.....	439
24.1.3	EWM_OUT 低功耗模式下的引脚状态.....	439
24.2	简介.....	440

小节编号	标题	页
24.2.1	特性.....	440
24.2.2	工作模式.....	440
24.2.3	结构框图.....	441
24.3	EWM 信号说明.....	442
24.4	存储器映射/寄存器定义.....	442
24.4.1	控制寄存器 (EWM_CTRL).....	443
24.4.2	服务寄存器 (EWM_SERV).....	444
24.4.3	比较低电平寄存器 (EWM_CMPL).....	444
24.4.4	比较高电平寄存器 (EWM_CMPH).....	445
24.4.5	时钟预分频器寄存器 (EWM_CLKPRESCALER).....	445
24.5	功能说明.....	446
24.5.1	EWM_out 信号.....	446
24.5.2	EWM_in 信号.....	447
24.5.3	EWM 计数器.....	447
24.5.4	EWM 比较寄存器.....	447
24.5.5	EWM 刷新机制.....	448
24.5.6	EWM 中断.....	448
24.5.7	计数器时钟预分频器.....	448

第 25 章 看门狗定时器 (WDOG)

25.1	此模块的芯片实现细节.....	449
25.1.1	WDOG 时钟.....	449
25.1.2	WDOG 低功耗模式.....	449
25.2	简介.....	450
25.3	特性.....	450
25.4	功能概述.....	451
25.4.1	解锁和更新看门狗.....	452
25.4.2	看门狗配置时间(WCT).....	453
25.4.3	刷新看门狗.....	453

小节编号	标题	页
25.4.4	窗口化操作模式.....	454
25.4.5	看门狗禁用操作模式.....	454
25.4.6	调试运行模式.....	454
25.5	测试看门狗.....	455
25.5.1	快速测试.....	455
25.5.2	字节测试.....	455
25.6	备份复位生成器.....	456
25.7	生成的复位和中断.....	457
25.8	存储器映射和寄存器定义.....	457
25.8.1	看门狗状态和控制寄存器高位 (WDOG_STCTRLH).....	458
25.8.2	看门狗状态和控制寄存器低位 (WDOG_STCTRL).....	460
25.8.3	看门狗超时值寄存器高位 (WDOG_TOVALH).....	460
25.8.4	看门狗超时值寄存器低位 (WDOG_TOVAL).....	460
25.8.5	看门狗窗口寄存器高位 (WDOG_WINH).....	461
25.8.6	看门狗窗口寄存器低位 (WDOG_WINL).....	461
25.8.7	看门狗刷新寄存器 (WDOG_REFRESH).....	462
25.8.8	看门狗解锁寄存器 (WDOG_UNLOCK).....	462
25.8.9	看门狗定时器输出寄存器高位 (WDOG_TMROUTH).....	462
25.8.10	看门狗定时器输出寄存器低位 (WDOG_TMROUTL).....	463
25.8.11	看门狗复位计数寄存器 (WDOG_RSTCNT).....	463
25.8.12	看门狗预分频器寄存器 (WDOG_PRESC).....	463
25.9	通过 8 位访问的看门狗操作.....	464
25.9.1	通用准则.....	464
25.9.2	通过 8 位访问的刷新和解锁操作.....	464
25.10	看门狗操作限制.....	465

第 26 章 多用途时钟产生器 (MCG)

26.1	此模块的芯片实现细节.....	467
26.1.1	MCG 振荡器时钟输入选项.....	467

小节编号	标题	页
26.1.2	MCG 实例化信息.....	467
26.2	简介.....	468
26.2.1	特性.....	468
26.2.2	操作模式.....	470
26.3	外部信号说明.....	471
26.4	存储器映射/寄存器定义.....	471
26.4.1	MCG 控制 1 寄存器 (MCG_C1).....	472
26.4.2	MCG 控制 2 寄存器 (MCG_C2).....	473
26.4.3	MCG 控制 3 寄存器 (MCG_C3).....	474
26.4.4	MCG 控制 4 寄存器 (MCG_C4).....	474
26.4.5	MCG 控制 5 寄存器 (MCG_C5).....	475
26.4.6	MCG 控制 6 寄存器 (MCG_C6).....	476
26.4.7	MCG 状态寄存器 (MCG_S).....	478
26.4.8	MCG 状态和控制寄存器 (MCG_SC).....	479
26.4.9	MCG 自动调整比较值高电平寄存器 (MCG_ATCVH).....	480
26.4.10	MCG 自动调整比较值低电平寄存器 (MCG_ATCVL).....	481
26.4.11	MCG 控制 7 寄存器 (MCG_C7).....	481
26.4.12	MCG 控制 8 寄存器 (MCG_C8).....	482
26.4.13	MCG 控制 12 寄存器 (MCG_C12).....	482
26.4.13	MCG 状态 2 寄存器 (MCG_S2).....	483
26.4.13	MCG 测试 3 寄存器 (MCG_T3).....	483
26.5	功能说明.....	483
26.5.1	MCG 模式状态图.....	483
26.5.2	低功耗位使用.....	487
26.5.3	MCG 内部参考时钟.....	487
26.5.4	外部参考时钟.....	488
26.5.5	MCG 固定频率时钟.....	488
26.5.6	MCG PLL 时钟.....	489
26.5.7	MCG 自动调整 (ATM).....	489

小节编号	标题	页
26.6	初始化/应用信息.....	490
26.6.1	MCG 模块初始化序列.....	490
26.6.2	使用 32.768 kHz 参考.....	492
26.6.3	MCG 模式切换.....	492

第 27 章 振荡器 (OSC)

27.1	此模块的芯片实现细节.....	501
27.1.1	OSC 在 MCG 下的操作模式.....	501
27.2	简介.....	501
27.3	特性和模式.....	501
27.4	结构框图.....	502
27.5	OSC 信号说明.....	502
27.6	外部晶振/谐振器连接.....	503
27.7	外部时钟连接.....	504
27.8	存储器映射/寄存器定义.....	505
27.8.1	OSC 存储器映射/寄存器定义.....	505
27.9	功能说明.....	507
27.9.1	OSC 模块状态.....	507
27.9.2	OSC 模块模式.....	509
27.9.3	计数器.....	510
27.9.4	参考时钟的引脚要求.....	511
27.10	复位.....	511
27.11	低功耗模式下的操作.....	511
27.12	中断.....	511

第 28 章 RTC 振荡器(OSC32K)

28.1	简介.....	513
28.1.1	特性和模式.....	513
28.1.2	结构框图.....	513

小节编号	标题	页
28.2	RTC 信号说明.....	514
28.2.1	EXTAL32 — 振荡器输入.....	514
28.2.2	XTAL32 — 振荡器输出.....	514
28.3	外部晶振连接.....	515
28.4	存储器映射/寄存器说明.....	515
28.5	功能说明.....	515
28.6	复位概述.....	515
28.7	中断.....	516

第 29 章 Flash 存储器控制器(FMC)

29.1	简介.....	517
29.1.1	概述.....	517
29.1.2	特性.....	517
29.2	工作模式.....	518
29.3	外部信号说明.....	518
29.4	存储器映射和寄存器说明.....	518
29.4.1	Flash 访问保护寄存器 (FMC_PFAPR).....	523
29.4.2	Flash 存储阵列 0 控制寄存器 (FMC_PFB0CR).....	526
29.4.3	Reserved (FMC_Reserved).....	528
29.4.4	高速缓存标签存储器 (FMC_TAGVDW0Sn).....	529
29.4.5	高速缓存标签存储器 (FMC_TAGVDW1Sn).....	530
29.4.6	高速缓存标签存储器 (FMC_TAGVDW2Sn).....	531
29.4.7	高速缓存标签存储器 (FMC_TAGVDW3Sn).....	532
29.4.8	高速缓存数据存储器 (上字) (FMC_DATAW0SnU).....	532
29.4.9	高速缓存数据存储器 (下字) (FMC_DATAW0SnL).....	533
29.4.10	高速缓存数据存储器 (上字) (FMC_DATAW1SnU).....	533
29.4.11	高速缓存数据存储器 (下字) (FMC_DATAW1SnL).....	534
29.4.12	高速缓存数据存储器 (上字) (FMC_DATAW2SnU).....	534
29.4.13	高速缓存数据存储器 (下字) (FMC_DATAW2SnL).....	535

小节编号	标题	页
29.4.14	高速缓存数据存储 (上字) (FMC_DATAW3SnU).....	535
29.4.15	高速缓存数据存储 (下字) (FMC_DATAW3SnL).....	536
29.5	功能说明.....	536
29.5.1	默认配置.....	536
29.5.2	配置选项.....	536
29.5.3	预测式读取.....	537
29.5.4	Flash 访问控制(FAC)功能.....	538
29.6	初始化和应用信息.....	547

第 30 章 Flash 存储器模块(FTFA)

30.1	简介.....	549
30.1.1	特性.....	549
30.1.2	框图.....	550
30.1.3	术语表.....	550
30.2	外部信号说明.....	552
30.3	存储器映射和寄存器.....	552
30.3.1	Flash 配置字段说明.....	552
30.3.2	程序 Flash IFR 映射.....	552
30.3.3	寄存器说明.....	553
30.4	功能说明.....	566
30.4.1	Flash 保护.....	566
30.4.2	Flash 访问保护.....	567
30.4.3	中断.....	568
30.4.4	低功耗模式下的 Flash 操作.....	569
30.4.5	功能操作模式.....	569
30.4.6	Flash 读取和忽略的写入操作.....	570
30.4.7	写入时读取 (RWW)	570
30.4.8	Flash 程序和擦除.....	570
30.4.9	Flash 命令操作.....	570

小节编号	标题	页
30.4.10	裕量读取命令.....	574
30.4.11	Flash 命令说明.....	575
30.4.12	加密.....	590
30.4.13	复位序列.....	592

第 31 章 循环冗余校验(CRC)

31.1	简介.....	593
31.1.1	特性.....	593
31.1.2	结构框图.....	593
31.1.3	工作模式.....	594
31.2	存储器映射和寄存器说明.....	594
31.2.1	CRC 数据寄存器 (CRC_DATA).....	595
31.2.2	CRC 多项式寄存器 (CRC_GPOLY).....	596
31.2.3	CRC 控制寄存器 (CRC_CTRL).....	596
31.3	功能说明.....	597
31.3.1	CRC 初始化/重新初始化.....	597
31.3.2	CRC 计算.....	598
31.3.3	转置特性.....	599
31.3.4	CRC 结果补码.....	600

第 32 章 随机数生成加速器 (RNGA)

32.1	简介.....	603
32.1.1	概述.....	603
32.2	工作模式.....	604
32.2.1	进入正常模式.....	604
32.2.2	进入睡眠模式.....	604
32.3	存储器映射和寄存器定义.....	604
32.3.1	RNGA 控制寄存器 (RNG_CR).....	605
32.3.2	RNGA 状态寄存器 (RNG_SR).....	607

小节编号	标题	页
32.3.3	RNGA 熵寄存器 (RNG_ER).....	609
32.3.4	RNGA 输出寄存器 (RNG_OR).....	609
32.4	功能说明.....	610
32.4.1	输出(OR)寄存器.....	610
32.4.2	内核引擎/控制逻辑.....	610
32.5	初始化/应用信息.....	611

第 33 章 模数转换器 (ADC)

33.1	此模块的芯片实现细节.....	613
33.1.1	ADC 实例化信息.....	613
33.1.2	ADC 的 DMA 支持.....	613
33.1.3	ADCx 连接/通道分配.....	613
33.1.4	ADC 通道 MUX 选择.....	616
33.1.5	ADC 参考电压选项.....	616
33.1.6	VBAT 与 ADC 输入通道的连接.....	617
33.1.7	ADC 触发器.....	617
33.1.8	ADC 转换时钟选项.....	617
33.1.9	ADC 低功耗模式.....	618
33.2	简介.....	618
33.2.1	特性.....	619
33.2.2	结构框图.....	619
33.3	ADC 信号说明.....	621
33.3.1	模拟电源(VDDA).....	622
33.3.2	模拟接地(VSSA).....	622
33.3.3	参考电压选择.....	622
33.3.4	模拟通道输入(ADx).....	622
33.3.5	差分模拟通道输入(DADx).....	623
33.4	存储器映射和寄存器定义.....	623
33.4.1	ADC 状态和控制寄存器 1 (ADCx_SC1n).....	624

小节编号	标题	页
33.4.2	ADC 配置寄存器 1 (ADCx_CFG1).....	627
33.4.3	ADC 配置寄存器 2 (ADCx_CFG2).....	628
33.4.4	ADC 数据结果寄存器 (ADCx_Rn).....	629
33.4.5	比较值寄存器 (ADCx_CVn).....	630
33.4.6	状态和控制寄存器 2 (ADCx_SC2).....	631
33.4.7	状态和控制寄存器 3 (ADCx_SC3).....	633
33.4.8	ADC 偏移校正寄存器 (ADCx_OFS).....	634
33.4.9	ADC 同相端增益寄存器 (ADCx_PG).....	634
33.4.10	ADC 反相端增益寄存器 (ADCx_MG).....	635
33.4.11	ADC 同相端通用校准值寄存器 (ADCx_CLPD).....	636
33.4.12	ADC 同相端通用校准值寄存器 (ADCx_CLPS).....	636
33.4.13	ADC 同相端通用校准值寄存器 (ADCx_CLP4).....	637
33.4.14	ADC 同相端通用校准值寄存器 (ADCx_CLP3).....	637
33.4.15	ADC 同相端通用校准值寄存器 (ADCx_CLP2).....	638
33.4.16	ADC 同相端通用校准值寄存器 (ADCx_CLP1).....	638
33.4.17	ADC 同相端通用校准值寄存器 (ADCx_CLP0).....	639
33.4.18	ADC 反相端通用校准值寄存器 (ADCx_CLMD).....	639
33.4.19	ADC 反相端通用校准值寄存器 (ADCx_CLMS).....	640
33.4.20	ADC 反相端通用校准值寄存器 (ADCx_CLM4).....	640
33.4.21	ADC 反相端通用校准值寄存器 (ADCx_CLM3).....	641
33.4.22	ADC 反相端通用校准值寄存器 (ADCx_CLM2).....	641
33.4.23	ADC 反相端通用校准值寄存器 (ADCx_CLM1).....	642
33.4.24	ADC 反相端通用校准值寄存器 (ADCx_CLM0).....	642
33.5	功能说明.....	643
33.5.1	时钟选择和分频控制.....	643
33.5.2	参考电压选择.....	644
33.5.3	硬件触发和通道选择.....	644
33.5.4	转换控制.....	645
33.5.5	自动比较功能.....	652

小节编号	标题	页
33.5.6	校准功能.....	653
33.5.7	用户自定义偏移功能.....	654
33.5.8	温度传感器.....	655
33.5.9	MCU 等待模式下的操作.....	656
33.5.10	MCU 在普通停止模式下的操作.....	656
33.5.11	MCU 低功耗停止模式操作.....	657
33.6	初始化信息.....	657
33.6.1	ADC 模块初始化示例.....	657
33.7	应用信息.....	659
33.7.1	外部引脚和布线.....	659
33.7.2	误差来源.....	661

第 34 章 比较器 (CMP)

34.1	此模块的芯片实现细节.....	665
34.1.1	CMP 输入连接.....	665
34.1.2	CMP 外部参考.....	665
34.1.3	外部窗口/样本输入.....	665
34.1.4	CMP 触发模式.....	666
34.2	简介.....	666
34.2.1	CMP 特性.....	666
34.2.2	6 位 DAC 关键特性.....	667
34.2.3	ANMUX 关键特性.....	667
34.2.4	CMP、DAC 和 ANMUX 结构框图.....	668
34.2.5	CMP 框图.....	669
34.3	存储器映射/寄存器定义.....	671
34.3.1	CMP 控制寄存器 0 (CMPx_CR0).....	671
34.3.2	CMP 控制寄存器 1 (CMPx_CR1).....	672
34.3.3	CMP 滤波器周期寄存器 (CMPx_FPR).....	673
34.3.4	CMP 状态和控制寄存器 (CMPx_SCR).....	674

小节编号	标题	页
34.3.5	DAC 控制寄存器 (CMPx_DACCR).....	675
34.3.6	MUX 控制寄存器 (CMPx_MUXCR).....	675
34.4	功能说明.....	676
34.4.1	CMP 功能模式.....	676
34.4.2	电源模式.....	685
34.4.3	启动和运行.....	686
34.4.4	低通滤波器.....	686
34.5	CMP 中断.....	688
34.6	DMA 支持.....	689
34.7	CMP 异步 DMA 支持.....	689
34.8	数模转换器.....	690
34.9	DAC 功能说明.....	690
34.9.1	参考电压源选择.....	690
34.10	DAC 复位.....	691
34.11	DAC 时钟.....	691
34.12	DAC 中断.....	691

第 35 章 12 位数模转换器 (DAC)

35.1	此模块的芯片实现细节.....	693
35.1.1	12 位 DAC 概述.....	693
35.1.2	12 位 DAC 输出.....	693
35.1.3	12 位 DAC 参考.....	693
35.2	简介.....	693
35.3	特性.....	693
35.4	结构框图.....	694
35.5	存储器映射/寄存器定义.....	695
35.5.1	DAC 数据低电平寄存器 (DACx_DATnL).....	697
35.5.2	DAC 数据高电平寄存器 (DACx_DATnH).....	697
35.5.3	DAC 状态寄存器 (DACx_SR).....	698

小节编号	标题	页
35.5.4	DAC 控制寄存器 (DACx_C0).....	699
35.5.5	DAC 控制寄存器 1 (DACx_C1).....	700
35.5.6	DAC 控制寄存器 2 (DACx_C2).....	700
35.6	功能说明.....	701
35.6.1	DAC 数据缓冲器操作.....	701
35.6.2	DMA 操作.....	702
35.6.3	复位.....	702
35.6.4	低功耗工作模式.....	703

第 36 章 可编程延时模块 (PDB)

36.1	此模块的芯片实现细节.....	705
36.1.1	PDB 实例化.....	705
36.1.2	PDB 模块互连.....	706
36.1.3	背靠背应答连接.....	706
36.1.4	PDB 间隔触发器与 DAC 的连接.....	707
36.1.5	DAC 外部触发器输入连接.....	707
36.1.6	脉冲输出连接.....	707
36.1.7	脉冲输出使能寄存器的实现.....	707
36.2	简介.....	707
36.2.1	特性.....	707
36.2.2	实施方法.....	708
36.2.3	背靠背应答连接.....	709
36.2.4	DAC 外部触发器输入连接.....	709
36.2.5	功能框图.....	709
36.2.6	操作模式.....	711
36.3	存储器映射和寄存器定义.....	711
36.3.1	状态和控制寄存器 (PDBx_SC).....	712
36.3.2	系数值寄存器 (PDBx_MOD).....	714
36.3.3	计数器寄存器 (PDBx_CNT).....	715

小节编号	标题	页
36.3.4	中断延迟寄存器 (PDBx_IDLY).....	715
36.3.5	通道 n 控制寄存器 1 (PDBx_CHnC1).....	716
36.3.6	通道 n 状态寄存器 (PDBx_CHnS).....	717
36.3.7	通道 n 延时 0 寄存器 (PDBx_CHnDLY0).....	717
36.3.8	通道 n 延时 1 寄存器 (PDBx_CHnDLY1).....	718
36.3.9	DAC 间隔触发器 n 控制寄存器 (PDBx_DACINTCn).....	718
36.3.10	DAC 间隔 n 寄存器 (PDBx_DACINTn).....	719
36.3.11	脉冲输出 n 使能寄存器 (PDBx_POEN).....	720
36.3.12	脉冲输出 n 延迟寄存器 (PDBx_POnDLY).....	720
36.4	功能说明.....	721
36.4.1	PDB 预触发器和触发器输出.....	721
36.4.2	PDB 触发器输入源选择.....	722
36.4.3	脉冲输出.....	722
36.4.4	更新延迟寄存器.....	723
36.4.5	中断.....	725
36.4.6	DMA.....	725
36.5	应用信息.....	725
36.5.1	使用预分频器和乘法因数对时序分辨率的影响.....	725

第 37 章 定时器/PWM 模块 (TPM)

37.1	此模块的芯片实现细节.....	727
37.1.1	TPM 实例化信息.....	727
37.1.2	时钟选项.....	727
37.1.3	触发选项.....	728
37.1.4	全局时基.....	728
37.1.5	TPM 中断.....	728
37.2	简介.....	729
37.2.1	TPM 理念.....	729
37.2.2	特性.....	729

小节编号	标题	页
37.2.3	工作模式.....	730
37.2.4	结构框图.....	730
37.3	TPM 信号说明.....	731
37.3.1	TPM_EXTCLK — TPM 外部时钟.....	731
37.3.2	TPM_CHn — TPM 通道 (n) I/O 引脚.....	732
37.4	存储器映射和寄存器定义.....	732
37.4.1	版本 ID 寄存器 (TPMx_VERID).....	734
37.4.2	参数寄存器 (TPMx_PARAM).....	735
37.4.3	TPM 全局寄存器 (TPMx_GLOBAL).....	736
37.4.4	状态和控制 (TPMx_SC).....	737
37.4.5	计数器寄存器 (TPMx_CNT).....	738
37.4.6	模数寄存器 (TPMx_MOD).....	739
37.4.7	捕捉和比较状态 (TPMx_STATUS).....	739
37.4.8	通道 (n) 状态和控制 (TPMx_CnSC).....	741
37.4.9	通道 (n) 值寄存器 (TPMx_CnV).....	743
37.4.10	组合通道寄存器 (TPMx_COMBINE).....	744
37.4.11	通道触发 (TPMx_TRIG).....	746
37.4.12	通道极性 (TPMx_POL).....	747
37.4.13	滤波器控制 (TPMx_FILTER).....	748
37.4.14	正交解码器控制和状态 (TPMx_QDCTRL).....	749
37.4.15	配置寄存器 (TPMx_CONF).....	750
37.5	功能说明.....	752
37.5.1	时钟域.....	753
37.5.2	预分频器.....	753
37.5.3	计数器寄存器.....	754
37.5.4	输入捕捉模式.....	756
37.5.5	输出比较模式.....	757
37.5.6	边沿对齐 PWM (EPWM) 模式.....	758
37.5.7	中心对齐 PWM (CPWM) 模式.....	760

小节编号	标题	页
37.5.8	组合 PWM 模式.....	761
37.5.9	组合输入捕捉模式.....	765
37.5.10	输入捕捉滤波器.....	766
37.5.11	死区时间插入.....	767
37.5.12	正交解码器模式.....	768
37.5.13	通过写缓存更新的寄存器.....	771
37.5.14	DMA.....	772
37.5.15	输出触发器.....	772
37.5.16	复位概述.....	773
37.5.17	TPM 中断.....	773

第 38 章 周期性中断定时器 (PIT)

38.1	此模块的芯片实现细节.....	775
38.1.1	PIT/DMA 周期性触发器分配	775
38.1.2	PIT/ADC 触发器.....	775
38.2	简介.....	775
38.2.1	结构框图.....	775
38.2.2	特性.....	776
38.3	信号说明.....	777
38.4	存储器映射/寄存器说明.....	777
38.4.1	PIT 模块控制寄存器 (PIT_MCR).....	777
38.4.2	PIT 高位寿命定时器寄存器 (PIT_LTMR64H).....	779
38.4.3	PIT 低位寿命定时器寄存器 (PIT_LTMR64L).....	779
38.4.4	定时器加载值寄存器 (PIT_LDVAL _n).....	780
38.4.5	当前定时器值寄存器 (PIT_CVAL _n).....	780
38.4.6	定时器控制寄存器 (PIT_TCTRL _n).....	780
38.4.7	定时器标志寄存器 (PIT_TFLG _n).....	781
38.5	功能说明.....	782
38.5.1	常规操作.....	782

小节编号	标题	页
38.5.2	中断.....	783
38.5.3	链接定时器.....	784
38.6	初始化和应用信息.....	784
38.7	链接定时器配置示例.....	785
38.8	使用寿命定时器的示例配置.....	785

第 39 章 低功耗定时器 (LPTMR)

39.1	此模块的芯片实现细节.....	787
39.1.1	LPTMR 预分频器/毛刺滤波器时钟选项.....	787
39.1.2	LPTMR 脉冲计数器输入选项.....	787
39.2	简介.....	788
39.2.1	特性.....	788
39.2.2	操作模式.....	788
39.3	LPTMR 信号说明.....	789
39.3.1	详细信号说明.....	789
39.4	存储器映射和寄存器定义.....	789
39.4.1	低功耗定时器控制状态寄存器 (LPTMRx_CSR).....	790
39.4.2	低功耗定时器预分频寄存器 (LPTMRx_PSR).....	791
39.4.3	低功耗定时器比较寄存器 (LPTMRx_CMR).....	792
39.4.4	低功耗定时器计数器寄存器 (LPTMRx_CNR).....	793
39.5	功能说明.....	793
39.5.1	LPTMR 功耗和复位.....	793
39.5.2	LPTMR 时钟.....	793
39.5.3	LPTMR 预分频器/毛刺滤波器.....	794
39.5.4	LPTMR 比较.....	795
39.5.5	LPTMR 计数器.....	795
39.5.6	LPTMR 硬件触发.....	796
39.5.7	LPTMR 中断.....	796

第 40 章

实时时钟 (RTC)

40.1	此模块的芯片实现细节.....	797
40.1.1	RTC_CLKOUT 信号.....	797
40.2	简介.....	797
40.2.1	特性.....	797
40.2.2	操作模式.....	798
40.2.3	RTC 信号说明.....	798
40.3	寄存器定义.....	799
40.3.1	RTC 时间秒寄存器 (RTC_TSR).....	800
40.3.2	RTC 时间预分频器寄存器 (RTC_TPR).....	800
40.3.3	RTC 时间警报寄存器 (RTC_TAR).....	800
40.3.4	RTC 时间补偿寄存器 (RTC_TCR).....	801
40.3.5	RTC 控制寄存器 (RTC_CR).....	802
40.3.6	RTC 状态寄存器 (RTC_SR).....	804
40.3.7	RTC 锁定寄存器 (RTC_LR).....	805
40.3.8	RTC 中断使能寄存器 (RTC_IER).....	806
40.3.9	RTC 写入访问寄存器 (RTC_WAR).....	807
40.3.10	RTC 读取访问寄存器 (RTC_RAR).....	808
40.4	功能说明.....	810
40.4.1	功耗、时钟和复位.....	810
40.4.2	时间计数器.....	811
40.4.3	补偿.....	811
40.4.4	时间警报.....	812
40.4.5	更新模式.....	812
40.4.6	寄存器锁定.....	812
40.4.7	访问控制.....	812
40.4.8	中断.....	813

第 41 章

通用串行总线全速 OTG 控制器 (USBFSOTG)

小节编号	标题	页
41.1	此模块的芯片实现细节.....	815
41.1.1	通用串行总线 (USB) FS 子系统.....	815
41.2	简介.....	819
41.2.1	参考资料.....	819
41.2.2	USB—概述.....	820
41.2.3	USB OTG.....	821
41.2.4	USBFS 特性.....	822
41.3	功能说明.....	822
41.3.1	数据结构.....	822
41.3.2	需要外部组件的片上收发器.....	823
41.4	程序员接口.....	825
41.4.1	缓冲区描述符表.....	825
41.4.2	RX 与 TX 用作 USB 外设或 USB 主机.....	826
41.4.3	寻址 BDT 条目.....	827
41.4.4	缓冲区描述符 (BD)	828
41.4.5	USB 事务.....	830
41.5	存储器映射/寄存器定义.....	832
41.5.1	外设 ID 寄存器 (USB _x _PERID).....	834
41.5.2	外设 ID 补码寄存器 (USB _x _IDCOMP).....	835
41.5.3	外设修订版寄存器 (USB _x _REV).....	835
41.5.4	外设其他信息寄存器 (USB _x _ADDINFO).....	836
41.5.5	OTG 中断状态寄存器 (USB _x _OTGISTAT).....	836
41.5.6	OTG 中断控制寄存器 (USB _x _OTGICR).....	837
41.5.7	OTG 状态寄存器 (USB _x _OTGSTAT).....	838
41.5.8	OTG 控制寄存器 (USB _x _OTGCTL).....	839
41.5.9	中断状态寄存器 (USB _x _ISTAT).....	840
41.5.10	中断使能寄存器 (USB _x _INTEN).....	841
41.5.11	错误中断状态寄存器 (USB _x _ERRSTAT).....	842
41.5.12	错误中断使能寄存器 (USB _x _ERREN).....	843

小节编号	标题	页
41.5.13	状态寄存器 (USB _x _STAT).....	844
41.5.14	控制寄存器 (USB _x _CTL).....	845
41.5.15	地址寄存器 (USB _x _ADDR).....	846
41.5.16	BDT 页寄存器 1 (USB _x _BDTPAGE1).....	846
41.5.17	低帧号寄存器 (USB _x _FRMNUML).....	847
41.5.18	高帧号寄存器 (USB _x _FRMNUMH).....	847
41.5.19	令牌寄存器 (USB _x _TOKEN).....	848
41.5.20	SOF 阈值寄存器 (USB _x _SOFTHLD).....	848
41.5.21	BDT 页寄存器 2 (USB _x _BDTPAGE2).....	849
41.5.22	BDT 页寄存器 3 (USB _x _BDTPAGE3).....	850
41.5.23	端点控制寄存器 (USB _x _ENDPT _n).....	850
41.5.24	USB 控制寄存器 (USB _x _USBCTRL).....	851
41.5.25	USB OTG 观察寄存器 (USB _x _OBSERVE).....	852
41.5.26	USB OTG 控制寄存器 (USB _x _CONTROL).....	853
41.5.27	USB 收发器控制寄存器 0 (USB _x _USBTRC0).....	853
41.5.28	帧调整寄存器 (USB _x _USBFRMADJUST).....	855
41.5.29	其他控制寄存器 (USB _x _MISCCTRL).....	855
41.5.30	用于 IN 方向 7-0 端点的外设模式停止禁用 (USB _x _STALL_IL_DIS).....	856
41.5.31	用于 IN 方向 15-8 端点的外设模式停止禁用 (USB _x _STALL_IH_DIS).....	857
41.5.32	用于 OUT 方向 7-0 端点的外设模式停止禁用 (USB _x _STALL_OL_DIS).....	859
41.5.33	用于 OUT 方向 15-8 端点的外设模式停止禁用 (USB _x _STALL_OH_DIS).....	860
41.5.34	USB 时钟恢复控制 (USB _x _CLK_RECOVER_CTRL).....	861
41.5.35	IRC48M 振荡器使能寄存器 (USB _x _CLK_RECOVER_IRC_EN).....	862
41.5.36	时钟恢复组合中断使能 (USB _x _CLK_RECOVER_INT_EN).....	863
41.5.37	时钟恢复分离中断状态 (USB _x _CLK_RECOVER_INT_STATUS).....	864
41.6	OTG 和主机模式操作.....	864
41.7	主机模式操作示例.....	865
41.8	On-The-Go 操作.....	867
41.8.1	OTG 双重角色 A 器件操作.....	867

小节编号	标题	页
41.8.2	OTG 双重角色 B 器件操作.....	869
41.9	从设备模式 IRC48M 操作.....	870

第 42 章 CAN (FlexCAN)

42.1	此模块的芯片实现细节.....	871
42.1.1	FlexCAN 模块的数量.....	871
42.1.2	MDIS 位复位值.....	871
42.1.3	消息缓冲区数.....	871
42.1.4	FlexCAN 时钟.....	871
42.1.5	FlexCAN 中断.....	872
42.1.6	低功耗模式下的 FlexCAN 操作.....	872
42.1.7	FlexCAN 等待模式.....	872
42.1.8	FlexCAN 毛刺滤波器.....	872
42.2	简介.....	873
42.2.1	概述.....	874
42.2.2	FlexCAN 模块特性.....	875
42.2.3	工作模式.....	876
42.3	FlexCAN 信号说明.....	878
42.3.1	CAN Rx.....	878
42.3.2	CAN Tx.....	878
42.4	存储器映射/寄存器定义.....	878
42.4.1	FlexCAN 存储器映射.....	878
42.4.2	模块配置寄存器 (CANx_MCR).....	883
42.4.3	控制 1 寄存器 (CANx_CTRL1).....	887
42.4.4	自由运行定时器 (CANx_TIMER).....	891
42.4.5	Rx 邮箱全局掩码寄存器 (CANx_RXMGMASK).....	892
42.4.6	Rx 14 掩码寄存器 (CANx_RX14MASK).....	893
42.4.7	Rx 15 掩码寄存器 (CANx_RX15MASK).....	894
42.4.8	错误计数器 (CANx_ECR).....	894

小节编号	标题	页
42.4.9	错误和状态 1 寄存器 (CAN _x _ESR1).....	896
42.4.10	中断掩码 1 寄存器 (CAN _x _IMASK1).....	902
42.4.11	中断标志 1 寄存器 (CAN _x _IFLAG1).....	902
42.4.12	控制 2 寄存器 (CAN _x _CTRL2).....	904
42.4.13	错误和状态 2 寄存器 (CAN _x _ESR2).....	908
42.4.14	CRC 寄存器 (CAN _x _CRCR).....	909
42.4.15	Rx FIFO 全局掩码寄存器 (CAN _x _RXFGMASK).....	909
42.4.16	Rx FIFO 信息寄存器 (CAN _x _RXFIR).....	910
42.4.17	CAN 位时序寄存器 (CAN _x _CBT).....	911
42.4.18	Rx 单独掩码寄存器 (CAN _x _RXIMR _n).....	912
42.4.53	消息缓冲器结构.....	913
42.4.54	Rx FIFO 结构.....	918
42.5	功能说明.....	920
42.5.1	传输流程.....	920
42.5.2	仲裁过程.....	921
42.5.3	接收过程.....	924
42.5.4	匹配过程.....	926
42.5.5	移动过程.....	930
42.5.6	数据一致性.....	931
42.5.7	Rx FIFO.....	934
42.5.8	CAN 协议相关特性.....	936
42.5.9	时钟域和限制.....	944
42.5.10	工作模式详情.....	944
42.5.11	中断.....	949
42.5.12	总线接口.....	949
42.6	初始化/应用信息.....	950
42.6.1	FlexCAN 初始化序列.....	950

第 43 章 串行外设接口 (SPI)

小节编号	标题	页
43.1	此模块的芯片实现细节.....	953
43.1.1	SPI 模块配置.....	953
43.1.2	SPI 时钟.....	953
43.1.3	CTAR 的数目.....	953
43.1.4	TX FIFO 大小.....	953
43.1.5	RX FIFO 大小.....	954
43.1.6	PCS 信号数量.....	954
43.1.7	低功耗模式下的 SPI 操作.....	954
43.1.8	SPI 等待模式.....	955
43.1.9	SPI 中断.....	955
43.1.10	SPI 时钟.....	955
43.2	简介.....	955
43.2.1	结构框图.....	955
43.2.2	特性.....	956
43.2.3	接口配置.....	958
43.2.4	工作模式.....	958
43.3	模块信号说明.....	960
43.3.1	PCS0/SS—外设片选/从机选择.....	960
43.3.2	PCS1–PCS3—外设片选 1–3.....	960
43.3.3	PCS4—外设片选 4.....	960
43.3.4	PCS5/PCSS—外设片选 5 /外设片选选通.....	960
43.3.5	SCK—串行时钟.....	961
43.3.6	SIN—串行输入.....	961
43.3.7	SOUT—串行输出.....	961
43.4	存储器映射/寄存器定义.....	961
43.4.1	模块配置寄存器 (SPIx_MCR).....	963
43.4.2	传输计数寄存器 (SPIx_TCR).....	966
43.4.3	时钟和传输属性寄存器 (主机模式下) (SPIx_CTAR _n).....	966
43.4.4	时钟和传输属性寄存器 (从机模式下) (SPIx_CTAR _n _SLAVE).....	971

小节编号	标题	页
43.4.5	状态寄存器 (SPLx_SR).....	972
43.4.6	DMA/中断请求选择和使能寄存器 (SPLx_RSER).....	974
43.4.7	主机模式下的 PUSH TX FIFO 寄存器 (SPLx_PUSHR).....	976
43.4.8	从机模式下的 PUSH TX FIFO 寄存器 (SPLx_PUSHR_SLAVE).....	978
43.4.9	POP RX FIFO 寄存器 (SPLx_POPR).....	978
43.4.10	发送 FIFO 寄存器 (SPLx_TXFRn).....	979
43.4.11	接收 FIFO 寄存器 (SPLx_RXFRn).....	979
43.5	功能说明.....	980
43.5.1	模块传输的开始和停止.....	980
43.5.2	串行外设接口(SPI)配置.....	981
43.5.3	模块波特率和时钟延迟生成.....	984
43.5.4	传输格式.....	987
43.5.5	连续串行通信时钟.....	996
43.5.6	从机模式操作限制.....	998
43.5.7	中断/DMA 请求.....	998
43.5.8	节电特性.....	1000
43.6	初始化/应用信息.....	1001
43.6.1	如何管理队列.....	1001
43.6.2	切换主机和从机模式.....	1002
43.6.3	在主机/从机模式下初始化模块.....	1002
43.6.4	波特率设置.....	1002
43.6.5	延迟设置.....	1003
43.6.6	FIFO 指针地址的计算.....	1004

第 44 章 低功耗 I2C (LPI2C)

44.1	此模块的芯片实现细节.....	1007
44.1.1	LPI2C 实例化信息.....	1007
44.2	简介.....	1007
44.2.1	概述.....	1007

小节编号	标题	页
44.2.2	特性.....	1008
44.2.3	框图.....	1009
44.2.4	操作模式.....	1009
44.2.5	信号说明.....	1010
44.3	存储器映射和寄存器.....	1010
44.3.1	版本 ID 寄存器 (LPI2Cx_VERID).....	1013
44.3.2	参数寄存器 (LPI2Cx_PARAM).....	1013
44.3.3	主机控制寄存器 (LPI2Cx_MCR).....	1014
44.3.4	主机状态寄存器 (LPI2Cx_MSR).....	1015
44.3.5	主机中断使能寄存器 (LPI2Cx_MIER).....	1017
44.3.6	主机 DMA 使能寄存器 (LPI2Cx_MDER).....	1019
44.3.7	主机配置寄存器 0 (LPI2Cx_MCFGR0).....	1020
44.3.8	主机配置寄存器 1 (LPI2Cx_MCFGR1).....	1021
44.3.9	主机配置寄存器 2 (LPI2Cx_MCFGR2).....	1023
44.3.10	主机配置寄存器 3 (LPI2Cx_MCFGR3).....	1023
44.3.11	主机数据匹配寄存器 (LPI2Cx_MDMR).....	1024
44.3.12	主时钟配置寄存器 0 (LPI2Cx_MCCR0).....	1024
44.3.13	主时钟配置寄存器 1 (LPI2Cx_MCCR1).....	1025
44.3.14	主机 FIFO 控制寄存器 (LPI2Cx_MFCR).....	1026
44.3.15	主机 FIFO 状态寄存器 (LPI2Cx_MFSR).....	1027
44.3.16	主机数据发送寄存器 (LPI2Cx_MTDR).....	1027
44.3.17	主机数据接收寄存器 (LPI2Cx_MRDR).....	1028
44.3.18	从机控制寄存器 (LPI2Cx_SCR).....	1029
44.3.19	从机状态寄存器 (LPI2Cx_SSR).....	1031
44.3.20	从机中断使能寄存器 (LPI2Cx_SIER).....	1033
44.3.21	从机 DMA 使能寄存器 (LPI2Cx_SDER).....	1035
44.3.22	从机配置寄存器 1 (LPI2Cx_SCFGR1).....	1036
44.3.23	从机配置寄存器 2 (LPI2Cx_SCFGR2).....	1038
44.3.24	从机地址匹配寄存器 (LPI2Cx_SAMR).....	1039

小节编号	标题	页
44.3.25	从机地址状态寄存器 (LPI2Cx_SASR).....	1040
44.3.26	从机 ACK 发送寄存器 (LPI2Cx_STAR).....	1040
44.3.27	从机数据发送寄存器 (LPI2Cx_STDR).....	1041
44.3.28	从机数据接收寄存器 (LPI2Cx_SRDR).....	1042
44.4	功能说明.....	1043
44.4.1	时钟和复位.....	1043
44.4.2	主机模式.....	1044
44.4.3	从机模式.....	1048
44.4.4	中断和 DMA 请求.....	1050
44.4.5	外设触发信号.....	1052

第 45 章 通用异步接收器/发送器 (UART)

45.1	此模块的芯片实现细节.....	1055
45.1.1	UART 配置信息.....	1055
45.1.2	UART 唤醒.....	1055
45.1.3	UART 中断.....	1056
45.2	简介.....	1057
45.2.1	特性.....	1057
45.2.2	工作模式.....	1059
45.3	UART 信号说明.....	1060
45.3.1	详细信号说明.....	1060
45.4	存储器映射和寄存器.....	1061
45.4.1	UART 波特率寄存器: 高位 (UARTx_BDH).....	1066
45.4.2	UART 波特率寄存器: 低位 (UARTx_BDL).....	1067
45.4.3	UART 控制寄存器 1 (UARTx_C1).....	1067
45.4.4	UART 控制寄存器 2 (UARTx_C2).....	1069
45.4.5	UART 状态寄存器 1 (UARTx_S1).....	1070
45.4.6	UART 状态寄存器 2 (UARTx_S2).....	1073
45.4.7	UART 控制寄存器 3 (UARTx_C3).....	1075

小节编号	标题	页
45.4.8	UART 数据寄存器 (UARTx_D).....	1076
45.4.9	UART 匹配地址寄存器 1 (UARTx_MA1).....	1077
45.4.10	UART 匹配地址寄存器 2 (UARTx_MA2).....	1077
45.4.11	UART 控制寄存器 4 (UARTx_C4).....	1078
45.4.12	UART 控制寄存器 5 (UARTx_C5).....	1078
45.4.13	UART 扩展数据寄存器 (UARTx_ED).....	1079
45.4.14	UART 调制解调器寄存器 (UARTx_MODEM).....	1080
45.4.15	UART 红外寄存器 (UARTx_IR).....	1081
45.4.16	UART FIFO 参数 (UARTx_PFIFO).....	1082
45.4.17	UART FIFO 控制寄存器 (UARTx_CFIFO).....	1083
45.4.18	UART FIFO 状态寄存器 (UARTx_SFIFO).....	1084
45.4.19	UART FIFO 发送水印 (UARTx_TWFIFO).....	1085
45.4.20	UART FIFO 发送计数 (UARTx_TCFIFO).....	1086
45.4.21	UART FIFO 接收水印 (UARTx_RWFIFO).....	1086
45.4.22	UART FIFO 接收计数 (UARTx_RCFIFO).....	1087
45.4.23	UART 7816 控制寄存器 (UARTx_C7816).....	1087
45.4.24	UART 7816 中断使能寄存器 (UARTx_IE7816).....	1089
45.4.25	UART 7816 中断状态寄存器 (UARTx_IS7816).....	1090
45.4.26	UART 7816 等待参数寄存器 (UARTx_WP7816).....	1091
45.4.27	UART 7816 等待 N 寄存器 (UARTx_WN7816).....	1092
45.4.28	UART 7816 等待 FD 寄存器 (UARTx_WF7816).....	1092
45.4.29	UART 7816 错误阈值寄存器 (UARTx_ET7816).....	1093
45.4.30	UART 7816 发送长度寄存器 (UARTx_TL7816).....	1093
45.4.31	UART 7816 ATR 持续定时器寄存器 A (UARTx_AP7816A_T0).....	1094
45.4.32	UART 7816 ATR 持续定时器寄存器 B (UARTx_AP7816B_T0).....	1094
45.4.33	UART 7816 等待参数寄存器 A (UARTx_WP7816A_T0).....	1095
45.4.34	UART 7816 等待参数寄存器 A (UARTx_WP7816A_T1).....	1095
45.4.35	UART 7816 等待参数寄存器 B (UARTx_WP7816B_T0).....	1096
45.4.36	UART 7816 等待参数寄存器 B (UARTx_WP7816B_T1).....	1096

小节编号	标题	页
45.4.37	UART 7816 等待和保护参数寄存器 (UART _x _WGP7816_T1).....	1097
45.4.38	UART 7816 等待参数寄存器 C (UART _x _WP7816C_T1).....	1097
45.5	功能说明.....	1098
45.5.1	发送器.....	1098
45.5.2	接收器.....	1103
45.5.3	波特率生成.....	1116
45.5.4	数据格式 (非 ISO-7816)	1118
45.5.5	单线操作.....	1121
45.5.6	循环操作.....	1122
45.5.7	ISO-7816/智能卡支持.....	1122
45.5.8	红外接口.....	1127
45.6	复位.....	1128
45.7	系统级中断源.....	1128
45.7.1	RXEDGIF 说明.....	1129
45.8	DMA 操作.....	1129
45.9	应用信息.....	1130
45.9.1	发送/接收数据缓冲区操作.....	1130
45.9.2	ISO-7816 初始化序列.....	1130
45.9.3	初始化序列 (非 ISO-7816)	1132
45.9.4	溢出(OR)标志位含义.....	1133
45.9.5	溢出 NACK 考虑因素.....	1134
45.9.6	匹配地址寄存器.....	1134
45.9.7	调制解调器的特性.....	1134
45.9.8	IrDA 最小脉宽.....	1135
45.9.9	正在清除 7816 等待定时器 (WT、BWT、CWT) 中断.....	1136
45.9.10	旧版和反向兼容性考虑因素.....	1136

第 46 章 低功耗通用异步接收器/发送器 0 (LPUART0)

46.1	此模块的芯片实现细节.....	1139
------	-----------------	------

小节编号	标题	页
46.1.1	LPUART0 概述.....	1139
46.2	简介.....	1139
46.2.1	特性.....	1139
46.2.2	操作模式.....	1140
46.2.3	信号说明.....	1141
46.2.4	功能框图.....	1141
46.3	寄存器定义.....	1143
46.3.1	LPUART 波特率寄存器 (LPUARTx_BAUD).....	1144
46.3.2	LPUART 状态寄存器 (LPUARTx_STAT).....	1146
46.3.3	LPUART 控制寄存器 (LPUARTx_CTRL).....	1150
46.3.4	LPUART 数据寄存器 (LPUARTx_DATA).....	1154
46.3.5	LPUART 匹配地址寄存器 (LPUARTx_MATCH).....	1156
46.3.6	LPUART 调制解调器 IrDA 寄存器 (LPUARTx_MODIR).....	1157
46.4	功能说明.....	1158
46.4.1	生成波特率.....	1158
46.4.2	传送器功能说明.....	1159
46.4.3	接收器功能说明.....	1161
46.4.4	其他 LPUART 功能.....	1167
46.4.5	红外接口.....	1169
46.4.6	中断和状态标志.....	1170

第 47 章 FlexIO

47.1	此模块的芯片实现细节.....	1173
47.1.1	FlexIO 在本器件上的配置.....	1173
47.1.2	FlexIO 触发器选项.....	1173
47.2	简介.....	1174
47.2.1	概述.....	1174
47.2.2	特性.....	1174
47.2.3	结构框图.....	1175

小节编号	标题	页
47.2.4	操作模式.....	1175
47.2.5	FlexIO 信号说明.....	1176
47.3	存储器映射和寄存器.....	1176
47.3.1	版本 ID 寄存器 (FLEXIO_VERID).....	1178
47.3.2	参数寄存器 (FLEXIO_PARAM).....	1179
47.3.3	FlexIO 控制寄存器 (FLEXIO_CTRL).....	1180
47.3.4	引脚状态寄存器 (FLEXIO_PIN).....	1181
47.3.5	移位器状态寄存器 (FLEXIO_SHIFTSTAT).....	1182
47.3.6	移位器错误寄存器 (FLEXIO_SHIFTERR).....	1182
47.3.7	定时器状态寄存器 (FLEXIO_TIMSTAT).....	1183
47.3.8	移位器状态中断使能 (FLEXIO_SHIFTSIEN).....	1184
47.3.9	移位器错误中断使能 (FLEXIO_SHIFTEIEN).....	1184
47.3.10	定时器中断使能寄存器 (FLEXIO_TIMIEN).....	1185
47.3.11	移位器状态 DMA 使能 (FLEXIO_SHIFTSDEN).....	1185
47.3.12	移位器控制 N 寄存器 (FLEXIO_SHIFTCTL _n).....	1186
47.3.13	移位器配置 N 寄存器 (FLEXIO_SHIFTCFG _n).....	1188
47.3.14	移位器缓冲器 N 寄存器 (FLEXIO_SHIFTBUF _n).....	1189
47.3.15	移位器缓冲器 N 位交换寄存器 (FLEXIO_SHIFTBUFBIS _n).....	1190
47.3.16	移位器缓冲器 N 字节交换寄存器 (FLEXIO_SHIFTBUFBYS _n).....	1190
47.3.17	移位器缓冲器 N 位字节交换寄存器 (FLEXIO_SHIFTBUFBBS _n).....	1190
47.3.18	定时器控制 N 寄存器 (FLEXIO_TIMCTL _n).....	1191
47.3.19	定时器配置 N 寄存器 (FLEXIO_TIMCFG _n).....	1193
47.3.20	定时器比较 N 寄存器 (FLEXIO_TIMCMP _n).....	1195
47.4	功能说明.....	1196
47.4.1	移位器操作.....	1196
47.4.2	定时器操作.....	1198
47.4.3	引脚操作.....	1200
47.5	应用信息.....	1200
47.5.1	UART 发送.....	1201

小节编号	标题	页
47.5.2	UART 接收.....	1201
47.5.3	SPI 主机.....	1203
47.5.4	SPI 从机.....	1205
47.5.5	I2C 主机.....	1206
47.5.6	I2S 主机.....	1208
47.5.7	I2S 从机.....	1209

第 48 章 集成芯片间声音 (I2S) /同步音频接口 (SAI)

48.1	此模块的芯片实现细节.....	1211
48.1.1	实例化信息.....	1211
48.1.2	I2S/SAI 时钟.....	1211
48.1.3	I2S/SAI 在低功耗模式下的操作.....	1213
48.2	简介.....	1214
48.2.1	特性.....	1214
48.2.2	结构框图.....	1214
48.2.3	工作模式.....	1215
48.3	外部信号.....	1216
48.4	存储器映射和寄存器定义.....	1216
48.4.1	SAI 发送控制寄存器 (I2Sx_TCSR).....	1219
48.4.2	SAI 发送配置 1 寄存器 (I2Sx_TCR1).....	1223
48.4.3	SAI 发送配置 2 寄存器 (I2Sx_TCR2).....	1223
48.4.4	SAI 发送配置 3 寄存器 (I2Sx_TCR3).....	1225
48.4.5	SAI 发送配置 4 寄存器 (I2Sx_TCR4).....	1226
48.4.6	SAI 发送配置 5 寄存器 (I2Sx_TCR5).....	1228
48.4.7	SAI 发送数据寄存器 (I2Sx_TDRn).....	1229
48.4.8	SAI 发送 FIFO 寄存器 (I2Sx_TFRn).....	1229
48.4.9	SAI 发送掩码寄存器 (I2Sx_TMR).....	1230
48.4.10	SAI 接收控制寄存器 (I2Sx_RCSR).....	1231
48.4.11	SAI 接收配置 1 寄存器 (I2Sx_RCR1).....	1235

小节编号	标题	页
48.4.12	SAI 接收配置 2 寄存器 (I2Sx_RCR2).....	1235
48.4.13	SAI 接收配置 3 寄存器 (I2Sx_RCR3).....	1237
48.4.14	SAI 接收配置 4 寄存器 (I2Sx_RCR4).....	1238
48.4.15	SAI 接收配置 5 寄存器 (I2Sx_RCR5).....	1240
48.4.16	SAI 接收数据寄存器 (I2Sx_RDRn).....	1241
48.4.17	SAI 接收 FIFO 寄存器 (I2Sx_RFRn).....	1241
48.4.18	SAI 接收掩码寄存器 (I2Sx_RMR).....	1242
48.4.19	SAI MCLK 控制寄存器 (I2Sx_MCR).....	1243
48.4.20	SAI MCLK 分频寄存器 (I2Sx_MDR).....	1244
48.5	功能说明.....	1244
48.5.1	SAI 时钟.....	1244
48.5.2	SAI 复位.....	1246
48.5.3	同步模式.....	1247
48.5.4	帧同步配置.....	1247
48.5.5	数据 FIFO.....	1248
48.5.6	字掩码寄存器.....	1250
48.5.7	中断和 DMA 请求.....	1250

第 49 章 通用输入/输出 (GPIO)

49.1	此模块的芯片实现细节.....	1253
49.1.1	GPIO 信号数量.....	1253
49.2	简介.....	1253
49.2.1	特性.....	1253
49.2.2	工作模式.....	1254
49.2.3	GPIO 信号说明.....	1254
49.3	存储器映射和寄存器定义.....	1255
49.3.1	端口数据输出寄存器 (GPIOx_PDOR).....	1256
49.3.2	端口置位输出寄存器 (GPIOx_PSOR).....	1257
49.3.3	端口清零输出寄存器 (GPIOx_PCOR).....	1257

小节编号	标题	页
49.3.4	端口翻转输出寄存器 (GPIOx_PTOR).....	1258
49.3.5	端口数据输入寄存器 (GPIOx_PDIR).....	1258
49.3.6	端口数据方向寄存器 (GPIOx_PDDR).....	1259
49.4	功能说明.....	1259
49.4.1	通用输入.....	1259
49.4.2	通用输出.....	1259

第 50 章 JTAG 控制器(JTAGC)

50.1	简介.....	1261
50.1.1	结构框图.....	1261
50.1.2	特性.....	1261
50.1.3	操作模式.....	1262
50.2	外部信号说明.....	1263
50.2.1	TCK—测试时钟输入.....	1263
50.2.2	TDI—测试数据输入.....	1263
50.2.3	TDO—测试数据输出.....	1263
50.2.4	TMS—测试模式选择.....	1264
50.3	寄存器说明.....	1264
50.3.1	指令寄存器.....	1264
50.3.2	旁通寄存器.....	1264
50.3.3	器件标识寄存器.....	1265
50.3.4	边界扫描寄存器.....	1265
50.4	功能说明.....	1266
50.4.1	JTAGC 复位配置.....	1266
50.4.2	IEEE 1149.1-2001 (JTAG)测试接入端口.....	1266
50.4.3	TAP 控制器状态机.....	1266
50.4.4	JTAGC 数据块指令.....	1268
50.4.5	边界扫描.....	1270
50.5	初始化/应用信息.....	1271



第 1 章 关于本手册

1.1 读者

本参考手册适用于想要通过此器件开发产品的系统软件和硬件开发人员 and 应用程序编程人员。它假定读者理解操作系统、微处理器系统设计以及软件和硬件的基本原理。

1.2 组织

本手册包含两组主要的章节。

1. 第一组章节中包含适用于此芯片上所有组件的信息。
2. 第二组章节分成不同的功能组，详述了特定的功能区域。
 - 这些功能组的示例包括时钟、定时器和通信接口。
 - 每组包括用于提供单个模块技术描述的章节。

1.3 模块说明

每个模块章节具有两个主要部分：

- 芯片特性信息：第一部分，芯片特性 [模块名称] 信息，包含芯片上模块实例的数量和模块实例间可能存在的实施上的不同，如 FIFO 深度或者支持的通道数量上的不同。这部分信息也可能包含此模块与其它模块之间的功能连接。请首先阅读此部分，因为此内容对于理解此章节的其他部分非常重要。
- 一般模块信息：后续部分提供了与模块相关的一般信息，包括其信号、寄存器和功能说明。

注

如果芯片特性信息（第一部分）与一般模块信息（后续部分）有冲突，芯片特性信息部分有优先级。

第49章 增强型串行通信接口(eSCI)

49.1 芯片特定eSCI信息

此芯片具有六个eSCI模块实例。部分特性因实例不同而不同。下表汇总了这些特性差异。此表未列出实例共有的特性详细信息。

表49-1. eSCI实例特性差异

实例	DMA支持
eSCI_A 和 eSCI_B	是
eSCI_C、eSCI_D、eSCI_E和eSCI_F	注：eSCI DMA功能描述不适用于这些实例

注释

对于eSCI_D，单线特性不适用于通过PCSA3的TX/RX，
因此此管脚仅用作输出。

49.2 简介

eSCI模块是一个具有LIN主机接口层和DMA支持的SCI模块。LIN主机层符合LIN 1.3、LIN 2.0、LIN2.1和SAE J2602/1规范。

49.2.1 参考文献

- LIN规范包修订版1.3，2002年12月12日
- LIN规范包修订版2.0，2003年9月23日

样本参考手册

NXP Semiconductors
2633

应首先读取的
芯片特性信息

一般模块信息的
开始

图 1-1. 示例：芯片特性信息和一般模块信息章节

1.3.1 示例：用于阐明相同章节中的内容的芯片特性信息

以下示例介绍了芯片特性信息，用于阐明本章节中后面展示的一般模块信息。在此示例中，芯片特性寄存器复位值替代寄存器图中出现的复位值。

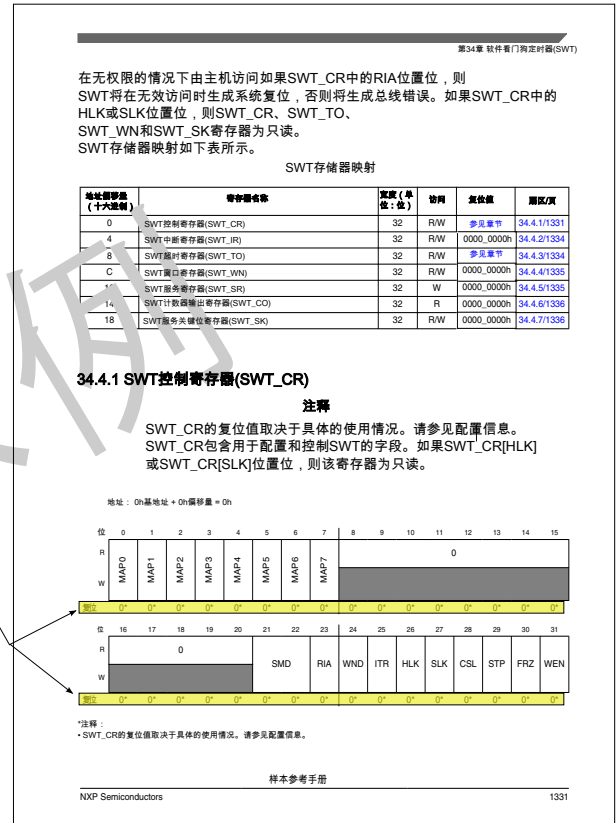
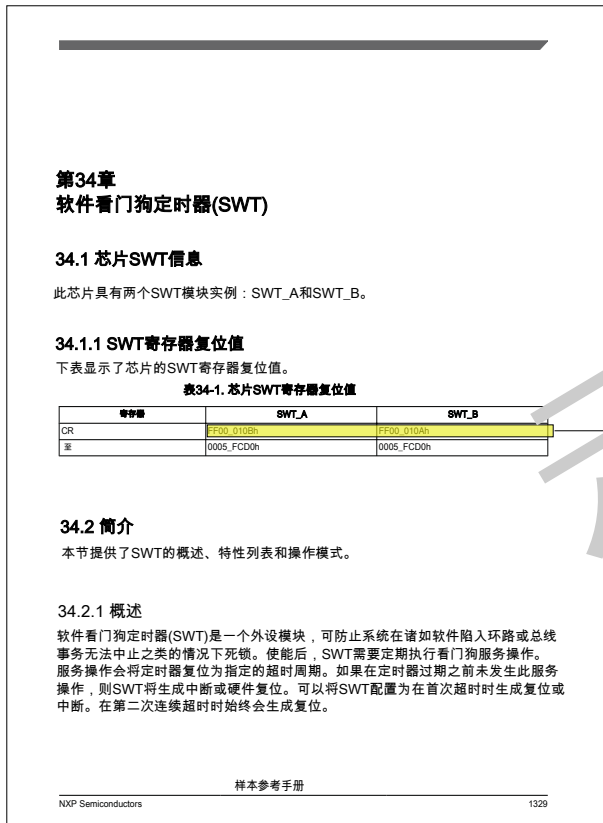


图 1-2. 示例：用于阐明相同章节中的内容的芯片特性信息

1.3.2 示例：参考不同章节的芯片特性信息

以下芯片特性信息是指其他章节的芯片特性信息。在此示例中，请在继续阅读本章节其他内容之前先阅读芯片特性信息组。

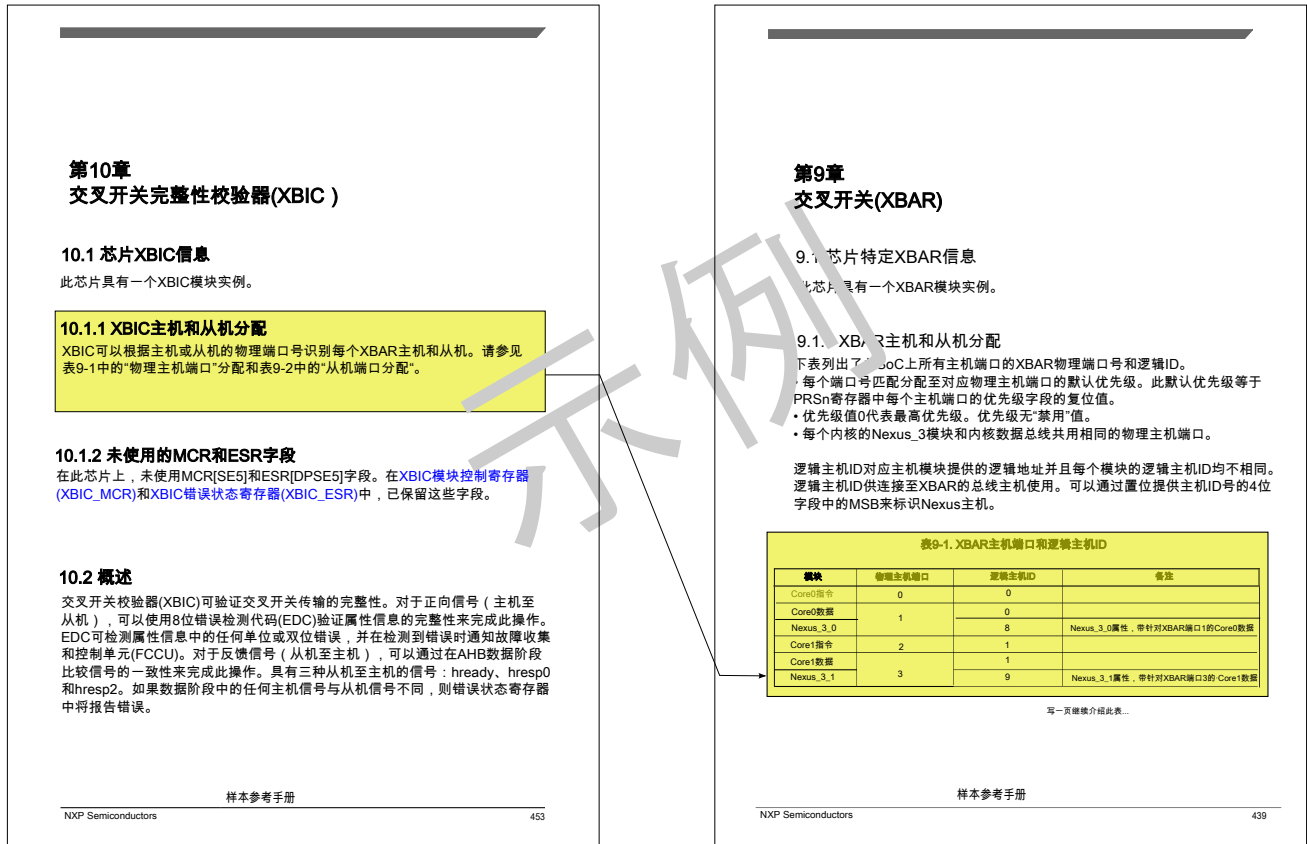


图 1-3. 示例：参考不同章节的芯片特性信息

1.4 寄存器说明

模块章节介绍了寄存器信息：

- 存储器映射，包括：
 - 地址
 - 每个寄存器的名称和首字母缩略词/缩写词
 - 每个寄存器的宽度（位）
 - 每个寄存器的复位值
 - 所描述的每个寄存器的页码
- 寄存器图
- 字段描述表
- 相关文本

下列寄存器图介绍了本手册使用字段结构的约定。

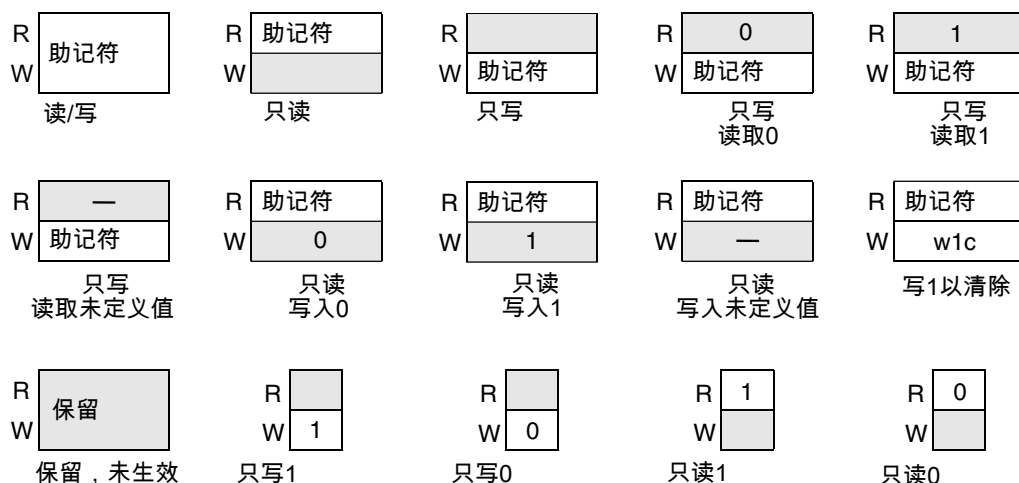


图 1-4. 寄存器图惯例

1.5 约定

1.5.1 编码系统

通过以下后缀可识别不同的编码系统：

后缀	用来识别
b	二进制数。例如，数字 5 的二进制数写为 101b。在某些情况下，二进制数冠以前缀 0b。
d	十进制数。当有可能出现混淆时才为十进制数加后缀。通常，十进制数没有后缀。
h	十六进制数。例如，数字 60 的十六进制数写为 3Ch。在某些情况下，十六进制数冠以前缀 0x。

1.5.2 印刷符号

以下印刷符号广泛应用于本文档：

示例	说明
占位符, x	斜体项目是用户所提供的信息的占位符。斜体文字也可以用于表示出版物或重点的标题。无格式的小写字母还用作单个字母和数字的占位符。
代码	固定宽度类型指的是文本必须严格按照所示格式输入。它可以用于指令助记符、指令符、符号、子命令、参数和运算符。固定宽度类型也用于示例代码。文本和表格中的指令助记符和指令符必须全部大写（例如，BSR）。
SR[SCM]	方括号里的内容表示寄存器的某一字段。此示例指的是状态寄存器(SR)中的缩放模式(SCM)字段。
REVNO[6:4]和 XAD[7:0]	数字放在方括号中，数字之间用冒号隔开可代表以下两种意思：

示例	说明
	<ul style="list-style-type: none"> 寄存器字段的子集 例如，REVNO[6:4]指 4-6 位，是在 REVNO 寄存器中占据 0-6 位的 COREREV 字段的一部分。 独立总线信号的连续范围 例如，XAD[7:0]指的是 XAD 总线的信 0-7。

1.5.3 特殊术语

以下术语具有特殊含义：

术语	含义
有效	指的是以下信号状态： <ul style="list-style-type: none"> 为 1: 引脚状态为高电平； 为 0: 引脚状态为低电平。
无效	指的是以下信号状态： <ul style="list-style-type: none"> 为 1: 引脚状态为低电平； 为 0: 引脚状态为高电平。 <p>在某些情况下，也称为取反操作。</p>
保留	请参阅存储器空间、寄存器、字段或编程设置。写入保留的位置可能会导致无法预测的功能或行为。 <ul style="list-style-type: none"> 请勿修改保留编程设置的默认值，例如保留寄存器字段的复位值。 请考虑存储器中要保留的未定义位置。
w1c	写入 1 以清零：是指必须写入 1 以“清零”的寄存器位域。

第 2 章 简介

2.1 概述

本章对本文档所涵盖器件上的可用模块进行了概要说明。

2.2 Kinetis KS 系列特性汇总

下表列出了 KS 系列 上集成的特性。

表 2-1. 特性汇总

特性	器件概述
硬件特性	
封装	64-pin LQFP (10 x 10 mm) 100-pin LQFP (14 x 14 mm)
电压范围	1.71 V 至 3.6 V
温度范围 (T _A)	-40°C 至 105°C
温度范围 (T _J)	-40°C 至 125°C
系统	
中央处理器 (CPU)	ARM Cortex M4
最大值 CPU 频率	高速运转/过驱时为 120 MHz, 正常运转时为 80 MHz
存储器保护单元 (MPU)	否
高速缓存	否
中断控制器	是
直接存储器访问 (DMA)	16 通道
DMA 请求多路复用器	是
不可屏蔽中断 (NMI)	是
软件看门狗	是
硬件看门狗	是
低漏电唤醒单元	外部监控器引脚 16 外部唤醒引脚, 带数字毛刺滤波器

下一页继续介绍此表...

表 2-1. 特性汇总 (继续)

特性	器件概述
	和 4 内部唤醒源 在 LLS 和 VLLS 模式下, RESET 引脚始终被用作复位唤醒
随机数生成器	是
硬件加密	否
调试	2 引脚串行线调试 (SWD) IEEE 1149.1 联合测试访问组 (JTAG) IEEE 1149.7 紧凑型 JTAG (cJTAG)
跟踪	跟踪端口接口单元 (TPIU) Flash 块和断电 (FPB) 数据观察点和跟踪 (DWT) 仪器跟踪宏单元 (ITM)
边界扫描	是
唯一标识 (ID) 号	128 位宽
篡改检测	否
安全密钥存储	否
存储器	
Flash 总量	256 KB
程序 Flash 存储器	256 KB
FlexMemory	否
Flash 缓存	4w x 8s x 64 位 = 256 字节
随机存取存储器 (RAM)	64 KB
外部 SDRAM 控制器	否
外部 NAND Flash 控制器	否
低漏电待机存储器	16 KB (在 LLS2 和 VLLS2 模式下) 在 VBAT 域中为 32 字节 在所有 VLLS 模式下的主电源域中为 32 字节
循环冗余校验 (CRC)	16 或 32 位 CRC, 带可编程生成器多项式
外部总线接口	否
时钟	
外部晶振振荡器或谐振器	低范围、低功耗或全摆幅: 32 - 40 kHz 高范围、低功耗或全摆幅: 3 - 32 MHz
外部方波输入时钟	直流为 50 MHz
内部时钟参考	32 kHz 振荡器 4 MHz 振荡器 1 kHz 振荡器 48 MHz
锁相环 (PLL)	最高 120 MHz VCO
锁频环 (FLL)	范围 1: 20 - 25 MHz 范围 2: 40 - 50 MHz

下一页继续介绍此表...

表 2-1. 特性汇总 (继续)

特性	器件概述
	范围 3: 60 - 75 MHz 范围 4: 80 - 100 MHz
48 MHz 内部参考 (IRC48M)	1.5% 精度 (开环) 0.25% 精度 (闭环)
人机接口 (HMI)	
通用输入/输出 (GPIO)	最高 66 引脚中断 / DMA 请求功能 仅限端口 D 上的数字毛刺滤波器 所有输入引脚上的迟滞、上拉电阻和下拉电阻 仅限 PTA4 和 RESET_B 输入引脚上的无源输入滤波器 所有输出引脚上的压摆率均可配置 8 个引脚上的大电流驱动能力可配置选项 (PTB0, PTB1, PTD4, PTD5, PTD6, PTD7, PTC3, 和 PTC4)
模拟	
电源管理控制器 (PMC)	低压警告和检测、高压检测, 带可选触发点
16 位数模转换器 0 (ADC0)	17 单端通道 4 差分对 (高精度)
可编程增益放大器 (PGA)	否
高速比较器 (CMP)	1
6 位数模转换器 (DAC)	1
12 位 DAC	1
运算放大器 (OPAMP)	否
跨导放大器 (TRIAMP)	否
可编程参考电压 (VREF)	否
定时器	
可编程延迟区块 (PDB)	每个 ADC 1 个触发器, 每个 ADC 2 个预触发器, 每个 DAC 1 个触发器, 每个 CMP 1 个脉冲输出
16 位低功耗定时器 PWM 模块 (TPM0)	6 条带 DMA 支持的通道 在停止/VLPS 模式下工作
16 位低功耗定时器 PWM 模块 (TPM1)	2 条带 DMA 支持的通道 在停止/VLPS 模式下工作
16 位低功耗定时器 PWM 模块 (TPM2)	2 条带 DMA 支持的通道 在停止/VLPS 模式下工作
以太网 1588 定时器	否
32 位可编程中断定时器 (PIT)	4 通道
载波调制器定时器 (CMT)	否
独立实时时钟 (IRTC)	辅助电源 32 kHz 外部振荡器

下一页继续介绍此表...

表 2-1. 特性汇总 (继续)

特性	器件概述
	32 字节待机 RAM
低功耗定时器 (LPT)	1 通道、16 位脉冲计数器或 周期性中断
通信接口	
以太网控制器	否
FlexIO	1 支持广泛的协议, 包括但不限于 UART、I2C、SPI、I2S 在停止/VLPS 模式下工作 支持 DMA
通用串行总线 (USB) 2.0 FS 控制器	低速/全速 支持主机、设备和 OTG USB 时钟恢复 (仅限器件)
控制器局域网 (CAN)	KS22: 2; KS20: 1
串行外设接口 (SPI)	2
低功耗 I2C (LPI2C)	2
低功耗通用异步接收器/发送器 0 (LPUART0)	1
通用异步接收器/发送器 0 (UART0)	1 硬件流控制 IrDA ISO-7816 较高波特率 (CPU 时钟)
UART 1-2	2 硬件流控制 IrDA 较高波特率 (CPU 时钟)
安全数码 (SD) 接口	否
同步音频接口 (SAI)	2 集成电路内置音频总线 (I2S) AC'97 支持

2.3 结构框图

下图展示了 SOC 超集器件的顶层结构框图。

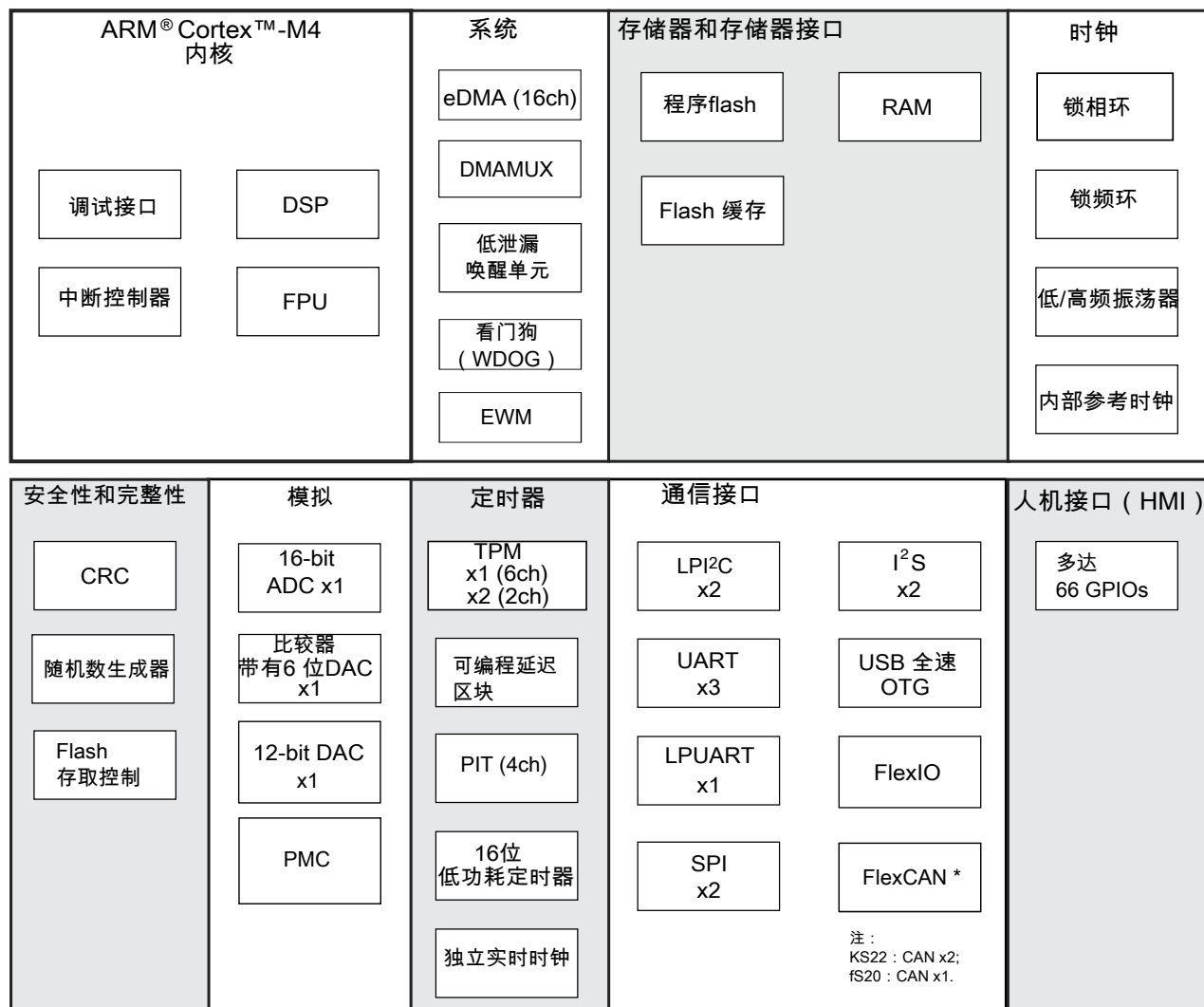


图 2-1. SOC 结构框图

2.4 模块功能类别

此设备上的模块按功能类别分组。下列章节更详细地说明了每一类的模块。

表 2-2. 模块功能类别

模块类别	说明
ARM® Cortex®-M4 core	<ul style="list-style-type: none"> ARM's Cortex-M 类别中的 32 位 MCU 内核添加了基于 ARMv7 架构的 DSP 指令和单精度浮点单元
系统	<ul style="list-style-type: none"> 系统集成模块 电源管理和模式控制器 <ul style="list-style-type: none"> 基于高速运行、运行、等待、停止和断电模式下的多种功耗模式 低漏电唤醒单元 其他控制模块 交叉开关 外设桥 直接内存访问 (DMA) 控制器, 带复用器, 以增加可用 DMA 请求。 外部看门狗监视器 看门狗
存储器	<ul style="list-style-type: none"> 内部存储器包括: <ul style="list-style-type: none"> 程序 flash 存储器 SRAM
时钟模块	<ul style="list-style-type: none"> 提供多个从内部和外部生成时钟的时钟生成选项 系统振荡器用于为 MCU 提供时钟源 RTC 振荡器用于为 RTC 提供时钟源
安全	<ul style="list-style-type: none"> 用于错误检测的循环冗余检查模块
模拟	<ul style="list-style-type: none"> 高速模数转换器 比较器 数模转换器
定时器	<ul style="list-style-type: none"> 可编程延迟块 低功耗定时器/PWM 周期中断定时器 低功耗定时器 独立实时时钟
通信	<ul style="list-style-type: none"> USB OTG 控制器, 具有内置 FS/LS 收发器 CAN 串行外设接口 低功耗集成电路总线 (I²C) UART 低功耗 UART (LPUART) 集成电路内置音频总线 (I²S) FlexIO
人机界面 (HMI)	<ul style="list-style-type: none"> 通用输入/输出控制器

2.4.1 ARM® Cortex®-M4 内核模块

以下内核模块可用于此设备。

表 2-3. 内核模块

模块	说明
ARM Cortex-M4	<p>ARM® Cortex®-M4 是 Cortex M 系列处理器中的最新产品, 旨在用作微控制器内核, 重点面向成本敏感、具有确定性且由中断驱动的环境。Cortex M4 处理器基于 ARMv7 架构和 Thumb®-2 ISA, 并且向上兼容 Cortex M3、Cortex M1 和 Cortex M0 架构。Cortex M4 的改进包括一个 ARMv7 Thumb-2 DSP (从 ARMv7-A/R 配置架构</p>

下一页继续介绍此表...

表 2-3. 内核模块 (继续)

模块	说明
	移植而来), 提供 32 位指令, 具有 SIMD (单指令多数据) DSP 类型的乘法/累加和饱和算法。
NVIC	ARMv7-M 异常模型和嵌套向量中断控制器 (NVIC) 集成一个可重定位的向量表, 支持许多外部中断、一个非屏蔽中断 (NMI) 和优先级。 NVIC 凭借等效系统和经简化的编程来替换影子寄存器。NVIC 包含中断响应的函数的地址。该地址通过指令端口提取, 允许并行寄存器堆栈与查找。前十六个条目分配给 ARM 内部中断源, 而其他条目则映射到由 MCU 定义的中断。
AWIC	异步唤醒中断控制器 (AWIC) 最基本的功能是在停止模式中检测异步唤醒事件和系统时钟控制逻辑信号用以重新获取系统时钟。时钟重新启动后, NVIC 开始执行挂起中断、正常中断或执行中的事件。
调试接口	该器件的大部分调试都基于 ARM CoreSight™ 架构。本器件支持 4 种调试接口: <ul style="list-style-type: none"> • IEEE 1149.1 JTAG • IEEE 1149.7 JTAG (cJTAG) • 串行线调试 (SWD) • ARM 实时跟踪接口

2.4.2 系统模块

以下系统模块可用于此设备。

表 2-4. 系统模块

模块	说明
系统集成模块 (SIM)	SIM 包括集成逻辑和多个模块配置设置。
系统模式控制器	SMC 可为进入和退出各功耗模式提供控制和保护、控制电源管理控制器 (PMC) 以及为整个 MCU 提供复位进入和退出。
电源管理控制器 (PMC)	PMC 为用户提供了多种功耗选项, 用户因此可针对所需的功能水平优化功耗。包括上电复位 (POR) 和集成的低压检测 (LVD) / 高压检测 (HVD) 功能, 具有复位 (掉电) 能力和可选 LVD/HVD 触发点。
低漏电唤醒单元 (LLWU)	LLWU 模块允许器件通过不同的内部外设和外部引脚源从低漏电功耗模式 (LLS 和 VLLS) 中唤醒。
其他控制模块 (MCM)	MCM 包括集成逻辑。
交叉开关 (XBS)	XBS 连接总线主机和总线从机, 从而使所有总线主机都能同时访问不同的总线从机, 并且还可在其访问相同从机时提供总线主机间的仲裁。
外设桥	外设桥将交叉开关接口转换为可访问器件上大多数外设的接口。
DMA 多路复用器 (DMAMUX)	DMA 多路复用器从多个 DMA 请求中选择少量 DMA 控制器。
直接存储器访问 (DMA) 控制器	DMA 控制器提供有带传输控制描述符的可编程通道, 用于通过 8 位、16 位、32 位、16 字节和 32 字节数据值的双地址传输实现数据移动。
外部看门狗监视器 (EWM)	EWM 是软件看门狗模块的冗余机制, 用于监控内部和外部系统操作是否存在故障情况。
软件看门狗 (WDOG)	WDOG 监控内部系统操作并在系统出现故障时强制复位。它可以通过带有独立可编程刷新窗口的 1KHz 低功耗振荡器, 检测程序流或者系统频率的偏差。

2.4.3 存储器和存储器接口

该器件提供下列存储器和存储器接口。

表 2-5. 存储器和存储器接口

模块	说明
Flash 存储器	<ul style="list-style-type: none"> 程序 Flash 存储器 — 可执行程序代码的非易失性 Flash 存储器
Flash 控制器	管理芯片和片上 Flash 的接口。
SRAM	内部系统 RAM。部分 SRAM 在 LLS2 和 VLLS2 低漏电模式下保持通电。
系统寄存器文件	在所有功耗模式下可以访问的 32 字节寄存器文件，通过 VDD 为其供电。
VBAT 寄存器文件	在所有功耗模式下可以访问的 32 字节寄存器文件，通过 VBAT 为其供电。

2.4.4 时钟

此芯片提供下列时钟模块。

表 2-6. 时钟模块

模块	说明
多功能时钟产生器 (MCG)	MCG 可为 MCU 提供多个时钟源，包括： <ul style="list-style-type: none"> 锁相环 (PLL) — 电压控制振荡器 (VCO) 锁频环 (FLL) — 数字控制振荡器 (DCO) 内部参考时钟 — 可用作其他片上外设的时钟源
48 MHz 内部参考时钟 (IRC48M)	IRC48M 提供由内部生成的时钟源。时钟恢复电路使用传入的 USB 数据流来调整内部振荡器并使能内部振荡器，以满足 USB 时钟容差的要求。
系统振荡器	系统振荡器与外部晶体或谐振器一同生成 MCU 的基准时钟。
实时时钟振荡器	RTC 振荡器具有独立的电源，支持 32 kHz 晶体振荡器以为 RTC 时钟供电。RTC 可选择性代替系统振荡器作为主振荡器源。

2.4.5 安全性和完整性模块

以下安全性和完整性模块可用于此设备：

表 2-7. 安全性和完整性模块

模块	说明
随机数发生器 (RNG)	支持数字签名标准中定义的密钥生成算法。
循环冗余校验 (CRC)	使用 16/32 位移位寄存器的硬件 CRC 生成器电路。用于所有单位、双位、奇数和大多数多位错误、可编程初始种子值和可选功能的错误检测，以便通过转置寄存器转置输入数据和 CRC 结果。

2.4.6 模拟模块

此器件提供下列模拟模块：

表 2-8. 模拟模块

模块	说明
16 位模数转换器 (ADC)	16 位逐次逼近型 ADC
模拟比较器	比较整个电源电压范围内的两个模拟输入电压。
6 位数模转换器 (DAC)	64 分接头的梯形电阻网络，可为需要参考电压的应用提供可选的参考电压。
12 位数模转换器 (DAC)	低功耗通用 DAC，其输出可置于外部引脚上或设置为模拟比较器或 ADC 的其中一个输入。

2.4.7 定时器模块

此器件提供下列定时器模块：

表 2-9. 定时器模块

模块	说明
可编程延迟区块 (PDB)	<ul style="list-style-type: none"> • 16 位分辨率。 • 3 位预分频器。 • 触发器正极转换事件信号将启动计数器。 • 支持两个已触发的延迟输出信号，每个信号具有来自触发器事件的独立控制延迟。 • 输出可以为同时 OR'd，以安排来自一个输入触发器事件的两个转换，并且可以为脉冲输出安排精确的边沿部署。此功能用于为 CMP 窗口功能生成控制信号和封装引脚的输出（如应用需要），例如临界导通模式功率系数校正。 • 支持持续脉冲输出或单发模式，每个输出均可通过可能的触发事件独立使能。 • 支持旁通模式。 • 支持 DMA。
定时器/PWM 模块 (TPM)	<ul style="list-style-type: none"> • 多通道定时器，支持输入捕捉、输出比较和生成用于控制电机和电源管理应用的 PWM 信号。 • 该计数器、比较和捕捉寄存器都由可在低功耗模式下保持使能的异步时钟进行计时。 • 支持 DMA。
周期性中断定时器 (PIT)	<ul style="list-style-type: none"> • 四个通用中断定时器。 • 适用于触发 ADC 转换。 • 32 位计数器分辨率。 • 支持 DMA。
低功耗定时器 (LPTimer)	<ul style="list-style-type: none"> • 用于 1 kHz 预分频器/去抖滤波器（内部 LPO）、32.768 kHz（外部晶振）或内部参考时钟的可选时钟。 • 带 16 位计数器的可配置去抖滤波器或预分频器。 • 16 位定时计数器或比较脉冲计数器。 • 完成定时器比较后生成中断。 • 完成定时器比较后生成硬件触发器信号。
实时时钟 (RTC)	<ul style="list-style-type: none"> • 独立的电源、POR 和 32 kHz 晶体振荡器。

表 2-9. 定时器模块

模块	说明
	<ul style="list-style-type: none"> • 32 位秒计数器，带 32 位闹钟。 • 带补偿功能的 16 位预分频器，可以校正 0.12 ppm 至 3906 ppm 的误差。

2.4.8 通信接口

该器件提供下列通信接口：

表 2-10. 通信模块

模块	说明
USB OTG (低速/全速)	USB 2.0 兼容模块，支持主机、设备和 On-The-Go 模式。包括全速和低速片上收发器。
控制器局域网 (CAN)	支持 CAN2.0B 规范的完整实现。
串行外设接口 (SPI)	同步串行总线，用于与外部设备通信。
低功耗内部集成电路 (LPI2C)	LPI2C 模块提供低功耗 I2C 模块，该模块可在低功耗停止模式下运行（如需要）。此外，它还符合系统管理总线 (SMBus) 规范版本 2。
通用异步收发器 (UART)	异步串行总线通信接口，带可编程 8 位和 9 位数据格式，支持 ISO 7816 智能卡接口。
LPUART	低功耗 UART 模块，支持停止模式下的工作。
I2S	I2S 为一个全双工串行端口，允许芯片与使用集成电路内置音频总线 (I2S) 和 Intel® AC97 标准的各种串行器件通信，例如标准编解码器、数字信号处理器 (DSP)、微处理器、外设和音频编解码器。
FlexIO	FlexIO 模块支持模拟其他 UART、SPI、LPI2C、I2S、PWM 和其他串行模块等。当其使用的时钟保持使能时，该模块可以在 VLPS 模式下保持运行。

2.4.9 人机接口

该芯片提供下列人机接口 (HMI)：

表 2-11. HMI 模块

模块	说明
通用输入/输出 (GPIO)	<p>所有通用输入或输出 (GPIO) 引脚均可生成中断和 DMA 请求。</p> <p>所有 GPIO 引脚均具有 3.3 V 公差。</p>

2.5 可订购部件编号

下表汇总了本文档所涵盖的芯片型号标识。

表 2-12. 可订购型号汇总

部件编号	CPU 频率	引脚数	封装	程序 flash	SRAM	GPIO	CAN
MKS22FN256VLL12	120 MHz	100	LQFP	256 KB	64 KB	66	2
MKS22FN256VLH12	120 MHz	64	LQFP	256 KB	64 KB	40	2
MKS22FN256VFT12	120 MHz	48	QFN	256 KB	64 KB	35	2
MKS22FN128VLL12	120 MHz	100	LQFP	128 KB	64 KB	66	2
MKS22FN128VLH12	120 MHz	64	LQFP	128 KB	64 KB	40	2
MKS22FN128VFT12	120 MHz	48	QFN	128 KB	64 KB	35	2
MKS20FN256VLL12	120 MHz	100	LQFP	256 KB	64 KB	66	1
MKS20FN256VLH12	120 MHz	64	LQFP	256 KB	64 KB	40	1
MKS20FN256VFT12	120 MHz	48	QFN	256 KB	64 KB	35	1
MKS20FN128VLL12	120 MHz	100	LQFP	128 KB	64 KB	66	1
MKS20FN128VLH12	120 MHz	64	LQFP	128 KB	64 KB	40	1
MKS20FN128VFT12	120 MHz	48	QFN	128 KB	64 KB	35	1

第 3 章 内核概述

3.1 ARM Cortex-M4 内核配置

本节总结如何在芯片上配置模块。本模块的完整文档由 ARM 提供，请访问：arm.com。

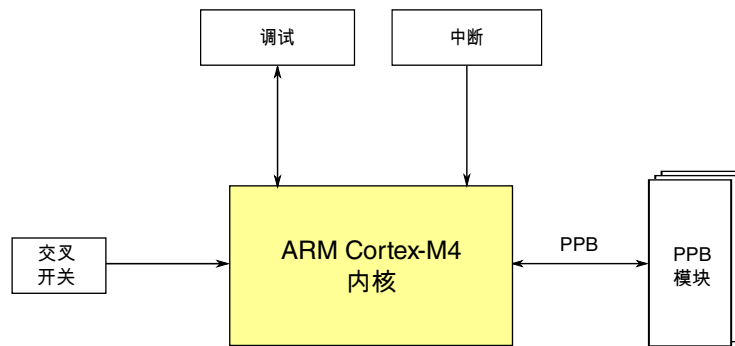


图 3-1. 内核配置

表 3-1. 相关信息的参考链接

主题	相关模块	参考
完整说明	ARM Cortex-M4 core	ARM Cortex-M4 技术参考手册
系统存储器映射		系统存储器映射
时钟		时钟分布
电源管理		电源管理
系统/指令/数据总线模块	交叉开关	交叉开关
调试	IEEE 1149.1 JTAG IEEE 1149.7 JTAG (cJTAG) 串行线调试 (SWD) ARM 实时跟踪接口	调试
中断	可嵌套向量的中断控制器 (NVIC)	NVIC

下一页继续介绍此表...

表 3-1. 相关信息的参考链接 (继续)

主题	相关模块	参考
私有外设总线 (PPB) 模块	其他控制模块 (MCM)	MCM
私有外设总线 (PPB) 模块	单精度浮点单元 (FPU)	FPU

3.1.1 总线、互连和接口

ARM Cortex-M4 内核有 4 条总线，如下表所示。

总线名称	说明
指令代码 (ICODE) 总线	ICODE 和 DCODE 总线发生多路复用。这一多路复用总线称为 CODE 总线，通过一个单主机端口连接到交叉开关。
数据代码 (DCODE) 总线	
系统总线	系统总线连接到交叉开关上的单独主机端口。
私有外设 (PPB) 总线	PPB 用于访问以下模块： <ul style="list-style-type: none"> ARM 模块，如 NVIC、ITM、DWT、FBP、和 ROM 表 其他控制模块 (MCM)

3.1.2 系统节拍定时器

系统节拍定时器的时钟源始终是内核时钟 FCLK。这导致如下结果：

- SysTick 控制和状态寄存器中的 CLKSOURCE 位始终设置为选择内核时钟。
- 由于时序基准 (FCLK) 是一个可变频率，因此 SysTick 校准值寄存器中的 TENMS 位将始终为零。
- SysTick 校准值寄存器中的 NOREF 位始终置位，这意味着 FCLK 是唯一可用的基准时序源。

3.1.3 调试设施

此芯片具有多种调试功能，包括运行控制和跟踪功能。标准 ARM 调试端口支持 JTAG 和 SWD 接口。此芯片还支持 cJTAG 接口。

3.1.4 内核权限级别

ARM 文档使用不同于本文档中的术语以区分权限级别。

如果遇到术语...	它也意味着该术语...
有权限	管理员
无权限或者用户	用户

3.2 嵌套向量中断控制器 (NVIC) 配置

本节总结如何在芯片上配置模块。本模块的完整文档由 ARM 提供，请访问：arm.com。

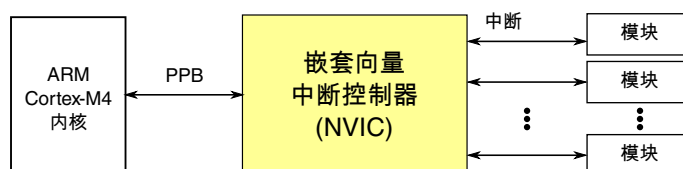


图 3-2. NVIC 配置

表 3-2. 相关信息的参考链接

主题	相关模块	参考
完整说明	可嵌套中断向量控制器 (NVIC)	ARM Cortex-M4 技术参考手册
系统存储器映射		系统存储器映射
时钟		时钟分布
电源管理		电源管理
私有外设总线 (PPB)	ARM Cortex-M4 内核	ARM Cortex-M4 内核

3.2.1 中断优先级

此设备支持 16 个中断优先级。因此，在 NVIC 中，IPR 寄存器中的各个源都包含 4 位。例如，IPR0 如下所示：

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读	IRQ3				0	0	0	0	IRQ2				0	0	0	0	IRQ1				0	0	0	0	IRQ0				0	0	0	0
写					■								■								■											

3.2.2 不可屏蔽中断

对 NVIC 的不可屏蔽中断请求由外部 $\overline{\text{NMI}}$ 信号控制。 $\overline{\text{NMI}}$ 信号复用的引脚，必须配置为 $\overline{\text{NMI}}$ 功能，以产生不可屏蔽中断请求。

3.2.3 中断通道分配

下表中定义了中断源分配。

- 向量编号 - 中断处于响应状态时，存储在堆栈上的值。
- IRQ 编号 - 非内核中断源编号，其为向量编号减 16。

IRQ 编号用于 ARM 的 NVIC 文档中。

表 3-4. 中断向量分配

地址	向量	IRQ ¹	NVIC 非 IPR 寄存器编号 ²	NVIC IPR 寄存器编号 ³	来源模块	来源描述
ARM 内核系统处理程序向量						
0x0000_0000	0	—	—	—	ARM 内核	初始堆栈指针
0x0000_0004	1	—	—	—	ARM 内核	初始程序计数
0x0000_0008	2	—	—	—	ARM 内核	不可屏蔽中断 (NMI)
0x0000_000C	3	—	—	—	ARM 内核	硬故障
0x0000_0010	4	—	—	—	ARM 内核	MemManage 故障
0x0000_0014	5	—	—	—	ARM 内核	总线故障
0x0000_0018	6	—	—	—	ARM 内核	使用故障
0x0000_001C	7	—	—	—	—	—
0x0000_0020	8	—	—	—	—	—
0x0000_0024	9	—	—	—	—	—
0x0000_0028	10	—	—	—	—	—
0x0000_002C	11	—	—	—	ARM 内核	管理程序调用 (SVCALL)
0x0000_0030	12	—	—	—	ARM 内核	调试监控器
0x0000_0034	13	—	—	—	—	—
0x0000_0038	14	—	—	—	ARM 内核	可挂起的系统服务请求 (PendableSrvReq)
0x0000_003C	15	—	—	—	ARM 内核	系统节拍定时器 (SysTick)
非内核向量						
0x0000_0040	16	0	0	0	DMA	DMA 通道 0 传输完成
0x0000_0044	17	1	0	0	DMA	DMA 通道 1 传输完成
0x0000_0048	18	2	0	0	DMA	DMA 通道 2 传输完成
0x0000_004C	19	3	0	0	DMA	DMA 通道 3 传输完成
0x0000_0050	20	4	0	1	DMA	DMA 通道 4 传输完成

下一页继续介绍此表...

表 3-4. 中断向量分配 (继续)

地址	向量	IRQ ¹	NVIC 非 IPR 寄存器编号 ²	NVIC IPR 寄存器编号 ³	来源模块	来源描述
0x0000_0054	21	5	0	1	DMA	DMA 通道 5 传输完成
0x0000_0058	22	6	0	1	DMA	DMA 通道 6 传输完成
0x0000_005C	23	7	0	1	DMA	DMA 通道 7 传输完成
0x0000_0060	24	8	0	2	DMA	DMA 通道 8 传输完成
0x0000_0064	25	9	0	2	DMA	DMA 通道 9 传输完成
0x0000_0068	26	10	0	2	DMA	DMA 通道 10 传输完成
0x0000_006C	27	11	0	2	DMA	DMA 通道 11 传输完成
0x0000_0070	28	12	0	3	DMA	DMA 通道 12 传输完成
0x0000_0074	29	13	0	3	DMA	DMA 通道 13 传输完成
0x0000_0078	30	14	0	3	DMA	DMA 通道 14 传输完成
0x0000_007C	31	15	0	3	DMA	DMA 通道 15 传输完成
0x0000_0080	32	16	0	4	DMA	DMA 错误中断通道 0-15
0x0000_0084	33	17	0	4	MCM	FPU 源
0x0000_0088	34	18	0	4	Flash 存储器	命令完成
0x0000_008C	35	19	0	4	Flash 存储器	读取冲突
0x0000_0090	36	20	0	5	模式控制器	低压检测, 低压警告
0x0000_0094	37	21	0	5	LLWU	低漏电唤醒 注: LLWU 中断不可通过中断控制器屏蔽, 以避免系统无法在 LLS 恢复时完全退出停止模式的现象。
0x0000_0098	38	22	0	5	WDOG 或 EWM	两个看门狗模块共用此中断
0x0000_009C	39	23	0	5	RNG	随机数生成器
0x0000_00A0	40	24	0	6	LPI ² C0	—
0x0000_00A4	41	25	0	6	LPI ² C1	—
0x0000_00A8	42	26	0	6	SPI0	所有源的单个中断向量
0x0000_00AC	43	27	0	6	SPI1	所有源的单个中断向量
0x0000_00B0	44	28	0	7	I ² S0	传送
0x0000_00B4	45	29	0	7	I ² S0	接收
0x0000_00B8	46	30	0	7	LPUART0	状态和错误
0x0000_00BC	47	31	0	7	UART0	UART 状态源的单个中断向量
0x0000_00C0	48	32	1	8	UART0	UART 错误源的单个中断向量
0x0000_00C4	49	33	1	8	UART1	UART 状态源的单个中断向量
0x0000_00C8	50	34	1	8	UART1	UART 错误源的单个中断向量
0x0000_00CC	51	35	1	8	UART2	UART 状态源的单个中断向量
0x0000_00D0	52	36	1	9	UART2	UART 错误源的单个中断向量
0x0000_00D4	53	37	1	9	—	—
0x0000_00D8	54	38	1	9	—	—
0x0000_00DC	55	39	1	9	ADC0	—

下一页继续介绍此表...

表 3-4. 中断向量分配 (继续)

地址	向量	IRQ ¹	NVIC 非 IPR 寄存器编号 ²	NVIC IPR 寄存器编号 ³	来源模块	来源描述
0x0000_00E0	56	40	1	10	CMP0	—
0x0000_00E4	57	41	1	10	—	—
0x0000_00E8	58	42	1	10	TPM0	—
0x0000_00EC	59	43	1	10	TPM1	—
0x0000_00F0	60	44	1	11	TPM2	—
0x0000_00F4	61	45	1	11	—	—
0x0000_00F8	62	46	1	11	RTC	警告中断
0x0000_00FC	63	47	1	11	RTC	秒中断
0x0000_0100	64	48	1	12	PIT	通道 0
0x0000_0104	65	49	1	12	PIT	通道 1
0x0000_0108	66	50	1	12	PIT	通道 2
0x0000_010C	67	51	1	12	PIT	通道 3
0x0000_0110	68	52	1	13	PDB	—
0x0000_0114	69	53	1	13	USB OTG	—
0x0000_0118	70	54	1	13	—	—
0x0000_011C	71	55	1	13	—	—
0x0000_0120	72	56	1	14	DAC0	—
0x0000_0124	73	57	1	14	MCG	—
0x0000_0128	74	58	1	14	低功耗定时器	—
0x0000_012C	75	59	1	14	端口控制模块	引脚检测 (Port A)
0x0000_0130	76	60	1	15	端口控制模块	引脚检测 (Port B)
0x0000_0134	77	61	1	15	端口控制模块	引脚检测 (Port C)
0x0000_0138	78	62	1	15	端口控制模块	引脚检测 (Port D)
0x0000_013C	79	63	1	15	端口控制模块	引脚检测 (Port E)
0x0000_0140	80	64	2	16	软件	软件中断 ⁴
0x0000_0144	81	65	2	16	—	—
0x0000_0148	82	66	2	16	—	—
0x0000_014C	83	67	2	16	—	—
0x0000_0150	84	68	2	17	—	—
0x0000_0154	85	69	2	17	—	—
0x0000_0158	86	70	2	17	FlexIO	FlexIO
0x0000_015C	87	71	2	17	—	—
0x0000_0160	88	72	2	18	—	—
0x0000_0164	89	73	2	18	—	—
0x0000_0168	90	74	2	18	—	—
0x0000_016C	91	75	2	18	CAN0	OR'ed 消息缓冲区 (0-15)
0x0000_0170	92	76	2	19	CAN0	总线关闭

下一页继续介绍此表...

表 3-4. 中断向量分配 (继续)

地址	向量	IRQ ¹	NVIC 非 IPR 寄存器编号 ²	NVIC IPR 寄存器编号 ³	来源模块	来源描述
0x0000_0174	93	77	2	19	CAN0	错误
0x0000_0178	94	78	2	19	CAN0	发送警告
0x0000_017C	95	79	2	19	CAN0	接收警告
0x0000_0180	96	80	2	20	CAN0	唤醒
0x0000_0184	97	81	2	20	—	—
0x0000_0188	98	82	2	20	—	—
0x0000_018C	99	83	2	20	—	—
0x0000_0190	100	84	2	21	—	—
0x0000_0194	101	85	2	21	—	—
0x0000_0198	102	86	2	21	—	—
0x0000_019C	103	87	2	21	—	—
0x0000_01A0	104	88	2	22	I ² S1	发送
0x0000_01A4	105	89	2	22	I ² S1	receive
0x0000_01A8	106	90	2	22	—	—
0x0000_01AC	107	91	2	22	—	—
0x0000_01B0	108	92	2	23	—	—
0x0000_01B4	109	93	2	23	—	—
0x0000_01B8	110	94	2	23	CAN1 (只针对 KS22)	OR'ed 消息缓冲区 (0-15)
0x0000_01BC	111	95	2	23	CAN1 (只针对 KS22)	总线关闭
0x0000_01C0	112	96	3	24	CAN1 (只针对 KS22)	错误
0x0000_01C4	113	97	3	24	CAN1 (只针对 KS22)	发送警告
0x0000_01C8	114	98	3	24	CAN1 (只针对 KS22)	接收警告
0x0000_01CC	115	99	3	24	CAN1 (只针对 KS22)	唤醒

1. 表示 NVIC 的中断源编号。
2. 表示用于该 IRQ 的 NVIC 的 ISER、ICER、ISPR、ICPR 和 IABR 寄存器编号。计算该值的公式为：IRQ 除 32
3. 表示用于该 IRQ 的 NVIC 的 IPR 寄存器编号。计算该值的公式为：IRQ 除 4
4. 此中断仅可通过 NVIC 寄存器挂起或清除。

3.2.3.1 确定位域和寄存器位置，以配置特定中断

假设用户需要配置低功耗定时器 (LPTMR) 中断。下表是[中断通道分配](#)中 LPTMR 行的摘录。

表 3-5. LPTMR 中断向量分配

地址	向量	IRQ ¹	NVIC 非 IPR 寄存器编号 ²	NVIC IPR 寄存器编号 ³	来源模块	源描述
0x0000_0128	74	58	1	14	低功耗定时器	—

1. 表示 NVIC 的中断源编号。
2. 表示用于该 IRQ 的 NVIC 的 ISER、ICER、ISPR、ICPR 和 IABR 寄存器编号。计算该值的公式为：IRQ 除 32
3. 表示用于该 IRQ 的 NVIC 的 IPR 寄存器编号。计算该值的公式为：IRQ 除 4

- 您将用于配置中断的 NVIC 寄存器为：
 - NVICISER1
 - NVICICER1
 - NVICISPR1
 - NVICICPR1
 - NVICIABR1
 - NVICIPR14
- 要确定这些特定寄存器中的特定 IRQ 的位域：
 - NVICISER1, NVICICER1, NVICISPR1, NVICICPR1, NVICIABR1 位位置 = $IRQ \bmod 32 = 26$
 - NVICIPR14 位域起始位置 = $8 * (IRQ \bmod 4) + 4 = 20$

由于 NVICIPR 位域为 4 位宽 (16 个优先级), NVICIPR14 位域范围为 20-23。

因此, 将使用以下位域位置配置 LPTMR 中断:

- NVICISER1[26]
- NVICICER1[26]
- NVICISPR1[26]
- NVICICPR1[26]
- NVICIABR1[26]
- NVICIPR14[23:20]

3.3 异步唤醒中断控制器 (AWIC) 配置

本节总结如何在芯片上配置模块。本模块的完整文档由 ARM 提供, 请访问: arm.com。

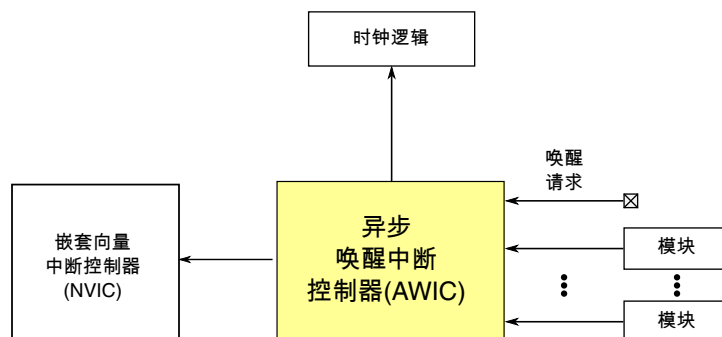


图 3-3. 异步唤醒中断控制器配置

表 3-6. 相关信息的参考链接

主题	相关模块	参考
系统存储器映射		系统存储器映射
时钟		时钟分布
电源管理		电源管理
	可嵌套中断向量控制器 (NVIC)	NVIC
唤醒请求		AWIC 唤醒源

3.3.1 唤醒源

该器件将以下内部和外部输入用于 AWIC 模块。

表 3-7. AWIC Partial Stop、Stop 和 VLPS 唤醒源

唤醒源	说明
可用系统复位	RESET 引脚, LPO 是其时钟源时的 WDOG, 以及 JTAG
低电压检测	电源模式控制器
低压警告	电源模式控制器
高电压检测	电源模式控制器
引脚中断	端口控制模块 — 任何已使能的引脚中断都能唤醒系统
ADC	使用内部时钟源时, ADC 起作用
CMP	由于无系统时钟可用, 因此, 功能受限, 触发模式可通过周期性采样提供唤醒功能
LPI ² C	当使用在 Stop 和 VLPS 模式下激活的时钟源时起作用
FlexIO	当使用在 Stop 和 VLPS 模式下激活的时钟源时起作用
TPM	当使用在 Stop 和 VLPS 模式下激活的时钟源时起作用
UART	RXD 上的有效边沿
LPUART	当使用在 Stop 和 VLPS 模式下激活的时钟源时起作用
USB FS/LS 控制器	唤醒
LPTMR	当使用在 Stop 和 VLPS 模式下激活的时钟源时起作用

下一页继续介绍此表...

表 3-7. AWIC Partial Stop、Stop 和 VLPS 唤醒源 (继续)

唤醒源	说明
RTC	在 Stop/VLPS 模式下起作用
I2S (SAI)	当使用外部位时钟或外部主机时钟时起作用
TPM	当使用在 Stop 和 VLPS 模式下激活的时钟源时起作用
CAN	在边沿上唤醒 (CANx_RX)
NMI	不可屏蔽中断

3.4 FPU 配置

本节总结如何在芯片上配置模块。

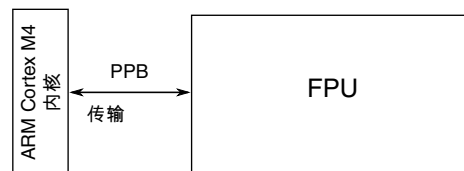


图 3-4. FPU configuration

表 3-8. 相关信息的参考链接

主题	相关模块	参考
完整说明	FPU	ARM Cortex-M4 技术参考手册
系统存储器映射		系统存储器映射
时钟		时钟分布
电源管理		电源管理
传输 私有外设总线 (PPB)	ARM Cortex M4 内核	ARM Cortex-M4 内核

3.5 JTAG 控制器配置

本节总结如何在芯片上配置模块。

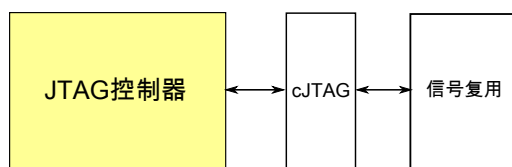


图 3-5. JTAGC 控制器配置

表 3-9. 相关信息的参考链接

主题	相关模块	参考
完整说明	JTAGC	JTAGC
信号复用	端口控制	信号复用

第 4 章 存储器和存储器接口

4.1 Flash 存储器配置

本节总结如何在芯片上配置模块。

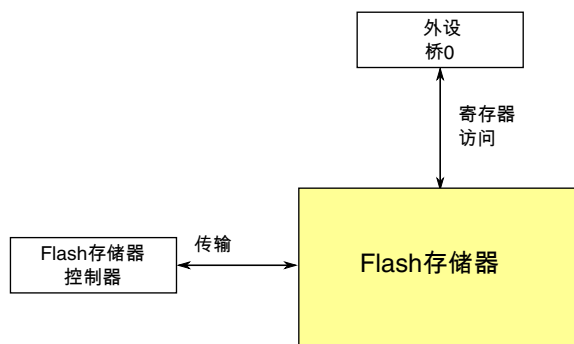


图 4-1. Flash 存储器配置

表 4-1. 相关信息的参考链接

主题	相关模块	参考
完整说明	Flash 存储器	
系统存储器映射		系统存储器映射
时钟		时钟分布
传输	Flash 存储器控制器	Flash 存储器控制器
寄存器访问	外设桥	外设桥

4.1.1 Flash 存储器类型

此器件包含以下类型的 Flash 存储器：

- 程序 Flash 存储器 — 可执行程序代码的非易失性 Flash 存储器

4.1.2 Flash 存储器大小

本文档涵盖的器件包括：

- 1 个块，包含 2 个 KB 扇区

本文所涉及器件的 Flash 存储器的大小有：

器件	程序 Flash (KB)	块 0 地址范围
MKS2xFN256Vxx12	256	0x0000_0000–0x0003_FFFF
MKS2xFN128Vxx12	128	0x0000_0000–0x0001_FFFF

4.1.3 Flash 加密

有关如何在该器件上实施 Flash 加密的信息，请参见[芯片加密](#)。

4.1.4 Flash 程序限制

在高速运行或 VLPR 功耗模式下，不得编程或擦除此器件上的 Flash 存储器。

4.1.5 Flash 模式

始终在 NVM 正常模式下配置 Flash 存储器。任何操作条件下均不可将 Flash 配置为 NVM 特殊模式。

4.1.6 擦除所有 Flash 内容

可以通过设置 FTFA_FSEC[MEEN] 寄存器位来禁止全部擦除 MCU 的 Flash 内容。如果该寄存器被置为 'b10，那么全部擦除操作就被禁止了。软件可通过对 Flash 寄存器的一系列外设总线写入操作来启动擦除所有 Flash 块操作。此外，可以通过设置 DAP_CONTROL[0] 从 SWJ-DP 调试端口擦除 Flash 存储器外部的整个 Flash 存储器。设置 DAP_STATUS[0] 表示已接受整体擦除命令。当批量擦除完成时 DAP_STATUS[0] 清零。

4.1.7 FTF_FOFT 寄存器

Flash 存储器的 FTF_FOFT 寄存器允许用户自定义 MCU 启动时的操作。关于其定义的详细信息，请参见 [FOFT 启动选项](#)。

4.2 Flash 存储器控制器配置

本节总结如何在芯片上配置模块。

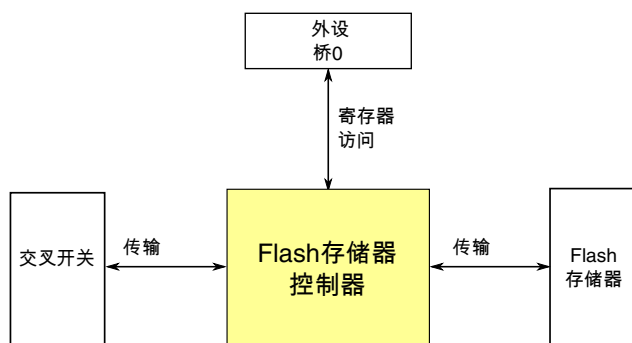


图 4-2. Flash 存储器控制器配置

表 4-2. 相关信息的参考链接

主题	相关模块	参考
完整说明	Flash 存储器控制器	Flash 存储器控制器
系统存储器映射		系统存储器映射
时钟		时钟分布
传输	Flash 存储器	Flash 存储器
传输	交叉开关	交叉开关
寄存器访问	外设桥	外设桥

4.2.1 主机数

Flash 存储器控制器最多支持 8 个交叉开关主机。但是，此器件具有不同的交叉开关主机数。有关主机端口分配的详细信息，请参见 [交叉开关配置](#)。

4.3 SRAM 配置

本节总结如何在芯片上配置模块。

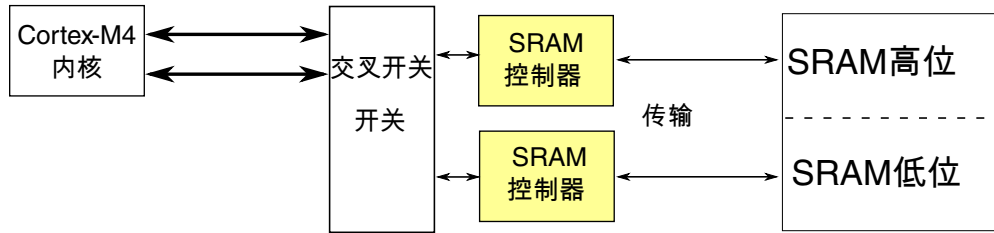


图 4-3. SRAM 配置

表 4-3. 相关信息的参考链接

主题	相关模块	参考
完整说明	SRAM	SRAM
系统存储器映射		系统存储器映射
时钟		时钟分布
传输	SRAM 控制器	SRAM 控制器
	ARM Cortex-M4 内核	ARM Cortex-M4 内核

4.3.1 SRAM 大小

该芯片集成了可由总线主机通过交叉开关访问的 SRAM。此片上 SRAM 分成 SRAM_L 和 SRAM_U 区域，在存储器中的固定地址 0x2000_0000 形成相邻的两块区域。例如：

- SRAM_L 固定至 0x1FFF_FFFF 并占用该结束地址前的空间。
- SRAM_U 固定至 0x2000_0000 并占用该起始地址后的空间。

注

ARM Cortex-M4 架构不支持跨越 0x2000_0000 边界的未对齐访问。

本文档中介绍的设备的 SRAM 大小如下表所示。

器件	SRAM_L 大小 (KB)	SRAM_U 大小 (KB)	总共 SRAM (KB)	地址范围
MKS2xFN256Vxx12	16	48	64	0x1FFF_C000-0x2000_BFFF
MKS2xFN128Vxx12	16	48	64	0x1FFF_C000-0x2000_BFFF

4.3.2 低功耗模式下的 SRAM 保留

SRAM 保留为 LLS3 和 VLLS3 模式。

在 LLS2 和 VLLS2 模式下，SRAM_U 从 0x2000_0000 起的 16 KB 区域将通电。
在 VLLS1 和 VLLS0 下，未保留任何 SRAM；但是，[32 字节寄存器文件](#) 可用。

4.4 系统寄存器文件配置

本节总结如何在芯片上配置模块。

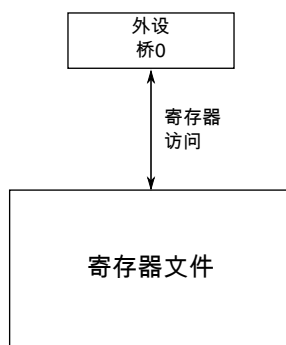


图 4-4. 系统寄存器文件配置

表 4-4. 相关信息的参考链接

主题	相关模块	参考
完整说明	寄存器文件	寄存器文件
系统存储器映射		系统存储器映射
时钟		时钟分布
电源管理		电源管理

4.4.1 系统寄存器文件

该器件包含一个可在所有功耗模式下运行的 32 字节寄存器文件。此系统寄存器文件由 8 个 4 字节寄存器 RFSYS_REG n 组成，其中 n 的范围为 0 至 7。

此外，它还会在低电压检测 (LVD) 和高电压检测 (HVD) 事件过程中保持内容，并且仅在上电复位时复位。

4.5 VBAT 寄存器文件配置

本节总结如何在芯片上配置模块。

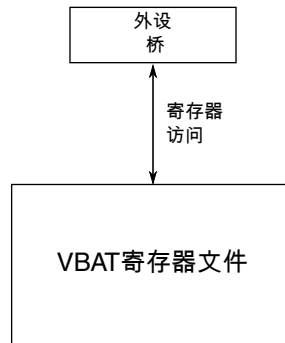


图 4-5. VBAT 寄存器文件配置

表 4-5. 相关信息的参考链接

主题	相关模块	参考
完整说明	VBAT 寄存器文件	VBAT 寄存器文件
系统存储器映射		系统存储器映射
时钟		时钟分布
电源管理		电源管理

4.5.1 VBAT 寄存器文件

该器件包含一个可在所有功耗模式下运行的 32 字节寄存器文件, 并且可通过 VBAT 上电。该 VBAT 寄存器文件由 8 个 4 字节寄存器 RFVBAT_REG n 组成, 其中 n 的范围为 0 至 7。

它仅可在 VBAT 上电复位过程中复位。

第 5 章 存储器映射

5.1 简介

本器件包含各种存储器和存储器映射外设，全部位于一个 32 位连续存储器空间内。本章介绍该存储器空间中存储器和外设的位置。

5.2 系统存储器映射

下表显示的是高级设备存储器映射。此映射提供了各部分的完整架构地址空间定义。基于存储器和外设的物理大小，实际使用的地址区域可能更小。

系统存储器映射包括用于特定目的的地址空间。

- 一个别名区域将系统地址空间映射到程序 Flash 部分。Flash 区域别名专门用于引用 Flash 中的只读数据系数，同时仍在处理器内核中保留了完整的 Harvard 存储器组织，以支持并行指令提取（例如从 RAM）和数据访问（通过别名空间从 Flash 访问）。
- 处理器内核支持的位带功能使用映射到基本 RAM 和外设地址空间的别名区域。此功能将别名地址空间的各 32 位字映射到底层 RAM 或外设地址空间中的唯一一位，以支持处理器发起的单比特插入和提取操作。

表 5-1. 系统存储器映射

系统 32 位地址范围	目标从机	访问
0x0000_0000–0x07FF_FFFF	程序 Flash 和只读数据 (包括前 1024 字节中的异常向量)	所有主机
0x0800_0000–0x0FFF_FFFF	保留	—
0x1000_0000–0x1BFF_FFFF	保留	—
0x1C00_0000–0x1FFF_FFFF	SRAM_L: Lower SRAM (ICODE/DCODE)	所有主机
0x2000_0000–0x200F_FFFF ²	SRAM_U: 高位 SRAM 位带区	所有主机
0x2010_0000–0x21FF_FFFF	保留	—

下一页继续介绍此表...

表 5-1. 系统存储器映射 (继续)

系统 32 位地址范围	目标从机	访问
0x2200_0000–0x23FF_FFFF	另外命名为 SRAM_U 位带	仅限 Cortex-M4 内核
0x2400_0000–0x2FFF_FFFF	保留	–
0x3000_0000–0x33FF_FFFF ¹	程序 Flash 和只读数据	仅限 Cortex-M4 内核
0x3400_0000–0x3FFF_FFFF	保留	–
0x4000_0000–0x4007_FFFF	外设桥 0 的位带区 (AIPS-Lite0)	Cortex-M4 内核 & DMA
0x4008_0000–0x400F_EFFF	保留	–
0x400F_F000–0x400F_FFFF	通用输入/输出 (GPIO) 的位带区	Cortex-M4 内核 & DMA
0x4010_0000–0x41FF_FFFF	保留	–
0x4200_0000–0x42FF_FFFF	另外命名为外设桥 (AIPS-Lite) 位带	仅限 Cortex-M4 内核
0x4300_0000–0x43FD_FFFF	保留	–
0x43FE_0000–0x43FF_FFFF	另外命名为通用输入/输出 (GPIO) 位带	仅限 Cortex-M4 内核
0x4400_0000–0xDFFF_FFFF	保留	–
0xE000_0000–0xE00F_FFFF	专用外设	仅限 Cortex-M4 内核
0xE010_0000–0xFFFF_FFFF	保留	–

1. 此映射提供 Flash 的完整架构地址空间定义。基于特定器件实施的存储器物理大小，实际使用的地址区域可能更小。详情请参见 [Flash 存储器大小](#)。
2. 此范围取决于特定器件实施的 SRAM 量。详情请参见 [SRAM 大小](#)。

注

1. 对 AIPS-Lite 外设桥和通用输入/输出 (GPIO) 模块地址空间的访问权仅限于内核、DMA 和 EzPort。
2. ARM Cortex-M4 内核访问权限还包含通过调试接口访问。

5.2.1 位带别名区

SRAM_U、AIPS-Lite 及通用输入/输出 (GPIO) 模块资源驻留在 Cortex-M4 处理器位带区。

处理器还包括两个 32 MB 别名位带区，其与两个 1 MB 位带空间相关联。32 MB 空间中的各 32 位位置映射到位带区中的某一单独位。别名区的 32 位写操作对位带区目标位的作用与读-修改-写操作相同。

写入位带区别名的位 0 数值确定哪个数值写入目标位：

- 位 0 置位时，向目标位写入 1。
- 位 0 清零时，向目标位写入 0。

位带区别名中进行的 32 位读操作返回以下任意一种数值：

- 数值为 0x0000_0000 表示目标位清零
- 数值为 0x0000_0001 表示目标位置位

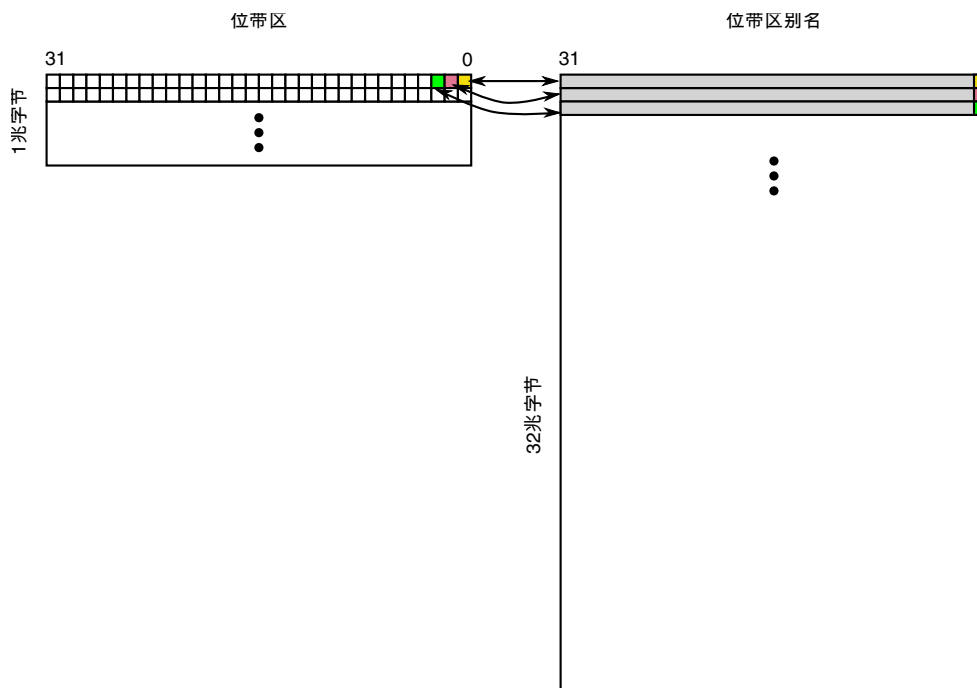


图 5-1. 位带区别名映射

注

位带区中的各位有一个等效位，通过别名位带区中的对应长字的位 0 可以操纵等效位。

5.2.2 Flash 访问控制简介

Flash 访问控制 (FAC) 是一个可由 NXP 或第三方配置的存储器优化保护方案，允许用户在为这些库提供可编程限制时使用软件库。Flash 存储器分为大小相等的区段，可为专有软件库提供保护。这些区段的保护受到控制，因为 FAC 对每一件访问片上 flash 存储器的事件进行循环的访问权限评估。可配置性允许保护更多的区段，同时支持两级供应商为器件添加专有软件。

Flash 访问控制对应于 ARM Cortex-M 系列产品支持的三种特权级别，最安全的状态——管理员/特权安全——对应于仅限执行和仅限管理员访问控制。用户不安全状态对应于未设置访问控制状态，中级状态中的用户安全对应于使用仅限执行的访问控制。

对此保护方案的控制在一性编程 NVM 位置中实现,可通过一性程序 Flash 命令操作进行配置。控制 FAC 的 NVM 位置不受“擦除所有块”的 Flash 命令和调试接口启动的批量擦除操作影响。

注

FAC 保护方案有 8 个 XACC 和 8 个 SACC 寄存器,最多可控制 64 段。对于 128 KB 或更小的程序 Flash,存储器划分为 32 段,受 4 个低位 XACC 和 SACC 寄存器控制。

5.3 Flash 存储器映射

Flash 存储器和 Flash 寄存器位于不同的基地址,如下图所示。相应的基地址在[系统存储器映射](#)中指定。

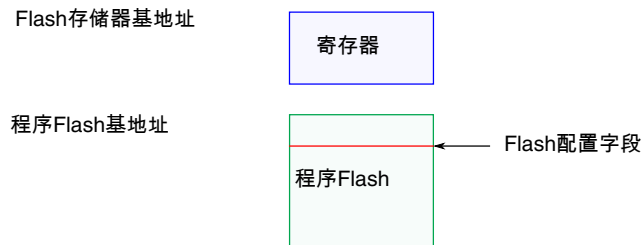


图 5-2. Flash 存储器映射

片上 Flash 在已分配 Flash 范围的一部分中实施,从而在起始地址为 0x0000_0000 的存储器映射中形成一个相邻块。有关支持范围的详情,请参见[Flash 存储器大小](#)。

如果访问 Flash 存储器范围时超出设备 Flash 大小,会导致总线周期终止,同时请求总线主机的相应响应生成错误。读取 Flash 存储的同时用 Flash 命令操作 Flash 存储资源,会导致读取冲突事件,产生总线响应错误。

5.3.1 可选的非易失性 IRC 用户调整说明

下列的非易失性地址 (4 字节) 是用于某些开发工具支持的定制 IRC 用户调整。出厂加载调整的可选的 IRC 调整可存储在此位置。要覆盖出厂调整值,用户软件必须将新值加载到 MCG 调整寄存器。以下地址仅供参考。

非易失性字节地址 (示例)	可选的 IRC 调整值
0x0000_03FC	保留
0x0000_03FD	保留
0x0000_03FE (位 0)	SCFTRIM
0x0000_03FE (位 4:1)	FCTRIM

下一页继续介绍此表...

非易失性字节地址（示例）	可选的 IRC 调整值
0x0000_03FE（位 5）	FCFTRIM
0x0000_03FF	SCTRIM

5.4 SRAM 存储器映射

片上 RAM 分为两个区域：SRAM_L 和 SRAM_U。RAM 实施方式为 SRAM_L 和 SRAM_U 在存储器映射中形成一个相邻块。详情参见 [SRAM 配置](#)。

如果访问 SRAM_L 和 SRAM_U 存储器范围时超出芯片 RAM 大小，会导致总线周期终止，同时请求总线主机的相应响应生成错误。

5.5 外设桥（AIPS-Lite）存储器映射

如果通过对应的 SIM 寄存器时钟选通控制位禁用了某模块，那么该模块 AIPS 槽也将被禁用。访问任何未实现或禁用的外设桥槽中的地址都会导致传输错误终止。

就通过外设桥进行编程模型访问而言，在 4 KB 槽中实现的通常只有很小一部分。访问未在外设中定义的地址会导致传输错误终止。

5.5.1 先写后读顺序和存储器操作所需的串行化

在某些情况下，执行后续操作前外设写入操作必须全部完成。这类情况示例有：

- 退出中断服务程序（ISR）
- 更改模式
- 配置函数

在这类情况下，应用软件必须执行先写后读的顺序，以确保正确执行存储器操作所需的串行化：

1. 对外设寄存器进行写操作。
2. 对已进行写操作的外设寄存器进行读操作，以便验证写操作。
3. 继续后续操作。

注

某个导致这类情况的原因是处理器写缓冲。处理器架构具有可编程配置位，可禁用写缓冲：ACTLR[DISDEFWBUF]。但是，禁用写缓冲很可能会降低系统性能，下降幅度远大于仅执行必要存储器串行化的情况。

5.5.2 外设桥 0 (AIPS-Lite 0) 存储器映射

表 5-2. 外设桥 0 槽分配

系统 32 位基地址	槽编号	模块
0x4000_0000	0	—
0x4000_1000	1	—
0x4000_2000	2	—
0x4000_3000	3	—
0x4000_4000	4	—
0x4000_5000	5	—
0x4000_6000	6	—
0x4000_7000	7	—
0x4000_8000	8	DMA 控制器
0x4000_9000	9	DMA 控制器传输控制描述符
0x4000_A000	10	—
0x4000_B000	11	—
0x4000_C000	12	—
0x4000_D000	13	—
0x4000_E000	14	—
0x4000_F000	15	—
0x4001_0000	16	—
0x4001_1000	17	—
0x4001_2000	18	—
0x4001_3000	19	—
0x4001_4000	20	—
0x4001_5000	21	—
0x4001_6000	22	—
0x4001_7000	23	—
0x4001_8000	24	—
0x4001_9000	25	—
0x4001_A000	26	—
0x4001_B000	27	—
0x4001_C000	28	—
0x4001_D000	29	—
0x4001_E000	30	—
0x4001_F000	31	Flash 控制器
0x4002_0000	32	Flash
0x4002_1000	33	DMA 通道多路复用器
0x4002_2000	34	—
0x4002_3000	35	—

下一页继续介绍此表...

表 5-2. 外设桥 0 槽分配 (继续)

系统 32 位基地址	槽编号	模块
0x4002_4000	36	FlexCAN 0
0x4002_5000	37	FlexCAN 1 (仅针对 KS22)
0x4002_6000	38	—
0x4002_7000	39	—
0x4002_8000	40	—
0x4002_9000	41	随机数生成器 (RNGA)
0x4002_A000	42	LPUART0
0x4002_B000	43	—
0x4002_C000	44	SPI 0
0x4002_D000	45	SPI 1
0x4002_E000	46	—
0x4002_F000	47	I2S 0
0x4003_0000	48	I2S 1
0x4003_1000	49	—
0x4003_2000	50	CRC
0x4003_3000	51	—
0x4003_4000	52	—
0x4003_5000	53	—
0x4003_6000	54	可编程延迟区块 (PDB)
0x4003_7000	55	周期性中断定时器 (PIT)
0x4003_8000	56	TPM 0
0x4003_9000	57	TPM 1
0x4003_A000	58	TPM 2
0x4003_B000	59	模数转换器 (ADC) 0
0x4003_C000	60	—
0x4003_D000	61	实时时钟 (RTC)
0x4003_E000	62	VBAT 寄存器文件
0x4003_F000	63	DAC0
0x4004_0000	64	低功耗定时器 (LPTMR)
0x4004_1000	65	系统寄存器文件
0x4004_2000	66	—
0x4004_3000	67	—
0x4004_4000	68	—
0x4004_5000	69	—
0x4004_6000	70	—
0x4004_7000	71	SIM 低功耗逻辑
0x4004_8000	72	系统集成模块 (SIM)
0x4004_9000	73	端口 A 复用控制
0x4004_A000	74	端口 B 复用控制

下一页继续介绍此表...

表 5-2. 外设桥 0 槽分配 (继续)

系统 32 位基地址	槽编号	模块
0x4004_B000	75	端口 C 复用控制
0x4004_C000	76	端口 D 复用控制
0x4004_D000	77	端口 E 复用控制
0x4004_E000	78	—
0x4004_F000	79	—
0x4005_0000	80	—
0x4005_1000	81	—
0x4005_2000	82	软件看门狗
0x4005_3000	83	—
0x4005_4000	84	—
0x4005_5000	85	—
0x4005_6000	86	—
0x4005_7000	87	—
0x4005_8000	88	—
0x4005_9000	89	—
0x4005_A000	90	—
0x4005_B000	91	—
0x4005_C000	92	—
0x4005_D000	93	—
0x4005_E000	94	—
0x4005_F000	95	FlexIO
0x4006_0000	96	—
0x4006_1000	97	外部看门狗
0x4006_2000	98	—
0x4006_3000	99	—
0x4006_4000	100	多用途时钟产生器 (MCG)
0x4006_5000	101	系统振荡器 (OSC)
0x4006_6000	102	LPI ² C 0
0x4006_7000	103	LPI ² C 1
0x4006_8000	104	—
0x4006_9000	105	—
0x4006_A000	106	UART 0
0x4006_B000	107	UART 1
0x4006_C000	108	UART 2
0x4006_D000	109	—
0x4006_E000	110	—
0x4006_F000	111	—
0x4007_0000	112	—
0x4007_1000	113	—

下一页继续介绍此表...

表 5-2. 外设桥 0 槽分配 (继续)

系统 32 位基地址	槽编号	模块
0x4007_2000	114	USB OTG FS/LS
0x4007_3000	115	模拟比较器 (CMP) / 6 位数模转换器 (DAC)
0x4007_4000	116	—
0x4007_5000	117	—
0x4007_6000	118	—
0x4007_7000	119	—
0x4007_8000	120	—
0x4007_9000	121	—
0x4007_A000	122	—
0x4007_B000	123	—
0x4007_C000	124	低漏电唤醒单元 (LLWU)
0x4007_D000	125	电源管理控制器 (PMC)
0x4007_E000	126	系统模式控制器 (SMC)
0x4007_F000	127	复位控制模块 (RCM)
0x400F_F000		GPIO 控制器

5.6 私有外设总线 (PPB) 存储器映射

PPB 是已定义 ARM 总线架构的一部分，提供对选择处理器-本地模块的访问。这些资源仅可从内核访问；其他系统主机无法访问。

表 5-3. PPB 存储器映射

系统 32 位地址范围	资源
0xE000_0000–0xE000_0FFF	仪器跟踪宏单元 (ITM)
0xE000_1000–0xE000_1FFF	数据观察点和跟踪 (DWT)
0xE000_2000–0xE000_2FFF	Flash 存储器块和断点 (FPB)
0xE000_3000–0xE000_DFFF	保留
0xE000_E000–0xE000_EFFF	系统控制空间 (SCS) (用于 NVIC 和 FPU)
0xE000_F000–0xE003_FFFF	保留
0xE004_0000–0xE004_0FFF	跟踪端口接口单元 (TPIU)
0xE004_1000–0xE004_1FFF	保留
0xE004_2000–0xE004_2FFF	保留
0xE004_3000–0xE004_3FFF	保留
0xE004_4000–0xE007_FFFF	保留
0xE008_0000–0xE008_0FFF	其他控制模块 (MCM)
0xE008_1000–0xE008_1FFF	保留
0xE008_2000–0xE00F_EFFF	保留
0xE00F_F000–0xE00F_FFFF	ROM 表 - 允许自动检测调试组件

第 6 章 时钟分布

6.1 简介

MCG 模块控制使用哪个时钟源来获得系统时钟。时钟生成逻辑将选定的时钟源分为不同的时钟域，包括用于系统总线主机、系统总线从机和 Flash 存储器的时钟。时钟生成逻辑还可实施特定模块时钟门控，从而允许模块时钟的打开和关闭。

系统的主要时钟都由 MCGOUTCLK 时钟生成。时钟生成电路提供多个时钟分频器，从而允许给设备的不同模块提供不同的时钟频率。这样的设计权衡考虑了性能和功耗。

有些模块（如 USB OTG 控制器）的特定时钟来自 IRC48MCLK、MCGPLLCLK 或 MCGFLLCLK 时钟。此外，还有各种其他特定模块时钟拥有其他可选择的时钟源。大部分模块的时钟选择都受 SIM 模块中 SOPT 寄存器的控制。

6.2 编程模型

系统时钟源的选择和多路复用通过 MCG 模块控制和编程来实现。该系统的时钟分频器设置和模块时钟门控通过 SIM 模块配置来实现。有关详细的寄存器和位说明，请参见以下章节。

6.3 高级设备时钟图

以下系统振荡器、MCG 和 SIM 模块寄存器控制下图所示的多路复用器、分频器和时钟门控：

	OSC	MCG	SIM
多路复用器	MCG_Cx	MCG_Cx	SIM_SOPT1, SIM_SOPT2
分频器	—	MCG_Cx	SIM_CLKDIVx
时钟门控	OSC_CR	MCG_C1	SIM_SCGCx

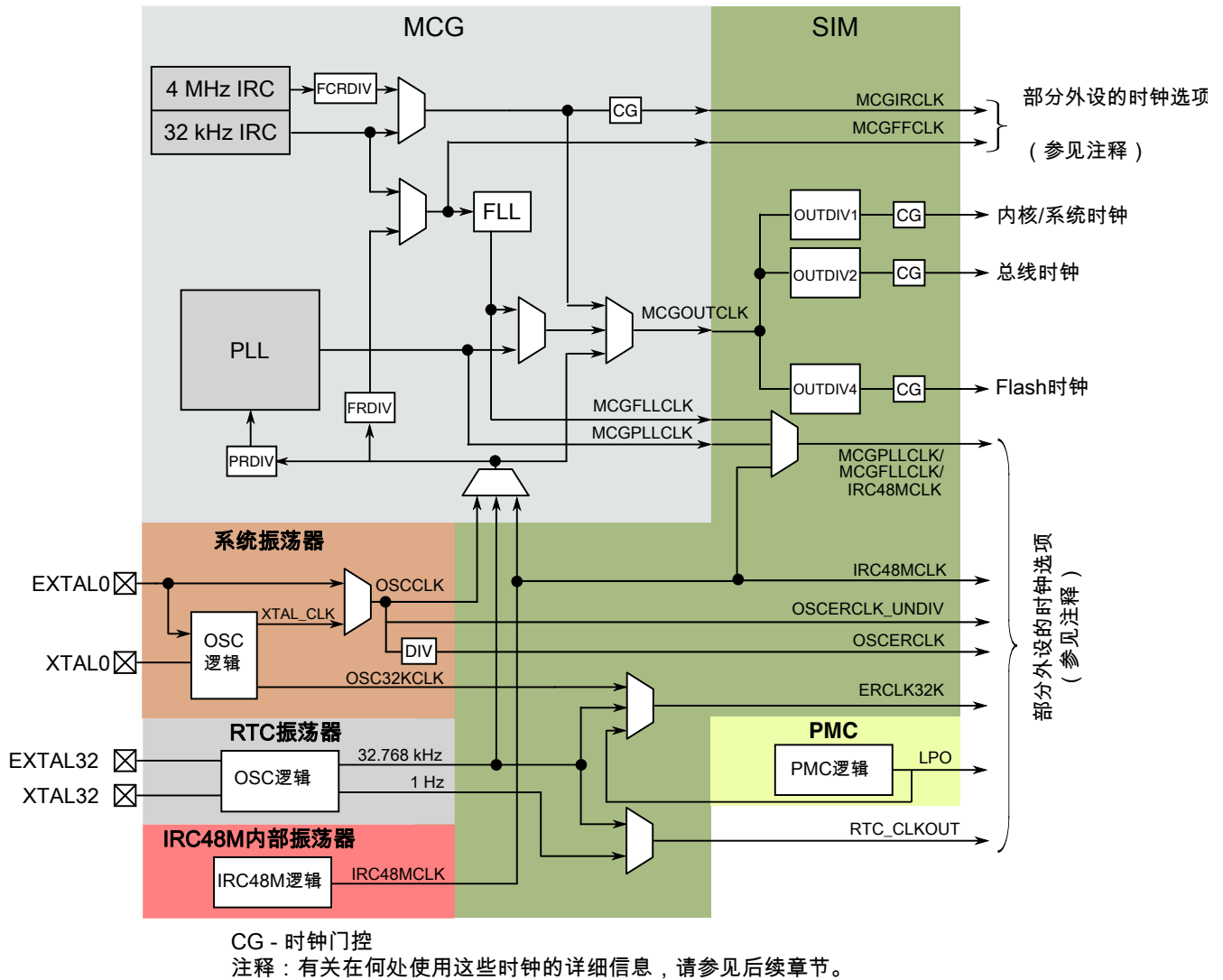


图 6-1. 时钟示意图

6.4 时钟定义

下表介绍了上一个框图中的时钟。

时钟名称	说明
内核时钟	由 OUTDIV1 分频的 MCGOUTCLK，为 ARM Cortex-M4 内核提供时钟。
平台时钟	由 OUTDIV1 分频的 MCGOUTCLK，为交叉开关和 NVIC 提供时钟。
系统时钟	由 OUTDIV1 分频的 MCGOUTCLK，直接为总线主机提供时钟。此外，此时钟用于 UART0 和 UART1。

下一页继续介绍此表...

时钟名称	说明
总线时钟	由 OUTDIV2 分频的 MCGOUTCLK, 为总线从机和外设 (不包括存储器) 提供时钟。
Flash 时钟	由 OUTDIV4 分频的 MCGOUTCLK, 为 Flash 存储器提供时钟。
MCGIRCLK	慢速或快速内部参考时钟的 MCG 输出。
MCGFFCLK	慢速内部参考时钟或分频 MCG 外部参考时钟的 MCG 输出。
MCGOUTCLK	为内核、系统、总线、和 Flash 时钟提供时钟源的 IRC、MCGFLLCLK、MCGPLLCLK 或 MCG 外部参考时钟的 MCG 输出。它也可以作为调试跟踪接口的时钟。
MCGFLLCLK	FLL 的 MCG 输出。MCGFLLCLK 可为部分模块提供时钟。
MCGPLLCLK	PLL 的 MCG 输出。MCGFLLCLK 或 MCGPLLCLK 可为部分模块提供时钟。
IRC48MCLK	内部 48 MHz 振荡器, 可用作 MCG 参考时钟并且还可为部分片上模块提供时钟。
OSCCLK	内部振荡器的系统振荡器输出, 或由 EXTAL 引脚直接输入的时钟源
OSCERCLK	源自 OSCCLK 的系统振荡器输出, 可为部分片上模块提供时钟。分频比可以为 1、2、4 或 8。
OSC32KCLK	系统振荡器 32 kHz 输出。
ERCLK32K	部分模块的时钟源可选为 OSC32KCLK 或 RTC 时钟。
RTC clock	RTC 振荡器输出, 用于 RTC 模块
LPO	PMC 1kHz 输出。

6.4.1 器件时钟汇总

下表提供了有关片上时钟的更多信息。

表 6-1. 时钟汇总

时钟名称	高速运行模式 时钟频率	运行模式 时钟频率	VLPR 模式 时钟频率	时钟源	当...时, 将禁用时钟
MCGOUTCLK	最高 120 MHz	最高 120 MHz	最高 4 MHz	MCG	除部分停止模式外的所有停止模式下, 以及 PLL 锁定过程中 (当 MCGOUTCLK 源自 PLL 时)
MCGFLLCLK	最高 100 MHz	最高 100 MHz	N/A	MCG	MCG 时钟控制未使能。在所有低功耗模式 (包括停止和 VLPx 模式) 下覆盖强制禁用。

下一页继续介绍此表...

表 6-1. 时钟汇总 (继续)

时钟名称	高速运行模式 时钟频率	运行模式 时钟频率	VLPR 模式 时钟频率	时钟源	当...时, 将禁用时钟
MCGPLLCLK	高达 120 MHz	高达 120 MHz	N/A	MCG	MCG 时钟控制未使能, 在停止模式下但 PLLSTEN=0, 或在 VLPS、LLS、和 VLLSx 模式下
内核时钟	最高 120 MHz	最高 80 MHz	最高 4 MHz	MCGOUTCLK 时钟分频器	在所有等待和停止模式下
系统时钟	最高 120 MHz	最高 80 MHz	最高 4 MHz	MCGOUTCLK 时钟分频器	在所有停止模式和运算操作模式下
总线时钟	最高 60 MHz	最高 50 MHz	最高 4 MHz	MCGOUTCLK 时钟分频器	在除部分 STOP2 模式外的所有停止模式以及运算操作模式下
Flash 时钟	最高 26.67 MHz	最高 26.67 MHz	BLPE 时最高 1 MHz, BLPI 时最高 800 kHz	MCGOUTCLK 时钟分频器	在除部分 STOP2 模式外的所有停止模式下
内部参考 (MCGIRCLK)	30-40 kHz 或 4 MHz	30-40 kHz 或 4 MHz	仅限 4 MHz	MCG	MCG_C1[IRCLKEN] 已清除, 停止或 VLPS 模式且 MCG_C1[IREFSTEN] 已清除, 或 LLS/VLLS 模式
外部基准时钟 (OSCERCLK)	最高 50 MHz (旁通), 30-40 kHz, 或 3-32 MHz (晶体)	最高 50 MHz (旁通) 30-40 kHz, 或 3-32 MHz (晶体)	最高 16 MHz (旁通), 30-40 kHz (低频晶体) 或最高 16 MHz (高频晶体)	系统 OSC	系统 OSC 的 OSC_CR[ERCLKEN] 已清除, 或停止模式且 OSC_CR[EREFS TEN] 已清除
外部参考 32 kHz (ERCLK32K)	30-40 kHz	30-40 kHz	30-40 kHz	系统 OSC 或 LPO 或 RTC OSC, 取决于 SIM_SOPT1[OSC32KSEL]	系统 OSC 的 OSC_CR[ERCLKEN] 已清除 或 RTC 的 RTC_CR[OSCE] 已清除
内部 48 MHz 时钟 (IRC48MCLK)	48 MHz	48 MHz	N/A	IRC48M	USB MCG 或 SIM 控制未使能。 在所有低功耗模式 (包括停止和 VLPx 模式) 下覆盖强制禁用。VLPS、LLSx、VLLSx。
RTC_CLKOUT	1 Hz 或 32 kHz	1 Hz 或 32 kHz	1 Hz 或 32 kHz	RTC 时钟	RTC_CLKOUT 在 LLS 和 VLLSx 模式下禁用。可以通过 SIM_SOPT1[OSC32KOUT]覆盖时钟, 以在所有低功耗模式下驱动输出 CLKOUT32K。

下一页继续介绍此表...

表 6-1. 时钟汇总 (继续)

时钟名称	高速运行模式 时钟频率	运行模式 时钟频率	VLPR 模式 时钟频率	时钟源	当...时, 将禁用时钟
CLKOUT32K	32 kHz	32 kHz	32 kHz	ERCLK32K - which is system OSC or LPO or RTC OSC depending on SIM_SOPT1[OSC32KSEL]	SIM_SOPT1[OSC32KOUT] not configured to drive ERCLK32K out.
LPO	1 kHz	1 kHz	1 kHz	PMC	VLLS0 时
USB FS 时钟	48 MHz	48 MHz	N/A	IRC48MCLK 或 MCGPLLCLK 或 MCGFLLCLK 带小数时钟分频器、或 USB_CLKIN	USB FS OTG 禁用
I2S 主时钟	最高 25 MHz	最高 25 MHz	最高 12.5 MHz	系统时钟、MCGPLLCLK、MCGFLLCLK、IRC48MCLK、OSCERCLK 带小数时钟分频器、或 I2S_CLKIN	I2S 禁用
跟踪时钟	最高 120 MHz	最高 120 MHz	最高 4 MHz	系统时钟或 MCGOUTCLK	跟踪禁用
LPUART0 时钟	最高 120 MHz	最高 100 MHz	最高 16 MHz	MCGPLLCLK 或 MCGFLLCLK 或 IRC48MCLK 或 MCGIRCLK 或 OSCERCLK	LPUART0 禁用
FlexIO 时钟	最高 120 MHz	最高 100 MHz	最高 16 MHz	MCGPLLCLK 或 MCGFLLCLK 或 IRC48MCLK 或 MCGIRCLK 或 OSCERCLK 或 I2S0_MCLK 或 系统时钟	FlexIO 禁用
LPI2C 时钟	最高 60 MHz	最高 60 MHz	最高 16 MHz	MCGPLLCLK 或 MCGFLLCLK 或 IRC48MCLK 或 MCGIRCLK 或 OSCERCLK	LPI2Cx 禁用
TPM 时钟	最高 60 MHz	最高 60 MHz	最高 4 MHz	MCGPLLCLK 或 MCGFLLCLK 或 IRC48MCLK 或 MCGIRCLK 或	所有 TPM 禁用

下一页继续介绍此表...

表 6-1. 时钟汇总 (继续)

时钟名称	高速运行模式 时钟频率	运行模式 时钟频率	VLPR 模式 时钟频率	时钟源	当...时, 将禁用时钟
				OSCERCLK	
FlexCAN 时钟	最高 60 MHz	最高 50 MHz	最高 4 MHz	总线时钟或 OSCERCLK	FlexCANx 禁用

6.5 内部时钟要求

时钟分频器通过 SIM 模块的 CLKDIV 寄存器编程。每个分频器的可编程范围是 1 分频到 16 分频。配置此器件的时钟时，必须满足下列要求：

1. 内核和系统时钟频率必须是 120 MHz 或更低 (HSRUN 下)、80 MHz 或更低 (RUN 下)。
2. 总线时钟频率必须设置为 60 MHz 或更低 (HSRUN 下)、50 MHz 或更低 (RUN 下)，且内核时钟的分频值为整数。内核时钟与总线时钟之比的最大值为 8。
3. Flash 时钟频率必须设置为 26.67 MHz 或更低，小于或等于总线时钟，且内核时钟的分频值为整数。内核时钟与 Flash 时钟之比的最大值为 8。

下面是该器件的一些较常用时钟配置：

方案 1:

时钟	频率
内核时钟	50 MHz
系统时钟	50 MHz
总线时钟	50 MHz
Flash 时钟	25 MHz

方案 2: 运行

时钟	频率
内核时钟	80 MHz
系统时钟	80 MHz
总线时钟	40 MHz
Flash 时钟	26.67 MHz

方案 3: 高速运行

时钟	频率
内核时钟	120 MHz
系统时钟	120 MHz
总线时钟	60 MHz
Flash 时钟	24 MHz

6.5.1 复位后的时钟分频器值

各时钟分频器通过 SIM 模块的 CLKDIV n 寄存器编程。Flash 存储器的 FTF_FOPT[LPBOOT] 位控制内核时钟、系统时钟、总线时钟和 Flash 时钟分频器的复位值，如下所示：

FTF_FOPT [LPBOOT]	内核/系统时钟	总线时钟	Flash 时钟	说明
0	0x7 (8 分频)	0x7 (8 分频)	0xF (16 分频)	低功耗启动
1	0x0 (1 分频)	0x0 (1 分频)	0x1 (2 分频)	快速时钟启动

这为用户提供灵活的低频率、低功耗启动选项。Flash 擦除状态默认值为快速时钟模式，因为在 Flash 擦除状态下，驻留在 Flash 中的低功耗启动 (FTF_FOPT[LPBOOT]) 位为逻辑 1。

要使能低功耗启动选项，应将 FTF_FOPT[LPBOOT] 设为 0。在复位过程中，若 LPBOOT 清零，系统将处于慢速时钟配置。发生系统复位时，时钟分频器回到此可配置复位状态。

6.5.2 VLPR 模式时钟

在 VLPR 模式下，无法更改时钟分频器。必须在进入 VLPR 模式之前对其进行编程，以确保：

- 内核/系统和总线时钟小于或等于 4 MHz，且
- Flash 存储器时钟小于或等于 1 MHz

注

当 MCG 处于 BLPI 且时钟源来自快速 IRC 时，必须设置时钟分频器控制 MCG_SC[FCRDIV] 和 SIM_CLKDIV1[OUTDIV4]，使所得 Flash 时钟标称频率为 800 kHz 或更低。在此情况下，正确配置的一个示例为 MCG_SC[FCRDIV]=000b 和 SIM_CLKDIV1[OUTDIV4]=0100b，得到 5 分频设置。

6.6 时钟门控

可使用 SIM 模块的 SCGC_x 寄存器单独对每个模块的时钟进行门控开启和关闭。可在任何复位后清除这些位，禁用相应模块的时钟以降低功耗。在初始化模块之前，设置 SCGC_x 寄存器中的相应位来开启时钟。关闭模块的时钟门控位前，务必先禁用相应的模块。

任何通过总线对外设的访问都会使其时钟禁用，从而产生错误终止。

6.7 模块时钟

下表汇总与各模块相关的时钟。

表 6-2. 模块时钟

模块	总线接口时钟	内部时钟	I/O 接口时钟
内核模块			
ARM Cortex-M4 内核	系统时钟	内核时钟	—
NVIC	系统时钟	—	—
DAP	系统时钟	—	—
ITM	系统时钟	—	—
κJTAG、JTAGC	—	—	JTAG_CLK
系统模块			
DMA	系统时钟	—	—
DMA 多路复用器	总线时钟	—	—
端口控制	总线时钟	LPO	—
交叉开关	系统时钟	—	—
外设桥	系统时钟	总线时钟、Flash 时钟	—
LLWU、PMC、SIM、RCM	Flash 时钟	LPO	—
模式控制器	Flash 时钟	—	—
MCM	系统时钟	—	—
EWM	总线时钟	LPO	—
WDOG 定时器	总线时钟	LPO	—
时钟模块			
MCG	Flash 时钟	MCGOUTCLK、MCGPLLCLK、MCGFLLCLK、MCGIRCLK、OSCCLK、RTC OSC、IRC48MCLK	—
OSC	总线时钟	OSCERCLK、OSCCLK、OSCERCLK_UNDIV、OSC32KCLK	—

下一页继续介绍此表...

表 6-2. 模块时钟 (继续)

模块	总线接口时钟	内部时钟	I/O 接口时钟
IRC48M	—	IRC48MCLK	—
存储器和存储器接口			
Flash 控制器	系统时钟	Flash 时钟	—
Flash 存储器	Flash 时钟	—	—
安全			
CRC	总线时钟	—	—
RNGA	总线时钟	—	—
模拟			
ADC	总线时钟	OSCERCLK、IRC48MCLK	—
CMP	总线时钟	—	—
DAC	总线时钟	—	—
定时器			
TPM	总线时钟	TPM 时钟	TPM_CLKIN0、TPM_CLKIN1
PDB	总线时钟	—	—
PIT	总线时钟	—	—
LPTMR	Flash 时钟	LPO、OSCERCLK、MCGIRCLK、ERCLK32K	—
RTC	Flash 时钟	EXTAL32	—
通信接口			
USB FS OTG	系统时钟	USB FS 时钟	—
DSPIC	总线时钟	—	DSPIC_SCK
LPI ² C	总线时钟	LPI ² C 时钟	I ² C_SCL
UART0、UART1	系统时钟	—	—
UART2	总线时钟	—	—
LPUART0	总线时钟	LPUART0 时钟	—
I ² S	总线时钟	I ² S 主机时钟	I ² S_TX_BCLK、I ² S_RX_BCLK
FlexCAN	总线时钟	FlexCAN 时钟	—
FlexIO	总线时钟	FlexIO 时钟	—
人机接口			
GPIO	平台时钟	—	—

6.7.1 PMC 1-kHz LPO 时钟

电源管理控制器 (PMC) 产生一个 1-kHz 时钟，在所有操作模式中均使能，包括除 VLLS0 外的所有低功耗模式。该 1-kHz 时钟源常被称为 LPO 时钟或 1-kHz LPO 时钟。

6.7.2 IRC 48MHz 时钟

集成的 48 MHz 内部参考时钟源 (IRC48MCLK) 可用于高速运行、运行、等待和停止操作模式。进入运行模式后，IRC48MCLK 也可用于运算操作模式、PSTOP2 和 PSTOP1 操作模式。当 MCU 转换至 VLPS、LLSx、和 VLLSx 低功耗模式时，IRC48MCLK 将被强制禁用。

注

IRC48MCLK 在停止模式下不会被强制禁用，因此应在进入停止模式之前由软件禁用，除非需要使用它。IRC48MCLK 在 VLPR 模式下不会被强制禁用，因此应在进入 VLPR 模式之前由软件禁用。

操作这些模式时，可通过以下任何控制设置使能 IRC48MCLK：

- USB 控制寄存器使能 — 当 USB_CLK_RECOVER_IRC_EN[IRC_EN]=1 时使能
- MCG 控制寄存器用于选择 IRC48 MHz 时钟（当 MCG_C7[OSCSEL]=10 时使能）并且在外部时钟模式下配置 MCG (PBE、BLPE、PEE、FBE 或 FEE) 或 MCG_C5[PLLCLKEN0] = 1。
- SIM 控制寄存器用于选择 IRC48 MHz 时钟 — 当 SIM_SOPT2[PLLFLSEL]=11 时使能

在 USB 器件应用中，可以在 USB 时钟恢复模式下使能 IRC48M 模块，在此模式下，将调谐内部 IRC48M 振荡器，以匹配从传入的 USB 数据流中提取的时钟。可通过此功能生成高精度 48MHz 时钟源，无需片上 PLL 或相关片外晶振电路。

如果断开 USB 器件与主机的连接，由于从 USB 数据流提取的时钟已断开，则 IRC48M USB 时钟恢复功能将停止调整 IRC48M 振荡器。将 USB 器件与 USB 主机断开后，48MHz 时钟源频率不会变化。如果将 IRC48M 时钟选作 PLL 的时钟源且 MCG_C7[OSCSEL]=10，则当 USB 器件连接至 USB 主机时，系统的时钟频率将变化，并开始时钟恢复调谐。

IRC48MCLK 也可用作：

- MCG 的振荡器参考 - 从中派生出内核、系统、总线和 Flash 时钟源
- ADC 备选时钟源
- 用于 LPUART 通信的时钟源
- 用于 I2S/SAI 通信的时钟源
- 用于 TPM 通信的时钟源
- 用于 FlexIO 通信的时钟源
- 用于 LPI2C 通信的时钟源

6.7.3 WDOG 时钟

按照下图所示，WDOG 可选两个时钟源。

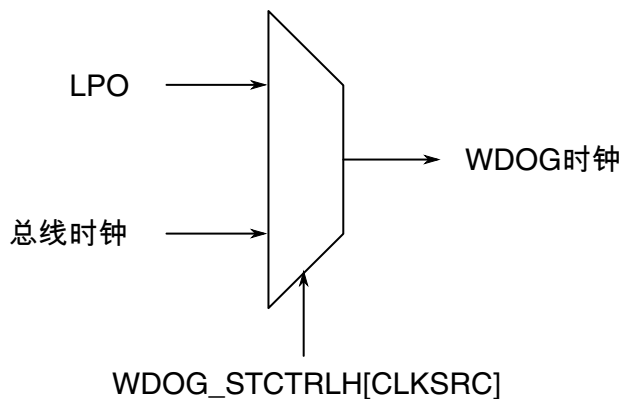


图 6-2. WDOG 时钟生成

6.7.4 调试跟踪时钟

可按照下图所示调试跟踪时钟源。

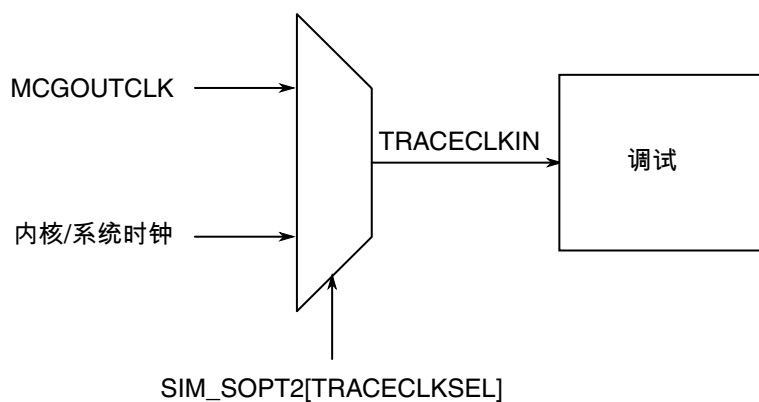


图 6-3. 跟踪时钟生成

6.7.5 PORT 数字滤波器时钟

各 PORT_x 模块中的数字滤波器可按下图进行计时。

注

在停止模式下，数字滤波器除非以 1kHz LPO 时钟源运行，否则它可以被旁路。

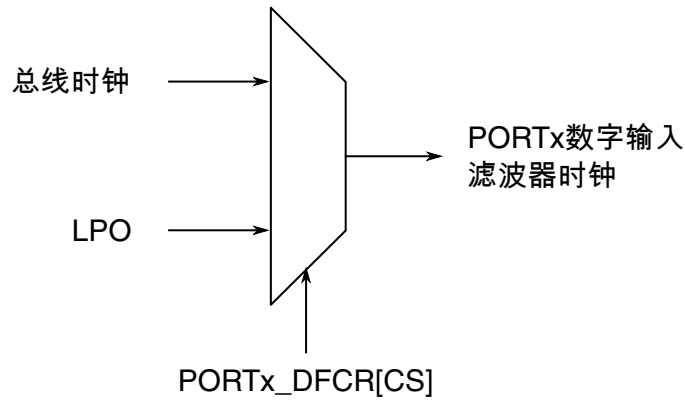


图 6-4. PORTx 数字输入滤波器时钟生成

6.7.6 LPTMR 时钟

LPTMR_x 模块中的预分频器和毛刺滤波器可按下图所示计时。

注

如果 LPTMR_x 将继续在所有要求的低功耗模式下运行，则选定时钟必须保持使能。

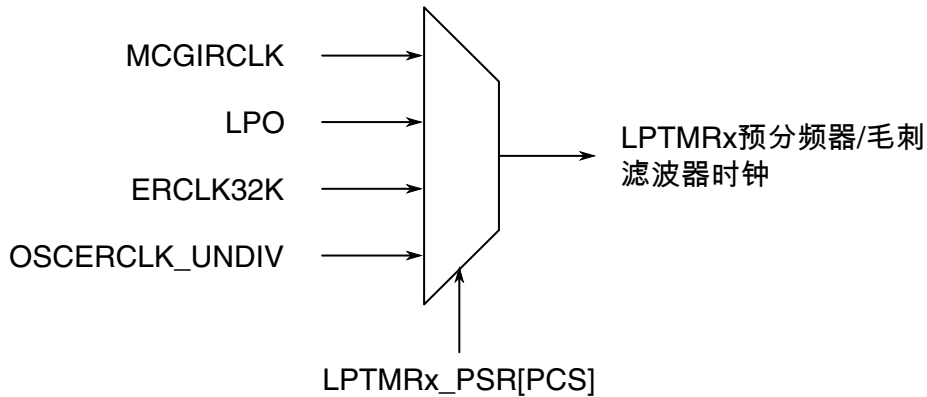


图 6-5. LPTMRx 预分频器/毛刺滤波器时钟生成

6.7.7 RTC_CLKOUT 和 CLKOUT32K 时钟

使能 RTC 时，RTC_CLKOUT 信号可以通过配置多路复用控制逻辑来驱动外部引脚，如下所示。

注

RTC_CLKOUT 在 LLS_x 和 VLLS_x 模式下禁用。

在 RTC_CLKOUT 信号为可选项时，由 SIM_SOPT1[OSC32KOUT] 控制的 CLKOUT32K 也可以覆盖该引脚现有的多路复用配置，用来驱动外部引脚。CLKOUT32K 功能可用于所有操作模式。在 VLLS0 模式下，仅 RTC 振荡器可用。

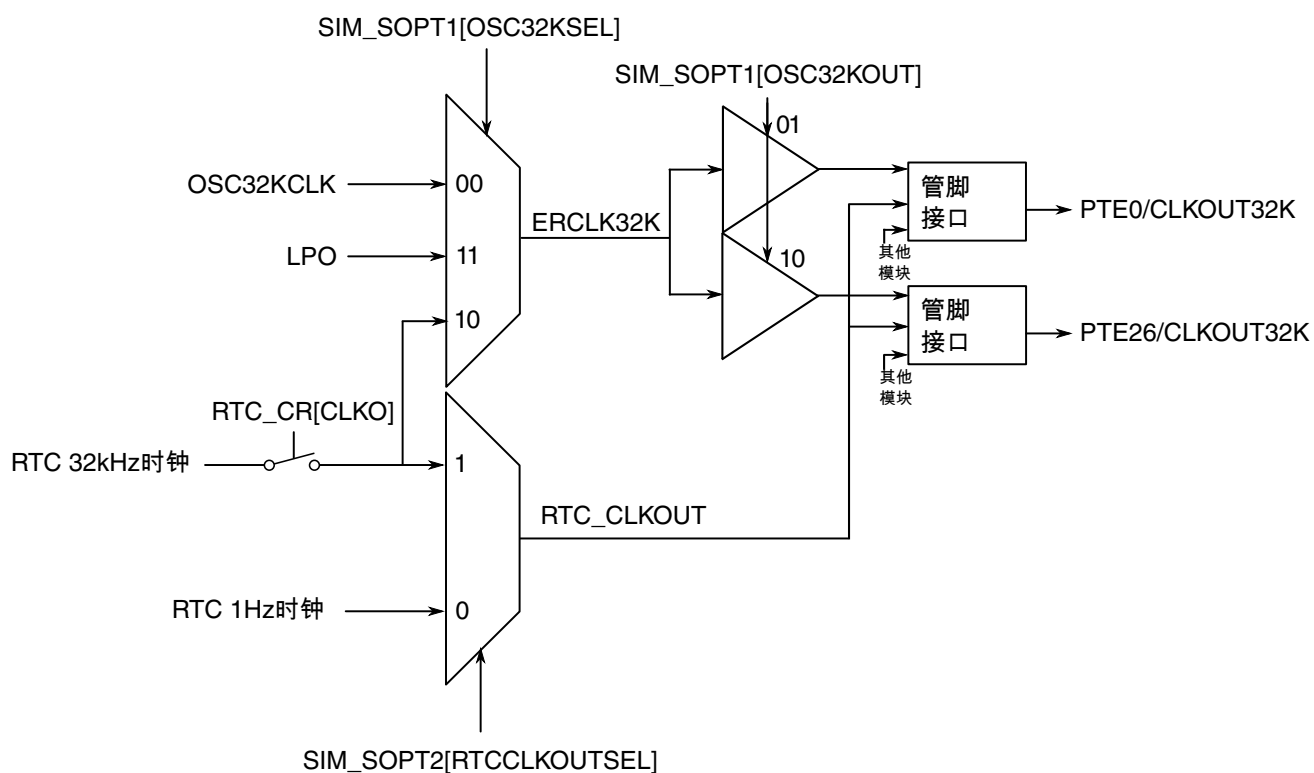


图 6-6. RTC_CLKOUT 和 CLKOUT32K 生成

6.7.8 USB FS OTG 控制器时钟

注

系统时钟的最小频率必须达到 20 MHz 才能使 USB FS OTG 控制器运行。

USB OTG 控制器还需要 48 MHz 时钟。时钟源选项如下所示。

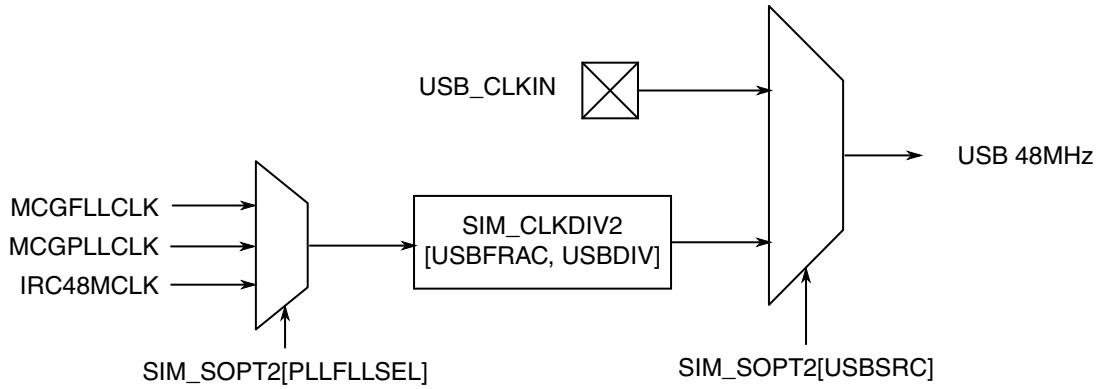


图 6-7. USB 48 MHz 时钟源

注

MCGFLLCLK 不符合认证所需的 USB 抖动特性。在 USB 时钟恢复功能已使能的 USB 器件操作中，IRC48MCLK 只能用作 USB 时钟源。

6.7.9 UART 时钟

UART0 和 UART1 模块采用内核/系统时钟运行，为这些模块提供更高的性能水平。所有其他 UART 模块采用总线时钟运行。

6.7.10 LPUART0 时钟

LPUART0 模块有一个可选的时钟，如下图所示。

注

如果 LPUART0 将继续在所有要求的低功耗模式下运行，则选定时钟必须保持使能。

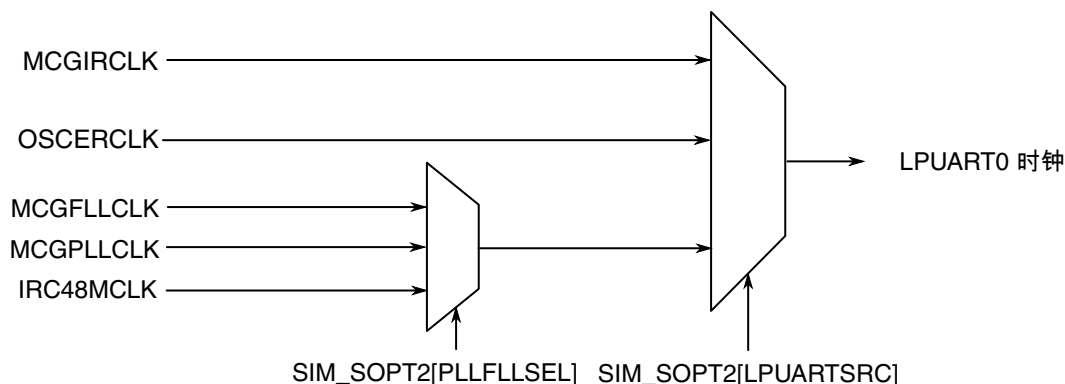


图 6-8. LPUART0 时钟生成

6.7.11 I²S/SAI 时钟

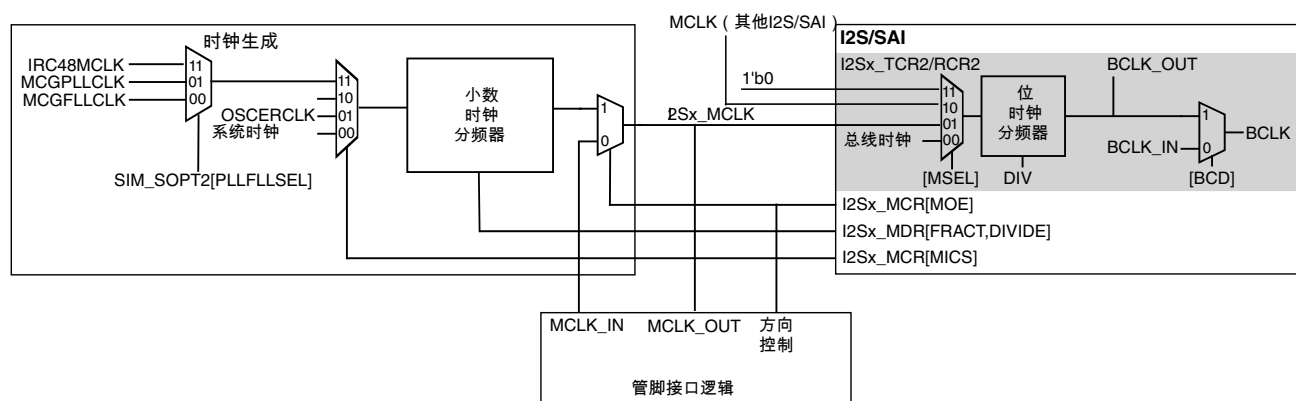
将接收器或发送器配置为内部生成的位时钟时，音频主机时钟（MCLK）用于生成位时钟。音频主机时钟还可作为引脚的输出或输入。发送器和接收器具有相同的音频主机时钟输入。

每个 SAI 外设均可控制输入时钟选择、引脚方向和一个音频主机时钟的分频比。

I²S/SAI 发送器和接收器支持异步位时钟（BCLK），该时钟可通过音频主机时钟从内部生成或从外部提供。该模块还支持接收器和发送器之间或两个单独 I²S/SAI 外设之间的同步操作选项。

发送器和接收器可以分别的选择总线时钟或者是音频主机时钟来生成位时钟。

下图所示为 MCLK 和 BCLK 时钟源选项。

图 6-9. I²S/SAI 时钟生成

6.7.12 FlexIO 时钟

FlexIO 定时器和移位器有一个可选的时钟，如下图所示。

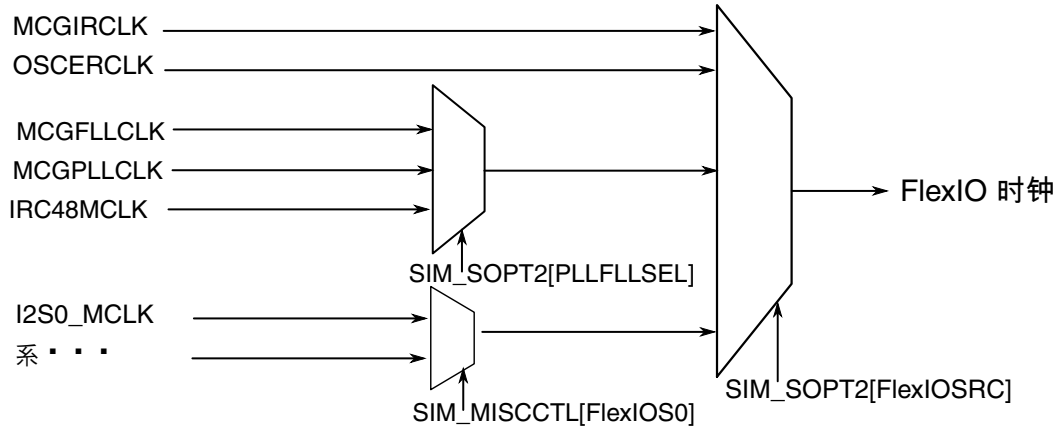


图 6-10. FlexIO 时钟生成

注

如果 FlexIO 将继续在要求的低功耗模式下运行，则选定时钟必须保持使能。

6.7.13 LPI2C 时钟

每个 LPI2C 模块有一个可选的时钟，如下图所示。

注

如果 LPI2C 将继续在所有要求的低功耗模式下运行，则选定时钟必须保持使能。

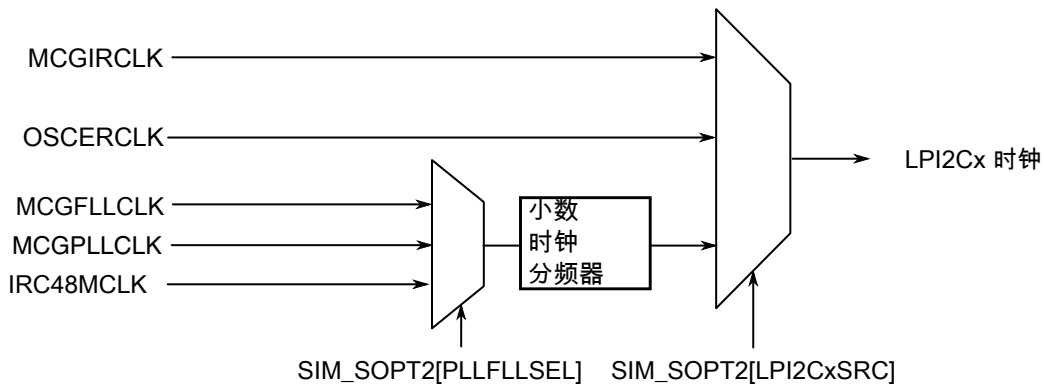


图 6-11. LPI2C 时钟生成

6.7.14 TPM 时钟

TPM 模块的计数器具有可选的时钟，如下图所示。

注

如果 TPM_x 将继续在所有所需的低功耗模式下运行，则选定时钟必须保持启用。

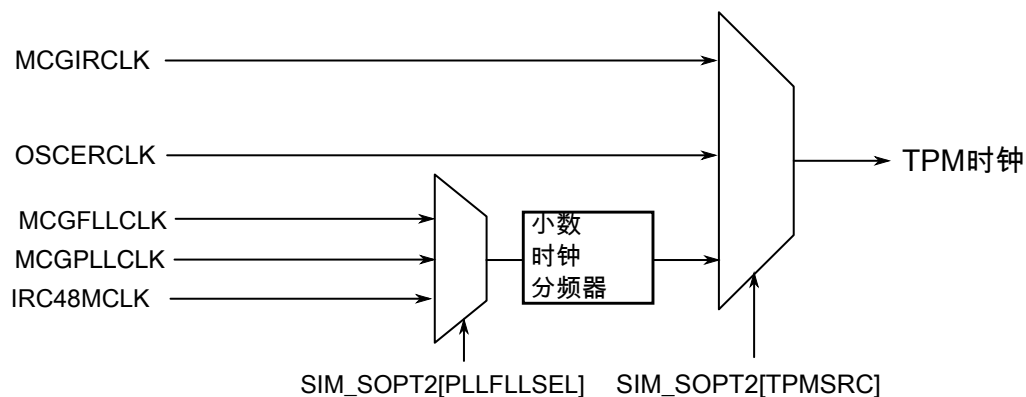


图 6-12. TPM 时钟生成

6.7.15 FlexCAN 时钟

FlexCAN 协议引擎的时钟可按下图所示进行选择。

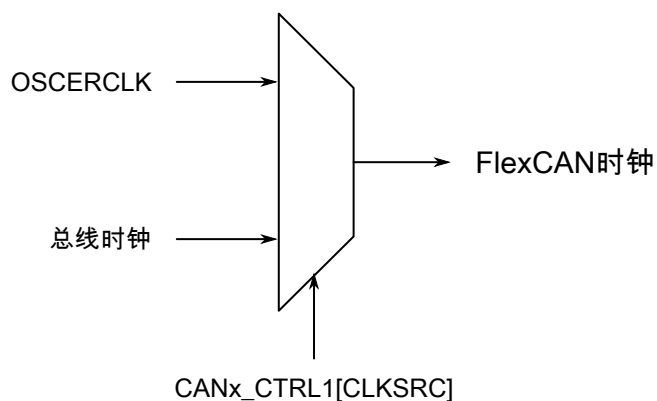


图 6-13. FlexCAN 时钟生成

第 7 章 复位与引导

7.1 简介

该 MCU 支持下列复位源：

表 7-1. 复位源

复位源	说明
POR 复位	<ul style="list-style-type: none">• 上电复位 (POR)
系统复位	<ul style="list-style-type: none">• 外部引脚复位 (PIN)• 低压检测 (LVD) 和高压检测 (HVD)• 计算机正常操作 (COP) 看门狗复位• 低漏电唤醒 (LLWU) 复位• 多用途时钟发生器时钟丢失 (LOC) 复位• 多用途时钟发生器锁定丢失 (LOL) 复位• 停止模式应答错误 (SACKERR)• 软件复位 (SW)• 锁定复位 (LOCKUP)• MDM DAP 系统复位
调试复位	<ul style="list-style-type: none">• JTAG 复位• nTRST 复位

每个系统复位源在系统复位状态 (SRS) 寄存器中都有一个相关位。有关寄存器详情，请参见[复位控制模块](#)。

在 CPU 执行代码的功能模式下，MCU 将退出复位。详情请参见[引导选项](#)。

7.2 复位

本节讨论基本复位机制和复位源。某些触发复位的模块可以配置成触发中断。如需更多信息，请参考独立的外设章节。

7.2.1 上电复位 (POR)

MCU 初始化上电或供电电压下降到低于上电复位重置电压电平 (V_{POR}) 时, POR 电路将导致 POR 复位条件。

随着供电电压升高, LVD 电路将 MCU 保持在复位状态, 直至电源超过 LVD 低阈值 (V_{LVDL})。在 POR 后置位 SRS0 寄存器中的 POR 和 LVD 位。

7.2.2 系统复位源

复位 MCU 提供一种从一组已知的初始条件重新开始处理的方法。系统复位开始时, 片上稳压器处于完全稳压状态, 系统时钟来源于内部基准时钟。当处理器退出复位时将执行以下操作:

- 从向量表偏移 0 处读取 SP (SP_main) 初始值
- 从向量表偏移 4 处读取开始 PC
- LR 设为 0xFFFF_FFFF

片上外设模块禁用且非模拟 I/O 引脚最初配置为禁用。分配了模拟功能的引脚在复位后会默认为其模拟功能。

在复位过程结束之后, JTAG 引脚的关联输入引脚配置为:

- TDI 上拉 (PU)
- TCK 下拉 (PD)
- TMS 上拉

关联的输出引脚配置为:

- 没有下拉或上拉电阻的 TDO

请注意, nTRST 信号最初被配置为禁用, 但是, 如果配置为其 JTAG 功能, 关联的输入引脚将被配置为:

- nTRST 上拉

7.2.2.1 外部引脚复位 (PIN)

在此器件上, \overline{RESET} 为专有引脚。此引脚为开漏引脚, 具有内部上拉电阻。 \overline{RESET} 可将器件从任意模式唤醒。在引脚复位过程中, RCM 的 SRS0[PIN] 位置位。

7.2.2.1.1 $\overline{\text{RESET}}$ 引脚滤波器

$\overline{\text{RESET}}$ 引脚滤波器支持用 1 kHz LPO 时钟和总线时钟进行滤波。

RCM_RPFC[RSTFLTSS]、RCM_RPFC[RSTFLTSRW]、和 RCM_RPFW[RSTFLTSEL] 控制此功能。请参见 RCM 章节。可通过芯片 POR 异步复位此滤波器。每个滤波器的复位值假定 $\overline{\text{RESET}}$ 引脚已取反。

对于 LPO 时钟仍继续工作的所有停止模式 (Stop、VLPS、LLS、VLLS3、VLLS2 和 VLLS1)，唯一的滤波选项为基于 LPO 的数字滤波器。滤波逻辑可以切换为旁通操作或继续滤波操作，具体取决于选择的滤波模式。进入 VLLS0 时， $\overline{\text{RESET}}$ 引脚滤波器被禁用和旁通。

LPO 滤波器具有固定的滤波值 3。由于输入数据上的同步器，因此，也存在相关延迟 (2 个周期)。因此，需要 5 个周期才能完成从低至高或从高至低的转换。

7.2.2.2 低压检测 (LVD) 和高压检测 (HVD)

该芯片包含一个用于管理低电压状况的系统，以便在供电电压波动期间保护存储器中的内容并控制 MCU 系统状态。此系统包含一个上电复位 (POR) 电路和一个用户可选择跳变电压的 LVD 电路。LVD 系统在 hsrun、正常运行、等待或停止模式下时钟使能。进入 VLPx、LLS 或 VLLSx 模式后，LVD 系统禁用。

可以将 LVD 配置为在检测到低电压情况时生成复位，方法是将 PMC 的 LVDSC1[LVDRE] 位置 1。低电压检测阈值由 PMC 的 LVDSC1[LVDV] 字段确定。LVD 复位后，LVD 系统将 MCU 保持在复位状态，直至供电电压升至低电压检测阈值之上。RCM 的 SRS0[LVD] 位在 LVD 复位或 POR 之后置位。

HVD 功能与 LVD 非常相似，并且共用相同的 RCM 状态位域 (RCM_SRS[LVD])。详情请参见 HVD 复位操作。

7.2.2.3 计算机正常运行 (COP) 看门狗定时器

计算机正常运行 (COP) 看门狗定时器 (WDOG) 通过定期的软件通信来对系统操作进行监控。此通讯一般被称为处理 (或刷新) COP 看门狗。如果此周期性刷新没有发生，则 WDOG 将产生一个系统复位的事件。COP 复位将引起 RCM 的 SRS0[WDOG] 位置位。

7.2.2.4 低漏电唤醒 (LLWU)

LLWU 为多个外部引脚、 $\overline{\text{RESET}}$ 引脚、以及多个内部外设提供将 MCU 从低漏电功耗模式中唤醒的方式。LLWU 模块仅在低漏电功耗模式下起作用。

- 在 LLS 模式下，仅通过 LLWU 的 $\overline{\text{RESET}}$ 引脚可生成系统复位。
- 在 VLLSx 模式下，所有使能输入至 LLWU 均可生成系统复位。

系统复位之后，LLWU 将保留指示最后唤醒输入源的标志，直到用户将其清除。

注

部分标志可在 LLWU 中清除，而另一部分需要在外设模块中才能清除。更多信息，请参见独立的外设章节。

7.2.2.5 多用途时钟产生器时钟丢失 (LOC)

MCG 模块支持外部参考时钟。

如果 MCG 模块中的 C6[CME] 位置位，则会使能时钟监控器。如果外部参考频率按照 MCG 模块中的 C2[RANGE] 字段所配置降至低于 f_{loc_low} 或 f_{loc_high} ，则 MCU 将复位。置位 RCM 的 SRS0[LOC] 位指示此复位源。

注

为了防止时钟复位事件意外丢失，在进入任何低功耗模块（包括 VLPR 和 VLPW）之前，应禁用所有时钟监控器。

7.2.2.6 MCG 锁定丢失 (LOL) 复位

MCG 包括一个 PLL 失锁检测器。当配置为 PEE 或完成锁定时将使能该检测器。如果置位 MCG 模块中的 MCG_C8[LOLRE] 位并且 PLL 锁定状态位 (MCG_S[LOLS0]) 置位，则 MCU 复位。置位 RCM_SRS0[LOL] 位指示此复位源。

注

如果芯片处于任何停止模式，则此复位源不会引起复位。

7.2.2.7 停止模式应答错误 (SACKERR)

在内核试图进入停止模式时，在 1025 个 1kHz LPO 时钟周期内，仍然有外设模块没有对该模式进行响应，则会产生该复位。

如果发生错误条件，模块对进入停止模式可能不会作出应答。该错误可能是由模块的外部时钟输入故障引起的。

7.2.2.8 软件复位 (SW)

通过设置 NVIC 应用中断及复位控制寄存器中的 SYSRESETREQ 位可以强制使设备产生软件复位。(有关寄存器字段的完整说明, 尤其是 VECTKEY 字段的要求, 请参见 ARM 的 NVIC 文档)。设置 SYSRESETREQ 可产生软件复位请求。软件复位对除调试模块以外的所有主要模块强制执行系统复位。软件复位将引起 RCM 的 SRS1[SW] 位置位。

7.2.2.9 锁定复位 (LOCKUP)

LOCKUP 可立即表明内核出现严重的软件错误。这是激活处理器内置系统状态保护硬件而产生的不可恢复的异常进而内核被锁定的结果。

LOCKUP 条件会导致系统复位, 还会引起 RCM 的 SRS1[LOCKUP] 位置位。

7.2.2.10 MDM-AP 系统复位请求

置位 MDM-AP 控制寄存器中的系统复位请求位, 以发起系统复位。这是通过 JTAG/SWD 接口进行复位的主要方法。在清除此位前系统一直保持复位状态。

设置 MDM-AP 控制寄存器中的内核保持复位位, 将内核保持在复位状态, 而系统其余模块退出复位状态。

7.2.3 MCU 复位

MCU 产生各种复位来复位不同模块。

7.2.3.1 VBAT POR

VBAT POR 在 VBAT POR 复位源上置位。它仅影响 VBAT 电源域内的模块: RTC 和 VBAT 寄存器文件。这些模块不受其他复位类型的影响。

7.2.3.2 仅限 POR

“仅限 POR”复位仅在 POR 复位源上置位。它将复位 PMC 和系统寄存器文件。

“仅限 POR”复位还会引起所有其他复位类型 (除 VBAT POR 外) 发生。

7.2.3.3 非 VLLS 的芯片 POR

“非 VLLS 的芯片 POR”复位在 POR 和 LVD 复位源上置位。它将复位部分 SMC 和 SIM。它还会复位 LPTMR。

“非 VLLS 的芯片 POR”复位还会引起这些复位发生：芯片 POR、非 VLLS 的芯片复位和芯片复位（包括早期芯片复位）。

7.2.3.4 芯片 POR

“芯片 POR”在 POR、LVD 和 VLLS 唤醒复位源上置位。它复位“复位引脚滤波器”寄存器和部分 SIM 和 MCG。

“芯片 POR”还会引起芯片复位（包括“早期芯片复位”）发生。

7.2.3.5 非 VLLS 的芯片复位

“非 VLLS 的芯片复位”复位在所有复位源上均置位，不通过 RESET_b 引脚发生的 VLLS 唤醒除外。它将复位部分 SMC、LLWU 以及在 VLLS 模式期间保持通电的其他模块。

“非 VLLS 的芯片复位”复位还会引起芯片复位（包括“早期芯片复位”）发生。

7.2.3.6 早期芯片复位

“早期芯片复位”在所有复位源上置位。它仅复位 Flash 存储器模块。它将在 Flash 存储器开始初始化之前撤除（“早于”芯片复位撤除）。

7.2.3.7 芯片复位

“芯片复位”在所有复位源上置位，仅在 Flash 初始化完成且 RESET_b 引脚也已失效后撤除。它将复位剩余模块（不被其他复位类型复位的模块）。

7.2.4 复位引脚

除无法通过 $\overline{\text{RESET}}$ 引脚发生的 VLLS 唤醒外，对于所有其他复位源， $\overline{\text{RESET}}$ 引脚将在至少 128 个总线时钟周期内被 MCU 拉低，直到 Flash 初始化完成。

Flash 初始化完成之后， $\overline{\text{RESET}}$ 引脚将被释放，且内部芯片复位将在 $\overline{\text{RESET}}$ 引脚被拉高后失效。 $\overline{\text{RESET}}$ 引脚从外部置位将延迟内部芯片复位的失效。

7.2.5 调试复位

以下章节详细介绍此器件上可用的调试复位。

7.2.5.1 JTAG 复位

执行某些 IR 代码时，JTAG 模块产生系统复位。当 EXTEST、HIGHZ 和 CLAMP 指令激活时，此功能复位置位。选择任何其他 IR 代码时，将释放 JTAG 模块的复位源。JTAG 复位将引起 RCM 的 SRS1[JTAG] 位置位。

7.2.5.2 nTRST 复位

nTRST 引脚有效时会引起 JTAG 逻辑复位。置位 nTRST 引脚允许调试器控制 TAP 控制器状态机（退出 LLS 或 VLLSx 后），无需复位调试模块的状态。

nTRST 引脚不会引起系统复位。

7.2.5.3 复位调试子系统

使用 SWJ-DP CTRL/STAT 寄存器中的 CDBGRSTREQ 位复位调试模块。但是，如下所述，使用 CDBGRSTREQ 位并不能复位所有和调试相关的寄存器。

CDBGRSTREQ 复位以下模块中和调试相关的寄存器：

- SWJ-DP
- AHB-AP
- TPIU
- MDM-AP (MDM 控制和状态寄存器)

CDBGRSTREQ 不会复位以下模块中和调试相关的寄存器：

- CM4 内核（内核调制寄存器：DHCSR、DCRSR、DCRDR、DEMCR）
- FPB
- DWT
- ITM
- NVIC
- 交叉总线开关

- AHB-AP¹
- 私有外设总线¹

7.3 启动

本小节介绍启动顺序，包括来源和选项。

7.3.1 启动源

此器件仅支持从内部 Flash 启动。任何二次启动都必须经过 Flash 中的初始化序列。

7.3.2 启动选项

在 CPU 执行代码的功能模式下，器件将退出复位。

7.3.3 FOPT 启动选项

Flash 存储器模块中的 Flash 选项 (FOPT) 寄存器允许用户定义 MCU 启动时的操作。该寄存器包含可从 Flash 配置字段的 NVM 选项字节加载的只读位。用户可以改写 Flash 中的选项字节，从而改变后续复位所用的 FOPT 值。有关选项字节编程的更多信息，请参见“Flash 存储器”一章。

复位时，MCU 利用 FOPT 寄存器位配置该器件，如下表所示。

注

选项字节中的保留位应保持默认已擦除状态（逻辑 1）。
FOPT[7:0] = 0x00 不是有效配置。若 Flash 配置字段中的 NVM 选项字节内容为 0x00，则向 FOPT 寄存器写入 0xFF。

表 7-2. Flash 选项寄存器位定义

位编号	字段	值	定义
7-6	保留		保留供将来扩展使用。
5	FAST_INIT		选择 POR、VLLSx、和任何系统复位时的初始化速度。
		0	慢速初始化。Flash 将以慢速初始化，其优点是可降低初始化期间的平均电流。恢复时长由 LPBOOT 设置决定的时钟分频器选择控制。

下一页继续介绍此表...

1. CDBGSTREQ 不会影响 AHB 源，因此私有外设总线上的调试资源在系统复位期间可用。

表 7-2. Flash 选项寄存器位定义 (继续)

位编号	字段	值	定义
		1	快速初始化。Flash 能以较快速度恢复，其代价是初始化期间的电流较大。
4-3	保留		保留供将来扩展使用。
2	NMI_DIS		NMI 功能的使能/禁用控制。
		0	NMI 中断始终被阻止。相关引脚继续默认为具有内部上拉使能的 NMI 引脚控制。
		1	NMI 引脚/中断复位默认为使能。
1	保留		保留供将来扩展使用。
0	LPBOOT		控制 SIM_CLKDIV1 寄存器中 OUTDIVx 的复位值。选择较大的分频值时，POR、VLLSx 恢复和复位序列期间以及退出复位后的平均功耗较低。如果未选择 FAST_INIT 选项，恢复时间也会延长。
		0	低功耗启动：SIM_CLKDIV1 寄存器中的 OUTDIVx 值在退出复位时自动配置为较高的分频值，以降低退出复位时的功耗。 <ul style="list-style-type: none"> 内核和系统时钟分频器 (OUTDIV1) 以及总线时钟分频器 (OUTDIV2) 为 0x7 (8 分频) Flash 时钟分频器 (OUTDIV4) 为 0xF (16 分频)
		1	正常启动：SIM_CLKDIV1 寄存器中的 OUTDIVx 值在退出复位时自动配置为较高的频率值，以提高退出复位时的工作频率。 <ul style="list-style-type: none"> 内核和系统时钟分频器 (OUTDIV1) 以及总线时钟分频器 (OUTDIV2) 为 0x0 (1 分频) Flash 时钟分频器 (OUTDIV4) 为 0x1 (2 分频)

7.3.4 启动序列

启动时，片上调节器使系统保持 POR 状态，直至输入供电电压高于 POR 阈值。系统继续保持在此静态状态下，直至内部调控的供电电压达到 LVD 确定的安全操作电压。模式控制器复位逻辑随后执行以下序列以退出复位。

1. 系统复位遵循一定的内部逻辑。 $\overline{\text{RESET}}$ 引脚被驱动至输出低电平，并且 MCG 按默认配置使能。
2. 所需时钟使能（内核时钟、系统时钟、Flash 时钟和任何未禁用时钟门控复位的总线时钟）。
3. 当复位控制逻辑继续将 $\overline{\text{RESET}}$ 引脚驱动输出为低电平时，内部逻辑上的系统复位继续保持，但 Flash 控制器将退出复位并开始初始化操作。
4. 在复位序列早期，NVM 选项字节将被读取和存储到 Flash 存储器模块的 FOPT 寄存器。如果将 LPBOOT 编程为备选时钟分频器复位值，则系统/内核时钟将切换至慢速时钟。
5. Flash 初始化完成之后，将释放 $\overline{\text{RESET}}$ 引脚。如果 $\overline{\text{RESET}}$ 继续为有效值（表示 $\overline{\text{RESET}}$ 引脚上升时间缓慢或外部驱动低电平），则系统继续保持在复位状态。一旦检测到 $\overline{\text{RESET}}$ 引脚处于高电平，内核时钟使能且系统解除复位状态。
6. 当系统退出复位时，处理器将初始化堆栈、程序计数器 (PC) 和连接寄存器 (LR)。处理器从向量表偏移 0 读取 SP (SP_main) 初始值。内核从向量表偏移

4 读取 PC 初始值。LR 设为 0xFFFF_FFFF。后续操作取决于 NMI 输入以及 Flash 存储器模块中的 FOPT[NMI_DIS] 字段。

- 如果 NMI 输入为高电平或在 NMI_DIS 字段中禁用 NMI 功能, 则 CPU 将在 PC 位置开始执行。
- 如果 NMI 输入为低电平且在 NMI_DIS 字段中启用 NMI 功能, 则会导致 NMI 中断。处理器将进入异常处理并读取向量表偏移量 8 中的 NMI 中断处理程序地址。CPU 将在 NMI 中断处理程序处开始执行。

后续系统复位遵循相同的复位流程。

第 8 章 电源管理

8.1 简介

本章介绍多种芯片电源模式以及各模块在这些模式下的功能。

8.2 时钟模式

此处信息介绍了本器件支持的各种时钟模式。

8.2.1 部分停止

部分停止是一个时钟选项，可替代停止模式，在 SMC 停止控制寄存器 (SMC_STOPCTRL) 中配置。仅部分模块进入停止模式，某些功能仍然有效，代价是功耗较高。部分停止可从运行模式或 VLP 运行模式进入。

配置为 PSTOP2 时，仅内核和系统时钟使能，总线时钟保持活动状态。由系统时钟提供时钟的总线主机和总线从机进入停止模式，但由总线时钟提供时钟的总线从机保持运行（或 VLP 运行）模式。MCG 中的时钟产生器和 PMC 中的片上稳压器也保持运行（或 VLP 运行）模式。复位、由系统时钟提供时钟的总线主机或总线从机所产生的异步中断，或者由总线时钟提供时钟的总线从机所产生的同步中断，可以启动从 PSTOP2 退出的程序。经过配置，也可以利用 DMA 请求（使用异步 DMA 唤醒）在 DMA 传输期间退出部分停止模式，然后器件再回到 PSTOP2 模式。

配置为 PSTOP1 时，系统时钟和总线时钟均使能。所有总线主机和总线从机都进入停止模式，但 MCG 中的时钟产生器和 PMC 中的片上稳压器保持运行（或 VLP 运行）模式。复位或者总线主机/总线从机产生的异步中断，可以启动从 PSTOP1 退出的程序。经过配置，也可以利用异步 DMA 请求在 DMA 传输期间退出部分停止模式，然后器件再回到 PSTOP1 模式。

PSTOP1 在功能上与停止模式相似，但唤醒速度更快，代价是功耗较高。另一个优点是它让所有 MCG 时钟保持使能，这对某些在停止模式下仍然工作的异步外设十分有用。

8.2.2 DMA 唤醒

在停止模式下，DMA 可配置为通过 DMA 请求信号来唤醒 MCU。可以为每个 DMA 通道配置唤醒，并且该唤醒在运算操作、PSTOP、STOP 和 VLPS 低功耗模式下均受支持。

在 PSTOP、STOP 或 VLPS 模式下检测到 DMA 唤醒时，器件将发起正常退出低功耗模式。包括还原片上稳压器和内部电源开关，以使能 MCG 中的时钟产生器，从而使能系统和总线时钟（但不包括内核时钟）并取消总线主机和总线从机的停止模式信号。唯一的差别在于 CPU 仍保持处于低功耗模式且 CPU 时钟禁用。

在运算操作模式，DMA 唤醒将发起正常退出运算操作模式。这包括使能时钟并取消总线主机和总线从机的停止模式信号。在运算操作过程始终使能内核时钟。

由于 DMA 唤醒将使能时钟并取消总线主机和总线从机的停止模式信号，因此，软件需要确保未涉及 DMA 唤醒和传输的总线主机和从机保持处于已知状态。可以通过在进入低功耗模式之前禁用这些模块或者置位选定模块中的等待使能位完成此任务。

DMA 请求发起唤醒取消且 DMA 完成当前传输之后，器件将转换回原来的低功耗模式。这包括请求所有非 CPU 总线主机进入停止模式，然后请求总线从机进入停止模式。在停止和 VLPS 模式下，MCG 和 PMC 随后也会进入其相应的模式。

注

如果请求的 DMA 传输无法引起 DMA 请求无效，则器件将保持处于较高功耗状态，直到完全退出低功耗模式。

如果 DMA 请求在停止模式进入序列过程中发出（或者在 DMA 唤醒过程中请求发出时重新进入）并引起 SMC 设置他的停止中止标志位，则使能的 DMA 唤醒可导致中止进入低功耗模式。DMA 唤醒完成之后，必须重新启动进入低功耗模式。

DMA 唤醒过程中发生的中断可导致立即退出低功耗模式（在运算操作模式，这是一个可选项），而不会影响 DMA 传输。

可以通过同步 DMA 请求或异步 DMA 请求产生 DMA 唤醒。一般来说，如果一个外设在停止模式可以产生同步 DMA 请求，也支持异步 DMA 中断，那么它就能够产生 DMA 异步请求，但并非所有外设在停止模式下均可产生异步 DMA 请求。

8.2.3 运算操作

运算操作是一种执行或仅运算操作模式，它使能 CPU 并且可完全访问 SRAM 和 Flash 读取端口，但其他总线主机和总线从机将会处于停止模式。运算操作可从运行模式、HSRUN 模式或 VLP 运行模式下进入。

注

切勿在未先退出运算操作之前进入任何停止模式。

因为运算操作重用了停止模式逻辑（包括在总线从机之前禁用总线主机的分阶段进入），任何可在停止模式下保持运行的总线主机或总线从机也可在运算模式下保持运行，包括产生异步中断和 DMA 请求。在运行模式下使能运算操作时，总线主机和总线从机的模块功能与在停止模式下相同。在 VLP 运行模式下使能运算操作时，总线主机和总线从机的模块功能与在 VLPS 模式下相同。尽管 Flash 寄存器接口禁用，但 MCG、PMC、SRAM 和 Flash 读取端口不受运算操作影响。

在运算操作过程中，禁用 AIPS 外设空间并且尝试的访问将产生总线错误。在运算操作过程中保持可访问私有外设总线 (PPB)，包括 MCM、系统控制空间 (SCS) (对于 NVIC)、和 SysTick。尽管支持访问 GPIO 寄存器，但 GPIO 端口数据输入寄存器不会返回有效的数据，因为至端口控制和中断模块的时钟禁用。在写 GPIO 端口数据输出寄存器后，可以控制配置为输出引脚的 GPIO 端口。

运算操作由 MCM 中的 CPO 寄存器控制，并且只有 CPU 能够访问。置位或清除 MCM 中的 CPOREQ 位将进入或退出运算操作。还可将运算操作配置为在检测到中断时自动退出，这是为了能够响应大部分的中断服务所要求的。仅内核系统中断（异常，包括 NMI 和 SysTick）和任何边沿敏感型中断可以在不退出运算操作的情况下运行。

当进入运算操作模式时，CPOACK 状态位指示什么时间完全进入。在运行模式下退出运算操作时，CPOACK 状态位将立即清零。在 VLP 运行模式下退出运算操作时，退出将被延迟，以允许 PMC 处理功耗变化。此延迟意味着将轮询 CPOACK 位，以确定什么时间可以访问 AIPS 外设空间并且不会生成总线错误。

在运算操作过程中也支持 DMA 唤醒且会引起 CPOACK 状态位清零，并且可以在 DMA 唤醒过程中访问 AIPS 外设空间。完成 DMA 唤醒之后，将转换回运算操作。

8.2.4 外设休眠

部分外设支持外设休眠模式，可在低功耗模式下使用寄存器位禁用此外设。还可以在外设休眠过程中通过 SIM 中的寄存器位将 Flash 存储器置于低功耗状态。

外设休眠包括下列所有操作模式。

- CPU 处于等待模式。

- CPU 处于停止模式，包括进入序列和 DMA 唤醒过程中。
- CPU 处于运算操作中，包括进入序列和 DMA 唤醒过程中。

因此，外设休眠可用于在 WAIT 或 VLPW 模式下禁用选定的总线主机或从机。它还可用于在进入任何停止模式（或运算操作）时立即禁用选定的总线主机，而无需等待总线主机识别进入到停止序列的那一部分了。最后，它还可用于禁用在 DMA 唤醒过程中保持未激活的选定总线主机或从机。

如果在 WAIT 和 PSTOP 模式下不需要访问 Flash 存储器，则 Flash 休眠可用于降低功耗，但执行 Flash 中的代码和向量时会稍微延迟唤醒。它还可在执行 SRAM 中的代码和向量时用于降低运算操作过程中的功耗。

8.2.5 时钟门控

为了节能，多数模块的时钟均可通过 SIM 模块中的 SCGCx 寄存器关闭。可在任何复位后清除这些位，从而将禁用相应模块的时钟。在初始化模块之前，将 SCGCx 寄存器中的相应位置位以使能时钟。关闭时钟前，务必禁用该模块。有关更多信息，请参见时钟分布和 SIM 章节。

8.3 功耗模式描述

电源管理控制器（PMC）为用户提供了多种功耗选项，用户因此可针对所需的功能优化功耗。

根据用户应用的停止要求，可以使用多种不同停止模式，它们提供特定逻辑和/或存储器的状态保持、部分降耗或完全降耗。I/O 状态在所有工作模式下保持。下表对各种功耗模式进行了比较。

对于 Run 和 VLPR 模式，存在对应的等待和停止模式。等待模式与 ARM 睡眠模式相似。停止模式（VLPS、STOP）与 ARM 深度睡眠模式相似。当不需要最高总线频率以满足应用需求时，超低功耗运行（VLPR）工作模式可以显著减少运行时功耗。

不支持从 HSRUN 直接进入停止模式，需要在尝试进入停止模式之前先转换至运行模式。

三种主要的工作模式为：运行、等待和停止。WFI 指令激活芯片的等待和停止模式。基本模式通过多种方式增强，可以根据应用需要，提供更低的功耗。

表 8-1. 芯片功耗模式

芯片模式	描述	内核模式	正常恢复方法
正常运行	复位后的默认状态；片上稳压器开启。	运行	-
High Speed run	允许的芯片最高性能。与正常运行模式相比，在此状态下，MCU 可以更高频率运行。	运行	-
通过 WFI 指令实现 Normal Wait (正常等待)	内核处于睡眠模式时允许外设运行以降低功耗。NVIC 依然监测中断；继续提供外设时钟。	睡眠	中断
通过 WFI 指令实现 Normal Stop (正常停止)	将芯片置于静态。是既能保持所有寄存器，同时还能维持 LVD 保护的最低功耗模式。NVIC 禁用；AWIC 用于从中断唤醒；外设时钟停止。	深度睡眠	中断
VLPR (超低功耗运行)	片内电压调节器处于低功耗模式，仅提供芯片低频运行所需的电压。降频 Flash 访问模式 (1 MHz)；LVD 关闭；内部振荡器为内核、总线和外设时钟提供低功耗 4 MHz 源。	运行	-
通过 WFI 指令实现 VLPW (超低功耗等待)	与 VLPR 相同，但内核处于睡眠模式下，以便进一步降低功耗；NVIC 仍然监测中断 (FCLK 开启)。片内电压调节器处于低功耗模式，仅提供芯片低频运行所需的电压。	睡眠	中断
通过 WFI 指令实现 VLPS (超低功耗停止)	将芯片置于静态，关闭 LVD 操作。ADC 和引脚中断可以工作的最低功耗模式。外设时钟停止，但 LPTimer、RTC、CMP、DAC 可以使用。NVIC 禁用 (FCLK 关闭)；AWIC 用来从中断唤醒。片内电压调节器处于低功耗模式，仅提供芯片低频运行所需的电压。所有 SRAM 均处于工作状态 (保留内容，保持 I/O 状态)。	深度睡眠	中断
LLS3 (低漏电停止 3)	状态保持功耗模式。大多数外设都处于状态保持模式 (时钟停止)，但 LLWU、LPTimer、RTC、CMP、DAC 可以使用。NVIC 禁用；LLWU 用来唤醒。 注：LLWU 中断不可通过中断控制器屏蔽，以避免系统无法在 LLS 恢复时完全退出停止模式的现象。 所有 SRAM 均处于工作状态 (保留内容，保持 I/O 状态)。	深度睡眠	唤醒中断 ¹
LLS2 (低漏电停止 2)	状态保持功耗模式。大多数外设都处于状态保持模式 (时钟停止)，但 LLWU、LPTimer、RTC、CMP、DAC 可以使用。NVIC 禁用；LLWU 用来唤醒。 注：LLWU 中断不可通过中断控制器屏蔽，以避免系统无法在 LLS 恢复时完全退出停止模式的现象。 SRAM_U 的一部分保持上电状态 (保留内容，保持 I/O 状态)。	深度睡眠	唤醒中断 ²
VLLS3 (极低漏电停止 3)	大多数外设禁用 (时钟停止)，但 LLWU、LPTimer、RTC、CMP、DAC 可以使用。NVIC 禁用；LLWU 用来唤醒。 SRAM_U 和 SRAM_L 保持上电 (保留内容，保持 I/O 状态)。	深度睡眠	唤醒复位
VLLS2 (极低漏电停止 2)	大多数外设禁用 (时钟停止)，但 LLWU、LPTimer、RTC、CMP、DAC 可以使用。NVIC 禁用；LLWU 用来唤醒。 SRAM_L 断电。SRAM_U 的一部分保持上电状态 (保留内容，保持 I/O 状态)。	深度睡眠	唤醒复位 ³
VLLS1 (极低漏电停止 1)	大多数外设禁用 (时钟停止)，但 LLWU、LPTimer、RTC、CMP、DAC 可以使用。NVIC 禁用；LLWU 用来唤醒。 所有 SRAM_U 和 SRAM_L 均断电。32 字节系统寄存器文件和 32 字节 VBAT 寄存器文件保持上电，以便保存客户关键型数据。	深度睡眠	唤醒复位 ³

下一页继续介绍此表...

表 8-1. 芯片功耗模式 (继续)

芯片模式	描述	内核模式	正常恢复方法
VLLS0 (极低漏电停止 0)	大多数外设禁用 (时钟停止), 但 LLWU 和 RTC 可以使用。NVIC 禁用; LLWU 用来唤醒。 所有 SRAM_U 和 SRAM_L 均断电。32 字节系统寄存器文件和 32 字节 VBAT 寄存器文件保持上电, 以便保存客户关键型数据。 POR 检测电路可选断电。	深度睡眠	Wakeup Reset ³
BAT (仅备用电池)	除了 VBAT 电源外, 此芯片将掉电。RTC 和 32 字节 VBAT 寄存器文件保持上电, 以便保存客户关键型数据。	关断	上电顺序

1. 通过执行 LLWU 中断服务程序恢复正常运行模式操作。
2. 通过执行 LLWU 中断服务例程恢复正常运行模式操作。
3. 遵循复位流程, 置位 NVIC 的 LLWU 中断标志。

8.4 进入和退出功耗模式

WFI 指令可为芯片调用等待和停止模式。处理器通过中断退出低功耗模式。The [嵌套矢量中断控制器 \(NVIC\)](#) 描述中断操作和哪些外设能引起中断。

注

WFE 指令对进入低功耗模式会产生副作用, 但这并非有意为之。有关 WFE 指令的更多信息, 请参见 ARM 文档。

从 VLLS_x 恢复需通过复位事件来唤醒。芯片通过复位、已使能引脚或已使能模块从 VLLS_x 唤醒。来源列表请参见 LLWU 配置部分中的“LLWU 输入”表。

从 VLLS_x 唤醒的流程是通过复位进行。RCM 中的 SRS 寄存器的唤醒位置位, 表示芯片正从低功耗模式恢复。代码已开始执行, 但 I/O 引脚仍处于进入低功耗模式前的状态, 系统振荡器和 MCG 寄存器复位 (即使在进入 VLLS_x 之前已设置 EREFSTEN 也是如此)。软件必须向 PMC 模块中的稳压器状态和控制寄存器的 ACKISO 位写入 1, 以清除此保持状态。

注

为避免引脚发生不需要的转换, 软件必须在释放保持状态之前, 将 I/O 引脚重新初始化到进入低功耗模式前的状态。

在 VLLS_x 模式下, 若振荡器配置为继续运行, 则在 ACKISO 位清零之前, 必须重新配置振荡器。VLLS_x 恢复之后, MCG 内的振荡器配置被清除; 当 ACKISO 清零时, 振荡器会停止, 除非重新配置该寄存器。

8.5 功耗模式转换

下图显示了功耗模式转换。任何复位都会使芯片返回正常运行状态。在运行、等待和停止模式下，使能主动功率调节。与正常模式相比，VLPx 模式是一种低功耗操作模式。VLPR 和 VLPW 模式下频率受限。从应用要保留的逻辑或存储器数来说，LLS 和 VLLSx 模式为最低功耗停止模式。

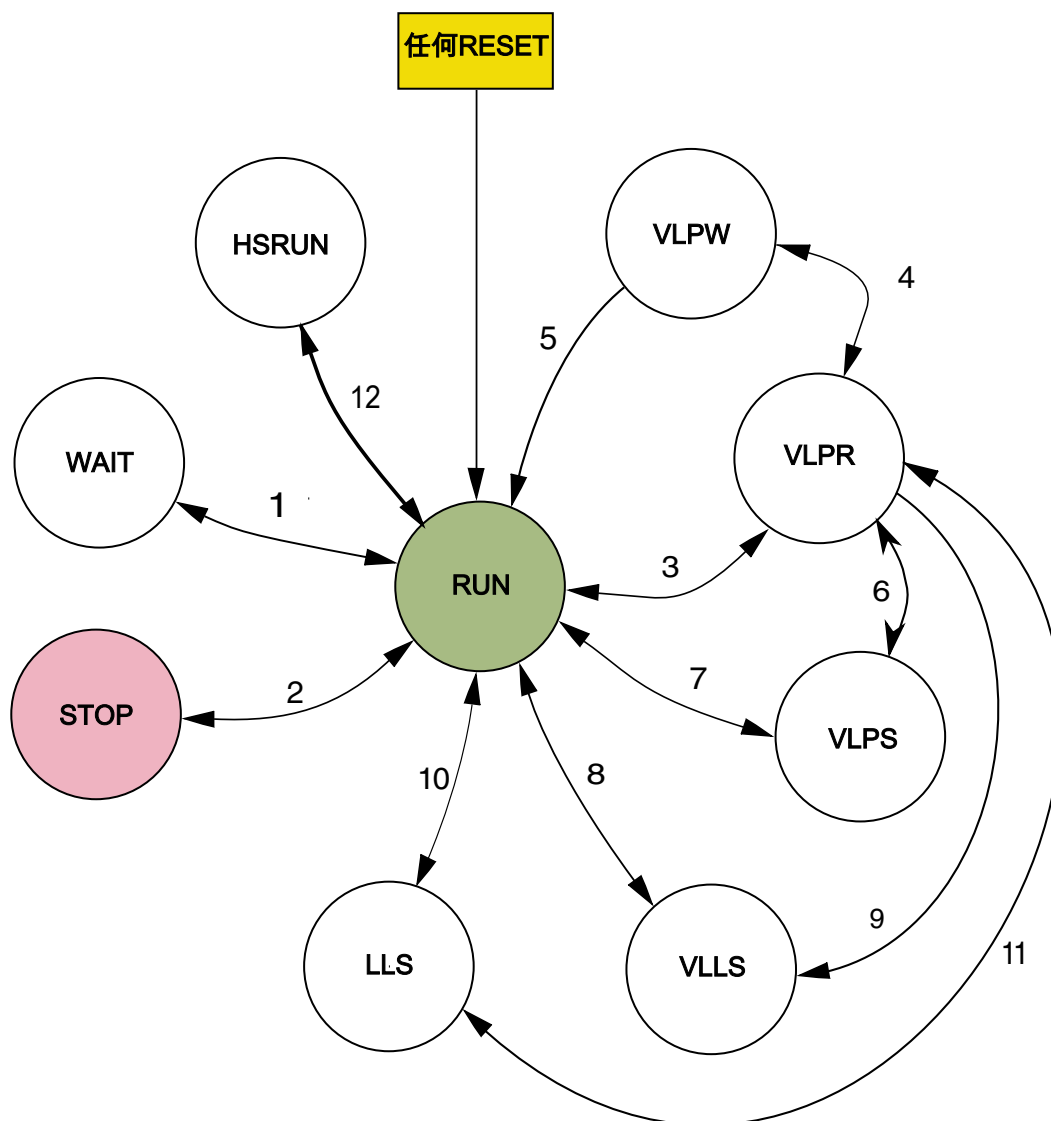


图 8-1. 功耗模式状态转换示意图

8.6 功耗模式关闭序列

进入停止或其他低功耗模式时，将按一定顺序关断时钟，以安全地将芯片置于目标低功耗状态。可以通过内核执行 WFI 指令来发起所有低功耗进入序列。ARM 内核的输出 SLEEPDEEP 和 SLEEPING 将触发进入不同的低功耗模式：

- 系统级等待和 VLPW 模式等于：SLEEPING 和 $\overline{\text{SLEEPDEEP}}$
- 所有其他低功耗模式等于：SLEEPING 和 SLEEPDEEP

进入非等待模式时，芯片执行以下序列：

- 立即关闭至 ARM Cortex-M4 内核的内核时钟和系统时钟。
- 轮询来自非内核交叉主机 (DMA)、支持外设 (SPI、PIT、RNG) 和 Flash 控制器的停止确认，以指示需要保持使能系统时钟、总线时钟和/或 Flash 时钟以完成先前发起的操作，从而有效停止进入目标低功耗模式。检测到所有确认之后，系统时钟、总线时钟和 Flash 时钟将被同时关闭。
- MCG 和模式控制器关闭按照目标低功耗模式定义由片上稳压器驱动的时钟源和/或内部电源。

在等待模式下，大部分系统时钟不受低功耗模式进入的影响。至 ARM Cortex-M4 内核的内核时钟被关闭。某些模块支持停止等待功能并且在这些配置下禁用其时钟。

当调试器使能时，调试器模块支持从 STOP、WAIT、VLPS 和 VLPW 转换回中止状态。可以通过置位 MDM-AP 控制寄存器中的调试请求位来发起此转换。作为此转换的一部分，系统时钟将被重新建立，等效于正常的 RUN//VLPR 模式时钟配置。

8.7 Flash 程序限制

在高速运行或 VLPR 功耗模式下，不得编程或擦除此器件上的 Flash 存储器。

8.8 低功率模式中的模块操作

下表说明了当芯片处于每个低功耗模式中时每个模块的功能。标准特性如表所示，其中包括运算操作 (CPO) 和部分停止 2 (PSTOP2) 的例外情况。

(调试模块另议，请参见[低功耗模式中的调试](#)。) 数字评级 (例如 2 MHz 和 1 Mbit/s) 代表每个模式的最高频率或数据率。同时，这些术语还用作：

- FF = 全部功能。在 VLPR 和 VLPW 下，系统频率受限，但如果模块的功能未受限，则仍列为 FF。

- 异步操作 = 如果可供选择的时钟源保持使能，则可实现全部功能。
- 静态 = 保留模块寄存器状态和相关存储器。
- 通电 = 存储器通电以保留内容。
- 低功耗 = 存储器通电在低功耗状态下保留内容。
- OFF = 模块断电；唤醒后模块处于复位状态。对于时钟，OFF 意味着禁用。
- 唤醒 = 模块可用作芯片的唤醒源。

表 8-2. 低功耗模式下的模块操作

模块	VLPR	VLPW	停止	VLPS	LLSx	VLLSx
内核模块						
NVIC	FF	FF	静态	静态	静态	关断
系统模块						
模式控制器	FF	FF	FF	FF	FF	FF
LLWU ¹	静态	静态	静态	静态	FF	FF ²
调节器	低功耗	低功耗	开	低功耗	低功耗	VLLS2/3 时为低功耗，VLLS0/1 时为关断
LVD	禁用	禁用	开	禁用	禁用	禁用
掉电检测	开	开	开	开	开	在 VLLS1/2/3 下为开，在 VLLS0 下可选择性禁用 ³
DMA	FF CPO 中为异步操作	FF	异步操作	异步操作	静态	关断
看门狗	FF	FF	FF	FF	静态	关断
EWM	FF CPO 中为静态	静态	静态 PSTOP2 中的 FF	静态	静态	关断
时钟						
1 kHz LPO	开	开	开	开	开	VLLS1/2/3 时为开，VLLS0 时为关断
系统振荡器 (OSC)	OSCERCLK (最大为 16 MHz 晶体)	OSCERCLK (最大为 16 MHz 晶体)	OSCERCLK 可选	OSCERCLK (最大为 16 MHz 晶体)	限定为低范围/低功耗	VLLS1/2/3 时限定为低范围/低功耗，VLLS0 时为关
MCG	4 MHz IRC	4 MHz IRC	静态 - MCGIRCLK 可选；PLL 选择性开启，但实施门控	静态 - MCGIRCLK 可选 (仅限 4 MHz IRC)	静态 - 无时钟输出	关
内核时钟	4 MHz (最大值)	关	关断	关断	关断	关
平台时钟	4 MHz (最大值)	4 MHz (最大值)	关	关断	关断	关断
系统时钟	4 MHz (最大值) CPO 为关	4 MHz (最大值)	关	关断	关断	关断
总线时钟	4 MHz (最大值)	4 MHz (最大值)	关	关	关断	关断

下一页继续介绍此表...

表 8-2. 低功耗模式下的模块操作 (继续)

模块	VLPR	VLPW	停止	VLPS	LLSx	VLLSx
	CPO 为关		PSTOP2 最大值为 MHz (从 RUN 模式进入) PSTOP2 最大值为 4 MHz (从 VLPR 模式进入)			
存储器 and 存储器接口						
Flash	1 MHz (最大值) 访问 - 无编程/擦除 CPO 中无寄存器访问	低功耗	低功耗	低功耗	关断	关
System RAM (SRAM_U and SRAM_L)	低功耗	低功耗	低功耗	低功耗	在 LLS3 下为低功耗, 在 LLS2 下为局部	在 LLS3 下为低功耗, 在 LLS2 下为局部 其它为关
VBAT 寄存器文件	通电	通电	通电	通电	通电	通电
系统寄存器文件	通电	通电	通电	通电	通电	通电
通信接口						
全速/低速 USB	静态, 在恢复时唤醒	静态, 在恢复时唤醒	静态, 在恢复时唤醒	静态, 在恢复时唤醒	静态	关断
UART0、UART1	250 kbit/s 静态, 在 CPO 边沿唤醒	250 kbit/s	静态, 在边沿唤醒	静态, 在边沿唤醒	静态	关
UART2	250kbit/s 静态, 在 CPO 边沿唤醒	250 kbit/s	静态, 在边沿唤醒 PSTOP2 中为 FF	静态, 在边沿唤醒	静态	关
LPUART0	4 Mbps CPO 中为异步操作	4 Mbps	异步操作 PSTOP2 中的 FF	异步操作	静态	关断
SPI	1 Mbit/s (从机) 2 Mbit/s (主机) CPO 中为静态	1 Mbit/s (从机) 2 Mbit/s (主机)	静态 PSTOP2 中的 FF	静态	静态	关
LPI2C	400 kbit/s FF, CPO 中为异步操作	400 kbit/s FF	异步操作	异步操作	静态	关断
CAN	500 kbit/s 在 CPO 中唤醒	500 kbit/s	唤醒 PSTOP2 中为 FF	唤醒	静态	关断
I ² S	FF CPO 中为异步操作	FF	采用外部时钟的异步操作 PSTOP2 中为 FF	带外部时钟的 FF ⁵	静态	关断
FlexIO	FF	FF	异步操作	异步操作	静态	关断

下一页继续介绍此表...

表 8-2. 低功耗模式下的模块操作 (继续)

模块	VLPR	VLPW	停止	VLPS	LLSx	VLLSx
			PSTOP2 中的 FF			
安全						
CRC	FF CPO 中为静态	FF	静态	静态	静态	关断
RNG	FF CPO 中为静态	FF CPO 中为静态	静态	静态	静态	关断
定时器						
TPM	FF	FF	异步操作 PSTOP2 中的 FF	异步操作	静态	关断
PIT	FF CPO 中为静态	FF	静态 PSTOP2 中的 FF	静态	静态	关断
PDB	FF CPO 中为静态	FF	静态 PSTOP2 中的 FF	静态	静态	关断
LPTMR	FF	FF	异步操作 PSTOP2 中的 FF	异步操作	异步操作	异步操作 ⁶
RTC - 32kHz OSC ⁴	FF CPO 中为异步操作	FF	异步操作 PSTOP2 中的 FF	异步操作	异步操作	异步操作 ⁷
模拟						
16 位 ADC	FF 仅在 CPO 中为 ADACK 和 ALTCLK 时钟	FF	仅为 ADACK、 ALTCLK、和 ALTCLK2 时钟 PSTOP2 中为 FF	仅为 ADACK 和 ALTCLK 时钟	静态	关
CMP ⁸	FF CPO 中为高速或 低速比较	FF	HS 或 LS 比较 PSTOP2 中的 FF	HS 或 LS 比较	低速比较	VLLS1/2/3 时为 低速比较, VLLS0 时为关断
6 位 DAC	FF CPO 中为静态	FF	静态 PSTOP2 中的 FF	静态	静态	静态, VLLS0 时 为关断
12 位 DAC	FF CPO 中为静态	FF	静态 PSTOP2 中的 FF	静态	静态	静态
人机接口						
GPIO	FF 仅在 CPO 中写 入 GPIO	FF	静态输出, 唤醒输 入 PSTOP2 中为 FF	静态输出, 唤醒输 入	静态, 引脚锁存	关, 引脚锁存

1. 使用 LLWU 模块，则用于此芯片的外部引脚不需要使能相关的外设功能。它只需将控制该引脚 (GPIO 或外设) 的功能配置为输入，以便允许 LLWU 上发生一次转换。
2. 由于 LPO 时钟源禁用，VLLS0 期间将旁路滤波器。
3. SMC 模块中的 SMC_STOPCTRL[PORPO] 位控制此选项。
4. 这些元件在 BAT 功耗模式下仍然保持通电状态。
5. 使用外部生成的位时钟或外部生成的音频主机时钟 (包括 EXTAL)。
6. VLLS0 时不提供系统 OSC 和 LPO 时钟源。所有模式均支持脉冲计数。
7. RTC_CLKOUT 不可用。可配置 CLKOUT32K 为可供选择的 32 Khz 时钟。
8. 停止模式或 VLPS 时的 CMP 支持高速或低速外部引脚间或外部引脚与 DAC 之间的比较。。LLSx 或 VLLSx 时的 CMP 仅支持低速外部引脚间或外部引脚与 DAC 之间的比较。停止、VLPS、LLSx、或 VLLSx 模式下不提供窗口、采样和滤波工作模式。

第 9 章 加密

9.1 简介

KS22 可以根据 Flash 模块的加密等级切换不同的安全模式。本章将讲述 Flash 加密以及 Flash 加密对芯片的其他模块的影响。

9.2 Flash 加密

Flash 模块根据 FSEC[SEC] 位保存的状态向 MCU 提供安全信息。而 MCU 根据 FSEC[SEC] 位的状态确认加密请求, 并对 Flash 的资源访问做限制。复位期间, Flash 模块使用从 Flash 配置字段读取的数据对 FSEC 寄存器进行初始化。

注

这种安全机制仅适用于通过外部调试器对芯片访问的情况。
CPU 对 Flash 的访问不受 FSEC 状态的影响。

在非安全状态下, 所有 Flash 命令都可供 (JTAG) 使用, 用户代码也可执行 Flash 控制器命令。当 Flash 处于加密状态时 (FSEC[SEC] 等于 00、01 或 11), 编程接口只允许发起全片擦除操作, 无权进行其他操作。

关于 Flash 加密选项和使能/禁用 Flash 加密的更多信息, 请参见 [Flash 存储器模块](#)。

9.3 与其它模块之间的安全性交互

系统通过 Flash 加密设置确定可用资源。下面的章节将介绍 Flash 加密位对片上其他模块的影响。

9.3.1 与调试器之间的安全性

当 Flash 安全特性激活时，JTAG 端口无法访问 MCU 的存储器资源。边界扫描链操作可以运行，但调试功能禁用，因此，调试端口无法读取 Flash 内容。

虽然大部分调试功能被禁用，但是调试器可以写 MDM-AP 控制寄存器的“Erase All Block”位以触发全片擦除（擦除所有数据块）命令。即使某些存储器位置受保护，也能通过调试器进行整体擦除。

禁用 Erase All Block 后，调试器也不能通过 Erase All Block 指令来擦除 Flash。

第 10 章 调试

10.1 简介

该器件的调试部分基于 ARM coresight 架构，在每个不同的器件中，可能由于引脚和其他资源优化分配的原因，coresight 功能可能会做部分裁剪。

本器件支持 4 种调试接口：

- IEEE 1149.1 JTAG
- IEEE 1149.7 JTAG (cJTAG)
- 串行线调试 (SWD)
- ARM 实时跟踪接口

Cortex-M4 调试架构非常灵活。下图显示了内核调试架构及其组件的拓扑。

图 10-1. Cortex-M4 调试拓扑

下表展示了每个调试组件的简短描述。

表 10-1. 调试组件描述

模块	说明
SWJ-DP	经过优化的调试端口，支持 SWD、JTAG
AHB-AP	连接 JTAG 至调试模块的 AHB 主机接口以及 SOC 系统存储器映射
MDM-AP	为外部调试器提供集中控制和状态寄存器，以控制器件。
ROM 表	标识可用调试 IP。
内核调试	单步、寄存器访问、运行、内核状态
ITM	软件指令信息 + 单数据跟踪消息 + 观察点消息
DWT (数据和地址观察点)	4 个数据和地址观察点
FPB (Flash 地址重载及断点)	FPB 实现硬件断点，并将代码空间中的代码与数据打包到系统空间 FPB 单元包含指令字面值比较器，用于匹配来自代码空间的数据，FPB 也可以将指令地址重新映射到其他系统空间内。

下一页继续介绍此表...

表 10-1. 调试组件描述 (继续)

模块	说明
	FPB 还包含六个指令比较器，用于匹配来自代码空间的指令提取以及重新映射至系统空间内的对应区域。或者，六个指令比较器可以单独地通过配置比较器，将断点指令（BKPT）返回至匹配上的处理器内核，从而提供硬件断点功能。
TPIU（跟踪端口接口单元）	异步模式（1 引脚）= TRACE_SWO（可用于 JTAG_TDO 上）

10.1.1 参考资料

有关 ARM 调试组件的更多信息，请参见以下文档：

- ARMv7-M Architecture Reference Manual
- ARM Debug Interface v5.1
- ARM CoreSight Architecture Specification

10.2 调试端口

下图介绍了 cJTAG 模块、JTAG 控制器和调试端口的配置：

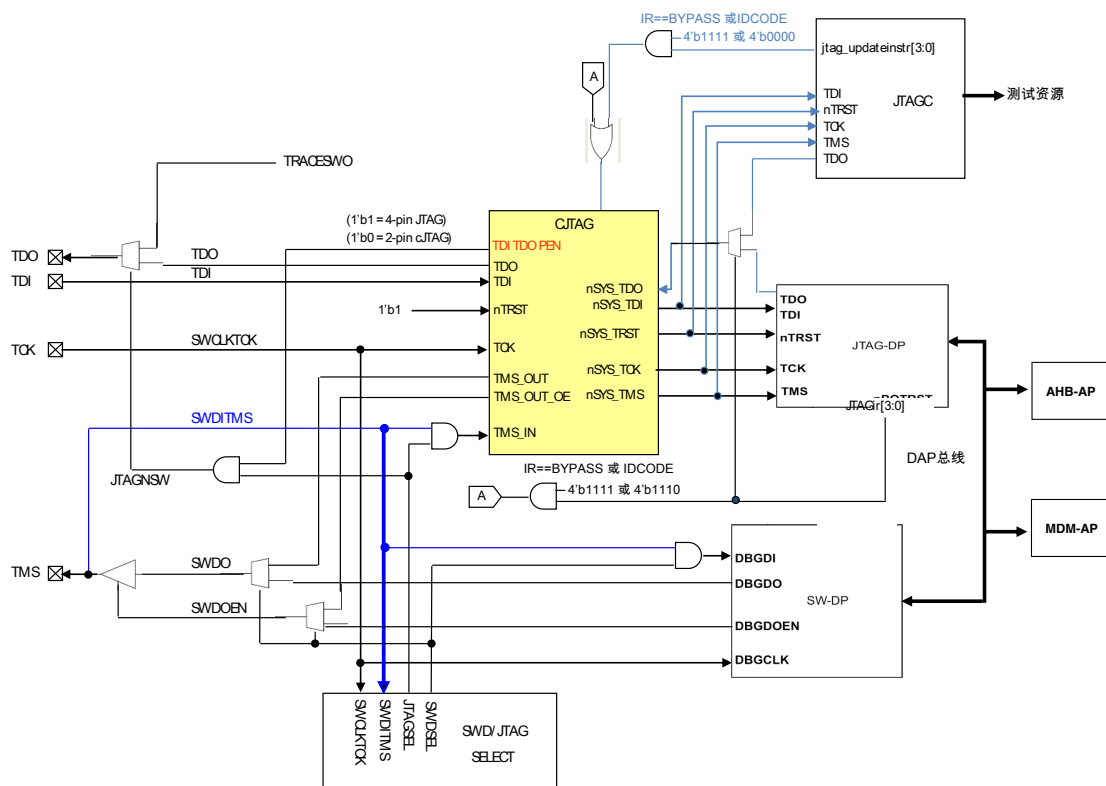


图 10-2. 已修改的调试端口

调试端口由标准 JTAG 模式下的复位产生，并且可按下面所给出的序列切换至 cJTAG 或 SWD 模式。在更改模式后，可以将未使用的调试引脚设置为其他复用功能。

10.2.1 JTAG 至 SWD 更改顺序

1. 发送超过 50 个 TCK 周期，且 TMS (SWDIO) = 1。
2. 发送 16 位序列，且 TMS (SWDIO) = 0111_1001_1110_0111 (MSB 先传输)。
3. 发送超过 50 个 TCK 周期，且 TMS (SWDIO) = 1。

注

请参见 ARM 文档，以了解 CoreSight DAP Lite 的相关限制。

10.2.2 JTAG 至 cJTAG 更改顺序

1. 复位调试端口。
2. 通过零位扫描将控制级别设置为 2。

3. 执行存储格式 (STFMT) 命令 (00011) 以将扫描格式寄存器设置为 1149.7 扫描格式。

10.3 调试端口引脚说明

调试端口的引脚在 POR (上电) 后, 默认设置为调试复用功能 (除 JTAG_TRST_b 外), 这些引脚可以后重新分配至其可选功能。在 cJTAG 和 SWD 模式下, 可配置 JTAG_TDI 和 JTAG_TRST_b 为 GPIO 功能。

表 10-2. 调试端口引脚

引脚名称	JTAG 调试端口		cJTAG 调试端口		SWD 调试端口		内部上拉/下拉
	类型	说明	类型	说明	类型	说明	
JTAG_TMS/ SWD_DIO	I	JTAG 测试模式选择	I	cJTAG 数据	I/O	串行线数据	上拉
JTAG_TCLK/ SWD_CLK	I	JTAG 测试时钟	I	cJTAG 时钟	I	串行线时钟	下拉
JTAG_TDI	I	JTAG 测试数据输入	-	-	-	-	上拉
JTAG_TDO/ TRACE_SWO	O	JTAG 测试数据输出	O	通过单个引脚跟踪输出	O	通过单个引脚跟踪输出	N/C
JTAG_TRST_b	I	JTAG 复位	I	cJTAG 复位	-	-	上拉

10.4 系统 TAP 连接

系统 JTAG 控制器以并联方式与 ARM TAP 控制器相连。系统 JTAG 控制器 IR 代码可覆盖 ARM JTAG 控制器 IR 代码, 而不会发生任何冲突。请参见 IR 代码表以获取可用 IR 代码列表。TAP 输出 (TDO) 根据所选的 IR 代码进行复用。该设计完全符合 JTAG 标准, 并且将 JTAG 链作为单个 TAP 显示。上电复位时, 默认选择 ARM 的 IDCODE (IR=4'b1110)。

10.4.1 IR 代码

表 10-3. JTAG 指令

指令	代码[3:0]	指令说明
IDCODE	0000	选择器件标识寄存器。

下一页继续介绍此表...

表 10-3. JTAG 指令 (继续)

指令	代码[3:0]	指令说明
SAMPLE/PRELOAD	0010	选择边界扫描寄存器，以便移位、采样和预载，而不影响功能运行。
SAMPLE	0011	选择边界扫描寄存器，以便移位和采样，而不影响功能运行。
EXTEST	0100	选择边界扫描寄存器，将预设值输出到对应引脚，同时复位系统。
HIGHZ	1001	选择旁路寄存器，将引脚设置为高阻态，复位系统。
CLAMP	1100	选择旁路寄存器，将预设值输出到调试引脚上，同时复位系统。
ARM_IDCODE	1110	ARM JTAG-DP 指令。
BYPASS	1111	为数据操作选择旁通寄存器。
Factory debug reserved	0101, 0110, 0111, 1101	仅供出厂调试使用。
ARM JTAG-DP Reserved	1000, 1010, 1011, 1110	这些指令控制 ARM JTAG-DP 控制器。有关这些指令的更多信息，请查看 ARM JTAG-DP 文档。
Reserved ¹	所有其他操作码	解码以选择旁通寄存器。

1. 制造商保留未来更改对预留指令代码解码的权利。

10.5 JTAG 状态和控制寄存器

通过 ARM 调试访问端口 (DAP) 调试器可以访问 (如下图所示) 以下寄存器。这组寄存器提供了运行时的状态和控制信息，以及低功耗恢复的状态信息。凭借状态寄存器位，调试器无需通过 Cross Bar 发起总线操作即能获得内核的已更新状态。

值得注意的是，这些 DAP 控制和状态寄存器并非在系统存储器映射中进行地址映射，并且只能通过使用 JTAG、cJTAG 或 SWD 的调试访问端口 (DAP) 进行访问。调试访问端口 1 的 MDM-AP 的可访问寄存器如下表所示。

表 10-4. MDM-AP 寄存器汇总

地址	寄存器	说明
0x0100_0000	状态	参见 MDM-AP 状态寄存器
0x0100_0004	控制	参见 MDM-AP 控制寄存器
0x0100_00FC	ID	始终读取为 0x001C_0000 的只读标识寄存器

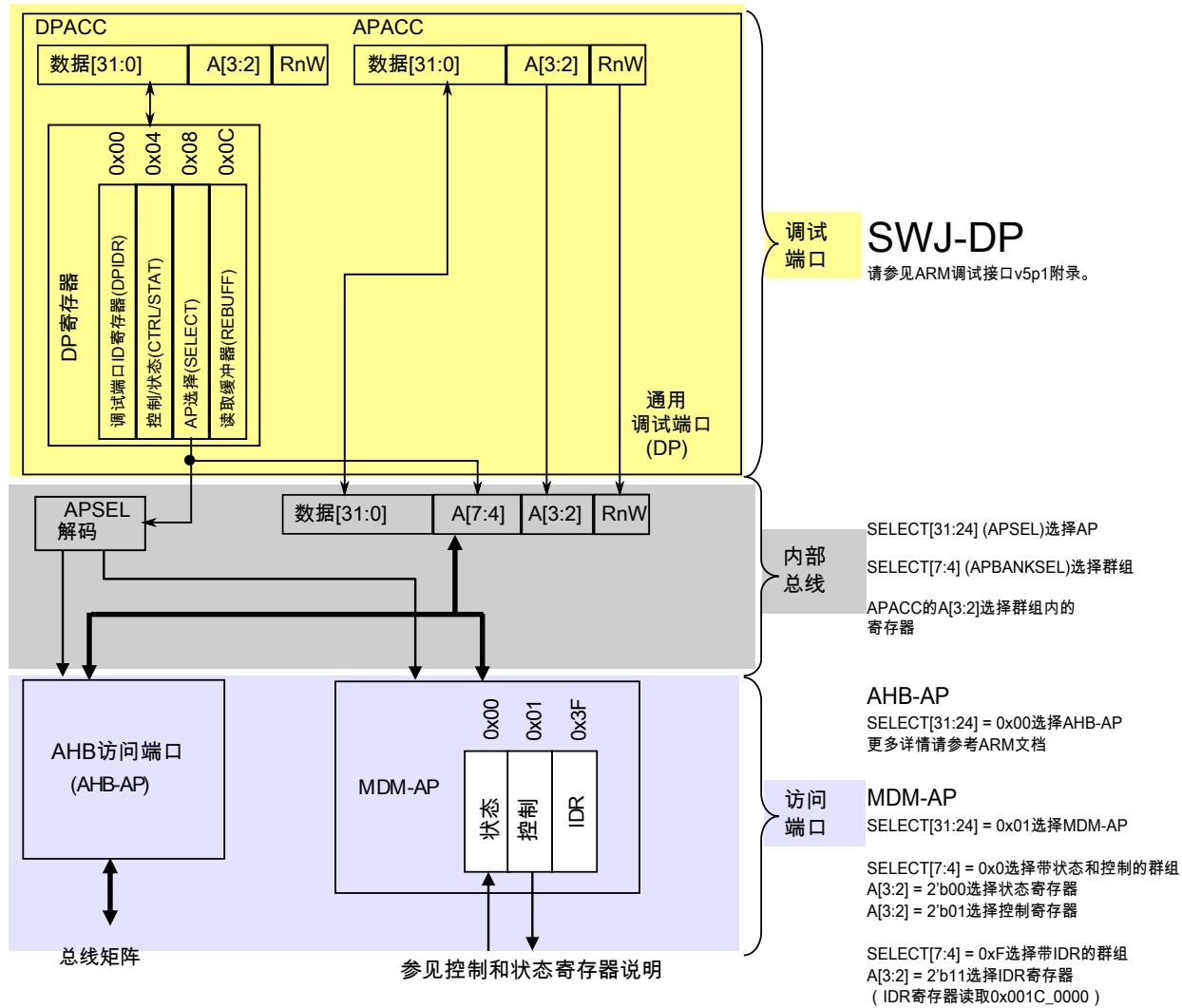


图 10-3. MDM AP 寻址

10.5.1 MDM-AP 控制寄存器

表 10-5. MDM-AP 控制寄存器分配

位	名称	安全 ¹	说明
0	正在执行 Flash 整体擦除	是	置位后开始执行 Mess Erase 操作。Mess Erase 完成后，由硬件自动清除此位。 当 Mess Erase (通过 MEEN 设置) 被禁用后，将不会执行 Mess Erase 擦除请求，并且次位将持续置位，直到下一个系统复位。
1	调试禁用	否	置位禁用调试。清零允许调试操作。设置后，它将覆盖 DHCSR 内的 C_DEBUGEN 位并将强制禁用调试逻辑。
2	调试请求	否	置位以强制暂停内核。

下一页继续介绍此表...

表 10-5. MDM-AP 控制寄存器分配 (继续)

位	名称	安全 ¹	说明
			若内核处于低功耗 (WAIT 或者 STOP) 模式, 该位可用于唤醒内核并转变到 HALT 状态。
3	系统复位请求	否	置位强制进行系统复位。系统将保持复位, 直至此位被清除。
4	内核保持复位	否	配置位, 可控制系统复位序列结束时的内核操作。 0 正常运行 - 在系统复位序列结束时, 解除内核与系统的其他部分的复位。 1 强制内核处于复位状态 - 在复位序列结束后将内核保持在复位状态。一旦系统进入挂起状态, 该控制位清零可立即解除内核的复位状态并且 CPU 开始工作。
5	VLLSx 调试请求 (VLLDBGREQ)	否	在下次从 VLLSx 模式恢复之后, 设置以配置在下次复位时使系统处于 hold 状态。 设置此位, 当系统从 VLLSx 恢复时, 系统处于 Hold 状态, 以便让调试器重新初始化 Debug 模块。 模式控制器将在进入 VLLSx 模式时捕捉此位逻辑。退出 VLLSx 模式时, 模式控制器将保持在复位状态, 直到 VLLDBGACK 被置位。 当系统进行上电复位 (POR) 时, 此位会被清 0。
6	VLLSx 调试应答 (VLLDBGACK)	否	释放一个由退出 VLLSx 模式所造成的系统 Hold 状态。 调试器可使用此位来退出系统复位状态 (在退出 VLLSx 模式时保持)。调试器重新初始化所有调试 IP, 然后再置位此位, 以准许 CPU 退出复位状态并开始运行。 VLLDBGACK 位通过调试器清零, 或者可保持原样, 在下次上电复位 (POR) 后会被自动清 0。
7	LLS、VLLSx 状态应答	否	置位此位以确认已读取 DAP LLS 和 VLLS 状态位。此确认会将状态位自动清零。 调试器可使用此位来清除 LLS 和 VLLSx 模式的进入状态位。此位可通过调试器置位和清零。

1. 安全模式下可用的命令。

10.5.2 MDM-AP 状态寄存器

表 10-6. MDM-AP 状态寄存器分配

位	名称	说明
0	Flash Mess Erase 完成	Flash 批量擦除应答位在任何系统复位后都会清零。当 MDM-AP 控制寄存器中的 Mess Erase In Progress 位被置位时, 此位也会被清零。当 Flash 控制器正在执行 Mess Erase 操作时, 这位被置 1。 当批量擦除 (通过 MEEN 设置) 被禁用后, 就不会应答置位“Mess Erase In Progress”位所引起的擦除请求。
1	Flash 就绪	表示 Flash 已完成初始化, 调试器可配置, 即使调试器继续使系统保持复位状态, 此位也仍然有效。
2	系统加密	表示加密状态。处于安全状态时, 调试器无法访问系统总线或任何存储器映射外设。该位表示器件锁定且无法进行系统总线访问。
3	系统复位	表示系统复位状态。

下一页继续介绍此表...

表 10-6. MDM-AP 状态寄存器分配 (继续)

位	名称	说明
		0 系统处于复位状态。 1 系统未处于复位状态。
4	保留	
5	批量擦除使能	表示 MCU 是否可以批量擦除。 0 禁用批量擦除 1 使能批量擦除
6	Backdoor Access Key Enable	表示 MCU 是否已使能后门密钥访问 0 禁用 1 使能
7	LP 使能	解码 LPLLSM 控制位，表示下次 ARM 内核进入深度睡眠模式时，VLPS、LLS、或 VLLSx 是选定的功耗模式。 0 未使能低功耗停止模式 1 已使能低功耗停止模式 旨在用于尝试从 RUN 到 VLPS 的调试操作。根据调试定义，系统实际上会进入停止状态。当此位置位时，调试器应当将深度睡眠指示（SLEEPDEEP 和 SLEEPING 置位时）解释为调试器-VLPS 状态指示。
8	超低功耗模式	表示当前功耗模式为 VLPx。此位始终表示 VLPx 是否使能。 此位用于上下调整 JTAG TCK 频率。
9	LLS 模式退出	此位表示已发生退出 LLS 模式事件。调试器将丧失通信，而系统处于 LLS 模式（包括访问此寄存器）。通信重新建立后，此位表示系统已处于 LLS 模式。调试模块在 LLS 期间会保持状态不变，因此无需重新配置。 在退出 LLS 模式的过程中，此位将一直置位。“LLS 模式退出”位保持到调试器有机会认识到已退出 LLS 为止，写入 1 到 MDM AP 控制寄存器的 LLS、VLLSx 状态应答位可将其清零。
10	VLLSx 模式退出	此位表示已发生退出 VLLSx 模式事件。调试器将丧失通信，而系统处于 VLLSx 模式（包括访问此寄存器）。通信重新建立后，此位表示系统已处于 VLLSx 模式。调试模块在 VLLSx 模式期间会丧失状态，因此需要重新配置。 在退出 VLLS 模式的过程中，此位将一直置位。“VLLSx 模式退出”位保持到调试器有机会认识到已退出 VLLS 模式为止，写入 1 到 MDM AP 控制寄存器的 LLS、VLLSx 状态应答位可将其清零。
11 – 15	保留以供将来使用	始终读取 0。
16	内核已停止	表示内核已进入调试暂停模式。
17	内核 SLEEPDEEP	表示内核已进入低功耗模式
18	内核 SLEEPING	SLEEPING==1 且 SLEEPDEEP==0 表示 WAIT 或 VLPW 模式。 SLEEPING==1 且 SLEEPDEEP==1 表示 STOP 或 VLPS 模式。
19 – 31	保留以供将来使用	始终读取 0。

10.6 调试复位

调试系统将收到以下复位源：

- 来自外部信号的 JTAG_TRST_b。有些封装上可能没有此信号的引脚。
- 在 TCLK 域中调试复位 (SWJ-DP CTRL/STAT 寄存器内的 CDBGSTREQ 位), 允许调试器复位调试逻辑。
- 通过 cJTAG 转义命令置位 TRST。
- 系统 POR 复位

相反, 调试系统能够使用以下机制产生系统复位:

- DAP 控制寄存器中的系统复位, 允许调试器保持系统在复位状态。
- NVIC 应用中断和复位控制寄存器中的 SYSRESETREQ 位。
- DAP 控制寄存器中的系统复位, 允许调试器保持内核在复位状态。

10.7 AHB-AP

AHB-AP 允许调试器访问系统中的所有存储器和寄存器, 包括通过 NVIC 访问处理器寄存器。系统访问与处理器状态无关。AHB-AP 不在总线上执行背靠背事务, 因此所有事务都是非序列化的。AHB-AP 可执行未对齐的事务和位带事务。事务会绕过 FPB, 因此 FPB 无法重新映射 AHB-AP 事务。SWJ/SW-DP 启动的事务中止会驱动一个 AHB-AP 支持的边带信号, 称为 HAbort。此信号被驱动到总线矩阵中, 它会复位总线矩阵状态, 以便 AHB-AP 能访问私有外设总线进行最后的调试努力, 如读取/停止/复位内核等。所有 AHB-AP 传输事物都是小端对齐的。

在系统复位事件开始时的短暂时间内系统处于安全状态, 此时禁止调试器访问所有 AHB-AP 事务。但是可以访问和监视 MDM-AP 状态寄存器, 以便确定初始周期何时完成。经过此初始周期后, 若通过置位 RESET 引脚让系统保持复位状态, 则调试器可通过总线矩阵访问私有外设总线以配置调试 IP, 即使系统复位置位也是如此。在系统复位期间, 禁止通过交叉开关访问其他存储器和寄存器资源。

10.8 ITM

ITM 是一种应用程序驱动的跟踪源, 支持类似 printf 的输出跟踪操作系统 (OS) 和应用程序事件, 并发出诊断系统信息。ITM 以数据包形式发出跟踪信息。有四个源可产生数据包。若多个源同时产生数据包, ITM 会仲裁数据包的输出顺序。这四个源按优先级从高到低排列是:

1. 软件跟踪 -- 软件可直接写入 ITM 刺激寄存器。这会发出数据包。
2. 硬件跟踪 -- DWT 产生这些数据包, ITM 将其发出。
3. 时间戳 -- 时间戳随数据包一起发出 ITM 包含一个 21 位计数器来产生时间戳。Cortex-M4 时钟或串行线查看器 (SWV) 输出的位时钟速率为该计数器提供时钟。
4. 全局系统时间戳。可选择利用全系统的 48 位计数值产生时间戳。

10.9 内核跟踪连接性

ITM 可将其数据路由至 TPIU。(请参见 [MCM \(其他控制模块\)](#) 以控制至 TPIU 的路由。) 此配置可以通过低成本工具实现使用跟踪, 并且同时保持与跟踪仿真器的兼容性。

10.10 TPIU

TPIU 将 ITM 产生的片上追踪数据转换为数据流, 其后可通过跟踪端口分析器 (TPA) 捕捉。TPIU 专门设计用于低成本调试。

10.11 DWT

DWT 是一个执行以下调试功能的单元:

- 它包含 4 个比较器, 可配置为硬件观察点、PC 采样器事件触发器或数据地址采样器事件触发器。第一个比较器 DWT_COMP0 还可比较时钟周期计数器 CYCCNT。第二个比较器 DWT_COMP1 还可用作数据比较器。
- DWT 包含用于以下事项的计数器:
 - 时钟周期 (CYCCNT)
 - 折叠命令
 - 负载储存单元 (LSU) 操作
 - 睡眠周期
 - CPI (除第一个周期外的所有指令周期)
 - 中断开支

注

每次计数器溢出时将发出一个事件。

- DWT 可配置为按定义的间隔发出 PC 样本, 以及发出中断事件信息。

10.12 低功耗模式中的调试

在调试模块处于静态或断电状态的低功耗模式中, 调试器在低功耗模式期间不能采集任何调试数据。当调试器处于静态状态, 只要退出低功耗模式且系统返回至具有有效调试的状态, 调试端口即返回全部功能。在调试器逻辑断电的情况下, 调试器在恢复时复位, 且必须在退出低功耗模式后立即重新配置。

系统模式控制模块输入逻辑监控调试端口中的“调试启动”和“系统启动”信号来作为调试器处于有效状态的指示。这些信号在 RUN、VLPR、WAIT 和 VLPW 模式中可以进行更改。如果调试信号处于有效状态且系统尝试进入 STOP 或 VLPS 模式，则 FCLK 将继续运行以支持内核寄存器访问。在 FCLK 处于有效状态的这些模式中，调试模块有权访问内核寄存器，但无权通过 Cross bar 访问系统存储器源。

使能调试后，不允许直接从 RUN 转换至 VLPS 模式，这会导致系统进入 STOP 模式。可以通过评估 MDM-AP 状态寄存器中的状态位来确定此伪 VLPS 状态。请注意，使能调试后，可以从 RUN 转换至 VLPR 再转换至 VLPS 模式，但这会导致系统进入 STOP 模式。

在 VLLS 模式下，所有调试模块均关断并在唤醒时复位。在 LLS 模式下，调试模块保留其状态，但不可进行任何调试活动。

注

使用 cJTAG 并进入 LLS 模式时，必须在退出 LLS 模式时复位 cJTAG 控制器。

进入 VLLSx 模式将导致所有调试控制和设置复位。为了留出时间使调试器与硬件同步，可对 MDM-AP 控制寄存器进行配置使系统在恢复时保持复位状态，这样调试器就能够在系统退出复位并恢复操作之前重新获得对调试逻辑的控制和重新配置权。

10.12.1 低功耗模式中的调试模块状态

下表显示低功耗模式中调试模块的状态。这些术语还用作：

- FF = 全部功能。在 VLPR 和 VLPW 下，系统频率受限，但如果模块的功能未受限，则仍列为 FF。
- 静态 = 保留模块寄存器状态和相关存储器。
- OFF = 模块断电；唤醒后模块处于复位状态。

表 10-7. 低功耗模式中的调试模块状态

模块	STOP	VLPR	VLPW	VLPS	LLS	VLLSx
调试端口	FF	FF	FF	关断	静态	关断
AHB-AP	FF	FF	FF	关断	静态	关断
ITM	FF	FF	FF	关断	静态	关断
TPIU	FF	FF	FF	关断	静态	关断
DWT	FF	FF	FF	关断	静态	关断

10.13 调试和加密

使能加密 (FSEC[SEC] != 10) 后, 调试端口的功能将受限, 以防数据泄密。在加密状态下, 调试器仍然可以访问 MDM-AP 状态寄存器, 并且可以确定该器件当前的加密状态。在加密器件的情况下, 如果已经使能整体擦除操作, 那么调试器也可以通过写入 MDM-AP 控制寄存器来执行 Mess Erase 指令来整体擦除操作。如果禁用 Mess Erase (FSEC[MEEN] = 10) 并且器件又处于加密状态, 则无法通过调试接口进行擦除 Flash 操作。

整体擦除被禁用 (FSEC[MEEN]= 10) 时, 调试器不能通过写入 MDM-AP 控制寄存器来实施整体擦除操作。

第 11 章

信号复用及信号说明

11.1 简介

为了优化小型封装的功能，引脚通过信号复用技术实现多种功能。本章阐述了芯片信号与外部引脚间的复用关系。

端口控制 模块控制着外部引脚上这些信号的当前功能。参考关于寄存器控制引脚功能的章节。

11.2 引脚分配

11.2.1 信号复用和引脚分配

下表显示的是各引脚上的信号以及这些引脚在本文档所支持芯片上的位置。“端口控制模块”负责选择每个引脚上可用的 ALT 功能。

注

KS20 中，只存在 CAN0。KS22 中，有两种 CAN 模块实例 (CAN0 和 CAN1)。

100 LQFP	64 LQFP	48 QFN	Pin Name	默认值	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
1	1	1	PTE0/ CLKOUT32K	ADC0_SE4a	ADC0_SE4a	PTE0/ CLKOUT32K	SPI1_PCS1	UART1_TX			LPI2C1_SDA	RTC_ CLKOUT
2	2	2	PTE1/ LLWU_P0	ADC0_SE5a	ADC0_SE5a	PTE1/ LLWU_P0	SPI1_SOUT	UART1_RX			LPI2C1_SCL	SPI1_SIN
3	—	3	PTE2/ LLWU_P1	ADC0_SE6a	ADC0_SE6a	PTE2/ LLWU_P1	SPI1_SCK	UART1_CTS_ b				
4	—	4	PTE3	ADC0_SE7a	ADC0_SE7a	PTE3	SPI1_SIN	UART1_RTS_ b				SPI1_SOUT
5	—	5	PTE4/ LLWU_P2	禁用		PTE4/ LLWU_P2	SPI1_PCS0	LPUART0_TX				LPI2C1_SDA
6	—	6	PTE5	禁用		PTE5	SPI1_PCS2	LPUART0_RX				LPI2C1_SCL

引脚分配

100 LQFP	64 LQFP	48 QFN	Pin Name	默认值	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
7	—	—	PTE6	禁用		PTE6	SPI1_PCS3	LPUART0_CTS_b	I2S0_MCLK			USB_SOF_OUT
8	3	7	VDD	VDD	VDD							
9	4	8	VSS	VSS	VSS							
10	5	9	USB0_DP	USB0_DP	USB0_DP							
11	6	10	USB0_DM	USB0_DM	USB0_DM							
12	7	11	USBVDD	USBVDD	USBVDD							
13	—	—	NC	NC	NC							
14	8	—	ADC0_DP1	ADC0_DP1	ADC0_DP1							
15	—	—	ADC0_DM1	ADC0_DM1	ADC0_DM1							
16	—	—	ADC0_DP2	ADC0_DP2	ADC0_DP2							
17	—	—	ADC0_DM2	ADC0_DM2	ADC0_DM2							
18	9	—	ADC0_DP0	ADC0_DP0	ADC0_DP0							
19	10	—	ADC0_DM0	ADC0_DM0	ADC0_DM0							
20	11	—	ADC0_DP3	ADC0_DP3	ADC0_DP3							
21	12	—	ADC0_DM3	ADC0_DM3	ADC0_DM3							
22	13	12	VDDA	VDDA	VDDA							
23	14	12	VREFH	VREFH	VREFH							
24	15	13	VREFL	VREFL	VREFL							
25	16	13	VSSA	VSSA	VSSA							
26	17	—	CMP0_IN5	CMP0_IN5	CMP0_IN5							
27	18	—	DAC0_OUT/ ADC0_SE23	DAC0_OUT/ ADC0_SE23	DAC0_OUT/ ADC0_SE23							
28	19	14	XTAL32	XTAL32	XTAL32							
29	20	15	EXTAL32	EXTAL32	EXTAL32							
30	21	16	VBAT	VBAT	VBAT							
31	—	—	PTE24	ADC0_SE17	ADC0_SE17	PTE24	CAN1_TX	TPM0_CH0	I2S1_TX_FS	LPI2C0_SCL	EWM_OUT_b	
32	—	—	PTE25	ADC0_SE18	ADC0_SE18	PTE25	CAN1_RX	TPM0_CH1	I2S1_TX_BCLK	LPI2C0_SDA	EWM_IN	
33	—	—	PTE26/ CLKOUT32K	禁用		PTE26/ CLKOUT32K			I2S1_TXD0		RTC_CLKOUT	USB_CLKIN
34	22	17	PTA0	JTAG_TCLK / SWD_CLK		PTA0	UART0_CTS_b	TPM0_CH5		EWM_IN		JTAG_TCLK / SWD_CLK
35	23	18	PTA1	JTAG_TDI		PTA1	UART0_RX		CMP0_OUT	LPI2C1_HREQ	TPM1_CH1	JTAG_TDI
36	24	19	PTA2	JTAG_TDO / TRACE_SWO		PTA2	UART0_TX				TPM1_CH0	JTAG_TDO / TRACE_SWO
37	25	20	PTA3	JTAG_TMS / SWD_DIO		PTA3	UART0_RTS_b	TPM0_CH0		EWM_OUT_b		JTAG_TMS / SWD_DIO
38	26	21	PTA4/ LLWU_P3	NMI_b		PTA4/ LLWU_P3		TPM0_CH1			I2S0_MCLK	NMI_b
39	27	—	PTA5	禁用		PTA5	USB_CLKIN	TPM0_CH2			I2S0_TX_BCLK	JTAG_TRST_b
40	—	—	VDD	VDD	VDD							

100 LQFP	64 LQFP	48 QFN	Pin Name	默认值	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
41	—	—	VSS	VSS	VSS							
42	28	—	PTA12	禁用		PTA12	CAN0_TX	TPM1_CH0			I2S0_TXD0	
43	29	—	PTA13/ LLWU_P4	禁用		PTA13/ LLWU_P4	CAN0_RX	TPM1_CH1			I2S0_TX_FS	
44	—	—	PTA14	禁用		PTA14	SPI0_PCS0	UART0_TX			I2S0_RX_ BCLK	
45	—	—	PTA15	禁用		PTA15	SPI0_SCK	UART0_RX			I2S0_RXD0	
46	—	—	PTA16	禁用		PTA16	SPI0_SOUT	UART0_CTS_ b			I2S0_RX_FS	
47	—	—	PTA17	禁用		PTA17	SPI0_SIN	UART0_RTS_ b			I2S0_MCLK	
48	30	22	VDD	VDD	VDD							
49	31	23	VSS	VSS	VSS							
50	32	24	PTA18	EXTAL0	EXTAL0	PTA18			TPM_CLKIN0			
51	33	25	PTA19	XTAL0	XTAL0	PTA19			TPM_CLKIN1		LPTMR0_ ALT1	
52	34	26	RESET_b	RESET_b	RESET_b							
53	35	27	PTB0/ LLWU_P5	ADC0_SE8	ADC0_SE8	PTB0/ LLWU_P5	LPI2C0_SCL	TPM1_CH0			FXIO0_D4	UART0_RX
54	36	28	PTB1	ADC0_SE9	ADC0_SE9	PTB1	LPI2C0_SDA	TPM1_CH1		EWM_IN	FXIO0_D5	UART0_TX
55	37	29	PTB2	ADC0_SE12	ADC0_SE12	PTB2	LPI2C0_SCL	UART0_RTS_ b			FXIO0_D6	CAN1_RX
56	38	30	PTB3	ADC0_SE13	ADC0_SE13	PTB3	LPI2C0_SDA	UART0_CTS_ b			FXIO0_D7	CAN1_TX
57	—	—	PTB9	禁用		PTB9	SPI1_PCS1	LPUART0_ CTS_b				
58	—	—	PTB10	禁用		PTB10	SPI1_PCS0	LPUART0_RX	I2S1_TX_ BCLK			
59	—	—	PTB11	禁用		PTB11	SPI1_SCK	LPUART0_TX	I2S1_TX_FS			
60	—	—	VSS	VSS	VSS							
61	—	—	VDD	VDD	VDD							
62	39	31	PTB16	禁用		PTB16	SPI1_SOUT	UART0_RX	TPM_CLKIN0		EWM_IN	I2S1_TXD0 (Note: 100LQFP only)
63	40	—	PTB17	禁用		PTB17	SPI1_SIN	UART0_TX	TPM_CLKIN1		EWM_OUT_b	FXIO0_D0
64	41	32	PTB18	禁用		PTB18	CAN0_TX	TPM2_CH0	I2S0_TX_ BCLK			FXIO0_D1
65	42	33	PTB19	禁用		PTB19	CAN0_RX	TPM2_CH1	I2S0_TX_FS			FXIO0_D2
66	—	—	PTB20	禁用		PTB20					CMP0_OUT	FXIO0_D4
67	—	—	PTB21	禁用		PTB21					FXIO0_D5	
68	—	—	PTB22	禁用		PTB22					FXIO0_D6	
69	—	—	PTB23	禁用		PTB23		SPI0_PCS5			FXIO0_D7	
70	43	—	PTC0	ADC0_SE14	ADC0_SE14	PTC0	SPI0_PCS4	PDB0_EXTRG	USB_SOF_ OUT		FXIO0_D3	SPI0_PCS0

引脚分配

100 LQFP	64 LQFP	48 QFN	Pin Name	默认值	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
71	44	34	PTC1/ LLWU_P6	ADC0_SE15	ADC0_SE15	PTC1/ LLWU_P6	SPI0_PCS3	UART1_RTS_ b	TPM0_CH0		I2S0_TXD0	LPUART0_ RTS_b
72	45	35	PTC2	ADC0_SE4b	ADC0_SE4b	PTC2	SPI0_PCS2	UART1_CTS_ b	TPM0_CH1		I2S0_TX_FS	LPUART0_ CTS_b
73	46	36	PTC3/ LLWU_P7	禁用		PTC3/ LLWU_P7	SPI0_PCS1	UART1_RX	TPM0_CH2	CLKOUT	I2S0_TX_ BCLK	LPUART0_RX
74	47	—	VSS	VSS	VSS							
75	48	—	VDD	VDD	VDD							
76	49	37	PTC4/ LLWU_P8	禁用		PTC4/ LLWU_P8	SPI0_PCS0	UART1_TX	TPM0_CH3		LPI2C0_ HREQ	LPUART0_TX
77	50	38	PTC5/ LLWU_P9	禁用		PTC5/ LLWU_P9	SPI0_SCK	LPTMR0_ ALT2	I2S0_RXD0		CMP0_OUT	TPM0_CH2
78	51	39	PTC6/ LLWU_P10	CMP0_IN0	CMP0_IN0	PTC6/ LLWU_P10	SPI0_SOUT	PDB0_EXTRG	I2S0_RX_ BCLK		I2S0_MCLK	LPI2C0_SCL
79	52	40	PTC7	CMP0_IN1	CMP0_IN1	PTC7	SPI0_SIN	USB_SOF_ OUT	I2S0_RX_FS			LPI2C0_SDA
80	53	—	PTC8	CMP0_IN2	CMP0_IN2	PTC8	LPI2C0_SCLS		I2S0_MCLK		FXIO0_D0	I2S1_RXD0
81	54	—	PTC9	CMP0_IN3	CMP0_IN3	PTC9	LPI2C0_SDAS		I2S0_RX_ BCLK		FXIO0_D1	I2S1_RX_ BCLK
82	55	—	PTC10	禁用		PTC10	LPI2C1_SCL		I2S0_RX_FS		FXIO0_D2	I2S1_RX_FS
83	56	—	PTC11/ LLWU_P11	禁用		PTC11/ LLWU_P11	LPI2C1_SDA				FXIO0_D3	I2S1_MCLK
84	—	—	PTC12	禁用		PTC12	LPI2C1_SCLS		TPM_CLKIN0			FXIO0_D0
85	—	—	PTC13	禁用		PTC13	LPI2C1_SDAS		TPM_CLKIN1			FXIO0_D1
86	—	—	PTC14	禁用		PTC14			LPUART0_ RTS_b			FXIO0_D2
87	—	—	PTC15	禁用		PTC15			LPUART0_ CTS_b			FXIO0_D3
88	—	—	VSS	VSS	VSS							
89	—	—	VDD	VDD	VDD							
90	—	—	PTC16	禁用		PTC16	CAN1_RX	LPUART0_RX				FXIO0_D4
91	—	—	PTC17	禁用		PTC17	CAN1_TX	LPUART0_TX				FXIO0_D5
92	—	—	PTC18	禁用		PTC18		LPUART0_ RTS_b				
93	57	41	PTD0/ LLWU_P12	禁用		PTD0/ LLWU_P12	SPI0_PCS0	UART2_RTS_ b			LPUART0_ RTS_b	FXIO0_D6
94	58	42	PTD1	ADC0_SE5b	ADC0_SE5b	PTD1	SPI0_SCK	UART2_CTS_ b			LPUART0_ CTS_b	FXIO0_D7
95	59	43	PTD2/ LLWU_P13	禁用		PTD2/ LLWU_P13	SPI0_SOUT	UART2_RX			LPUART0_RX	LPI2C0_SCL
96	60	44	PTD3	禁用		PTD3	SPI0_SIN	UART2_TX			LPUART0_TX	LPI2C0_SDA
97	61	45	PTD4/ LLWU_P14	禁用		PTD4/ LLWU_P14	SPI0_PCS1	UART0_RTS_ b	TPM0_CH4		EWM_IN	SPI1_PCS0
98	62	46	PTD5	ADC0_SE6b	ADC0_SE6b	PTD5	SPI0_PCS2	UART0_CTS_ b	TPM0_CH5		EWM_OUT_b	SPI1_SCK

100 LQFP	64 LQFP	48 QFN	Pin Name	默认值	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
99	63	47	PTD6/ LLWU_P15	ADC0_SE7b	ADC0_SE7b	PTD6/ LLWU_P15	SPI0_PCS3	UART0_RX				SPI1_SOUT
100	64	48	PTD7	禁用		PTD7		UART0_TX				SPI1_SIN

11.2.2 引脚配置

下图显示的是本文档所支持芯片的引脚分配。多个信号可通过单个引脚进行复用。要确定每个引脚上可以使用哪些信号，请参见上一节“信号复用与引脚分配”。

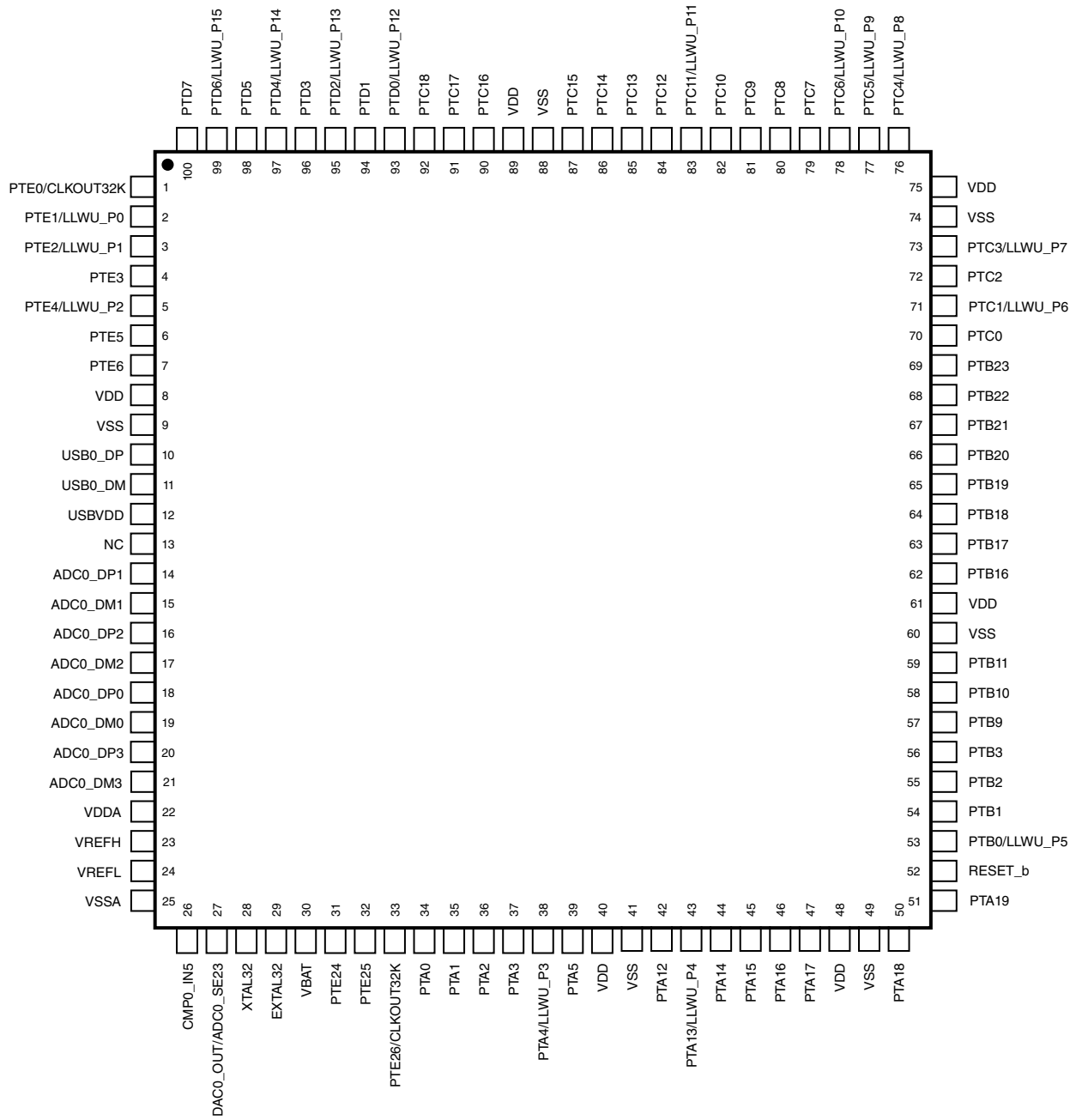


图 11-1. 100 LQFP 引脚分配图

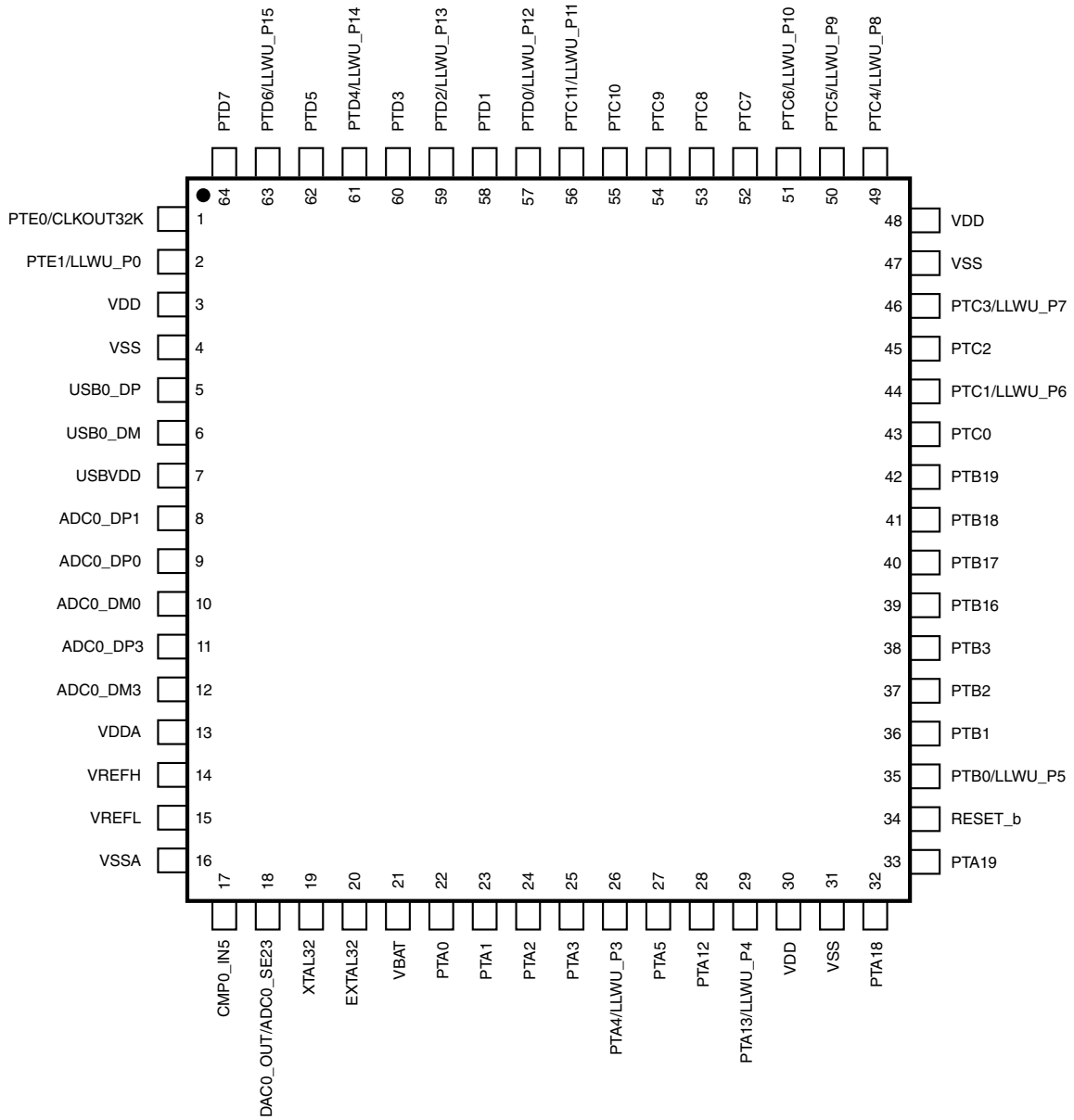


图 11-2. 64 LQFP 引脚分配图

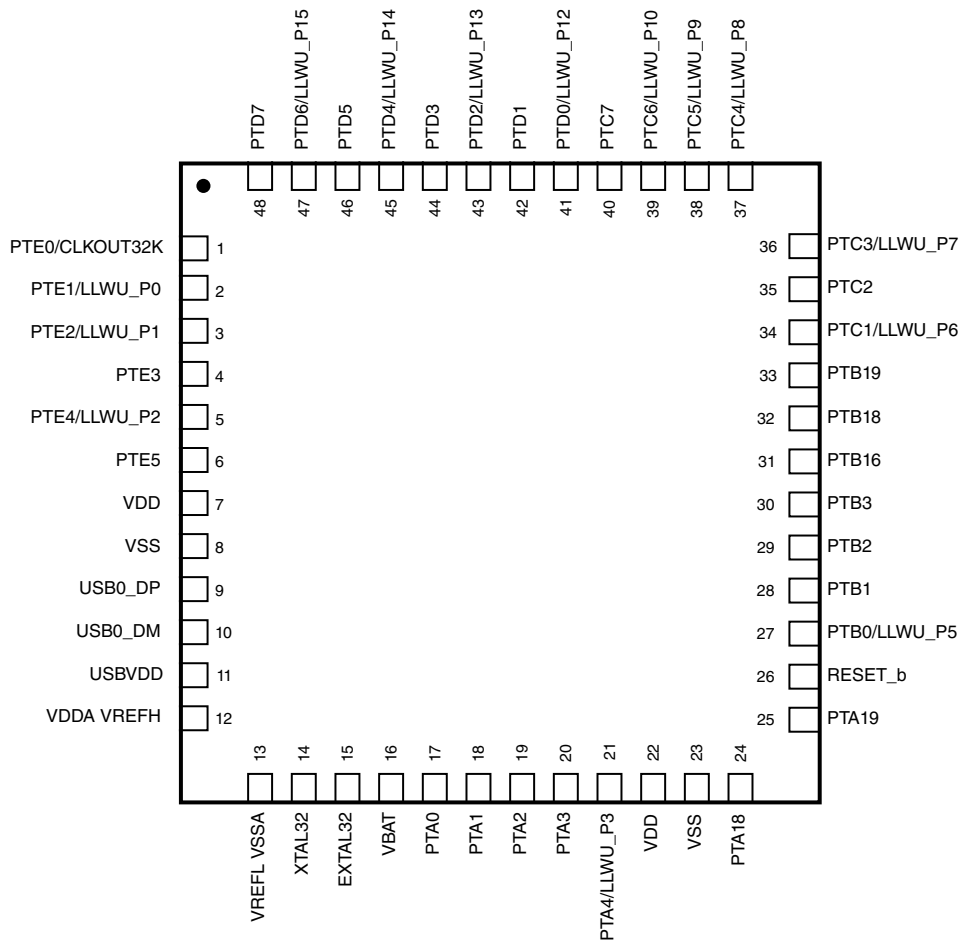


图 11-3. 48 QFN 引脚分配图

11.3 模块信号说明表

下面的章节说明芯片级信号名称与模块章节中使用的信号名称的关联。同时简要介绍信号功能和方向。

11.3.1 内核模块

表 11-1. JTAG 信号说明

芯片信号名称	模块信号名称	说明	I/O
JTAG_TMS	JTAG_TMS/ SWD_DIO	JTAG 测试模式选择	I
JTAG_TCLK	JTAG_TCLK / SWD_CLK	JTAG 测试时钟	I
JTAG_TDI	JTAG_TDI	JTAG 测试数据输入	I
JTAG_TDO	JTAG_TDO / TRACE_SWO	JTAG 测试数据输出	O
JTAG_TRST	JTAG_TRST_b	JTAG 复位	I

表 11-2. SWD 信号说明

芯片信号名称	模块信号名称	说明	I/O
SWD_DIO	JTAG_TMS/ SWD_DIO	串行线数据	I
SWD_CLK	JTAG_TCLK / SWD_CLK	串行线时钟	I

表 11-3. TPIU 信号说明

芯片信号名称	模块信号名称	说明	I/O
TRACE_SWO	JTAG_TDO / TRACE_SWO	通过单引脚跟踪 ARM CoreSight 调试块的输出数据	O

11.3.2 系统模块

表 11-4. EWM 信号说明

芯片信号名称	模块信号名称	说明	I/O
EWM_IN	EWM_in	用于外部安全电路安全状态的 EWM 输入。EWM_in 的极性可通过 EWM_CTRL[ASSIN]位进行编程。默认极性为低电平有效。	I
EWM_OUT	EWM_out	EWM 复位输出信号	O

11.3.3 时钟模块

表 11-5. OSC 信号说明

芯片信号名称	模块信号名称	说明	I/O
EXTAL0	EXTAL	外部时钟/振荡器输入	I
XTAL0	XTAL	振荡器输出	O

表 11-6. RTC OSC 信号说明

芯片信号名称	模块信号名称	说明	I/O
EXTAL32	EXTAL32	32.768 kHz 振荡器输入	I
XTAL32	XTAL32	32.768 kHz 振荡器输出	O

11.3.4 模拟

表 11-7. ADC 0 信号说明

芯片信号名称	模块信号名称	说明	I/O
ADC0_DP[3:0]	DADP3–DADP0	差分模拟通道输入	I
ADC0_DM[3:0]	DADM3–DADM0	差分模拟通道输入	I
ADC0_SEn	ADn	单端模拟通道输入	I
VREFH	V _{REFSH}	参考电压（高）	I
VREFL	V _{REFSL}	参考电压（低）	I
VDDA	V _{DDA}	模拟电源	I
VSSA	V _{SSA}	模拟接地	I

表 11-8. CMP 0 信号说明

芯片信号名称	模块信号名称	说明	I/O
CMP0_IN[5:0]	IN[5:0]	模拟电压输入	I
CMP0_OUT	CMPO	比较器输出	O

表 11-9. DAC 0 信号说明

芯片信号名称	模块信号名称	说明	I/O
DAC0_OUT	—	DAC 输出	O

11.3.5 定时器模块

表 11-10. PDB 0 信号说明

芯片信号名称	模块信号名称	说明	I/O
PDB0_EXTRG	EXTRG	外部触发器输入源 如果已使能 PDB 且已选择外部触发器输入源, 则 EXTRG 信号上的正边沿将复位并启动计数器。	I

表 11-11. LPTMR 0 信号说明

芯片信号名称	模块信号名称	说明	I/O
LPTMR0_ALT[2:1]	LPTMR0_ALT n	脉冲计数器输入引脚	I

表 11-12. RTC 信号说明

芯片信号名称	模块信号名称	说明	I/O
VBAT	—	RTC 和 VBAT 寄存器文件的备用电池电源	I
RTC_CLKOUT	RTC_CLKOUT	1 Hz 方波输出或 OSCERCLK	O

表 11-13. TPM 0 信号说明

芯片信号名称	模块信号名称	说明	I/O
TPM_CLKIN[1:0]	TPM_EXTCLK	外部时钟。当与计数器时钟同步后, TPM 外部时钟可让计数器在每个时钟上升沿递增。	I
TPM0_CH[5:0]	TPM_CH n	TPM 通道 ($n = 5$ 至 0)。在输出比较或 PWM 模式下且 TPM 计数器使能时, TPM 通道引脚将配置为输出, 否则 TPM 通道引脚为输入。	I/O

表 11-14. TPM 1 信号说明

芯片信号名称	模块信号名称	说明	I/O
TPM_CLKIN[1:0]	TPM_EXTCLK	外部时钟。当与计数器时钟同步后, TPM 外部时钟可让计数器在每个时钟上升沿递增。	I
TPM1_CH[1:0]	TPM_CH n	TPM 通道 ($n = 5$ 至 0)。在输出比较或 PWM 模式下且 TPM 计数器使能时, TPM 通道引脚将配置为输出, 否则 TPM 通道引脚为输入。	I/O

表 11-15. TPM 2 信号说明

芯片信号名称	模块信号名称	说明	I/O
TPM_CLKIN[1:0]	TPM_EXTCLK	外部时钟。当与计数器时钟同步后, TPM 外部时钟可让计数器在每个时钟上升沿递增。	I
TPM2_CH[1:0]	TPM_CH n	TPM 通道 ($n = 5$ 至 0)。在输出比较或 PWM 模式下且 TPM 计数器使能时, TPM 通道引脚将配置为输出, 否则 TPM 通道引脚为输入。	I/O

11.3.6 通信接口

表 11-16. USB FS OTG 信号说明

芯片信号名称	模块信号名称	说明	I/O
USB0_DM	usb_dm	USB 总线上的 USB D- 模拟数据信号。	I/O
USB0_DP	usb_dp	USB 总线上的 USB D+ 模拟数据信号。	I/O
USB_CLKIN	—	备用 USB 时钟输入	I
USB_SOF_OUT	—	USB 帧起始信号。可让 USB 帧起点可用于外部同步。	O

表 11-17. CAN 0 信号说明

芯片信号名称	模块信号名称	说明	I/O
CAN0_RX	CAN Rx	CAN 接收引脚	输入
CAN0_TX	CAN Tx	CAN 发送引脚	输出

表 11-18. CAN 1 (仅为 KS22) 信号说明

芯片信号名称	模块信号名称	说明	I/O
CAN1_RX	CAN Rx	CAN 接收引脚	输入
CAN1_TX	CAN Tx	CAN 发送引脚	输出

表 11-19. SPI 0 信号说明

芯片信号名称	模块信号名称	说明	I/O
SPI0_PCS0	PCS0/SS	外设芯片选择 0 (O)	I/O
SPI0_PCS[3:1]	PCS[1:3]	外设芯片选择 1–3	O
SPI0_PCS4	PCS4	外设芯片选择 4	O
SPI0_PCS5	PCS5/ PCSS	外设芯片选择 5 /外设芯片选择选通	O
SPI0_SIN	SIN	串行数据输入	I
SPI0_SOUT	SOUT	串行数据输出	O
SPI0_SCK	SCK	串行时钟(O)	I/O

表 11-20. SPI 1 信号说明

芯片信号名称	模块信号名称	说明	I/O
SPI1_PCS0	PCS0/SS	外设芯片选择 0 (O)	I/O
SPI1_PCS[3:1]	PCS[1:3]	外设芯片选择 1–3	O
SPI1_SIN	SIN	串行数据输入	I
SPI1_SOUT	SOUT	串行数据输出	O
SPI1_SCK	SCK	串行时钟(O)	I/O

表 11-21. LPI2C 0 信号说明

芯片信号名称	模块信号名称	说明	I/O
LPI2C0_SCL	SCL	LPI2C 时钟线路。	I/O
LPI2C0_SDA	SDA	LPI2C 数据线路。	I/O
LPI2C0_HREQ	HREQ	主机请求，可在置位且 I2C 总线闲置时启动 LPI2C 主机传输。	I
LPI2C0_SCLS	SCLS	辅助 I2C 时钟线路。如果将 LPI2C 主机/从机配置为使用单独引脚，则此为 LPI2C 从机 SCL 引脚。	I/O
LPI2C0_SDAS	SDAS	辅助 I2C 数据线路。如果将 LPI2C 主机/从机配置为使用单独引脚，则此为 LPI2C 从机 SDA 引脚。	I/O

表 11-22. LPI2C 1 信号说明

芯片信号名称	模块信号名称	说明	I/O
LPI2C1_SCL	SCL	LPI2C 时钟线路。	I/O
LPI2C1_SDA	SDA	LPI2C 数据线路。	I/O
LPI2C1_HREQ	HREQ	主机请求，可在置位且 I2C 总线闲置时启动 LPI2C 主机传输。	I
LPI2C1_SCLS	SCLS	辅助 I2C 时钟线路。如果将 LPI2C 主机/从机配置为使用单独引脚，则此为 LPI2C 从机 SCL 引脚。	I/O
LPI2C1_SDAS	SDAS	辅助 I2C 数据线路。如果将 LPI2C 主机/从机配置为使用单独引脚，则此为 LPI2C 从机 SDA 引脚。	I/O

表 11-23. LPUART 信号说明

芯片信号名称	模块信号名称	说明	I/O
LPUART0_TX	LPUART_TX	发送数据。此引脚一般为输出，但在单线模式（发射器禁用或发射方向配置为接收数据）下为输入（三态）。	O/I
LPUART0_RX	LPUART_RX	接收数据	I
LPUART0_CTS	LPUART_CTS	清除发送	I
LPUART0_CTS	LPUART_RTS	请求发送	I

表 11-24. UART 0 信号说明

芯片信号名称	模块信号名称	说明	I/O
UART0_CTS	$\overline{\text{CTS}}$	清除发送	I
UART0_RTS	$\overline{\text{RTS}}$	请求发送	O
UART0_TX	TXD	发送数据	O
UART0_RX	RXD	接收数据	I

表 11-25. UART 1 信号说明

芯片信号名称	模块信号名称	说明	I/O
UART1_CTS	$\overline{\text{CTS}}$	清除发送	I

下一页继续介绍此表...

表 11-25. UART 1 信号说明 (继续)

芯片信号名称	模块信号名称	说明	I/O
UART1_RTS	RTS	请求发送	O
UART1_TX	TXD	发送数据	O
UART1_RX	RXD	接收数据	I

表 11-26. UART 2 信号说明

芯片信号名称	模块信号名称	说明	I/O
UART2_CTS	CTS	清除发送	I
UART2_RTS	RTS	请求发送	O
UART2_TX	TXD	发送数据	O
UART2_RX	RXD	接收数据	I

表 11-27. I²S0 信号说明

芯片信号名称	模块信号名称	说明	I/O
I2S0_MCLK	SAI_MCLK	音频主机时钟。从外部生成时，此主机时钟为输入；从内部生成时，为输出。	I/O
I2S0_RX_BCLK	SAI_RX_BCLK	接收位时钟。从外部生成时，此位时钟为输入；从内部生成时，为输出。	I/O
I2S0_RX_FS	SAI_RX_SYNC	接收帧同步。从外部生成时，此帧同步为位时钟同步采样的输入；从内部生成时，为位时钟同步生成的输出。	I/O
I2S0_RXD	SAI_RX_DATA	接收数据。接收数据由位时钟同步采样。	I
I2S0_TX_BCLK	SAI_TX_BCLK	传送位时钟。从外部生成时，此位时钟为输入；从内部生成时，为输出。	I/O
I2S0_TX_FS	SAI_TX_SYNC	发送帧同步。从外部生成时，此帧同步为位时钟同步采样的输入；从内部生成时，为位时钟同步生成的输出。	I/O
I2S0_TXD	SAI_TX_DATA	发送数据。发送数据由位时钟同步生成，当未发送字时为三态。	O

表 11-28. I²S1 信号说明

芯片信号名称	模块信号名称	说明	I/O
I2S1_MCLK	SAI_MCLK	音频主机时钟。从外部生成时，此主机时钟为输入；从内部生成时，为输出。	I/O
I2S1_RX_BCLK	SAI_RX_BCLK	接收位时钟。从外部生成时，此位时钟为输入；从内部生成时，为输出。	I/O
I2S1_RX_FS	SAI_RX_SYNC	接收帧同步。从外部生成时，此帧同步为位时钟同步采样的输入；从内部生成时，为位时钟同步生成的输出。	I/O
I2S1_RXD	SAI_RX_DATA	接收数据。接收数据由位时钟同步采样。	I
I2S1_TX_BCLK	SAI_TX_BCLK	传送位时钟。从外部生成时，此位时钟为输入；从内部生成时，为输出。	I/O
I2S1_TX_FS	SAI_TX_SYNC	发送帧同步。从外部生成时，此帧同步为位时钟同步采样的输入；从内部生成时，为位时钟同步生成的输出。	I/O
I2S1_TXD	SAI_TX_DATA	发送数据。发送数据由位时钟同步生成，当未发送字时为三态。	O

表 11-29. FlexIO 信号说明

芯片信号名称	模块信号名称	说明	I/O
FXIO0_Dn	FXIO_Dn (n=0...7)	双向 FlexIO 移位器和定时器引脚输入/输出	I/O

11.3.7 人机接口 (HMI)

表 11-30. GPIO 信号说明

芯片信号名称	模块信号名称	说明	I/O
PTA[31:0]	PORTA31–PORTA0	通用输入/输出	I/O
PTB[31:0] ¹	PORTB31–PORTB0	通用输入/输出	I/O
PTC[31:0] ¹	PORTC31–PORTC0	通用输入/输出	I/O
PTD[31:0] ¹	PORTD31–PORTD0	通用输入/输出	I/O
PTE[31:0] ¹	PORTE31–PORTE0	通用输入/输出	I/O

1. 可用 GPIO 引脚取决于特定封装。有关具体哪些 GPIO 信号可用，请参见“信号复用”部分。

第 12 章 端口控制和中断 (PORT)

12.1 此模块的芯片实现细节

12.1.1 信号复用集成

本节总结如何将模块集成至芯片。有关模块本身的全面阐述，请参见各模块相关章节。

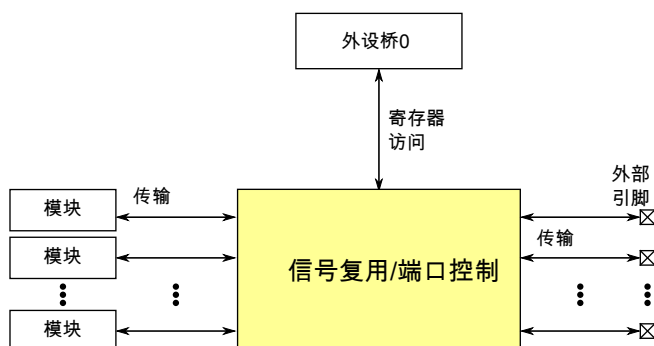


图 12-1. 信号复用集成

表 12-1. 相关信息的参考链接

主题	相关模块	参考
完整说明	端口控制	端口控制
系统存储器映射		系统存储器映射
时钟		时钟分配
寄存器访问	外设总线控制器	外设桥

12.1.1.1 端口控制和中断模块特性

- 32 引脚端口

注

器件未提供全部引脚。详情请参见以下章节。

- 每个 32 引脚端口均配有一个中断。

表 12-2. 端口汇总

功能	端口 A	端口 B	端口 C	端口 D	端口 E
上拉或下拉选择控制	有	有	有	有	有
复位时的上拉或下拉选择	PTA1/PTA2/PTA3/ PTA4=上拉, 其他= 下拉	下拉	下拉	下拉	下拉
上拉或下拉使能控制	有	有	有	有	有
复位时的上拉或下拉使能	PTA0/PTA1/PTA2/ PTA3/PTA4=使能; 其他=禁用	禁用	禁用	禁用	禁用
压摆率使能控制	是	有	有	有	是
复位时的压摆率使能	禁用	禁用	禁用	禁用	禁用
无源滤波器使能控制	PTA4=是; 其它=否	否	无	无	否
复位时的无源滤波器使能	禁用	禁用	禁用	禁用	禁用
开漏使能控制	有	有	有	有	是
复位时的开漏使能	禁用	禁用	禁用	禁用	禁用
驱动强度使能控制	否	仅限于 PTB0/PTB1	仅限于 PTC3/PTC4	仅限于 PTD4/ PTD5/PTD6/PTD7	否
复位时的驱动强度使能	禁用	禁用	禁用	禁用	禁用
引脚复用控制	有	有	有	有	有
复位时的引脚复用	PTA0/PTA1/PTA2/ PTA3/ PTA4=ALT7; 其他 =ALT0	ALT0	ALT0	ALT0	ALT0
锁定位	有	有	有	有	有
中断和 DMA 请求	支持	有	有	有	支持
数字毛刺滤波器	无	无	无	是	否

注

芯片只有端口 D 提供数字滤波器。与数字滤波器相关的寄存器（例如 DFER、DFCR、DFWR）不适用于端口 A、B、C 和 E。

12.1.1.2 时钟门控

可使用 SIM 模块中的 SCGC5[PORTx] 位开启和关闭端口控制模块的时钟。可在任何复位后清除这些位，禁用相应模块的时钟以节省电力。在初始化对应模块之前，将 SIM 模块中的 SCGC5[PORTx] 位置位以使能时钟。关闭时钟前，务必禁用该模块。详情请参阅时钟分布章节。

12.1.1.3 信号复用限制

1. 给定外设功能最多只能分配给一个封装引脚。请勿将同一功能分配给一个以上的引脚。
2. 为确保给定外设接口的信号时序最佳，应选择相互最接近的引脚。

12.2 简介

12.3 概述

端口控制和中断 (PORT) 模块可以为端口控制、数字滤波、和外部中断功能提供支持。

大部分功能可以针对 32 位端口中的每个引脚单独配置，无论引脚复用状态如何，均能对引脚产生影响。

每个端口有一个 PORT 模块实例。并非每个端口中的所有引脚都实现在具体每个芯片上。

12.3.1 特性

PORT 模块具有以下特性：

- 引脚中断
 - 每个引脚均有中断标志和使能寄存器
 - 支持每个引脚配置边沿敏感（上升、下降、二者兼而有之）或电平敏感（低、高）选项
 - 支持每个引脚配置中断或 DMA 请求
 - 低功耗模式下的异步唤醒
 - 引脚中断在所有数字引脚复用模式下均有效
- 选定引脚上的数字输入滤波器
 - 每个引脚的数字输入滤波器，可供多路复用至引脚的任何数字外设使用
 - 每个引脚的单独使能或旁路控制字段

- 数字输入滤波器的可选时钟源，且滤波器大小为五位分辨率
- 在所有数字引脚多路复用模式下均有效
- 端口控制
 - 有单独的上/下拉控制字段，支持上拉、下拉和上/下拉禁用
 - 有单独的驱动强度字段，支持高和低驱动强度
 - 有单独的压摆率字段，支持快速和慢速压摆率
 - 有单独的输入无源滤波器字段，支持各输入无源滤波器的使能和禁用
 - 单独开漏字段，支持使能和禁用上的单个开漏输出
 - 单独的多路复用控制字段，支持模拟或引脚禁用、GPIO 和最多 6 个芯片特性数字功能
- 端口配置字段在所有数字引脚多路复用模式下均有效

12.3.2 工作模式

12.3.2.1 运行模式

在运行模式下，PORT 正常运行。

12.3.2.2 等待模式

在等待模式下，PORT 继续正常工作并且可配置为当检测到已使能中断时退出低功耗模式。在等待下仍会生成 DMA 请求，但不会退出低功耗模式。

12.3.2.3 停止模式

在停止模式下，PORT 可配置为当检测到已使能中断时通过异步唤醒信号退出低功耗模式。

在停止模式下，将旁通数字输入滤波器，除非将其配置为从 LPO 时钟源运行。

12.3.2.4 调试模式

在调试模式下，PORT 正常运行。

12.4 外部信号说明

本表对 PORT 外部信号加以说明。

表 12-3. 信号属性

名称	函数	I/O	复位	上/下拉
PORTx[31:0]	外部中断	I/O	0	-

注

并非每个端口内的所有引脚都在不同芯片上实现。

12.5 详细信号描述

本表格包含 PORT 接口的详细信号说明。

表 12-4. PORT 接口 — 详细信号说明

信号	I/O	说明	
PORTx[31:0]	I/O	外部中断。	
		状态含义	有效 — 引脚为逻辑 1。 无效 — 引脚为逻辑 0。
		时序	有效声明 — 可能在任何时刻发生, 且可能和系统时钟不同步。 无效声明 — 可能在任何时刻发生, 且可能和系统时钟不同步。

12.6 存储器映射和寄存器定义

在有效存储器映射外对 PORT 存储器空间进行任何读写访问都将导致总线错误。所有寄存器访问都能在零等待状态下完成。

PORT 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4004_9000	引脚控制寄存器 n (PORTA_PCR0)	32	R/W	参见章节	12.6.1/177
4004_9004	引脚控制寄存器 n (PORTA_PCR1)	32	R/W	参见章节	12.6.1/177
4004_9008	引脚控制寄存器 n (PORTA_PCR2)	32	R/W	参见章节	12.6.1/177
4004_900C	引脚控制寄存器 n (PORTA_PCR3)	32	R/W	参见章节	12.6.1/177
4004_9010	引脚控制寄存器 n (PORTA_PCR4)	32	R/W	参见章节	12.6.1/177

下一页继续介绍此表...

PORT 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4004_9014	引脚控制寄存器 n (PORTA_PCR5)	32	R/W	参见章节	12.6.1/177
4004_9018	引脚控制寄存器 n (PORTA_PCR6)	32	R/W	参见章节	12.6.1/177
4004_901C	引脚控制寄存器 n (PORTA_PCR7)	32	R/W	参见章节	12.6.1/177
4004_9020	引脚控制寄存器 n (PORTA_PCR8)	32	R/W	参见章节	12.6.1/177
4004_9024	引脚控制寄存器 n (PORTA_PCR9)	32	R/W	参见章节	12.6.1/177
4004_9028	引脚控制寄存器 n (PORTA_PCR10)	32	R/W	参见章节	12.6.1/177
4004_902C	引脚控制寄存器 n (PORTA_PCR11)	32	R/W	参见章节	12.6.1/177
4004_9030	引脚控制寄存器 n (PORTA_PCR12)	32	R/W	参见章节	12.6.1/177
4004_9034	引脚控制寄存器 n (PORTA_PCR13)	32	R/W	参见章节	12.6.1/177
4004_9038	引脚控制寄存器 n (PORTA_PCR14)	32	R/W	参见章节	12.6.1/177
4004_903C	引脚控制寄存器 n (PORTA_PCR15)	32	R/W	参见章节	12.6.1/177
4004_9040	引脚控制寄存器 n (PORTA_PCR16)	32	R/W	参见章节	12.6.1/177
4004_9044	引脚控制寄存器 n (PORTA_PCR17)	32	R/W	参见章节	12.6.1/177
4004_9048	引脚控制寄存器 n (PORTA_PCR18)	32	R/W	参见章节	12.6.1/177
4004_904C	引脚控制寄存器 n (PORTA_PCR19)	32	R/W	参见章节	12.6.1/177
4004_9050	引脚控制寄存器 n (PORTA_PCR20)	32	R/W	参见章节	12.6.1/177
4004_9054	引脚控制寄存器 n (PORTA_PCR21)	32	R/W	参见章节	12.6.1/177
4004_9058	引脚控制寄存器 n (PORTA_PCR22)	32	R/W	参见章节	12.6.1/177
4004_905C	引脚控制寄存器 n (PORTA_PCR23)	32	R/W	参见章节	12.6.1/177
4004_9060	引脚控制寄存器 n (PORTA_PCR24)	32	R/W	参见章节	12.6.1/177
4004_9064	引脚控制寄存器 n (PORTA_PCR25)	32	R/W	参见章节	12.6.1/177
4004_9068	引脚控制寄存器 n (PORTA_PCR26)	32	R/W	参见章节	12.6.1/177
4004_906C	引脚控制寄存器 n (PORTA_PCR27)	32	R/W	参见章节	12.6.1/177
4004_9070	引脚控制寄存器 n (PORTA_PCR28)	32	R/W	参见章节	12.6.1/177
4004_9074	引脚控制寄存器 n (PORTA_PCR29)	32	R/W	参见章节	12.6.1/177
4004_9078	引脚控制寄存器 n (PORTA_PCR30)	32	R/W	参见章节	12.6.1/177
4004_907C	引脚控制寄存器 n (PORTA_PCR31)	32	R/W	参见章节	12.6.1/177
4004_9080	全局引脚控制低位寄存器 (PORTA_GPCLR)	32	W (始终读 0)	0000_0000h	12.6.2/180
4004_9084	全局引脚控制低位寄存器 (PORTA_GPCHR)	32	W (始终读 0)	0000_0000h	12.6.3/180
4004_90A0	中断状态标志寄存器 (PORTA_ISFR)	32	w1c	0000_0000h	12.6.4/181
4004_90C0	数字滤波器使能寄存器 (PORTA_DFER)	32	R/W	0000_0000h	12.6.5/181
4004_90C4	数字滤波器时钟寄存器 (PORTA_DFCR)	32	R/W	0000_0000h	12.6.6/182
4004_90C8	数字滤波器宽度寄存器 (PORTA_DFWR)	32	R/W	0000_0000h	12.6.7/182
4004_A000	引脚控制寄存器 n (PORTB_PCR0)	32	R/W	参见章节	12.6.1/177
4004_A004	引脚控制寄存器 n (PORTB_PCR1)	32	R/W	参见章节	12.6.1/177
4004_A008	引脚控制寄存器 n (PORTB_PCR2)	32	R/W	参见章节	12.6.1/177
4004_A00C	引脚控制寄存器 n (PORTB_PCR3)	32	R/W	参见章节	12.6.1/177

下一页继续介绍此表...

PORT 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4004_A010	引脚控制寄存器 n (PORTB_PCR4)	32	R/W	参见章节	12.6.1/177
4004_A014	引脚控制寄存器 n (PORTB_PCR5)	32	R/W	参见章节	12.6.1/177
4004_A018	引脚控制寄存器 n (PORTB_PCR6)	32	R/W	参见章节	12.6.1/177
4004_A01C	引脚控制寄存器 n (PORTB_PCR7)	32	R/W	参见章节	12.6.1/177
4004_A020	引脚控制寄存器 n (PORTB_PCR8)	32	R/W	参见章节	12.6.1/177
4004_A024	引脚控制寄存器 n (PORTB_PCR9)	32	R/W	参见章节	12.6.1/177
4004_A028	引脚控制寄存器 n (PORTB_PCR10)	32	R/W	参见章节	12.6.1/177
4004_A02C	引脚控制寄存器 n (PORTB_PCR11)	32	R/W	参见章节	12.6.1/177
4004_A030	引脚控制寄存器 n (PORTB_PCR12)	32	R/W	参见章节	12.6.1/177
4004_A034	引脚控制寄存器 n (PORTB_PCR13)	32	R/W	参见章节	12.6.1/177
4004_A038	引脚控制寄存器 n (PORTB_PCR14)	32	R/W	参见章节	12.6.1/177
4004_A03C	引脚控制寄存器 n (PORTB_PCR15)	32	R/W	参见章节	12.6.1/177
4004_A040	引脚控制寄存器 n (PORTB_PCR16)	32	R/W	参见章节	12.6.1/177
4004_A044	引脚控制寄存器 n (PORTB_PCR17)	32	R/W	参见章节	12.6.1/177
4004_A048	引脚控制寄存器 n (PORTB_PCR18)	32	R/W	参见章节	12.6.1/177
4004_A04C	引脚控制寄存器 n (PORTB_PCR19)	32	R/W	参见章节	12.6.1/177
4004_A050	引脚控制寄存器 n (PORTB_PCR20)	32	R/W	参见章节	12.6.1/177
4004_A054	引脚控制寄存器 n (PORTB_PCR21)	32	R/W	参见章节	12.6.1/177
4004_A058	引脚控制寄存器 n (PORTB_PCR22)	32	R/W	参见章节	12.6.1/177
4004_A05C	引脚控制寄存器 n (PORTB_PCR23)	32	R/W	参见章节	12.6.1/177
4004_A060	引脚控制寄存器 n (PORTB_PCR24)	32	R/W	参见章节	12.6.1/177
4004_A064	引脚控制寄存器 n (PORTB_PCR25)	32	R/W	参见章节	12.6.1/177
4004_A068	引脚控制寄存器 n (PORTB_PCR26)	32	R/W	参见章节	12.6.1/177
4004_A06C	引脚控制寄存器 n (PORTB_PCR27)	32	R/W	参见章节	12.6.1/177
4004_A070	引脚控制寄存器 n (PORTB_PCR28)	32	R/W	参见章节	12.6.1/177
4004_A074	引脚控制寄存器 n (PORTB_PCR29)	32	R/W	参见章节	12.6.1/177
4004_A078	引脚控制寄存器 n (PORTB_PCR30)	32	R/W	参见章节	12.6.1/177
4004_A07C	引脚控制寄存器 n (PORTB_PCR31)	32	R/W	参见章节	12.6.1/177
4004_A080	全局引脚控制低位寄存器 (PORTB_GPCLR)	32	W (始终读 0)	0000_0000h	12.6.2/180
4004_A084	全局引脚控制低位寄存器 (PORTB_GPCHR)	32	W (始终读 0)	0000_0000h	12.6.3/180
4004_A0A0	中断状态标志寄存器 (PORTB_ISFR)	32	w1c	0000_0000h	12.6.4/181
4004_A0C0	数字滤波器使能寄存器 (PORTB_DFER)	32	R/W	0000_0000h	12.6.5/181
4004_A0C4	数字滤波器时钟寄存器 (PORTB_DFCCR)	32	R/W	0000_0000h	12.6.6/182
4004_A0C8	数字滤波器宽度寄存器 (PORTB_DFWR)	32	R/W	0000_0000h	12.6.7/182
4004_B000	引脚控制寄存器 n (PORTC_PCR0)	32	R/W	参见章节	12.6.1/177
4004_B004	引脚控制寄存器 n (PORTC_PCR1)	32	R/W	参见章节	12.6.1/177
4004_B008	引脚控制寄存器 n (PORTC_PCR2)	32	R/W	参见章节	12.6.1/177

下一页继续介绍此表...

PORT 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4004_B00C	引脚控制寄存器 n (PORTC_PCR3)	32	R/W	参见章节	12.6.1/177
4004_B010	引脚控制寄存器 n (PORTC_PCR4)	32	R/W	参见章节	12.6.1/177
4004_B014	引脚控制寄存器 n (PORTC_PCR5)	32	R/W	参见章节	12.6.1/177
4004_B018	引脚控制寄存器 n (PORTC_PCR6)	32	R/W	参见章节	12.6.1/177
4004_B01C	引脚控制寄存器 n (PORTC_PCR7)	32	R/W	参见章节	12.6.1/177
4004_B020	引脚控制寄存器 n (PORTC_PCR8)	32	R/W	参见章节	12.6.1/177
4004_B024	引脚控制寄存器 n (PORTC_PCR9)	32	R/W	参见章节	12.6.1/177
4004_B028	引脚控制寄存器 n (PORTC_PCR10)	32	R/W	参见章节	12.6.1/177
4004_B02C	引脚控制寄存器 n (PORTC_PCR11)	32	R/W	参见章节	12.6.1/177
4004_B030	引脚控制寄存器 n (PORTC_PCR12)	32	R/W	参见章节	12.6.1/177
4004_B034	引脚控制寄存器 n (PORTC_PCR13)	32	R/W	参见章节	12.6.1/177
4004_B038	引脚控制寄存器 n (PORTC_PCR14)	32	R/W	参见章节	12.6.1/177
4004_B03C	引脚控制寄存器 n (PORTC_PCR15)	32	R/W	参见章节	12.6.1/177
4004_B040	引脚控制寄存器 n (PORTC_PCR16)	32	R/W	参见章节	12.6.1/177
4004_B044	引脚控制寄存器 n (PORTC_PCR17)	32	R/W	参见章节	12.6.1/177
4004_B048	引脚控制寄存器 n (PORTC_PCR18)	32	R/W	参见章节	12.6.1/177
4004_B04C	引脚控制寄存器 n (PORTC_PCR19)	32	R/W	参见章节	12.6.1/177
4004_B050	引脚控制寄存器 n (PORTC_PCR20)	32	R/W	参见章节	12.6.1/177
4004_B054	引脚控制寄存器 n (PORTC_PCR21)	32	R/W	参见章节	12.6.1/177
4004_B058	引脚控制寄存器 n (PORTC_PCR22)	32	R/W	参见章节	12.6.1/177
4004_B05C	引脚控制寄存器 n (PORTC_PCR23)	32	R/W	参见章节	12.6.1/177
4004_B060	引脚控制寄存器 n (PORTC_PCR24)	32	R/W	参见章节	12.6.1/177
4004_B064	引脚控制寄存器 n (PORTC_PCR25)	32	R/W	参见章节	12.6.1/177
4004_B068	引脚控制寄存器 n (PORTC_PCR26)	32	R/W	参见章节	12.6.1/177
4004_B06C	引脚控制寄存器 n (PORTC_PCR27)	32	R/W	参见章节	12.6.1/177
4004_B070	引脚控制寄存器 n (PORTC_PCR28)	32	R/W	参见章节	12.6.1/177
4004_B074	引脚控制寄存器 n (PORTC_PCR29)	32	R/W	参见章节	12.6.1/177
4004_B078	引脚控制寄存器 n (PORTC_PCR30)	32	R/W	参见章节	12.6.1/177
4004_B07C	引脚控制寄存器 n (PORTC_PCR31)	32	R/W	参见章节	12.6.1/177
4004_B080	全局引脚控制低位寄存器 (PORTC_GPCLR)	32	W (始终读 0)	0000_0000h	12.6.2/180
4004_B084	全局引脚控制低位寄存器 (PORTC_GPCHR)	32	W (始终读 0)	0000_0000h	12.6.3/180
4004_B0A0	中断状态标志寄存器 (PORTC_ISFR)	32	w1c	0000_0000h	12.6.4/181
4004_B0C0	数字滤波器使能寄存器 (PORTC_DFER)	32	R/W	0000_0000h	12.6.5/181
4004_B0C4	数字滤波器时钟寄存器 (PORTC_DFCCR)	32	R/W	0000_0000h	12.6.6/182
4004_B0C8	数字滤波器宽度寄存器 (PORTC_DFWR)	32	R/W	0000_0000h	12.6.7/182
4004_C000	引脚控制寄存器 n (PORTD_PCR0)	32	R/W	参见章节	12.6.1/177
4004_C004	引脚控制寄存器 n (PORTD_PCR1)	32	R/W	参见章节	12.6.1/177

下一页继续介绍此表...

PORT 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4004_C008	引脚控制寄存器 n (PORTD_PCR2)	32	R/W	参见章节	12.6.1/177
4004_C00C	引脚控制寄存器 n (PORTD_PCR3)	32	R/W	参见章节	12.6.1/177
4004_C010	引脚控制寄存器 n (PORTD_PCR4)	32	R/W	参见章节	12.6.1/177
4004_C014	引脚控制寄存器 n (PORTD_PCR5)	32	R/W	参见章节	12.6.1/177
4004_C018	引脚控制寄存器 n (PORTD_PCR6)	32	R/W	参见章节	12.6.1/177
4004_C01C	引脚控制寄存器 n (PORTD_PCR7)	32	R/W	参见章节	12.6.1/177
4004_C020	引脚控制寄存器 n (PORTD_PCR8)	32	R/W	参见章节	12.6.1/177
4004_C024	引脚控制寄存器 n (PORTD_PCR9)	32	R/W	参见章节	12.6.1/177
4004_C028	引脚控制寄存器 n (PORTD_PCR10)	32	R/W	参见章节	12.6.1/177
4004_C02C	引脚控制寄存器 n (PORTD_PCR11)	32	R/W	参见章节	12.6.1/177
4004_C030	引脚控制寄存器 n (PORTD_PCR12)	32	R/W	参见章节	12.6.1/177
4004_C034	引脚控制寄存器 n (PORTD_PCR13)	32	R/W	参见章节	12.6.1/177
4004_C038	引脚控制寄存器 n (PORTD_PCR14)	32	R/W	参见章节	12.6.1/177
4004_C03C	引脚控制寄存器 n (PORTD_PCR15)	32	R/W	参见章节	12.6.1/177
4004_C040	引脚控制寄存器 n (PORTD_PCR16)	32	R/W	参见章节	12.6.1/177
4004_C044	引脚控制寄存器 n (PORTD_PCR17)	32	R/W	参见章节	12.6.1/177
4004_C048	引脚控制寄存器 n (PORTD_PCR18)	32	R/W	参见章节	12.6.1/177
4004_C04C	引脚控制寄存器 n (PORTD_PCR19)	32	R/W	参见章节	12.6.1/177
4004_C050	引脚控制寄存器 n (PORTD_PCR20)	32	R/W	参见章节	12.6.1/177
4004_C054	引脚控制寄存器 n (PORTD_PCR21)	32	R/W	参见章节	12.6.1/177
4004_C058	引脚控制寄存器 n (PORTD_PCR22)	32	R/W	参见章节	12.6.1/177
4004_C05C	引脚控制寄存器 n (PORTD_PCR23)	32	R/W	参见章节	12.6.1/177
4004_C060	引脚控制寄存器 n (PORTD_PCR24)	32	R/W	参见章节	12.6.1/177
4004_C064	引脚控制寄存器 n (PORTD_PCR25)	32	R/W	参见章节	12.6.1/177
4004_C068	引脚控制寄存器 n (PORTD_PCR26)	32	R/W	参见章节	12.6.1/177
4004_C06C	引脚控制寄存器 n (PORTD_PCR27)	32	R/W	参见章节	12.6.1/177
4004_C070	引脚控制寄存器 n (PORTD_PCR28)	32	R/W	参见章节	12.6.1/177
4004_C074	引脚控制寄存器 n (PORTD_PCR29)	32	R/W	参见章节	12.6.1/177
4004_C078	引脚控制寄存器 n (PORTD_PCR30)	32	R/W	参见章节	12.6.1/177
4004_C07C	引脚控制寄存器 n (PORTD_PCR31)	32	R/W	参见章节	12.6.1/177
4004_C080	全局引脚控制低位寄存器 (PORTD_GPCLR)	32	W (始终读 0)	0000_0000h	12.6.2/180
4004_C084	全局引脚控制低位寄存器 (PORTD_GPCHR)	32	W (始终读 0)	0000_0000h	12.6.3/180
4004_C0A0	中断状态标志寄存器 (PORTD_ISFR)	32	w1c	0000_0000h	12.6.4/181
4004_C0C0	数字滤波器使能寄存器 (PORTD_DFER)	32	R/W	0000_0000h	12.6.5/181
4004_C0C4	数字滤波器时钟寄存器 (PORTD_DFCR)	32	R/W	0000_0000h	12.6.6/182
4004_C0C8	数字滤波器宽度寄存器 (PORTD_DFWR)	32	R/W	0000_0000h	12.6.7/182
4004_D000	引脚控制寄存器 n (PORTE_PCR0)	32	R/W	参见章节	12.6.1/177

下一页继续介绍此表...

PORT 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4004_D004	引脚控制寄存器 n (PORTE_PCR1)	32	R/W	参见章节	12.6.1/177
4004_D008	引脚控制寄存器 n (PORTE_PCR2)	32	R/W	参见章节	12.6.1/177
4004_D00C	引脚控制寄存器 n (PORTE_PCR3)	32	R/W	参见章节	12.6.1/177
4004_D010	引脚控制寄存器 n (PORTE_PCR4)	32	R/W	参见章节	12.6.1/177
4004_D014	引脚控制寄存器 n (PORTE_PCR5)	32	R/W	参见章节	12.6.1/177
4004_D018	引脚控制寄存器 n (PORTE_PCR6)	32	R/W	参见章节	12.6.1/177
4004_D01C	引脚控制寄存器 n (PORTE_PCR7)	32	R/W	参见章节	12.6.1/177
4004_D020	引脚控制寄存器 n (PORTE_PCR8)	32	R/W	参见章节	12.6.1/177
4004_D024	引脚控制寄存器 n (PORTE_PCR9)	32	R/W	参见章节	12.6.1/177
4004_D028	引脚控制寄存器 n (PORTE_PCR10)	32	R/W	参见章节	12.6.1/177
4004_D02C	引脚控制寄存器 n (PORTE_PCR11)	32	R/W	参见章节	12.6.1/177
4004_D030	引脚控制寄存器 n (PORTE_PCR12)	32	R/W	参见章节	12.6.1/177
4004_D034	引脚控制寄存器 n (PORTE_PCR13)	32	R/W	参见章节	12.6.1/177
4004_D038	引脚控制寄存器 n (PORTE_PCR14)	32	R/W	参见章节	12.6.1/177
4004_D03C	引脚控制寄存器 n (PORTE_PCR15)	32	R/W	参见章节	12.6.1/177
4004_D040	引脚控制寄存器 n (PORTE_PCR16)	32	R/W	参见章节	12.6.1/177
4004_D044	引脚控制寄存器 n (PORTE_PCR17)	32	R/W	参见章节	12.6.1/177
4004_D048	引脚控制寄存器 n (PORTE_PCR18)	32	R/W	参见章节	12.6.1/177
4004_D04C	引脚控制寄存器 n (PORTE_PCR19)	32	R/W	参见章节	12.6.1/177
4004_D050	引脚控制寄存器 n (PORTE_PCR20)	32	R/W	参见章节	12.6.1/177
4004_D054	引脚控制寄存器 n (PORTE_PCR21)	32	R/W	参见章节	12.6.1/177
4004_D058	引脚控制寄存器 n (PORTE_PCR22)	32	R/W	参见章节	12.6.1/177
4004_D05C	引脚控制寄存器 n (PORTE_PCR23)	32	R/W	参见章节	12.6.1/177
4004_D060	引脚控制寄存器 n (PORTE_PCR24)	32	R/W	参见章节	12.6.1/177
4004_D064	引脚控制寄存器 n (PORTE_PCR25)	32	R/W	参见章节	12.6.1/177
4004_D068	引脚控制寄存器 n (PORTE_PCR26)	32	R/W	参见章节	12.6.1/177
4004_D06C	引脚控制寄存器 n (PORTE_PCR27)	32	R/W	参见章节	12.6.1/177
4004_D070	引脚控制寄存器 n (PORTE_PCR28)	32	R/W	参见章节	12.6.1/177
4004_D074	引脚控制寄存器 n (PORTE_PCR29)	32	R/W	参见章节	12.6.1/177
4004_D078	引脚控制寄存器 n (PORTE_PCR30)	32	R/W	参见章节	12.6.1/177
4004_D07C	引脚控制寄存器 n (PORTE_PCR31)	32	R/W	参见章节	12.6.1/177
4004_D080	全局引脚控制低位寄存器 (PORTE_GPCLR)	32	W (始终读 0)	0000_0000h	12.6.2/180
4004_D084	全局引脚控制低位寄存器 (PORTE_GPCHR)	32	W (始终读 0)	0000_0000h	12.6.3/180
4004_D0A0	中断状态标志寄存器 (PORTE_ISFR)	32	w1c	0000_0000h	12.6.4/181
4004_D0C0	数字滤波器使能寄存器 (PORTE_DFER)	32	R/W	0000_0000h	12.6.5/181
4004_D0C4	数字滤波器时钟寄存器 (PORTE_DFCCR)	32	R/W	0000_0000h	12.6.6/182
4004_D0C8	数字滤波器宽度寄存器 (PORTE_DFWR)	32	R/W	0000_0000h	12.6.7/182

12.6.1 引脚控制寄存器 n (PORTx_PCRn)

注

有关该器件的复位值信息，请参阅信号复用和引脚分配章节。

有关每个引脚可用功能的详情，请参阅通用 IO 配置部分。

请不要修改与选定封装中不可用引脚相关的引脚配置寄存器。封装中不可用的未绑定引脚将默认处于禁用状态，以实现最低功耗。

地址: 基址 + 0h 偏移 + (4d × i), 其中 i=0d 到 31d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ISF	0				IRQC			
W								w1c								
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LK	0			MUX				0	DSE	ODE	PFE	0	SRE	PE	PS
W																
复位	0	0	0	0	*	*	*	*	0	*	0	*	0	*	*	*

* 注:

- MUX 字段: 随端口变化。有关每个端口的复位值，请参见“信号复用”和“信号说明”章节。
- DSE 字段: 随端口变化。有关每个端口的复位值，请参见“信号复用”和“信号说明”章节。
- PFE 字段: 随端口变化。有关每个端口的复位值，请参见“信号复用”和“信号说明”章节。
- SRE 字段: 随端口变化。有关每个端口的复位值，请参见“信号复用”和“信号说明”章节。
- PE 字段: 随端口变化。有关每个端口的复位值，请参见“信号复用”和“信号说明”章节。
- PS 字段: 随端口变化。有关每个端口的复位值，请参见“信号复用”和“信号说明”章节。

PORTx_PCRn 字段描述

字段	描述
31–25 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
24 ISF	中断状态标志 在所有数字引脚复用模式下，引脚中断配置有效。 0 未检测到已配置的中断。 1 检测到已配置的中断。如果引脚配置为生成 DMA 请求，则在完成所请求的 DMA 传输之后将自动清除对应的标志。否则，标志位将保持置位，直到向标志写入逻辑 1。如果引脚配置为电平触发中断且引脚保持有效，则标志将在清零之后立即再次置位。

下一页继续介绍此表...

PORTx_PCRn 字段描述 (继续)

字段	描述
23–20 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
19–16 IRQC	<p>中断配置</p> <p>在所有数字引脚复用模式下，引脚中断配置有效。对应引脚采用如下配置以生成中断/DMA 请求：</p> <p>0000 中断状态标志 (ISF) 已禁用。 0001 上升沿上的 ISF 标志和 DMA 请求。 0010 下降沿上的 ISF 标志和 DMA 请求。 0011 任意沿上的 ISF 标志和 DMA 请求。 0100 保留。 0101 保留。 0110 保留。 0111 保留。 1000 逻辑为 0 时的 ISF 标志和中断。 1001 上升沿上的 ISF 标志和中断。 1010 下降沿上的 ISF 标志和中断。 1011 任意沿上的 ISF 标志和中断。 1100 逻辑为 1 时的 ISF 标志和中断。 1101 保留。 1110 保留。 1111 保留。</p>
15 LK	<p>锁定寄存器</p> <p>0 引脚控制寄存器字段 [15:0] 未锁定。 1 引脚控制寄存器字段 [15:0] 锁定且在下次系统复位之前无法更新。</p>
14–12 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
11–8 MUX	<p>引脚复用控制</p> <p>并非所有引脚都支持所有引脚复用槽。保留未生效的引脚复用插槽可能导致为引脚配置不同的引脚复用槽。 对应引脚采用如下引脚复用槽配置：</p> <p>0000 引脚禁用。 0001 备选 1 (GPIO)。 0010 备用 2 (芯片特性)。 0011 备用 3 (芯片特性)。 0100 备用 4 (芯片特性)。 0101 备用 5 (芯片特性)。 0110 备用 6 (芯片特性)。 0111 备用 7 (芯片特性)。 1000 备选 1 (GPIO)。 1001 备用 8 (芯片特性)。 1010 备用 10 (芯片特性)。 1011 备用 11 (芯片特性)。 1100 备用 12 (芯片特性)。 1101 备用 13 (芯片特性)。</p>

下一页继续介绍此表...

PORTx_PCRn 字段描述 (继续)

字段	描述
	1110 备用 14 (芯片特性)。 1111 备用 15 (芯片特性)。
7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
6 DSE	驱动强度使能 在所有数字引脚复用模式下，驱动强度配置有效。 0 如果引脚配置为数字输出，则对应引脚上的驱动强度将配置为弱驱动。 1 如果引脚配置为数字输出，则对应引脚上的驱动强度将配置为强驱动。
5 ODE	开漏使能 在所有数字引脚复用模式下，开漏配置有效。 0 对应引脚上的开漏输出禁用。 1 如果引脚配置为数字输出，则对应引脚上的开漏输出使能。
4 PFE	无源滤波器使能 在所有数字引脚复用模式下，无源滤波器配置有效。 0 对应引脚上的无源输入滤波器禁用。 1 如果引脚配置为数字输入，则对应引脚上的无源输入滤波器使能。有关滤波器特性的信息，请参阅芯片数据手册。
3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 SRE	压摆率使能 在所有数字引脚复用模式下，压摆率配置有效。 0 如果引脚配置为数字输入，则对应引脚上的压摆率将配置为快速。 1 如果引脚配置为数字输出，则对应引脚上的压摆率将配置为慢速。
1 PE	上/下拉使能 在所有数字引脚复用模式下，上/下拉配置有效。 0 对应引脚上的内部上拉或下拉 电阻未使能。 1 如果引脚配置为数字输入，则对应引脚上的内部上拉 或下拉 电阻使能。
0 PS	上/下拉选择 在所有数字引脚复用模式下，上/下拉配置有效。 0 如果置位对应的 PE 字段，则对应引脚上的内部下拉电阻使能。 1 如果置位对应的 PE 字段，则对应引脚上的内部上拉电阻使能。

12.6.2 全局引脚控制低位寄存器 (PORTx_GPCLR)

此寄存器只支持 32 位写入。

地址: 基址 基准 + 80h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W	GPWE																GPWD															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_GPCLR 字段描述

字段	描述
31-16 GPWE	<p>全局引脚写入使能</p> <p>选择哪个引脚控制寄存器 (31 至 16) 位[15:0] 采用 GPWD 中的值进行更新。如果选中的引脚控制寄存器被锁定，则向该寄存器的写入将被忽略。</p> <p>0 对应的引脚控制寄存器不使用 GPWD 中的值进行更新。 1 对应的引脚控制寄存器使用 GPWD 中的值进行更新。</p>
GPWD	<p>全局引脚写数据</p> <p>写入在所有被 GPWE 选中的引脚控制寄存器位 [15:0] 中写入的值。</p>

12.6.3 全局引脚控制低位寄存器 (PORTx_GPCHR)

此寄存器只支持 32 位写入。

地址: 基址 基准 + 84h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W	GPWE																GPWD															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_GPCHR 字段描述

字段	描述
31-16 GPWE	<p>全局引脚写入使能</p> <p>选择哪个引脚控制寄存器 (31 至 16) 位[15:0] 采用 GPWD 中的值进行更新。如果选中的引脚控制寄存器被锁定，则向该寄存器的写入将被忽略。</p> <p>0 对应的引脚控制寄存器不使用 GPWD 中的值进行更新。 1 对应的引脚控制寄存器使用 GPWD 中的值进行更新。</p>
GPWD	<p>全局引脚写数据</p> <p>写入在所有被 GPWE 选中的引脚控制寄存器位 [15:0] 中写入的值。</p>

12.6.4 中断状态标志寄存器 (PORTx_ISFR)

在所有数字引脚复用模式下，引脚中断配置有效。每个引脚的中断状态标志在对应的引脚控制寄存器中也可见，且每个标志都可在任意位置清除。

地址: 基址 基准 + A0h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISF																															
W	w1c																															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_ISFR 字段描述

字段	描述
ISF	<p>中断状态标志</p> <p>字段中的每个位表示检测到与字段相同的已配置中断数。</p> <p>0 未检测到已配置的中断。</p> <p>1 检测到已配置的中断。如果引脚配置为生成 DMA 请求，则在完成所请求的 DMA 传输之后将自动清除对应的标志。否则，标志位将保持置位，直到向标志写入逻辑 1。如果引脚配置为电平触发中断且引脚保持有效，则标志将在清零之后立即再次置位。</p>

12.6.5 数字滤波器使能寄存器 (PORTx_DFER)

对于不支持数字滤波器的引脚而言，对应位为只读位。参阅信号复用和信号说明章节，了解支持数字滤波器的引脚的信息。

在所有数字引脚复用模式下，数字滤波器配置有效。

地址: 基址 基准 + C0h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DFE																															
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_DFER 字段描述

字段	描述
DFE	<p>数字滤波器使能</p> <p>在所有数字引脚复用模式下，数字滤波器配置有效。在系统复位以及每次数字滤波器禁用时，每个数字滤波器的输出复位为零。字段中的每个位使能与字段数字相同的数字过滤器。</p> <p>0 对应引脚上的数字滤波器禁用且数字滤波器输出将复位为 0。</p> <p>1 如果引脚配置为数字输入，则对应引脚上的数字滤波器使能。</p>

12.6.6 数字滤波器时钟寄存器 (PORTx_DFCR)

对于不支持数字滤波器的端口而言，此寄存器为只读寄存器。
 在所有数字引脚复用模式下，数字滤波器配置有效。

地址: 基址 基准 + C4h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															CS
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_DFCR 字段描述

字段	描述
31-1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
0 CS	时钟源 在所有数字引脚复用模式下，数字滤波器配置有效。配置数字输入滤波器的时钟源。只有在所有数字滤波器都禁用时才能更改滤波器时钟源。 0 数字滤波器由总线时钟计时。 1 数字滤波器由 LPO 时钟计时。

12.6.7 数字滤波器宽度寄存器 (PORTx_DFWR)

对于不支持数字滤波器的端口而言，此寄存器为只读寄存器。
 在所有数字引脚复用模式下，数字滤波器配置有效。

地址: 基址 基准 + C8h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																FILT															
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_DFWR 字段描述

字段	描述
31-5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

PORTx_DFWR 字段描述 (继续)

字段	描述
FILT	<p>滤波器长度</p> <p>在所有数字引脚复用模式下，数字滤波器配置有效。配置时钟周期中数字滤波器为已使能数字滤波器吸收的毛刺的最大尺寸。大于此寄存器设置的毛刺将通过数字滤波器，而等于或小于此寄存器设置的毛刺将被滤除。必须在所有滤波器禁用时才更改滤波器长度。</p>

12.7 功能说明

12.7.1 引脚控制

每个端口引脚具有一个与其关联的对应引脚控制寄存器 PORT_PCRn。

该引脚控制寄存器的上半部分用于配置引脚的功能，以中断 CPU 或请求在上升/下降沿或两者上进行 DMA 传输，以及配置为端口引脚上发生的逻辑电平。它还包括一个用于表示已发生中断的标志。

引脚控制寄存器的下半部分用于配置 32 位端口内每个引脚的以下功能。

- 上拉或下拉使能
- 驱动强度和压摆率配置
- 开漏使能
- 无源输入滤波器使能
- 引脚复用模式

这些功能应用于所有数字引脚复用模式，并且单独外设不会覆盖引脚控制寄存器。例如，如果引脚上的 I²C 功能使能，则不会覆盖该引脚的上拉或开漏配置。

当引脚复用模式配置为模拟或禁用时，该引脚上的所有数字功能将禁用。这包括上拉和下拉使能、输出缓冲区使能、输入缓冲区使能以及无源滤波使能。

LK 位（引脚控制寄存器 PCR_n 的引脚 15）支持每个引脚的配置锁定，直到系统复位。锁定后，尽管对锁定寄存器进行的尝试写入操作未生成总线错误，但对该引脚控制寄存器下半部分的写入操作将被忽略。

PORT 模块禁用时，每个引脚控制寄存器的配置保留。

每当引脚在任何数字引脚复用模式下配置时，该引脚的输入缓冲区使能，允许通过对应通用 IO 端口数据输入寄存器 (GPIO_PDIR) 读取引脚状态，或者允许生成引脚中断或 DMA 请求。如果引脚缓冲区使能时引脚总是浮空，这可能导致功耗增加，必须避免。引脚可能因为输入引脚未连接或输出引脚为三态（输出缓冲区禁用）而浮空。

使能内部上/下拉电阻（或设置外部上/下拉电阻）将确保引脚在其输入缓冲区使能时不浮空；注意当外部缓冲区使能时内部上/下拉电阻自动禁用，允许上/下拉使能位保持置位。配置引脚复用模式为禁用或模拟将禁用引脚的输入缓冲区，达到最低功耗。

12.7.2 全局引脚控制

两个全局引脚控制寄存器允许单个寄存器在多至 16 个引脚上写入更新引脚控制寄存器的下半部分，全部引脚具有相同值。锁定的寄存器无法使用全局引脚控制寄存器写入。

全局引脚控制寄存器经过设计，可使软件在一个端口内为相同外设功能快速配置多个引脚。但是，中断功能无法使用全局引脚控制寄存器配置。

全局引脚控制寄存器为只写寄存器，始终读取为 0。

12.7.3 外部中断

只要 PORT 模块使能，则 PORT 模块的外部中断功能在所有数字引脚复用模式下均可用。

每个引脚可以针对以下外部中断模式单独配置：

- 中断禁用，复位后默认
- 有效高级别敏感中断
- 有效低级别敏感中断
- 上升沿敏感中断
- 下降沿敏感中断
- 上升和下降沿敏感中断
- 上升沿敏感 DMA 请求
- 下降沿敏感 DMA 请求
- 上升和下降沿敏感 DMA 请求

当在引脚或数字输入滤波器的输出端检测到配置边沿或电平时，如果数字输入滤波器已使能，则会置位中断状态标志。如果未处于停止模式，则输入将首先同步至总线时钟，以检测配置的电平或边沿转换。

PORT 模式产生单个中断，当端口的任何使能的中断置位中断状态标志位时有效。可以通过将逻辑 1 写入 PORT_ISFR 或 PORT_PCRn 的 ISF 标志位，来清空任何使能的中断的状态标志位，使中断无效。

PORT 模式产生单个 DMA 请求，当端口的任何使能中断，置位 DMA 请求状态标志位时有效。DMA 传输完成后 DMA 就无效了，因为这置位了所有使能 DMA 请求的中断状态标志。

在停止模式下，如果检测到所需电平或边沿，则会异步置位任何已使能中断的中断状态标志。这也会产生退出低功耗模式的异步唤醒信号。

12.7.4 数字滤波器

如果 PORT 模块已使能，则 PORT 模块的数字滤波器功能可用于所有数字引脚复用模式。

可以将某一端口内所有数字滤波器使用的时钟配置为总线时钟或 LPO 时钟。仅当该端口的所有数字滤波器禁用时才必须更改此选择。如果某一端口的数字滤波器配置为使用总线时钟，则停止模式期间数字滤波器将被旁通。数字滤波器旁通时，每个数字滤波器的输出始终等于输入引脚，但数字滤波器的内部状态保持静态且不会因输入引脚上的任何变化而更新。

在一个端口中使能的所有数字滤波器的滤波宽度（以时钟大小为单位）相同，只有在禁用该端口的所有数字滤波器时才能更改此宽度。

系统复位后以及每次数字滤波器禁用时，每个数字滤波器的输出为逻辑零。数字滤波器使能后，输入会同步到滤波器时钟，其为总线时钟或 LPO 时钟。如果在与滤波宽度寄存器配置等同数量的滤波器时钟周期中同步输入和数字滤波器输出不同，则数字滤波器输出更新至与同步滤波输入相等。

通过数字滤波器的最长延迟等于三个滤波器时钟周期加滤波宽度配置寄存器。

第 13 章

系统集成模块 (SIM)

13.1 简介

系统集成模块 (SIM) 提供系统控制和芯片配置寄存器。

13.1.1 特性

SIM 的特性包括:

- 系统时钟配置
 - 系统时钟分频值
 - 架构时钟选通控制
 - USB 时钟选择和分频值
 - FlexIO 时钟源选择
 - LPI2C 时钟源选择
 - TPM 时钟源选择
 - LPUART 时钟源选择
- Flash 和系统 RAM 大小配置
- TPM 外部时钟选择, 通道 0 输入捕获源选择
- UART0 和 UART1 接收/发送源选择/配置
- LPUART 发送源选择
- USB 引脚上的 UART 选择
- FlexIO 时钟槽 0 选择

13.2 存储器映射和寄存器定义

SIM 模块包含许多字段，用来为不同的模块时钟选择时钟源和分频器。有关更多信息（包括结构框图和时钟定义），请参见[时钟分布](#)一章。

注

SIM_SOPT1 寄存器位于与其他 SIM 寄存器不同的基地址中。

SIM 存储器映射

绝对地址（十六进制）	寄存器名称	宽度（单位：位）	访问	复位值	小节/页
4004_7000	系统选项寄存器 1 (SIM_SOPT1)	32	R/W	参见章节	13.2.1/189
4004_8004	系统选项寄存器 2 (SIM_SOPT2)	32	R/W	0000_1000h	13.2.2/190
4004_8010	系统选项寄存器 5 (SIM_SOPT5)	32	R/W	0000_0000h	13.2.3/192
4004_8018	系统选项寄存器 7 (SIM_SOPT7)	32	R/W	0000_0000h	13.2.4/194
4004_8020	系统选项寄存器 9 (SIM_SOPT9)	32	R/W	0000_0000h	13.2.5/195
4004_8024	系统能够设备标识寄存器 (SIM_SDID)	32	R	参见章节	13.2.6/197
4004_8034	系统时钟门控寄存器 4 (SIM_SCGC4)	32	R/W	F000_0030h	13.2.7/199
4004_8038	系统时钟门控寄存器 5 (SIM_SCGC5)	32	R/W	0004_0182h	13.2.8/201
4004_803C	系统时钟门控寄存器 6 (SIM_SCGC6)	32	R/W	4000_0001h	13.2.9/202
4004_8040	系统时钟门控寄存器 7 (SIM_SCGC7)	32	R/W	0000_0002h	13.2.10/206
4004_8044	系统时钟分频器寄存器 1 (SIM_CLKDIV1)	32	R/W	参见章节	13.2.11/206
4004_8048	系统时钟分频器寄存器 2 (SIM_CLKDIV2)	32	R/W	0000_0000h	13.2.12/208
4004_804C	Flash 配置寄存器 1 (SIM_FCFG1)	32	R	参见章节	13.2.13/209
4004_8050	Flash 配置寄存器 2 (SIM_FCFG2)	32	R	参见章节	13.2.14/211
4004_8054	高电平寄存器唯一标识 (SIM_UIDH)	32	R	参见章节	13.2.15/211
4004_8058	中高电平寄存器唯一标识 (SIM_UIDMH)	32	R	参见章节	13.2.16/212
4004_805C	中低电平寄存器唯一标识 (SIM_UIDML)	32	R	参见章节	13.2.17/212
4004_8060	低电平寄存器唯一标识 (SIM_UIDL)	32	R	参见章节	13.2.18/213
4004_8064	系统时钟分频器寄存器 3 (SIM_CLKDIV3)	32	R/W	0000_0000h	13.2.19/213
4004_806C	其他控制寄存器 (SIM_MISCCTL)	32	R/W	0000_0003h	13.2.20/214

13.2.1 系统选项寄存器 1 (SIM_SOPT1)

注

SOPT1 寄存器仅可在 POR 或 LVD 上复位。

地址: 4004_7000h 基准 + 0h 偏移 = 4004_7000h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				0								OSC32KSEL		OSC32KOUT	
W																
复位	x*	x*	x*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RAMSIZE				0				0							
W																
复位	x*	x*	x*	x*	0*	0*	0*	0*	0*	0*	x*	x*	x*	x*	x*	x*

* 注:

- 重置值在系统从 Flash IFR 复位时加载。
- x = 复位时未定义。

SIM_SOPT1 字段描述

字段	描述
31–29 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
28–20 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
19–18 OSC32KSEL	32K 振荡器时钟选择 为选择 32 kHz 时钟源 (ERCLK32K)。仅可在 POR/LVD 上复位此字段。 00 系统振荡器 (OSC32KCLK) 01 保留 10 RTC 32.768kHz 振荡器 11 LPO 1 kHz
17–16 OSC32KOUT	32K 振荡器时钟输出 所有模式 (包括 LLS/VLLS 和系统复位) 下在选定引脚上输出 ERCLK32K, 覆盖引脚现有的引脚复用配置。 仅可在 POR/LVD 上复位此字段。 00 ERCLK32K 不是输出。 01 ERCLK32K 是 PTE0 上的输出。 10 ERCLK32K 是 PTE26 上的输出。 11 保留。
15–12 RAMSIZE	RAM 大小 该字段用于指定器件上可用的系统 RAM 大小。

下一页继续介绍此表...

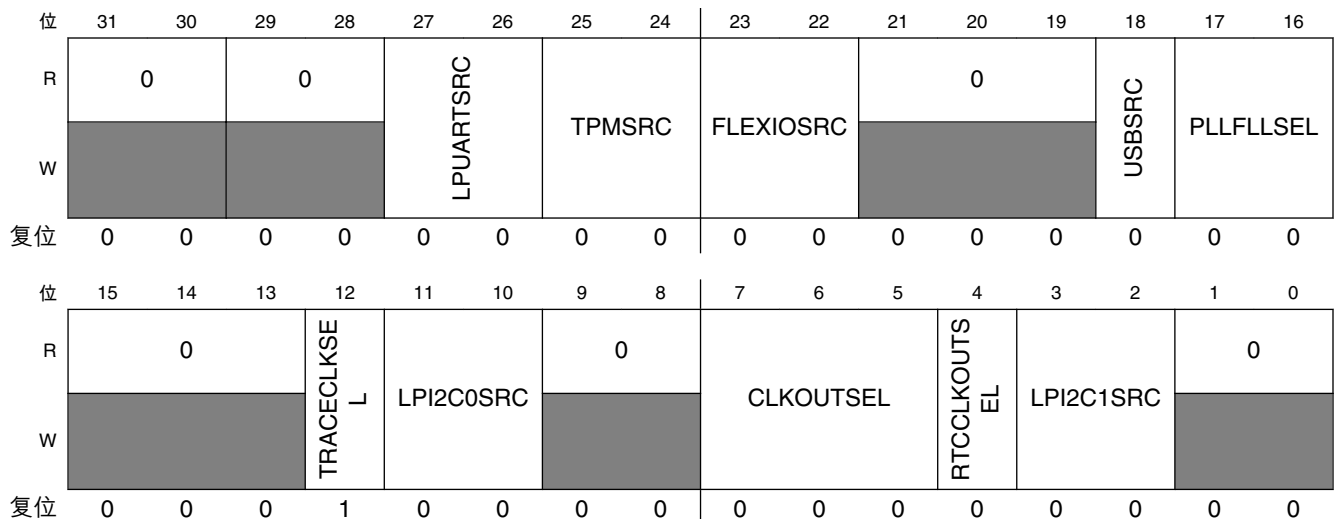
SIM_SOPT1 字段描述 (继续)

字段	描述
0001	8 KB
0011	16 KB
0100	24 KB
0101	32 KB
0110	48 KB
0111	64 KB
1000	96 KB
1001	128 KB
1011	256 KB
11-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

13.2.2 系统选项寄存器 2 (SIM_SOPT2)

SOPT2 包含用于选择器件上的多个模块时钟源选项的控制。有关更多信息 (包括时钟示意图和器件时钟定义), 请参见时钟分配章节。

地址: 4004_7000h 基准 + 1004h 偏移 = 4004_8004h



SIM_SOPT2 字段描述

字段	描述
31-30 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
29-28 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

SIM_SOPT2 字段描述 (继续)

字段	描述
27-26 LPUARTSRC	<p>LPUART 时钟源选择</p> <p>选择 LPUART 发送与接收时钟的时钟源。</p> <p>00 时钟禁用 01 SOPT2[PLLFLSEL] 选择 MCGFLLCLK、MCGPLLCLK、或 IRC48M 时钟。 10 OSCERCLK 时钟 11 MCGIRCLK 时钟</p>
25-24 TPMSRC	<p>TPM 时钟源选择</p> <p>选择 TPM 计数器时钟的时钟源。</p> <p>00 时钟禁用 01 SOPT2[PLLFLSEL] 选择 MCGFLLCLK、MCGPLLCLK、或 IRC48M 时钟。 10 OSCERCLK 时钟 11 MCGIRCLK 时钟</p>
23-22 FLEXIOSRC	<p>FlexIO 模块时钟源选择</p> <p>选择 FlexIO 发送与接收时钟的时钟源。</p> <p>00 I2S0_MCLK 或系统时钟，通过 SIM_MISCCTRL[FIEXIOS0]选择。 01 SOPT2[PLLFLSEL] 选择 MCGFLLCLK、MCGPLLCLK、或 IRC48M 时钟。 10 OSCERCLK 时钟 11 MCGIRCLK 时钟</p>
21-19 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
18 USBSRC	<p>USB 时钟源选择</p> <p>选择 USB 48 MHz 时钟的时钟源。</p> <p>0 外部旁路时钟 (USB_CLKIN)。 1 SOPT2[PLLFLSEL]选择 MCGFLLCLK、MCGPLLCLK、IRC48M 然后 USB 小数分频器按照 SIM_CLKDIV2[USBFRAC, USBDIV] 的配置分频。</p>
17-16 PLLFLSEL	<p>PLL/FLL 时钟选择</p> <p>选择高频时钟作为各种外设时钟选项。</p> <p>00 MCGFLLCLK 时钟 01 MCGPLLCLK 时钟 10 Reserved 11 IRC48 MHz clock</p>
15-13 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
12 TRACECLKSEL	<p>调试跟踪时钟选择</p> <p>选择内核/系统时钟，或 MCG 输出时钟 (MCGOUTCLK) 作为跟踪时钟源。</p> <p>0 MCGOUTCLK 1 内核/系统时钟</p>
11-10 LPI2C0SRC	<p>LPI2C0 源</p>

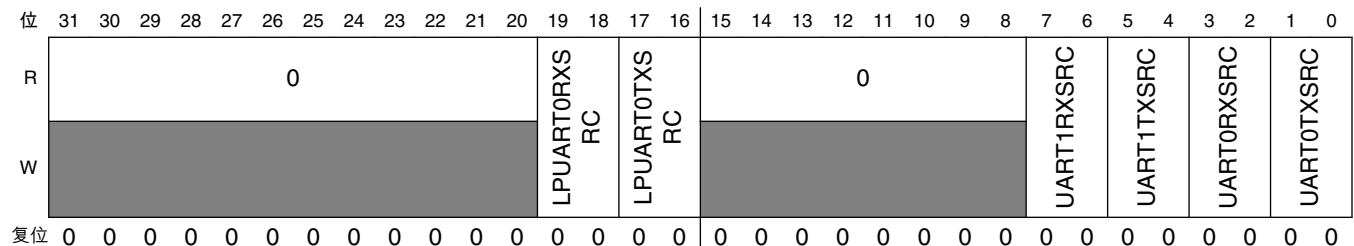
下一页继续介绍此表...

SIM_SOPT2 字段描述 (继续)

字段	描述
	00 时钟禁用 01 SOPT2[PLLFLSEL] 选择 MCGFLLCLK、MCGPLLCLK、或 IRC48M 时钟。 10 OSCERCLK 时钟 11 MCGIRCLK 时钟
9-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7-5 CLKOUTSEL	CLKOUT 选择 选择此时钟作为 CLKOUT 引脚上的输出。 000 保留 001 保留 010 Flash 时钟 011 LPO 时钟 (1 kHz) 100 MCGIRCLK 101 RTC 32.768kHz 时钟 110 OSCERCLK0 111 IRC 48 MHz 时钟
4 RTCCLKOUTSEL	RTC 时钟输出检测 选择 RTC 1 Hz 时钟或 32.768kHz 时钟作为 RTC_CLKOUT 引脚上的输出。 0 RTC 1 Hz 时钟为 RTC_CLKOUT 引脚上的输出。 1 RTC 32.768kHz 时钟为 RTC_CLKOUT 引脚上的输出。
3-2 LPI2C1SRC	LPI2C1 源 00 时钟禁用 01 SOPT2[PLLFLSEL] 选择 MCGFLLCLK、MCGPLLCLK、或 IRC48M 时钟。 10 OSCERCLK 时钟 11 MCGIRCLK 时钟
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

13.2.3 系统选项寄存器 5 (SIM_SOPT5)

地址: 4004_7000h 基准 + 1010h 偏移 = 4004_8010h



SIM_SOPT5 字段描述

字段	描述
31-20 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
19-18 LPUART0RXSRC	LPUART0 接收数据源选择 选择 LPUART0 接收数据的源。 00 LPUART0_RX 引脚 01 CMP0 输出 10 保留 11 保留
17-16 LPUART0TXSRC	LPUART0 发送数据源选择 选择 UART0 发送数据的源。 00 LPUART0_TX 引脚 01 LPUART0_TX 引脚, 通过 TPM1 通道 0 输出调制 10 LPUART0_TX 引脚, 通过 TPM2 通道 0 输出调制 11 保留
15-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7-6 UART1RXSRC	UART 1 接收数据源选择 选择 UART 1 接收数据的源。 00 UART1_RX 引脚 01 CMP0 输出 10 保留 11 保留
5-4 UART1TXSRC	UART 1 发送数据源选择 选择 UART 1 发送数据的源。 00 UART1_TX 引脚 01 UART1_TX 引脚, 通过 TPM1 通道 0 输出调制 10 UART1_TX 引脚, 通过 TPM2 通道 0 输出调制解调 11 保留
3-2 UART0RXSRC	UART 0 接收数据源选择 选择 UART 0 接收数据的源。 00 UART0_RX 引脚 01 CMP0 输出 10 保留 11 保留
UART0TXSRC	UART 0 发送数据源选择 选择 UART 0 发送数据的源。 00 UART0_TX 引脚 01 UART0_TX 引脚, 通过 TPM1 通道 0 输出调制

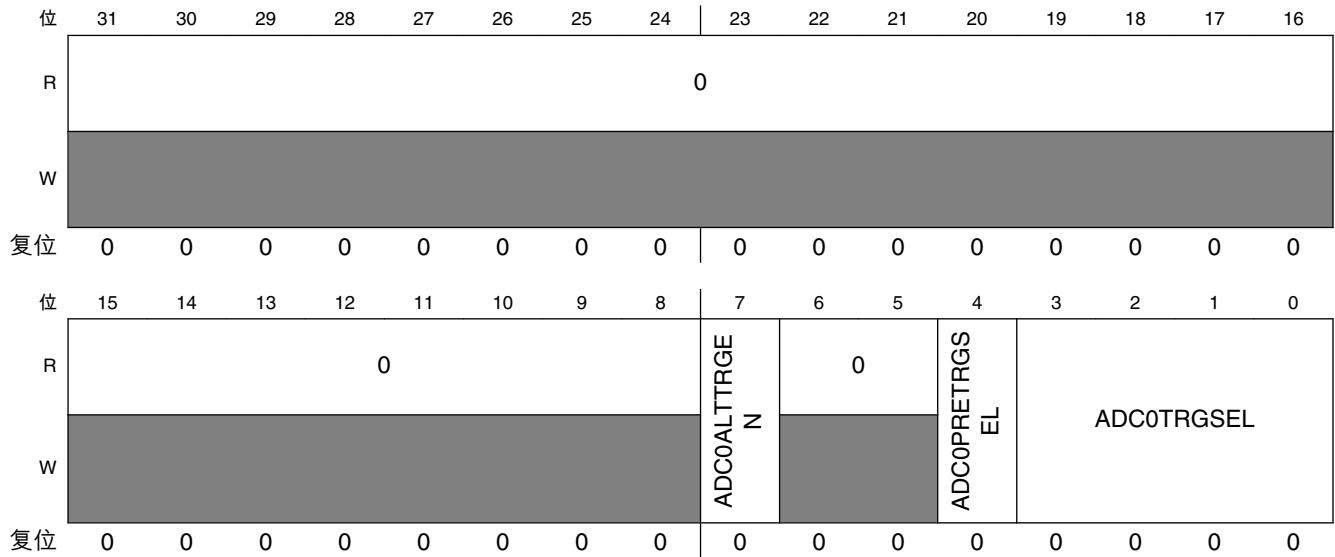
下一页继续介绍此表...

SIM_SOPT5 字段描述 (继续)

字段	描述
10	UART0_TX 引脚，通过 TPM2 通道 0 输出调制
11	保留

13.2.4 系统选项寄存器 7 (SIM_SOPT7)

地址: 4004_7000h 基准 + 1018h 偏移 = 4004_8018h



SIM_SOPT7 字段描述

字段	描述
31–16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15–8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 ADC0ALTTRGEN	ADC0 备用触发器使能 使能 ADC0 的备用转换触发器。 0 选择 ADC0 的 PDB 触发器。 1 选择 ADC0 的备用触发器。
6–5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
4 ADC0PRETRGSEL	ADC0 预触发器选择 当备用触发器通过 ADC0ALTTRGEN 使能时，选择 ADC0 预触发器源。选择 TPM 触发器源时不使用此字段。 0 预触发器 A 1 预触发器 B

下一页继续介绍此表...

SIM_SOPT7 字段描述 (继续)

字段	描述
ADC0TRGSEL	<p>ADC0 触发器选择</p> <p>当备用触发器在停止和 VLPS 模式下工作时，选择 ADC0 触发器源。</p> <p>0000 PDB 外部触发器引脚输入 (PDB0_EXTRG)</p> <p>0001 高速比较器 0 输出</p> <p>0010 保留</p> <p>0011 保留</p> <p>0100 PIT 触发器 0</p> <p>0101 PIT 触发器 1</p> <p>0110 PIT 触发器 2</p> <p>0111 PIT 触发器 3</p> <p>1000 TPM0 溢出</p> <p>1001 TPM1 溢出</p> <p>1010 TPM2 溢出</p> <p>1011 保留</p> <p>1100 RTC 警报</p> <p>1101 RTC 秒数</p> <p>1110 低功耗定时器 (LPTMR) 触发器</p> <p>1111 TPM1 通道 0 (预触发器 A) 和通道 (预触发器 B)</p>

13.2.5 系统选项寄存器 9 (SIM_SOPT9)

地址: 4004_7000h 基准 + 1020h 偏移 = 4004_8020h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R	0					TPM2CLKSEL	TPM1CLKSEL	TPM0CLKSEL	0			TPM2CH0SRC		TPM1CH0SRC		0	
W	保留					TPM2CLKSEL	TPM1CLKSEL	TPM0CLKSEL	保留			TPM2CH0SRC		TPM1CH0SRC		保留	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0																
W	保留																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

SIM_SOPT9 字段描述

字段	描述
31-27 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

SIM_SOPT9 字段描述 (继续)

字段	描述
26 TPM2CLKSEL	<p>TPM2 外部时钟引脚选择</p> <p>选择用于将时钟驱动至 TPM2 模块的外部引脚。</p> <p>注：必须通过端口控制模块中对应的引脚控制寄存器为 TPM 外部时钟功能配置选定的引脚。</p> <p>0 TPM_CLKIN0 引脚 1 TPM_CLKIN1 引脚</p>
25 TPM1CLKSEL	<p>TPM1 外部时钟引脚选择</p> <p>选择用于将时钟驱动至 TPM1 模块的外部引脚。</p> <p>注：必须通过端口控制模块中对应的引脚控制寄存器为 TPM 外部时钟功能配置选定的引脚。</p> <p>0 TPM_CLKIN0 引脚 1 TPM_CLKIN1 引脚</p>
24 TPM0CLKSEL	<p>TPM0 外部时钟引脚选择</p> <p>选择用于将时钟驱动至 TPM0 模块的外部引脚。</p> <p>注：必须通过端口控制模块中对应的引脚控制寄存器为 TPM 外部时钟功能配置选定的引脚。</p> <p>0 TPM_CLKIN0 引脚 1 TPM_CLKIN1 引脚</p>
23–22 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
21–20 TPM2CH0SRC	<p>TPM2 通道 0 输入捕捉源选择</p> <p>选择 TPM2 通道 0 输入捕捉的源。</p> <p>注：如果 TPM 未处于输入捕捉模式，则将此字段清零。</p> <p>00 TPM2_CH0 信号 01 CMP0 输出 10 保留 11 保留</p>
19–18 TPM1CH0SRC	<p>TPM1 通道 0 输入捕捉源选择</p> <p>选择 TPM1 通道 0 输入捕捉的源。</p> <p>注：如果 TPM 未处于输入捕捉模式，则将此字段清零。</p> <p>00 TPM1_CH0 信号 01 CMP0 输出 10 保留 11 USB 帧脉冲起始</p>
保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>

13.2.6 系统能够设备标识寄存器 (SIM_SDID)

地址: 4004_7000h 基准 + 1024h 偏移 = 4004_8024h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FAMILYID				SUBFAMID				SERIESID				0				REVID				DIEID				FAMID				PINID			
W	0																															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	0*	0*	0*	0*	x*	x*	x*	x*	1*	0*	1*	1*	1*	x*	x*	x*	x*	x*	x*	x*

* 注:

- 重置值在系统从 Flash IFR 复位时加载。
- x = 复位时未定义。

SIM_SDID 字段描述

字段	描述
31–28 FAMILYID	Kinetis 系列 ID 指定器件的 Kinetis 系列。 0000 KS0x 系列 0001 KS1x 系列 0010 KS2x 系列 0011 KS3x 系列 0100 KS4x 系列 0101 KS5x 系列 0110 KS6x 系列 0111 KS7x 系列 1000 KS8x 系列 1001 KS9x 系列
27–24 SUBFAMID	Kinetis 子系列 ID 指定器件的 Kinetis 子系列。 0000 KSx0 子系列 0010 KSx2 子系列 0100 KSx4 子系列 0110 KSx6 子系列 0111 KSx7 子系列
23–20 SERIESID	Kinetis 系列 ID 指定器件的 Kinetis 系列。 0000 Kinetis K 系列 0001 Kinetis L 系列 0101 Kinetis W 系列 0110 Kinetis V 系列 0111 Kinetis KS 系列
19–16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

SIM_SDID 字段描述 (继续)

字段	描述
15-12 REVID	器件版本号 指定器件的硅实施编号。
11-7 DIEID	器件芯片 ID 指定器件的硅特征集识别编号。
6-4 FAMID	Kinetis 系列标识 仅为兼容性保持此字段，但在此寄存器中已被 SERIESID、FAMILYID 和 SUBFAMID 字段替代。 000 KS0x 或 KS1x 001 KS2x 010 KS3x 011 KS4x 100 KS5x 101 KS6x 110 KS7x 111 KS8x 或 KS9x
PINID	引脚数标识 指定器件的引脚数。 0000 保留 0001 保留 0010 32 引脚 0011 保留 0100 48 引脚 0101 64 引脚 0110 80 引脚 0111 81 引脚或 121 引脚 1000 100 引脚 1001 121 引脚 1010 144 引脚 1011 定制引脚分配 (WLCSP) 1100 169 引脚 1101 保留 1110 256 引脚 1111 保留

13.2.7 系统时钟门控寄存器 4 (SIM_SCGC4)

地址: 4004_7000h 基准 + 1034h 偏移 = 4004_8034h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1	1	1	1	0				0					USBOTG	0	
W	[保留]													CMP		[保留]
复位	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	UART2	UART1	UART0	0		LPI2C1	LPI2C0	1	1	0	0	EWM	0
W	[保留]															
复位	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

SIM_SCGC4 字段描述

字段	描述
31 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
30 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
29 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
28 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
27-21 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
20 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
19 CMP	比较器时钟门控 此位控制比较器模块的时钟门控。 0 时钟禁用 1 时钟使能
18 USBOTG	USB 时钟门控 此位控制 USB 模块的时钟门控。 0 时钟禁用 1 时钟使能
17-14 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
13 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

SIM_SCGC4 字段描述 (继续)

字段	描述
12 UART2	UART2 时钟选通控制 此位控制 UART2 模块的时钟门控。 0 时钟禁用 1 时钟使能
11 UART1	UART1 时钟选通控制 此位控制 UART1 模块的时钟门控。 0 时钟禁用 1 时钟使能
10 UART0	UART0 时钟选通控制 此位控制 UART0 模块的时钟门控。 0 时钟禁用 1 时钟使能
9-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 LPI2C1	LPI2C1 时钟门控 此位控制 LPI2C1 模块的时钟门控。 0 时钟禁用 1 时钟使能
6 LPI2C0	LPI2C0 时钟门控 此位控制 LPI2C0 模块的时钟门控。 0 时钟禁用 1 时钟使能
5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 EWM	EWM 时钟门控 此位控制 EWM 模块的时钟门控。 0 时钟禁用 1 时钟使能
0 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

13.2.8 系统时钟门控寄存器 5 (SIM_SCGC5)

地址: 4004_7000h 基准 + 1038h 偏移 = 4004_8038h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
R	0														1	0								
W	[Reserved]																							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0								
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
R	0	PORTE			PORTD			PORTC			PORTB			PORTA			1	0	0	0	0	0	1	LPTMR
W	[Reserved]																							
复位	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0							

SIM_SCGC5 字段描述

字段	描述
31 FLEXIO	FlexIO 时钟门控 此位控制 FlexIO 模块的时钟门控。 0 时钟禁用 1 时钟使能
30-19 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
18 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
17-14 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
13 PORTE	Port E 时钟门控 此位控制端口 E 模块的时钟门控。 0 时钟禁用 1 时钟使能
12 PORTD	Port D 时钟门控 此位控制端口 D 模块的时钟门控。 0 时钟禁用 1 时钟使能
11 PORTC	Port C 时钟门控 此位控制端口 C 模块的时钟门控。

下一页继续介绍此表...

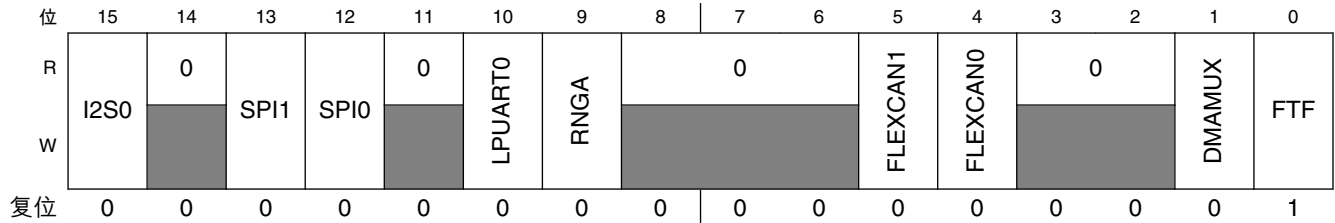
SIM_SCGC5 字段描述 (继续)

字段	描述
	0 时钟禁用 1 时钟使能
10 PORTB	Port B 时钟门控 此位控制端口 B 模块的时钟门控。 0 时钟禁用 1 时钟使能
9 PORTA	Port A 时钟门控 此位控制端口 A 模块的时钟门控。 0 时钟禁用 1 时钟使能
8-7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3-2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
0 LPTMR	低功耗定时器访问控制 此位控制低功耗定时器模块的软件访问。 0 访问禁用 1 访问使能

13.2.9 系统时钟门控寄存器 6 (SIM_SCGC6)

地址: 4004_7000h 基准 + 103Ch 偏移 = 4004_803Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R		1		0							0	0			0	
W	DAC0		RTC		ADC0	TPM2	TPM1	TPM0	PIT	PDB				CRC		I2S1
复位	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0



SIM_SCGC6 字段描述

字段	描述
31 DAC0	DAC0 时钟门控 此位控制 DAC0 模块的时钟门控。 0 时钟禁用 1 时钟使能
30 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
29 RTC	RTC 访问控制 此位控制 RTC 模块的软件访问和中断。 0 访问和中断禁用 1 访问和中断使能
28 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
27 ADC0	ADC0 时钟门控 此位控制 ADC0 模块的时钟门控。 0 时钟禁用 1 时钟使能
26 TPM2	TPM2 时钟门控 此位控制 TPM2 模块的时钟门控。 0 时钟禁用 1 时钟使能
25 TPM1	TPM1 时钟门控 此位控制 TPM1 模块的时钟门控。 0 时钟禁用 1 时钟使能
24 TPM0	TPM0 时钟门控 此位控制 TPM0 模块的时钟门控。 0 时钟禁用 1 时钟使能
23 PIT	PIT 时钟选通控制 此位控制 PIT 模块的时钟门控。

下一页继续介绍此表...

SIM_SCGC6 字段描述 (继续)

字段	描述
	0 时钟禁用 1 时钟使能
22 PDB	PDB 时钟门控 此位控制 PDB 模块的时钟门控。 0 时钟禁用 1 时钟使能
21 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
20-19 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
18 CRC	CRC 时钟选通控制 此位控制 CRC 模块的时钟门控。 0 时钟禁用 1 时钟使能
17 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
16 I2S1	I2S1 时钟门控 此位控制 I2S1 模块的时钟门控。 0 时钟禁用 1 时钟使能
15 I2S0	I2S0 时钟门控 此位控制 I2S0 模块的时钟门控。 0 时钟禁用 1 时钟使能
14 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
13 SPI1	SPI1 时钟选通控制 此位控制 SPI1 模块的时钟门控。 0 时钟禁用 1 时钟使能
12 SPI0	SPI0 时钟选通控制 此位控制 SPI0 模块的时钟门控。 0 时钟禁用 1 时钟使能
11 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
10 LPUART0	LPUART0 时钟门控

下一页继续介绍此表...

SIM_SCGC6 字段描述 (继续)

字段	描述
	此位控制 LPUART0 模块的时钟门控。 0 时钟禁用 1 时钟使能
9 RNGA	RNGA 时钟门控 控制 RNGA 模块的时钟门控。
8-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 FLEXCAN1	FlexCAN1 时钟门控 此位控制 FlexCAN1 模块的时钟门控。 注: 此位仅适用于 KS22。KS20 没有 CAN1 模块。 0 时钟禁用 1 时钟使能
4 FLEXCAN0	FlexCAN0 时钟门控 此位控制 FlexCAN0 模块的时钟门控。 0 时钟禁用 1 时钟使能
3-2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 DMAMUX	DMA 复用器时钟门控 此位控制 DMA 复用器模块的时钟门控。 0 时钟禁用 1 时钟使能
0 FTF	Flash 存储器时钟门控 此位控制 Flash 存储器模块的时钟门控。当时钟门控 Flash 存储器时, 仍支持 Flash 读取, 但进入低功耗模式将受阻。 0 时钟禁用 1 时钟使能

13.2.10 系统时钟门控寄存器 7 (SIM_SCGC7)

地址: 4004_7000h 基准 + 1040h 偏移 = 4004_8040h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											0	0	DMA	0	
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

SIM_SCGC7 字段描述

字段	描述
31-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 DMA	DMA 时钟门控 此位控制 DMA 模块的时钟门控。 0 时钟禁用 1 时钟使能
0 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

13.2.11 系统时钟分频器寄存器 1 (SIM_CLKDIV1)

更新 CLKDIV1 时，使用一个写入命令更新全部字段。

注

当器件处于 VLPR 模式时，无法写入 CLKDIV1 寄存器。

地址: 4004_7000h 基准 + 1044h 偏移 = 4004_8044h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OUTDIV1				OUTDIV2				0				OUTDIV4				0															
W																																
复位	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	1*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

*注:

- 重置值在系统从 FTF_FOFT[LPBOOT] 复位时加载。

SIM_CLKDIV1 字段描述

字段	描述
31–28 OUTDIV1	<p>时钟 1 输出分频器值</p> <p>该字段设置来自 MCGOUTCLK 的内核/系统时钟的分频值。复位结束后，将向其加载 0000 或 0111，具体取决于 FTF_FOFT[LPBOOT]。</p> <p>0000 1 分频。 0001 2 分频。 0010 3 分频。 0011 4 分频。 0100 5 分频。 0101 6 分频。 0110 7 分频。 0111 8 分频。 1000 9 分频。 1001 10 分频。 1010 11 分频。 1011 12 分频。 1100 13 分频。 1101 14 分频。 1110 15 分频。 1111 16 分频。</p>
27–24 OUTDIV2	<p>时钟 2 输出分频器值</p> <p>该字段设置来自 MCGOUTCLK 的总线时钟的分频值。复位结束后，将向其加载 0000 或 0111，具体取决于 FTF_FOFT[LPBOOT]。总线时钟频率必须是内核/系统时钟的整数分频。</p> <p>0000 1 分频。 0001 2 分频。 0010 3 分频。 0011 4 分频。 0100 5 分频。 0101 6 分频。 0110 7 分频。 0111 8 分频。 1000 9 分频。 1001 10 分频。 1010 11 分频。 1011 12 分频。 1100 13 分频。 1101 14 分频。 1110 15 分频。 1111 16 分频。</p>
23–20 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
19–16 OUTDIV4	<p>时钟 4 输出分频器值</p> <p>该字段设置来自 MCGOUTCLK 的 Flash 时钟的分频值。复位结束后，将向其加载 0001 或 1111，具体取决于 FTF_FOFT[LPBOOT]。Flash 时钟频率必须是系统时钟的整数分频。</p>

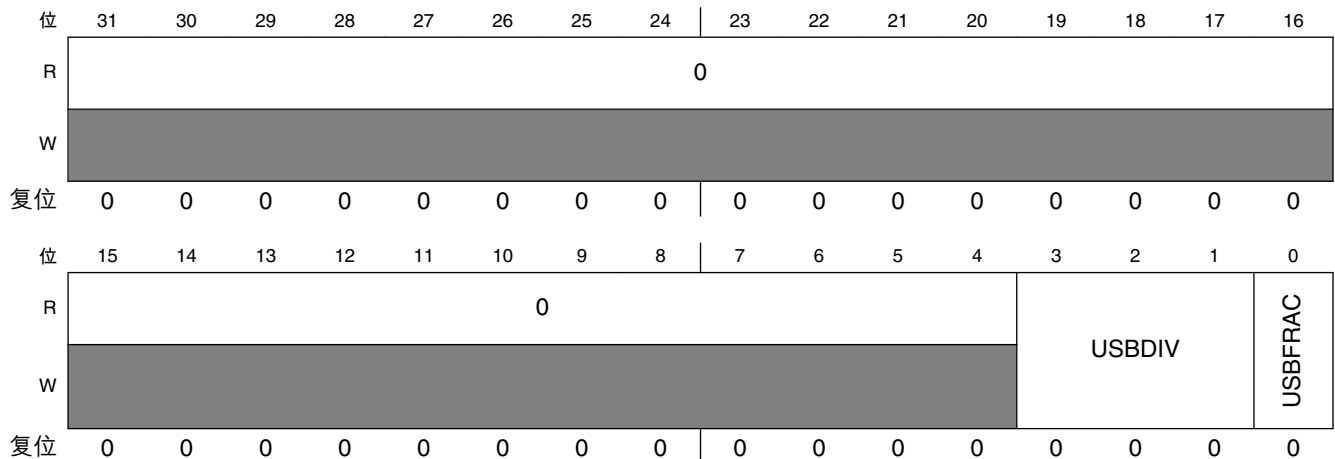
下一页继续介绍此表...

SIM_CLKDIV1 字段描述 (继续)

字段	描述
0000	1 分频。
0001	2 分频。
0010	3 分频。
0011	4 分频。
0100	5 分频。
0101	6 分频。
0110	7 分频。
0111	8 分频。
1000	9 分频。
1001	10 分频。
1010	11 分频。
1011	12 分频。
1100	13 分频。
1101	14 分频。
1110	15 分频。
1111	16 分频。
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

13.2.12 系统时钟分频器寄存器 2 (SIM_CLKDIV2)

地址: 4004_7000h 基准 + 1048h 偏移 = 4004_8048h



SIM_CLKDIV2 字段描述

字段	描述
31-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3-1 USBDIV	USB 时钟分频器除数 此字段设置当 MCGFLLCLK、MCGPLLCLK、或 IRC48M 时钟用作 USB 时钟源 (SOPT2[USBSRC] = 1) 时小数时钟分频器的分频值。

下一页继续介绍此表...

SIM_CLKDIV2 字段描述 (继续)

字段	描述
	分频器输出时钟 = 分频器输入时钟 × [(USBFRAC+1) / (USBDIV+1)]
0 USBFRAC	USB 时钟分频器小数 此字段设置当 MCGFLLCLK、MCGPLLCLK、或 IRC48M 时钟用作 USB 时钟源 (SOPT2[USBSRC] = 1) 时小数时钟分频器的小数乘法值。

13.2.13 Flash 配置寄存器 1 (SIM_FCFG1)

地址: 4004_7000h 基准 + 104Ch 偏移 = 4004_804Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R		1			PFSIZE					0					1	
W																
复位	1*	1*	1*	1*	1*	1*	1*	1*	0*	0*	0*	0*	1*	1*	1*	1*
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R		0				1						0				
W																
复位	0*	0*	0*	0*	1*	1*	1*	1*	0*	0*	0*	0*	0*	0*	0*	0*
															FLASHDOZE	FLASHDIS

* 注:

- 重置值在系统从 Flash IFR 复位时加载。

SIM_FCFG1 字段描述

字段	描述
31-28 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。

下一页继续介绍此表...

SIM_FCFG1 字段描述 (继续)

字段	描述
27-24 PFSIZE	<p>程序 flash 大小</p> <p>该字段用于指定器件上提供的程序 Flash 存储器大小。保留未定义值。</p> <p>0011 32 KB 的程序 Flash 存储器 0101 64 KB 的程序 Flash 存储器 0111 128 KB 的程序 Flash 存储器 1001 256 KB 的程序 Flash 存储器 1011 512 KB 的程序 Flash 存储器 1101 1024 KB 的程序 Flash 存储器 1111 256 KB 程序 flash 存储器。1111 表示此 SoC 的最大可用 flash 大小。</p>
23-20 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
19-16 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 1。</p>
15-12 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
11-8 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 1。</p>
7-2 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
1 FLASHDOZE	<p>Flash 休眠</p> <p>置位后，Flash 存储器在等待模式期间禁用。Flash 禁用时，DMA 或其他总线主机至 Flash 的任何访问尝试将导致总线错误。在 VLP 模式下不得将此位清零。Flash 将在等待模式结束时再次自动使能，因此，无需重定位 Flash 存储器中的中断向量。当此位置位时，自等待模式起的唤醒时间将延长。</p> <p>0 Flash 在等待模式期间保持使能 1 Flash 在等待模式期间保持禁用</p>
0 FLASHDIS	<p>Flash 禁用</p> <p>Flash 访问禁用（并生成总线错误）且 Flash 存储器处于低功耗状态。在 VLP 模式下不得更改此位。禁用 Flash 之前，重定位 Flash 存储器的中断向量。</p> <p>0 Flash 使能 1 Flash 禁用</p>

13.2.14 Flash 配置寄存器 2 (SIM_FCFG2)

地址: 4004_7000h 基准 + 1050h 偏移 = 4004_8050h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAXADDR0							1	0						
W	[Reserved]															
复位	0*	0*	1*	0*	0*	0*	0*	0*	1*	0*	0*	0*	0*	0*	0*	0*
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	[Reserved]															
复位	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* 注:

- 重置值在系统从 Flash IFR 复位时部分加载。

SIM_FCFG2 字段描述

字段	描述
31 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
30-24 MAXADDR0	最大地址块 0 此字段与 13 个后导零级联，表示每个程序 Flash 块的第一个无效地址。 例如，如果 MAXADDR0 = 0x20，则第一个 Flash 块 0 的无效地址为 0x0004_0000。对于 Flash 块中带 256 KB 程序 Flash 的器件，这是 MAXADDR0 值。
23 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

13.2.15 高电平寄存器唯一标识 (SIM_UIDH)

地址: 4004_7000h 基准 + 1054h 偏移 = 4004_8054h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID																															
W	[Reserved]																															
复位	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	

* 注:

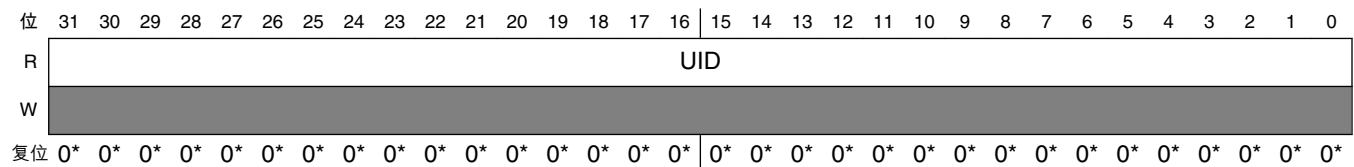
- 重置值在系统从 Flash IFR 复位时加载。

SIM_UIDH 字段描述

字段	描述
UID	唯一标识 器件的唯一标识。

13.2.16 中高电平寄存器唯一标识 (SIM_UIDMH)

地址: 4004_7000h 基准 + 1058h 偏移 = 4004_8058h



*注:

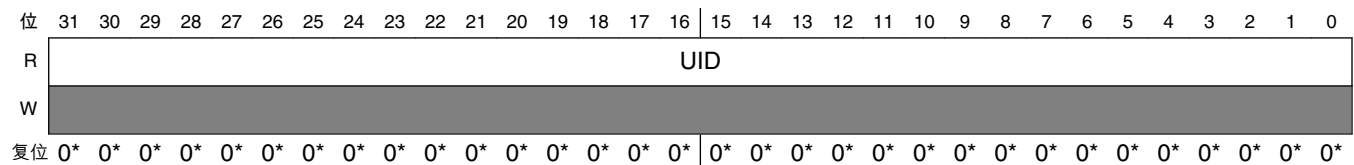
- 重置值在系统从 Flash IFR 复位时加载。

SIM_UIDMH 字段描述

字段	描述
UID	唯一标识 器件的唯一标识。

13.2.17 中低电平寄存器唯一标识 (SIM_UIDML)

地址: 4004_7000h 基准 + 105Ch 偏移 = 4004_805Ch



*注:

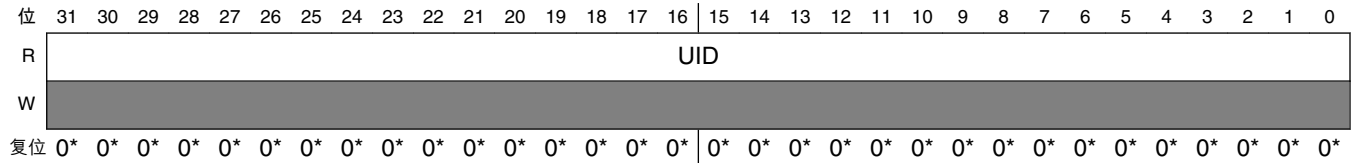
- 重置值在系统从 Flash IFR 复位时加载。

SIM_UIDML 字段描述

字段	描述
UID	唯一标识 器件的唯一标识。

13.2.18 低电平寄存器唯一标识 (SIM_UIDL)

地址: 4004_7000h 基准 + 1060h 偏移 = 4004_8060h



*注:

- 重置值在系统从 Flash IFR 复位时加载。

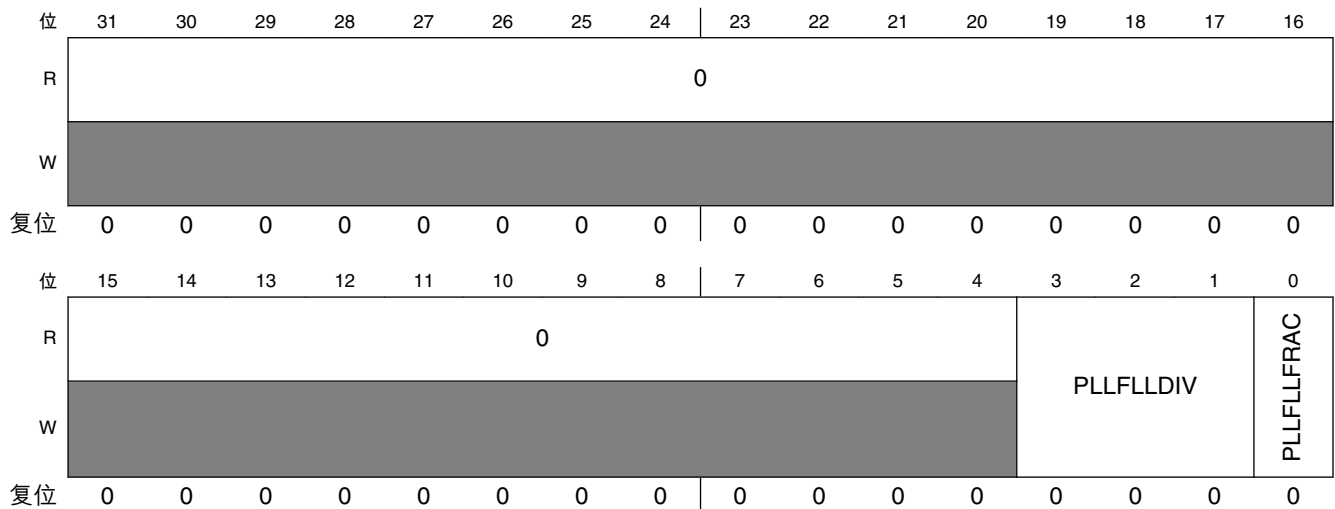
SIM_UIDL 字段描述

字段	描述
UID	唯一标识 器件的唯一标识。

13.2.19 系统时钟分频器寄存器 3 (SIM_CLKDIV3)

此寄存器应仅在 LPUART、LPI2C 和 TPM 模块禁用时写入。

地址: 4004_7000h 基准 + 1064h 偏移 = 4004_8064h



SIM_CLKDIV3 字段描述

字段	描述
31-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3-1 PLLFLLDIV	PLLFLL 时钟分频器除数 此字段设置用作各种外设时钟源的小数时钟分频器的分频值。小数时钟分频器的源时钟由 SOPT2 PLLFLLSEL 寄存器位设置。分频器输出时钟 = 分频器输入时钟 $\times ((\text{PLLFLLFRAC}+1)/(\text{PLLFLLDIV}+1))$
0 PLLFLLFRAC	PLLFLL 时钟分频器小数 此字段设置用作各种外设源的小数时钟分频器的分频值。小数时钟分频器的源时钟由 SOPT2 PLLFLLSEL 寄存器位设置。分频器输出时钟 = 分频器输入时钟 $\times ((\text{PLLFLLFRAC}+1)/(\text{PLLFLLDIV}+1))$

13.2.20 其他控制寄存器 (SIM_MISCCTL)

地址: 4004_7000h 基准 + 106Ch 偏移 = 4004_806Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	[Reserved]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														FlexIOS0	UARTSELONU SB
W	[Reserved]														FlexIOS0	UARTSELONU SB
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

SIM_MISCCTL 字段描述

字段	描述
31-3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 FlexIOS0	FlexIO 时钟槽 0 选择 0 系统时钟 1 I2S0_MCLK
UARTSELONUSB	USB DP/DM 引脚上的 UART 选择。详情请参见 USB 章节中的“USB 上的 UART 性能”一节。 00 UART0 01 UART1

下一页继续介绍此表...

SIM_MISCCTL 字段描述 (继续)

字段	描述
10	UART2
11	LPUART (默认)

13.3 功能说明

有关 SIM 功能的更多信息，请参见[简介](#) 一节。

第 14 章

Kinetis Flashloader

14.1 此模块的芯片实现细节

14.1.1 Kinetis Flashloader

- Kinetis Flashloader 已在芯片制造过程中预先写入 Kinetis Flash，其支持 Flash 编程而无需调试器。对于 KS 系列微控制器，Kinetis Flashloader 支持 UART、LPI2C、SPI、USB-HID 和 CAN 外设接口与主机通信，从而向 Flash 上烧写用户应用程序镜像。下表显示了 Flashloader 对于支持的外设所使用的引脚配置。

表 14-1. Kinetis Flashloader 引脚分配

外设接口	分配的引脚	模块实例	ALTMUX 栏
UART	PTE0,UART1_TX	1	3
	PTE1,UART1_RX		
LPI2C	PTB0,LPI2C0_SCL	0	2
	PTB1,LPI2C0_SDA		
SPI	PTD4,SPI1_PCS0	1	7
	PTD5,SPI1_SCK		
	PTD6,SPI1_SOUT		
	PTD7,SPI1_SIN		
CAN	PTB18,CAN0_TX	0	2
	PTB19,CAN0_RX		
USB	USB0_DP	0	-
	USB0_DM		
	USBVDD		

注

对于该器件，在本章的后续部分，“I2C”表示“LPI2C”。

14.2 简介

不具备片上 ROM 的 Kinetis 器件在交货时，在出厂前通过在线系统编程方式一次性在片上 Flash 存储器上预烧录 Kinetis Flashloader 程序。Kinetis Flashloader 的主要目的在于将客户的固件镜像加载到 Flash 存储器上。Flash 上的镜像有两种程序：flashloader_loader 和 Flashloader。器件复位之后，flashloader_loader 程序先执行。flashloader_loader 程序将来自 Flash 的 Flashloader 图像的内容复制到片上 RAM；之后器件切换为 Flashloader 程式，从 RAM 执行。

对于此设备，Kinetis Flashloader 可与处于从机模式的 USB、UART、CAN、I2C 和 SPI 外设连接，并响应通过其中一个端口通信的主设备（或主机）发出的命令。主机/主设备可能是一个在计算机或嵌入式主机（与 Kinetis Flashloader 通信）上运行的固件下载应用程序。与主机/主设备（计算机或嵌入式主机）无关，该 Kinetis Flashloader 始终采用命令协议与该主机/主设备通信。命令用于写入存储器（Flash 或 RAM）、清除 Flash、获取/置位 Flashloader 选项和属性值。主机应用程序可以查询可用的命令集。

本章描述了 Kinetis Flashloader 特性、功能、命令结构以及支持哪些外设。

支持的特性：

- 支持 USB FS、UART、CAN、I2C 以及 SPI 外设接口。
- 自动检测活动外设
- 带自动波特率的 UART 与 CAN 外设
- 面向所有外设，基于数据包的通用协议
- 数据包错误检测和重传
- Flashloader 在运行期间所用的 RAM 保护
- 提供读取设备属性的命令，例如 Flash 和 RAM 大小

表 14-2. Kinetis Flashloader 支持的命令

命令	说明	当 Flash 安全被禁用时, 此命令被
Call	运行用户应用程序代码并将控制返还给引导加载程序	不支持
Execute	运行用户应用程序代码，不再返回 Flashloader	不支持
FillMemory	按照字模式将一系列字节填充到 Flash 中	不支持
FlashEraseAll	擦除整个 Flash 阵列	不支持
FlashEraseRegion	擦除 Flash 中的一系列扇区	不支持
FlashProgramOnce	写入数据（命令数据包中提供）到一次性编程字段中的指定字节范围	不支持
FlashReadOnce	通过给定的索引和字节数返回一次性编程字段的内容	不支持
FlashReadResource	通过给定的偏移、字节数和选项，返回 IFR 字段或 Flash 固件 ID 的内容	不支持

下一页继续介绍此表...

表 14-2. Kinetis Flashloader 支持的命令 (继续)

命令	说明	当 Flash 安全被禁用时, 此命令被
WriteMemory	将数据写入存储器	不支持
ReadMemory	从存储器读取数据	不支持
GetProperty	获取属性的当前值	支持
ReceiveSbFile	接收 SB 文件 (由 elftosb 工具生成)	如果 SB 文件加密, 予以支持。参阅 SB 文件加密支持一节, 了解更多详情。
Reset	复位芯片	支持
SetProperty	尝试更改可写属性	支持

14.3 功能说明

以下子章节描述 Kinetis Flashloader 功能。

14.3.1 存储器映射

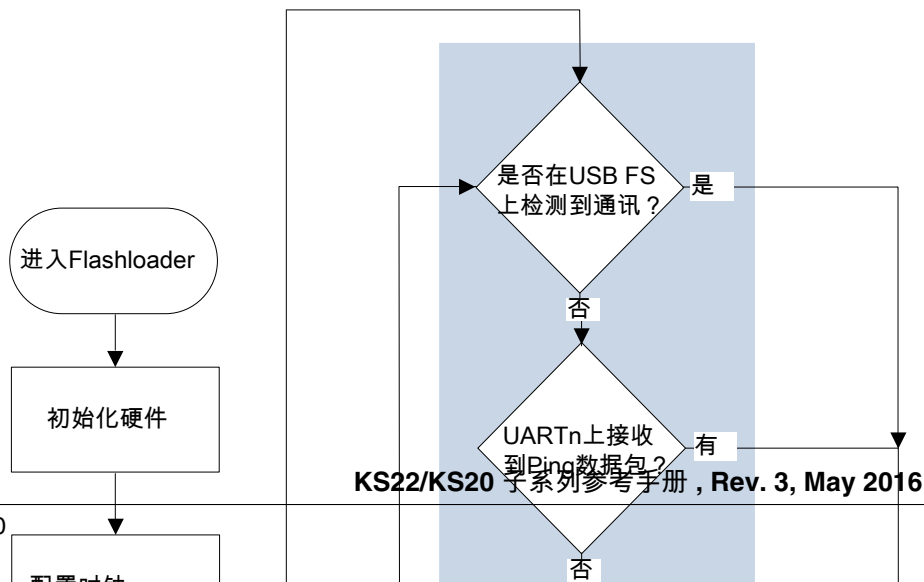
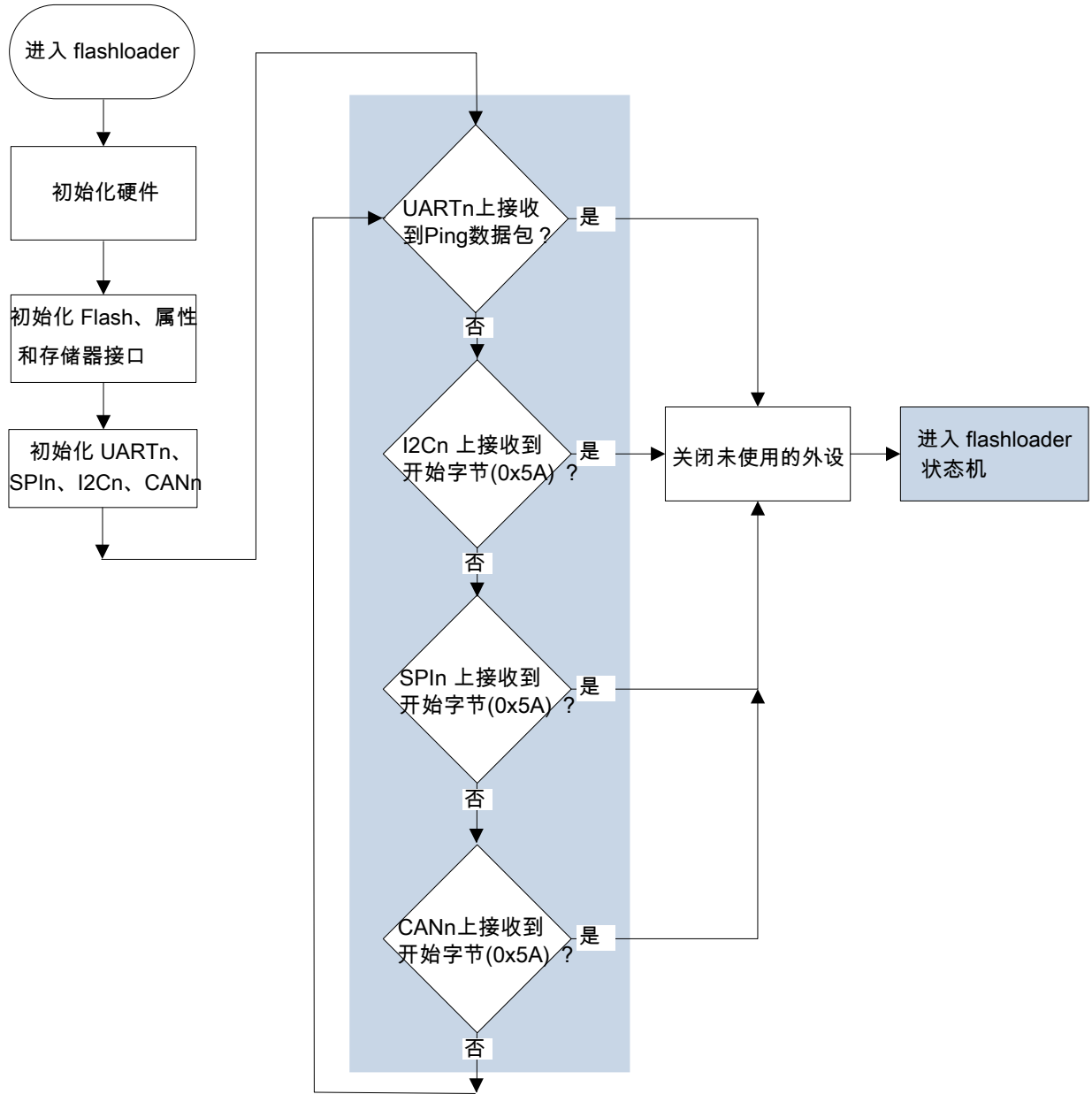
执行时, Kinetis Flashloader 使用随机存储器 (RAM)。

Kinetis Flashloader 至少需要 32 KB 的 RAM 内存空间。对于内存量小于此片上 RAM 的 Kinetis 器件, Kinetis Flashloader 不可用。

14.3.2 启动流程

Kinetis Flashloader 开始执行操作之后, flashloader 操作随之开始:

1. flashloader 在 RAM 中的临时工作区域被初始化。
2. 所有受支持的外设均被初始化。
3. flashloader 等待外设开始通信。
 - 有效的外设检测流程不存在超时问题。
 - 检测到通信之后, 所有无效外设均会关闭, 并进入命令接受状态。



14.3.3 时钟配置

Kinetis Flashloader 将使能内部 48 MHz 参考时钟，在 Flashloader 退出时使用芯片的复位时钟配置。

14.3.4 Flashloader 协议

本节用于说明主机和 Kinetis Flashloader 之间的数据包传输通用协议。包括对不同事务数据包传输的描述，例如不带数据阶段的命令和带传入或传出数据阶段的命令。下一节用于说明事务中所用的各种数据包类型。

主机发出的每个命令都会由响应命令予以应答。

命令可能包含一个可选的数据阶段：

- 如果数据阶段正在传入（从主机至 Flashloader），那么数据阶段是原始命令的组成部分。
- 如果数据阶段正在传出（从 Flashloader 至主机），那么数据阶段是响应命令的组成部分。

注

在所有协议（在下一小节中描述）中，为了响应命令或数据包而发出的 Ack 可在处理命令/数据包之前、期间或之后送达。

14.3.4.1 无数据阶段的命令

无数据阶段的命令协议包含：

- 命令数据包（从主机）
- 通用响应命令数据包（至主机）

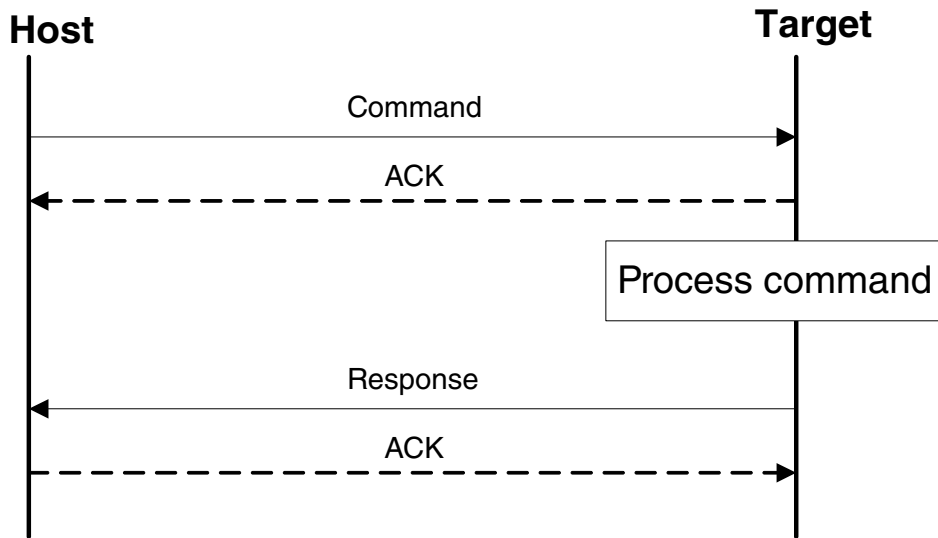


图 14-2. 无数据阶段的命令

14.3.4.2 带输入数据阶段的命令

带输入数据阶段的命令协议包含：

- 命令数据包（从主机）
- 通用响应命令数据包（至主机）
- 输入数据包（从主机）
- 通用响应命令数据包（至主机）

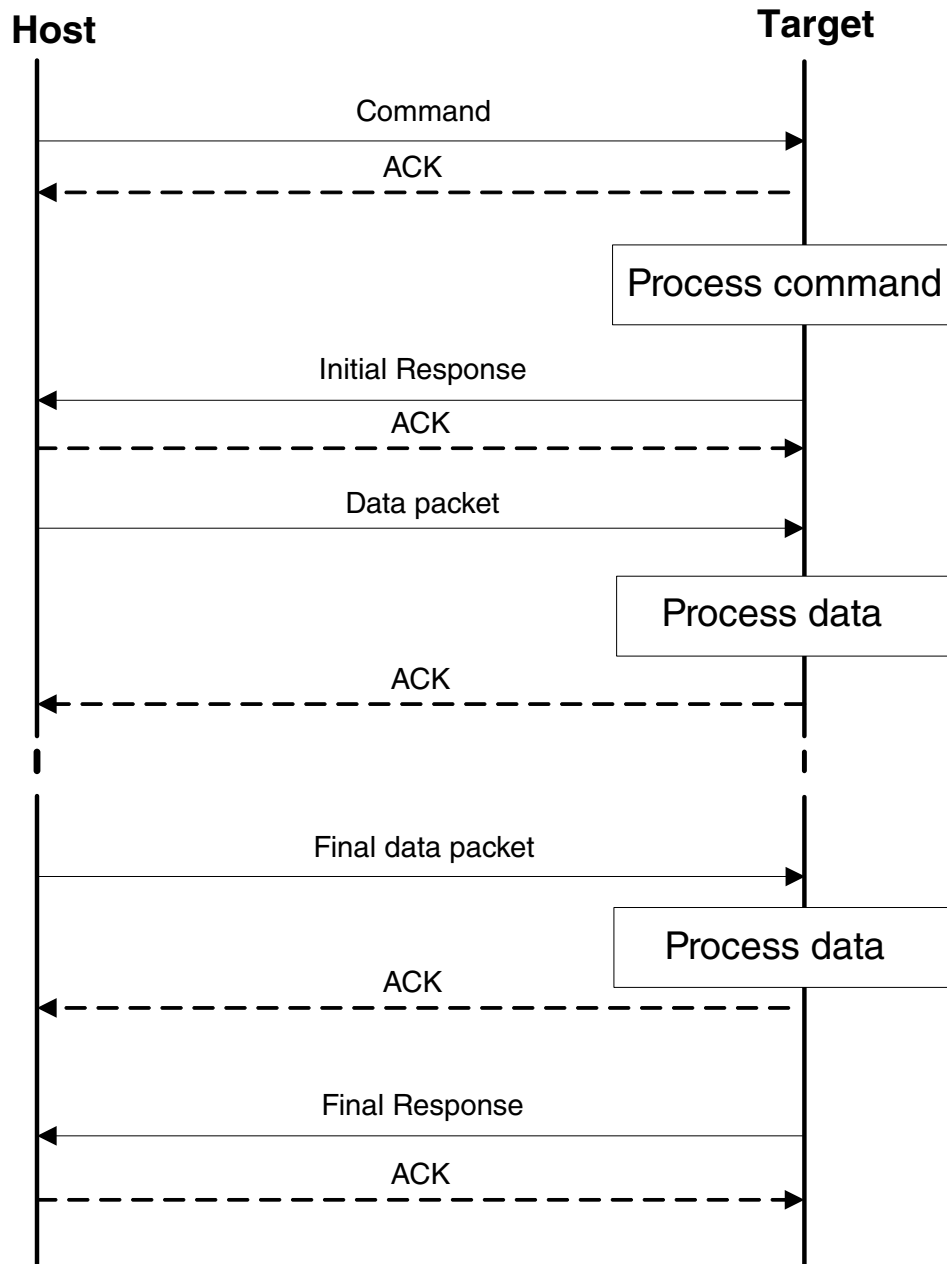


图 14-3. 带输入数据阶段的命令

注

- 主机等待对命令的响应时，可能不会继续发送数据包。
- 如果数据阶段起始之前的通用响应数据包不含 `kStatus_Success` 状态，那么数据阶段会终止。
- 接收方可能通过发送 `kStatus_AbortDataPhase` 状态的最终通用响应而提前终止数据阶段。主机可能通过发送长度为零的数据包而提前终止数据阶段。
- 在数据阶段之后发送的最终通用响应数据包包含整个操作的状态信息。

14.3.4.3 带传出数据阶段的命令

带传出数据阶段的命令协议包含：

- 命令数据包（从主机）
- ReadMemory 响应命令数据包（至主机）(kCommandFlag_HasDataPhase set)
- 传出数据包（至主机）
- 通用响应命令数据包（至主机）

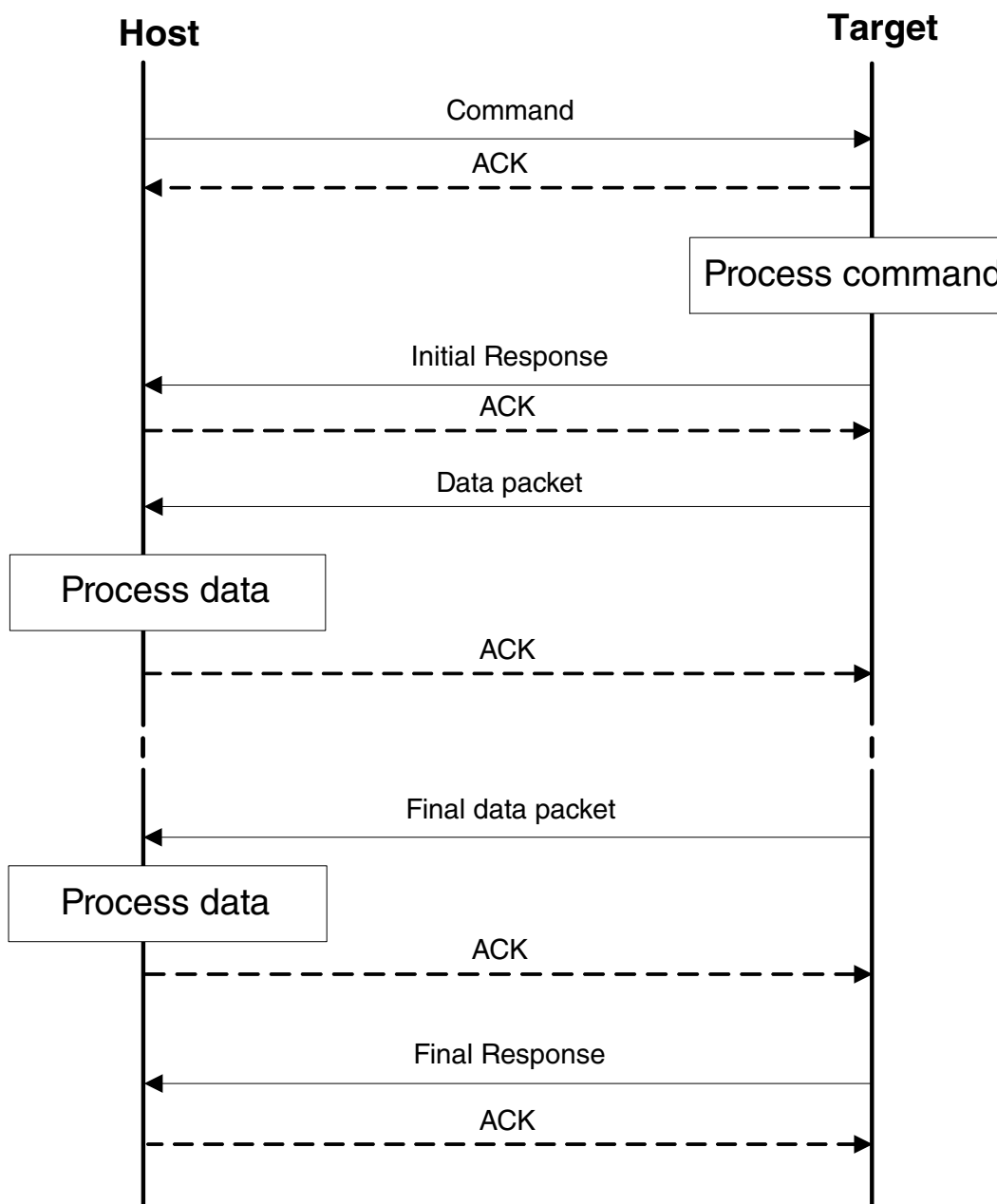


图 14-4. 带传出数据阶段的命令

注

- 对于上述的传出数据阶段序列，数据阶段确实被视为响应命令的组成部分。
- 主机等待对命令的响应时，可能不会继续发送数据包。
- 如果 ReadMemory 在数据阶段（未包含 kCommandFlag_HasDataPhase 标志）开始之前响应命令数据包，则数据阶段会终止。
- 数据阶段可能因主机过早发送 kStatus_AbortDataPhase 状态的最终通用响应而终止。发送方可能因发送长度为零的数据包而过早终止数据阶段。
- 在数据阶段之后发送的最终通用响应数据包包含整个操作的状态信息。

14.3.5 Flashloader 包类型

Kinetis Flashloader 器件以从机模式工作。所有数据通信都由主机发起，主机可以是个人计算机或嵌入式主机。Kinetis Flashloader 器件是目标器件，用于接收命令或数据包。主机和目标器件之间的所有数据通信均以包的形式进行。

注

词汇“目标”指的是“Kinetis Flashloader 器件”。

本器件共使用 6 种类型的包：

- Ping 包
- Ping 响应包
- 帧包
- 命令包
- 数据包
- 响应包

包中的所有字段均采用低字节序。

14.3.5.1 Ping 包

Ping 包是从主机发送到目标的第一个数据包 (Kinetis Flashloader)，以此和选定外设建立连接。对于 UART 外设，Ping 包用于确定波特率。Ping 包必须于其他通信之前发送。对应于 Ping 包，目标回送 Ping 响应包。

表 14-3. Ping 包格式

字节 #	数值	名称
0	0x5A	start byte
1	0xA6	ping

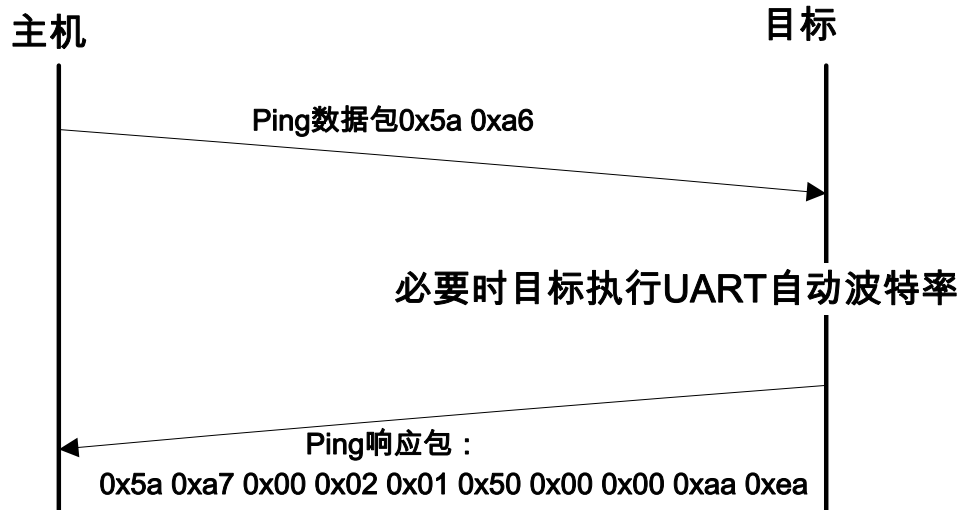


图 14-5. Ping 包协议序列

14.3.5.2 Ping 响应包

目标（Kinetic Flashloader）在接收一个 Ping 包之后会向主机发回一个 Ping 响应包。如果通过 UART 外设通信，则目标在采用 Ping 响应包进行答复之前会使用输入 Ping 包确定波特率。一旦主机接收 Ping 响应包，连接即建立，且主机开始向目标（Kinetic Flashloader）发送命令。

表 14-4. Ping 响应包格式

字节 #	数值	参数
0	0x5A	start byte
1	0xA7	Ping response code
2		Protocol bugfix
3		Protocol minor
4		Protocol major
5		Protocol name = 'P' (0x50)
6		Options low
7		Options high
8		CRC16 low
9		CRC16 high

14.3.5.3 帧包

帧包用于流量控制和错误检测，并且它（帧包）还可以包覆命令和数据包。

本节中介绍的帧包用于串行外设，包括 UART、I2C 和 SPI。USB HID 外设并不使用帧包。相反，它使用 USB 协议中固有的分包。更多详情，请参阅 [USB 外设](#)。

表 14-5. 帧包格式

字节 #	数值	参数	说明
0	0x5A	start byte	
1		packetType	
2		length_low	长度是 16 位字段，它用于指定整个命令或数据包大小（以字节为单位）。
3		length_high	
4		crc16_low	这是 16 位字段。CRC16 值包含整个帧包，其中包括开始字节和命令或数据包，但不包括 CRC 字节。请参见此表后面的 CRC16 算法。
5		crc16_high	
6...n		Command or Data packet payload	

一种特殊帧包，它仅包含一个开始字节和一种数据包类型，用于实现主机与目标之间的同步。

表 14-6. 特殊帧包格式

字节 #	数值	参数
0	0x5A	start byte
1	0xA n	packetType

“数据包类型”字段从其中一种已定义的类型（如下所示）中指定数据包类型：

表 14-7. packetType 字段

packetType	名称	说明
0xA1	kFramingPacketType_Ack	已成功接收上一个数据包；允许发送更多数据包。
0xA2	kFramingPacketType_Nak	上一个数据包已损坏，必须重新发送。
0xA3	kFramingPacketType_AckAbort	将中止数据阶段。
0xA4	kFramingPacketType_Command	帧包包含命令包有效载荷。
0xA5	kFramingPacketType_Data	帧包包含数据包有效载荷。
0xA6	kFramingPacketType_Ping	发送以验证另一端是否有效。也用于 UART 自动波特率。
0xA7	kFramingPacketType_PingResponse	响应 Ping；包含帧协议版本号和选项。

14.3.5.4 命令包

命令包具有一个 32 位命令标题和一个 32 位参数列表。

表 14-8. 命令包格式

命令包格式 (32 字节)										
命令标题 (4 字节)				28 字节参数 (最大 7 个参数)						
标签	标志	Rsvd	参数计数	参数 1 (32 位)	参数 2 (32 位)	参数 3 (32 位)	参数 4 (32 位)	参数 5 (32 位)	参数 6 (32 位)	参数 7 (32 位)
字节 0	字节 1	字节 2	字节 3							

表 14-9. 命令标题格式

字节 #	命令标题字段	
0	命令或响应标签	命令标题长度为 4 个字节，具有这些字段。
1	标志	
2	保留。应为 0x00。	
3	参数计数	

紧跟着标题的,是标题中制定的 ParameterCount 字段值大小的 32 位参数。由于命令包长度为 32 字节,因此,命令包中仅可包含 7 个参数。

命令包还可供目标用于将响应发送回主机。如先前所述,对于所有传输,命令包和数据包均嵌入到帧数据包中。

表 14-10. 受支持的命令

命令	名称
0x01	FlashEraseAll
0x02	FlashEraseRegion
0x03	ReadMemory
0x04	WriteMemory
0x05	FillMemory
0x06	Reserved
0x07	GetProperty
0x08	ReceiveSBFile
0x09	Execute
0x0A	Call
0x0B	Reset
0x0C	SetProperty
0x0D	Reserved

下一页继续介绍此表...

表 14-10. 受支持的命令 (继续)

命令	名称
0x0E	FlashProgramOnce
0x0F	FlashReadOnce
0x10	FlashReadResource
0x11	Reserved
0x12	ReliableUpdate

表 14-11. 受支持的响应

响应	名称
0xA0	GenericResponse
0xA3	ReadMemoryResponse (仅用于将响应发送至 ReadMemory 命令)
0xA7	GetPropertyResponse (仅用于将响应发送至 GetProperty 命令)
0xAF	FlashReadOnceResponse (仅用于将响应发送至 FlashReadOnce 命令)
0xB0	FlashReadResourceResponse (仅用于将响应发送至 FlashReadResource 命令)

Flags: 每个命令包包含一个标志字节。仅使用标志字节的位 0。如果标志字节的位 0 设为 1, 则数据包将遵循命令序列。在数据阶段传输的字节数由参数阵列中的特定命令参数决定。

ParameterCount: 命令包中包含的参数数目。

Parameters: 参数为字长 (32 位)。如果默认的最大数据包大小为 32 字节, 则命令包可包含最多 7 个参数。

14.3.5.5 数据包

无论是从主机发送数据到目标, 还是从目标发送数据到主机, 数据包均只传输数据。数据传输方向由主机发送的最后一个命令决定。数据包封装在帧包里, 以确保接收到正确的包数据。

数据包的内容只是数据本身。没有其它的字段, 因此每个数据包中的数据均可传输。帧包负责确保接收到正确的包数据。

14.3.5.6 响应包

响应使用的是相同的命令数据格式, 包覆帧数据包数据。响应类型包括:

- GenericResponse

- GetPropertyResponse
- ReadMemoryResponse
- FlashReadOnceResponse
- FlashReadResourceResponse

GenericResponse: Kinetis Flashloader 处理完命令之后, flashloader 会将通用响应以及状态和命令标签信息发送至主机。通用响应是命令协议序列中的最后一个数据包。通用响应包包含帧数据包数据和命令包数据 (通用响应标签 = 0xA0) 和参数列表 (在下一节介绍)。对于状态代码和命令标签参数, 将标题中的参数计数字段始终设为 2。

表 14-12. GenericResponse 参数

字节 #	参数	描述
0 .. 3	Status code	Status code 是指目标 (Kinetis Flashloader) 在执行命令过程中遇到的错误。如果命令成功, 则会返回 kStatus_Success 代码。表 14-49、Kinetis Flashloader 状态错误代码, 列出了 Kinetis Flashloader 返回至主机的状态代码。
4 - 7	Command tag	Command tag 标识至主机发送的命令的响应。

GetPropertyResponse: GetPropertyResponse 数据包由目标发送, 以响应使用 GetProperty 命令的主机查询。GetPropertyResponse 数据包包含帧包数据和命令包数据, 且命令/响应标签置位为 GetPropertyResponse 标签值 (0xA7)。

标题中的参数计数字段置位为大于 1, 以始终包含状态码以及一个或多个属性值。

表 14-13. GetPropertyResponse 参数

字节 #	数值	参数
0 .. 3		Status code
4 - 7		Property value
...		...
		Can be up to maximum 6 property values, limited to the size of the 32-bit command packet and property type.

ReadMemoryResponse: ReadMemoryResponse 数据包由目标发出, 以响应发出 ReadMemory 命令的主机。ReadMemoryResponse 数据包包含帧包数据和命令包数据, 且命令/响应标签置位为 ReadMemoryResponse 标签值 (0xA3)。标志置位于 kCommandFlag_HasDataPhase (1)。

下述状态码和数据字节计数参数的参数计数置位为 2。

表 14-14. ReadMemoryResponse 参数

字节 #	参数	描述
0 .. 3	Status code	相关读存储器命令的状态。
4 - 7	Data byte count	数据阶段发出的字节数。

FlashReadOnceResponse: FlashReadOnceResponse 数据包由目标发出，以响应发出 FlashReadOnce 命令的主机。FlashReadOnceResponse 数据包包含帧包数据和命令包数据，命令/响应标签置位为 FlashReadOnceResponse 标签值 (0xAF)，且标志置位为 0。参数计数置位为 2 加上所请求的将在 FlashReadOnceCommand 中读取的字数。

表 14-15. FlashReadOnceResponse 参数

字节 #	数值	参数
0 - 3		Status code
4 - 7		Byte count to read
...		...
		Can be up to 20 bytes of requested read data.

FlashReadResourceResponse: FlashReadResourceResponse 数据包由目标发出，以响应发出 FlashReadResource 命令的主机。FlashReadResourceResponse 数据包包含帧包数据和命令包数据，命令/响应标签置位为 FlashReadResourceResponse 标签值 (0xB0)，且标志置位为 kCommandFlag_HasDataPhase (1)。

表 14-16. FlashReadResourceResponse Parameters

字节 #	数值	参数
0 - 3		Status code
4 - 7		Data byte count

14.3.6 Flashloader 命令 API

如以前的章节所述，所有 Kinetis Flashloader 命令 API 均采用帧包包含的命令数据包格式。

- 对于受 Flashloader 支持的命令列表，请参阅，see [表 14-2](#)。
- 如需查看由 Kinetis Flashloader 返回的状态代码列表，请参阅 Flashloader，see [表 14-49](#)。

注

本节中的所有示例描述了使用帧包的串行外设上的字节流量。USB HID 事务使用 USB HID 报告数据包，而不是使用此节中所示的序列帧包。请参阅 [HID 报告](#) 了解有关 USB HID 数据包结构的详情。

14.3.6.1 Call 命令

Call 命令会执行命令中指定的存储器地址上的函数。该地址必须是位于可访问 Flash (内部或外部) 或 RAM 中的一个有效的存储器位置。该命令支持 32 位参数的传递。尽管命令支持堆栈地址，但此次调用仍会使用当前堆栈指针予以执行。执行功能之后，一个 32 位的返回值将在通用响应信息中返回。

表 14-17. Call 命令的参数

字节 #	命令
0 .. 3	调用地址
4 - 7	参数字组
8 - 11	堆栈指针

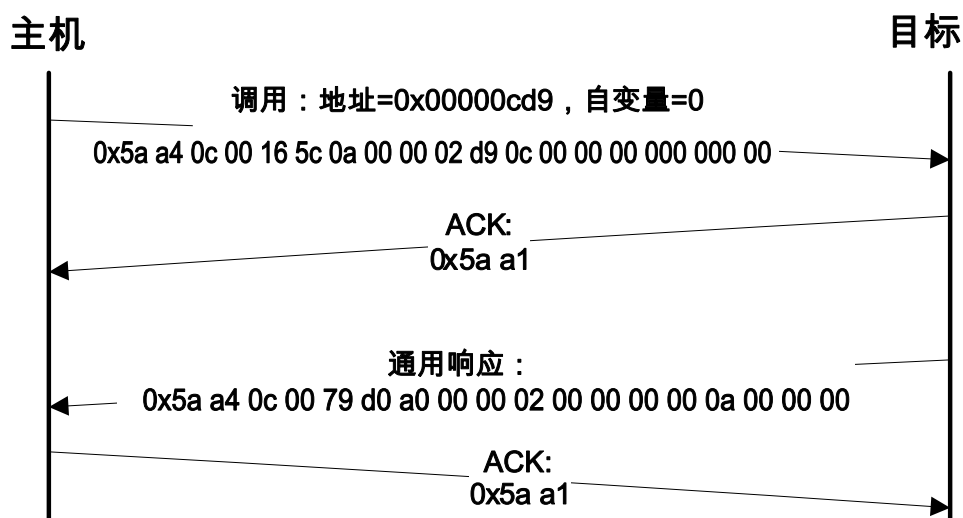


图 14-6. Call 命令的协议序列

响应: 目标 (Kinetis Flashloader) 会返回一个状态码置位为调用函数返回值或置位为 `kStatus_InvalidArgument (105)` 的 `GenericResponse` 数据包。

14.3.6.2 GetProperty 命令

GetProperty 命令被用于查询 Flashloader 的多种属性和设置。每个受支持的属性都含有与之相关的独有 32 位标志。该标志占据命令数据包的第一个参数位。目标返回一个 GetPropertyResponse 数据包，该数据包的属性值是 GetProperty 命令中的标记位确定的属性值。

属性是指可通过 GetProperty 或 SetProperty 命令访问的已限定的数据单元。属性可能为只读或只写。所有读写属性都是 32 位整数，因此很容易携带在命令参数中。

如需查看 Kinetis Flashloader 支持的属性及与之相关的 32 位属性标记位的列表，请参阅 [表 14-45](#)。

该 32 位属性是 GetProperty 命令所需的唯一参数。

表 14-18. GetProperty 命令参数

字节 #	命令
0 .. 3	属性标签

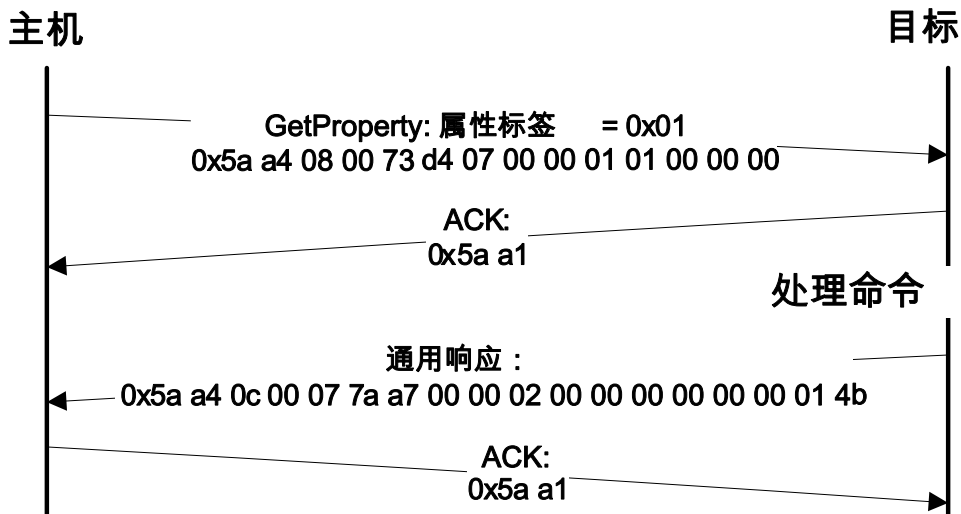


图 14-7. GetProperty 命令的协议序列

表 14-19. GetProperty 命令数据包格式 (范例)

GetProperty	参数	数值
帧包	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x08 0x00
	crc16	0x73 0xD4
命令包	commandTag	0x07 – GetProperty

下一页继续介绍此表...

表 14-19. GetProperty 命令数据包格式（范例）（继续）

GetProperty	参数	数值
	flags	0x00
	reserved	0x00
	parameterCount	0x01
	propertyTag	0x00000001 - CurrentVersion

GetProperty 命令没有数据阶段。

响应：响应 GetProperty 命令期间，目标会发送一个相应标记位置位为 0xA7 的 GetPropertyResponse 数据包。参数量表示发送的用作属性值的参数数量，首个参数显示状态代码 0，之后的参数显示属性值。下个图标显示一个 GetPropertyResponse 数据包范例。

表 14-20. GetProperty 响应数据包格式（范例）

GetPropertyResponse	参数	数值
帧包	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x0c 0x00 (12 bytes)
	crc16	0x07 0x7a
命令包	responseTag	0xA7
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	status	0x00000000
	propertyValue	0x0000014b - CurrentVersion

14.3.6.3 SetProperty 命令

SetProperty 命令用于更改 Kinetis Flashloader 属性或选项的值。然而，SetProperty 命令只能改变可写属性的值 — 参见表 14-45。Get/SetProperty 命令使用的属性，如果想为只读属性设置值，Kinetis Flashloader 将返回一个错误。

SetProperty 命令所需的 2 个参数为属性标签和要设置的新值。

表 14-21. SetProperty 命令的参数

字节 #	命令
0 .. 3	属性标签
4 - 7	属性值

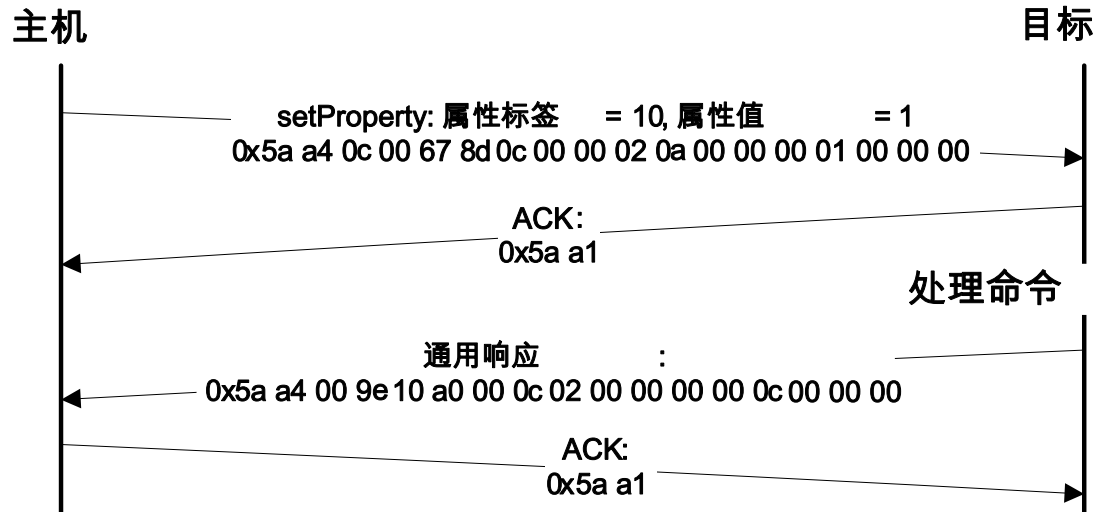


图 14-8. SetProperty 命令的协议序列

表 14-22. SetProperty 命令数据包格式 (范例)

SetProperty	参数	数值
帧包	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x0C 0x00
	crc16	0x67 0x8D
命令包	commandTag	0x0C – SetProperty with property tag 10
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	propertyTag	0x0000000A - VerifyWrites
	propertyValue	0x00000001

SetProperty 命令没有数据阶段。

响应：目标（Kinetis Flashloader）会返回一个带有以下状态码的通用响应数据包：

表 14-23. SetProperty 响应状态代码

状态代码
kStatus_Success
kStatus_ReadOnly
kStatus_UnknownProperty
kStatus_InvalidArgument

14.3.6.4 FlashEraseAll 命令

FlashEraseAll 命令执行擦除整个 Flash 存储器 or of QuadSPI memory (if that is the case)。如果任何 Flash 区域被保护，FlashEraseAll 命令将失败并返回一个错误状态代码。如果 Flash 加密是启用状态，当执行 FlashEraseAll 命令时会通过设置 FTFA_FSEC 寄存器解除 Flash 加密。然而 Flash 配置字段中的 FSEC 字段被擦除，因此除非重新编程，否则 Flash 加密在下次系统重置时会被重新使能。FlashEraseAll 命令的命令标签是设置在命令数据包中的 commandTag 字段值为 0x01。

FlashEraseAll 命令无需任何参数。

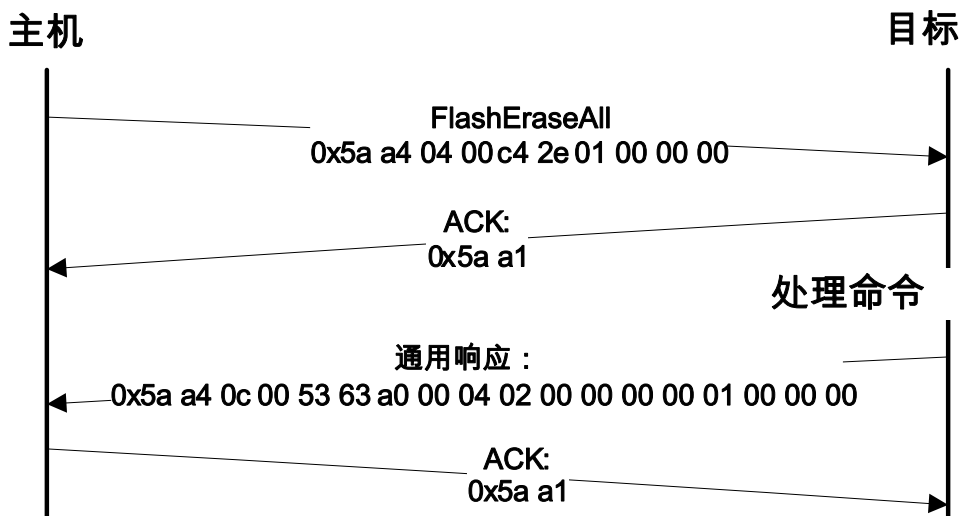


图 14-9. FlashEraseAll 命令的协议序列

表 14-24. FlashEraseAll 命令数据包格式 (范例)

FlashEraseAll	参数	数值
帧包	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x04 0x00

下一页继续介绍此表...

表 14-24. FlashEraseAll 命令数据包格式（范例）（继续）

FlashEraseAll	参数	数值
	crc16	<ul style="list-style-type: none"> If MemoryID = 0 (internal flash), then crc16 = 0x0C 0x22 If MemoryID = 1 (QSPI0 memory), then crc16 = 0xB8 0x54
	crc16	0xC4 0x2E
命令包	commandTag	0x01 - FlashEraseAll
	flags	0x00
	reserved	0x00
	parameterCount	0x00
	MemoryID	<ul style="list-style-type: none"> If MemoryID = 0x00h, then internal flash. If MemoryID = 0x01h, then QSPI0 memory.

FlashEraseAll 命令没有数据阶段。

响应：目标（Kinetis Flashloader）会返回一个带有状态码的通用响应数据包，如果命令成功执行，该状态码被设置为 kStatus_Success，否则该状态码为对应的错误状态码。

14.3.6.5 FlashEraseRegion 命令

FlashEraseRegion 命令执行擦除 Flash 存储器的一个或多个扇区，或擦除连接的 SPI Flash 器件的 Flash 内的指定区域。

FlashEraseRegion 命令所需的 2 个参数是起始地址和字节数。起始地址和字节数参数必须是，否则 FlashEraseRegion 命令会出错并返回 kStatus_FlashAlignmentError (0x101)。如果指定区域不在 Flash 存储器空间内，FlashEraseRegion 命令会出错并返回 kStatus_FlashAddressError (0x102)。如果指定区域有任何部分受到保护，FlashEraseRegion 命令会出错并返回 kStatus_MemoryRangeInvalid (0x10200)。

表 14-25. FlashEraseRegion 命令的参数

字节 #	参数
0 .. 3	Start address
4 - 7	Byte count

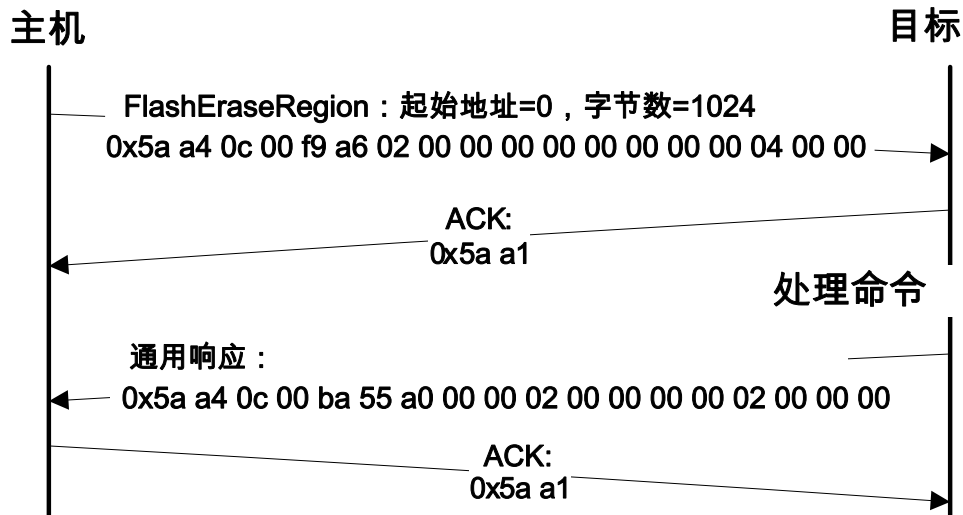


图 14-10. FlashEraseRegion 命令的协议序列

表 14-26. FlashEraseRegion 命令数据包格式 (范例)

FlashEraseRegion	参数	数值
帧包	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x0C 0x00
	crc16	0xF9 0x A6
命令包	commandTag	0x02, kCommandTag_FlashEraseRegion
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	startAddress	0x00 0x00 0x00 0x00 (0x0000_0000)
	byte count	0x00 0x04 0x00 0x00 (0x400)

FlashEraseRegion 命令没有数据阶段。

响应：目标（Kinetic Flashloader）会返回一个带有以下错误状态代码的通用响应数据包。

表 14-27. FlashEraseRegion 响应状态代码

状态代码
kStatus_Success (0x0)
kStatus_MemoryRangeInvalid (0x10200)
kStatus_FlashAlignmentError (0x101)
kStatus_FlashAddressError (0x102)
kStatus_FlashAccessError (0x103)
kStatus_FlashProtectionViolation (0x104)
kStatus_FlashCommandFailure (0x105)

14.3.6.6 FillMemory 命令

FillMemory 命令按一种数据模式填充存储器的一系列字节。它与 WriteMemory 命令采用同一规则。FillMemory 和 WriteMemory 之间的区别在于 FillMemory 命令参数中包含一种数据模式，且 FillMemory 命令没有数据阶段，但 WriteMemory 有数据阶段。

表 14-28. FillMemory 命令的参数

字节 #	命令
0 .. 3	待填充存储器的起始地址
4 - 7	待按照该模式写入的字节数 <ul style="list-style-type: none"> • 起始地址应是 32 位对齐。 • 字节数应能被 4 整除。
8 - 11	32 位模式

- 要按一种字节模式（8 位）填充，必须在 32 位模式中将字节复制 4 次。
- 要按一种短型模式（16 位）填充，必须在 32 位模式中将该短型值复制 2 次。

例如，要填充 0xFE 字节值，则字模式应是 0xFEFEFEFE；要填充 0x5AFE 短型值，则字模式应是 0x5AFE5AFE。

向 flash 写入时必须特别注意。

- 首先，任何被写入的 flash 扇区必须已利用 FlashEraseAll 或 FlashEraseRegion 命令进行过擦除。
- 向 flash 写入需要起始地址。
- 如果 VerifyWrites 属性为真，则向 flash 写入还会执行 flash 验证编程操作。

向 RAM 写入时，起始地址无需对齐，且数据不会被填补。

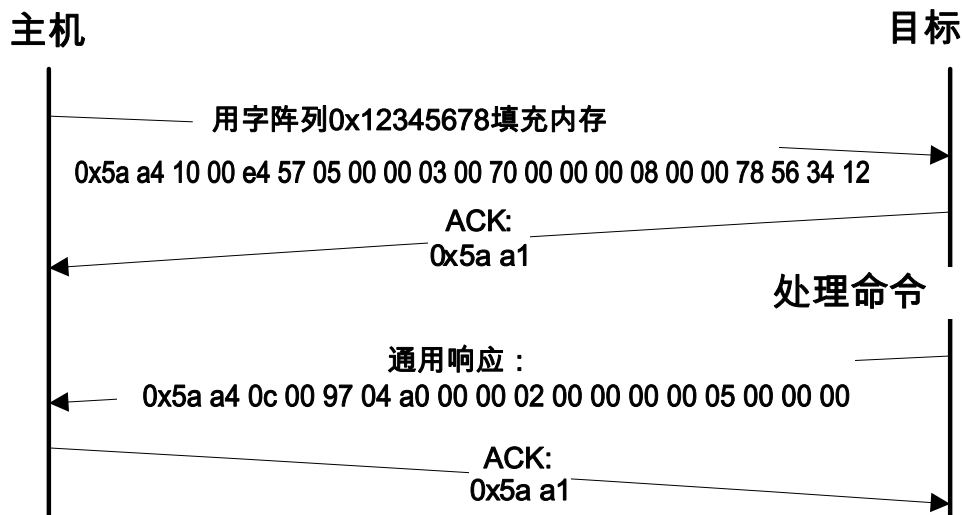


图 14-11. FillMemory 命令的协议序列

表 14-29. FillMemory 命令数据包格式 (范例)

FillMemory	参数	数值
帧包	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x10 0x00
	crc16	0xE4 0x57
命令包	commandTag	0x05 – FillMemory
	flags	0x00
	Reserved	0x00
	parameterCount	0x03
	startAddress	0x00007000
	byteCount	0x00000800
	patternWord	0x12345678

FillMemory 命令没有数据阶段。

响应：目标（Kinetis Flashloader）会返回一个带有状态码的通用响应数据包，如果命令成功执行，该状态码被设置为 kStatus_Success，否则该状态码为合适的错误状态码。

14.3.6.7 FlashProgramOnce 命令

FlashProgramOnce 命令将命令数据包中提供的数据写入一次编程字段中的指定字节范围。写入一次编程字段时必须特别注意。

- 一次编程字段只支持编程一次，所以任何想要对一次编程字段重新编程的尝试会得到错误的响应。
- 写入一次编程字段需要字节数为 4 字节对齐或 8 字节对齐。

FlashProgramOnce 命令使用 3 个参数：index、byteCount、和 data。

表 14-30. FlashProgramOnce 命令的参数

字节 #	命令
0 .. 3	一次编程字段的索引
4 - 7	字节数（必须被 4 整除）
8 - 11	数据
12 - 16	数据

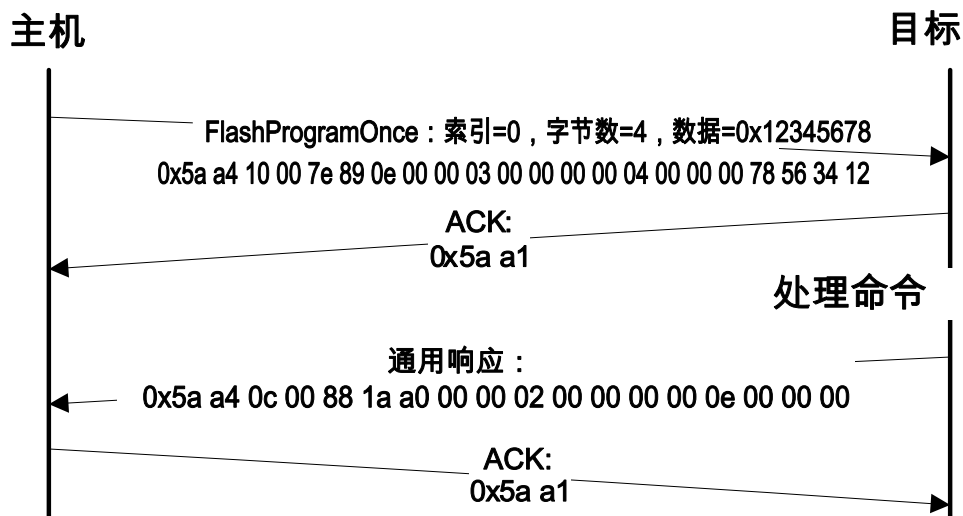


图 14-12. FlashProgramOnce 命令的协议序列

表 14-31. FlashProgramOnce 命令数据包格式（范例）

FlashProgramOnce	参数	数值
帧包	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x10 0x00
	crc16	0x7E4 0x89
命令包	commandTag	0x0E – FlashProgramOnce
	flags	0
	reserved	0
	parameterCount	3
	index	0x0000_0000
	byteCount	0x0000_0004

下一页继续介绍此表...

表 14-31. FlashProgramOnce 命令数据包格式（范例）（继续）

FlashProgramOnce	参数	数值
	data	0x1234_5678

响应：如果命令成功执行，目标（Kinetis Flashloader）会返回一个带有状态码的 GenericResponse 数据包，该状态码被设置为 kStatus_Success，或者该状态码为合适的错误状态码。

14.3.6.8 FlashReadOnce 命令

FlashReadOnce 命令通过给定的索引和字节数返回一次编程字段的内容。
FlashReadOnce 命令使用 2 个参数：index 和 byteCount。

表 14-32. FlashReadOnce 命令的参数

字节 #	参数	说明
0 .. 3	索引	一次编程字段的索引（从中读取）
4 - 7	byteCount	待读取和返回到调用方的字节数

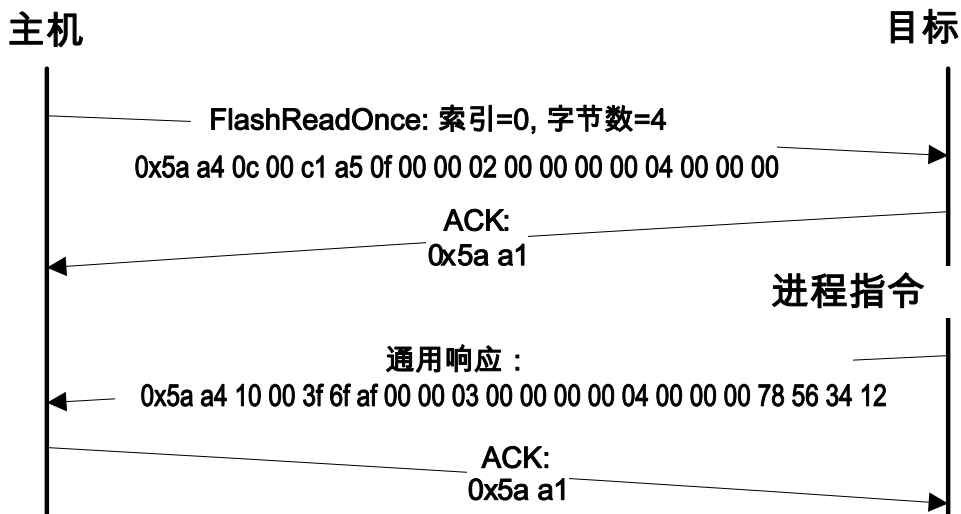


图 14-13. FlashReadOnce 命令的协议序列

表 14-33. FlashReadOnce 命令数据包格式（范例）

FlashReadOnce	参数	数值
帧包	start byte	0x5A
	packetType	0xA4

下一页继续介绍此表...

表 14-33. FlashReadOnce 命令数据包格式（范例）（继续）

FlashReadOnce	参数	数值
	length	0x0C 0x00
	crc	0xC1 0xA5
命令包	commandTag	0x0F – FlashReadOnce
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	index	0x0000_0000
	byteCount	0x0000_0004

表 14-34. FlashReadOnce 响应格式（范例）

FlashReadOnce 响应	参数	数值
帧包	start byte	0x5A
	packetType	0xA4
	length	0x10 0x00
	crc	0x3F 0x6F
命令包	commandTag	0xAF
	flags	0x00
	reserved	0x00
	ParameterCount	0x03
	status	0x0000_0000
	byteCount	0x0000_0004
	data	0x1234_5678

响应：命令执行成功时，目标（Kinetis Flashloader）会返回一个 FlashReadOnceResponse 数据包，其中状态代码设置为 kStatus_Success，并有从一次编程字段中读取的字节数和相应的数据，否则将会返回设置为适当错误状态码的状态代码，并且字节数设置为 0。

14.3.6.9 FlashReadResource 命令

FlashReadResource 命令通过给定的偏移、字节数和选项，返回 IFR 字段或 Flash 固件 ID 的内容。FlashReadResource 命令使用 3 个参数：start address、byteCount、和 option。

表 14-35. FlashReadResource 命令的参数

字节 #	参数	命令
0 .. 3	start address	待读取的指定非易失性存储器的起始地址
4 - 7	byteCount	待读取的字节数
8 - 11	option	0: IFR 1: Flash 固件 ID

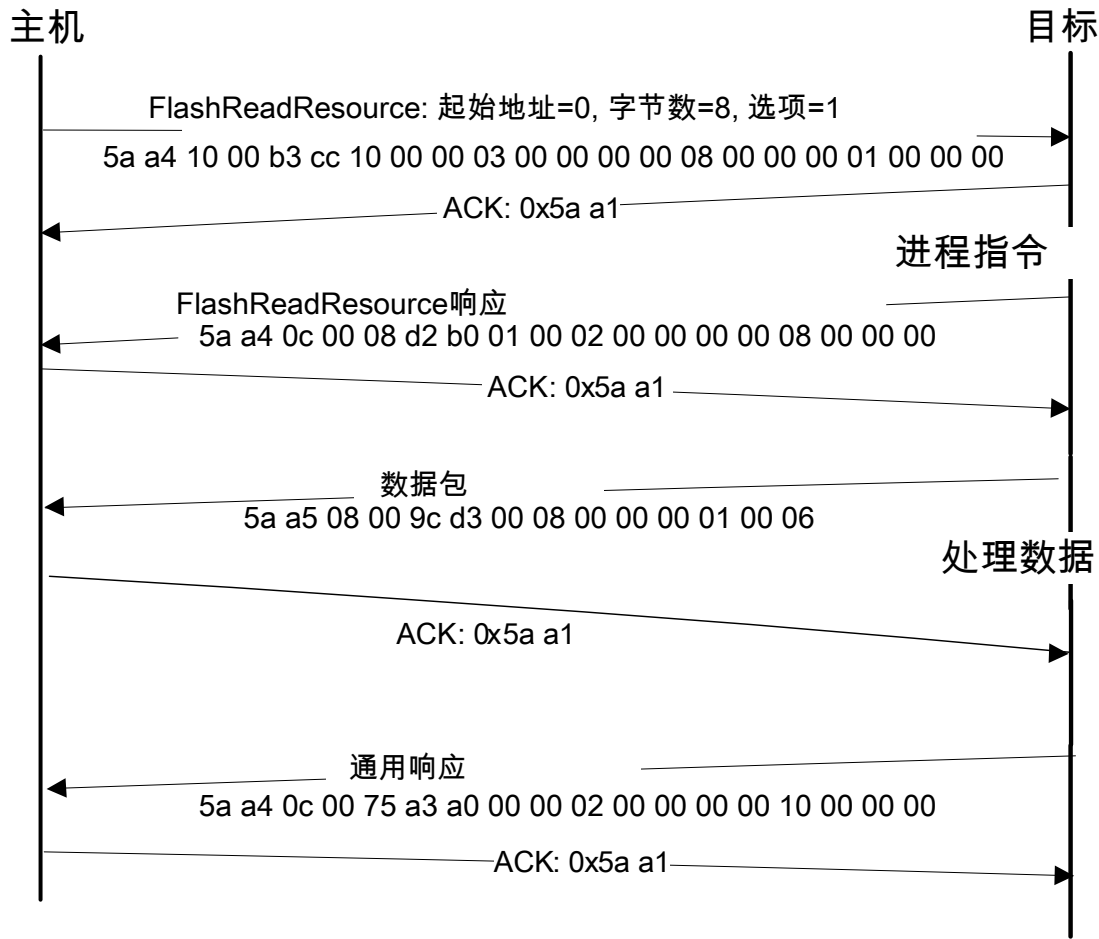


图 14-14. FlashReadResource 命令的协议序列

表 14-36. FlashReadResource 命令数据包格式 (范例)

FlashReadResource	参数	数值
帧包	start byte	0x5A
	packetType	0xA4
	length	0x10 0x00
	crc	0xB3 0xCC
命令包	commandTag	0x10 – FlashReadResource
	flags	0x00

下一页继续介绍此表...

表 14-36. FlashReadResource 命令数据包格式 (范例) (继续)

FlashReadResource	参数	数值
	reserved	0x00
	ParameterCount	0x03
	startAddress	0x0000_0000
	byteCount	0x0000_0008
	option	0x0000_0001

表 14-37. FlashReadResource 响应格式 (范例)

FlashReadResource 响应	参数	数值
帧包	start byte	0x5A
	packetType	0xA4
	length	0x0C 0x00
	crc	0xD2 0xB0
命令包	commandTag	0xB0
	flags	0x01
	Reserved	0x00
	parameterCount	0x02
	status	0x0000_0000
	byteCount	0x0000_0008

数据阶段：FlashReadResource 命令具有数据阶段。因为目标 (Kinetis Flashloader) 工作于从机模式，所以主机必须持续接收数据包直到接收到 *FlashReadResource* 命令在 *byteCount* 参数中指定的字节数。

14.3.6.10 WriteMemory 命令

WriteMemory 命令将数据阶段中提供的数据写入存储器 (flash 或 RAM) 中指定的字节范围。。但是，如果已使能 flash 保护，则无法成功写入受保护扇区。

向 flash 写入时必须特别注意。

- 首先，任何被写入的 flash 扇区必须已利用 FlashEraseAll 或 FlashEraseRegion 命令进行过擦除。
- 向 flash 写入需要起始地址 4 位对齐 ([1:0] = 00)。
- 如果 VerifyWrites 属性为真，则向 flash 写入还会执行 flash 验证编程操作。

向 RAM 写入时，起始地址无需对齐，且数据不会被填补。

WriteMemory 命令所需的 2 个参数是起始地址和字节数。

表 14-38. WriteMemory 命令的参数

字节 #	命令
0 .. 3	起始地址
4 - 7	字节计数

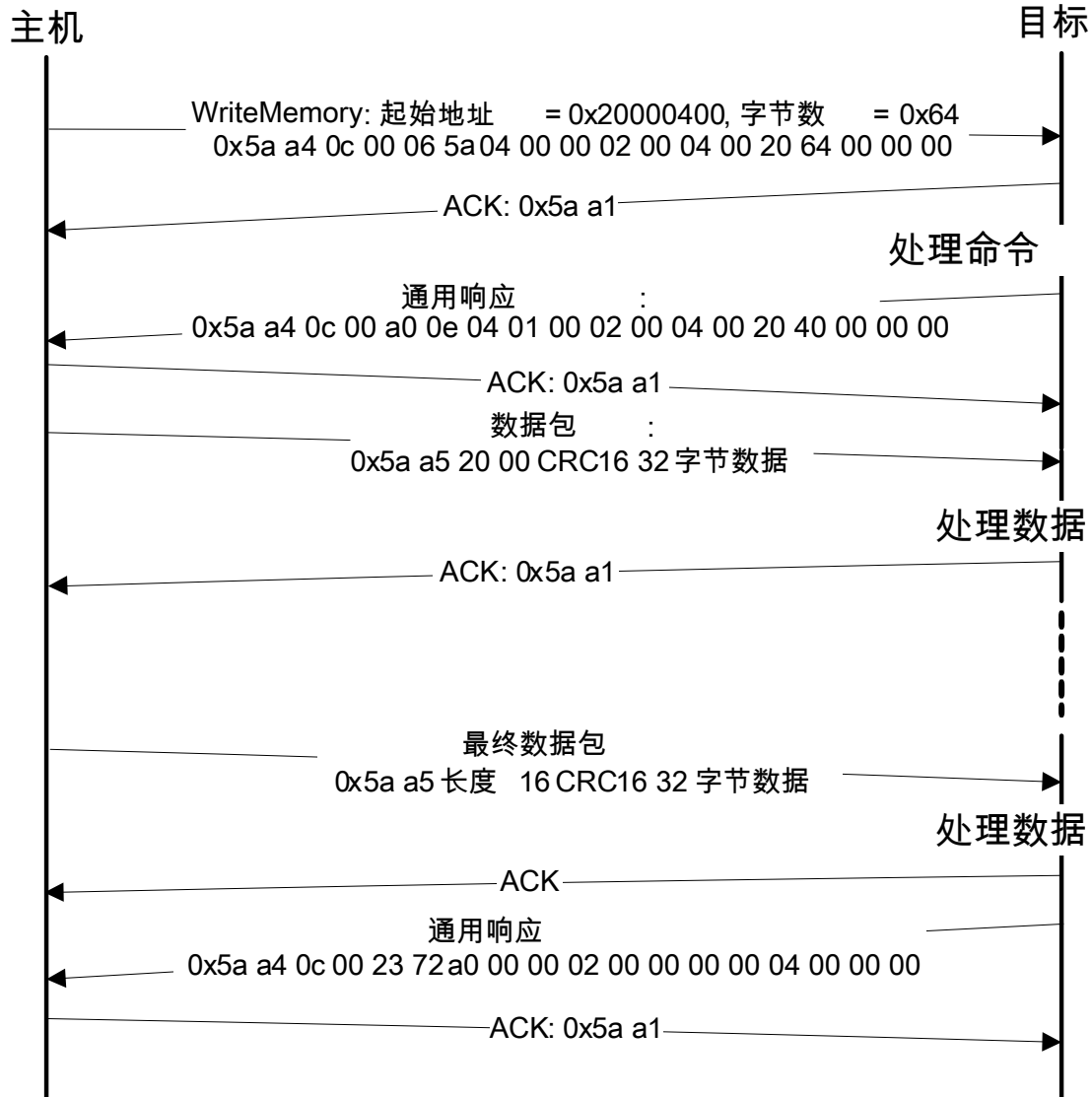


图 14-15. WriteMemory 命令的协议序列

表 14-39. WriteMemory 命令数据包格式 (范例)

WriteMemory	参数	数值
帧包	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x0C 0x00

下一页继续介绍此表...

表 14-39. WriteMemory 命令数据包格式（范例）（继续）

WriteMemory	参数	数值
	crc16	0x06 0x5A
命令包	commandTag	0x04 - writeMemory
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	startAddress	0x20000400
	byteCount	0x00000064

数据阶段：WriteMemory 命令有数据阶段；直至目标收到由 WriteMemory 命令的 byteCount 参数中指定的字节数数据之后，主机才会发送数据包。

响应：目标（Kinetis Flashloader）会返回一个带有状态码的 GenericResponse 数据包，如果命令成功执行，该状态码被设置为 kStatus_Success，或设置为合适的错误状态码。

14.3.6.11 ReadMemory 命令

ReadMemory 命令在给定地址中按照指定的字节数返回存储器内容。该命令可读取存储器中 CPU 可以访问且无安全保护的任意区域。

ReadMemory 命令所需的 2 个参数是起始地址和字节数。

表 14-40. ReadMemory 命令的参数

字节	参数	说明
0-3	Start address	待读取存储器的起始地址
4-7	Byte count	待读取和返回到调用方的字节数

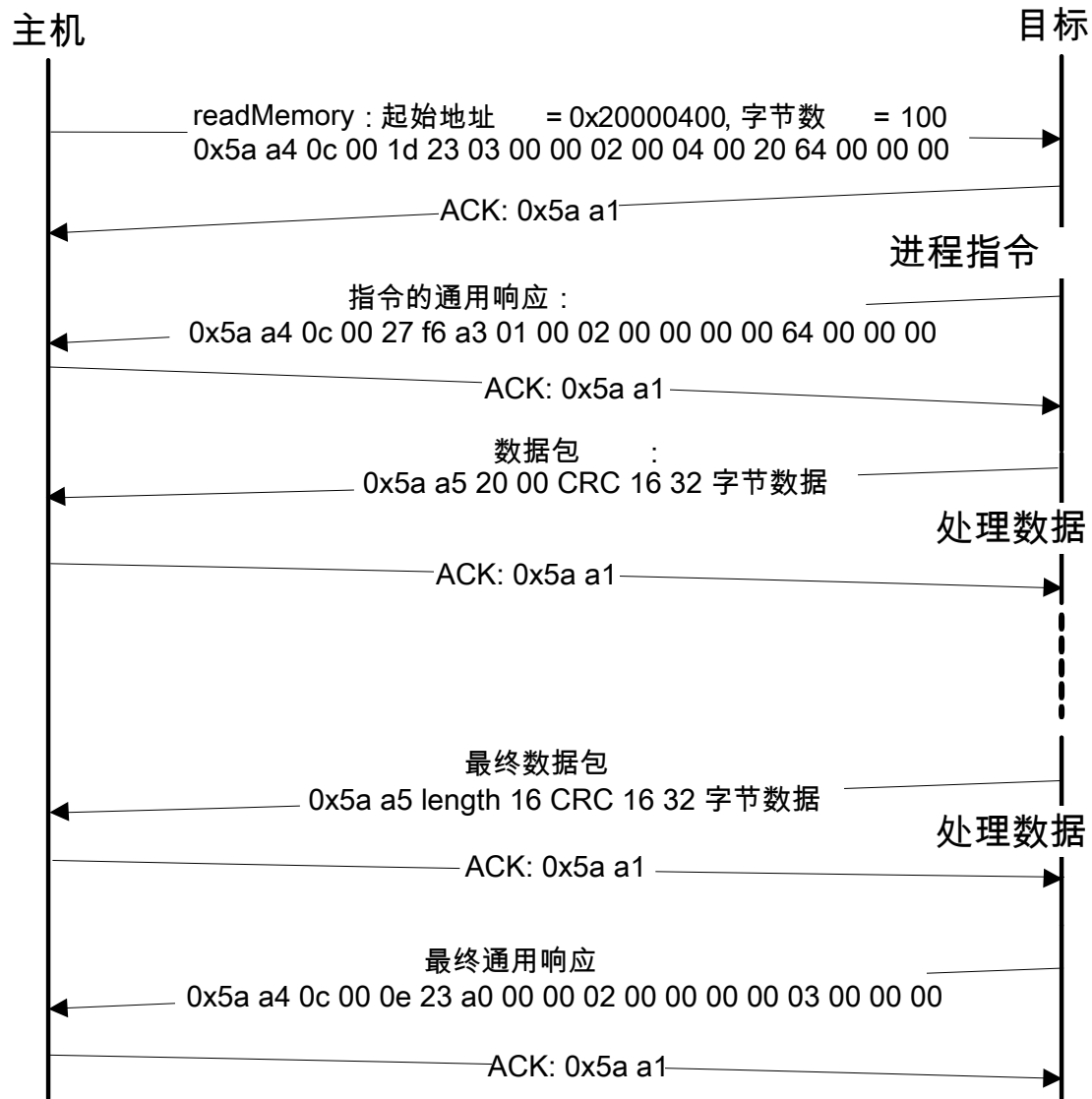


图 14-16. ReadMemory 命令的协议序列

表 14-41. ReadMemory 命令数据包格式 (范例)

ReadMemory	参数	数值
帧包	Start byte	0x5A0xA4,
	packetType	kFramingPacketType_Command
	length	0x0C 0x00
	crc16	0x1D 0x23
命令包	commandTag	0x03 - readMemory
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	startAddress	0x20000400
	byteCount	0x00000064

数据阶段：ReadMemory 命令具有数据阶段。由于目标（Kinetis Flashloader）工作于从机模式，因此主机必须持续接收数据包直到接收到 ReadMemory 命令在 byteCount 参数中指定的字节数数据。

响应：目标（Kinetis Flashloader）会返回一个带有状态码的 GenericResponse 数据包，如果命令成功执行，该状态码被设置为 kStatus_Success，否则该状态码为合适的错误状态码。

14.3.6.12 Execute 命令

Execute 命令使 flashloader 将程序计数器设置为提供的跳转地址处的代码，R0 设置为所提供的参数，堆栈指针设置为提供的堆栈指针地址。在跳转之前，系统返回到复位状态。

跳转地址、函数参数指针和堆栈指针是 Execute 命令所需的参数。

表 14-42. Execute 命令的参数

字节 #	命令
0 .. 3	跳转地址
4 - 7	参数字组
8 - 11	堆栈指针地址

Execute 命令没有数据阶段。

响应：在执行 Execute 命令之前，目标（Kinetis Flashloader）会验证参数，并返回一个带有状态码的 GenericResponse 数据包，该状态码被设置为 kStatus_Success 或为适当的错误状态代码。

14.3.6.13 Reset 命令

Reset 命令将导致 flashloader 复位芯片。

Reset 命令无需参数。

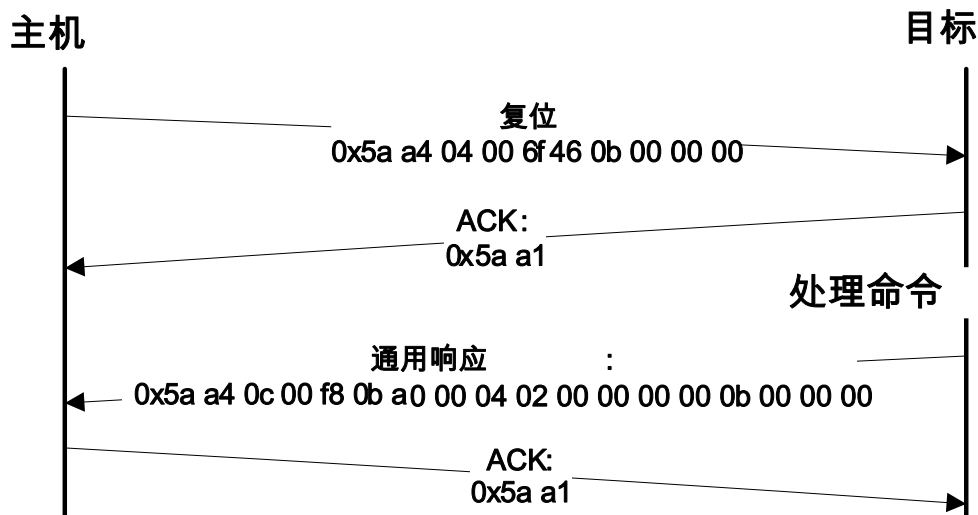


图 14-17. Reset 命令的协议序列

表 14-43. Reset 命令数据包格式 (范例)

Reset	参数	数值
帧包	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x04 0x00
	crc16	0x6F 0x46
命令包	commandTag	0x0B - reset
	flags	0x00
	reserved	0x00
	parameterCount	0x00

Reset 命令没有数据阶段。

响应：在复位芯片之前，目标 (Kinetis Flashloader) 会返回一个带有状态码为 `kStatus_Success` 的 `GenericResponse` 数据包。

14.3.6.14 ReceiveSBFile 命令

ReceiveSBFile 命令 (ReceiveSbFile) 将开始向目标传输 SB 文件。该命令只指定将在数据阶段中传输的 SB 文件的大小 (字节)。会将 SB 文件视作由 Bootloader 接收的文件进行处理。

表 14-44. ReceiveSBFile 命令的参数

字节 #	命令
0..3	字节计数

数据阶段: Receive SB 文件命令没有数据阶段; 直至目标收到由 Receive SB 文件命令的 byteCount 参数指定的数据字节数之后, 主机才会发送数据包。

响应: 目标 (Flashloader) 会返回一个带有状态码的 GenericResponse 数据包, 如果命令成功执行, 该状态码被设置为 kStatus_Success, 否则该状态码为合适的错误码。

14.4 支持的外设

本节描述了受 Kinetis Flashloader 支持的外设。

14.4.1 I2C 外设

Kinetis Flashloader 支持数据通过 I2C 外设加载到 Flash, I2C 外设作为 I2C 从机。7 位从地址在传输中使用。

Kinetis Flashloader 将 0x10 用作 I2C 从机地址, 并且支持具有 400 kbps 的 I2C 波特率。

由于 I2C 外设用作 I2C 从设备, 因此每次传输都应由主机开始, 并且每个传出数据包都应由主机获取。

- 传入数据包由采用所选 I2C 从机地址的主机发送, 并且方向位设置为写入。
- 传出数据包由采用所选 I2C 从机地址的主机读取, 并且方向位设置为读取。
- 如果目标正忙于处理或准备数据, 则 0x00 将作为响应发送至主机。

以下流程图演示主机如何读取来自目标的 ping 数据包、ACK 和响应。

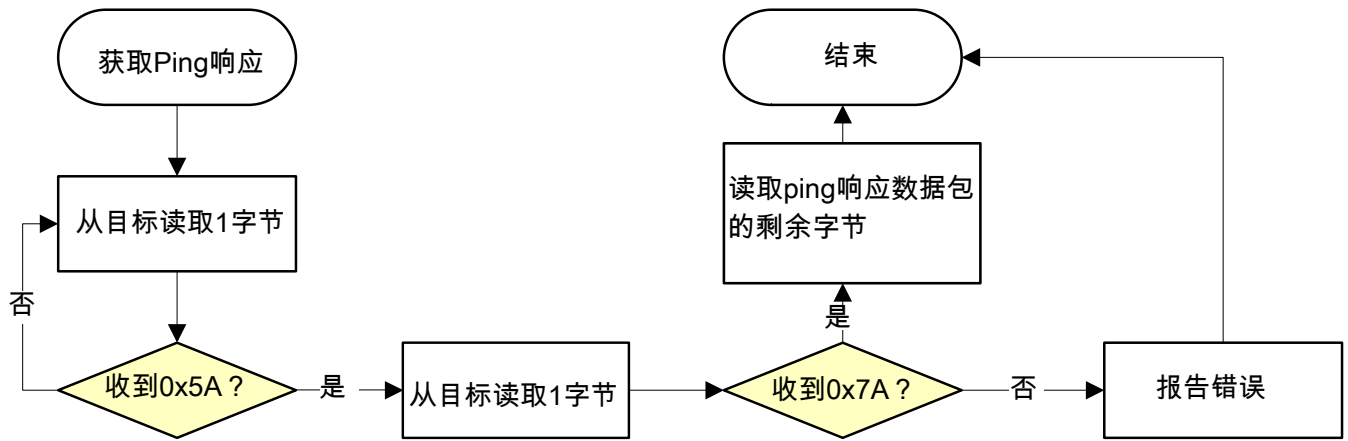


图 14-18. 主机通过 I2C 读取来自目标的 ACK 数据包

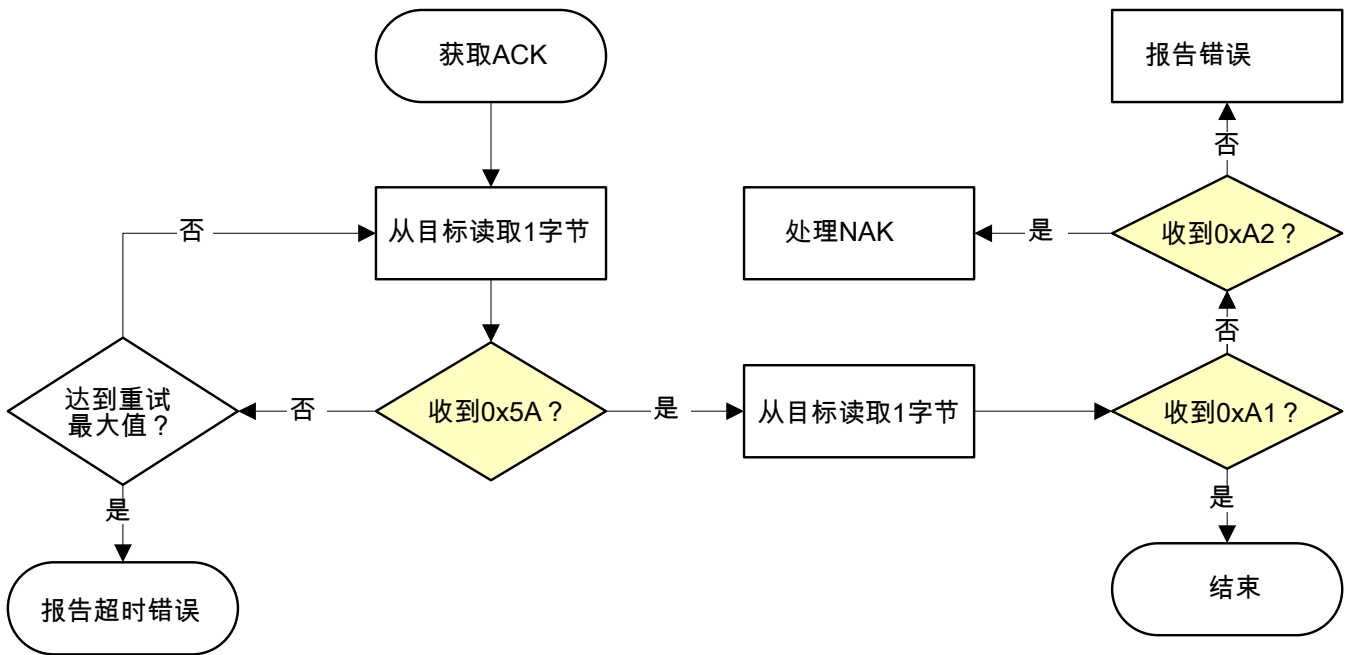


图 14-19. 主机通过 I2C 读取来自目标的响应

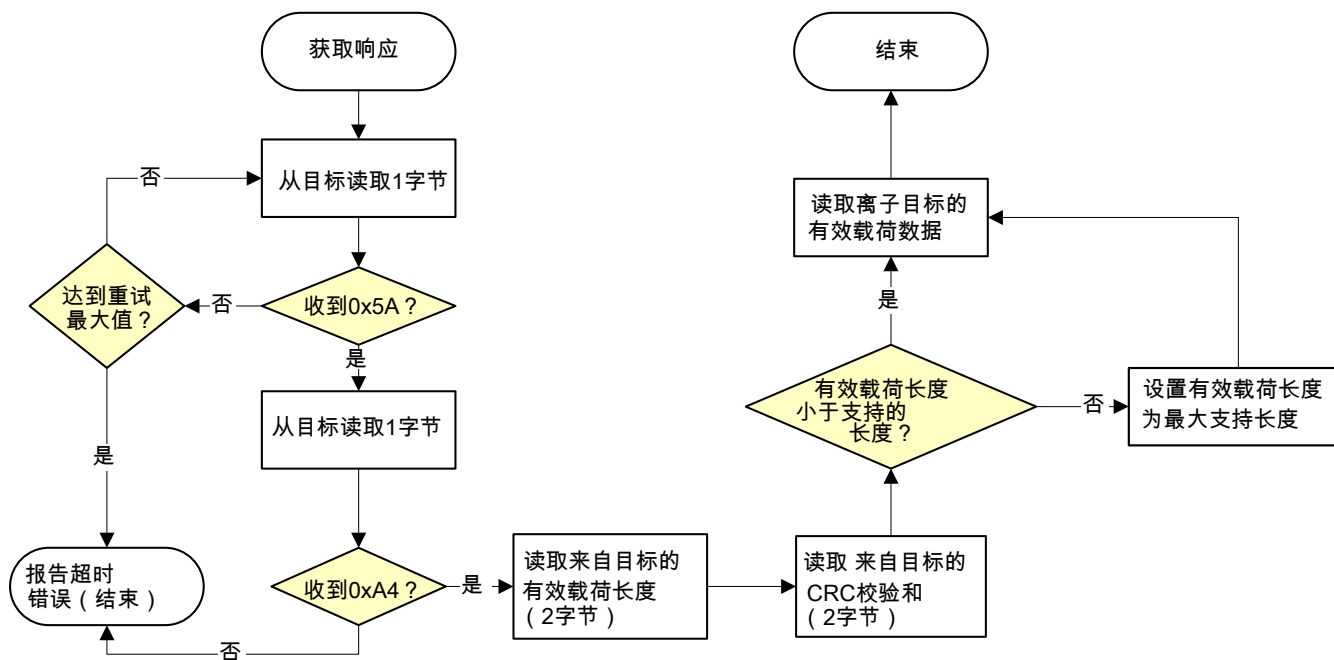


图 14-20. 主机通过 I2C 读取来自目标的响应

14.4.2 SPI 外设

Kinetis Flashloader 支持通过 SPI 外设将数据加载到 Flash, 其中 SPI 外设用作 SPI 从机。

Kinetis Flashloader 支持 400 kbps 的 SPI 波特率。

SPI 外设使用以下总线属性:

- 时钟相位 = 1 (第二边沿)
- 时钟极性 = 1 (低电平有效)

由于 Kinetis Flashloader 用作 SPI 从设备, 因此每次传输都应由主机开始, 并且每个传出数据包都应由主机获取。

SPI 上的传输与 I2C 稍有不同:

- 在发送出任何字节后, 主机将接收 1 个字节。
- 当主机将字节发送给目标时, 应忽略收到的字节。
- 主机通过向目标发送 0x00 来开始读取字节
- 如果目标处于以下条件下, 则字节 0x00 将作为响应发送给主机:
 - 正在处理传入数据包
 - 正在准备传出数据
 - 收到无效数据

SPI 总线配置如下所示:

- 相位 = 1; 在上升沿上采样数据
- 极性 = 1; 空闲为高
- MSB 先进行传输

对于目标没有实际数据进行发送的任何传输，目标（从机）负责确保 0x00 字节将返回到主机。主机使用帧包识别实际数据并且忽略 0x00 字节（它没有帧包）。

以下流程图显示主机如何通过 SPI 读取来自目标的 ping 响应、ACK 和命令响应。

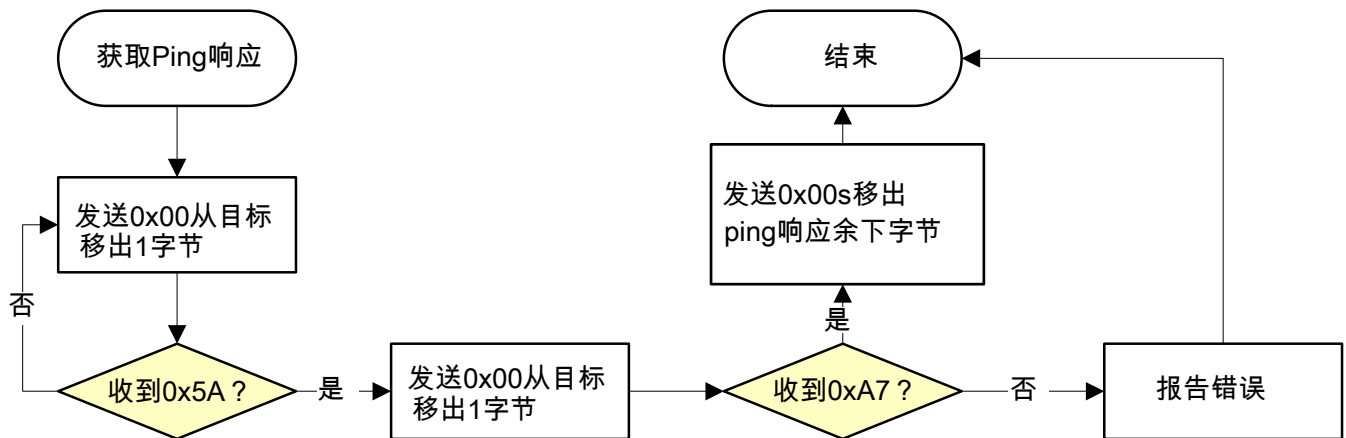


图 14-21. 主机通过 SPI 读取来自目标的 ping 数据包

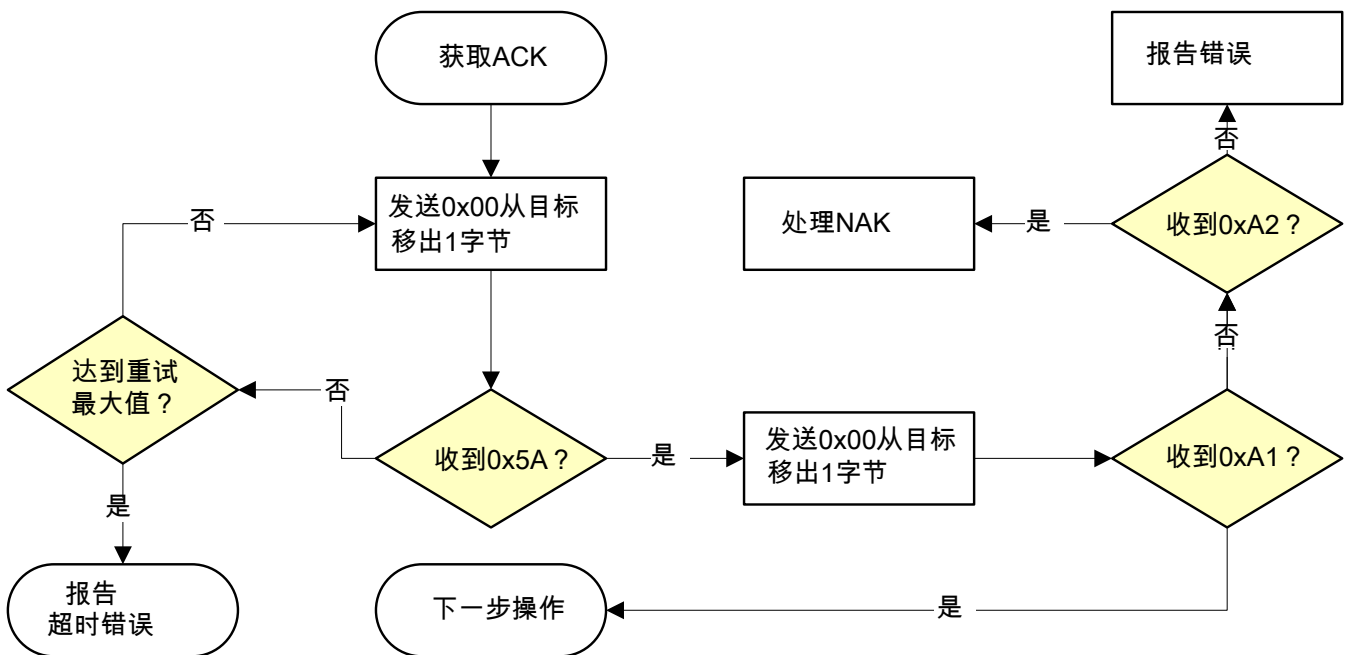


图 14-22. 主机通过 SPI 读取来自目标的 ACK

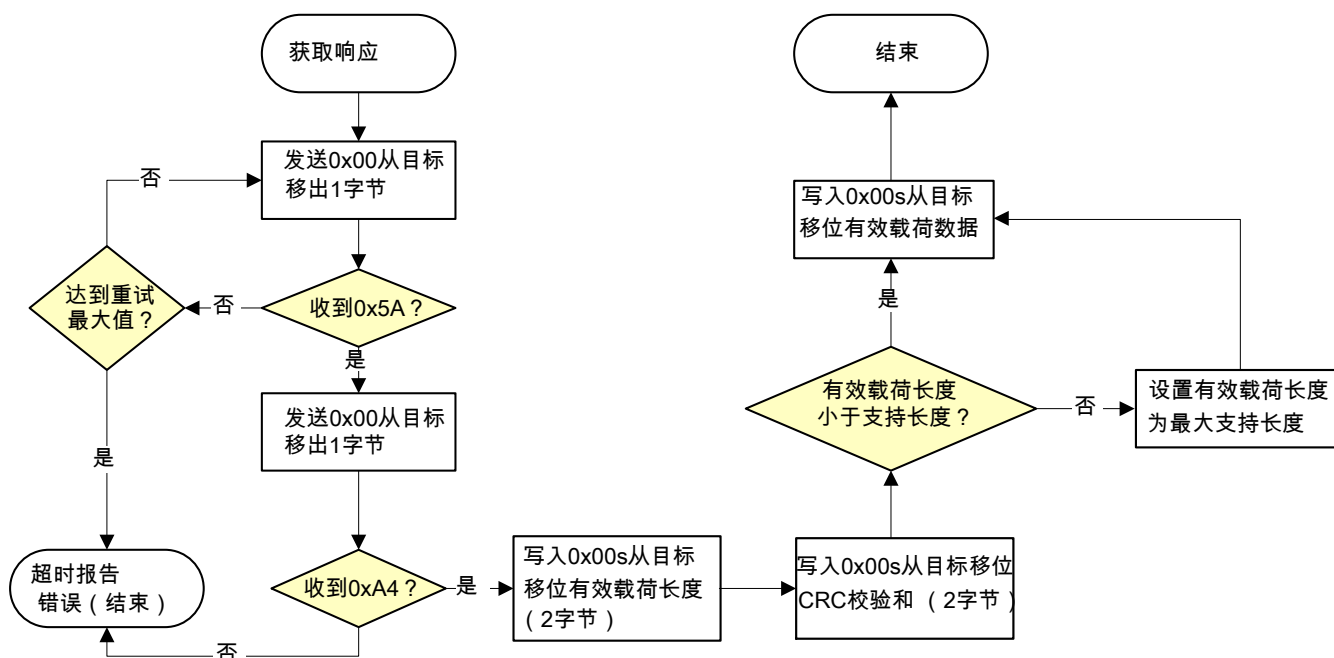


图 14-23. 主机通过 SPI 读取来自目标的响应

14.4.3 UART 外设

Kinetis Flashloader 针对 UART 外设提供了自动波特率检测算法，因此可提供灵活的波特率选择。

自动波特率功能：如果 UART n 用于连接到 flashloader，则 UART n _RX 引脚必须在检测阶段保持高电平并且不得保持浮空，这样才符合自动波特率检测算法。当 flashloader 检测到 UART n _RX 上的 ping 数据包 (0x5A 0xA6) 后，flashloader 固件将执行自动波特率序列。如果成功检测到波特率，则 flashloader 将以检测到的波特率发送 ping 数据包响应 [(0x5A 0xA7)、协议版本 (4 个字节)、协议版本选项 (2 个字节)、和 crc16 (2 个字节)]。Kinetis Flashloader 随后会进入一个循环，等待 flashloader 通过 UART 外设发出命令。

注

- 自动波特功能需要较高精确率的 ping 包 (+/-3%)，否则 ping 包会被当做噪音忽略。
- 必须以固定 UART 传输模式 (8 位数据、无校验位和 1 个停止位) 不断发生 ping 数据包的数据字节 (字节之间的延迟不超过 80 ms)。如果在逐个发送 ping 数据包的数据字节时它们之间的延迟超过 80 ms，则自动波特率检测算法可能会计算出错误的波特率。在这种情况下，应复位自动波特率检测状态机。

支持的波特率：波特率与 MCU 内核和系统时钟频率密切相关。所支持的典型波特率为 9600、19200、38400、57600、和 115200。

数据包传输：当自动波特率检测成功后，flashloader 通信可通过 UART 外设进行。以下流程图显式：

- 主机如何检测来自目标的 ACK
- 主机如何检测来自目标的 ping 响应
- 主机如何检测来自目标的命令响应

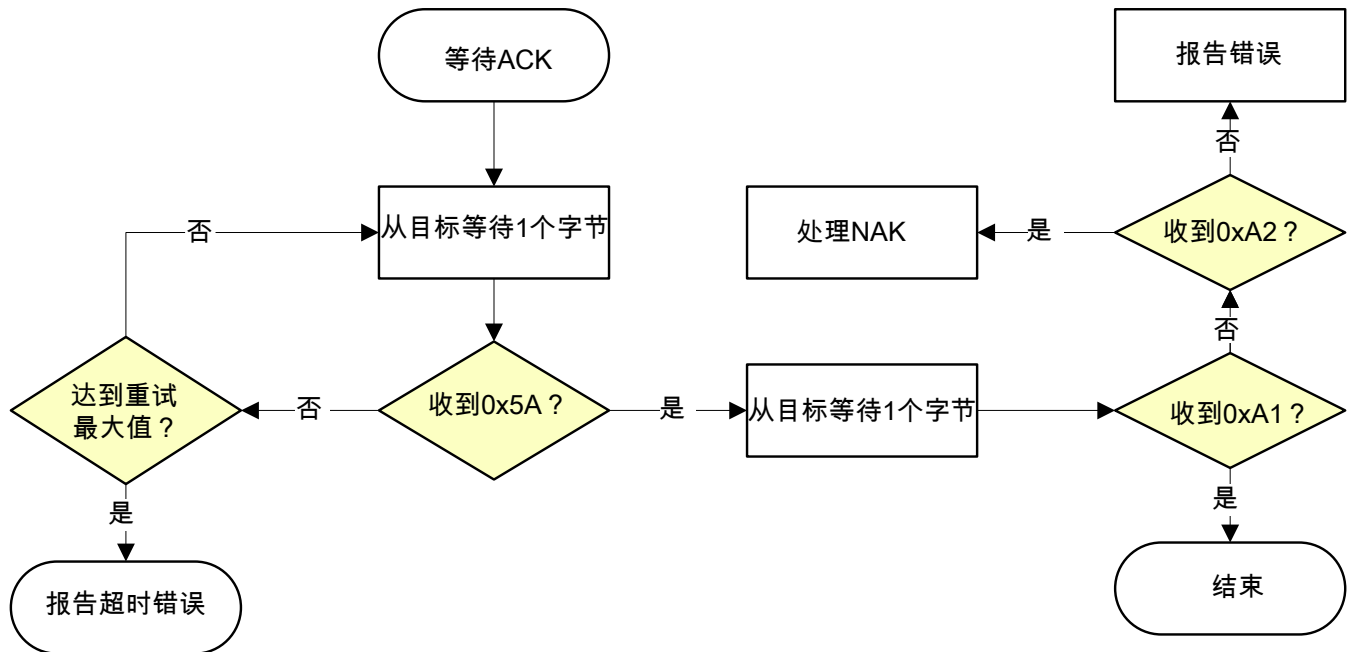


图 14-24. 主机通过 UART 读取来自目标的 ACK

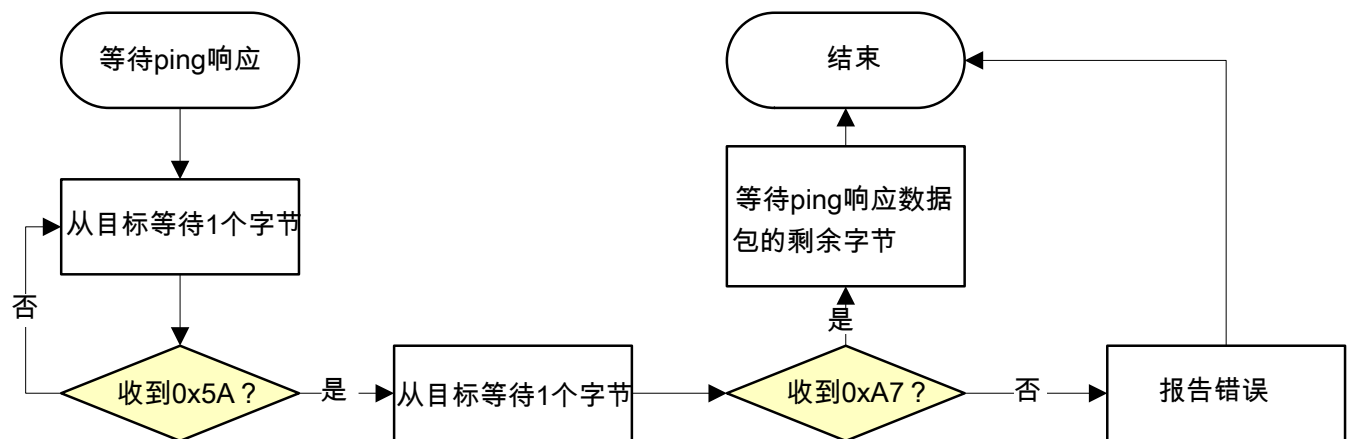


图 14-25. 主机通过 UART 读取来自目标的 ping 响应

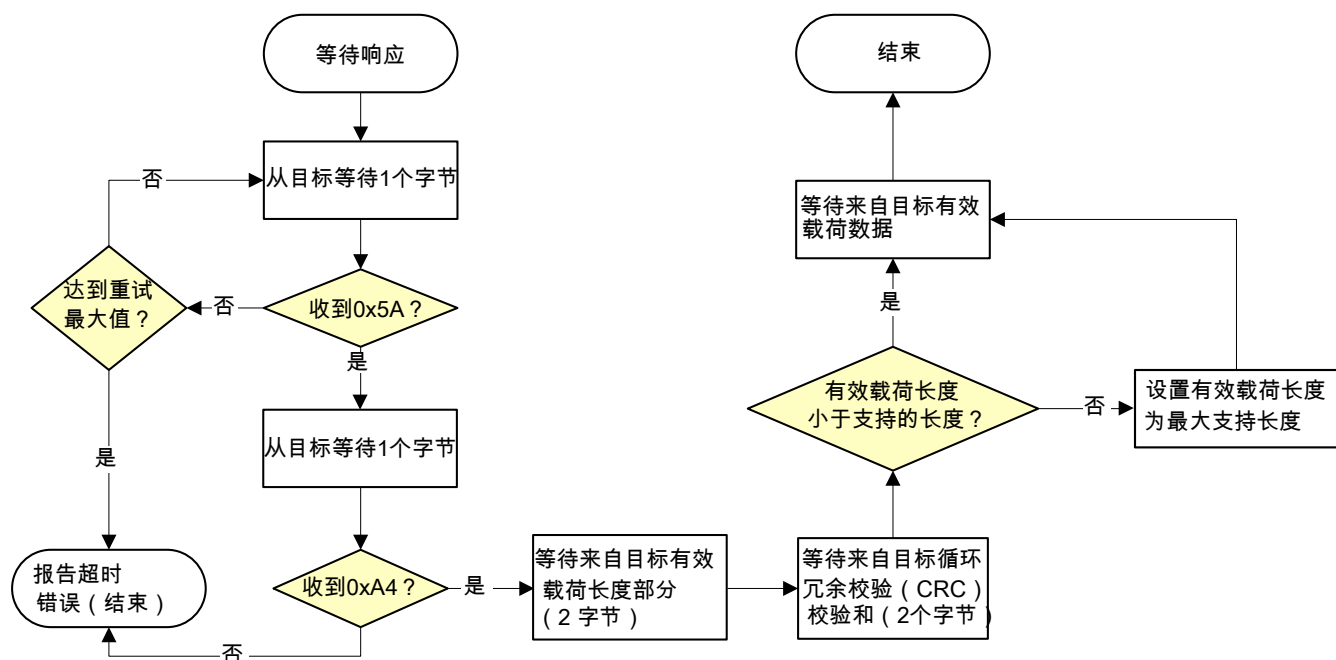


图 14-26. 主机通过 UART 读取来自目标的命令响应

14.4.4 USB 外设

Kinetis Flashloader 支持通过 USB 外设将数据加载到 Flash。目标作为 USB HID 类设备。

USB HID 并不使用帧包；相反，它使用 USB 协议中固有的分包。设备至 NAK 输出传输（在可以接收之前）的功能可提供所需的流量控制；每个 USB 数据包的内置 CRC 均可提供所需的错误检测。

14.4.4.1 时钟配置

Flashloader 支持无晶振 USB 功能。如果 USB 外设已使能，则 Flashloader 将使能 48-MHz IRC。Flashloader 还可以使能 USB 时钟恢复功能（通过将 `USBx_CLK_RECOVER_CTRL[CLOCK_RECOVER_EN]` 设置为 1 并将 `USB_CLK_RECOVER_IRC_EN[IRC_EN]` 设置为 1）。

14.4.4.2 设备描述符

Kinetis Flashloader 可配置默认 USB VID/PID/Strings，如下所示：

默认 VID/PID:

- VID = 0x15A2
- PID = 0x0073

默认 Strings:

- 制造商 [1] = "Freescale Semiconductor Inc." (注意 Freescale Semiconductor 已改为 NXP Semiconductors。)
- 产品 [2] = "Kinetis Bootloader"

14.4.4.3 端点

HID 外设使用以下 3 个端点:

- 控制 (0)
- 中断 IN (1)
- 中断 OUT (2)

中断 OUT 端点对 HID 类器件为可选，但 Kinetis Flashloader 将其用作管道，其中固件可以通过 USB 主机来 NAK 发送请求。

14.4.4.4 HID 报告

Flashloader USB HID 外设已定义并使用 4 种 HID 报告。报告 ID 可确定报告中发送的数据包的方向和类型；否则所有报告的内容都相同。

报告 ID	数据包类型	方向
1	命令	OUT
2	数据	OUT
3	命令	IN
4	数据	IN

对于所有报告，这些属性应用:

使用最小	1
使用最大	1
逻辑最小	0
逻辑最大	255
报告大小	8
报告计数	34

每个报告最大为 34 字节。它从最小 Flashloader 数据包大小 (32 个字节加上 2 个字节的报告标题, 用于指明报告中发送的数据包的长度 (以字节为单位)) 派生而来。

注

在未来, 最大报告大小可能会增大, 以便支持较大数据包的传输。或者, 也可以添加具有较大最大大小的其他报告。

所有报告中发送的实际数据都类似于:

0	报告 ID
1	数据包长度 LSB
2	数据包长度 MSB
3	数据包[0]
4	数据包[1]
5	数据包[2]
	...
N+3-1	数据包[N-1]

此数据包包含报告 ID, 如果在 HID 报告描述符中定义了多个报告, 则需要提供报告 ID。实际发送和接收的数据的最大长度为 35 个字节。数据包长度标题以低字节序格式进行编写, 并且它设置为报告中发送的数据包的大小 (以字节为单位)。此大小不包含报告 ID 或自身的数据包长度标题。在数据阶段中, 数据包大小为 0 表示来自接收器的数据阶段中止请求。

14.4.5 CAN (或 FlexCAN) 外设

Kinetis Flashloader 支持通过 USB 外设将数据加载到 Flash。支持以多种预定义的速度传输至 FlexCAN:

- 125 kHz
- 250 kHz
- 500 kHz
- 1 MHz (默认传输速率)

主机应用程序必须使用受支持 FlexCAN 速度之一。在 Flashloader 中, 它支持在支持速度内的自动速度检测。Flashloader 开始时将以初始速度 (默认速度为 1 MHz) 进入监听模式。主机向特定节点发送 ping 数据包之后, 它将在 FlexCAN 总线上生成流量。由于 Flashloader 处于监听模式, 因此, 它可以通过检测错误来检查本地节点速度是否正确。

- 如果存在错误, 则某些传输的速度可能不正确。
- Flashloader 将更改速度设置并再次检查。

- 如果没有错误，则这意味着传输速度正确，它会将设置更改回正常接收模式，以检查此节点是否存在数据包。
- 主机侧在自动速度检测周期内也应具有合理的时间容差。如果发生超时，则这意味着特定节点无响应，或者存在实际错误（并且它会将错误报告至应用程序）。

以下流程图演示了主机如何读取来自目标的 ping 数据包、ACK 的响应。

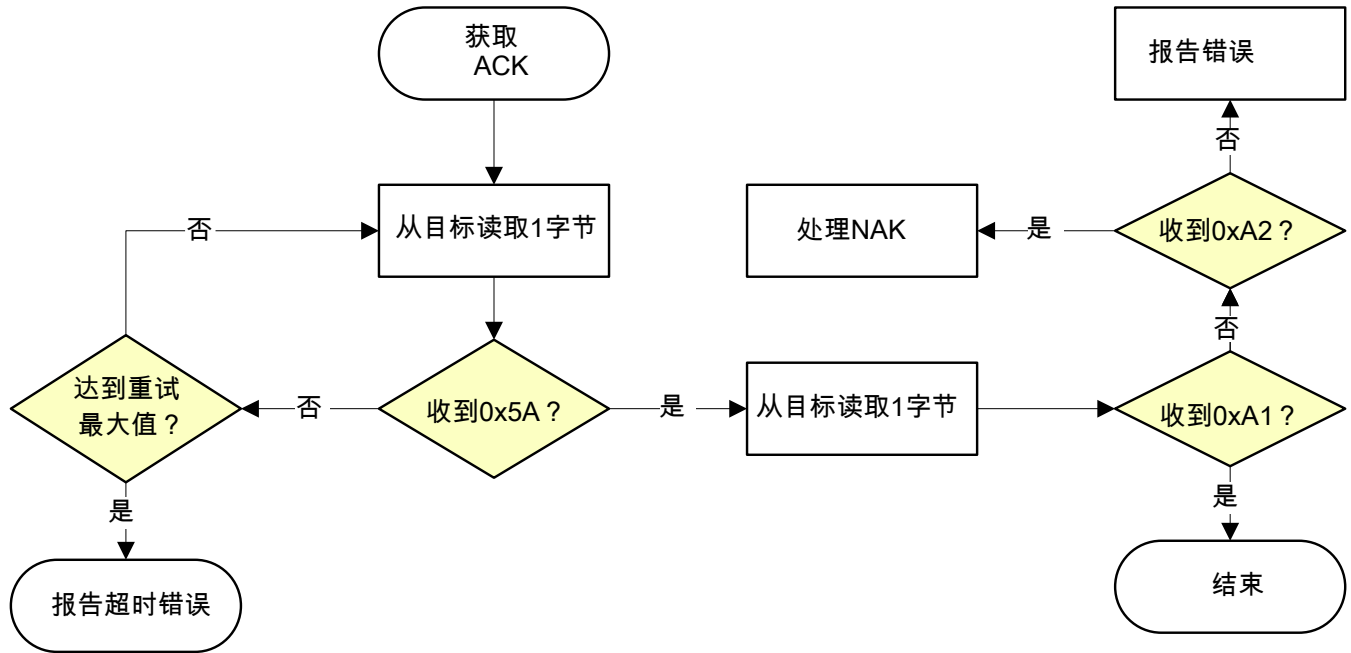


图 14-27. 主机通过 FlexCAN 来读取来自目标的 ACK

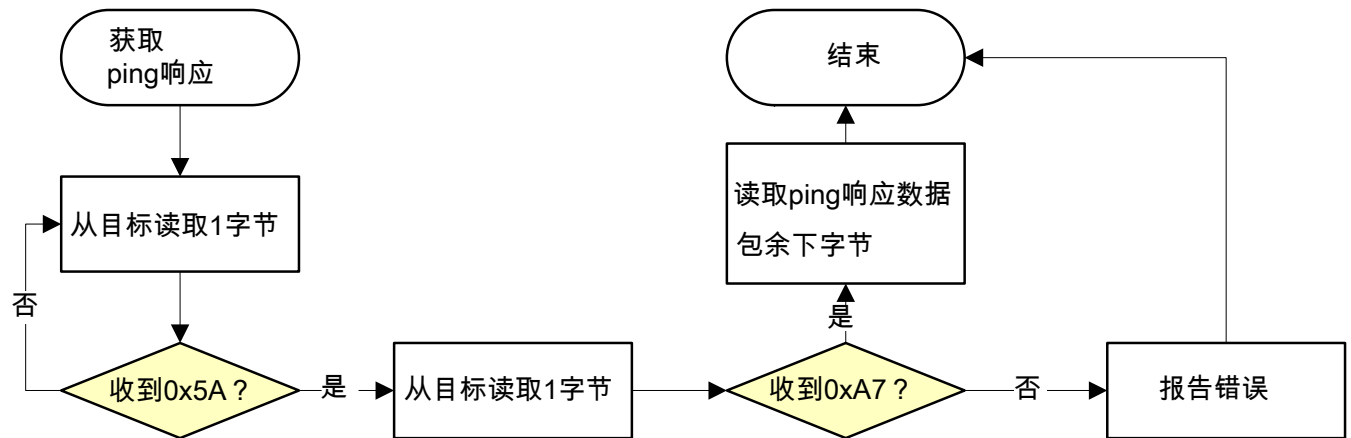


图 14-28. 主机通过 FlexCAN 来读取来自目标的 ping

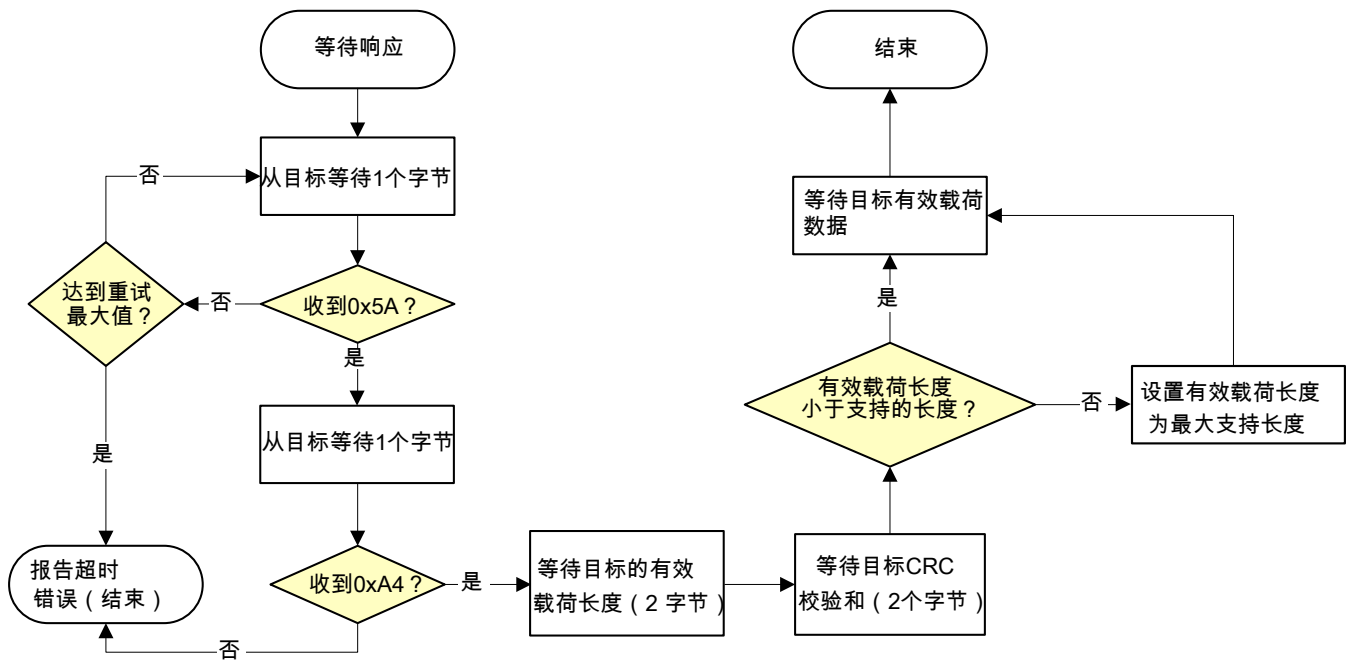


图 14-29. 主机通过 FlexCAN 来读取来自目标的命令响应

14.5 Get/SetProperty 命令属性

本节列出了 GetProperty 和 SetProperty 命令的属性。

表 14-45. Get/SetProperty 命令使用的属性，按值排序

属性	可写入	标签值	大小	描述
CurrentVersion	否	01h	4	当前 Flashloader 版本。
AvailablePeripherals	否	02h	4	此芯片支持的外设集。
FlashStartAddress	否	03h	4	程序 Flash 的起始地址。
FlashSizeInBytes	否	04h	4	程序 Flash 的大小 (字节)。
FlashSectorSize	否	05h	4	程序 Flash 其中一个扇区的大小 (字节)。 这是最小擦除大小。
FlashBlockCount	否	06h	4	Flash 阵列中的块数。
AvailableCommands	否	07h	4	Flashloader 支持的命令集。
VerifyWrites	是	0Ah	4	控制 Flashloader 是否验证对 Flash 的写入操作。 VerifyWrites 功能默认已使能。 0 - 未进行验证。 1 - 使能验证。
MaxPacketSize	否	0Bh	4	当前激活外设接口支持的最大数据包大小。
ReservedRegions	否	0Ch	16	Flashloader 保留的存储器区域列表。返回为值对 (<start-address-of-region>, <end-address-of-region>)。

下一页继续介绍此表...

表 14-45. Get/SetProperty 命令使用的属性，按值排序 (继续)

属性	可写入	标签值	大小	描述
				<ul style="list-style-type: none"> 如果未置位 HasDataPhase 标志，则响应数据包参数计数表示对数。 如果已置位 HasDataPhase 标志，则第二个参数为数据阶段中的字节数。
RAMStartAddress	否	0Eh	4	RAM 区段的起始地址。
RAMSizeInBytes	否	0Fh	4	RAM 区段的大小 (字节)。
SystemDeviceId	否	10h	4	Kinetis 系统设备标识寄存器的值。
FlashSecurityState	否	11h	4	表示是否已使能 Flash 安全。 0 - Flash 安全已禁用。 1 - Flash 安全已使能。
UniqueDeviceId	否	12h	16	唯一器件标识，Kinetis 唯一标识寄存器的值 (K 系列器件为 16, KL 系列器件为 12)。
FacSupport	否	13h	4	FAC (Flash 访问控制) 支持标志 0 - FAC 不受支持。 1 - FAC 受支持。
FlashAccessSegmentSize	否	14h	4	1 个 Flash 区段的大小 (字节)。
FlashAccessSegmentCount	否	15h	4	FAC 区段计数 (Flash 模型内的 Flash 访问区段计数)。
FlashReadMargin	是	16h	4	Flash 擦除和程序验证命令的裕量级别设置。 0 = 正常 1 = 用户 (默认) 2 = 出厂
TargetVersion	否	18h	4	SoC 目标构建版本号。

14.5.1 属性定义

本节提供了 Get/Set 属性定义。

14.5.1.1 CurrentVersion 属性

此属性值为 4 字节结构，包含当前版本的 Flashloader。

表 14-46. CurrentVersion 字段属性

位	31:24	[23:16]	1:58	[7:0]
字段	名称 = 'K' (0x4B)	主要版本	次要版本	Bugfix 版本

14.5.1.2 AvailablePeripherals 属性

此属性值为位域，列出了 Flashloader 及其所运行硬件支持的外设。

表 14-47. 外设位

位	[31:7]	[6]	[5]	[4]	[3]	2	[1]	[0]
外设	保留	保留	保留	USB HID	CAN 从机	SPI 从机	I2C 从机	UART

如果外设可用，则将在属性值中置位对应的位。所有保留位必须置位为 0。

14.5.1.3 AvailableCommands 属性

此属性值为位域，带置位的位，表示 Flashloader 中使能的命令。此位域中只会列出从主机发送至目标的命令。不包括响应命令，例如 GenericResponse。

表示是否存在命令的位号等于命令的标签值减去 1。之所以要减去 1 是因为最低命令标签值为 0x01。要获取给定命令的位掩码，请使用此表达式：

$$\text{mask} = 1 \ll (\text{tag} - 1)$$

表 14-48. 命令位

位	[31:18]	[17]	[16]	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	2	[1]	[0]
命令	Reserved	Reserved	Reserved	FlashReadResource	FlashReadOnce	FlashProgramOnce	Reserved	SetProperty	Reset	Reserved	Execute	ReceiveSBFile	GetProperty	FlashSecurityDisable	FillMemory	WriteMemory	ReadMemory	FlashEraseRegion	FlashEraseAll

14.6 Kinetis Flashloader 状态错误代码

本节介绍了 Kinetis Flashloader 返回至主机的状态错误代码。

表 14-49. Kinetis Flashloader 状态错误代码，按值排序

错误代码	数值	说明
kStatus_Success	0	操作成功，无错误。
kStatus_Fail	1	操作失败，发生通用错误。
kStatus_ReadOnly	2	无法更改请求的值，因为它为只读。
kStatus_OutOfRange	3	请求的值超出范围。
kStatus_InvalidArgument	4	未定义请求的命令参数。
kStatus_Timeout	5	发生超时。
kStatus_FlashSizeError	100	未使用。
kStatus_FlashAlignmentError	101	地址或长度与所要求不匹配。
kStatus_FlashAddressError	102	地址或长度超出可寻址存储器。
kStatus_FlashAccessError	103	FTFA_FSTAT[ACCERR] 位已置位。
kStatus_FlashProtectionViolation	104	FTFA_FSTAT[FPVIOL] 位已置位。
kStatus_FlashCommandFailure	105	FTFA_FSTAT[MGSTAT0] 位已置位。
kStatus_FlashUnknownProperty	106	未知 Flash 属性。
kStatus_FlashEraseKeyError	107	提供的密钥与设定的 Flash 密钥不匹配。
kStatus_FlashRegionExecuteOnly	108	Flash 区域保护为仅执行。
kStatus_I2C_SlaveTxUnderrun	200	I2C 从机 TX 欠载错误。
kStatus_I2C_SlaveRxOverrun	201	I2C 从机 RX 溢出错误。
kStatus_I2C_ArbitrationLost	202	I2C 仲裁丢失错误。
kStatus_SPI_SlaveTxUnderrun	300	SPI 从机 TX 下溢错误。
kStatus_SPI_SlaveRxOverrun	301	SPI 从机 RX 溢出错误。
kStatus_SPI_Timeout	302	SPI 传输超时。
kStatus_SPI_Busy	303	SPI 实例正忙于执行传输。
kStatus_SPI_NoTransferInProgress	304	在未进行任何传输时尝试中止传输。
kStatus_UnknownCommand	10000	未定义请求的命令值。
kStatus_SecurityViolation	10001	不允许执行此命令，因为已使能 Flash 安全。
kStatus_AbortDataPhase	10002	过早中止数据阶段。
kStatusRomLdrSectionOverrun	10100	加载程序已处理完 SB 文件。
kStatusRomLdrSignature	10101	SB 文件的签名不正确。
kStatusRomLdrSectionLength	10102	块中的扇区长度无效。
kStatusRomLdrUnencryptedOnly	10103	已发送加密的 SB 文件且解密支持不可用。
kStatusRomLdrEOFReached	10104	已达到 SB 文件末尾。
kStatusRomLdrChecksum	10105	命令标签块的校验和无效。
kStatusRomLdrCrc32Error	10106	加载命令的 CRC-32 数据不正确。
kStatusRomLdrUnknownCommand	10107	SB 文件中发现未知命令。
kStatusRomLdrIdNotFound	10108	在 SB 文件中未找到可引导的扇区。
kStatusRomLdrDataUnderrun	10109	SB 状态机正等待更多数据。
kStatusRomLdrJumpReturned	10110	SB 文件中的跳转函数已返回。
kStatusRomLdrCallFailed	10111	SB 文件中的调用命令失败。
kStatusRomLdrKeyNotFound	10112	在 SB 文件的密钥目录下找不到匹配的密钥，因此无法解密扇区。

下一页继续介绍此表...

表 14-49. Kinetis Flashloader 状态错误代码，按值排序 (继续)

错误代码	数值	说明
kStatusRomLdrSecureOnly	10113	发送的 SB 文件未解密且已使能目标上的安全。
kStatusRomLdrResetReturned	10114	意外返回 SB 文件复位操作。
kStatusMemoryRangeInvalid	10200	存储器范围与保护区域冲突。
kStatus_UnknownProperty	10300	未定义请求的属性值。
kStatus_ReadOnlyProperty	10301	无法写入请求的属性值。
kStatus_InvalidPropertyValue	10302	指定的属性值无效。
kStatus_AppCrcCheckPassed	10400	CRC 校验有效且已通过。
kStatus_AppCrcCheckFailed	10401	CRC 校验有效但失败。
kStatus_AppCrcCheckInactive	10402	CRC 校验无效。
kStatus_AppCrcCheckInvalid	10403	CRC 校验无效，因为 BCA 无效或未置位 CRC 参数 (所有 0xFF 字节)。
kStatus_AppCrcCheckOutOfRange	10404	CRC 校验有效，但地址超出范围。

第 15 章

复位控制模块 (RCM)

15.1 简介

这些信息用于介绍复位控制模块 (RCM) 的寄存器。RCM 使用多种芯片复位功能。详情请参见芯片复位章节。

有关使用 PMC 的详情，请参见 [AN4503: Power Management for Kinetis MCUs](#)。

15.2 复位存储器映射和寄存器说明

可在此处查看 RCM 存储器映射/寄存器定义。

复位控制模块 (RCM) 寄存器提供复位状态信息查询和复位滤波器控制。

注

RCM 寄存器只可在管理模式写入。在用户模式下进行写入访问将导致总线错误。

RCM 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4007_F000	系统复位状态寄存器 0 (RCM_SRS0)	8	R	82h	15.2.1/268
4007_F001	系统复位状态寄存器 1 (RCM_SRS1)	8	R	00h	15.2.2/269
4007_F004	复位引脚滤波器控制寄存器 (RCM_RPFC)	8	R/W	00h	15.2.3/270
4007_F005	复位引脚滤波器带宽寄存器 (RCM_RPFW)	8	R/W	00h	15.2.4/271
4007_F008	粘滞系统复位状态寄存器 0 (RCM_SSRS0)	8	R/W	82h	15.2.5/273
4007_F009	粘滞系统复位状态寄存器 1 (RCM_SSRS1)	8	R/W	00h	15.2.6/274

15.2.1 系统故为状态寄存器 0 (RCM_SRS0)

此寄存器包含只读状态位，用于表示最近复位源。这些位的复位状态取决于导致 MCU 复位的复位源。

注

此寄存器的复位值取决于复位源：

- POR (包括 LVD) — 0x82
- LVD (不包括 POR) — 0x02
- 由 $\overline{\text{RESET}}$ 引脚有效而引起的 VLLS 唤醒 — 0x41
- 由其他唤醒源引起的 VLLS 唤醒 — 0x01
- 其他复位 — 如果其对应的复位源导致复位, 则会置位该位

地址: 4007_F000h 基准 + 0h 偏移 = 4007_F000h

位	7	6	5	4	3	2	1	0
读	POR	PIN	WDOG	0	LOL	LOC	LVD	WAKEUP
写								
复位	1	0	0	0	0	0	1	0

RCM_SRS0 字段描述

字段	描述
7 POR	<p>上电复位</p> <p>表示复位由上电检测逻辑引起。由于此时内部供电电压是逐渐上升的，因此，低压复位 (LVD) 状态字段也置位，指示已因内部供电电压低于 LVD 阈值而发生复位。</p> <p>0 复位不是由 POR 引起 1 复位是由 POR 引起</p>
6 PIN	<p>外部复位引脚</p> <p>指示复位是由外部 $\overline{\text{reset}}$ 引脚上的有效低电平引起。</p> <p>0 复位不是由外部复位引脚引起 1 复位是由外部复位引脚引起</p>
5 WDOG	<p>看门狗</p> <p>表示复位由看门狗定时器 超时引起。此复位源可通过禁用看门狗来禁止。</p> <p>0 复位不是由看门狗定时溢出引起。 1 复位是由看门狗定时溢出引起。</p>
4 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
3 LOL	<p>失锁复位</p> <p>表示复位由 MCG PLL 中失锁引起。与失锁事件相关的信息，请参见 MCG 描述。</p>

下一页继续介绍此表...

RCM_SRS0 字段描述 (继续)

字段	描述
	0 复位不是由 PLL 中失锁引起。 1 复位由 PLL 中失锁引起
2 LOC	时钟丢失复位 表示复位由外部时钟丢失引起。MCG 时钟监控器必须使能才能检测到时钟丢失。与使能时钟监控器相关的信息，请参阅 MCG 详细信息。 0 复位不是由外部时钟丢失引起。 1 复位由外部时钟丢失引起。
1 LVD	低压检测复位 如果 PMC_LVDSC1[LVDRE] 置位且供电电压降至低于 LVD 跳变电压，则会发生 LVD 复位。该字段也会因 POR 而置位。 0 复位不是由 LVD 跳变或 POR 引起。 1 复位是由 LVD 跳变或 POR 引起。
0 WAKEUP	低漏电唤醒复位 表示复位由芯片处于低漏电模式时已使能的 LLWU 模块唤醒源引起。在 LLS 模式下，RESET 引脚为引起此复位的唯一唤醒源。VLLSx 模式下的任何已使能唤醒源均会导致复位。此位由除 WAKEUP 外的其他复位源清除。 0 复位不是由 LLWU 模块唤醒源引起的。 1 复位是由 LLWU 模块唤醒源引起。

15.2.2 系统复位状态寄存器 1 (RCM_SRS1)

此寄存器包含只读状态位，用于表示最近复位源。这些位的复位状态取决于导致 MCU 复位的复位源。

注

此寄存器的复位值取决于复位源：

- POR (包括 LVD) — 0x00
- LVD (不包括 POR) — 0x00
- VLLS 模式唤醒 — 0x00
- 其他复位 — 如果其对应的复位源导致复位，则会置位该位

地址: 4007_F000h 基准 + 1h 偏移 = 4007_F001h

位	7	6	5	4	3	2	1	0
读	0	0	SACKERR	0	MDM_AP	SW	LOCKUP	JTAG
写								
复位	0	0	0	0	0	0	0	0

RCM_SRS1 字段描述

字段	描述
7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 SACKERR	停止模式应答错误复位 表示尝试进入停止模式后，因一个或多个外设未能在约一秒时间内对此作出应答而引起复位。 0 复位不是因为外设未能对尝试进入停止模式作出应答而引起。 1 复位是因为外设未能对尝试进入停止模式作出应答而引起。
4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 MDM_AP	MDM-AP 系统复位请求 指示复位是由主调试器系统设置 MDM-AP 控制寄存器中的系统复位请求位而引起。 0 复位不是由主调试器设置系统复位请求位而引起。 1 复位是由主调试器系统设置系统复位请求位而引起。
2 SW	软件 指示复位是由软件对 ARM 内核中的应用中断和复位控制寄存器的 SYSRESETREQ 位进行设置而引起。 0 复位不是由软件设置 SYSRESETREQ 位引起。 1 复位是由 SYSRESETREQ 位的软件设置引起。
1 LOCKUP	内核锁定 指示 LOCKUP 事件的 ARM 内核指示已引起复位。 0 复位不是由内核 LOCKUP 事件引起。 1 复位是由内核 LOCKUP 事件引起。
0 JTAG	JTAG 生成的复位 表示复位由 JTAG 选择的某些 IR 指令引起：EXTEST、HIGHZ 和 CLAMP。 0 复位不是由 JTAG 引起。 1 复位是由 JTAG 引起。

15.2.3 复位引脚滤波器控制寄存器 (RCM_RPFC)

注

位 2-0 的复位值仅适用于芯片 POR。它们不受其他复位类型影响。

注

当禁用或进入停止模式时将复位总线时钟滤波器。

地址: 4007_F000h 基准 + 4h 偏移 = 4007_F004h

位	7	6	5	4	3	2	1	0
读	0				RSTFLTSS		RSTFLTSRW	
写	0				0		0	
复位	0	0	0	0	0	0	0	0

RCM_RPFC 字段描述

字段	描述
7-3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 RSTFLTSS	在停止模式下选择复位引脚滤波器 选择在停止和 VLPS 模式下，以及在 LLS 和 VLLS 模式期间，如何复位引脚滤波器。一退出 VLLS 模式，应在清除 PMC_REGSC[ACKISO] 之前配置此位。 0 所有滤波器禁用 1 LPO 时钟滤波器使能
RSTFLTSRW	在运行和等待模式下选择复位引脚滤波器 选择在运行和等待模式下如何使能复位引脚滤波器。 00 所有滤波器禁用 01 使能总线时钟滤波器，以实现正常操作 10 使能 LPO 时钟滤波器，以实现正常操作 11 保留

15.2.4 复位引脚滤波器带宽寄存器 (RCM_RPFW)**注**

RSTFLTSEL 字段中位的复位值仅适用于芯片 POR。它们不受其他复位类型影响。

地址: 4007_F000h 基准 + 5h 偏移 = 4007_F005h

位	7	6	5	4	3	2	1	0
读	0				RSTFLTSEL			
写	0				0			
复位	0	0	0	0	0	0	0	0

RCM_RPFW 字段描述

字段	描述
7-5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
RSTFLTSEL	复位引脚滤波器总线时钟选择 选择复位引脚总线时钟滤波器带宽。

下一页继续介绍此表...

RCM_RPFW 字段描述 (继续)

字段	描述
00000	总线时钟滤波器计数为 1
00001	总线时钟滤波器计数为 2
00010	总线时钟滤波器计数为 3
00011	总线时钟滤波器计数为 4
00100	总线时钟滤波器计数为 5
00101	总线时钟滤波器计数为 6
00110	总线时钟滤波器计数为 7
00111	总线时钟滤波器计数为 8
01000	总线时钟滤波器计数为 9
01001	总线时钟滤波器计数为 10
01010	总线时钟滤波器计数为 11
01011	总线时钟滤波器计数为 12
01100	总线时钟滤波器计数为 13
01101	总线时钟滤波器计数为 14
01110	总线时钟滤波器计数为 15
01111	总线时钟滤波器计数为 16
10000	总线时钟滤波器计数为 17
10001	总线时钟滤波器计数为 18
10010	总线时钟滤波器计数为 19
10011	总线时钟滤波器计数为 20
10100	总线时钟滤波器计数为 21
10101	总线时钟滤波器计数为 22
10110	总线时钟滤波器计数为 23
10111	总线时钟滤波器计数为 24
11000	总线时钟滤波器计数为 25
11001	总线时钟滤波器计数为 26
11010	总线时钟滤波器计数为 27
11011	总线时钟滤波器计数为 28
11100	总线时钟滤波器计数为 29
11101	总线时钟滤波器计数为 30
11110	总线时钟滤波器计数为 31
11111	总线时钟滤波器计数为 32

15.2.5 粘滞系统复位状态寄存器 0 (RCM_SSRS0)

此寄存器包含以指示自上次 POR、LVD 或 VLLS 唤醒之后尚未被软件清除的所有复位源的状态标志。软件可通过将逻辑 1 写入标志来清除状态标志。

地址: 4007_F000h 基准 + 8h 偏移 = 4007_F008h

位	7	6	5	4	3	2	1	0
读	SPOR	SPIN	SWDOG	0	SLOL	SLOC	SLVD	SWAKEUP
写	w1c	w1c	w1c		w1c	w1c	w1c	w1c
复位	1	0	0	0	0	0	1	0

RCM_SSRS0 字段描述

字段	描述
7 SPOR	<p>粘滞上电复位</p> <p>表示复位由上电检测逻辑引起。由于此时内部供电电压是逐渐上升的，因此，低压复位 (LVD) 状态字段也置位，指示已因内部供电电压低于 LVD 阈值而发生复位。</p> <p>0 复位不是由 POR 引起 1 复位是由 POR 引起</p>
6 SPIN	<p>粘滞外部复位引脚</p> <p>指示复位是由外部 $\overline{\text{reset}}$ 引脚上的有效低电平引起。</p> <p>0 复位不是由外部复位引脚引起。 1 复位是由外部复位引脚引起</p>
5 SWDOG	<p>粘滞看门狗</p> <p>表示复位由看门狗定时器超时引起。此复位源可通过禁用看门狗来禁止。</p> <p>0 复位不是由看门狗定时溢出引起。 1 复位是由看门狗定时溢出引起。</p>
4 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
3 SLOL	<p>粘滞失锁复位</p> <p>表示复位由 MCG PLL 中失锁引起。与失锁事件相关的信息，请参见 MCG 描述。</p> <p>0 复位不是由 PLL 中失锁引起。 1 复位由 PLL 中失锁引起。</p>
2 SLOC	<p>粘滞时钟丢失复位。</p> <p>表示复位由外部时钟丢失引起。MCG 时钟监控器必须使能才能检测到时钟丢失。与使能时钟监控器相关的信息，请参阅 MCG 详细信息。</p> <p>0 复位不是由外部时钟丢失引起。 1 复位由外部时钟丢失引起。</p>

下一页继续介绍此表...

RCM_SSRS0 字段描述 (继续)

字段	描述
1 SLVD	<p>粘滞低压检测复位</p> <p>如果 PMC_LVDSC1[LVDRE] 置位且供电电压降至低于 LVD 跳变电压，则会发生 LVD 复位。该字段也会因 POR 而置位。</p> <p>0 复位不是由 LVD 跳变或 POR 引起。 1 复位是由 LVD 跳变或 POR 引起。</p>
0 SWAKEUP	<p>粘滞低漏电唤醒复位</p> <p>表示复位由芯片处于低漏电模式时已使能的 LLWU 模块唤醒源引起。在 LLS 模式下，RESET 引脚为引起此复位的唯一唤醒源。VLLSx 模式下的任何已使能唤醒源均会导致复位。</p> <p>0 复位不是由 LLWU 模块唤醒源引起。 1 复位是由 LLWU 模块唤醒源引起。</p>

15.2.6 粘滞系统复位状态寄存器 1 (RCM_SSRS1)

此寄存器包含以指示自上次 POR、LVD 或 VLLS 唤醒之后尚未被软件清除的所有复位源的状态标志。软件可通过将逻辑 1 写入标志来清除状态标志。

地址: 4007_F000h 基准 + 9h 偏移 = 4007_F009h

位	7	6	5	4	3	2	1	0
读	0	0	SSACKERR	0	SMDM_AP	SSW	SLOCKUP	SJTAG
写			w1c		w1c	w1c	w1c	w1c
复位	0	0	0	0	0	0	0	0

RCM_SSRS1 字段描述

字段	描述
7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 SSACKERR	<p>粘滞停止模式应答错误复位</p> <p>表示尝试进入停止模式后，因一个或多个外设未能在约一秒时间内对此作出应答而引起复位。</p> <p>0 复位不是因为外设未能对尝试进入停止模式作出应答而引起。 1 复位是因为外设未能对尝试进入停止模式作出应答而引起。</p>
4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 SMDM_AP	<p>粘滞 MDM-AP 系统复位请求</p> <p>指示复位是由主调试器系统设置 MDM-AP 控制寄存器中的系统复位请求位而引起。</p>

下一页继续介绍此表...

RCM_SSRS1 字段描述 (继续)

字段	描述
	0 复位不是由主调试器设置系统复位请求位而引起。 1 复位是由主调试器系统设置系统复位请求位而引起。
2 SSW	粘滞软件 指示 ARM 内核中应用程序中断和复位控制寄存器的 SYSRESETREQ 位的软件设置已引起复位。 0 复位不是由软件设置 SYSRESETREQ 位引起 1 复位是由 SYSRESETREQ 位的软件设置引起。
1 SLOCKUP	粘滞内核锁定 指示 LOCKUP 事件的 ARM 内核指示已引起复位。 0 复位不是由内核 LOCKUP 事件引起。 1 复位是由内核 LOCKUP 事件引起。
0 SJTAG	粘滞 JTAG 生成的复位 表示复位由 JTAG 选择的某些 IR 指令引起 : EXTEST、HIGHZ 和 CLAMP。 0 复位不是由 JTAG 引起。 1 复位是由 JTAG 引起

第 16 章

系统模式控制器 (SMC)

16.1 简介

系统模式控制器 (SMC) 负责使系统按顺序进入和退出所有低功耗停止和运行模式。

特别是，它可以监控触发功耗模式转换的事件，同时控制系统电源、时钟和存储器，以达到所需功耗并实现该模式的功能。

本章介绍所有可用低功耗模式、进入/退出各模式的顺序以及各模式中可用的功能。即便是在最低功耗模式下，SMC 也能够运行。

如需进一步了解使用 LLWU 的详情，请参见 [AN4503: Power Management for Kinetis MCUs](#)。

16.2 操作模式

ARM CPU 具有三种主要操作模式：

- 运行
- 睡眠
- 深度睡眠

WFI 或 WFE 指令用于调用睡眠和深度睡眠模式。运行、等待和停止是 Kinetis 微控制器主要操作模式的常用术语。

下表介绍了 ARM CPU 模式与 Kinetis MCU 功耗模式之间的转换。

ARM CPU 模式	MCU 模式
睡眠	等待
深度睡眠	停止

相应地，ARM CPU 文档是指睡眠和深度睡眠，而 Kinetis MCU 文档一般使用等待和停止。

此外，Kinetis MCUs 还以多种方式介绍了停止、等待和运行模式。电源管理控制器 (PMC) 包含运行和停止模式调节器。运行调节用于正常的运行、等待和停止模式。停止模式调节用于所有超低功耗和低漏电模式。在停止模式调节过程中，总线频率在超低功耗模式下受限。

SMC 为用户提供多个功率选择。当不需要最高总线频率以满足应用需求时，超低功耗运行 (VLPR) 工作模式可以显著减少运行时功耗。当有限频率即可满足应用需求时，可以从正常运行模式修改运行模式 (RUNM) 字段，把 MCU 更改为 VLPR 模式。可以从 VLPR 模式，进入相应的等待 (VLPW) 和停止 (VLPS) 模式。

根据用户应用的要求，可以使用多种不同停止模式，它们支持特定逻辑和/或存储器的状态保持、部分降耗或完全降耗。I/O 状态在所有工作模式下保持。用多个寄存器来配置器件的各种操作模式。

下表介绍了该器件可用的功耗模式。

表 16-1. 功耗模式

模式	说明
RUN	MCU 可全速运行且内部电源充分调节，也就是说，处于运行调节状态。此模式也称为正常运行模式。
HSRUN	与 RUN 模式相比，MCU 的运行频率更高且内部电源充分调节。有关最大允许频率的详情，参见“功耗管理”章节。
WAIT	内核时钟被关闭。系统时钟继续运行。总线时钟（如果使能）继续运行。维持运行调节。
STOP	内核时钟被关闭。来自支持外设的所有停止确认信号均有效之后，其他主机的系统时钟以及总线时钟将关闭。
VLPR	内核、系统、总线和 Flash 时钟最大频率在此模式下受限。有关最大允许频率的详情，请参见“功耗管理”章节。
VLPW	内核时钟被关闭。尽管最大频率受限，但系统、总线和 Flash 时钟继续运行。有关最大允许频率的详情，请参见“电源管理”一章。
VLPS	内核时钟被关闭。来自支持外设的所有停止确认信号均有效之后，其他主机的系统时钟以及总线时钟将关闭。
LLS3	内核时钟被关闭。来自支持外设的所有停止确认信号均有效之后，其他主机的系统时钟以及总线时钟将关闭。通过降低内部逻辑的电压，MCU 进入低漏电模式。保留所有系统 RAM 内容，内部逻辑，且保持 I/O 状态。
LLS2	内核时钟被关闭。来自支持外设的所有停止确认信号均有效之后，其他主机的系统时钟以及总线时钟将关闭。通过降低内部逻辑的电压，同时关断系统 RAM2 分区，MCU 进入低漏电模式。保留系统 RAM1 分区内容，内部逻辑，及 I/O 状态。 ¹
VLLS3	内核时钟被关闭。来自支持外设的所有停止确认信号均有效之后，其他主机的系统时钟以及总线时钟将关闭。通过关断内部逻辑，MCU 进入低漏电模式。保留所有系统 RAM 内容且保持 I/O 状态。内部逻辑状态不会保留。
VLLS2	内核时钟被关闭。来自支持外设的所有停止确认信号均有效之后，其他主机的系统时钟以及总线时钟将关闭。通过关断内部逻辑和系统 RAM2 分区，MCU 进入低漏电模式。在此模式下，会保留系统 RAM1 分区内容。内部逻辑状态不会保留。 ¹
VLLS1	内核时钟被关闭。来自支持外设的所有停止确认信号均有效之后，其他主机的系统时钟以及总线时钟将关闭。通过关断内部逻辑和所有系统 RAM，MCU 进入低漏电模式。保持 I/O 状态。内部逻辑状态不会保留。

下一页继续介绍此表...

表 16-1. 功耗模式 (继续)

模式	说明
VLLS0	内核时钟被关闭。来自支持外设的所有停止确认信号均有效之后，其他主机的系统时钟以及总线时钟将关闭。通过关断内部逻辑和所有系统 RAM，MCU 进入低漏电模式。保持 I/O 状态。内部逻辑状态不会保留。1kHz LPO 时钟禁用，并且可使用 STOPCTRL[PORPO] 选择性地使能上电复位 (POR) 电路。

1. 有关系统 RAM 分区的大小和位置，请参见器件芯片配置详细信息。

16.3 存储器映射和寄存器说明

此处提供系统模式控制器相关寄存器的信息。

不同的 SMC 寄存器按不同的复位类型复位。各寄存器的描述提供了细节。有关此芯片复位类型的更多信息，请参见复位部分的详细信息。

注

SMC 寄存器只可在超级用户模式下写入。在用户模式下进行写入访问将导致总线错误。

注

执行 WFI 指令之前，最后写入的寄存器必须回读。这可以确保所有的与设置成低功耗模式相关的寄存器写入均在 MCU 进入低功耗模式之前完成。此操作如果失败则会导致无法正确进入低功耗模式。

SMC 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4007_E000	功耗模式保护寄存器 (SMC_PMPROT)	8	R/W	00h	16.3.1/279
4007_E001	功耗模式控制寄存器 (SMC_PMCTRL)	8	R/W	00h	16.3.2/281
4007_E002	停止控制寄存器 (SMC_STOPCTRL)	8	R/W	03h	16.3.3/282
4007_E003	功耗模式状态寄存器 (SMC_PMSTAT)	8	R	01h	16.3.4/283

16.3.1 功耗模式保护寄存器 (SMC_PMPROT)

该寄存器为进入任何低功耗运行或停止模式提供保护。配置功耗模式控制寄存器 (PMCTRL) 将使能低功耗运行或停止模式。

仅可在任何系统复位之后写入一次 PMPROT 寄存器。

如果 MCU 配置为不允许或保留功耗模式，则 MCU 将继续处于其当前功耗模式。例如，如果 MCU 处于正常的 RUN 模式且 AVLP 为 0，则使用 PMCTRL[RUNM] 尝试进入 VLPR 将受阻并且 PMCTRL[RUNM] 仍为 00b，表示 MCU 仍处于正常 RUN 模式。

注

此寄存器在非 VLLS 芯片复位和触发非 VLLS 芯片复位的复位类型时复位。不触发非 VLLS 型的芯片复位类型对其无影响。请参见复位章节详细内容，了解更多信息。

地址: 4007_E000h 基准 + 0h 偏移 = 4007_E000h

位	7	6	5	4	3	2	1	0
读	AHSRUN	0	AVLP	0	ALLS	0	AVLLS	0
写								
复位	0	0	0	0	0	0	0	0

SMC_PMPROT 字段描述

字段	描述
7 AHSRUN	允许高速运行模式 假设在 PMCTRL 中设置了响应的控制位，则该一次性写入位将允许 MCU 进入高速运行模式 (HSRUN)。 0 不允许 HSRUN。 1 允许 HSRUN。
6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 AVLP	允许极低功耗模式 如果在 PMCTRL 中建立了相应的控制位，则该一次性写入位将允许 MCU 进入任何超低功耗模式 (VLPR、VLPW、和 VLPS)。 0 不允许 VLPR、VLPW 和 VLPS。 1 允许 VLPR、VLPW 和 VLPS。
4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 ALLS	允许低泄漏停止模式 假设在 PMCTRL 中设置了响应的控制位，则该一次性写入位将允许 MCU 进入任何低泄漏停止模式 (LLS)。 0 不允许任何 LLSx 模式。 1 允许任何 LLSx 模式。
2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 AVLLS	允许极低泄漏停止模式 如果在 PMCTRL 中建立了相应的控制位，则该一次性写入位将允许 MCU 进入任何超低漏电停止模式 (VLLSx)。 0 不允许进入任何 VLLSx 模式。 1 允许进入任何 VLLSx 模式。

下一页继续介绍此表...

SMC_PMPROT 字段描述 (继续)

字段	描述
0 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

16.3.2 功耗模式控制寄存器 (SMC_PMCTRL)

如果允许通过相应的保护 (PMPROT) 寄存器设置进入选定功耗模式, 则 PMCTRL 寄存器控制低功耗运行和停止模式的进入。

注

在非 VLLS 芯片 POR 和触发非 VLLS 芯片 POR 的复位类型时复位。不触发非 VLLS 型的芯片 POR 的复位类型对其无影响。请参见复位章节详细内容, 了解更多信息。

地址: 4007_E000h 基准 + 1h 偏移 = 4007_E001h

位	7	6	5	4	3	2	1	0	
读	保留	RUNM			0	STOPA	STOPM		
写	保留	RUNM			0	STOPA	STOPM		
复位	0	0	0	0	0	0	0	0	

SMC_PMCTRL 字段描述

字段	描述
7 Reserved	此字段为保留字段。 保留此位以便未来扩展。软件应在此位写入 0 以维持兼容性。
6-5 RUNM	运行模式控制 写入后, 将导致进入选定的运行模式。如果没有使能 PMPROT 保护寄存器, 则对此字段进行的写操作将受阻。 注: 只有当 PMSTAT=RUN 时, RUNM 应置位为 VLPR。写入 VLPR 后, 不应将 RUNM 写回 RUN, 直到 PMSTAT=VLPR。 注: 可能只有当 PMSTAT=RUN 时, RUNM 才置位为 HSRUN。编程为 HSRUN 之后, 直至 PMSTAT=HSRUN 之前, 不应针对 RUNM 编程使其还原运行。此外, 在 RUNM=HSRUN 或 PMSTAT=HSRUN 时, 也不应尝试进入停止模式。 00 正常运行模式 (RUN) 01 保留 10 超低功耗运行模式 (VLPR) 11 高速运行模式 (HSRUN)
4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 STOPA	停止中止 当置位时, 这个只读状态位表示在前一个停止模式进入的过程中发生了终止, 以防止系统进入该模式。此字段将由复位或硬件在任何停止模式进入序列开始时清零, 并且将在该序列中止时置位。

下一页继续介绍此表...

SMC_PMCTRL 字段描述 (继续)

字段	描述
	0 上一停止模式进入已成功。 1 上一停止模式进入已中止。
STOPM	<p>停止模式控制</p> <p>写入后，当进入立即睡眠或退出时睡眠模式 (SLEEPDEEP=1) 时，控制进入选定停止模式。如果没有使能 PMPROT 保护寄存器，则对此字段进行的写操作将受阻。系统复位之后，硬件将在对 PMPROT 寄存器成功执行写入操作时将此字段清零。</p> <p>注：设为 VLLSx 或 LLSx 时，STOPCTRL 寄存器中的 LLSM 被用于进一步选择拟进入的特定 VLLS 或 LLS 子模式。</p> <p>注：设为 STOP 时，STOPCTRL 寄存器中的 PSTPOPO 可用于选择部分停止模式 (如需要)。</p> <p>000 常规停止模式 (STOP) 001 保留 010 超低功耗停止模式 (VLPS) 011 低漏电停止 (LLSx) 100 超低漏电停止 (VLLSx) 101 保留 110 保留 111 保留</p>

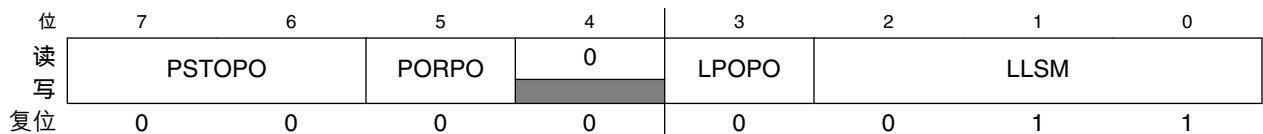
16.3.3 停止控制寄存器 (SMC_STOPCTRL)

The STOPCTRL 寄存器提供各种控制位，允许用户在 STOPM 字段选定的停止模式下微调功耗。

注

在非 VLLS 芯片 POR 和触发非 VLLS 芯片 POR 的复位类型时复位。不触发非 VLLS 型的芯片 POR 的复位类型对其无影响。请参见复位章节详细内容，了解更多信息。

地址: 4007_E000h 基准 + 2h 偏移 = 4007_E002h



SMC_STOPCTRL 字段描述

字段	描述
7-6 PSTOPO	<p>部分停止选项</p> <p>这些位用于控制当 STOPM=STOP 时是否进入部分停止模式。从 RUN (或 VLPR) 模式进入部分停止模式时，PMC、MCG 和 Flash 保持完全通电，从而允许器件以较高功耗实现即刻唤醒。在 PSTOP2 下，仅选通系统时钟，以使总线时钟上运行的外设保持完整功能。在 PSTOP1 模式下，仅选通系统和总线时钟。</p> <p>00 STOP - 正常停止模式</p>

下一页继续介绍此表...

SMC_STOPCTRL 字段描述 (继续)

字段	描述
	01 PSTOP1 - 部分停止，且系统和总线时钟禁用 10 PSTOP2 - 部分停止，且系统时钟禁用但总线时钟使能 11 保留
5 PORPO	POR 电源选项 此位用于控制在 VLLS0 模式下是否使能 POR 检测电路。 0 POR 检测电路在 VLLS0 模式下使能。 1 POR 检测电路在 VLLS0 模式下未使能。
4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 LPOPO	LPO 功耗选项 控制 1 kHz LPO 时钟是否在 LLS/VLLSx 模式中使能。 注：处于 VLLS0 模式期间，此 LPO 时钟被硬件禁用，此位没有影响。 0 LPO 时钟在 LLS/VLLSx 中使能。 1 LPO 时钟在 LLS/VLLSx 中禁用。
LLSM	LLS or VLLS 模式控制 如果 STOPM = LLSx 或 VLLSx，则此字段控制拟进入的 LLS 或 VLLS 子模式。 000 如果 PMCTRL[STOPM]=VLLSx，则为 VLLS0，如果 PMCTRL[STOPM]=LLSx，则保留。 001 如果 PMCTRL[STOPM]=VLLSx，则为 VLLS1，如果 PMCTRL[STOPM]=LLSx，则保留 010 如果 PMCTRL[STOPM]=VLLSx，则为 VLLS2，如果 PMCTRL[STOPM]=LLSx，则为 LLS2 011 如果 PMCTRL[STOPM]=VLLSx，则为 VLLS3，如果 PMCTRL[STOPM]=LLSx，则为 LLS3 100 保留 101 保留 110 保留 111 保留

16.3.4 功耗模式状态寄存器 (SMC_PMSTAT)

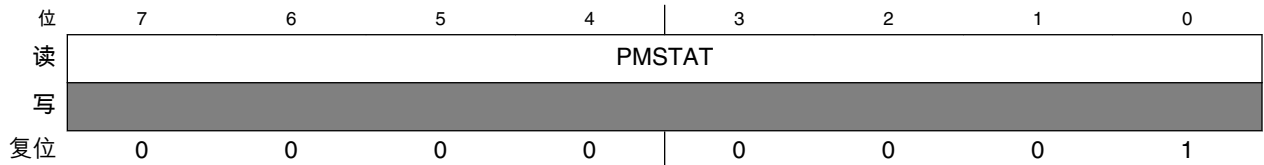
PMSTAT 为只读、热编码寄存器，表示系统的当前功耗模式。

注

在非 VLLS 芯片 POR 和触发非 VLLS 芯片 POR 的复位类型时复位。不触发非 VLLS 型的芯片 POR 的复位类型对其无影响。请参见复位章节详细内容，了解更多信息。

功能说明

地址: 4007_E000h 基准 + 3h 偏移 = 4007_E003h



SMC_PMSTAT 字段描述

字段	描述
PMSTAT	功耗模式状态 注: 使能调试时, PMSTAT 不会更新为 STOP 或 VLPS 注: 使能调试时, PMSTAT 不会更新为 STOP 或 VLPS。 0000_0001 当前功耗模式为 RUN。 0000_0010 当前功耗模式为 STOP。 0000_0100 当前功耗模式为 VLPR。 0000_1000 当前功耗模式为 VLPW。 0001_0000 当前功耗模式为 VLPS。 0010_0000 当前功耗模式为 LLS。 0100_0000 当前功耗模式为 VLLS。 1000_0000 当前功耗模式为 HSRUN。

16.4 功能说明

16.4.1 功耗模式转换

下图所示为此芯片可行的功耗模式状态转换。任何复位都会使 MCU 返回正常 RUN 状态。

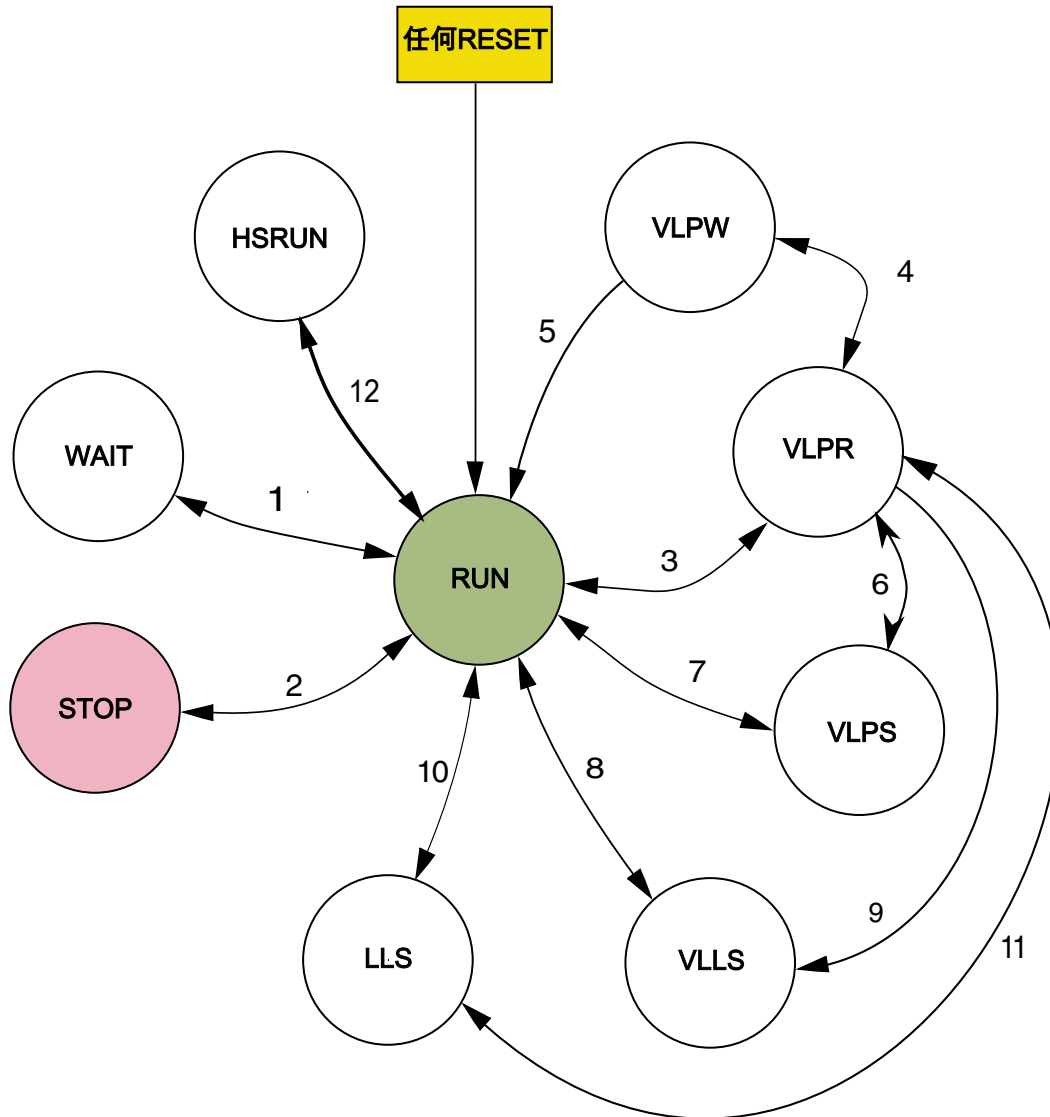


图 16-5. 功耗模式状态示意图

下表定义上图所示的不同状态转换的触发：

表 16-7. 功耗模式转换触发

转换编号	从	到	触发条件
1	RUN	WAIT	立即睡眠模式或退出时睡眠模式可在 SLEEPDEEP 清零时进入，通过 ARM 内核的系统控制寄存器进行控制。 参见注释。
	WAIT	RUN	中断或复位
2	RUN	STOP	PMCTRL[RUNM]=00, PMCTRL[STOPM]=000 ² 立即睡眠模式或退出时睡眠模式可在 SLEEPDEEP 置位时进入，通过 ARM 内核的系统控制寄存器进行控制。 参见注释。 ¹

下一页继续介绍此表...

表 16-7. 功耗模式转换触发 (继续)

转换编号	从	到	触发条件
	STOP	RUN	中断或复位
3	RUN	VLPR	内核、系统、总线和 Flash 时钟频率及 MCG 时钟模式在此模式下受限。有关支持的最大允许频率以及 MCG 模式的详情, 请参见“电源管理”章节。 设置 PMPROT[AVLP]=1, PMCTRL[RUNM]=10。
	VLPR	运行	设置 PMCTRL[RUNM]=00 或复位。
4	VLPR	VLPW	立即睡眠模式或退出时睡眠模式可在 SLEEPDEEP 清零时进入, 通过 ARM 内核的系统控制寄存器进行控制。 参见注释。 ¹
	VLPW	VLPR	
5	VLPW	RUN	复位
6	VLPR	VLPS	PMCTRL[STOPM]=000 ³ 或 010, 立即睡眠模式或退出时睡眠模式可在 SLEEPDEEP 置位时进入, 通过 ARM 内核的系统控制寄存器进行控制。 参见注释。 ¹
	VLPS	VLPR	注: 如果 VLPS 从 RUN 直接进入 (转换编号为 7), 则硬件将强制退回到 RUN, 并不允许转换到 VLPR。
7	RUN	VLPS	PMPROT[AVLP]=1, PMCTRL[STOPM]=010, 立即睡眠模式或退出时睡眠模式可在 SLEEPDEEP 置位时进入, 通过 ARM 内核的系统控制寄存器进行控制。 参见注释。 ¹
	VLPS	RUN	直接从 RUN 或 复位模式进入中断和 VLPS 模式。
8	RUN	VLLSx	PMPROT[AVLLS]=1, PMCTRL[STOPM]=100, STOPCTRL[LLSM]=x (VLLSx), Sleep-now 模式或 sleep-on-exit 模式可在 SLEEPDEEP 置位时进入, 在 ARM 内核的系统控制寄存器中进行控制。
	VLLSx	RUN	从已使能的 LLWU 输入源或复位引脚唤醒。
9	VLPR	VLLSx	PMPROT[AVLLS]=1, PMCTRL[STOPM]=100, STOPCTRL[LLSM]=x (VLLSx), Sleep-now 模式或 sleep-on-exit 模式可在 SLEEPDEEP 置位时进入, 在 ARM 内核的系统控制寄存器中进行控制。
10	RUN	LLSx	PMPROT[ALLS]=1, PMCTRL[STOPM]=011, STOPCTRL[LLSM]=x (LLSx), Sleep-now 模式或 sleep-on-exit 模式可在 SLEEPDEEP 置位时进入, 在 ARM 内核的系统控制寄存器中进行控制。
	LLSx	RUN	从已使能的 LLWU 输入源唤醒且从 RUN 或 RESET 引脚直接进入 LLSx 模式。
11	VLPR	LLSx	PMPROT[ALLS]=1, PMCTRL[STOPM]=011, Sleep-now 模式或 sleep-on-exit 模式可在 SLEEPDEEP 置位时进入, 在 ARM 内核的系统控制寄存器中进行控制。
	LLSx	VLPR	从已使能的 LLWU 输入源唤醒且从 VLPR 直接进入 LLSx 模式。

下一页继续介绍此表...

表 16-7. 功耗模式转换触发 (继续)

转换编号	从	到	触发条件
			注: 如果 LLSx 是从 RUN 直接进入的, 则硬件将不允许这次转换, 并将强制返回 RUN。
12	RUN	HSRUN	设置 PMPROT[AHSRUN]=1, PMCTRL[RUNM]=11。
	HSRUN	RUN	设置 PMCTRL[RUNM]=00 或复位。

1. 如果已使能调试, 则内核时钟保持支持调试。
2. 如果 PMCTRL[STOPM]=000 并且 STOPCTRL[PSTOPO]=01 或 10, 则仅进入部分停止模式, 而非进入 STOP 模式
3. 如果 PMCTRL[STOPM]=000 且 STOPCTRL[PSTOPO]=00, 则会进入 VLPS 模式而非 STOP 模式。如果 PMCTRL[STOPM]=000 且 STOPCTRL[PSTOPO]=01 或 10, 则仅进入部分停止模式, 而非进入 VLPS 模式

16.4.2 功耗模式进入/退出顺序

进入或退出低功耗模式时, 系统必须符合相应的流程, 以安全地管理转换。

SMC 管理系统进入和退出所有功耗模式。此图演示了 SMC 与芯片上其他系统组件之间的连接, 这些组件是所有功耗模式下进行系统排序所必需的。

图 16-6. 低功耗系统组件和连接

16.4.2.1 停止模式进入顺序

CPU 执行 WFI 指令将发起进入低功耗停止模式 (停止、VLPS、LLS、VLLSx) 执行完该指令之后, 将会出现以下序列:

1. CPU 时钟立即关闭。
2. 发出所有非 CPU 总线主机进入停止模式的请求。
3. 所有主机确认已准备好进入停止模式之后, 将会发出所有总线从机进入停止模式的请求。
4. 所有从机确认已准备好进入停止模式之后, 所有系统和总线时钟将关闭。
5. 禁用 MCG 中的时钟产生器。
6. 配置 PMC 中的片上稳压器和内部电源开关, 使其符合目标低功耗模式的功耗目标。

16.4.2.2 停止模式退出顺序

复位或中断事件将引起从低功耗停止模式退出。随后将执行以下序列, 以使系统恢复至运行模式 (RUN 或 VLPR):

1. 恢复 PMC 中的片上稳压器和内部电源开关。
2. MCG 中的时钟产生器使能。

3. 所有主机和从机的系统和总线时钟使能。
4. CPU 时钟使能且 CPU 开始执行用于发起从低功耗停止模式退出的复位或中断。

16.4.2.3 已中止停止模式进入

如果在停止进入序列期间发生了中断，则 SMC 可以提前中止转换并返回 RUN 模式，无需完全进入停止模式。仅当在 PMC 开始停止模式调节转换之前发生中断时才可能发生中止进入。此时之后，中断将被忽略，直到 PMC 完成其进入停止模式调节的转换。当停止模式进入序列发生中止时，SMC_PMCTRL[STOPA] 将置 1。

16.4.2.4 转换至等待模式

对于等待模式 (WAIT 和 VLPW)，CPU 时钟关闭，而其他所有时钟如在 RUN 和 VLPR 模式操作中一样继续运行。某些支持停止等待功能的模块在这些配置下禁用其时钟。

16.4.2.5 从停止模式转换至调试模式

当调试器使能时，调试器模块支持从 STOP、WAIT、VLPS 和 VLPW 转换回中止状态。作为此转换的一部分，系统时钟将被重新建立，且等效于正常的 RUN 和 VLPR 模式时钟配置。

16.4.3 运行模式

此处显示了器件支持的运行模式。

- 运行 (RUN)
- 超低功耗运行 (VLPR)
- 高速运行 (HSRUN)

16.4.3.1 RUN 模式

这是器件的正常工作模式。

此模式将在任意复位后选定。当 ARM 处理器退出复位时，它将置位协议栈、程序计数器 (PC) 和链路寄存器 (LR):

- 处理器从向量表偏移 0x000 读取开始 SP (SP_main)。

- 处理器从向量表偏移 0x004 读取开始 PC。
- LR 设为 0xFFFF_FFFF。

要降低此模式下的功耗，请禁用未使用的模块。

16.4.3.2 超低功耗运行 (VLPR) 模式

在 VLPR 模式下，片上稳压器处于停止模式稳压状态。在此状态下，稳压器提供足够的电流保证 MCU 在低时钟频率下运行。要进一步降低此模式下的功耗，请禁用未使用的模块，方法是使用 SIM 寄存器中其对应的时钟门控控制位。

进入此模式之前，必须满足以下条件：

- 必须将 MCG 配置为在 VLPR 模式下支持的模式。有关这些 MCG 模式的信息，请参见电源管理详细信息。
- 必须禁用 MCG 中的所有时钟监视器。
- 限制系统、总线、Flash 和内核的最大频率。关于支持频率的更多信息，请参见电源管理详细信息。
- 模式保护必须设为允许 VLP 模式，即 PMPROT[AVLP] 为 1。
- PMCTRL[RUNM] 必须设置为 10b 以进入 VLPR。
- 不允许进行 Flash 编程/擦除。

注

处于 VLPR 模式时，请勿提高时钟频率，因为调整器响应缓慢并且无法管理快速负载转换。此外，请勿调整 MCG 模块中的时钟源或任何时钟分频器寄存器。SIM 中的模块时钟使能可以置位，但不能清零。

要重新进入正常 RUN 模式，请清零 PMCTRL[RUNM]。PMSTAT 是一个只读状态寄存器，可用于确定何时系统已完成退出 RUN 模式。当 PMSTAT=RUN 时，系统处于 RUN 调整并且 MCU 可在任何时钟模式下以全速运行。如果需要更高的执行频率，则当从 VLPR 模式返回时，将会轮询 PMSTAT 寄存器，直到其置位为 RUN。

任何复位均会导致退出 VLPR 并在 MCU 退出其复位流程之后使器件返回至 RUN 模式。

16.4.3.3 高速运行 (HSRUN) 模式

在 HSRUN 模式下，片上稳压器保持运行调节状态，但电压输出稍有升高。与正常 RUN 模式相比，在此状态下，MCU 可以更高频率运行。有关最大允许频率的详情，请参见“电源管理”章节。

处于此模式时，必须遵守以下限制：

- 系统、Flash 或内核时钟允许的最大频率变化限制为 2x（两倍频率）
- 退出 HSRUN 模式前，时钟频率应减小到 RUN 模式接收的值。
- HSRUN 不支持进入到停止模式。
- 禁止修改模块的时钟门控位。
- 不允许进行 Flash 编程/擦除。

要进入 HSRUN 模式，设置 $PM\text{PROT}[AHSRUN]=HSRUN$ 且 $PM\text{CTRL}[RUNM]=HSRUN$ 。提高时钟频率前， $PM\text{STAT}$ 寄存器应轮询决定何时系统完成进入 HSRUN 模式。要重新进入正常 RUN 模式，清零 $RUNM$ 。在 MCU 退出其复位流程后，任何复位也将清零 $RUNM$ 并导致系统退出正常 RUN 模式。

16.4.4 等待模式

此器件包括下列两种不同的等待模式。

- 等待
- 超低功耗等待 (VLPW)

16.4.4.1 WAIT 模式

当 ARM 内核进入立即睡眠或 SLEEPDEEP 清零退出睡眠模式时，将进入 WAIT 模式。ARM CPU 进入其未被计时的低功耗状态，但只要外设已使能将继续被计时。通过 SIM 模块使能外设时钟。

发生中断请求时，CPU 退出 WAIT 模式并恢复 RUN 模式下的处理，并从导致中断服务程序的堆栈操作开始。

系统复位将导致退出 WAIT 模式，从而使器件返回正常 RUN 模式。

16.4.4.2 超低功耗等待 (VLPW) 模式

在 MCU 处于 VLPR 模式时，可通过进入立即睡眠或 SLEEPDEEP 清零退出睡眠模式且进入 VLPW。

在 VLPW 模式下，片上稳压器处于停止稳压状态。在此状态下，稳压器提供足够的电流保证 MCU 以较低的频率运行。要进一步降低此模式下的功耗，请禁用未使用的模块的时钟，方法是将 SIM（或 PCC）中外设的对应时钟门控控制位清零。

VLPR 模式限制也适应于 VLPW。

在 VLPW 模式，当一个中断发生时，MCU 将会返回到 VLPR 模式，并且执行中断服务程序。

系统复位将导致退出 VLPW 模式，从而使器件返回正常 RUN 模式。

16.4.5 停止模式

该器件包含多种停止模式，以满足用户的应用需求。

停止模式的范围如下：

- CPU 停止运行，保留所有 I/O、逻辑和存储器状态，特定异步模式外设运行

至：

- CPU 掉电，仅保留 I/O 和较小的寄存器文件，极少异步模式外设运行，同时剩余 MCU 掉电。

停止模式的选择取决于用户应用，并且需要权衡电源使用和状态保留、功能需求和恢复时间。

置位 PMPROT 和 PMCTRL 中正确的字段可选择各种停止模式。当 ARM 内核中的系统控制寄存器的 SLEEPDEEP 位置位时，将在立即睡眠或退出睡眠模式的流程开始时进入选定的停止模式。

可用的停止模式如下：

- 常规停止模式 (STOP)
- 超低功耗停止模式 (VLPS)
- 低漏电停止模式 (LLS)
- 极低漏电停止模式 (VLLSx)

16.4.5.1 STOP 模式

当 ARM 内核中的系统控制寄存器的 SLEEPDEEP 位置位时，可通过立即睡眠或退出睡眠时进入 STOP 模式。

MCG 模块可配置用于使参考时钟保持运行状态。

能够为器件提供异步中断的模块可以使器件退出 STOP 模式并使器件返回至正常的 RUN 模式。对于 STOP 模式下的外设、I/O 和存储器操作，请参见器件的电源管理章节。发生中断请求时，CPU 退出 STOP 模式并恢复处理，并从导致中断服务程序的堆栈操作开始。

系统复位将导致退出 STOP 模式，从而通过 MCU 复位使器件返回正常 RUN 模式。

16.4.5.2 超低功耗停止 (VLPS) 模式

可以进入 VLPS 模式的两种方法如此处所列。

- 当 MCU 处于 VLPR 模式且 PMCTRL 寄存器中的 PMCTRL[STOPM]=010 或 000 时，可通过立即睡眠或退出睡眠时进入停止模式，且 ARM 内核中系统控制寄存器中的 SLEEPDEEP 位置位。
- 当 MCU 处于 RUN 模式且 PMCTRL 寄存器中的 PMCTRL[STOPM]=010 时，可通过立即睡眠或退出睡眠时进入停止模式，且 ARM 内核中系统控制寄存器中的 SLEEPDEEP 位置位。请注意，从 RUN 模式直接进入 VLPS 模式时，硬件将禁用退出 VLPR 模式并且系统将始终退回 RUN 模式。

在 VLPS 模式下，片上稳压器处于和 VLPR 一样的停止稳压状态。

能够为器件提供异步中断的模块可以使器件退出 VLPS 模式并使器件返回至正常 VLPR 模式。

系统复位也会导致退出 VLPS 模式，从而使器件返回正常 RUN 模式。

16.4.5.3 低漏电停止 (LLSx) 模式

器件包含两种低漏电停止模式：LLS3 和 LLS2。本文档中常用的 LLS 或 LLSx 即指这两种模式。可从正常 RUN 或 VLPR 模式进入所有 LLS 模式。

在以下情况下，MCU 将进入 LLS 模式：

- 在立即睡眠或退出睡眠模式时，ARM 内核中系统控制寄存器中的 SLEEPDEEP 将置位；
- 器件按照 [表 16-7](#) 中所示配置。

在 LLS 模式下，片上稳压器处于停止稳压状态。大部分外设处于状态保留模式，在该模式下，它们无法在 LLS 模式下运行。

进入 LLS 模式之前，用户应配置低漏电唤醒 (LLWU) 模块，以使能所需的唤醒源。LLS 中可用的唤醒源如该器件芯片配置详细信息所述。

从 LSS 中唤醒之后，器件将通过挂起的 LLWU 模块中断返回到进入 LLS 时的运行模式（正常 RUN 或 VLPR）。在 LLWU 中断服务程序 (ISR) 中，用户可以轮询 LLWU 模块唤醒标志，以确定唤醒源。

注

LLWU 中断不可通过中断控制器屏蔽，以避免系统无法在 LLS 恢复时完全退出停止模式的情况。

有效的 $\overline{\text{RESET}}$ 引脚信号将导致退出 LLS 模式，使器件返回正常 RUN 模式。当通过 $\overline{\text{RESET}}$ 引脚退出 LLS 时，RCM_SRS[PIN] 和 RCM_SRS[WAKEUP] 置位。

16.4.5.4 超低漏电停止 (VLLSx) 模式

该器件包含三种超低漏电模式：

- VLLS3
- VLLS2
- VLLS1
- VLLS0

本文档中常用的 VLLSx 即指所有这些模式。

可从正常 RUN 或 VLPR 模式进入所有 VLLSx 模式。

在以下情况下，MCU 将进入配置的 LLS 模式：

- 在立即睡眠或退出睡眠模式时，ARM 内核中系统控制寄存器中的 SLEEPDEEP 位将置位；
- 器件按照 [表 16-7](#) 所示配置。

在 VLLS 下，当大部分数字逻辑关断时，片上稳压器处于其停止稳压状态。

进入 VLLS 模式之前，用户应配置低漏电唤醒 (LLWU) 模块，以使能所需的唤醒源。VLLS 中可用的唤醒源如该器件芯片配置详细信息所述。

从 VLLS 中唤醒之后，器件将通过挂起的 LLWU 中断返回至正常 RUN 模式。在 LLWU 中断服务程序 (ISR) 中，用户可以轮询 LLWU 模块唤醒标志，以确定唤醒源。

进入 VLLS 后，执行 VLLS 之前，每个 I/O 引脚将按照配置锁存。由于 MCU 中的所有数字逻辑均已关断，因此，VLLS 下的所有端口和外设数据将丢失。置位 PMC_REGSC[ACKISO]之前，必须恢复此信息。

有效的 $\overline{\text{RESET}}$ 引脚信号将导致退出任何 VLLS 模式，使器件返回正常 RUN 模式。当通过 $\overline{\text{RESET}}$ 引脚退出 VLLS 时，RCM_SRS[PIN] 和 RCM_SRS[WAKEUP] 置位。

16.4.6 低功耗模式中的调试

当 MCU 在加密状态时，器件禁用/限制调试器操作。当 MCU 不在加密状态时，ARM 调试器可以置位两个上电请求信号：

- 系统上电，通过端口控制/开始寄存器中的 SYSPWR
- 调试上电，通过调试端口控制/开始寄存器中的 CDBGPWRUPREQ

在 RUN、WAIT、VLPR 或 VLPW 模式下置位时，模式控制器将驱动每个信号 (CDBGPWRUPACK、CSYSPWRUPACK) 对应的应答。当两个请求均被置位时，模式控制器处理通过进入模拟停止状态进入 STOP 和 VLPS 的尝试。在此模拟停止状态下：

- 调节器处于运行调节状态；
- MCG 生成的时钟源使能；
- 所有系统时钟禁用，除内核时钟；
- 调试模块访问内核寄存器；
- 至片上外设的访问受阻。

当 MCU 处于 LLS 或 VLLS 模式时，无调试可用。LLS 是一种状态保留模式，在从 LLS 唤醒之后，所有调试操作均可继续，即使是在系统因系统复位事件唤醒的情况下。

进入 VLLS 模式将导致所有调试控制和设置关断。为了腾出时间让调试器与 MCU 同步，MDM AP 控制寄存器包括一个超低漏电调试请求 (VLLDBGREQ) 位，用于配置复位控制器逻辑，以使系统在下次从 VLLS 模式恢复时保持复位。此位允许调试器在调试会话继续之前重新初始化调试模块。

MDM AP 控制寄存器还包括一个超低漏电调试应答 (VLLDBGACK) 位，用于释放 VLLS 恢复之后保持复位的 ARM 内核。调试器重新初始化所有调试 IP，然后使能 VLLDBGACK 控制位有效，以允许 RCM 将 ARM 内核从复位中释放并允许 CPU 开始工作。

VLLDBGACK 位通过调试器清零（或者可保持原样），或者可因为下一次 VLLS 恢复产生的复位而自动清零。

第 17 章

电源管理控制器 (PMC)

17.1 简介

电源管理控制器 (PMC) 包含内部稳压器、上电复位 (POR)、低压检测系统 (LVD) 和高压检测系统 ((HVD))。

有关使用 PMC 的详情，请参见 [AN4503: Power Management for Kinetis MCUs](#)。

本器件支持内核电压旁通 (CVB)，旁通期间 PMC 的内部电压稳压器被禁用而内部逻辑由外部调控电源设备供电。

17.2 特性

下文列出了 PMC 具备的特性。

- 内部稳压器；
- 提供掉电检测的有效 POR；
- 支持两个低压跳变点、每个跳变点具有四个警告级别的低压检测；
- 支持两个高压跳变点的高压检测；
- 内核电压旁通，允许外部 PMIC 对期间的内部逻辑进行供电，以加强系统的电源效率。

17.3 低压检测 (LVD) 系统

此器件包含一个用于防止出现低压状况的系统。它会在供电电压波动期间，保护存储器内容和控制 MCU 系统状态。

此系统包含一个上电复位 (POR) 电路，和具有可供用户选择触发电压的 LVD 电路：高电压 (V_{LVDH}) 或低电压 (V_{LVDL})。触发电压根据 LVDS1[LVDV] 进行选择。进入 VLPx、LLS、和 VLLSx 模式后，LVD 禁用。

有两种标志可用于显示低压检测系统的状态:

- 低压状态下的低压检测标志以及以电平灵敏方式操作的控制 1 寄存器 (LVDF)。LVDF 在供电电压降低至低于选择的启动点 (VLVD) 之后置位。向 LVACK 写入 1 可清除 LVDF, 但只是在内部电压回升至高于启动点时如此; 否则, LVDF 保持置位。
- 低压状态下的低压警告标志 (LVWF) 以及以电平灵敏方式操作的控制 2 寄存器 (LVWF)。LVWF 在供电电压降低至低于选定的监控器启动点 (VLVW) 之后置位。向 LVWACK 写入 1 可清除 LVWF, 但只是在内部电压回升至高于启动点时如此; 否则, LVWF 保持置位。

17.3.1 LVD 复位操作

通过设置 LVDR, LVD 在检测到低压条件时会生成一个复位。低压检测阈值由 LVDR 决定。LVD 复位发生之后, LVD 系统会让 MCU 保持复位, 直至供电电压升高至高于此阈值。RCM 模块的 SRS 寄存器的 LVD 字段 (RCM_SRS[LVD]) 继 LVD 或上电复位之后置位。

17.3.2 LVD 中断操作

通过配置中断操作 (LVDR 置位和 LVDR 清除) 的 LVD 电路, LVDF 被置位且在检测到低压条件时发生 LVD 中断请求。通过向 LVACK 写入 1 清除 LVDF。

17.3.3 低压警告 (LVW) 中断操作

LVD 系统的低压检测状态和控制 2 寄存器中包含一个低压警告标志 (LVWF), 用于指出供电电压正接近 (但高于) LVD 电压。LVW 也有一个中断, 通过置位 LVWF 使能该中断。使能后, 若 LVWF 置位, 就会产生 LVW 中断请求。向 LVWACK 写入 1 可将 LVWF 清零。

LVW 选择四个触发电压中的一个:

- 最高: V_{LVW4}
- 两个中间电平: V_{LVW3} 和 V_{LVW2}
- 最低: V_{LVW1}

17.4 高压检测 (HVD) 系统

此器件包含一个用于防止出现高压状况的系统。

此系统包含一个用户可选择触发电压的高压电路：高电压 (V_{HVDH}) 或低电压 (V_{HVDL})。触发电压由 HV DSC1[HV DV] 选择。进入 VLP_x、LLS 和 VLLS_x 模式后，HVD 禁用。

一个标志可用于显示高压检测系统的状态：

- 高压状态下的高压检测标志以及以电平灵敏方式操作的控制 1 寄存器 (HV DSC1[DV DF])。HV DSC1[HV DF] 在供电电压升至高于选择的启动点 (HLVD) 之后置位。向 HV DSC1[HV DACK] 写入 1 可清除 HV DSC1[HV DF]，但只是在内部电压降低至低于启动点时如此；否则，HV DSC1[LV DF] 保持置位。

17.4.1 HVD 复位操作

通过设置 HV DSC1[HV DRE]，HVD 在检测到高压条件时会生成一个复位。高压检测阈值由 HV DSC1[HV DV] 决定。HVD 复位发生之后，HVD 系统会让 MCU 保持复位，直至供电电压降低到低于此阈值。RCM 模块的 SRS 寄存器的 LVD 字段 (RCM_SRS[LVD]) 继 HVD 复位之后设置。

17.4.2 HVD 中断操作

通过配置中断操作 (HV DSC1[HV DIE] 置位和 HV DSC1[HV DRE] 清除) 的 HVD 电路，HV DSC1[HV DF] 被置位且在检测到高压条件时发生 HVD 中断请求。通过向 HV DSC1[HV DACK] 写入 1 来清除 HV DSC1[HV DF]。

17.5 I/O 保留

处于 LLS 模式时，I/O 引脚被置于输入或输出状态。

PMC 一经唤醒即重新使能，完成上电时序之后进入充分调节状态，并从状态保留模式释放逻辑。I/O 在唤醒或复位之后立即释放。如果 LLS 通过 RESET 引脚退出，则 I/O 默认为其复位状态。

处于 VLLS 模式时，I/O 继续保持唤醒事件（由复位事件唤醒的情况除外）状态，直到唤醒经由写入 REGSC[ACKISO] 予以确认。如果 VLLS 通过 RESET 引脚退出，则 I/O 被释放并默认为其复位状态。在这种情况下，无需向 REGSC[ACKISO] 写入。

17.6 存储器映射和寄存器说明

可在此处查看有关 PMC 寄存器的详细信息。

注

PMC 寄存器的不同部分只能通过特定的复位类型进行复位。各寄存器的描述提供了细节。有关此芯片复位类型的更多信息，请参见复位部分的详细信息。

这些寄存器只可在超级用户模式下写入。在用户模式下进行写入访问将导致总线错误。

PMC 存储器映射

绝对地址（十六进制）	寄存器名称	宽度（单位：位）	访问	复位值	小节/页
4007_D000	低压检测状态和控制 1 寄存器 (PMC_LVDSC1)	8	R/W	10h	17.6.1/298
4007_D001	低压检测状态和控制 2 寄存器 (PMC_LVDSC2)	8	R/W	00h	17.6.2/299
4007_D002	稳压器状态和控制寄存器 (PMC_REGSC)	8	R/W	04h	17.6.3/301
4007_D00B	高压检测状态和控制 1 寄存器 (PMC_HVDSC1)	8	R/W	01h	17.6.4/302

17.6.1 低压检测状态和控制 1 寄存器 (PMC_LVDSC1)

本寄存器包含有状态和控制位，以支持低功耗检测功能。即使所需设置与复位设置相同，该寄存器也应当在复位初始化程序运行期间写入数据以设置所需的控制。

当器件处于极低功耗或低漏电模式时，不管 LVDSC1 如何设置，LVD 系统都被禁用。要保护需一直开启 LVD 的系统，请配置 SMC 模块 (SMC_PMPROT) 的功耗模式保护 (PMPROT) 寄存器，禁止使能任何极低功耗或低漏电模式。

请参见器件数据手册，了解实际的 LVD 跳变电压。

注

LVDV 位只会在收到 POR 事件时复位。寄存器的其它位会在收到非 VLLS 芯片复位时复位。有关复位类型的更多信息，请参见复位部分的详细信息。

地址: 4007_D000h 基准 + 0h 偏移 = 4007_D000h

位	7	6	5	4	3	2	1	0
读	LVDF	0	LVDIE	LVDRE	0		LVDV	
写		LVDACK						
复位	0	0	0	1	0	0	0	0

PMC_LVDSC1 字段描述

字段	描述
7 LVDF	<p>低电压检测标志</p> <p>该只读状态字段显示低压检测事件。</p> <p>0 未检测到低压事件。 1 检测到低压事件。</p>
6 LVDACK	<p>低电压检测确认</p> <p>该只写字段用于确认低压检测错误。写 1 可清除 LVDF。读数始终归 0。</p>
5 LVDIE	<p>低压检测中断使能</p> <p>使能 LVDF 的硬件中断请求。</p> <p>0 已禁用硬件中断 (使用轮询)。 1 LVDF = 1 时请求硬件中断。</p>
4 LVDRE	<p>低压检测复位使能</p> <p>该一次性写入位可使能 LVDF 事件以产生硬件复位。忽略其他写入。</p> <p>0 LVDF 不产生硬件复位。 1 LVDF = 1 时, 强制实施 MCU 复位。</p>
3-2 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
LVDV	<p>低压检测电压选择</p> <p>选择 LVD 跳变点电压 (V_{LVD})。</p> <p>00 已选择低跳变点 ($V_{LVD} = V_{LVDL}$) 01 已选择高跳变点 ($V_{LVD} = V_{LVDH}$) 10 保留 11 保留</p>

17.6.2 低压检测状态和控制 2 寄存器 (PMC_LVDSC2)

本寄存器包含有状态和控制位, 以支持低压警告功能。

当器件处于极低功耗或低漏电模式时, 不管 LVDSC2 如何设置, LVD 都被禁用。

请参见器件数据手册, 了解实际的 LVD 跳变电压。

注

LVW 跳变低压取决于 LVWV 和 LVDV。

注

LVWV 只会在收到 POR 事件时复位。寄存器的其他字段会在收到非 VLLS 芯片复位时复位。有关复位类型的更多信息，请参见复位部分的详细信息。

地址: 4007_D000h 基准 + 1h 偏移 = 4007_D001h

位	7	6	5	4	3	2	1	0
读	LVWF	0	LVWIE	0		LVWV		
写		LVWACK						
复位	0	0	0	0	0	0	0	0

PMC_LVDSC2 字段描述

字段	描述
7 LVWF	<p>低压警报标志</p> <p>该只读状态字段显示低压警告事件。当 V_{Supply} 转换低于跳变点或在复位后 V_{Supply} 已低于 V_{LVW}，那么 LVWF 将置位。LVWF 在上电复位后可能是 1，因此要使用 LVW 中断功能，在启用 LVWIE 前必须首先通过写入 LVWACK 来清除 LVWF。</p> <p>0 未检测到低压警告事件。 1 检测到低压警告事件。</p>
6 LVWACK	<p>低压警报应答</p> <p>该只写字段用于确认低压警告错误。写 1 可清除 LVWF。读数始终归 0。</p>
5 LVWIE	<p>低压警报中断使能</p> <p>使能 LVWF 的硬件中断请求。</p> <p>0 已禁用硬件中断 (使用轮询)。 1 LVWF = 1 时请求硬件中断。</p>
4-2 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
LVWV	<p>低压警报电压选择</p> <p>选择 LVW 跳变点电压 (V_{LVW})。实际的警告电压取决于 LVDSC1[LVDV]。</p> <p>00 选择低电平跳变点 ($V_{LVW} = V_{LVW1}$) 01 选择中间电平 1 跳变点 ($V_{LVW} = V_{LVW2}$) 10 选择中间电平 2 跳变点 ($V_{LVW} = V_{LVW3}$) 11 选择高电平跳变点 ($V_{LVW} = V_{LVW4}$)</p>

17.6.3 稳压器状态和控制寄存器 (PMC_REGSC)

该 PMC 包含一个内部稳压器。该稳压器设计采用的带隙参考电压也可以通过一个缓冲器作为一些内部外设 (如 CMP 和 ADC) 的输入。内部调节器提供一个状态位 (REGONS), 该状态位显示调节器正在运行调节。

注

此寄存器在如下情况下复位: 1. 收到由非 VLLS 信号导致的芯片复位时 2. 收到可以触发由非 VLLS 信号导致的芯片复位时有关更多信息, 请参见复位章节详细内容。

地址: 4007_D000h 基准 + 2h 偏移 = 4007_D002h

位	7	6	5	4	3	2	1	0
读	0	0	Reserved	BGEN	ACKISO	REGONS	Reserved	BGBE
写					w1c			
复位	0	0	0	0	0	1	0	0

PMC_REGSC 字段描述

字段	描述
7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 Reserved	此字段为保留字段。
4 BGEN	在 VLPx 操作时带隙使能 BGEN 控制带隙是否在低能耗操作模式 (VLPx, LLS 以及 VLLSx) 下使能。当片上外设需要带隙电压基准处于低功耗操作模式时, 设置 BGEN 以继续使能带隙操作。 注: 当带隙电压基准无需处于低功耗模式时, 清除 BGEN, 避免过度消耗电能。 0 在 VLPx、LLS 和 VLLSx 模式下, 带隙电压基准已被禁用。 1 在 VLPx、LLS 和 VLLSx 模式下, 带隙电压基准已使能。
3 ACKISO	确认隔离 读取此字段可显示某些外设和 I/O 引脚是否因处于 VLLS 模式而处于锁闭状态。设置释放 I/O 引脚和某些外设, 让它们返回正常运行模式状态时, 请往此字段中写入 1。 注: 从 VLLS 模式恢复之后, 用户应该先还原芯片配置, 然后再清除 ACKISO。尤其应该还原已使能的 LLWU 唤醒引脚的引脚配置, 以免在清除 ACKISO 时错误设置 LLWU 标志。 0 外设和 I/O 面板均处于正常运行状态。 1 某些外设和 I/O 面板处于隔离和锁闭状态。
2 REGONS	稳压器处于运行调节状态 该只读字段提供内部稳压器的当前状态。

下一页继续介绍此表...

PMC_REGSC 字段描述 (继续)

字段	描述
	0 调节器处于停止调节状态，或正在向/从其转变。 1 调节器处于运行调节状态。
1 Reserved	此字段为保留字段。 注：该保留为必须保持清零（置位为 0）。
0 BGBE	带隙缓冲区使能 使能带隙基准缓冲区。 0 带隙基准缓冲区未使能。 1 带隙基准缓冲区已使能。

17.6.4 高压检测状态和控制 1 寄存器 (PMC_HVDSC1)

本寄存器包含有状态和控制位，以支持高压检测功能。即使所需设置与复位设置相同，该寄存器也应当在复位初始化程序运行期间写入数据以设置所需的控制。

当器件处于极低功耗或低漏电模式时，不管 HVDSC1 何设置，HVD 系统都被禁用。要保护需一直开启 HVD 的系统，请配置 SMC 模块 (SMC_PMPROT) 的功耗模式保护 (PMPROT) 寄存器，禁止使能任何极低功耗或低漏电模式。

请参见器件数据手册，了解实际的 HVD 跳变电压。

注

该寄存器只会在收到 POR 事件时复位。有关复位类型的更多信息，请参见复位部分的详细信息。

地址: 4007_D000h 基准 + Bh 偏移 = 4007_D00Bh

位	7	6	5	4	3	2	1	0
读	HVDF	0	HVDIE	HVDRE	0			HVDV
写		HVDACK						
复位	0	0	0	0	0	0	0	1

PMC_HVDSC1 字段描述

字段	描述
7 HVDF	高压检测标志 该只读状态字段显示高压检测事件。 0 未检测到高压事件。 1 检测到高压事件。
6 HVDACK	高压检测确认

下一页继续介绍此表...

PMC_HVDSC1 字段描述 (继续)

字段	描述
	该只写字段用于确认高压检测错误。写 1 可清除 HVDF。读数始终归 0。
5 HVDIE	<p>高压检测中断使能</p> <p>使能 HVDF 的硬件中断请求。</p> <p>0 已禁用硬件中断 (使用轮询)。 1 LVWF = 1 时请求硬件中断。</p>
4 HVDRE	<p>高压检测复位使能</p> <p>该一次性写入位可使能 HVDF 事件以产生硬件复位。其它写入会被忽略, 直到下一次芯片复位为止。</p> <p>0 HVDF 不产生硬件复位。 1 HVDF = 1 时, 强制实施 MCU 复位</p>
3-1 保留	<p>此字段为保留字段。</p> <p>此只读字段为保留字段且值始终为 0。</p>
0 HVDV	<p>高压检测电压选择</p> <p>选择 HVD 跳变点电压 (V_{HVD})。</p> <p>0 已选择低跳变点 ($V_{HVD} = V_{HVDL}$) 1 已选择高跳变点 ($V_{HVD} = V_{HVDH}$)</p>

第 18 章

低漏电唤醒单元 (LLWU)

18.1 此模块的芯片实现细节

18.1.1 唤醒源

该器件使用以下内部外设和外部引脚输入作为 LLWU 模块的唤醒源。LLWU_Px 为外部引脚输入，LLWU_M0IF-M7IF 为至内部外设中断标志的连接。

注

除了 LLWU 唤醒源之外，当 NMI 或 RESET 引脚已使能且相应引脚已置位时，该器件还可从低功耗模式唤醒。

表 18-1. LLWU 输入的唤醒源

输入	唤醒源
LLWU_P0	PTE1/LLWU_P0 引脚
LLWU_P1	PTE2/LLWU_P1 引脚
LLWU_P2	PTE4/LLWU_P2 引脚
LLWU_P3	PTA4/LLWU_P3 引脚 ¹
LLWU_P4	PTA13/LLWU_P4 引脚
LLWU_P5	PTB0/LLWU_P5 引脚
LLWU_P6	PTC1/LLWU_P6 引脚
LLWU_P7	PTC3/LLWU_P7 引脚
LLWU_P8	PTC4/LLWU_P8 引脚
LLWU_P9	PTC5/LLWU_P9 引脚
LLWU_P10	PTC6/LLWU_P10 引脚
LLWU_P11	PTC11/LLWU_P11 引脚
LLWU_P12	PTD0/LLWU_P12 引脚
LLWU_P13	PTD2/LLWU_P13 引脚
LLWU_P14	PTD4/LLWU_P14 引脚

下一页继续介绍此表...

表 18-1. LLWU 输入的唤醒源 (继续)

输入	唤醒源
LLWU_P15	PTD6/LLWU_P15 引脚
LLWU_P16	保留
LLWU_P17	保留
LLWU_P18	保留
LLWU_P19	保留
LLWU_P20	保留
LLWU_P21	保留
LLWU_P22	保留
LLWU_P23	保留
LLWU_P24	保留
LLWU_P25	保留
LLWU_P26	USBVDD
LLWU_P27	USB0_DP
LLWU_P28	USB0_DM ²
LLWU_P29	保留
LLWU_P30	保留
LLWU_P31	保留
LLWU_M0IF	LPTMR
LLWU_M1IF	CMP0
LLWU_M2IF	保留
LLWU_M3IF	保留
LLWU_M4IF	保留
LLWU_M5IF	RTC 警报 ³
LLWU_M6IF	保留
LLWU_M7IF	RTC 秒数 ³

1. 如果在进入 LLS/VLLS 时使能了 NMI，则置位 NMI 引脚会在退出低功耗模式时产生 NMI 中断。也可以通过 FOPT[NMI_DIS] 位禁用 NMI。
2. 作为 LLWU 的唤醒源，USB0_DP 和 USB0_DM 仅在芯片处于 USB 主机模式时才可用。
3. 需要使能外设和外设中断。LLWU 的 WUME 位可使能内部模块标志作为唤醒输入。唤醒后，会根据外设清除机制清除这些标志。

18.2 简介

LLWU 模式允许用户选择多达 32 个外部引脚和多达 8 个内部模块作为低漏电功耗模式的中断唤醒源。

输入源如器件的芯片配置详情所述。可单独使能每个可用唤醒源。

$\overline{\text{RESET}}$ 引脚是一个额外的触发源，可使器件退出低漏电功耗模式，并导致 MCU 通过复位流退出 LLS 和 VLLS 模式。

LLWU 模块还包含 4 个用于外部唤醒引脚的可选数字引脚滤波器。

如需进一步了解有关使用 LLWU 的详情，请参见 [AN4503: Power Management for Kinetis MCUs](#)。

18.2.1 特性

LLWU 模块特性包括：

- 低漏电模式支持多达 32 个外部输入引脚和多达 8 个内部模块（配有单独的 MCU 中断使能位）。
- 输入源可能是外部引脚，或者来自可以运行于 LLS 或 VLLS 模式下的内部外设。请参见芯片配置信息，了解此器件的唤醒输入源。
- 外部引脚唤醒输入，每个都能编程为下降沿、上升沿或任何变化。
- 在 MCU 之后激活的唤醒输入进入低泄漏功耗模式。
- 提供可选数字滤波器，以实现外部引脚检测。请注意在禁用 LPO 时钟后，滤波器会被禁用和旁通。

18.2.2 操作模式

LLWU 模块仅在进入低漏电功耗模式时有效。从 LLS 恢复以后，LLWU 立即禁用。从 VLLS 恢复以后，LLWU 继续检测唤醒事件，直到用户通过向 PMC_REGSC[ACKISO] 的一个写入操作确认唤醒为止。

18.2.2.1 LLS 模式

外部引脚输入 (LLWU_Px) 和内部模块中断输入 (LLWU_MxIF) 导致的唤醒事件在退出 LLS 时产生中断。

注

LLWU 中断不可通过中断控制器屏蔽，以避免系统无法在 LLS 恢复时完全退出停止模式的现象。

18.2.2.2 VLLS 模式

所有唤醒和复位事件将导致 VLLS 模式退出，并执行复位流程。

18.2.2.3 非低漏电模式

LLWU 在所有非低漏电模式下处于休眠状态，其中，检测和控制逻辑处于静态。LLWU 寄存器可在非低漏电模式下访问，并且可以进行读写。

当唤醒引脚滤波器使能时，滤波器将立即开始工作。如果在有效边沿的 5 个 LPO 时钟周期内进入低漏电模式，则 LLWU 将检测到边沿事件。

18.2.2.4 调试模式

当芯片处于调试模式，然后进入 LLS 或 VLLS_x 模式时，所有的调试逻辑无法在低漏电模式下工作。退出 LLS 或 VLLS_x 模式时，LLWU 会进入休眠状态。

18.2.3 结构框图

下图为 LLWU 模块的框图。

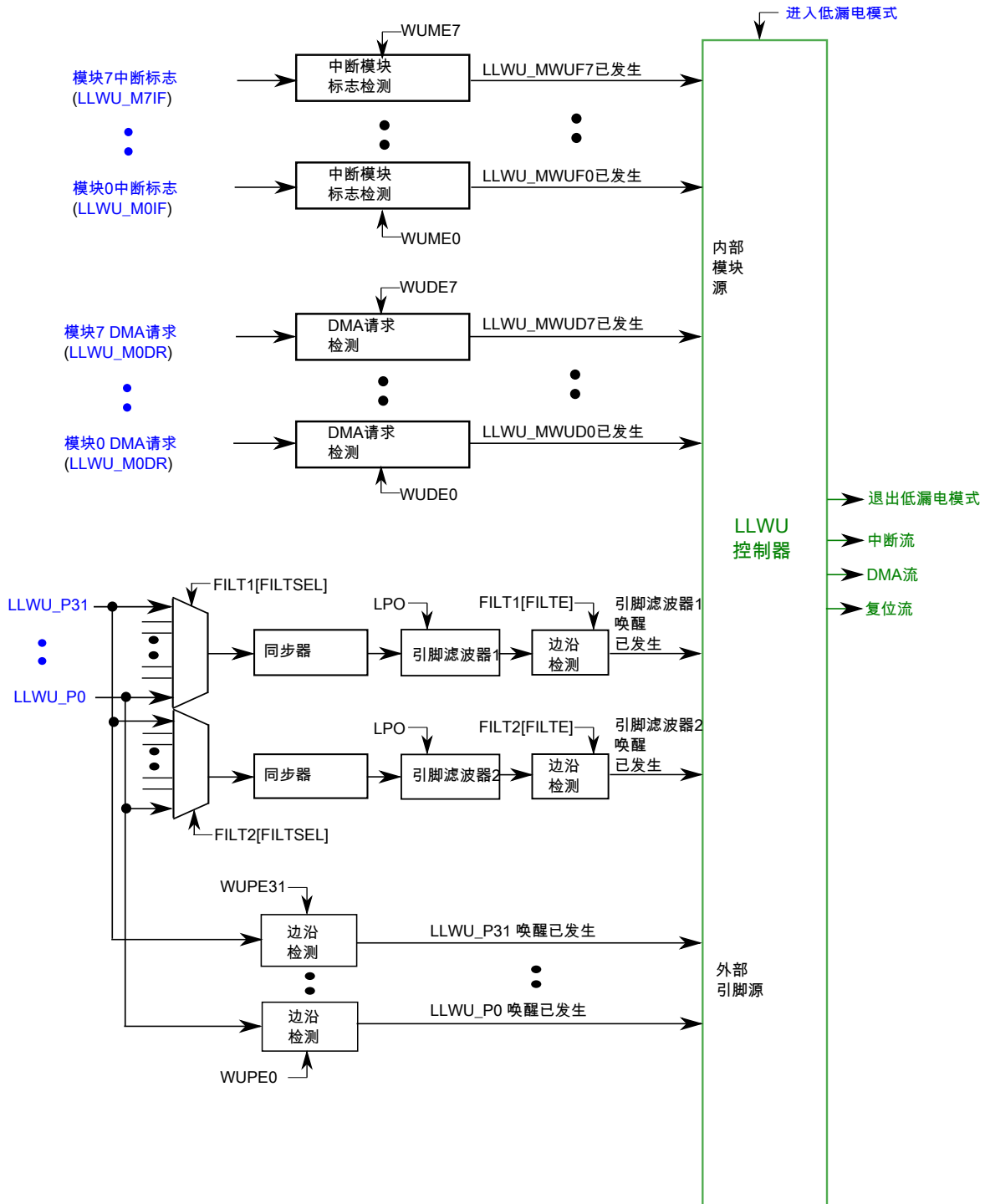


图 18-1. LLWU 框图

18.3 LLWU 信号说明

LLWU 的信号属性如下表所示。

可启用外部唤醒引脚以检测上升沿、下降沿或任何变化。

表 18-2. LLWU 信号说明

信号	说明	I/O
LLWU_Pn	唤醒输入(n = 0- 31)	I

18.4 存储器映射/寄存器定义

LLWU 包括以下寄存器：

- 唤醒源使能寄存器
 - 使能外部引脚输入源
 - 使能内部外设中断源
- 唤醒标志寄存器
 - 表示导致退出低漏电功耗模式的唤醒源包括外部引脚或内部模块中断
- 唤醒引脚滤波器使能寄存器

注

LLWU 寄存器只可在管理模式写入。在用户模式下进行写入访问将导致总线错误。

所有 LLWU 寄存器由芯片复位，而非 VLLS 复位，且由触发芯片复位而非 VLLS 的复位类型复位。不触发非 VLLS 类型的芯片复位对 LLWU 寄存器无影响。有关此芯片复位类型的更多信息，请参见[简介](#) 详细信息。

LLWU 存储器映射

绝对地址（十六进制）	寄存器名称	宽度（单位：位）	访问	复位值	小节/页
4007_C000	LLWU 引脚使能 1 寄存器 (LLWU_PE1)	8	R/W	00h	18.4.1/311
4007_C001	LLWU 引脚使能 2 寄存器 (LLWU_PE2)	8	R/W	00h	18.4.2/312
4007_C002	LLWU 引脚使能 3 寄存器 (LLWU_PE3)	8	R/W	00h	18.4.3/313
4007_C003	LLWU 引脚使能 4 寄存器 (LLWU_PE4)	8	R/W	00h	18.4.4/314
4007_C004	LLWU 引脚使能 5 寄存器 (LLWU_PE5)	8	R/W	00h	18.4.5/315
4007_C005	LLWU 引脚使能 6 寄存器 (LLWU_PE6)	8	R/W	00h	18.4.6/317
4007_C006	LLWU 引脚使能 7 寄存器 (LLWU_PE7)	8	R/W	00h	18.4.7/318
4007_C007	LLWU 引脚使能 8 寄存器 (LLWU_PE8)	8	R/W	00h	18.4.8/319
4007_C008	LLWU 模块使能寄存器 (LLWU_ME)	8	R/W	00h	18.4.9/320
4007_C009	LLWU 引脚标志 1 寄存器 (LLWU_PF1)	8	R/W	00h	18.4.10/322

下一页继续介绍此表...

LLWU 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4007_C00A	LLWU 引脚标志 2 寄存器 (LLWU_PF2)	8	R/W	00h	18.4.11/323
4007_C00B	LLWU 引脚标志 3 寄存器 (LLWU_PF3)	8	R/W	00h	18.4.12/325
4007_C00C	LLWU 引脚标志 4 寄存器 (LLWU_PF4)	8	R/W	00h	18.4.13/326
4007_C00D	LLWU 模块标志 5 寄存器 (LLWU_MF5)	8	R	00h	18.4.14/328
4007_C00E	LLWU 引脚滤波器 1 寄存器 (LLWU_FILT1)	8	R/W	00h	18.4.15/330
4007_C00F	LLWU 引脚滤波器 2 寄存器 (LLWU_FILT2)	8	R/W	00h	18.4.16/331

18.4.1 LLWU 引脚使能 1 寄存器 (LLWU_PE1)

LLWU_PE1 包含用于使能和选择外部唤醒输入引脚 LLWU_P3-LLWU_P0 边沿检测类型的字段。

注

此寄存器在如下情况下复位:

1. 收到由非 VLLS 信号导致的芯片复位时;
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + 0h 偏移 = 4007_C000h

位	7	6	5	4	3	2	1	0
读	WUPE3		WUPE2		WUPE1		WUPE0	
写								
复位	0	0	0	0	0	0	0	0

LLWU_PE1 字段描述

字段	描述
7-6 WUPE3	LLWU_P3 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
5-4 WUPE2	LLWU_P2 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入

下一页继续介绍此表...

LLWU_PE1 字段描述 (继续)

字段	描述
	01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
3-2 WUPE1	LLWU_P1 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
WUPE0	LLWU_P0 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测

18.4.2 LLWU 引脚使能 2 寄存器 (LLWU_PE2)

LLWU_PE2 包含用于使能和选择外部唤醒输入引脚 LLWU_P7-LLWU_P4 边沿检测类型的字段。

注

此寄存器在如下情况下复位:

1. 收到由非 VLLS 信号导致的芯片复位时;
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + 1h 偏移 = 4007_C001h

位	7	6	5	4	3	2	1	0
读	WUPE7		WUPE6		WUPE5		WUPE4	
写	0		0		0		0	
复位	0		0		0		0	

LLWU_PE2 字段描述

字段	描述
7-6 WUPE7	LLWU_P7 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。

下一页继续介绍此表...

LLWU_PE2 字段描述 (继续)

字段	描述
	00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
5-4 WUPE6	LLWU_P6 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
3-2 WUPE5	LLWU_P5 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
WUPE4	LLWU_P4 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测

18.4.3 LLWU 引脚使能 3 寄存器 (LLWU_PE3)

LLWU_PE3 包含用于使能和选择外部唤醒输入引脚 LLWU_P11-LLWU_P8 边沿检测类型的信息。

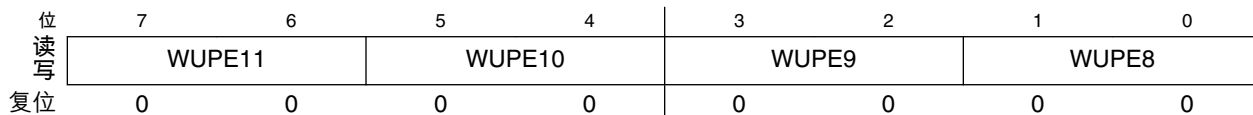
注

此寄存器在如下情况下复位:

1. 收到由非 VLLS 信号导致的芯片复位时;
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + 2h 偏移 = 4007_C002h



LLWU_PE3 字段描述

字段	描述
7-6 WUPE11	LLWU_P11 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
5-4 WUPE10	LLWU_P10 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
3-2 WUPE9	LLWU_P9 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
WUPE8	LLWU_P8 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测

18.4.4 LLWU 引脚使能 4 寄存器 (LLWU_PE4)

LLWU_PE4 包含用于使能和选择外部唤醒输入引脚 LLWU_P15-LLWU_P12 边沿检测类型的字段。

注

此寄存器在如下情况下复位:

1. 收到由非 VLLS 信号导致的芯片复位时;

2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + 3h 偏移 = 4007_C003h

位	7	6	5	4	3	2	1	0
读写	WUPE15		WUPE14		WUPE13		WUPE12	
复位	0	0	0	0	0	0	0	0

LLWU_PE4 字段描述

字段	描述
7-6 WUPE15	LLWU_P15 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
5-4 WUPE14	LLWU_P14 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
3-2 WUPE13	LLWU_P13 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
WUPE12	LLWU_P12 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测

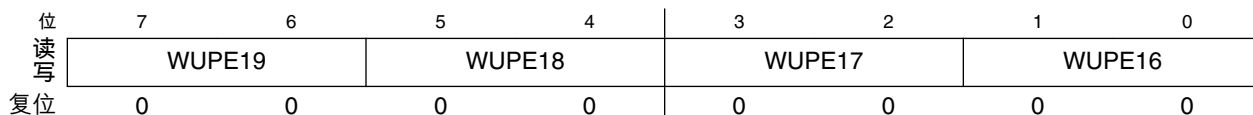
18.4.5 LLWU 引脚使能 5 寄存器 (LLWU_PE5)

LLWU_PE5 包含用于使能和选择外部唤醒输入引脚 LLWU_P19-LLWU_P16 边沿检测类型的字段。

注

此寄存器在非 VLLS 芯片复位和触发非 VLLS 芯片复位的复位类型时复位。不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + 4h 偏移 = 4007_C004h



LLWU_PE5 字段描述

字段	描述
7-6 WUPE19	LLWU_P19 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
5-4 WUPE18	LLWU_P18 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
3-2 WUPE17	LLWU_P17 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
WUPE16	LLWU_P16 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测

18.4.6 LLWU 引脚使能 6 寄存器 (LLWU_PE6)

LLWU_PE6 包含用于使能和选择外部唤醒输入引脚 LLWU_P23-LLWU_P20 边沿检测类型的字段。

注

此寄存器在如下情况下复位:

1. 收到由非 VLLS 信号导致的芯片复位时;
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + 5h 偏移 = 4007_C005h

位	7	6	5	4	3	2	1	0
读	WUPE23		WUPE22		WUPE21		WUPE20	
写								
复位	0	0	0	0	0	0	0	0

LLWU_PE6 字段描述

字段	描述
7-6 WUPE23	LLWU_P23 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
5-4 WUPE22	LLWU_P22 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
3-2 WUPE21	LLWU_P21 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
WUPE20	LLWU_P20 的唤醒引脚使能

下一页继续介绍此表...

LLWU_PE6 字段描述 (继续)

字段	描述
	使能和配置唤醒引脚的边沿检测。
00	外部输入引脚禁用作为唤醒输入
01	使能外部输入引脚的上升沿检测
10	使能外部输入引脚的下降沿检测
11	使能外部输入引脚的任何变化检测

18.4.7 LLWU 引脚使能 7 寄存器 (LLWU_PE7)

LLWU_PE7 包含用于使能和选择外部唤醒输入引脚 LLWU_P27-LLWU_P24 边沿检测类型的字段。

注

此寄存器在如下情况下复位:

1. 收到由非 VLLS 信号导致的芯片复位时;
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + 6h 偏移 = 4007_C006h

位 读 写	7	6	5	4	3	2	1	0
	WUPE27		WUPE26		WUPE25		WUPE24	
复位	0	0	0	0	0	0	0	0

LLWU_PE7 字段描述

字段	描述
7-6 WUPE27	LLWU_P27 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
5-4 WUPE26	LLWU_P26 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测

下一页继续介绍此表...

LLWU_PE7 字段描述 (继续)

字段	描述
	10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
3-2 WUPE25	LLWU_P25 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
WUPE24	LLWU_P24 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测

18.4.8 LLWU 引脚使能 8 寄存器 (LLWU_PE8)

LLWU_PE8 包含用于使能和选择外部唤醒输入引脚 LLWU_P31-LLWU_P28 边沿检测类型的字段。

注

此寄存器在如下情况下复位:

1. 收到由非 VLLS 信号导致的芯片复位时;
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + 7h 偏移 = 4007_C007h

位	7	6	5	4	3	2	1	0
读	WUPE31		WUPE30		WUPE29		WUPE28	
写	0		0		0		0	
复位	0		0		0		0	

LLWU_PE8 字段描述

字段	描述
7-6 WUPE31	LLWU_P31 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。

下一页继续介绍此表...

LLWU_PE8 字段描述 (继续)

字段	描述
	00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
5-4 WUPE30	LLWU_P30 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
3-2 WUPE29	LLWU_P29 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测
WUPE28	LLWU_P28 的唤醒引脚使能 使能和配置唤醒引脚的边沿检测。 00 外部输入引脚禁用作为唤醒输入 01 使能外部输入引脚的上升沿检测 10 使能外部输入引脚的下降沿检测 11 使能外部输入引脚的任何变化检测

18.4.9 LLWU 模块使能寄存器 (LLWU_ME)

LLWU_ME 包含用于将内部模块标志使能为输入 MWUF7-MWUF0 的唤醒输入源的位。

注

此寄存器在如下情况下复位:

1. 收到由非 VLLS 信号导致的芯片复位时;
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + 8h 偏移 = 4007_C008h

位	7	6	5	4	3	2	1	0
读	WUME7	WUME6	WUME5	WUME4	WUME3	WUME2	WUME1	WUME0
写								
复位	0	0	0	0	0	0	0	0

LLWU_ME 字段描述

字段	描述
7 WUME7	<p>模块 7 的唤醒模块使能</p> <p>将内部模块使能为唤醒源输入。</p> <p>0 内部模块标志未用作唤醒源 1 内部模块标志用作唤醒源</p>
6 WUME6	<p>模块 6 的唤醒模块使能</p> <p>将内部模块使能为唤醒源输入。</p> <p>0 内部模块标志未用作唤醒源 1 内部模块标志用作唤醒源</p>
5 WUME5	<p>模块 5 的唤醒模块使能</p> <p>将内部模块使能为唤醒源输入。</p> <p>0 内部模块标志未用作唤醒源 1 内部模块标志用作唤醒源</p>
4 WUME4	<p>模块 4 的唤醒模块使能</p> <p>将内部模块使能为唤醒源输入。</p> <p>0 内部模块标志未用作唤醒源 1 内部模块标志用作唤醒源</p>
3 WUME3	<p>模块 3 的唤醒模块使能</p> <p>将内部模块使能为唤醒源输入。</p> <p>0 内部模块标志未用作唤醒源 1 内部模块标志用作唤醒源</p>
2 WUME2	<p>模块 2 的唤醒模块使能</p> <p>将内部模块使能为唤醒源输入。</p> <p>0 内部模块标志未用作唤醒源 1 内部模块标志用作唤醒源</p>
1 WUME1	<p>模块 1 的唤醒模块使能</p> <p>将内部模块使能为唤醒源输入。</p> <p>0 内部模块标志未用作唤醒源 1 内部模块标志用作唤醒源</p>
0 WUME0	<p>模块 0 的唤醒模块使能</p> <p>将内部模块使能为唤醒源输入。</p>

下一页继续介绍此表...

LLWU_ME 字段描述 (继续)

字段	描述
0	内部模块标志未用作唤醒源
1	内部模块标志用作唤醒源

18.4.10 LLWU 引脚标志 1 寄存器 (LLWU_PF1)

LLWU_PF1 包含用于指示导致 MCU 退出 LLS 或 VLLS 模式的唤醒源的唤醒标志。对于 LLS，此触发源造成 CPU 中断。对于 VLLS，此触发源造成 MCU 复位流。

外部唤醒标志为只读，向对应的 WUF_x 位写入 1 可将标志清零。如果相关 WUPE_x 清零，则唤醒标志(WUF_x) (如已置 1) 将保持为 1。

注

此寄存器在如下情况下复位：

1. 收到由非 VLLS 信号导致的芯片复位时；
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + 9h 偏移 = 4007_C009h

位	7	6	5	4	3	2	1	0
读	WUF7	WUF6	WUF5	WUF4	WUF3	WUF2	WUF1	WUF0
写	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
复位	0	0	0	0	0	0	0	0

LLWU_PF1 字段描述

字段	描述
7 WUF7	LLWU_P7 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF7 写入 1。 0 LLWU_P7 输入不是唤醒源 1 LLWU_P7 输入是唤醒源
6 WUF6	LLWU_P6 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF6 写入 1。 0 LLWU_P6 输入不是唤醒源 1 LLWU_P6 输入是唤醒源
5 WUF5	LLWU_P5 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF5 写入 1。

下一页继续介绍此表...

LLWU_PF1 字段描述 (继续)

字段	描述
	0 LLWU_P5 输入不是唤醒源 1 LLWU_P5 输入是唤醒源
4 WUF4	LLWU_P4 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF4 写入 1。 0 LLWU_P4 输入不是唤醒源 1 LLWU_P4 输入是唤醒源
3 WUF3	LLWU_P3 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF3 写入 1。 0 LLWU_P3 输入不是唤醒源 1 LLWU_P3 输入是唤醒源
2 WUF2	LLWU_P2 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF2 写入 1。 0 LLWU_P2 输入不是唤醒源 1 LLWU_P2 输入是唤醒源
1 WUF1	LLWU_P1 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF1 写入 1。 0 LLWU_P1 输入不是唤醒源 1 LLWU_P1 输入是唤醒源
0 WUF0	LLWU_P0 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF0 写入 1。 0 LLWU_P0 输入不是唤醒源 1 LLWU_P0 输入是唤醒源

18.4.11 LLWU 引脚标志 2 寄存器 (LLWU_PF2)

LLWU_PF2 包含用于指示导致 MCU 退出 LLS 或 VLLS 模式的唤醒源的唤醒标志。对于 LLS，此触发源造成 CPU 中断流。对于 VLLS，此触发源造成 MCU 复位流。外部唤醒标志为只读，向对应的 WUF_x 位写入 1 可将标志清零。如果相关 WUPE_x 清零，则唤醒标志(WUF_x) (如已设置) 将保持设置。

注

此寄存器在如下情况下复位：

1. 收到由非 VLLS 信号导致的芯片复位时；
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + Ah 偏移 = 4007_C00Ah

位	7	6	5	4	3	2	1	0
读	WUF15	WUF14	WUF13	WUF12	WUF11	WUF10	WUF9	WUF8
写	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
复位	0	0	0	0	0	0	0	0

LLWU_PF2 字段描述

字段	描述
7 WUF15	LLWU_P15 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF15 写入 1。 0 LLWU_P15 输入不是唤醒源 1 LLWU_P15 输入是唤醒源
6 WUF14	LLWU_P14 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF14 写入 1。 0 LLWU_P14 输入不是唤醒源 1 LLWU_P14 输入是唤醒源
5 WUF13	LLWU_P13 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF13 写入 1。 0 LLWU_P13 输入不是唤醒源 1 LLWU_P13 输入是唤醒源
4 WUF12	LLWU_P12 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF12 写入 1。 0 LLWU_P12 输入不是唤醒源 1 LLWU_P12 输入是唤醒源
3 WUF11	LLWU_P11 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF11 写入 1。 0 LLWU_P11 输入不是唤醒源 1 LLWU_P11 输入是唤醒源
2 WUF10	LLWU_P10 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF10 写入 1。 0 LLWU_P10 输入不是唤醒源 1 LLWU_P10 输入是唤醒源
1 WUF9	LLWU_P9 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF9 写入 1。

下一页继续介绍此表...

LLWU_PF2 字段描述 (继续)

字段	描述
	0 LLWU_P9 输入不是唤醒源 1 LLWU_P9 输入是唤醒源
0 WUF8	LLWU_P8 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF8 写入 1。 0 LLWU_P8 输入不是唤醒源 1 LLWU_P8 输入是唤醒源

18.4.12 LLWU 引脚标志 3 寄存器 (LLWU_PF3)

LLWU_PF3 包含用于指示导致 MCU 退出 LLS 或 VLLS 模式的唤醒源的唤醒标志。对于 LLS，此触发源造成 CPU 中断流。对于 VLLS，此触发源造成 MCU 复位流。

外部唤醒标志为只读，向对应的 WUF_x 位写入 1 可将标志清零。如果相关 WUPE_x 清零，则唤醒标志(WUF_x) (如已设置) 将保持设置。

注

此寄存器在如下情况下复位：

1. 收到由非 VLLS 信号导致的芯片复位时；
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + Bh 偏移 = 4007_C00Bh

位	7	6	5	4	3	2	1	0
读	WUF23	WUF22	WUF21	WUF20	WUF19	WUF18	WUF17	WUF16
写	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
复位	0	0	0	0	0	0	0	0

LLWU_PF3 字段描述

字段	描述
7 WUF23	LLWU_P23 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF23 写入 1。 0 LLWU_P23 输入不是唤醒源 1 LLWU_P23 input was a wakeup source
6 WUF22	LLWU_P22 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF22 写入 1。

下一页继续介绍此表...

LLWU_PF3 字段描述 (继续)

字段	描述
	0 LLWU_P22 输入不是唤醒源 1 LLWU_P22 input was a wakeup source
5 WUF21	LLWU_P21 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF21 写入 1。 0 LLWU_P21 输入不是唤醒源 1 LLWU_P21 input was a wakeup source
4 WUF20	LLWU_P20 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF20 写入 1。 0 LLWU_P20 输入不是唤醒源 1 LLWU_P20 input was a wakeup source
3 WUF19	LLWU_P19 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF19 写入 1。 0 LLWU_P19 输入不是唤醒源 1 LLWU_P19 input was a wakeup source
2 WUF18	LLWU_P18 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF18 写入 1。 0 LLWU_P18 输入不是唤醒源 1 LLWU_P18 input was a wakeup source
1 WUF17	LLWU_P17 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF17 写入 1。 0 LLWU_P17 输入不是唤醒源 1 LLWU_P17 input was a wakeup source
0 WUF16	LLWU_P16 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF16 写入 1。 0 LLWU_P16 输入不是唤醒源 1 LLWU_P16 input was a wakeup source

18.4.13 LLWU 引脚标志 4 寄存器 (LLWU_PF4)

LLWU_PF4 包含用于指示导致 MCU 退出 LLS 或 VLLS 模式的唤醒源的唤醒标志。对于 LLS，此触发源造成 CPU 中断流。对于 VLLS，此触发源造成 MCU 复位流。

外部唤醒标志为只读，向对应的 WUF_x 位写入 1 可将标志清零。如果相关 WUPE_x 清零，则唤醒标志(WUF_x) (如已设置) 将保持设置。

注

此寄存器在如下情况下复位:

1. 收到由非 VLLS 信号导致的芯片复位时;
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + Ch 偏移 = 4007_C00Ch

位	7	6	5	4	3	2	1	0
读	WUF31	WUF30	WUF29	WUF28	WUF27	WUF26	WUF25	WUF24
写	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
复位	0	0	0	0	0	0	0	0

LLWU_PF4 字段描述

字段	描述
7 WUF31	LLWU_P31 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF31 写入 1。 0 LLWU_P31 输入不是唤醒源 1 LLWU_P31 input was a wakeup source
6 WUF30	LLWU_P30 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF30 写入 1。 0 LLWU_P30 输入不是唤醒源 1 LLWU_P30 input was a wakeup source
5 WUF29	LLWU_P29 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF29 写入 1。 0 LLWU_P29 输入不是唤醒源 1 LLWU_P29 input was a wakeup source
4 WUF28	LLWU_P28 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF28 写入 1。 0 LLWU_P28 输入不是唤醒源 1 LLWU_P28 input was a wakeup source
3 WUF27	LLWU_P27 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF27 写入 1。 0 LLWU_P27 输入不是唤醒源 1 LLWU_P27 input was a wakeup source
2 WUF26	LLWU_P26 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF26 写入 1。

下一页继续介绍此表...

LLWU_PF4 字段描述 (继续)

字段	描述
	0 LLWU_P26 输入不是唤醒源 1 LLWU_P26 input was a wakeup source
1 WUF25	LLWU_P25 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF25 写入 1。 0 LLWU_P25 输入不是唤醒源 1 LLWU_P25 input was a wakeup source
0 WUF24	LLWU_P24 的唤醒标志 表示已使能外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 WUF24 写入 1。 0 LLWU_P24 输入不是唤醒源 1 LLWU_P24 input was a wakeup source

18.4.14 LLWU 模块标志 5 寄存器 (LLWU_MF5)

LLWU_MF5 包含用于指示导致 MCU 退出 LLS 或 VLLS 模式的内部唤醒源的唤醒标志。对于 LLS，此触发源造成 CPU 中断流。对于 VLLS，此触发源造成 MCU 复位。

对于能够在低漏电功耗模式下运行的内部外设，例如实时时钟模块或 CMP 模块，相关外设的标志可作为 MWUFx 位访问。需要在外设中将此标志清零，而不是向 MWUFx 位写入 1。

注

此寄存器在如下情况下复位：

1. 收到由非 VLLS 信号导致的芯片复位时；
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + Dh 偏移 = 4007_C00Dh

位	7	6	5	4	3	2	1	0
读	MWUF7	MWUF6	MWUF5	MWUF4	MWUF3	MWUF2	MWUF1	MWUF0
写								
复位	0	0	0	0	0	0	0	0

LLWU_MF5 字段描述

字段	描述
7 MWUF7	<p>模块 7 的唤醒标志</p> <p>表示已使能内部外设是退出低漏电功耗模式的源。要将此标志清零,请按照内部外设标志清零机制进行操作。</p> <p>0 模块 7 输入不是唤醒源 1 模块 7 输入是唤醒源</p>
6 MWUF6	<p>模块 6 的唤醒标志</p> <p>表示已使能内部外设是退出低漏电功耗模式的源。要将此标志清零,请按照内部外设标志清零机制进行操作。</p> <p>0 模块 6 输入不是唤醒源 1 模块 6 输入是唤醒源</p>
5 MWUF5	<p>模块 5 的唤醒标志</p> <p>表示已使能内部外设是退出低漏电功耗模式的源。要将此标志清零,请按照内部外设标志清零机制进行操作。</p> <p>0 模块 5 输入不是唤醒源 1 模块 5 输入是唤醒源</p>
4 MWUF4	<p>模块 4 的唤醒标志</p> <p>表示已使能内部外设是退出低漏电功耗模式的源。要将此标志清零,请按照内部外设标志清零机制进行操作。</p> <p>0 模块 4 输入不是唤醒源 1 模块 4 输入是唤醒源</p>
3 MWUF3	<p>模块 3 的唤醒标志</p> <p>表示已使能内部外设是退出低漏电功耗模式的源。要将此标志清零,请按照内部外设标志清零机制进行操作。</p> <p>0 模块 3 输入不是唤醒源 1 模块 3 输入是唤醒源</p>
2 MWUF2	<p>模块 2 的唤醒标志</p> <p>表示已使能内部外设是退出低漏电功耗模式的源。要将此标志清零,请按照内部外设标志清零机制进行操作。</p> <p>0 模块 2 输入不是唤醒源 1 模块 2 输入是唤醒源</p>
1 MWUF1	<p>模块 1 的唤醒标志</p> <p>表示已使能内部外设是退出低漏电功耗模式的源。要将此标志清零,请按照内部外设标志清零机制进行操作。</p> <p>0 模块 1 输入不是唤醒源 1 模块 1 输入是唤醒源</p>
0 MWUF0	<p>模块 0 的唤醒标志</p> <p>表示已使能内部外设是退出低漏电功耗模式的源。要将此标志清零,请按照内部外设标志清零机制进行操作。</p> <p>0 模块 0 输入不是唤醒源 1 模块 0 输入是唤醒源</p>

18.4.15 LLWU 引脚滤波器 1 寄存器 (LLWU_FILT1)

LLWU_FILT1 是一个控制和状态寄存器，用于使能/禁用外部引脚的数字滤波器 1 功能。

注

此寄存器在如下情况下复位：

1. 收到由非 VLLS 信号导致的芯片复位时；
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + Eh 偏移 = 4007_C00Eh

	位	7	6	5	4	3	2	1	0
读		FILTF	FILTE			FILTSEL			
写		w1c							
复位		0	0	0	0	0	0	0	0

LLWU_FILT1 字段描述

字段	描述
7 FILTF	滤波器检测标志 表示 FILTSEL 选择的已滤波外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 FILTF 写入 1。 0 引脚滤波器 1 不是唤醒源 1 引脚滤波器 1 是唤醒源
6-5 FILTE	外部引脚上的数字滤波器 控制外部引脚检测的数字滤波器选项。 00 滤波器已禁用 01 滤波器上升沿检测已使能 10 滤波器下降沿检测已使能 11 滤波器任何边沿检测已使能
FILTSEL	滤波器引脚选择 从唤醒引脚中选择一个以复用至滤波器。 00000 为滤波器选择 LLWU_P0 11111 为滤波器选择 LLWU_P31

18.4.16 LLWU 引脚滤波器 2 寄存器 (LLWU_FILT2)

LLWU_FILT2 是一个控制和状态寄存器，用于使能/禁用外部引脚的数字滤波器 2 功能。

注

此寄存器在如下情况下复位：

1. 收到由非 VLLS 信号导致的芯片复位时；
2. 收到可以触发由非 VLLS 信号导致的芯片复位时。

不触发非 VLLS 型的芯片复位对其无影响。详情请参见[简介](#) 详细信息。

地址: 4007_C000h 基准 + Fh 偏移 = 4007_C00Fh

位	7	6	5	4	3	2	1	0
读	FILTF	FILTE			FILTSEL			
写	w1c							
复位	0	0	0	0	0	0	0	0

LLWU_FILT2 字段描述

字段	描述
7 FILTF	<p>滤波器检测标志</p> <p>表示 FILTSEL 选择的已滤波外部唤醒引脚是退出低漏电功耗模式的源。要将此标志清零，请向 FILTF 写入 1。</p> <p>0 引脚滤波器 2 不是唤醒源 1 引脚滤波器 2 是唤醒源</p>
6-5 FILTE	<p>外部引脚上的数字滤波器</p> <p>控制外部引脚检测的数字滤波器选项。</p> <p>00 滤波器已禁用 01 滤波器上升沿检测已使能 10 滤波器下降沿检测已使能 11 滤波器任何边沿检测已使能</p>
FILTSEL	<p>滤波器引脚选择</p> <p>从唤醒引脚中选择一个以复用至滤波器。</p> <p>00000 为滤波器选择 LLWU_P0 11111 为滤波器选择 LLWU_P31</p>

18.5 功能说明

低漏电唤醒单元 (LLWU) 模块允许将内部外设和外部输入引脚作为低漏电模式的唤醒源。

它仅在 LLS 和 VLLS_x 模式下有效。

LLWU 模块包含各外部引脚和内部模块的引脚使能。对于每个外部引脚，用户可以禁用或选择唤醒边沿类型，选项如下：

- 下降沿
- 上升沿
- 任意沿

当外部引脚使能为唤醒源时，引脚必须配置为输入引脚。

LLWU 基于 LPO 时钟设置可选的 3 周期毛刺滤波器。一旦检测到的外部引脚必须保持有效电平，直至已使能的毛刺滤波器超时。另外同步操作还需要 2 个周期，当滤波器功能使能时，这使得检测电路置位系统为唤醒或复位事件之前出现总共高达 5 个周期的延迟。4 个唤醒检测滤波器可用于选定的外部引脚。内部模块上未提供毛刺滤波器。内部模块上未提供毛刺滤波器。

对于内部模块中断，WUMEx 位可将相关模块中断使能为唤醒源。

18.5.1 LLS 模式

通过外部引脚输入或内部模块中断触发的唤醒事件将导致 CPU 产生中断并开始执行用户代码。

18.5.2 VLLS 模式

任何来自 VLLS 的唤醒始终通过复位来恢复，且设置 RCM_SRS[WAKEUP] 以指示低漏电模式已激活。在对 PMC_REGSC[ACKISO] 执行写操作之后，状态保留数据将丢失且 I/O 将恢复。

因 $\overline{\text{RESET}}$ 引脚置位导致的 VLLS 退出事件通过系统复位导致退出。状态保留数据丢失，I/O 状态将立即返回至其复位状态。在通过复位向量提取开始 CPU 操作之前，RCM_SRS[WAKEUP] 和 RCM_SRS[PIN] 位将置位且系统将执行复位流程。

18.5.3 初始化

对于已使能的外围唤醒输入，必须在进入 LLS 或 VLLS_x 模式之前通过软件将外围标志清零，以免立即退出该模式。

在进入 LLS 或 VLLS_x 模式之前，还必须通过软件将与外部输入引脚（已滤波和未滤波）相关的标志清零。

使能外部引脚滤波器或更改源引脚之后，先等待至少 5 个 LPO 时钟周期，然后再进入 LLS 或 VLLS_x 模式，以便滤波器初始化。

注

从 VLLS 模式恢复之后，用户必须重新初始化外设等的配置，然后再清除 PMC_REGSC[ACKISO]。尤其必须还原已使能的 LLWU 唤醒引脚的引脚配置，以免在清除 PMC_REGSC[ACKISO] 时错误设置 LLWU 标志。

选为唤醒源引脚的信号必须为数字引脚，可以在 PORT 模块中的引脚复用中设置。

第 19 章

杂项控制模块(MCM)

19.1 简介

杂项控制模块 (MCM) 可提供多种杂项控制功能。

19.1.1 特性

MCM 包含下列特性：

- 程序可见的有关平台配置和版本的信息

19.2 存储器映射/寄存器说明

以下存储器映射和寄存器说明介绍了使用字节地址的寄存器。

MCM 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
E008_0008	交叉开关 (AXBS) 从机配置 (MCM_PLASC)	16	R	000Fh	19.2.1/336
E008_000A	交叉开关 (AXBS) 主机配置 (MCM_PLAMC)	16	R	0017h	19.2.2/336
E008_000C	交叉开关 (AXBS) 控制寄存器 (MCM_PLACR)	32	R/W	0000_0000h	19.2.3/337
E008_0010	中断状态和控制寄存器 (MCM_ISCR)	32	R	0002_0000h	19.2.4/337
E008_0040	计算操作控制寄存器 (MCM_CPO)	32	R/W	0000_0000h	19.2.5/340

19.2.1 交叉开关(AXBS)从机配置 (MCM_PLASC)

PLASC 是一个 16 位只读寄存器，用于识别是否存在连接到交叉开关的总线从机。

地址: E008_0000h 基准 + 8h 偏移 = E008_0008h

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读	0								ASC							
写																
复位	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

MCM_PLASC 字段描述

字段	描述
15-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
ASC	ASC 字段中的每个位表示交叉开关的从输入端口是否存在对应连接。 0 不存在至 AXBS 输入端口 <i>n</i> 的总线从连接 1 存在至 AXBS 输入端口 <i>n</i> 的总线从连接

19.2.2 交叉开关(AXBS)主机配置 (MCM_PLAMC)

PLASC 是一个 16 位只读寄存器，用于识别是否存在连接到交叉开关的总线主机。

地址: E008_0000h 基准 + Ah 偏移 = E008_000Ah

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读	0								AMC							
写																
复位	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

MCM_PLAMC 字段描述

字段	描述
15-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
AMC	AMC 字段中的每个位表示 AXBS 主输入端口是否存在对应连接。 0 不存在至 AXBS 输入端口 <i>n</i> 的总线主连接。 1 存在至 AXBS 输入端口 <i>n</i> 的总线主连接。

19.2.3 交叉开关 (AXBS) 控制寄存器 (MCM_PLACR)

PLACR 寄存器选择适合交叉开关主机的仲裁策略。

地址: E008_0000h 基准 + Ch 偏移 = E008_000Ch

位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R	0																
W	[Shaded]																
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R	0							ARB	Reserved								
W	[Shaded]							[Shaded]	[Shaded]								
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0

MCM_PLACR 字段描述

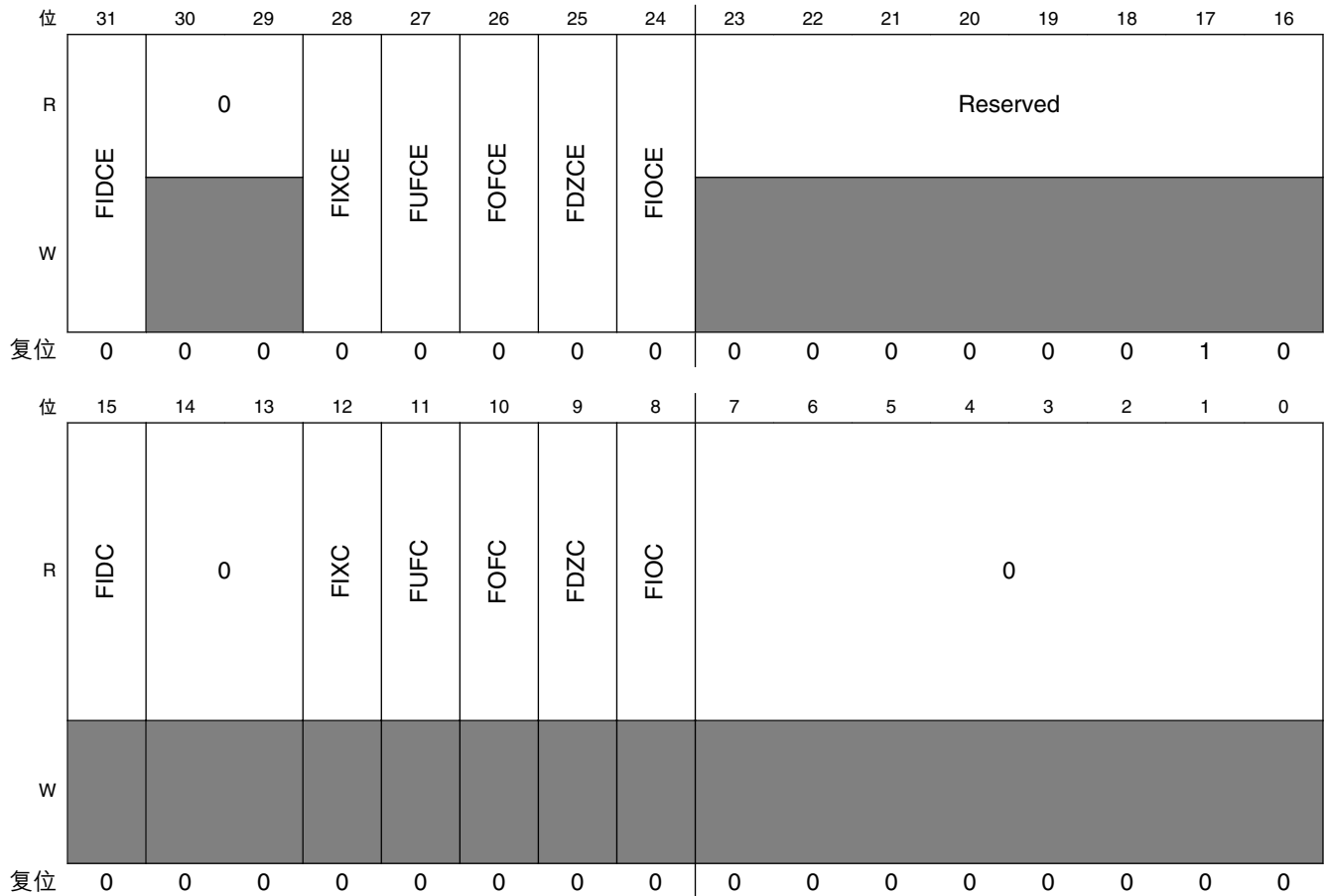
字段	描述
31-10 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
9 ARB	仲裁选择 0 交叉开关主机的固定优先级仲裁 1 交叉开关主机的轮询调度仲裁
Reserved	此字段为保留字段。

19.2.4 中断状态和控制寄存器 (MCM_ISCR)

MCM_ISCR 寄存器包含与内核浮点异常相关的使能位和状态位。单个活动指示器首先在异常情况使能时获得资格，然后逻辑综合形成一个发送至内核 NVIC 的中断请求。

位 15-8 是基于处理器 FPSCR 寄存器的只读指示器标志位。尝试写入这些位的操作被忽略。一旦置位，标志位保持有效，直到软件清除对应的 FPSCR 位。

地址: E008_0000h 基准 + 10h 偏移 = E008_0010h



MCM_ISR 字段描述

字段	描述
31 FIDCE	FPU 输入非规范化中断使能 0 禁用中断 1 使能中断
30-29 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
28 FIXCE	FPU 不精确中断使能 0 禁用中断 1 使能中断
27 FUFCE	FPU 下溢中断使能 0 禁用中断 1 使能中断
26 FOFCE	FPU 溢出中断使能 0 禁用中断 1 使能中断

下一页继续介绍此表...

MCM_ISCR 字段描述 (继续)

字段	描述
25 FDZCE	FPU 除数为零中断使能 0 禁用中断 1 使能中断
24 FIOCE	FPU 无效操作中中断使能 0 禁用中断 1 使能中断
23–16 Reserved	此字段为保留字段。
15 FIDC	FPU 输入非规范化中断状态 这个只读位是内核 FPSCR[IDC]位的副本，表示在处理器的 FPU 内检测到输入非规范化数字。一旦置位，此位保持置位，直到软件清除 FPSCR[IDC]位。 0 无中断 1 发生中断
14–13 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
12 FIXC	FPU 不精确中断状态 这个只读位是内核 FPSCR[IXC]位的副本，表示在处理器的 FPU 内检测到不精确数字。一旦置位，此位保持置位，直到软件清除 FPSCR[IXC]位。 0 无中断 1 发生中断
11 FUFC	FPU 下溢中断状态 这个只读位是内核 FPSCR[UFC]位的副本，表示在处理器的 FPU 内检测到下溢。一旦置位，此位保持置位，直到软件清除 FPSCR[UFC]位。 0 无中断 1 发生中断
10 FOFC	FPU 溢出中断状态 这个只读位是内核 FPSCR[OFC]位的副本，表示在处理器的 FPU 内检测到溢出。一旦置位，此位保持置位，直到软件清除 FPSCR[OFC]位。 0 无中断 1 发生中断
9 FDZC	FPU 除以零中断状态 这个只读位是内核 FPSCR[DZC]位的副本，表示在处理器的 FPU 内检测到除数为零。一旦置位，此位保持置位，直到软件清除 FPSCR[DZC]位。 0 无中断 1 发生中断
8 FIOC	FPU 无效操作中中断状态 这个只读位是内核 FPSCR[IOC]位的副本，表示在处理器的 FPU 内检测到非法操作。一旦置位，此位保持置位，直到软件清除 FPSCR[IOC]位。

下一页继续介绍此表...

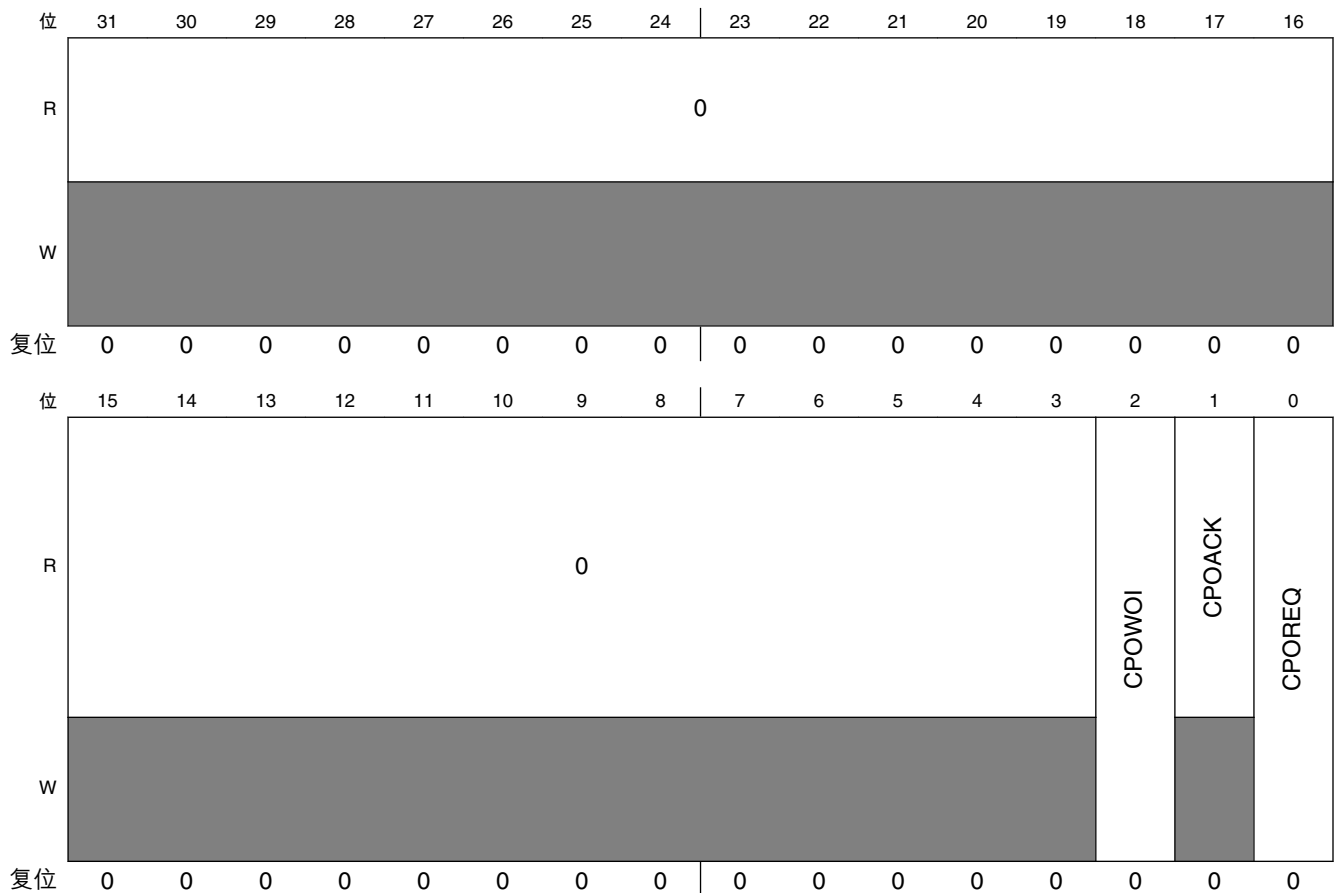
MCM_ISCR 字段描述 (继续)

字段	描述
	0 无中断 1 发生中断
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

19.2.5 计算操作控制寄存器 (MCM_CPO)

此寄存器控制计算操作。

地址: E008_0000h 基准 + 40h 偏移 = E008_0040h



MCM_CPO 字段描述

字段	描述
31-3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 CPOWOI	中断时计算操作唤醒

下一页继续介绍此表...

MCM_CPO 字段描述 (继续)

字段	描述
	0 无效。 1 置位后，CPOREQ 将在任何中断或异常向量提取时清除。
1 CPOACK	计算操作确认 0 计算操作进入未完成或者计算操作退出已完成。 1 计算操作进入已完成或者计算操作退出未完成。
0 CPOREQ	计算操作请求 如果 CPOW0I 为 1，则该位将由向量提取自动清除。 0 请求被清除。 1 请求计算操作。

19.3 功能说明

本节介绍 MCM 模块的功能说明。

19.3.1 中断

如果以下项目之一为真，将生成 MCM 中断：

- 使能 FPU 输入非规范化中断 (FIDCE)，同时输入为非规范化 (FIDC)
- 使能 FPU 不精确中断 (FIXCE)，同时数字不精确 (FIXC)
- 使能 FPU 下溢中断 (FUFCE)，同时发生了下溢 (FUFC)
- 使能 FPU 上溢中断 (FOFCE)，同时发生了上溢 (FOFC)
- 使能 FPU 除数为零中断 (FDZCE)，同时发生了除以零 (FDZC)
- 使能 FPU 无效操作中断 (FDZCE)，同时发生了无效操作 (FDZC)

19.3.1.1 确定中断的来源

要确定中断的确切来源，使用相应的中断启用位限定中断状态标志。

1. 计算 MCM_ISCR[31:16]&&MCM_ISCR[15:0]
2. 从结果中找到有效的标志位，从而找到正确的中断源

第 20 章 交叉开关-精简版 (AXBS-Lite)

20.1 此模块的芯片实现细节

20.1.1 简化交叉开关主机分配

与交叉开关相连的主机按以下方式分配：

主机模块	主机端口号
ARM 内核代码总线	0
ARM 内核系统总线	1
DMA	2
USB OTG	4

20.1.2 简化交叉开关从机分配

与交叉开关相连的从机按以下方式分配：

从机模块	从机端口号
Flash 存储器控制器	0
SRAM 控制器	1,2
外设桥 0/GPIO	3

20.2 简介

此处所示信息包含与交叉开关的布局、配置和编程有关的信息。

交叉开关凭借交叉开关结构连接总线主机和总线从机。该结构允许全部 4 个总线主机同时访问不同的总线从机，同时在其访问相同从机时提供总线主机间的仲裁。

20.2.1 特性

交叉开关具有以下功能：

- 对称交叉总线开关结构
 - 允许不同主机到不同从机间的并发操作
- 最多单时钟 32 位传输
- 可编程配置为固定优先级仲裁或轮询调度仲裁（参见芯片特性信息）。

20.3 存储器映射/寄存器定义

此交叉开关通过最少的门电路实现，因此没有存储器映射的配置寄存器。

要了解交叉开关中的仲裁方法是否可以编程以及由哪个模块编程，请参见有关芯片特性信息。

20.4 功能说明

20.4.1 常规操作

当主机访问交叉开关时，会立即执行该访问。如果该访问的目标从机端口可用，则将立即出现在该从机端口上。通过交叉开关可进行单时钟或零等待状态访问。如果该访问的目标从机端口繁忙或停驻在其他主机端口上，则请求主机将处于等待状态，直至目标从机端口可执行该主机的请求。为该请求服务时的延迟取决于每个主机的优先级以及响应从机的访问时间。

由于交叉开关可看作是主机设备的另一个从机，因此主机设备不知道它是否实际上拥有其目标从机端口。当主机无法控制其目标从机端口时，只需等待。

当该主机控制了目标从机端口后，将一直保有该端口的控制权，直至它运行了 IDLE 周期，或将另一个从机端口作为下次访问的目标，才会放弃对该从机端口的控制。

如果另一优先级更高的主机向此从机端口提出请求，该主机也会失去对该从机端口的控制。

交叉开关可终止所有主机 IDLE 传输，而不是终止来自任一从机总线的传输。另外，如果没有主机请求访问从机端口，交叉开关将在从机总线上执行 IDLE 传输，即便授权了某默认主机访问该从机端口。

当某个从机总线被交叉开关配置为空闲时，它将停驻在使用该从机端口的最后一个主机上。这样做是为了节省仲裁延迟的初始时钟，否则会出现该主机不得不重新通过仲裁来获取该从机端口的控制权。

20.4.2 仲裁

交叉开关支持两种仲裁算法：

- 固定优先级
- 循环

全局从机端口仲裁的选择在 MCM 模块中进行控制。对于固定优先级，将 MCM_PLACR[ARB]置 0。对于循环模式，将 MCM_PLACR[ARB]置 1。此仲裁设置适用于所有从机端口。

20.4.2.1 固定优先级操作

在固定优先级模式下执行操作时，将为每个主机分配一个唯一的优先级，编号越大，主机的优先级越高（例如，在有 5 个主机的系统中，主机 1 的优先级比主机 3 低）。如果两个主机请求访问同一从机端口，则优先级高的主机将获得该从机端口的控制权。

注

在此仲裁模式下，优先级高的主机可以独占某从机端口，并阻止任何优先级较低的主机访问该端口。

当某主机向某从机端口提出请求后，该从机端口将检查新请求主机的优先级是否高于目前控制该端口的主机的优先级，除非该从机端口处于暂停状态。该从机端口在每个时钟边沿执行一次仲裁检查，以确保由正确的主机（如果有）控制该端口。

下表根据请求主机的端口介绍了可能的情形：

表 20-1. 交叉开关如何将将从机端口的控制权授予主机

当	则交叉开关将控制权授予请求主机
下面两种情形都为真： <ul style="list-style-type: none"> • 当前主机没有运行传输。 • 新发出请求的主机的优先级高于当前主机的优先级。 	在下一时钟边沿
下列两种情形都为真：	在不定长突发传输的下一个仲裁点

下一页继续介绍此表...

表 20-1. 交叉开关如何将主机端口的控制权授予主机 (继续)

当	则交叉开关将控制权授予请求主机
<ul style="list-style-type: none"> • 当前主机正在运行不定长突发传输。 • 发出请求的主机的优先级高于当前主机的优先级。 	
发出请求的主机的优先级低于当前主机的优先级。	在以下任一周期结束时: <ul style="list-style-type: none"> • IDLE 周期 • 对非当前从机端口执行的非 IDLE 周期

20.4.2.2 循环优先级操作

在循环模式下执行操作时，会根据主机端口号为每个主机分配一个相对优先级。之后，将此相对优先级与最后一个执行传输的主机的主机端口号 (ID) 进行比较。优先级最高的请求主机将在下一个传输边界成为该从机总线的所有者。优先级取决于请求主机的 ID 与最后一个主机的 ID 之间的偏差程度。

在获得从机端口的访问权限后，主机可以根据需要对该端口执行多次传输，直至另一个主机向同一从机端口提出请求。队列中的下一个主机获得在下一个传输边界处访问该从机端口的权限；如果当前主机没有挂起的访问请求，也可能就在下一个时钟周期开始访问。

举例说明循环优先级模式下的仲裁机制：假定主机端口 0、1、4 和 5 可访问交叉开关。如果从机端口的最后一个主机为主机 1，且主机 0、4 和 5 同时提出请求，则其工作顺序为：4、5、0。

循环仲裁模式对可用从机端口带宽的分配一般更公平一些 (与固定优先级相比)，因为固定主机优先级不影响主机的选择。

20.5 初始化/应用信息

交叉开关无需初始化。有关仲裁方案复位状态的信息，请参见配置一章中的 AXBS 小节。

第 21 章

外设桥 (AIPS-Lite)

21.1 此模块的芯片实现细节

21.1.1 外设桥的数量

此器件含有一个外设桥。

21.1.2 存储器映射

外设桥用于访问该器件上大多数模块的寄存器。有关每个模块存储器槽分配的信息，请参见 [AIPS0 存储器映射](#)。

21.2 简介

外设桥将交叉开关接口转连至一个能访问片上大多数从机外设的接口。

外设桥占用 64 MB 地址空间（划分为许多 4 KB 的外设槽）。(也可能并未使用任何外设槽。有关槽分配的详情，请参见“存储器映射”章节。) 该桥接器包含单独的时钟使能输入，使每个槽都能支持速度较低的外设。

21.2.1 特性

外设桥的主要特性：

- 支持 8 位、16 位和 32 位数据路径宽度的外设槽

21.2.2 一般操作

连接外设桥的从设备模块均有可编程的控制和状态寄存器。系统主机通过外设桥对这些寄存器进行读写操作。

外设的寄存器映射位于 4 KB 边界上。每个外设均分配了一个或多个 4-KB 存储器映射块。两个全局外部模块使能功能可用于剩下的地址空间，以便自定义和扩展寻址的外设。

21.3 存储器映射/寄存器定义

此器件上的 AIPS 模块不含任何用户可编程寄存器。

21.4 功能说明

外设桥用作交叉开关和从机外设总线之间的总线协议转换器。

外设桥管理面向连接的从设备的所有事务，并通过对连接的地址空间的访问进行解码，从而为外设总线上的模块生成选择信号。

21.4.1 访问支持

本设备支持对 32 位外设执行对齐/不对齐的 32 位、16 位以及字节访问。由于支持不对齐的访问，因此可将存储器置于从机外设总线上。虽然外设桥不执行明确的检查，但外设寄存器必须对齐。外设电桥将对允许的 64 位事务执行两次从机外设总线传输，仅传输至 32 位大小的外设槽。其他访问仅通过一次传输来完成。

对外设槽执行的所有访问的数据位宽必须小于或等于指定的外设槽宽度。如果执行的访问的数据宽度大于目标端口的位宽，则会生成错误响应。

第 22 章 直接存储器访问多路复用器 (DMAMUX)

22.1 此模块的芯片实现细节

22.1.1 DMA MUX 请求源

该器件集成了一个 DMA 请求多路复用器,允许多达 63 个 DMA 请求信号映射到 16 个 DMA 通道中的任意一个。由于采用多路复用器,因此任何 DMA 请求源与特定 DMA 通道之间不存在硬相关性。

有些模块支持异步 DMA 操作,如下面的 DMA 来源分配表的最后一列所示。

表 22-1. DMA 请求源 - MUX 0

请求源编号	请求源模块	请求源说明	支持异步 DMA
0	—	通道禁用 ¹	
1	保留	未使用	
2	UART0	接收	
3	UART0	发送	
4	UART1	接收	
5	UART1	发送	
6	UART2	接收	
7	UART2	发送	
8	保留	—	
9	保留	—	
10	保留	—	
11		—	
12	I ² S0	接收	是
13	I ² S0	传送	是
14	SPI0	接收	
15	SPI0	发送	

表 22-1. DMA 请求源 - MUX 0 (继续)

请求源编号	请求源模块	请求源说明	支持异步 DMA
16	SPI1	发送或接收	
17	保留	—	
18	LPI ² C0	主机/从机接收	是
19	LPI ² C1	主机/从机接收	是
20	TPM0	通道 0	是
21	TPM0	通道 1	是
22	TPM0	通道 2	是
23	TPM0	通道 3	是
24	TPM0	通道 4	是
25	TPM0	通道 5	是
26	LPI ² C0	主机/从机发送	是
27	LPI ² C1	主机/从机发送	是
28	TPM1	通道 0	是
29	TPM1	通道 1	是
30	TPM2	通道 0	是
31	TPM2	通道 1	是
32	保留	—	
33	保留	—	
34	保留	—	
35	保留	—	
36	保留	—	
37	保留	—	
38	FlexIO	移位器 0	是
39	FlexIO	移位器 1	是
40	ADC0	—	是
41	FlexCAN1 (只针对 KS22)	—	
42	CMP0	—	是
43	FlexIO	移位器 2	是
44	FlexIO	移位器 3	是
45	DAC0	—	
46	I ² S1	接收	是
47	I ² S1	发送	是
48	PDB	—	
49	端口控制模块	端口 A	是
50	端口控制模块	端口 B	有
51	端口控制模块	端口 C	有
52	端口控制模块	端口 D	有
53	端口控制模块	端口 E	有
54	TPM0	溢出	是

下一页继续介绍此表...

表 22-1. DMA 请求源 - MUX 0 (继续)

请求源编号	请求源模块	请求源说明	支持异步 DMA
55	TPM1	溢出	是
56	TPM2	溢出	是
57	保留	—	
58	LPUART0	接收	是
59	LPUART0	发送	是
60	DMA 多路复用器	始终使能	
61	DMA 多路复用器	始终使能	
62	DMA 多路复用器	始终使能	
63	DMA 多路复用器	始终使能	

1. 配置某个 DMA 通道以选择请求源 0 或任何保留请求源可禁用该 DMA 通道。

22.1.2 DMA 通过 PIT 触发器传输

PIT 模块可以在前四个 DMA 通道上触发 DMA 传输。分配详情请参见 [PIT/DMA 周期性触发器分配](#)。

22.2 简介

22.2.1 概述

直接存储器访问多路复用器 (DMAMUX) 把称为槽的 DMA 源路由至 16 个 DMA 通道中的任意一个。此过程如下图所示。

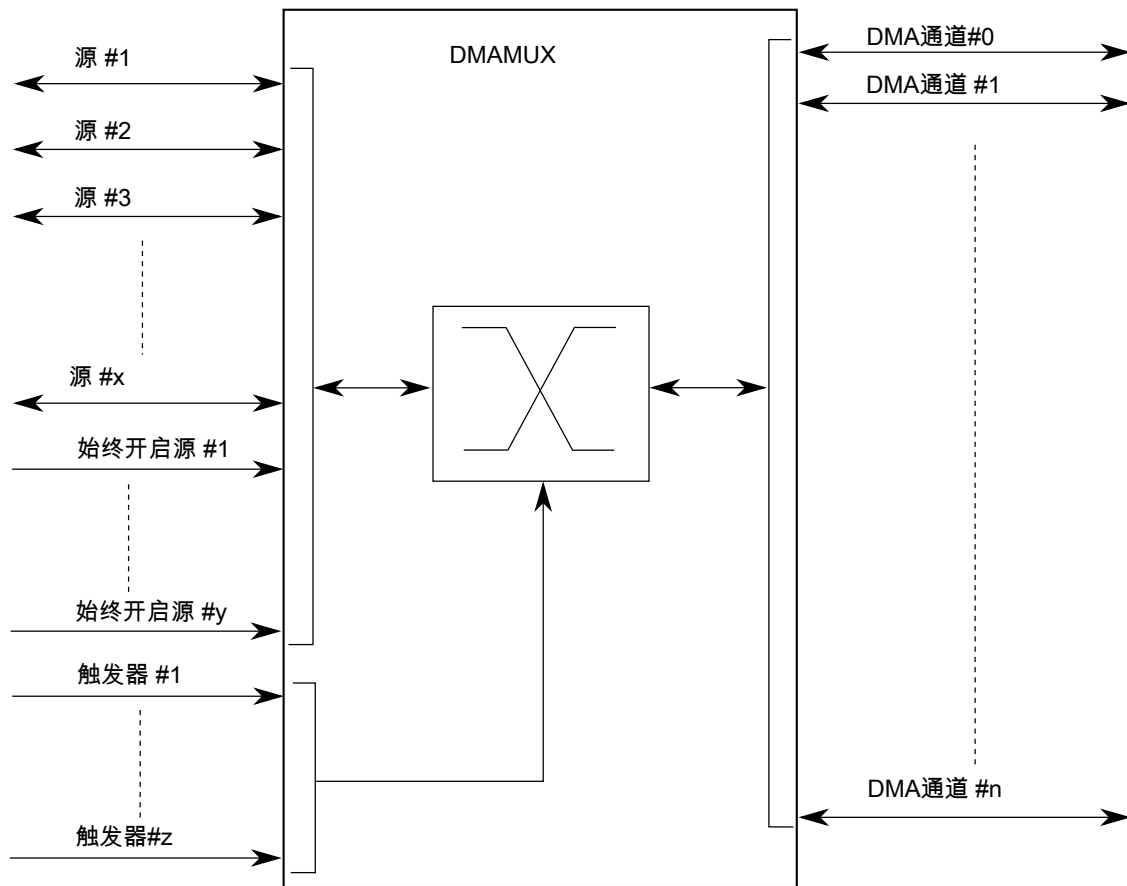


图 22-1. DMAMUX 结构框图

22.2.2 特性

DMAMUX 模块具有以下特性：

- 可以将最多 3459 个外设槽和最多 4 个始终开启槽路由至 16 个通道。
- 16 可独立选择的 DMA 通道路由器。
 - 前 4 个通道还提供触发器功能。
- 可以将各通道路由器分配给可能的的外设 DMA 槽或始终开启槽中的一个。

22.2.3 操作模式

有下列操作模式可用：

- 禁用模式

在此模式下，DMA 通道被禁用。由于 DMA 通道的禁用和使能主要通过 DMA 配置寄存器来实现，因此，此模式主要用作 DMA 通道多路复用器中 DMA 通道的复位状态。在对系统进行重新配置时（比如，更改 DMA 触发器的周期），也可用于暂停 DMA 通道。

- 正常模式

在此模式下，将 DMA 源直接连接至指定的 DMA 通道。DMAMUX 在此模式下的操作完全对系统透明。

- 周期触发模式

此模式下，一个 DMA 源只能请求一次 DMA 传输，比如，当发送缓冲器变空时，或者当接收缓冲器变满时，这都具有周期性。

周期配置是在周期性中断定时器(PIT)的寄存器中完成的。此模式只对 0–3 通道有效。

22.3 外部信号说明

DMAMUX 无外部引脚。

22.4 存储器映射/寄存器定义

本节详细说明 DMAMUX 中所有存储器映射的寄存器。

22.4.1 端字节序

此模块的存储器映射使用高字节序顺序。这表示：

- 对于 8 位寄存器，低地址字节将读取为最高有效字节。
- 对于 16 位寄存器，低地址字将读取为最高有效字。

下图提供了示例。

示例1：8位寄存器结构

地址	寄存器数据
00h	AAh
01h	BBh
02h	CCh
03h	DDh

对于此结构，地址00h的8位读取将得到DDh。

示例2：16位寄存器结构

地址	寄存器数据
00h	AABBh
02h	CCDDh

对于此结构，地址00h的16位读取将得到CCDDh。

图 22-2. 高字节序寄存器访问结果示例

DMAMUX 存储器映射

绝对地址（十六进制）	寄存器名称	宽度（单位：位）	访问	复位值	小节/页
4002_1000	通道配置寄存器 (DMAMUX_CHCFG0)	8	R/W	00h	22.4.2/354
4002_1001	通道配置寄存器 (DMAMUX_CHCFG1)	8	R/W	00h	22.4.2/354
4002_1002	通道配置寄存器 (DMAMUX_CHCFG2)	8	R/W	00h	22.4.2/354
4002_1003	通道配置寄存器 (DMAMUX_CHCFG3)	8	R/W	00h	22.4.2/354
4002_1004	通道配置寄存器 (DMAMUX_CHCFG4)	8	R/W	00h	22.4.2/354
4002_1005	通道配置寄存器 (DMAMUX_CHCFG5)	8	R/W	00h	22.4.2/354
4002_1006	通道配置寄存器 (DMAMUX_CHCFG6)	8	R/W	00h	22.4.2/354
4002_1007	通道配置寄存器 (DMAMUX_CHCFG7)	8	R/W	00h	22.4.2/354
4002_1008	通道配置寄存器 (DMAMUX_CHCFG8)	8	R/W	00h	22.4.2/354
4002_1009	通道配置寄存器 (DMAMUX_CHCFG9)	8	R/W	00h	22.4.2/354
4002_100A	通道配置寄存器 (DMAMUX_CHCFG10)	8	R/W	00h	22.4.2/354
4002_100B	通道配置寄存器 (DMAMUX_CHCFG11)	8	R/W	00h	22.4.2/354
4002_100C	通道配置寄存器 (DMAMUX_CHCFG12)	8	R/W	00h	22.4.2/354
4002_100D	通道配置寄存器 (DMAMUX_CHCFG13)	8	R/W	00h	22.4.2/354
4002_100E	通道配置寄存器 (DMAMUX_CHCFG14)	8	R/W	00h	22.4.2/354
4002_100F	通道配置寄存器 (DMAMUX_CHCFG15)	8	R/W	00h	22.4.2/354

22.4.2 通道配置寄存器 (DMAMUX_CHCFGn)

每个 DMA 通道都可以独立使能/禁用，并与系统中的一个 DMA 槽（外设槽或常开槽）相关联。

注

以同一源值设置多个 CHCFG 寄存器会导致不可预测的行为。即使禁用通道(ENBL==0)，也会如此。

在更改触发器或源设置之前，必须通过 CHCFGn[ENBL]禁用 DMA 通道。

Address: 4002_1000h base + 0h offset + (1d × i), where i=0d to 15d

位 读 写	7	6	5	4	3	2	1	0
	ENBL	TRIG	SOURCE					
复位	0	0	0	0	0	0	0	0

DMAMUX_CHCFGn 字段描述

字段	描述
7 ENBL	<p>DMA 通道使能</p> <p>使能 DMA 通道。</p> <p>0 DMA 通道已禁用。此模式主要用在 DMAMux 的配置中。DMA 有着独立的通道使用/禁用，应通过此功能来禁用或重新配置 DMA 通道。</p> <p>1 DMA 通道已使能</p>
6 TRIG	<p>DMA 通道触发器使能</p> <p>使能被触发 DMA 通道的周期触发能力。</p> <p>0 触发已禁用。如果触发被禁用且 ENBL 置位，则 DMA 通道会简单地把指定源路由至 DMA 通道。(正常模式)</p> <p>1 触发已使能。如果触发被使能且 ENBL 置位，则 DMAMUX 处于周期触发器模式。</p>
SOURCE	<p>DMA 通道源 (槽)</p> <p>指定将哪个 DMA 源 (若有) 路由至特定的 DMA 通道。请参阅芯片特性的 DMAMUX 信息，了解关于外设及其槽数的详情。</p>

22.5 功能说明

DMAMUX 的主要目的是使系统能灵活地使用可用 DMA 通道。

为此，DMAMUX 的配置是一个静态程序，在系统引导代码执行过程中实现。然而，如果执行[使能和配置源](#)中所列程序，则可在系统正常运行过程中更改 DMAMUX 的配置。

就功能而言，DMAMUX 通道可以分为两类：

- 实施正常路由功能和周期触发功能的通道
- 只实施正常路由功能的通道

22.5.1 带定期触发功能的 DMA 通道

在正常的路由功能以外, DMAMUX 的前 4 个通道提供了一种特殊的周期触发功能, 可用于为字节、帧或固定间隔的数据包提供一种自动发送机制, 无需处理器的干预。

触发器由周期性中断定时器(PIT)生成; 如此, 周期性触发间隔的配置通过 PIT 中的配置寄存器完成。有关本主题的更多信息, 请参见关于周期性中断定时器的章节。

注

受系统动态性的影响 (由于 DMA 通道优先级、总线仲裁、中断服务例程长度等), 触发器与实际 DMA 传输之间的时钟周期数无法得到保证。

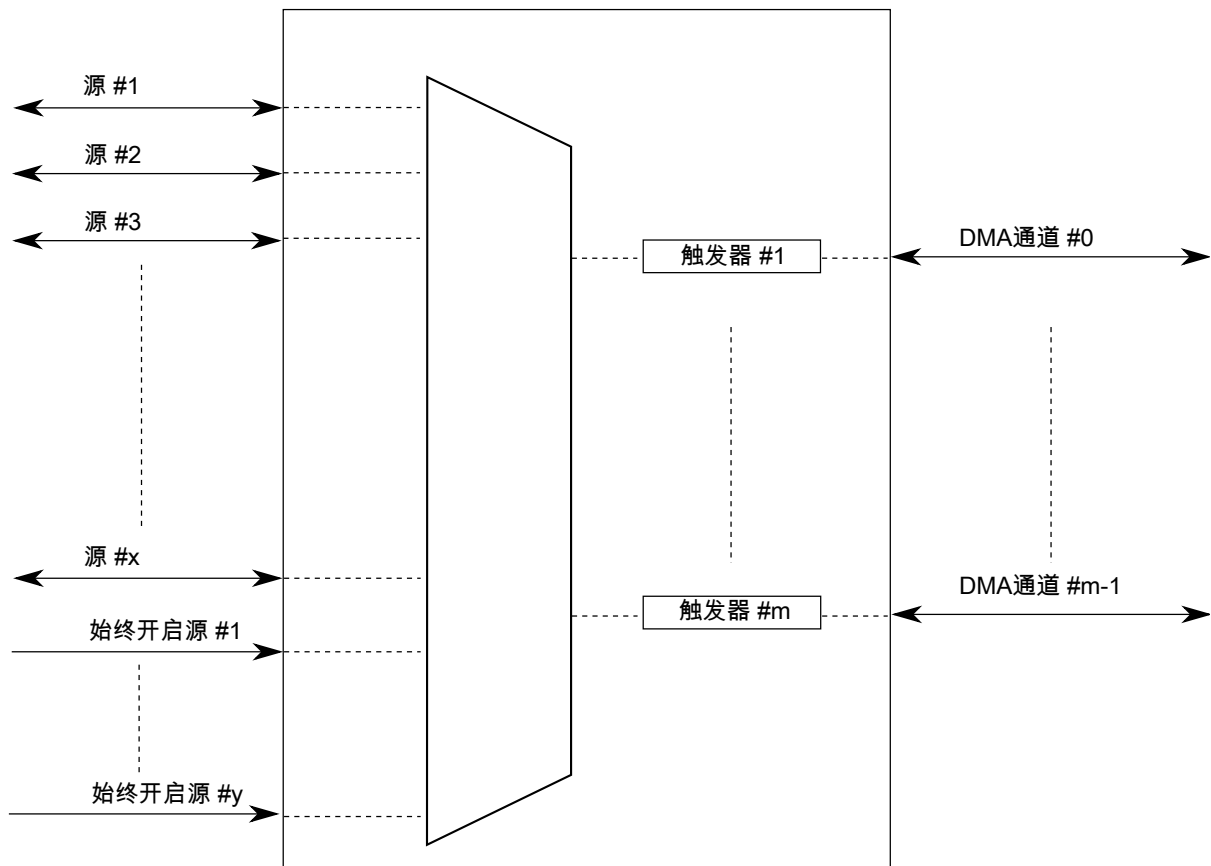


图 22-20. DMAMUX 触发的通道

借助 DMA 通道触发功能, 系统可以安排定期 DMA 传输操作, 通常是在某些外设的发送一侧, 并且不需要处理器介入。此触发器的工作方式是在触发事件到来时, 响应外设到 DMA 的请求。如下图所示。

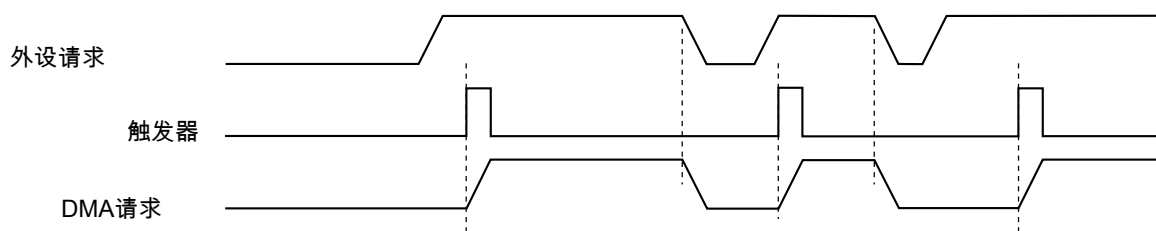


图 22-21. DMAMUX 通道触发: 正常操作

在满足 DMA 请求之后，外设将使其请求取反，实际上是复位其门控机制，直到外设重新置位其请求并看到下一个触发器事件为止。这意味着，如果看到一个触发事件，但外设未请求进行传输，则该触发事件会被忽略。这种情况如下图所示。

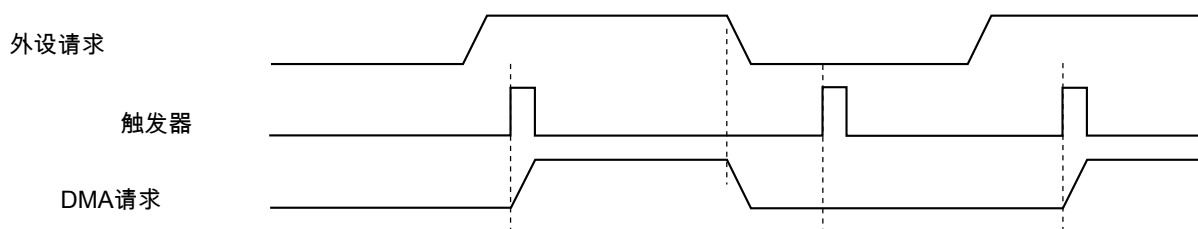


图 22-22. DMAMUX 通道触发: 被忽略的触发器

这种触发能力可以用于支持 DMA 传输的任意外设，对以下两种情况最有用：

- 定期轮询特定总线上的外部器件

例如，如上所述，将 SPI 的发送端分配给带触发器的 DMA 通道。设置后，SPI 将从存储器请求 DMA 传输，只要其发送缓冲器为空即可。通过在此通道上使用一个触发器，SPI 传输可以每 5 μs （作为示例）自动执行一次。在 SPI 的接收端，SPI 和 DMA 可以配置为把接收数据传输进存储器，这是一种有效的定期从外部器件读取数据并在无处理器干预的条件下把结果传输至存储器的方法。

- 用 GPIO 端口驱动波形或对波形采样

通过将 DMA 配置为传输数据到一个或多个 GPIO 端口，可以利用片上存储器中存储的表格数据创建复杂的波形。相反，通过 DMA 定期从一个或多个 GPIO 端口传输数据，可以对复杂波形采样并以表格形式把结果存储在片上存储器中。

有关各触发器功能的详细描述，包括分辨率、取值范围等，请参见“定期中断定时器”一节。

22.5.2 无触发功能的 DMA 通道

DMAMUX 的其他通道提供了正常的路由功能，如[操作模式](#)所述。

22.5.3 始终使能的 DMA 源

除了可以用作 DMA 源的外设以外，还有可以始终使能的 4 个额外 DMA 源。与外设 DMA 源不同（其中，外设控制着 DMA 传输过程中的数据流动），始终使能的源无法对数据传输进行这种“调节”。这些源在下列情况下最有用：

- 执行来往 GPIO 的 DMA 传输——使用无节流的（即尽量快），或者周期性的（利用 DMA 触发功能）将数据移自/至一个或者多个 GPIO 引脚。
- 执行从存储器到存储器的 DMA 传输——将数据从存储器移至存储器，一般以尽量快的方式，有时则用软件激活方式。
- 执行从存储器到外部总线（或相反）的 DMA 传输——与从存储器到存储器的传输类似，这种传输一般是尽快完成。
- 要求软件激活的任何 DMA 传输——应由软件显性启动的任何 DMA 传输。

在须由软件启动 DMA 传输的情况下，可以用一个始终使能的 DMA 源来实现最大的灵活性。当通过软件激活 DMA 通道时，后续执行次循环时会要求发送一个新的启动事件。这既可以是一个新的软件激活，也可以是来自 DMA 通道 MUX 的一个传输请求。这样做时的选项为：

- 在单个次循环中传输全部数据。

通过将 DMA 配置为在单个次循环中传输全部数据（即主循环计数器 = 1）时，不需要激活通道。这一选项的不足在于，在确定 DMA 传输对系统形成的负载时，精细度会降低。对于此选项，必须在 DMA 通道 MUX 中禁用 DMA 通道。

- 使用显性软件再激活。

在此选项中，将 DMA 配置为同时通过次循环和主循环传输数据，但要求在每个次循环之后，通过写入 DMA 寄存器来用处理器来重新激活通道。对于此选项，必须在 DMA 通道 MUX 中禁用 DMA 通道。

- 使用始终使能 DMA 源。

在此选项中，将 DMA 配置为同时通过次循环和主循环传输数据，并由 DMA 通道 MUX 实现通道再激活。对于此选项，DMA 通道应使能并指向一个“始终使能的”源。请注意，通道的再激活既可以是连续的（DMA 触发被禁用），也可以使用 DMA 触发功能。这样就可以在源之间逐一执行定期数据包传输，无需处理器的干预。

22.6 初始化/应用信息

本节就 DMA 通道多路复用器的初始化提供说明。

22.6.1 复位

各个位的复位状态如[存储器映射/寄存器定义](#)所示。总之，在复位之后，所有通道都被禁用，并且必须在使用前显性使能。

22.6.2 使能和配置源

要使能周期性触发源：

1. 确定该源将与哪个 DMA 通道关联。注意，只有前 4 个 DMA 通道具有周期性触发能力。
2. 清零 DMA 通道的 CHCFG[ENBL]和 CHCFG[TRIG]字段。
3. 确保在 DMA 中正常配置 DMA 通道。可以此时使能 DMA 通道。
4. 配置对应的定时器。
5. 选择要路由至 DMA 通道的源。写入对应的 CHCFG 寄存器，确保 CHCFG[ENBL]和 CHCFG[TRIG]字段置位。

注

以下是一个示例。参阅芯片配置详情，了解此器件具备触发能力的 DMA 通道数量。

要配置发送源#5 以配合 DMA 通道 1 使用且带周期性触发能力：

1. 将 0x00 写入 CHCFG1。
2. 在 DMA 中配置通道 1，包括使能该通道。
3. 为所需触发器间隔配置一个定时器。
4. 将 0xC5 写入 CHCFG1。

下面的代码示例展示了上面的第 1 至第 4 步：

```
void DMAMUX_Init(uint8_t DMA_CH, uint8_t DMAMUX_SOURCE)
{
    DMAMUX_0.CHCFG[DMA_CH].B.SOURCE = DMAMUX_SOURCE;
    DMAMUX_0.CHCFG[DMA_CH].B.ENBL   = 1;
    DMAMUX_0.CHCFG[DMA_CH].B.TRIG   = 1;
}
```

要使能源，(不带周期性触发能力)：

1. 确定该源将与哪个 DMA 通道关联。注意, 只有前 4 个 DMA 通道具有周期性触发能力。
2. 清零 DMA 通道的 CHCFG[ENBL]和 CHCFG[TRIG]字段。
3. 确保在 DMA 中正常配置 DMA 通道。可以此时使能 DMA 通道。
4. 选择要路由至 DMA 通道的源。写入对应的 CHCFG 寄存器, 确保 CHCFG[ENBL]置位在 CHCFG[TRIG]清零时。

注

以下是一个示例。参阅芯片配置详情, 了解此器件具备触发能力的 DMA 通道数量。

例如, 要配置发送源#5 以配合 DMA 通道 1 使用 (不带周期性触发能力):

1. 将 0x00 写入 CHCFG1。
2. 在 DMA 中配置通道 1, 包括使能该通道。
3. 将 0x85 写入 CHCFG1。

下面的代码示例展示了上面的第 1 至第 3 步:

```
In File registers.h:
#define DMAMUX_BASE_ADDR      0x40021000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCFG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCFG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCFG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCFG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCFG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCFG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCFG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCFG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCFG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCFG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCFG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCFG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCFG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCFG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCFG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCFG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);
```

```
In File main.c:
#include "registers.h"
:
:
*CHCFG1 = 0x00;
*CHCFG1 = 0x85;
```

要禁用源:

通过不将对应的源值写入任何 CHCFG 寄存器, 可禁用特定的 DMA 源。另外, 可能需要进行一些因模块而异的配置。有关更多详细信息, 请参见相应部分。

要切换 DMA 通道源:

1. 在 DMA 中禁用 DMA 通道, 为新源重新配置通道。
2. 清零 DMA 通道的 CHCFG[ENBL]和 CHCFG[TRIG]位。

3. 选择要路由至 DMA 通道的源。写入对应的 CHCFG 寄存器，确保 CHCFG[ENBL]和 CHCFG[TRIG]字段置位。

要将 DMA 通道 8 从发送源#5 切换至发送源#7:

1. 在 DMA 配置寄存器中，禁用 DMA 通道 8 并重新配置它以处理至外设槽 7 的传输。本例假设通道 8 不具备触发能力。
2. 将 0x00 写入 CHCFG8。
3. 将 0x87 写入 CHCFG8。(在本例中，置位 CHCFG[TRIG]不起作用，因为本例假设，通道 8 不支持周期性触发功能。)

下面的代码示例展示了上面的第 2 至第 3 步:

```
In File registers.h:
#define DMAMUX_BASE_ADDR      0x40021000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCFG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCFG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCFG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCFG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCFG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCFG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCFG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCFG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCFG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCFG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCFG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCFG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCFG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCFG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCFG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCFG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);

In File main.c:
#include "registers.h"
:
:
*CHCFG8 = 0x00;
*CHCFG8 = 0x87;
```


第 23 章

增强型直接存储器访问(eDMA)

23.1 简介

增强型直接存储器访问 (eDMA) 控制器是二代模块，能够执行复杂数据传输，且主机处理器干预最少。硬件微架构包括：

- DMA 引擎，用于执行：
 - 源地址和目标地址计算
 - 数据移动操作
- 本地存储器，包含用于每个 16 通道的传输控制描述符

23.1.1 eDMA 系统结构框图

图 23-1 描述了 eDMA 系统的组件，包括 eDMA 模块 (“引擎”)。

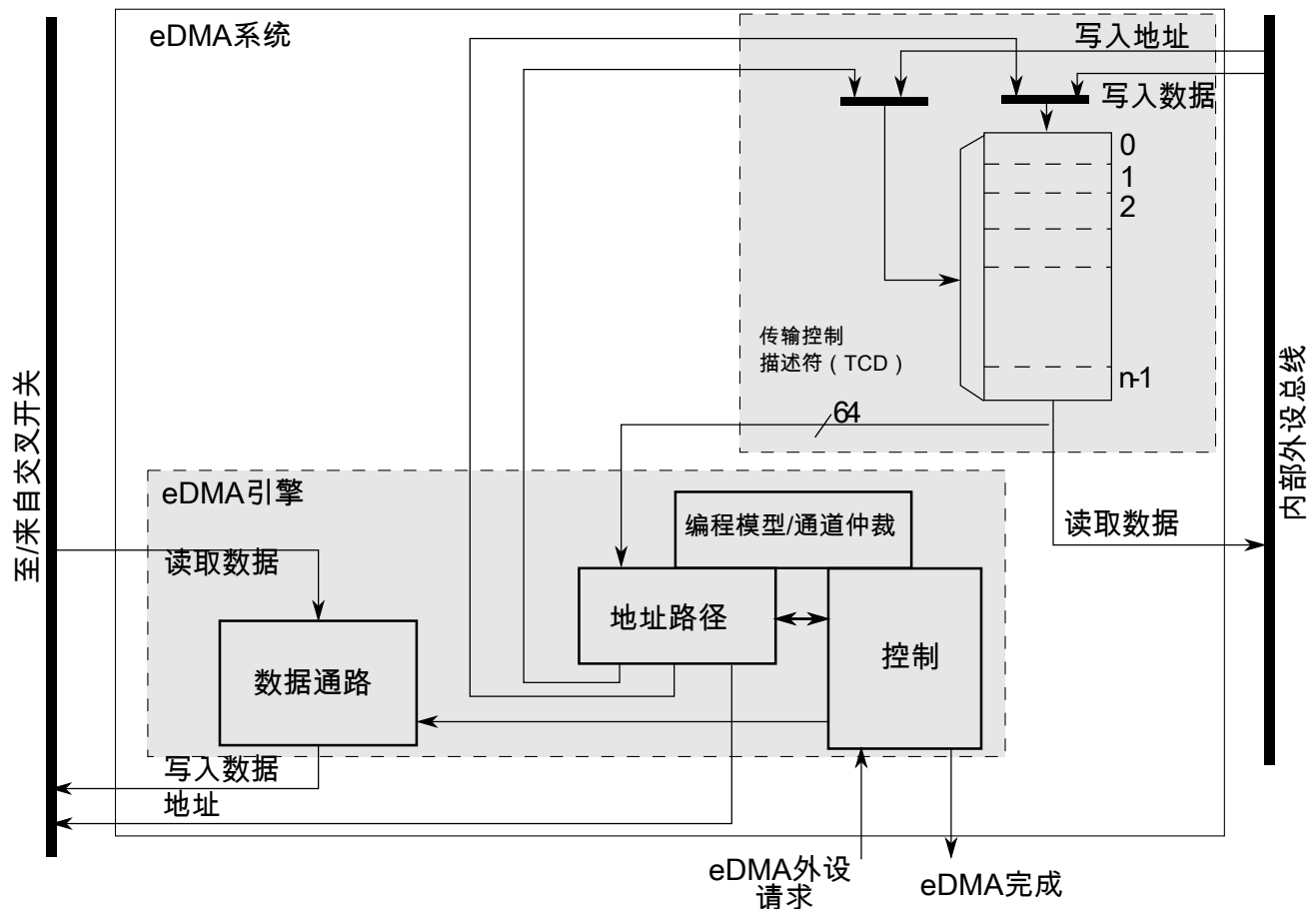


图 23-1. eDMA 系统结构框图

23.1.2 结构块

eDMA 模块分为两大主要模块：eDMA 引擎和传输控制描述符本地存储器。

eDMA 引擎可进一步细分为四个子模块：

表 23-1. eDMA 引擎子模块

子模块	功能
地址路径	<p>此结构块实现注册版本的两个通道传输控制描述符 (通道 x 和通道 y)，并管理所有主机总线地址计算。所有通道的功能相同。在此结构下，第一个通道处于激活状态时，如果认定优先级更高的通道激活，则读/写序列完成后，可抢占与一个通道关联的数据传输。通道被激活后，除非被优先级更高的通道抢占，否则会一直运行，直到次循环结束。这提供了一种机制 (由 DCHPRI_n[ECP]使能)，即可抢占大数据移动操作，以最大限度地缩短阻塞的另一个通道执行的时间。</p> <p>当选定执行某个通道时，会从本地存储器读取其 TCD 内容，并加载到地址路径通道 x 寄存器中正常启动，或者加载到通道 y 寄存器中，以抢占开始。次循环操作完成后，地址路径硬件将 TCD_n{SADDR, DADDR, CITER} 的新数值写回本地存储器。如果主迭代数已用尽，则执行额外处理，包括最终地址指针更新、重新加载 TCD_n_CITER 字段以及作为分散/聚集操作的一部分，从存储器获取下个 TCD_n。</p>

下一页继续介绍此表...

表 23-1. eDMA 引擎子模块 (继续)

子模块	功能
数据通路	本结构块实施总线主机读/写数据通路。它包括一个数据缓冲区及必要的多路复用逻辑，以支持所需的数据对齐。内部 读取数据总线是主要输入，内部 写入数据总线是主要输出。 地址和数据通路模块直接支持两级管路内部 总线。地址通路模块表示总线管路（地址阶段）的第一级，而数据通路模块执行管路（数据阶段）的第二级。
编程模型/通道仲裁	此结构块执行 eDMA 编程模型及通道仲裁逻辑的第一部分。编程模型寄存器与内部外设总线相连。eDMA 外设请求输入和中断请求输出也与此结构块相连（通过控制逻辑）。
控制	此结构块提供 eDMA 引擎所需的所有控制功能。对于源和目标大小相同的数据传输，eDMA 引擎执行一系列源读取/目标写入操作，直到已达到次循环字节数中指定的字节数。对于传输单位大小不同的描述符，对于每次大块的数据传输，都需要执行多次的小数据的访问。例如，如果源数据大小为 16 位，而目标数据为 32 位，则需要执行两次读取，然后执行一次 32 位写操作。

传输控制描述符本地存储器可进一步细分为：

表 23-2. 传输控制描述符存储器

子模块	说明
存储控制器	此逻辑实现所需的双端口控制器，管理来自 eDMA 引擎的访问及来自内部外设总线的引用。如之前所述，如果同时访问，eDMA 引擎具备优先访问权，而外设事务会延缓。
存储器阵列	每个通道的传输配置文件 TCD 存储。

23.1.3 特性

eDMA 是一个高度可编程的，经过优化最大程度减少主处理器干预的数据传输引擎。它适合于要传输的数据大小已静态已知且未被定义在传输数据内的应用。

eDMA 模块具有以下特性：

- 所有数据通过双地址传输移动：从源位置读取，写入目标位置
 - 可编程源、目标地址及传输大小
 - 支持增强型寻址模式
- 16 通道，可以最少的主机处理器干预实现复杂的数据传输
 - 内部数据缓冲区用作临时存储器，支持 16- 和 32-字节传输
 - 与交叉开关连接，以便总线管理数据传输
- 传输控制描述符(TCD)，支持双深嵌套传输操作
 - 存储在本地存储器中的适用于每种通道的 32 字节 TCD

- 由次字节传输数目定义的内部数据传输循环
- 由主迭代数目定义的外部数据传输循环
- 可通过三种方法激活通道：
 - 显式的软件初始化
 - 对于连续性传输，则通过通道-通道连接进行初始化
 - 配合外设节奏的硬件请求，每个通道一个
- 固定优先级和循环通道两种仲裁方式
- 通过可编程中断请求报告通道完成
 - 每个通道一次中断，可在主迭代数目完成时设置成有效
 - 每个通道支持可编程的错误终止事件，并且合并成一个错误中断给到中断控制器
- 可编程支持分散/集中 DMA 处理
- 支持复杂的数据结构

在对本模块的讨论中， n 指的是通道数。

23.2 操作模式

eDMA 有以下几种操作模式：

表 23-3. 操作模式

模式	说明
正常	正常模式下，eDMA 在源和目标之间传输数据。源和目标可以是可与 eDMA 配合使用的内存区或 I/O 数据块。 服务请求会引发传输控制描述符（TCD）所指定的特定数目字节（NBYTES）的传输。次循环按照读-写操作的顺序循环，为每个服务请求传输这些 NBYTES。每次服务请求都会执行一次主循环迭代，从而传输数据 NBYTES。
调试	在调试模式下，可通过控制寄存器配置 DMA 操作： <ul style="list-style-type: none"> • 如果 CR[EDBG] 已被清除，则 DMA 继续运行。 • 如果 CR[EDBG] 置位，eDMA 停止传输数据。如果在通道处于激活状态时进入调试模式，则 eDMA 持续运行，直至通道停用。
等待	进入等待模式前，DMA 尝试完成其当前的传输。传输完成后，器件即进入等待模式。

23.3 存储器映射/寄存器定义

eDMA 的编程模型分为两个区域：

- 第一个区域定义一些提供控制功能的寄存器
- 第二个区域则对应本地传输控制描述符(TCD) 存储器

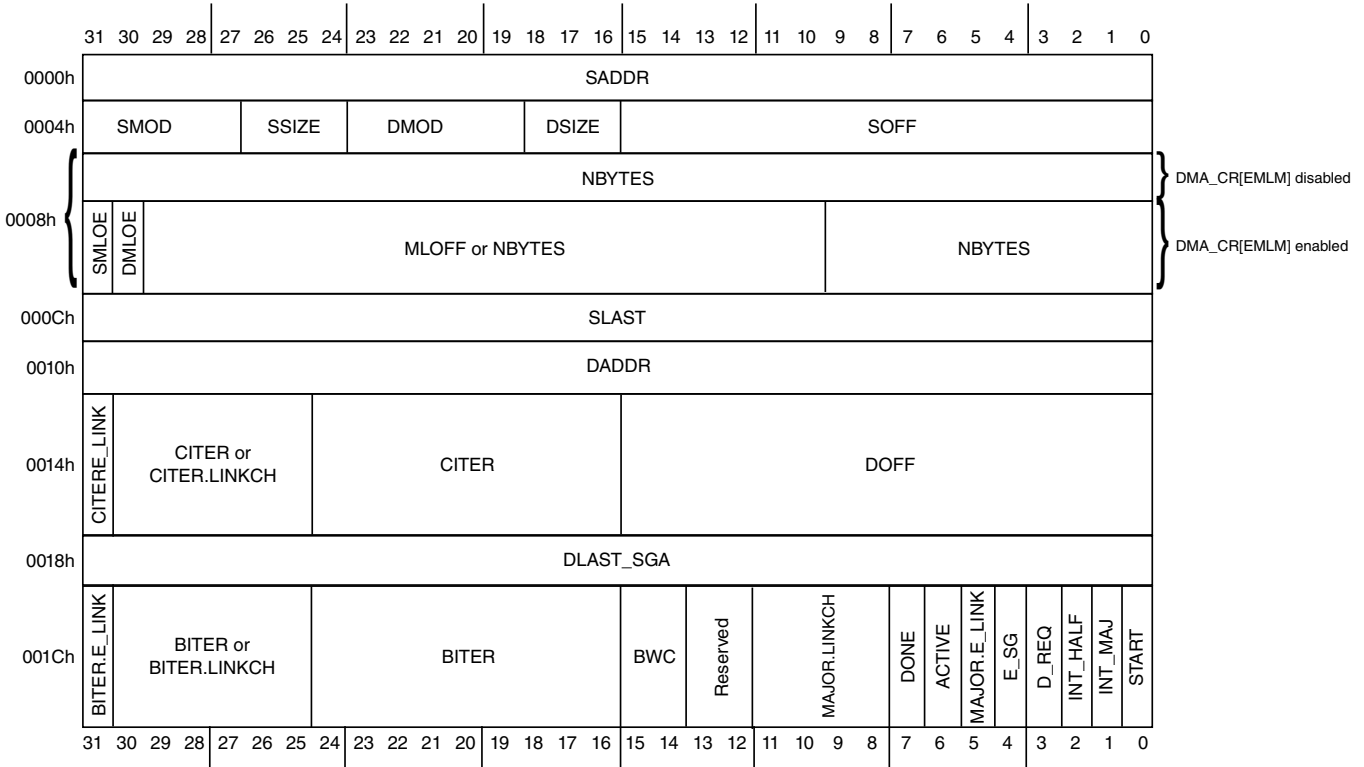
23.3.1 TCD 存储器

每个通道需要一个 32 字节传输控制描述符(TCD)来定义所需的数据传输操作。通道描述符按顺序存储在本地存储器中：通道 0、通道 1...通道 15。每个 TCD_{*n*} 定义均表示为 11 个 16 位或 32 位的寄存器。

23.3.2 TCD 初始化

激活通道前，必须先使用正确的传输配置文件初始化其 TCD。

23.3.3 TCD 结构



23.3.4 保留存储器和位字段

- 读取寄存器中的保留位会返回零值。
- 向寄存器的保留位写入会被忽略。
- 读取或写入一个预留存储器单元会产生一个总线错误。

DMA 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4000_8000	控制寄存器 (DMA_CR)	32	R/W	0000_0000h	23.3.5/378
4000_8004	错误状态寄存器 (DMA_ES)	32	R	0000_0000h	23.3.6/380
4000_800C	使能请求寄存器 (DMA_ERQ)	32	R/W	0000_0000h	23.3.7/382
4000_8014	使能错误中断寄存器 (DMA_EEI)	32	R/W	0000_0000h	23.3.8/384
4000_8018	清除使能错误中断寄存器 (DMA_CEEI)	8	W (始终读 0)	00h	23.3.9/386
4000_8019	置位使能错误中断寄存器 (DMA_SEEI)	8	W (始终读 0)	00h	23.3.10/387
4000_801A	清除使能请求寄存器 (DMA_CERQ)	8	W (始终读 0)	00h	23.3.11/388
4000_801B	置位使能请求寄存器 (DMA_SERQ)	8	W (始终读 0)	00h	23.3.12/389

下一页继续介绍此表...

DMA 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4000_801C	清除 DONE 状态位寄存器 (DMA_CDNE)	8	W (始终读 0)	00h	23.3.13/390
4000_801D	置位 START 位寄存器 (DMA_SSRT)	8	W (始终读 0)	00h	23.3.14/391
4000_801E	清除错误寄存器 (DMA_CERR)	8	W (始终读 0)	00h	23.3.15/392
4000_801F	清除中断请求寄存器 (DMA_CINT)	8	W (始终读 0)	00h	23.3.16/393
4000_8024	中断请求寄存器 (DMA_INT)	32	R/W	0000_0000h	23.3.17/393
4000_802C	错误寄存器 (DMA_ERR)	32	R/W	0000_0000h	23.3.18/396
4000_8034	硬件请求状态寄存器 (DMA_HRS)	32	R	0000_0000h	23.3.19/398
4000_8044	使能在停止状态下的异步请求寄存器 (DMA_EARS)	32	R/W	0000_0000h	23.3.20/401
4000_8100	通道 n 优先级寄存器 (DMA_DCHPRI3)	8	R/W	参见章节	23.3.21/403
4000_8101	通道 n 优先级寄存器 (DMA_DCHPRI2)	8	R/W	参见章节	23.3.21/403
4000_8102	通道 n 优先级寄存器 (DMA_DCHPRI1)	8	R/W	参见章节	23.3.21/403
4000_8103	通道 n 优先级寄存器 (DMA_DCHPRI0)	8	R/W	参见章节	23.3.21/403
4000_8104	通道 n 优先级寄存器 (DMA_DCHPRI7)	8	R/W	参见章节	23.3.21/403
4000_8105	通道 n 优先级寄存器 (DMA_DCHPRI6)	8	R/W	参见章节	23.3.21/403
4000_8106	通道 n 优先级寄存器 (DMA_DCHPRI5)	8	R/W	参见章节	23.3.21/403
4000_8107	通道 n 优先级寄存器 (DMA_DCHPRI4)	8	R/W	参见章节	23.3.21/403
4000_8108	通道 n 优先级寄存器 (DMA_DCHPRI11)	8	R/W	参见章节	23.3.21/403
4000_8109	通道 n 优先级寄存器 (DMA_DCHPRI10)	8	R/W	参见章节	23.3.21/403
4000_810A	通道 n 优先级寄存器 (DMA_DCHPRI9)	8	R/W	参见章节	23.3.21/403
4000_810B	通道 n 优先级寄存器 (DMA_DCHPRI8)	8	R/W	参见章节	23.3.21/403
4000_810C	通道 n 优先级寄存器 (DMA_DCHPRI15)	8	R/W	参见章节	23.3.21/403
4000_810D	通道 n 优先级寄存器 (DMA_DCHPRI14)	8	R/W	参见章节	23.3.21/403
4000_810E	通道 n 优先级寄存器 (DMA_DCHPRI13)	8	R/W	参见章节	23.3.21/403
4000_810F	通道 n 优先级寄存器 (DMA_DCHPRI12)	8	R/W	参见章节	23.3.21/403
4000_9000	TCD 源地址 (DMA_TCD0_SADDR)	32	R/W	未定义	23.3.22/404
4000_9004	TCD 有符号的源地址偏移 (DMA_TCD0_SOFF)	16	R/W	未定义	23.3.23/404
4000_9006	TCD 传输属性 (DMA_TCD0_ATTR)	16	R/W	未定义	23.3.24/405
4000_9008	TCD 次字节数目 (次循环映射禁用) (DMA_TCD0_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_9008	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD0_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_9008	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD0_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_900C	TCD 最后源地址调整 (DMA_TCD0_SLAST)	32	R/W	未定义	23.3.28/409
4000_9010	TCD 目标地址 (DMA_TCD0_DADDR)	32	R/W	未定义	23.3.29/409
4000_9014	TCD 有符号的目标地址偏移 (DMA_TCD0_DOFF)	16	R/W	未定义	23.3.30/410

下一页继续介绍此表...

DMA 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4000_9016	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD0_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_9016	DMA_TCD0_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_9018	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD0_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_901C	TCD 控制和状态 (DMA_TCD0_CSR)	16	R/W	未定义	23.3.34/413
4000_901E	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD0_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_901E	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD0_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_9020	TCD 源地址 (DMA_TCD1_SADDR)	32	R/W	未定义	23.3.22/404
4000_9024	TCD 有符号的源地址偏移 (DMA_TCD1_SOFF)	16	R/W	未定义	23.3.23/404
4000_9026	TCD 传输属性 (DMA_TCD1_ATTR)	16	R/W	未定义	23.3.24/405
4000_9028	TCD 次字节数目 (次循环映射禁用) (DMA_TCD1_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_9028	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD1_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_9028	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD1_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_902C	TCD 最后源地址调整 (DMA_TCD1_SLAST)	32	R/W	未定义	23.3.28/409
4000_9030	TCD 目标地址 (DMA_TCD1_DADDR)	32	R/W	未定义	23.3.29/409
4000_9034	TCD 有符号的目标地址偏移 (DMA_TCD1_DOFF)	16	R/W	未定义	23.3.30/410
4000_9036	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD1_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_9036	DMA_TCD1_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_9038	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD1_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_903C	TCD 控制和状态 (DMA_TCD1_CSR)	16	R/W	未定义	23.3.34/413
4000_903E	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD1_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_903E	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD1_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_9040	TCD 源地址 (DMA_TCD2_SADDR)	32	R/W	未定义	23.3.22/404
4000_9044	TCD 有符号的源地址偏移 (DMA_TCD2_SOFF)	16	R/W	未定义	23.3.23/404
4000_9046	TCD 传输属性 (DMA_TCD2_ATTR)	16	R/W	未定义	23.3.24/405
4000_9048	TCD 次字节数目 (次循环映射禁用) (DMA_TCD2_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_9048	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD2_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_9048	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD2_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_904C	TCD 最后源地址调整 (DMA_TCD2_SLAST)	32	R/W	未定义	23.3.28/409
4000_9050	TCD 目标地址 (DMA_TCD2_DADDR)	32	R/W	未定义	23.3.29/409

下一页继续介绍此表...

DMA 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4000_9054	TCD 有符号的目标地址偏移 (DMA_TCD2_DOFF)	16	R/W	未定义	23.3.30/410
4000_9056	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD2_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_9056	DMA_TCD2_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_9058	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD2_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_905C	TCD 控制和状态 (DMA_TCD2_CSR)	16	R/W	未定义	23.3.34/413
4000_905E	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD2_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_905E	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD2_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_9060	TCD 源地址 (DMA_TCD3_SADDR)	32	R/W	未定义	23.3.22/404
4000_9064	TCD 有符号的源地址偏移 (DMA_TCD3_SOFF)	16	R/W	未定义	23.3.23/404
4000_9066	TCD 传输属性 (DMA_TCD3_ATTR)	16	R/W	未定义	23.3.24/405
4000_9068	TCD 次字节数目 (次循环映射禁用) (DMA_TCD3_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_9068	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD3_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_9068	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD3_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_906C	TCD 最后源地址调整 (DMA_TCD3_SLAST)	32	R/W	未定义	23.3.28/409
4000_9070	TCD 目标地址 (DMA_TCD3_DADDR)	32	R/W	未定义	23.3.29/409
4000_9074	TCD 有符号的目标地址偏移 (DMA_TCD3_DOFF)	16	R/W	未定义	23.3.30/410
4000_9076	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD3_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_9076	DMA_TCD3_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_9078	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD3_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_907C	TCD 控制和状态 (DMA_TCD3_CSR)	16	R/W	未定义	23.3.34/413
4000_907E	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD3_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_907E	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD3_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_9080	TCD 源地址 (DMA_TCD4_SADDR)	32	R/W	未定义	23.3.22/404
4000_9084	TCD 有符号的源地址偏移 (DMA_TCD4_SOFF)	16	R/W	未定义	23.3.23/404
4000_9086	TCD 传输属性 (DMA_TCD4_ATTR)	16	R/W	未定义	23.3.24/405
4000_9088	TCD 次字节数目 (次循环映射禁用) (DMA_TCD4_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_9088	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD4_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_9088	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD4_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_908C	TCD 最后源地址调整 (DMA_TCD4_SLAST)	32	R/W	未定义	23.3.28/409

下一页继续介绍此表...

DMA 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4000_9090	TCD 目标地址 (DMA_TCD4_DADDR)	32	R/W	未定义	23.3.29/409
4000_9094	TCD 有符号的目标地址偏移 (DMA_TCD4_DOFF)	16	R/W	未定义	23.3.30/410
4000_9096	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD4_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_9096	DMA_TCD4_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_9098	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD4_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_909C	TCD 控制和状态 (DMA_TCD4_CSR)	16	R/W	未定义	23.3.34/413
4000_909E	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD4_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_909E	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD4_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_90A0	TCD 源地址 (DMA_TCD5_SADDR)	32	R/W	未定义	23.3.22/404
4000_90A4	TCD 有符号的源地址偏移 (DMA_TCD5_SOFF)	16	R/W	未定义	23.3.23/404
4000_90A6	TCD 传输属性 (DMA_TCD5_ATTR)	16	R/W	未定义	23.3.24/405
4000_90A8	TCD 次字节数目 (次循环映射禁用) (DMA_TCD5_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_90A8	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD5_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_90A8	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD5_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_90AC	TCD 最后源地址调整 (DMA_TCD5_SLAST)	32	R/W	未定义	23.3.28/409
4000_90B0	TCD 目标地址 (DMA_TCD5_DADDR)	32	R/W	未定义	23.3.29/409
4000_90B4	TCD 有符号的目标地址偏移 (DMA_TCD5_DOFF)	16	R/W	未定义	23.3.30/410
4000_90B6	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD5_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_90B6	DMA_TCD5_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_90B8	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD5_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_90BC	TCD 控制和状态 (DMA_TCD5_CSR)	16	R/W	未定义	23.3.34/413
4000_90BE	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD5_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_90BE	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD5_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_90C0	TCD 源地址 (DMA_TCD6_SADDR)	32	R/W	未定义	23.3.22/404
4000_90C4	TCD 有符号的源地址偏移 (DMA_TCD6_SOFF)	16	R/W	未定义	23.3.23/404
4000_90C6	TCD 传输属性 (DMA_TCD6_ATTR)	16	R/W	未定义	23.3.24/405
4000_90C8	TCD 次字节数目 (次循环映射禁用) (DMA_TCD6_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_90C8	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD6_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_90C8	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD6_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408

下一页继续介绍此表...

DMA 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4000_90CC	TCD 最后源地址调整 (DMA_TCD6_SLAST)	32	R/W	未定义	23.3.28/409
4000_90D0	TCD 目标地址 (DMA_TCD6_DADDR)	32	R/W	未定义	23.3.29/409
4000_90D4	TCD 有符号的目标地址偏移 (DMA_TCD6_DOFF)	16	R/W	未定义	23.3.30/410
4000_90D6	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD6_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_90D6	DMA_TCD6_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_90D8	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD6_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_90DC	TCD 控制和状态 (DMA_TCD6_CSR)	16	R/W	未定义	23.3.34/413
4000_90DE	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD6_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_90DE	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD6_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_90E0	TCD 源地址 (DMA_TCD7_SADDR)	32	R/W	未定义	23.3.22/404
4000_90E4	TCD 有符号的源地址偏移 (DMA_TCD7_SOFF)	16	R/W	未定义	23.3.23/404
4000_90E6	TCD 传输属性 (DMA_TCD7_ATTR)	16	R/W	未定义	23.3.24/405
4000_90E8	TCD 次字节数目 (次循环映射禁用) (DMA_TCD7_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_90E8	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD7_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_90E8	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD7_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_90EC	TCD 最后源地址调整 (DMA_TCD7_SLAST)	32	R/W	未定义	23.3.28/409
4000_90F0	TCD 目标地址 (DMA_TCD7_DADDR)	32	R/W	未定义	23.3.29/409
4000_90F4	TCD 有符号的目标地址偏移 (DMA_TCD7_DOFF)	16	R/W	未定义	23.3.30/410
4000_90F6	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD7_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_90F6	DMA_TCD7_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_90F8	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD7_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_90FC	TCD 控制和状态 (DMA_TCD7_CSR)	16	R/W	未定义	23.3.34/413
4000_90FE	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD7_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_90FE	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD7_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_9100	TCD 源地址 (DMA_TCD8_SADDR)	32	R/W	未定义	23.3.22/404
4000_9104	TCD 有符号的源地址偏移 (DMA_TCD8_SOFF)	16	R/W	未定义	23.3.23/404
4000_9106	TCD 传输属性 (DMA_TCD8_ATTR)	16	R/W	未定义	23.3.24/405
4000_9108	TCD 次字节数目 (次循环映射禁用) (DMA_TCD8_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_9108	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD8_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406

下一页继续介绍此表...

DMA 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4000_9108	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD8_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_910C	TCD 最后源地址调整 (DMA_TCD8_SLAST)	32	R/W	未定义	23.3.28/409
4000_9110	TCD 目标地址 (DMA_TCD8_DADDR)	32	R/W	未定义	23.3.29/409
4000_9114	TCD 有符号的目标地址偏移 (DMA_TCD8_DOFF)	16	R/W	未定义	23.3.30/410
4000_9116	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD8_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_9116	DMA_TCD8_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_9118	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD8_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_911C	TCD 控制和状态 (DMA_TCD8_CSR)	16	R/W	未定义	23.3.34/413
4000_911E	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD8_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_911E	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD8_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_9120	TCD 源地址 (DMA_TCD9_SADDR)	32	R/W	未定义	23.3.22/404
4000_9124	TCD 有符号的源地址偏移 (DMA_TCD9_SOFF)	16	R/W	未定义	23.3.23/404
4000_9126	TCD 传输属性 (DMA_TCD9_ATTR)	16	R/W	未定义	23.3.24/405
4000_9128	TCD 次字节数目 (次循环映射禁用) (DMA_TCD9_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_9128	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD9_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_9128	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD9_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_912C	TCD 最后源地址调整 (DMA_TCD9_SLAST)	32	R/W	未定义	23.3.28/409
4000_9130	TCD 目标地址 (DMA_TCD9_DADDR)	32	R/W	未定义	23.3.29/409
4000_9134	TCD 有符号的目标地址偏移 (DMA_TCD9_DOFF)	16	R/W	未定义	23.3.30/410
4000_9136	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD9_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_9136	DMA_TCD9_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_9138	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD9_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_913C	TCD 控制和状态 (DMA_TCD9_CSR)	16	R/W	未定义	23.3.34/413
4000_913E	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD9_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_913E	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD9_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_9140	TCD 源地址 (DMA_TCD10_SADDR)	32	R/W	未定义	23.3.22/404
4000_9144	TCD 有符号的源地址偏移 (DMA_TCD10_SOFF)	16	R/W	未定义	23.3.23/404
4000_9146	TCD 传输属性 (DMA_TCD10_ATTR)	16	R/W	未定义	23.3.24/405
4000_9148	TCD 次字节数目 (次循环映射禁用) (DMA_TCD10_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406

下一页继续介绍此表...

DMA 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4000_9148	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD10_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_9148	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD10_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_914C	TCD 最后源地址调整 (DMA_TCD10_SLAST)	32	R/W	未定义	23.3.28/409
4000_9150	TCD 目标地址 (DMA_TCD10_DADDR)	32	R/W	未定义	23.3.29/409
4000_9154	TCD 有符号的目标地址偏移 (DMA_TCD10_DOFF)	16	R/W	未定义	23.3.30/410
4000_9156	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD10_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_9156	DMA_TCD10_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_9158	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD10_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_915C	TCD 控制和状态 (DMA_TCD10_CSR)	16	R/W	未定义	23.3.34/413
4000_915E	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD10_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_915E	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD10_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_9160	TCD 源地址 (DMA_TCD11_SADDR)	32	R/W	未定义	23.3.22/404
4000_9164	TCD 有符号的源地址偏移 (DMA_TCD11_SOFF)	16	R/W	未定义	23.3.23/404
4000_9166	TCD 传输属性 (DMA_TCD11_ATTR)	16	R/W	未定义	23.3.24/405
4000_9168	TCD 次字节数目 (次循环映射禁用) (DMA_TCD11_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_9168	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD11_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_9168	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD11_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_916C	TCD 最后源地址调整 (DMA_TCD11_SLAST)	32	R/W	未定义	23.3.28/409
4000_9170	TCD 目标地址 (DMA_TCD11_DADDR)	32	R/W	未定义	23.3.29/409
4000_9174	TCD 有符号的目标地址偏移 (DMA_TCD11_DOFF)	16	R/W	未定义	23.3.30/410
4000_9176	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD11_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_9176	DMA_TCD11_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_9178	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD11_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_917C	TCD 控制和状态 (DMA_TCD11_CSR)	16	R/W	未定义	23.3.34/413
4000_917E	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD11_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_917E	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD11_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_9180	TCD 源地址 (DMA_TCD12_SADDR)	32	R/W	未定义	23.3.22/404
4000_9184	TCD 有符号的源地址偏移 (DMA_TCD12_SOFF)	16	R/W	未定义	23.3.23/404
4000_9186	TCD 传输属性 (DMA_TCD12_ATTR)	16	R/W	未定义	23.3.24/405

下一页继续介绍此表...

DMA 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4000_9188	TCD 次字节数目 (次循环映射禁用) (DMA_TCD12_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_9188	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD12_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_9188	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD12_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_918C	TCD 最后源地址调整 (DMA_TCD12_SLAST)	32	R/W	未定义	23.3.28/409
4000_9190	TCD 目标地址 (DMA_TCD12_DADDR)	32	R/W	未定义	23.3.29/409
4000_9194	TCD 有符号的目标地址偏移 (DMA_TCD12_DOFF)	16	R/W	未定义	23.3.30/410
4000_9196	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD12_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_9196	DMA_TCD12_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_9198	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD12_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_919C	TCD 控制和状态 (DMA_TCD12_CSR)	16	R/W	未定义	23.3.34/413
4000_919E	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD12_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_919E	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD12_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_91A0	TCD 源地址 (DMA_TCD13_SADDR)	32	R/W	未定义	23.3.22/404
4000_91A4	TCD 有符号的源地址偏移 (DMA_TCD13_SOFF)	16	R/W	未定义	23.3.23/404
4000_91A6	TCD 传输属性 (DMA_TCD13_ATTR)	16	R/W	未定义	23.3.24/405
4000_91A8	TCD 次字节数目 (次循环映射禁用) (DMA_TCD13_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_91A8	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD13_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_91A8	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD13_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_91AC	TCD 最后源地址调整 (DMA_TCD13_SLAST)	32	R/W	未定义	23.3.28/409
4000_91B0	TCD 目标地址 (DMA_TCD13_DADDR)	32	R/W	未定义	23.3.29/409
4000_91B4	TCD 有符号的目标地址偏移 (DMA_TCD13_DOFF)	16	R/W	未定义	23.3.30/410
4000_91B6	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD13_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_91B6	DMA_TCD13_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_91B8	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD13_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_91BC	TCD 控制和状态 (DMA_TCD13_CSR)	16	R/W	未定义	23.3.34/413
4000_91BE	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD13_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_91BE	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD13_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_91C0	TCD 源地址 (DMA_TCD14_SADDR)	32	R/W	未定义	23.3.22/404
4000_91C4	TCD 有符号的源地址偏移 (DMA_TCD14_SOFF)	16	R/W	未定义	23.3.23/404

下一页继续介绍此表...

DMA 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4000_91C6	TCD 传输属性 (DMA_TCD14_ATTR)	16	R/W	未定义	23.3.24/405
4000_91C8	TCD 次字节数目 (次循环映射禁用) (DMA_TCD14_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_91C8	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD14_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_91C8	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD14_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_91CC	TCD 最后源地址调整 (DMA_TCD14_SLAST)	32	R/W	未定义	23.3.28/409
4000_91D0	TCD 目标地址 (DMA_TCD14_DADDR)	32	R/W	未定义	23.3.29/409
4000_91D4	TCD 有符号的目标地址偏移 (DMA_TCD14_DOFF)	16	R/W	未定义	23.3.30/410
4000_91D6	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD14_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_91D6	DMA_TCD14_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_91D8	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD14_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_91DC	TCD 控制和状态 (DMA_TCD14_CSR)	16	R/W	未定义	23.3.34/413
4000_91DE	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD14_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_91DE	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD14_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416
4000_91E0	TCD 源地址 (DMA_TCD15_SADDR)	32	R/W	未定义	23.3.22/404
4000_91E4	TCD 有符号的源地址偏移 (DMA_TCD15_SOFF)	16	R/W	未定义	23.3.23/404
4000_91E6	TCD 传输属性 (DMA_TCD15_ATTR)	16	R/W	未定义	23.3.24/405
4000_91E8	TCD 次字节数目 (次循环映射禁用) (DMA_TCD15_NBYTES_MLNO)	32	R/W	未定义	23.3.25/406
4000_91E8	TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCD15_NBYTES_MLOFFNO)	32	R/W	未定义	23.3.26/406
4000_91E8	TCD 有符号的次循环偏移 (次循环映射和偏移已使能) (DMA_TCD15_NBYTES_MLOFFYES)	32	R/W	未定义	23.3.27/408
4000_91EC	TCD 最后源地址调整 (DMA_TCD15_SLAST)	32	R/W	未定义	23.3.28/409
4000_91F0	TCD 目标地址 (DMA_TCD15_DADDR)	32	R/W	未定义	23.3.29/409
4000_91F4	TCD 有符号的目标地址偏移 (DMA_TCD15_DOFF)	16	R/W	未定义	23.3.30/410
4000_91F6	TCD 当前次循环链接, 主循环数目 (通道链接使能) (DMA_TCD15_CITER_ELINKYES)	16	R/W	未定义	23.3.31/410
4000_91F6	DMA_TCD15_CITER_ELINKNO	16	R/W	未定义	23.3.32/411
4000_91F8	TCD 最后的目标地址调整/分散集中地址 (DMA_TCD15_DLASTSGA)	32	R/W	未定义	23.3.33/412
4000_91FC	TCD 控制和状态 (DMA_TCD15_CSR)	16	R/W	未定义	23.3.34/413
4000_91FE	TCD 开始次循环链接, 主循环数目 (通道链接使能) (DMA_TCD15_BITER_ELINKYES)	16	R/W	未定义	23.3.35/415
4000_91FE	TCD 开始次循环链接, 主循环数目 (通道链接禁用) (DMA_TCD15_BITER_ELINKNO)	16	R/W	未定义	23.3.36/416

23.3.5 控制寄存器 (DMA_CR)

CR 定义了 DMA 的基本操作配置。

仲裁可配置成使用固定优先级或循环方案。对于固定优先级仲裁，选定执行最高优先级通道的服务请求。通道优先级寄存器分配优先级；请查看 DCHPRI_n 寄存器。对于循环仲裁，其通道优先级被忽略，且在不考虑优先级的情况下对通道进行循环（按从高到低的通道编号）。

注

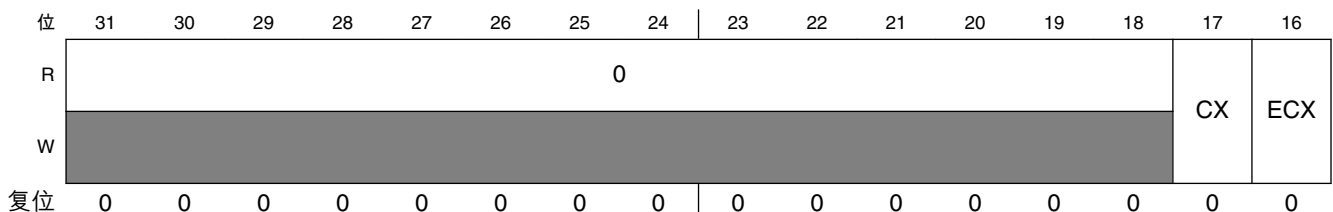
只在 DMA 通道处于非活动状态时向 CR 寄存器执行写入才能保证配置的正确性；也就是说，当 TCD_n_CSR[ACTIVE] 位被清除时。

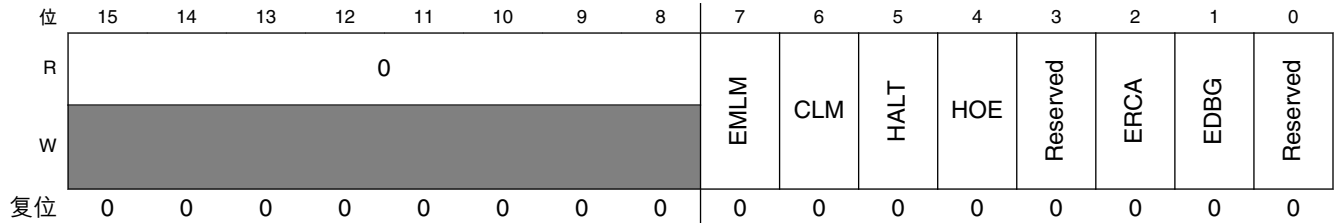
次循环偏移量是在次循环完成时，向最终源地址(TCD_n_SADDR) 或目标地址 (TCD_n_DADDR)添加的地址偏移数值。次循环偏移量使能后，在地址回写入 TCD 之前，会将次循环偏移量(MLOFF)添加到最终源地址(TCD_n_SADDR)、最终目标地址 (TCD_n_DADDR)上，或添加到两者上。如果主循环完成，则次循环偏移量会被忽略，而主循环地址偏移量 (TCD_n_SLAST 和 TCD_n_DLAST_SGA)则用于计算下个 TCD_n_SADDR 和 TCD_n_DADDR 值。

次循环映射使能 (EMLM 为 1) 后，TCD_n word2 被重新定义。部分 TCD_n word2 用于指明多个字段：用于指明次循环偏移量的源使能位(SMLOE)应在次循环完成时，应用于源地址(TCD_n_SADDR)；用于指明次循环偏移量的目标使能位(DMLOE)应在次循环完成时应用于目标地址 (TCD_n_DADDR)，以及标记扩展次循环偏移量值。相同的偏移量值(MLOFF)同时用作源和目标次循环偏移量。当任一次循环偏移使能 (SMLOE 置位或 DMLOE 置位) 时，NBYTES 字段减少到 10 位。当两个次循环偏移均禁用 (SMLOE 被清除，DMLOE 被清除) 时，NBYTES 字段是一个 30 位矢量。

当次循环映射禁用 (EMLM 为 0) 时，所有 32 位 TCD_n word2 均分配至 NBYTES 字段。

地址: 4000_8000h 基准 + 0h 偏移 = 4000_8000h





DMA_CR 字段描述

字段	描述
31-18 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
17 CX	取消传输 0 正常操作 1 取消剩余的数据传输。停止执行中的通道,迫使次循环完成。取消在最后写入当前读/写序列后生效。CX 位在取消生效之后实行自我清除。此次取消会以正常方式停用通道,如同次循环已结束。
16 ECX	错误取消传输 0 正常操作 1 采用与 CX 位相同的方式取消剩余的数据传输。停止执行中的通道,迫使次循环完成。取消在最后写入当前读/写序列后生效。ECX 位在取消生效之后实行自我清除。除了取消传输之外,ECX 将取消视作一次错误状况,并因此更新错误状态寄存器(DMAx_ES)和生成一个可选的错误中断。
15-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 EMLM	使能次循环映射 0 已禁用。TCDn.word2 定义为一个 32 位 NBYTES 字段。 1 已使能。TCDn.word2 被重新定义为包含单个使能字段、一个偏移字段和 NBYTES 字段。单个使能字段使得次循环偏移能够应用到源地址、目标地址及这两种地址上。当任一种偏移使能时,NBYTES 字段被缩减。
6 CLM	持续链接模式 注: 如果每个服务请求只有一个次循环迭代,则在有通道与其自身链接时,不要使用持续链接模式,例如,当通道的 NBYTES 值与源或目标的大小相同时。可以简单地通过增加 NBYTES 值来实现相同的数据传输配置文件,这样可以提供更高效、更快速的处理。 0 如果其自身与次循环通道链接,则在再次激活之前,需要经历通道仲裁。 1 如果其自身与次循环通道链接,则在再次激活之前,不会经历通道仲裁。在次循环完成时,如果该通道的次循环通道链接使能且链接通道就是它自身,则通道再次激活。这样可以有效地应用次循环偏移量,并重新开始下一个次循环。
5 HALT	挂起 DMA 操作 0 正常操作 1 停止任何新通道的启动。可完成执行通道任务。此位被清除后,通道继续执行程序。
4 HOE	因错误停止 0 正常操作 1 任何错误都会导致 HALT 位置位。然后,所有服务请求均被忽略,直到 HALT 位被清除。
3 Reserved	此字段为保留字段。 保留

下一页继续介绍此表...

DMA_CR 字段描述 (继续)

字段	描述
2 ERCA	使能循环通道仲裁 0 固定优先级仲裁用于选择通道。 1 循环仲裁用于选择通道。
1 EDBG	使能调试 0 处于调试模式时，DMA 继续运行。 1 处于调试模式时，DMA 暂停启动新的通道。完成正在执行的通道任务。当系统退出调试模式 或者 EDBG 位被清除后，通道继续执行程序。
0 Reserved	此字段为保留字段。 保留

23.3.6 错误状态寄存器 (DMA_ES)

ES 提供有关最后一次记录通道错误的信息。通道错误的原因可能是：

- 配置错误，即：
 - 传输-控制描述符中的违规设置，或
 - 固定仲裁中的一个违规优先级寄存器设置
- 总线主机读取或写入周期出现错误终止
- 当传输通过相应的取消传输控制位取消时，相应的错误位会被置位

更多详情，请参见[故障报告和处理](#)。

地址: 4000_8000h 基准 + 4h 偏移 = 4000_8004h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	0														ECX
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CPE	0	ERRCHN				SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE	
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_ES 字段描述

字段	描述
31 VLD	所有 ERR 状态位的逻辑 OR 0 没有 ERR 位置位。 1 至少有一个 ERR 位置位，表示有一个有效的错误未被清除。

下一页继续介绍此表...

DMA_ES 字段描述 (继续)

字段	描述
30-17 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
16 ECX	传输已取消 0 没有已取消的传输 1 记录的最后一个条目是由错误引起的取消传输
15 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
14 CPE	通道优先级错误 0 没有通道优先级错误 1 最后一个记录的错误是通道优先级配置错误。通道优先级并不是唯一的。
13-12 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
11-8 ERRCHN	错误通道编号或取消的通道编号 最后一个记录错误的通道编号，不包含 CPE 错误，或最后一个记录的错误取消传输。
7 SAE	源地址错误 0 没有源地址配置错误。 1 最后一个记录的错误是在 TCDn_SADDR 字段中检测到的配置错误。TCDn_SADDR 与 TCDn_ATTR[SSIZE]不一致。
6 SOE	源偏移错误 0 没有源偏移配置错误 1 最后一个记录的错误是在 TCDn_SOFF 字段中检测到的配置错误。TCDn_SOFF 与 TCDn_ATTR[SSIZE]不一致。
5 DAE	目标地址错误 0 没有目标地址配置错误 1 最后一个记录的错误是在 TCDn_DADDR 字段中检测到的配置错误。TCDn_DADDR 与 TCDn_ATTR[DSIZE]不一致。
4 DOE	目标偏移错误 0 没有目标偏移配置错误 1 最后一个记录的错误是在 TCDn_DOFF 字段中检测到的配置错误。TCDn_DOFF 与 TCDn_ATTR[DSIZE]不一致。
3 NCE	NBYTES/CITER 配置错误 0 没有 NBYTES/CITER 配置错误 1 最后一个记录的错误是在 TCDn_NBYTES 或 TCDn_CITER 字段中检测到的配置错误。 • TCDn_NBYTES 不是 TCDn_ATTR[SSIZE] 和 TCDn_ATTR[DSIZE]的倍数，或者 • TCDn_CITER[CITER]等于 0，或 • TCDn_CITER[ELINK]不等于 TCDn_BITER[ELINK]
2 SGE	分散/集中配置错误 0 没有分散/集中配置错误 1 最后一个记录的错误是在 TCDn_DLASTSGA 字段中检测到的配置错误。如果 TCDn_CSR[ESG]使能，则在主循环完成之后、分散/集中操作开始时检查此字段。TCDn_DLASTSGA 不是 32 字节对齐。

下一页继续介绍此表...

DMA_ES 字段描述 (继续)

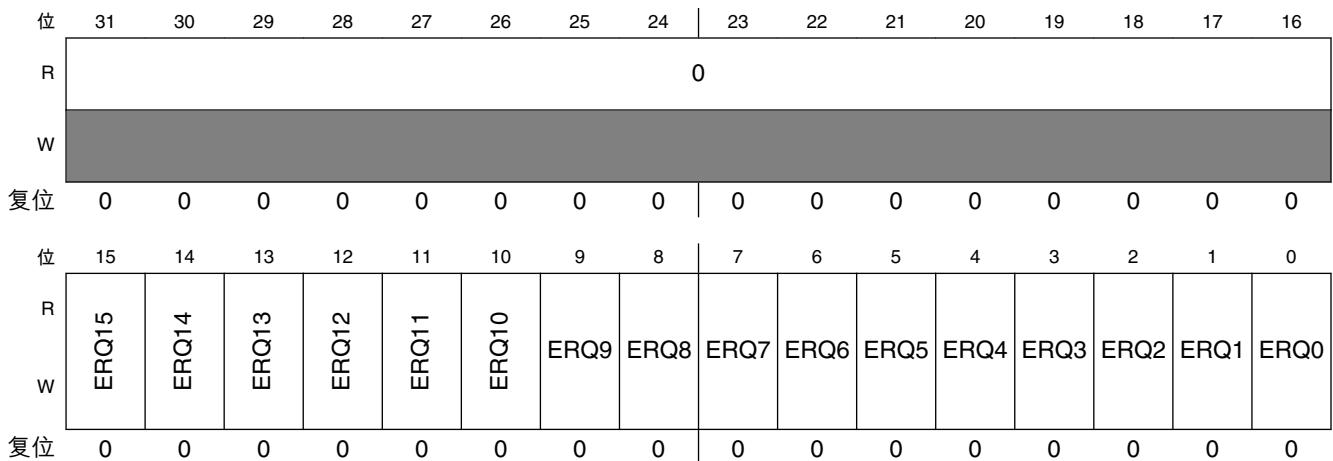
字段	描述
1 SBE	源总线错误 0 没有源总线错误 1 最后一个记录的错误是源读取的总线错误
0 DBE	目标总线错误 0 没有目标总线错误 1 最后一个记录的错误是目标写入的总线错误

23.3.7 使能请求寄存器 (DMA_ERQ)

ERQ 寄存器 提供一个 16 通道的位映射，以便为每个通道使能请求信号。任何给定通道使能的状态直接受向此寄存器的写入影响；它还受向 SERQ 和 CERQ 寄存器的写入影响。提供这些寄存器之后，无需对 ERQ 执行读-更改-写序列操作，即可轻松更改单一通道的请求使能。

DMA 请求输入信号及此使能请求标志位必须在通道的硬件服务请求被受理之前置位有效。DMA 使能请求标志位的状态不会对由软件或链接通道请求明确给出的通道服务请求造成影响。

地址: 4000_8000h 基准 + Ch 偏移 = 4000_800Ch



DMA_ERQ 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15 ERQ15	使能 DMA 请求 15 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能

下一页继续介绍此表...

DMA_ERQ 字段描述 (继续)

字段	描述
14 ERQ14	使能 DMA 请求 14 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
13 ERQ13	使能 DMA 请求 13 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
12 ERQ12	使能 DMA 请求 12 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
11 ERQ11	使能 DMA 请求 11 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
10 ERQ10	使能 DMA 请求 10 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
9 ERQ9	使能 DMA 请求 9 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
8 ERQ8	使能 DMA 请求 8 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
7 ERQ7	使能 DMA 请求 7 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
6 ERQ6	使能 DMA 请求 6 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
5 ERQ5	使能 DMA 请求 5 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
4 ERQ4	使能 DMA 请求 4 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
3 ERQ3	使能 DMA 请求 3 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能

下一页继续介绍此表...

DMA_ERQ 字段描述 (继续)

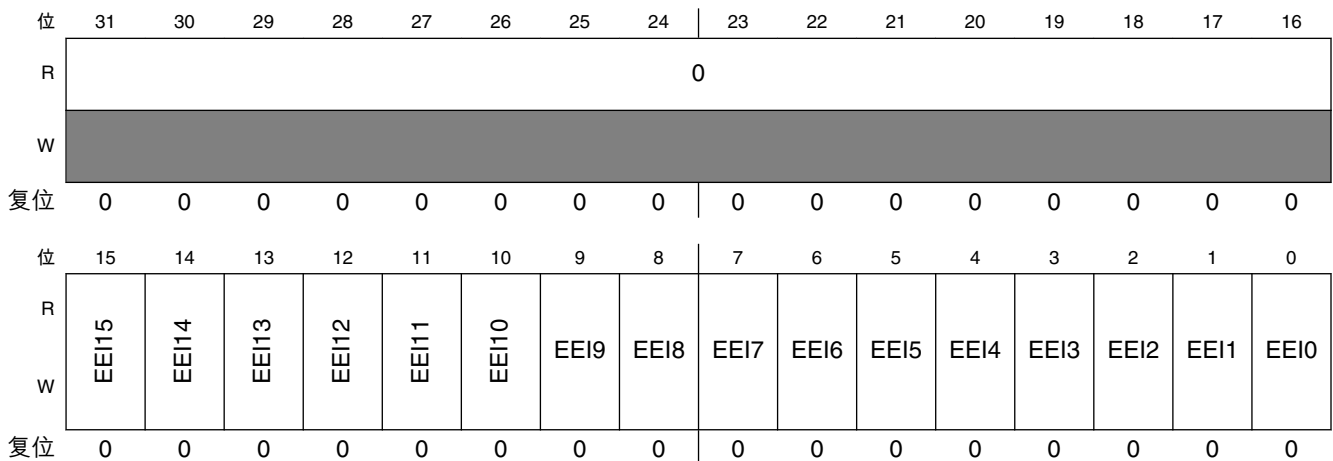
字段	描述
2 ERQ2	使能 DMA 请求 2 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
1 ERQ1	使能 DMA 请求 1 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能
0 ERQ0	使能 DMA 请求 0 0 相应通道的 DMA 请求信号已禁用 1 相应通道的 DMA 请求信号已使能

23.3.8 使能错误中断寄存器 (DMA_EEI)

EEI 寄存器提供一个 16 通道的位映射, 以便为每个通道使能错误中断信号。任何给定通道的错误中断使能的状态直接受向此寄存器的写入影响; 它还受向 SEEI 和 CEEI 的写入影响。提供这些寄存器之后, 无需对 EEI 寄存器执行读-更改-写序列操作, 即可轻松更改单一通道的错误中断使能。

DMA 错误指示器和错误中断使能标志位必须在一个给定通道的错误中断请求被置有效 (至中断控制器) 之前生效。

地址: 4000_8000h 基准 + 14h 偏移 = 4000_8014h



DMA_EEI 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

DMA_EEI 字段描述 (继续)

字段	描述
15 EEI15	使能错误中断 15 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
14 EEI14	使能错误中断 14 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
13 EEI13	使能错误中断 13 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
12 EEI12	使能错误中断 12 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
11 EEI11	使能错误中断 11 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
10 EEI10	使能错误中断 10 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
9 EEI9	使能错误中断 9 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
8 EEI8	使能错误中断 8 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
7 EEI7	使能错误中断 7 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
6 EEI6	使能错误中断 6 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
5 EEI5	使能错误中断 5 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
4 EEI4	使能错误中断 4 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求

下一页继续介绍此表...

DMA_EEI 字段描述 (继续)

字段	描述
3 EEI3	使能错误中断 3 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
2 EEI2	使能错误中断 2 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
1 EEI1	使能错误中断 1 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求
0 EEI0	使能错误中断 0 0 相应通道的错误信号不会生成一个错误中断 1 相应通道的错误信号生效会生成一个错误中断请求

23.3.9 清除使能错误中断寄存器 (DMA_CEEI)

CEEI 提供一种简单的存储器映射机制来清除 EEI 中的给定位，以便禁用某个给定通道的错误中断。寄存器写入的数据值导致 EEI 中相应的位被清除。置位 CAEE 位提供了全局清除功能，迫使 EEI 内容被清除，从而禁用所有的 DMA 请求输入。如果 NOP 位置位，则命令被忽略。这样可以将多字节寄存器的写操作在一个 32 位的寄存器里实现。对此寄存器的读取都将返回零。

地址: 4000_8000h 基准 + 18h 偏移 = 4000_8018h

位	7	6	5	4	3	2	1	0
读	0	0			0			
写	NOP	CAEE	0		CEEI			
复位	0	0	0	0	0	0	0	0

DMA_CEEI 字段描述

字段	描述
7 NOP	没有操作使能 0 正常操作 1 没有操作，忽略此寄存器中的其他位
6 CAEE	清除所有使能错误中断 0 只清除 CEEI 字段中指定的 EEI 位 1 清除 EEI 中的所有位

下一页继续介绍此表...

DMA_CEEI 字段描述 (继续)

字段	描述
5-4 Reserved	此字段为保留字段。
CEEI	清除使能错误中断 清除 EEI 中相应的位

23.3.10 置位使能错误中断寄存器 (DMA_SEEI)

SEEI 提供一种简单的存储器映射机制来置位 EEI 中的给定位, 以便使能某个给定通道的错误中断。寄存器写入的数据值导致 EEI 中相应的位置位。置位 SAEE 位提供全局置位功能, 迫使整个 EEI 内容置位。如果 NOP 位置位, 则命令被忽略。这样可以将多字节寄存器的写操作在一个 32 位的寄存器里实现。对此寄存器的读取都将返回零。

地址: 4000_8000h 基准 + 19h 偏移 = 4000_8019h

位	7	6	5	4	3	2	1	0
读	0	0	0		0			
写	NOP	SAEE	0		SEEI			
复位	0	0	0	0	0	0	0	0

DMA_SEEI 字段描述

字段	描述
7 NOP	没有操作使能 0 正常操作 1 没有操作, 忽略此寄存器中的其他位
6 SAEE	置位所有使能错误中断 0 只置位 SEEI 字段中指定的 EEI 位 1 置位 EEI 中的所有位
5-4 Reserved	此字段为保留字段。
SEEI	置位使能错误中断 置位 EEI 中相应的位

23.3.11 清除使能请求寄存器 (DMA_CERQ)

CERQ 提供一种简单的存储器映射机制来清除 ERQ 中的给定位, 以便禁用某个给定通道的 DMA 请求。寄存器写入的数据值导致 ERQ 中相应的位被清除。置位 CAER 位提供全局清除功能, 迫使 ERQ 的整个内容被清除, 从而禁用所有的 DMA 请求输入。如果 NOP 置位, 则命令被忽略。这样可以将多字节寄存器的写操作在一个 32 位的寄存器里实现。对此寄存器的读取都将返回零。

地址: 4000_8000h 基准 + 1Ah 偏移 = 4000_801Ah

位	7	6	5	4	3	2	1	0
读	0	0					0	
写	NOP	CAER		0			CERQ	
复位	0	0	0	0	0	0	0	0

DMA_CERQ 字段描述

字段	描述
7 NOP	没有操作使能 0 正常操作 1 没有操作, 忽略此寄存器中的其他位
6 CAER	清除全部使能请求 0 只清除 CERQ 字段中指定的 ERQ 位 1 清除 ERQ 中的所有位
5-4 Reserved	此字段为保留字段。
CERQ	清除使能请求 清除 ERQ 中相应的位

23.3.12 置位使能请求寄存器 (DMA_SERQ)

SERQ 提供一种简单的存储器映射机制来置位 ERQ 中的给定位，以便使能某个给定通道的 DMA 请求。寄存器写入的数据值导致 ERQ 中相应的位置位。置位 SAER 位提供全局置位功能，迫使整个 ERQ 的整个内容置位。如果 NOP 位置位，则命令被忽略。这样可以将多字节寄存器的写操作在一个 32 位的寄存器里实现。此寄存器的读取都将返回零。

地址: 4000_8000h 基准 + 1Bh 偏移 = 4000_801Bh

位	7	6	5	4	3	2	1	0
读	0	0					0	
写	NOP	SAER		0			SERQ	
复位	0	0	0	0	0	0	0	0

DMA_SERQ 字段描述

字段	描述
7 NOP	没有操作使能 0 正常操作 1 没有操作，忽略此寄存器中的其他位
6 SAER	置位全部使能请求 0 只置位 SERQ 字段中指定的 ERQ 位 1 置位 ERQ 中的所有位
5-4 Reserved	此字段为保留字段。
SERQ	置位使能请求 置位 ERQ 中相应的位

23.3.13 清除 DONE 状态位寄存器 (DMA_CDNE)

CDNE 提供一种简单的存储器映射机制来清除给定通道 TCD 中的 DONE 位。寄存器写入的数据值导致相应的传输控制描述符的 DONE 位被清除。置位 CADN 位提供全局清除功能，迫使所有 DONE 位被清除。如果 NOP 位置位，则命令被忽略。这样可以将多字节寄存器的写操作在一个 32 位的寄存器里实现。对此寄存器的读取都将返回零。

地址: 4000_8000h 基准 + 1Ch 偏移 = 4000_801Ch

位	7	6	5	4	3	2	1	0
读	0	0					0	
写	NOP	CADN	0				CDNE	
复位	0	0	0	0	0	0	0	0

DMA_CDNE 字段描述

字段	描述
7 NOP	没有操作使能 0 正常操作 1 没有操作，忽略此寄存器中的其他位
6 CADN	清除所有 DONE 位 0 只清除 CDNE 字段中指定的 TCDn_CSR[DONE] 位 1 清除 TCDn_CSR[DONE]中所有的位
5-4 Reserved	此字段为保留字段。
CDNE	清除 DONE 位 清除 TCDn_CSR[DONE]中相应的位

23.3.14 置位 START 位寄存器 (DMA_SSRT)

SSRT 提供一种简单的存储器映射机制来置位给定通道 TCD 中的 START 位。寄存器写入的数据值导致相应的传输控制描述符的 START 位被置位。置位 SAST 位提供全局置位功能，迫使所有 START 位被置位。如果 NOP 位置位，则命令被忽略。这样可以将多字节寄存器的写操作在一个 32 位的寄存器里实现。对此寄存器的读取都将返回零。

地址: 4000_8000h 基准 + 1Dh 偏移 = 4000_801Dh

位	7	6	5	4	3	2	1	0
读	0	0			0			
写	NOP	SAST	0		SSRT			
复位	0	0	0	0	0	0	0	0

DMA_SSRT 字段描述

字段	描述
7 NOP	没有操作使能 0 正常操作 1 没有操作，忽略此寄存器中的其他位
6 SAST	置位所有 START 位 (激活全部通道) 0 只置位 SSRT 字段中指定的 TCDn_CSR[START] 位 1 置位 TCDn_CSR[START]中所有的位
5-4 Reserved	此字段为保留字段。
SSRT	置位 START 位 置位 TCDn_CSR[START]中相应的位

23.3.15 清除错误寄存器 (DMA_CERR)

CERR 提供一种简单的存储器映射机制来清除 ERR 中的给定位，以便禁用某个给定通道的错误状态标志位。寄存器写入的给定值导致 ERR 中相应的位被清除。置位 CAEI 位提供全局清除功能，迫使 ERR 内容被清除，从而清除所有的通道错误指示器。如果 NOP 位置位，则命令被忽略。这样可以将多字节寄存器的写操作在一个 32 位的寄存器里实现。对此寄存器的读取都将返回零。

地址: 4000_8000h 基准 + 1Eh 偏移 = 4000_801Eh

位	7	6	5	4	3	2	1	0
读	0	0					0	
写	NOP	CAEI		0			CERR	
复位	0	0	0	0	0	0	0	0

DMA_CERR 字段描述

字段	描述
7 NOP	没有操作使能 0 正常操作 1 没有操作，忽略此寄存器中的其他位
6 CAEI	清除所有错误指示器 0 只清除 CERR 字段中指定的 ERR 位 1 清除 ERR 中的所有位
5-4 Reserved	此字段为保留字段。
CERR	清除错误指示器 清除 ERR 中相应的位

23.3.16 清除中断请求寄存器 (DMA_CINT)

CINT 提供一种简单的存储器映射机制来清除 INT 中的给定位，以便禁用某个给定通道的中断请求。寄存器写入的给定值导致 INT 中相应的位被清除。置位 CAIR 位提供全局清除功能，迫使 INT 的整个内容被清除，从而禁用所有的 DMA 中断请求。如果 NOP 位置位，则命令被忽略。这样可以将多字节寄存器的写操作在一个 32 位的寄存器里实现。对此寄存器的读取都将返回零。

地址: 4000_8000h 基准 + 1Fh 偏移 = 4000_801Fh

位	7	6	5	4	3	2	1	0
读	0	0					0	
写	NOP	CAIR	0				CINT	
复位	0	0	0	0	0	0	0	0

DMA_CINT 字段描述

字段	描述
7 NOP	没有操作使能 0 正常操作 1 没有操作，忽略此寄存器中的其他位
6 CAIR	清除所有中断请求 0 只清除 CINT 字段中指定的 INT 位 1 清除 INT 中的所有位
5-4 Reserved	此字段为保留字段。
CINT	清除中断请求 清除 INT 中相应的位

23.3.17 中断请求寄存器 (DMA_INT)

INT 寄存器提供一个适用于 16 通道的位映射，表示每个通道存在一个中断请求。根据传输控制描述符中适当的位设置，eDMA 引擎会在数据传输完成时生成一个中断。此寄存器的输出直接路由至中断控制器。在任何给定通道的相关中断服务例程中，软件必须负责清除对应的位，清除中断请求。一般情况下，在中断服务例程中，向 CINT 寄存器写入用于此目的。

任何给定通道的中断请求状态直接受向此寄存器的写入影响；它还受向 CINT 寄存器的写入影响。一旦写入 INT，任何位位置中的 1 都会清除相应通道的中断请求。在任何位位置上的 0 都不会对相应通道的当前中断状态造成影响。提供 CINT 寄存器之后，无需对 INT 寄存器执行读-更改-写序列操作，即可轻松清除单一通道的中断请求。

地址: 4000_8000h 基准 + 24h 偏移 = 4000_8024h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_INT 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15 INT15	中断请求 15 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
14 INT14	中断请求 14 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
13 INT13	中断请求 13 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
12 INT12	中断请求 12 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
11 INT11	中断请求 11

下一页继续介绍此表...

DMA_INT 字段描述 (继续)

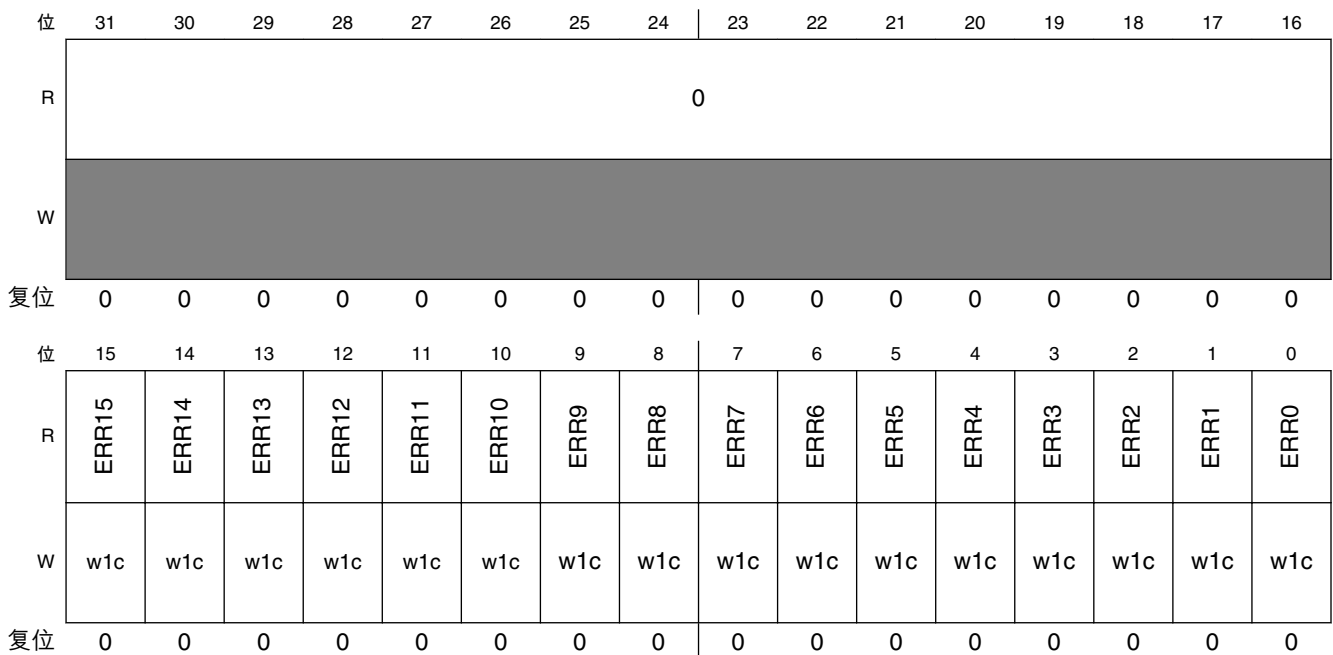
字段	描述
	0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
10 INT10	中断请求 10 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
9 INT9	中断请求 9 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
8 INT8	中断请求 8 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
7 INT7	中断请求 7 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
6 INT6	中断请求 6 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
5 INT5	中断请求 5 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
4 INT4	中断请求 4 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
3 INT3	中断请求 3 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
2 INT2	中断请求 2 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
1 INT1	中断请求 1 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态
0 INT0	中断请求 0 0 相应通道的中断请求已清除 1 相应通道的中断请求处于激活状态

23.3.18 错误寄存器 (DMA_ERR)

ERR 提供一个适用于 16 通道的位映射，表示每个通道存在一个错误。eDMA 通过在此寄存器中置位合适的位来发出存在错误状况的信号。本寄存器的输出由 EEI 内容控制，然后路由至中断控制器。在执行任何 DMA 错误的相关中断服务例程中，软件必须负责清除对应的位，清除错误中断请求。一般情况下，在中断服务例程中，向 CERR 写入用于此目的。普通的 DMA 通道完成标志（传输控制描述符中的 DONE 标志和可能的中断请求）不会被检测到的错误影响。

也可对此寄存器的内容进行轮询，因为一个非零数值表示存在通道错误（不考虑 EEI 的状态）。任何给定通道的错误指示器的状态受向此寄存器的写入影响；它还受向 CERR 的写入影响。一旦写入 ERR，任何位位置中的 1 都会清除相应通道的错误状态。在任何位位置上的 0 都不会对相应通道的当前错误状态造成影响。提供 CERR，以便轻松清除单个通道的错误指示器。

地址: 4000_8000h 基准 + 2Ch 偏移 = 4000_802Ch



DMA_ERR 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15 ERR15	通道 15 中的错误 0 通道中未发生错误 1 通道中发生错误

下一页继续介绍此表...

DMA_ERR 字段描述 (继续)

字段	描述
14 ERR14	通道 14 中的错误 0 通道中未发生错误 1 通道中发生错误
13 ERR13	通道 13 中的错误 0 通道中未发生错误 1 通道中发生错误
12 ERR12	通道 12 中的错误 0 通道中未发生错误 1 通道中发生错误
11 ERR11	通道 11 中的错误 0 通道中未发生错误 1 通道中发生错误
10 ERR10	通道 10 中的错误 0 通道中未发生错误 1 通道中发生错误
9 ERR9	通道 9 中的错误 0 通道中未发生错误 1 通道中发生错误
8 ERR8	通道 8 中的错误 0 通道中未发生错误 1 通道中发生错误
7 ERR7	通道 7 中的错误 0 通道中未发生错误 1 通道中发生错误
6 ERR6	通道 6 中的错误 0 通道中未发生错误 1 通道中发生错误
5 ERR5	通道 5 中的错误 0 通道中未发生错误 1 通道中已发生错误
4 ERR4	通道 4 中的错误 0 通道中未发生错误 1 通道中发生错误
3 ERR3	通道 3 中的错误 0 通道中未发生错误 1 通道中发生错误

下一页继续介绍此表...

DMA_ERR 字段描述 (继续)

字段	描述
2 ERR2	通道 2 中的错误 0 通道中未发生错误 1 通道中发生错误
1 ERR1	通道 1 中的错误 0 通道中未发生错误 1 通道中发生错误
0 ERR0	通道 0 中的错误 0 通道中未发生错误 1 通道中发生错误

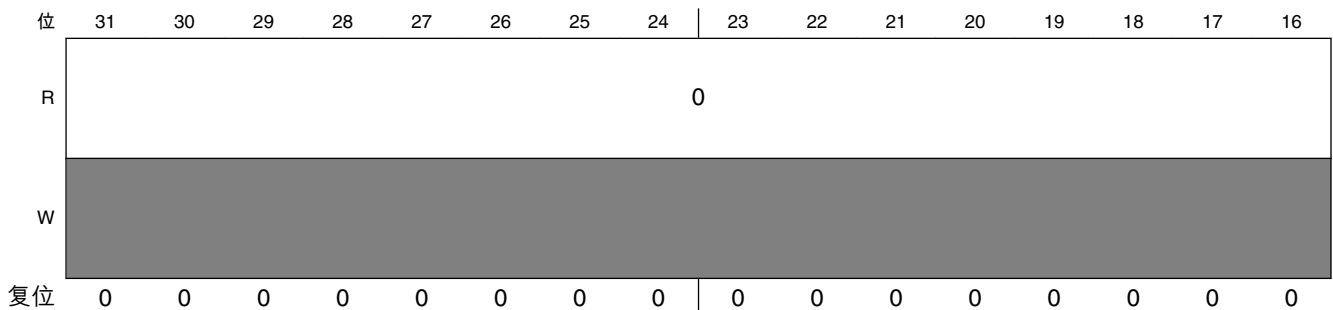
23.3.19 硬件请求状态寄存器 (DMA_HRS)

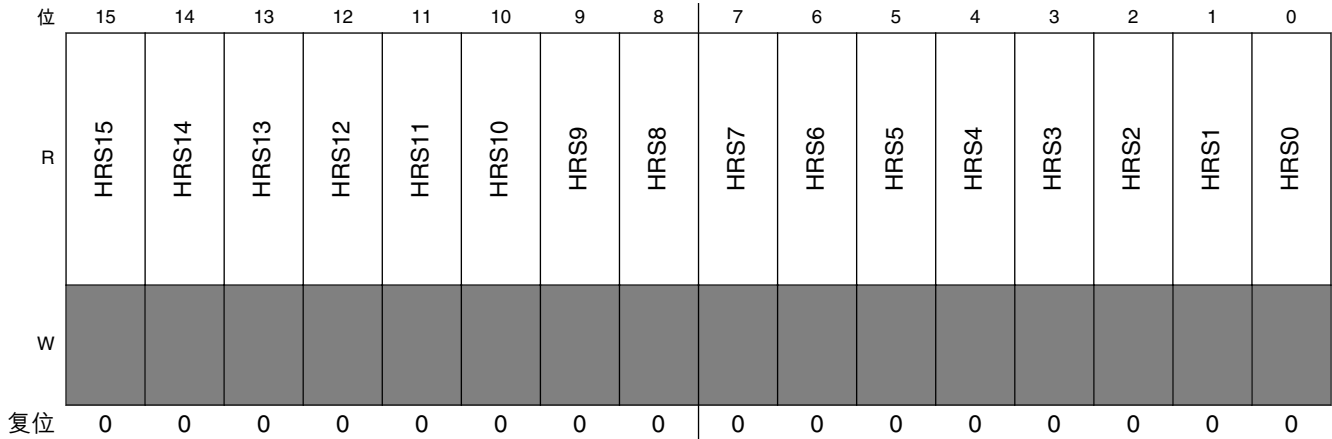
HRS 寄存器提供一个适用于 DMA 通道的位映射，表示每个通道存在一个硬件请求。这些硬件请求状态位反映了当前寄存器的状态以及从 DMA 仲裁逻辑看到的 DMA 请求信号（通过 ERQ）。该寄存器主要用于调试。

注

这些位反映了由仲裁逻辑验证的请求状态。因此，此状态受到 ERQ 位的影响。

地址: 4000_8000h 基准 + 34h 偏移 = 4000_8034h





DMA_HRS 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15 HRS15	硬件请求状态通道 15 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 15 的硬件服务请求 1 存在针对通道 15 的硬件服务请求
14 HRS14	硬件请求状态通道 14 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 14 的硬件服务请求 1 存在针对通道 14 的硬件服务请求
13 HRS13	硬件请求状态通道 13 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 13 的硬件服务请求 1 存在针对通道 13 的硬件服务请求
12 HRS12	硬件请求状态通道 12 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 12 的硬件服务请求 1 存在针对通道 12 的硬件服务请求
11 HRS11	硬件请求状态通道 11 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。

下一页继续介绍此表...

DMA_HRS 字段描述 (继续)

字段	描述
	0 不存在针对通道 11 的硬件服务请求 1 存在针对通道 11 的硬件服务请求
10 HRS10	硬件请求状态通道 10 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 10 的硬件服务请求 1 存在针对通道 10 的硬件服务请求
9 HRS9	硬件请求状态通道 9 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 9 的硬件服务请求 1 存在针对通道 9 的硬件服务请求
8 HRS8	硬件请求状态通道 8 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 8 的硬件服务请求 1 存在针对通道 8 的硬件服务请求
7 HRS7	硬件请求状态通道 7 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 7 的硬件服务请求 1 存在针对通道 7 的硬件服务请求
6 HRS6	硬件请求状态通道 6 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 6 的硬件服务请求 1 存在针对通道 6 的硬件服务请求
5 HRS5	硬件请求状态通道 5 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 5 的硬件服务请求 1 存在针对通道 5 的硬件服务请求
4 HRS4	硬件请求状态通道 4 其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。 0 不存在针对通道 4 的硬件服务请求 1 存在针对通道 4 的硬件服务请求

下一页继续介绍此表...

DMA_HRS 字段描述 (继续)

字段	描述
3 HRS3	<p>硬件请求状态通道 3</p> <p>其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。</p> <p>0 不存在针对通道 3 的硬件服务请求 1 存在针对通道 3 的硬件服务请求</p>
2 HRS2	<p>硬件请求状态通道 2</p> <p>其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。</p> <p>0 不存在针对通道 2 的硬件服务请求 1 存在针对通道 2 的硬件服务请求</p>
1 HRS1	<p>硬件请求状态通道 1</p> <p>其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。</p> <p>0 不存在针对通道 1 的硬件服务请求 1 存在针对通道 1 的硬件服务请求</p>
0 HRS0	<p>硬件请求状态通道 0</p> <p>其各自通道的 HRS 位在通道上存在硬件请求期间保持有效。在请求已完成且通道空闲之后，硬件会自动清除 HRS 位。</p> <p>0 不存在针对通道 0 的硬件服务请求 1 存在针对通道 0 的硬件服务请求</p>

23.3.20 使能在停止状态下的异步请求寄存器 (DMA_EARS)

地址: 4000_8000h 基准 + 44h 偏移 = 4000_8044h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDREQ_15	EDREQ_14	EDREQ_13	EDREQ_12	EDREQ_11	EDREQ_10	EDREQ_9	EDREQ_8	EDREQ_7	EDREQ_6	EDREQ_5	EDREQ_4	EDREQ_3	EDREQ_2	EDREQ_1	EDREQ_0
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_EARS 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15 EDREQ_15	在停止模式下，使能通道 15 的异步 DMA 请求 0 禁用通道 15 的异步 DMA 请求。 1 使能通道 15 的异步 DMA 请求。
14 EDREQ_14	在停止模式下，使能通道 14 的异步 DMA 请求 0 禁用通道 14 的异步 DMA 请求。 1 使能通道 14 的异步 DMA 请求。
13 EDREQ_13	在停止模式下，使能通道 13 的异步 DMA 请求 0 禁用通道 13 的异步 DMA 请求。 1 使能通道 13 的异步 DMA 请求。
12 EDREQ_12	在停止模式下，使能通道 12 的异步 DMA 请求 0 禁用通道 12 的异步 DMA 请求。 1 使能通道 12 的异步 DMA 请求。
11 EDREQ_11	在停止模式下，使能通道 11 的异步 DMA 请求 0 禁用通道 11 的异步 DMA 请求。 1 使能通道 11 的异步 DMA 请求。
10 EDREQ_10	在停止模式下，使能通道 10 的异步 DMA 请求 0 禁用通道 10 的异步 DMA 请求。 1 使能通道 10 的异步 DMA 请求。
9 EDREQ_9	在停止模式下，使能通道 9 的异步 DMA 请求 0 禁用通道 9 的异步 DMA 请求。 1 使能通道 9 的异步 DMA 请求。
8 EDREQ_8	在停止模式下，使能通道 8 的异步 DMA 请求 0 禁用通道 8 的异步 DMA 请求。 1 使能通道 8 的异步 DMA 请求。
7 EDREQ_7	在停止模式下，使能通道 7 的异步 DMA 请求 0 禁用通道 7 的异步 DMA 请求。 1 使能通道 7 的异步 DMA 请求。
6 EDREQ_6	在停止模式下，使能通道 6 的异步 DMA 请求 0 禁用通道 6 的异步 DMA 请求。 1 使能通道 6 的异步 DMA 请求。
5 EDREQ_5	在停止模式下，使能通道 5 的异步 DMA 请求 0 禁用通道 5 的异步 DMA 请求。 1 使能通道 5 的异步 DMA 请求。
4 EDREQ_4	在停止模式下，使能通道 4 的异步 DMA 请求

下一页继续介绍此表...

DMA_EARS 字段描述 (继续)

字段	描述
	0 禁用通道 4 的异步 DMA 请求。 1 使能通道 4 的异步 DMA 请求。
3 EDREQ_3	在停止模式下, 使能通道 3 的异步 DMA 请求。 0 禁用通道 3 的异步 DMA 请求。 1 使能通道 3 的异步 DMA 请求。
2 EDREQ_2	在停止模式下, 使能通道 2 的异步 DMA 请求。 0 禁用通道 2 的异步 DMA 请求。 1 使能通道 2 的异步 DMA 请求。
1 EDREQ_1	在停止模式下, 使能通道 1 的异步 DMA 请求。 0 禁用通道 1 的异步 DMA 请求 1 使能通道 1 的异步 DMA 请求。
0 EDREQ_0	在停止模式下, 使能通道 0 的异步 DMA 请求。 0 禁用通道 0 的异步 DMA 请求。 1 使能通道 0 的异步 DMA 请求。

23.3.21 通道 n 优先级寄存器 (DMA_DCHPRIn)

使能固定优先级通道仲裁 (CR[ERCA] = 0)后, 这些寄存器的内容定义与每个通道关联的唯一优先级。通道优先级以数值评估; 例如, 0 表示最低优先级, 1 表示高一级的优先级, 依次类推。软件必须设定通道优先级的唯一值; 否则, 则可能会报告出现配置错误。优先级的数值范围为 0 至 15。

Address: 4000_8000h base + 100h offset + (1d × i), where i=0d to 15d

位	7	6	5	4	3	2	1	0
读	ECP	DPA	0		CHPRI			
写								
复位	0	0	0	0	*	*	*	*

* 注:

- CHPRI 字段: 参见位字段描述。

DMA_DCHPRIn 字段描述

字段	描述
7 ECP	使能通道抢占。 0 较高优先级通道的服务请求不能使通道 n 挂起。 1 较高优先级通道的服务请求可以使通道 n 暂时挂起

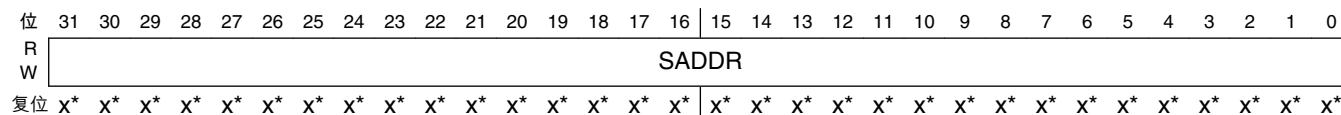
下一页继续介绍此表...

DMA_DCHPRI_n 字段描述 (继续)

字段	描述
6 DPA	禁用抢占能力。 0 通道 n 可挂起较低优先级通道。 1 无论通道是何种优先级，通道 n 都不能挂起任何通道。
5-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CHPRI	通道 n 仲裁优先级 固定优先级仲裁使能时的通道优先级 注：通道优先级字段 CHPRI 的复位值等于每个优先级寄存器的对应通道编号，也就是说， DCHPRI15[CHPRI] = 0b1111。

23.3.22 TCD 源地址 (DMA_TCD_n_SADDR)

Address: 4000_8000h base + 1000h offset + (32d × i), where i=0d to 15d



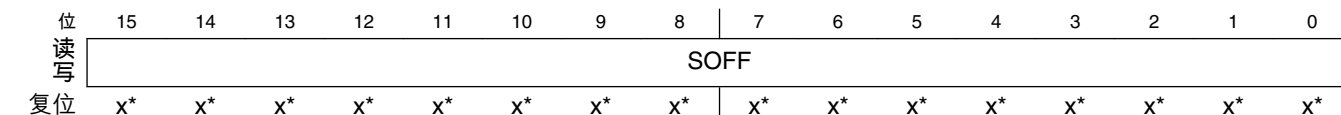
- * 注:
- x = 复位时未定义。

DMA_TCD_n_SADDR 字段描述

字段	描述
SADDR	源地址 指向源数据的存储器地址。

23.3.23 TCD 有符号的源地址偏移 (DMA_TCD_n_SOFF)

Address: 4000_8000h base + 1004h offset + (32d × i), where i=0d to 15d



- * 注:
- x = 复位时未定义。

DMA_TCDn_SOFF 字段描述

字段	描述
SOFF	带符号的源地址偏移 带符号的地址偏移，在每次源读取完成后，加到当前的源地址上，形成下一状态的源地址。

23.3.24 TCD 传输属性 (DMA_TCDn_ATTR)

Address: 4000_8000h base + 1006h offset + (32d × i), where i=0d to 15d

位 读 写	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SMOD					SSIZE			DMOD				DSIZE			
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* 注:

- x = 复位时未定义。

DMA_TCDn_ATTR 字段描述

字段	描述
15-11 SMOD	源地址模数 0 源地址模数功能已禁用 ≠0 此值限定了一个特定的地址范围，即对原寄存器值执行 SADDR + SOFF 计算之后指定的值。置位此字段之后，即可轻松实施一个循环数据队列。对于需要 2 的幂数字节的数据队列，应以 0 模数大小的地址开始，且应将 SMOD 字段设置为适用于队列的数值，以此保持所需高位地址位不变。为此字段设置的数值指定了可更改的低位地址位的数目。对于循环队列应用，SOFF 一般设置为传输大小，以便在 SMOD 功能将地址限制在 0 模数大小范围内时，执行后增寻址操作。
10-8 SSIZE	源数据传输大小 注：使用保留值会导致配置错误。 000 8 位 001 16 位 010 32 位 011 保留 100 16 字节 突发 101 32 字节 突发 110 保留 111 保留
7-3 DMOD	目标地址模数 查看 SMOD 定义
DSIZE	目标数据传输大小 查看 SSIZE 定义

23.3.25 TCD 次字节数目 (次循环映射禁用) (DMA_TCDn_NBYTES_MLNO)

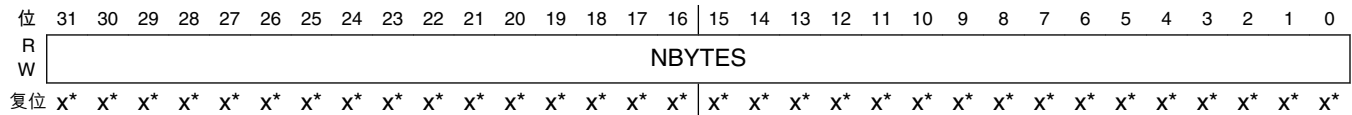
此寄存器或随后两个寄存器 (TCD_NBYTES_MLOFFNO、TCD_NBYTES_MLOFFYES) 中的一个指定了每次请求传输的字节数。具体使用哪个寄存器，取决于次循环映射是否禁用、启用但不用于此通道，或已启用且已使用。

在以下情况下，TCD 字 2 的定义如下：

- 次循环映射已禁用 (CR[EMLM] = 0)

如果次循环映射已使能，请查看 TCD_NBYTES_MLOFFNO 和 TCD_NBYTES_MLOFFYES 寄存器描述符了解 TCD 字 2 的定义。

Address: 4000_8000h base + 1008h offset + (32d × i), where i=0d to 15d



- *注:
- x = 复位时未定义。

DMA_TCDn_NBYTES_MLNO 字段描述

字段	描述
NBYTES	<p>次字节传输数目</p> <p>通道的每个服务请求会传输的字节数目。通道激活时，适当的 TCD 内容会加载到 eDMA 引擎，并且执行适当的读写操作，直到传输了次字节传输数目。此操作不可分割，不可中途停止。但是，可通过带宽控制字段或通过抢占加以延迟。次字节传输完毕之后，SADDR 和 DADDR 数值会回写入 TCD 存储器，主迭代数目会减少并回存入 TCD 存储器。如果主迭代数目传输完毕，则会进行额外处理。</p> <p>注：当 NBYTES 数值为 0x0000_0000 时，NBYTES 会被解析为一个 4 GB 的传输。</p>

23.3.26 TCD 有符号的次循环偏移 (次循环映射使能, 偏移禁用) (DMA_TCDn_NBYTES_MLOFFNO)

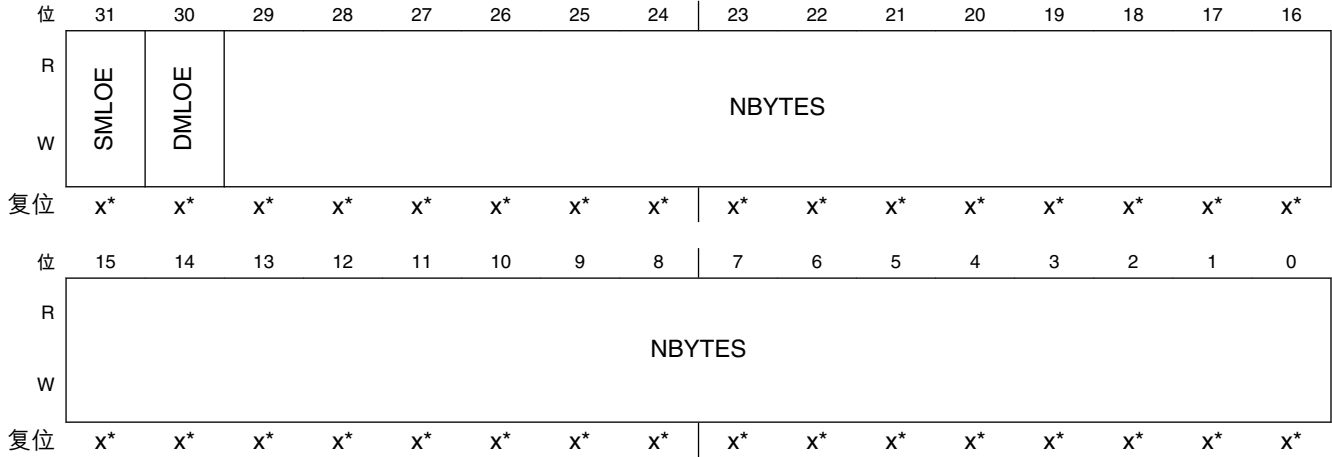
三个寄存器 (此寄存器、TCD_NBYTES_MLNO 或 TCD_NBYTES_MLOFFYES) 中的一个指定了每次请求传输的字节数。具体使用哪个寄存器，取决于次循环映射是否禁用、启用但不用于此通道，或已启用且已使用。

在以下情况下，TCD 字 2 的定义如下：

- 次循环映射已使能 (CR[EMLM] = 1) 且
- SMLOE = 0, DMLOE = 0

如果次循环映射已使能且 SMLOE 或 DMLOE 已置位，则参考 TCD_NBYTES_MLOFFYES 寄存器描述。如果次循环映射已禁用，则参考 TCD_NBYTES_MLNO 寄存器描述。

Address: 4000_8000h base + 1008h offset + (32d × i), where i=0d to 15d



* 注:
 • x = 复位时未定义。

DMA_TCDn_NBYTES_MLOFFNO 字段描述

字段	描述
31 SMLOE	源次循环偏移使能 选择是否在次循环完成时对源地址应用次循环偏移。 0 未对 SADDR 应用次循环偏移 1 已对 SADDR 应用次循环偏移
30 DMLOE	目标次循环偏移使能 选择是否在次循环完成时对目标地址应用次循环偏移。 0 未对 DADDR 应用次循环偏移 1 已对 DADDR 应用次循环偏移
NBYTES	次字节传输数目 通道的每个服务请求会传输的字节数目。 通道激活时，适当的 TCD 内容会加载到 eDMA 引擎，并且执行适当的读写操作，直到传输了次字节传输数目。此操作不可分割，不可中途停止。但是，可通过带宽控制字段或通过抢占加以延迟。次字节传输完毕之后，SADDR 和 DADDR 数值会回写入 TCD 存储器，主迭代数目会减少并回存入 TCD 存储器。如果主迭代数目传输完毕，则会进行额外处理。

23.3.27 TCD 有符号的次循环偏移（次循环映射和偏移已使能） (DMA_TCDn_NBYTES_MLOFFYES)

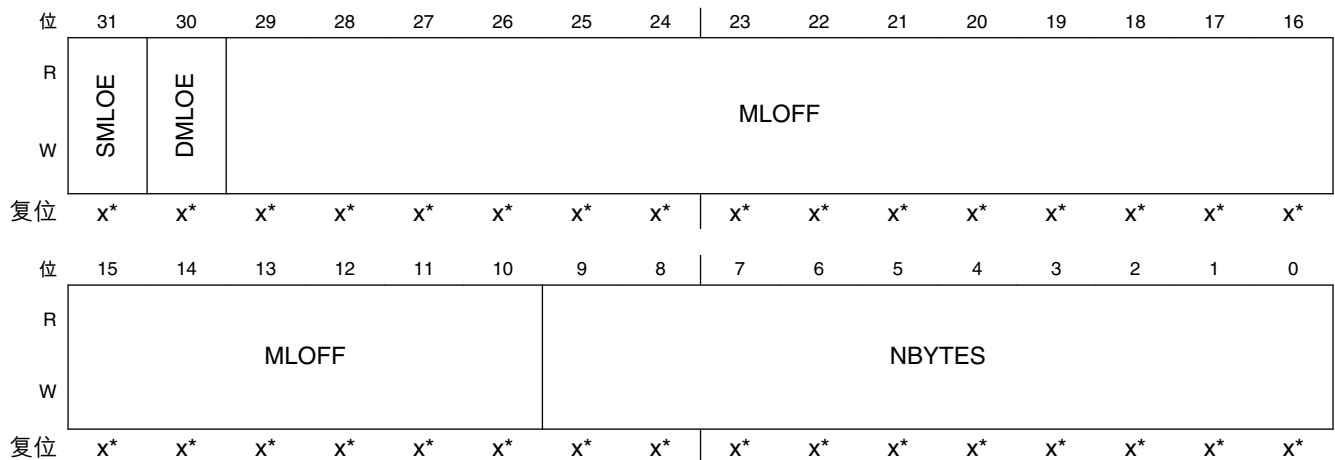
三个寄存器（此寄存器、TCD_NBYTES_MLNO 或 TCD_NBYTES_MLOFFNO）中的一个指定了每次请求传输的字节数。具体使用哪个寄存器，取决于次循环映射是否禁用、启用但不用于此通道，或已启用且已使用。

在以下情况下，TCD 字 2 的定义如下：

- 次循环映射已启用 (CR[EMLM] = 1)且
- 次循环偏移已使能 (SMLOE 或 DMLOE = 1)

如果次循环映射已使能且 SMLOE 和 DMLOE 已被清除，则参考 TCD_NBYTES_MLOFFNO 寄存器描述。如果次循环映射已禁用，则参考 TCD_NBYTES_MLNO 寄存器描述。

Address: 4000_8000h base + 1008h offset + (32d × i), where i=0d to 15d



- *注:
- x = 复位时未定义。

DMA_TCDn_NBYTES_MLOFFYES 字段描述

字段	描述
31 SMLOE	源次循环偏移使能 选择是否在次循环完成时对源地址应用次循环偏移。 0 未对 SADDR 应用次循环偏移 1 已对 SADDR 应用次循环偏移
30 DMLOE	目标次循环偏移使能 选择是否在次循环完成时对目标地址应用次循环偏移。

下一页继续介绍此表...

DMA_TCDn_NBYTES_MLOFFYES 字段描述 (继续)

字段	描述
	0 未对 DADDR 应用次循环偏移 1 已对 DADDR 应用次循环偏移
29–10 MLOFF	如果 SMLOE 或 DMLOE 已置位，则此字段代表一个已应用到源地址或目标地址的带符号扩展偏移，以便在次循环完成后形成下一个状态值。
NBYTES	次字节传输数目 通道的每个服务请求会传输的字节数目。 通道激活时，适当的 TCD 内容会加载到 eDMA 引擎，并且执行适当的读写操作，直到传输了次字节传输数目。此操作不可分割，不可中途停止。但是，可通过带宽控制字段或通过抢占加以延迟。次字节传输完毕之后，SADDR 和 DADDR 数值会回写入 TCD 存储器，主迭代数目会减少并回存入 TCD 存储器。如果主迭代数目传输完毕，则会进行额外处理。

23.3.28 TCD 最后源地址调整 (DMA_TCDn_SLAST)

Address: 4000_8000h base + 100Ch offset + (32d × i), where i=0d to 15d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SLAST																															
W	SLAST																															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	

* 注:

- x = 复位时未定义。

DMA_TCDn_SLAST 字段描述

字段	描述
SLAST	最后源地址调整 在主迭代数目传输完成时添加到源地址的调整值。可应用此值将源地址还原为初始值，或将此地址调整为下一个数据结构的地址。 此寄存器使用了二进制补码记数法；丢弃了溢出位。

23.3.29 TCD 目标地址 (DMA_TCDn_DADDR)

Address: 4000_8000h base + 1010h offset + (32d × i), where i=0d to 15d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DADDR																															
W	DADDR																															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	

* 注:

- x = 复位时未定义。

DMA_TCDn_DADDR 字段描述

字段	描述
DADDR	目标地址 指向目标数据的存储器地址。

23.3.30 TCD 有符号的目标地址偏移 (DMA_TCDn_DOFF)

Address: 4000_8000h base + 1014h offset + (32d × i), where i=0d to 15d



*注:

- x = 复位时未定义。

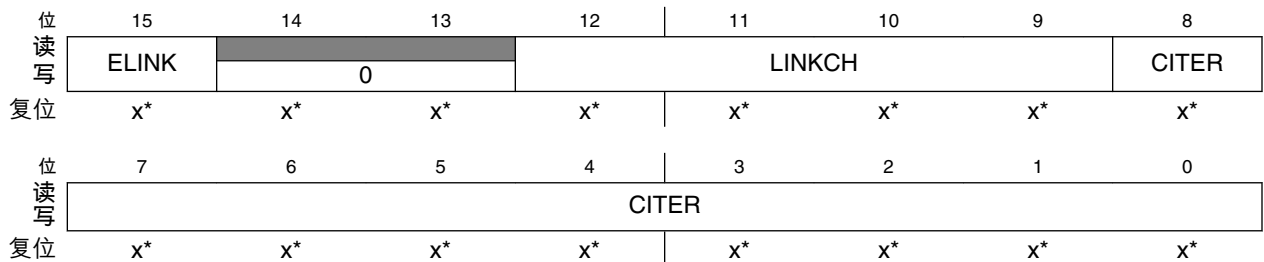
DMA_TCDn_DOFF 字段描述

字段	描述
DOFF	目标地址带符号的偏移 对当前目标地址应用带符号扩展偏移，以便在每个目标写入完成时形成下个状态值。

23.3.31 TCD 当前次循环链接，主循环数目（通道链接使能） (DMA_TCDn_CITER_ELINKYES)

如果 TCDn_CITER[ELINK]置位，则 TCDn_CITER 寄存器的定义如下:

Address: 4000_8000h base + 1016h offset + (32d × i), where i=0d to 15d



*注:

- x = 复位时未定义。

DMA_TCDn_CITER_ELINKYES 字段描述

字段	描述
15 ELINK	<p>在次循环完成时使能通道至通道链接</p> <p>通道完成次循环时，此标志位使能与其他通道（由 LINKCH 字段定义）的连接。链接目标通道通过置位指定通道的 TCDn_CSR[START] 位的这种内部机制来发起一个通道服务请求。</p> <p>如果通道链接被禁用，则 CITER 会扩展到 15 位，覆盖 LINKCH。如果主循环已完成，则此链接机制会因支持 MAJORELINK 通道链接而受到抑制。</p> <p>注：此位必须等于 BITER[ELINK]位；否则，会报告出现一个配置错误。</p> <p>0 通道至通道链接已禁用 1 通道至通道链接已使能</p>
14–13 Reserved	此字段为保留字段。
12–9 LINKCH	<p>次循环链接通道编号</p> <p>如果通道至通道链接已使能(ELINK = 1)，则在次循环完成之后，eDMA 引擎会通过置位该通道的 TCDn_CSR[START] 位向此字段定义的通道发起一个通道服务请求。</p>
CITER	<p>当前主循环迭代数目</p> <p>这个 9 位 (ELINK=1)或 15 位 (ELINK=0) 数目代表了通道的当前主循环数目。此数目在每次完成次循环且在传输控制描述符存储器中更新时都会减少。主迭代数目完成之后，通道会执行几种操作，例如，最后的源和目标地址计算，从开始迭代数目(BITER)字段重新加载 CITER 字段之前，先选择性地生成一个信号通道完成中断。</p> <p>注：当 CITER 字段最初由软件加载时，它所设置的数值必须与 BITER 字段中所含的数值相同。</p> <p>注：如果配置通道是为了执行单个服务请求，则 BITER 和 CITER 的初始值应为 0x0001。</p>

23.3.32 TCD 开始次循环链接，主循环数目（通道链接禁用） (DMA_TCDn_CITER_ELINKNO)

如果 TCDn_CITER[ELINK]被清零，则 TCDn_CITER 寄存器的定义如下：

Address: 4000_8000h base + 1016h offset + (32d × i), where i=0d to 15d

位 读 写	15	14	13	12	11	10	9	8
复位	x*	x*	x*	x*	x*	x*	x*	x*
位 读 写	7	6	5	4	3	2	1	0
复位	x*	x*	x*	x*	x*	x*	x*	x*

* 注:

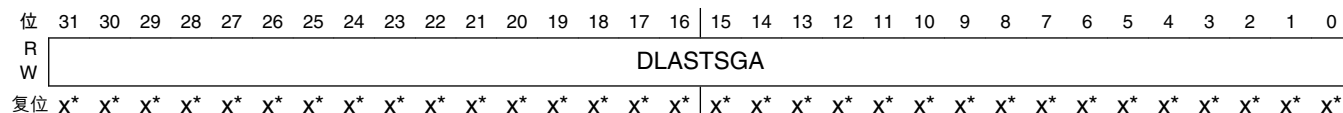
- x = 复位时未定义。

DMA_TCDn_CITER_ELINKNO 字段描述

字段	描述
15 ELINK	<p>在次循环完成时使能通道至通道链接</p> <p>通道完成次循环时，此标志位使能与其他通道（由 LINKCH 字段定义）的连接。链接目标通道通过置位指定通道的 TCDn_CSR[START] 位的这种内部机制来发起一个通道服务请求。</p> <p>如果通道链接被禁用，则 CITER 会扩展到 15 位，覆盖 LINKCH。如果主循环已完成，则此链接机制会因支持 MAJORELINK 通道链接而受到抑制。</p> <p>注：此位必须等于 BITER[ELINK]位；否则，会报告出现一个配置错误。</p> <p>0 通道至通道链接已禁用 1 通道-通道链接已使能</p>
CITER	<p>当前主循环迭代数目</p> <p>这个 9 位 (ELINK=1)或 15 位 (ELINK=0) 数目代表了通道的当前主循环数目。此数目在每次完成次循环且在传输控制描述符存储器中更新时都会减少。主迭代数目完成之后，通道会执行几种操作，例如，最后的源和目标地址计算，从开始迭代数目(BITER)字段重新加载 CITER 字段之前，先选择性地生成一个信号通道完成中断。</p> <p>注：当 CITER 字段最初由软件加载时，它所设置的数值必须与 BITER 字段中所含的数值相同。</p> <p>注：如果配置通道是为了执行单个服务请求，则 BITER 和 CITER 的初始值应为 0x0001。</p>

23.3.33 TCD 最后的目标地址调整/分散集中地址 (DMA_TCDn_DLASTSGA)

Address: 4000_8000h base + 1018h offset + (32d × i), where i=0d to 15d



- * 注:
- x = 复位时未定义。

DMA_TCDn_DLASTSGA 字段描述

字段	描述
DLASTSGA	<p>目标最后地址调整或将加载至此通道（分散/集中）的下一个传输控制描述符的存储器地址。</p> <p>如果 (TCDn_CSR[ESG] = 0)，则：</p> <ul style="list-style-type: none"> • 在主迭代数目传输完成时添加到目标地址的调整值。可应用此值将目标地址还原为初始值，或将此地址调整为下一个数据结构的地址。 • 此字段使用二进制补码记数法来调整最后的目标地址。 <p>否则：</p> <ul style="list-style-type: none"> • 此地址指向 32 字节对齐的，包含待加载到此通道的下一个传输控制描述符的起始地址。此通道在主迭代数目完成时重新加载。分散/集中地址必须为 32 字节对齐，否则会报告出现一个配置错误。

23.3.34 TCD 控制和状态 (DMA_TCDn_CSR)

Address: 4000_8000h base + 101Ch offset + (32d × i), where i=0d to 15d

位	15	14	13	12	11	10	9	8
读	BWC				MAJORLINKCH			
写			0					
复位	x*	x*	x*	x*	x*	x*	x*	x*
位	7	6	5	4	3	2	1	0
读	DONE	ACTIVE	MAJORELINK	ESG	DREQ	INTHALF	INTMAJOR	START
写								
复位	x*	x*	x*	x*	x*	x*	x*	x*

*注:

- x = 复位时未定义。

DMA_TCDn_CSR 字段描述

字段	描述
15-14 BWC	<p>带宽控制</p> <p>节制 eDMA 所消耗的总线带宽的量。通常，当 eDMA 执行次循环时，它会持续生成读/写序列，直到次循环数目完成。此字段迫使 eDMA 在每个读/写访问完成后产生延迟，以便控制交叉开关监控的总线请求带宽。</p> <p>注：如果源和目标的大小相同，则此字段在第一二次传输之间及每个次循环结束之前的最后一次写入会被忽略。这是缩短启动延迟的副作用。</p> <p>00 禁止 eDMA 引擎延迟。 01 保留 10 eDMA 引擎在每次 R/W 后延迟 4 个周期。 11 eDMA 引擎在每次 R/W 后延迟 8 个周期。</p>
13-12 Reserved	此字段为保留字段。
11-8 MAJORLINKCH	<p>主循环链接通道编号</p> <p>如果(MAJORELINK = 0)，则：</p> <ul style="list-style-type: none"> • 在主循环计数器完成之后，不会执行通道至通道链接（或挂链）。 <p>否则：</p> <ul style="list-style-type: none"> • 在主循环计数器完成之后，eDMA 引擎会通过置位该通道的 TCDn_CSR[START] 位在此字段定义的通道上发起一个通道服务请求。
7 DONE	<p>通道完成</p> <p>此标志位表示 eDMA 已完成主循环。eDMA 引擎在 CITER 数目达到 0 时将其置位。由软件将其清除，或者在通道被激活后则由硬件将其清除。</p> <p>注：必须清除此位，以便写入 MAJORELINK 或 ESG 位。</p>
6 ACTIVE	通道激活

下一页继续介绍此表...

DMA_TCDn_CSR 字段描述 (继续)

字段	描述
	此标志位表示此通道当前正在执行服务。它在通道开始服务时置位，在次循环完成时或检测到任何错误状况时由 eDMA 清除。
5 MAJORELINK	<p>在主循环完成时使能通道至通道链接</p> <p>通道完成主循环时，此标志位使能与其他通道（由 MAJORLINKCH 定义）的连接。链接目标通道通过置位指定通道的 TCDn_CSR[START] 位的这种内部机制来发起一个通道服务请求。</p> <p>注：为了支持动态链接一致性模型，此字段在 TCDn_CSR[DONE] 位置位时如被写入，则此字段被强制归 0。</p> <p>0 通道至通道链接已禁用 1 通道至通道链接已使能</p>
4 ESG	<p>使能分散/集中处理</p> <p>通道完成主循环时，此标志位使能当前通道的分散/集中处理。使能之后，eDMA 引擎将 DLASTSGA 用作 32 字节对齐地址（包含一个作为传输控制描述符加载到本地存储器的 32 字节数据结构）的存储器指针。</p> <p>注：为了支持动态分散/集中一致性模型，此字段在 TCDn_CSR[DONE] 位置位时如被写入，则此字段被强制归 0。</p> <p>0 当前通道的 TCD 为正常格式。 1 当前通道的 TCD 指定一种分散集中格式。在主循环执行完成后，DLASTSGA 字段会为下一个加载至此通道的 TCD 提供一个存储器指针。</p>
3 DREQ	<p>禁用请求</p> <p>如果此标志位已置位，则 eDMA 硬件在当前的主迭代数目达到 0 时自动清除相应的 ERQ 位。</p> <p>0 通道的 ERQ 位不受影响。 1 主循环完成时，通道的 ERQ 位被清除。</p>
2 INTHALF	<p>在主迭代数目完成一半时使能一个中断。</p> <p>如果此标志位置位，则通道会在当前的主迭代数目达到中点时在 INT 寄存器置位适当的位，以此生成一个中断请求。特别提醒，eDMA 引擎执行的比较操作为 (CITER == (BITER >> 1))。提供此中点中断请求，以支持双缓冲（也称作 ping-pong）方案或其他类型的数据移动，此时处理器需要及早获得传输进度指示。</p> <p>注：如果 BITER = 1，请不要使用 INTHALF。请使用 INTMAJOR。</p> <p>0 中点中断禁用。 1 中点中断使能。</p>
1 INTMAJOR	<p>在主迭代数目完成时使能一个中断。</p> <p>如果此标志位置位，则通道会在当前的主迭代数目达到零时在 INT 置位适当的位，以此生成一个中断请求。</p> <p>0 主循环结束中断禁用。 1 主循环结束中断使能。</p>
0 START	<p>通道开始</p> <p>如果此标志位置位，则通道正在请求服务。eDMA 硬件在通道开始执行之后，自动清除此标志位。</p> <p>0 通道尚未明确开始。 1 通道已通过软件发起的服务请求明确开始。</p>

23.3.35 TCD 开始次循环链接，主循环数目（通道链接使能） (DMA_TCDn_BITER_ELINKYES)

如果 TCDn_BITER[ELINK]位置位，则 TCDn_BITER 寄存器的定义如下：

Address: 4000_8000h base + 101Eh offset + (32d × i), where i=0d to 15d

位	15	14	13	12	11	10	9	8
读	ELINK		0		LINKCH			BITER
写	x*		x*		x*			x*
复位	x*		x*		x*			x*
位	7	6	5	4	3	2	1	0
读	BITER							
写	x*							
复位	x*							

* 注:

- x = 复位时未定义。

DMA_TCDn_BITER_ELINKYES 字段描述

字段	描述
15 ELINK	<p>在次循环完成时使能通道至通道链接</p> <p>通道完成次循环时，此标志位使能与其他通道（由 BITER[LINKCH]定义）的连接。链接目标通道通过置位指定通道的 TCDn_CSR[START] 位的这种内部机制来发起一个通道服务请求。如果通道链接被禁用，则 CITER 会扩展到 15 位，覆盖 LINKCH。如果主循环已完成，则此链接机制会因支持 MAJORELINK 通道链接而受到抑制。</p> <p>注：软件加载 TCD 时，此字段必须与相应的 CITER 字段采用相同设置；否则，会报告出现一个配置错误。主迭代数目完成时，此字段的内容会重新加载至 CITER 字段。</p> <p>0 通道至通道链接已禁用 1 通道至通道链接已使能</p>
14-13 Reserved	此字段为保留字段。
12-9 LINKCH	<p>链接通道编号</p> <p>如果通道至通道链接已使能(ELINK = 1)，则在次循环完成之后，eDMA 引擎会通过置位该通道的 TCDn_CSR[START] 位在此字段定义的通道上发起一个通道服务请求。</p> <p>注：软件加载 TCD 时，此字段必须与相应的 CITER 字段采用相同设置；否则，会报告出现一个配置错误。主迭代数目完成时，此字段的内容会重新加载至 CITER 字段。</p>
BITER	<p>起始主迭代数目</p> <p>由于传输控制描述符最初由软件加载，因此 9 位(ELINK = 1)或 15 位 (ELINK = 0)字段必须等于 CITER 字段中的值。主迭代数目完成时，此字段的内容会重新加载至 CITER 字段。</p> <p>注：软件加载 TCD 时，此字段必须与相应的 CITER 字段采用相同设置；否则，会报告出现一个配置错误。主迭代数目完成时，此字段的内容会重新加载至 CITER 字段。如果配置通道是为了执行单个服务请求，则 BITER 和 CITER 的初始值应为 0x0001。</p>

23.3.36 TCD 开始次循环链接，主循环数目（通道链接禁用） (DMA_TCDn_BITER_ELINKNO)

如果 TCDn_BITER[ELINK]位被清除，则 TCDn_BITER 寄存器的定义如下：

Address: 4000_8000h base + 101Eh offset + (32d × i), where i=0d to 15d

位	15	14	13	12	11	10	9	8
读写	ELINK	BITER						
复位	x*	x*	x*	x*	x*	x*	x*	x*
位	7	6	5	4	3	2	1	0
读写	BITER							
复位	x*	x*	x*	x*	x*	x*	x*	x*

* 注:

- x = 复位时未定义。

DMA_TCDn_BITER_ELINKNO 字段描述

字段	描述
15 ELINK	<p>在次循环完成时使能通道至通道链接</p> <p>通道完成次循环时，此标志位使能与其他通道（由 BITER[LINKCH]定义）的连接。链接目标通道通过置位指定通道的 TCDn_CSR[START] 位的这种内部机制来发起一个通道服务请求。如果通道链接被禁用，则 CITER 会扩展到 15 位，覆盖 LINKCH。如果主循环已完成，则此链接机制会因支持 MAJORELINK 通道链接而受到抑制。</p> <p>注：软件加载 TCD 时，此字段必须与相应的 CITER 字段采用相同设置；否则，会报告出现一个配置错误。主迭代数目完成时，此字段的内容会重新加载至 CITER 字段。</p> <p>0 通道至通道链接已禁用 1 通道至通道链接已使能</p>
BITER	<p>起始主迭代数目</p> <p>由于传输控制描述符最初由软件加载，因此 9 位 (ELINK = 1) 或 15 位 (ELINK = 0) 字段必须等于 CITER 字段中的值。主迭代数目完成时，此字段的内容会重新加载至 CITER 字段。</p> <p>注：软件加载 TCD 时，此字段必须与相应的 CITER 字段采用相同设置；否则，会报告出现一个配置错误。主迭代数目完成时，此字段的内容会重新加载至 CITER 字段。如果配置通道是为了执行单个服务请求，则 BITER 和 CITER 的初始值应为 0x0001。</p>

23.4 功能说明

以下小节描述了 eDMA 的操作过程。

23.4.1 eDMA 基础数据流

基础数据传输流可分成三个部分。

下图所示的是第一部分，涉及通道的激活：

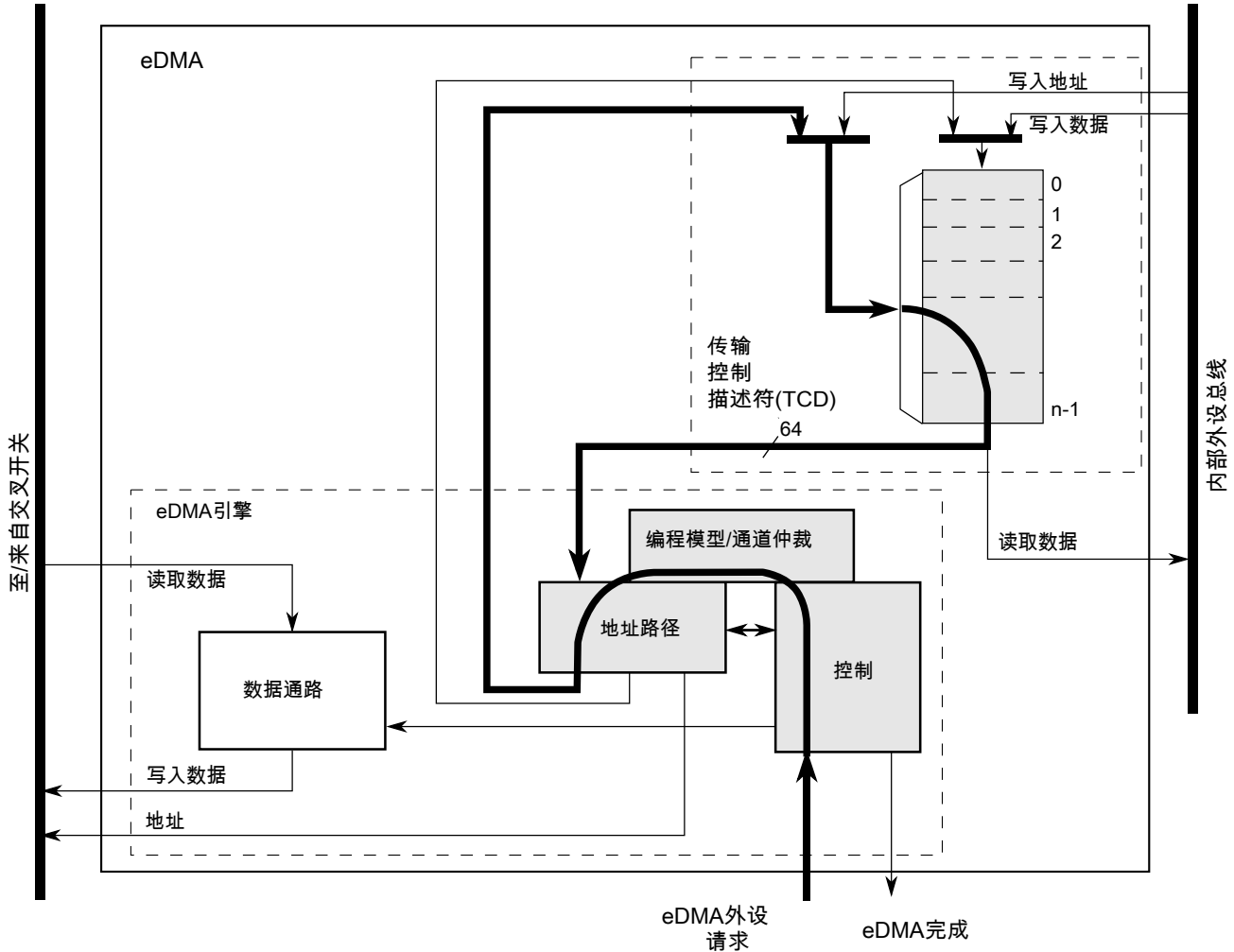


图 23-290. eDMA 操作，第 1 部分

本示例使用 eDMA 外设请求信号的有效性来为通道 n 请求服务。通过软件设置 $TCD_n_CSR[START]$ 位来激活通道的基本流与外设请求的基本流相同。eDMA 请求输入信号在内部注册，然后通过 eDMA 引擎进行路由：先通过控制模块，然后进入编程模型和通道仲裁。在下个循环，按照固定优先级或循环算法执行通道仲裁。完成仲裁后，激活的通道编号会经由地址通路发送，然后转换成访问 TCD_n 本地存储器所需的地址。接着，会访问 TCD 存储器，从本地存储器读取所需的描述符，加载到 eDMA 引擎地址通路通道 x 或 y 寄存器中。TCD 存储器的宽度为 64 位，以便最大限度地缩短获取已激活的通道描述符及其加载至地址通路通道 x 或 y 寄存器所需的时间。

下图显示了基础数据流的第 2 部分：

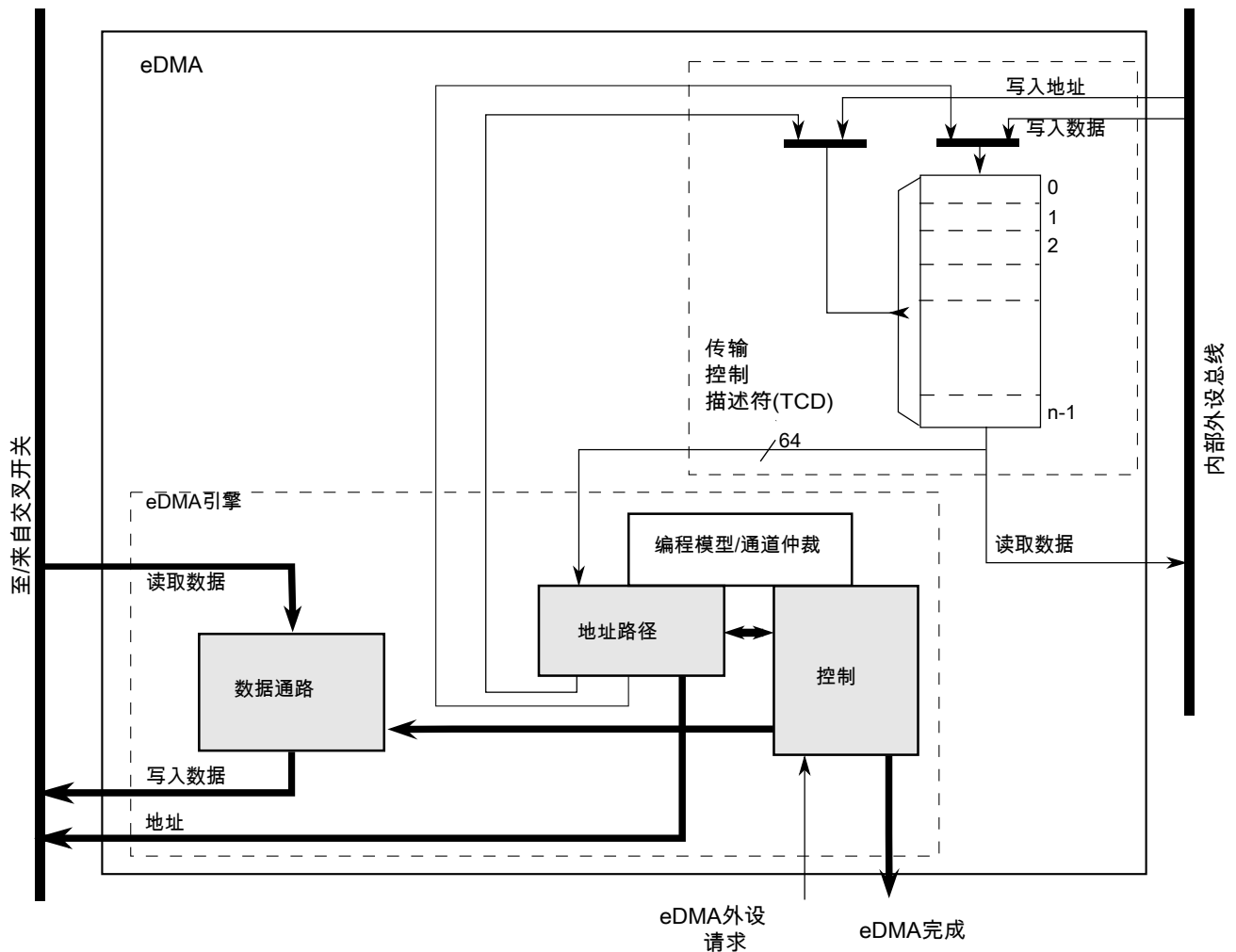


图 23-291. eDMA 操作，第 2 部分

与数据传输（地址通路、数据通路和控制）序列相关的模块通过所需的源读取和目标写入来执行实际的数据移动。读取源流程开始且获取的数据被暂时存储在数据通路块中，直到在写入目标期间进入内部总线。源读取/目标写入流程一直持续，直到传输了次字节数目。

移动完次循环字节数目之后，开始执行基本数据流的最后阶段。在本部分中，地址通路逻辑在适当的 TCD（例如 SADDR、DADDR、CITER）中对特定字段执行所需的更新。如果主迭代数目已用尽，则需执行其他操作。这些操作包括：最终的地址调整、再次将 BITER 字段加载到 CITER。此时，也会发生可选中中断请求有效，会利用描述符中的分散/聚集地址指针从存储器中获取新 TCD（如果分散/聚集已使能）。下图显示了对 TCD 存储器的更新及一个有效中断请求。

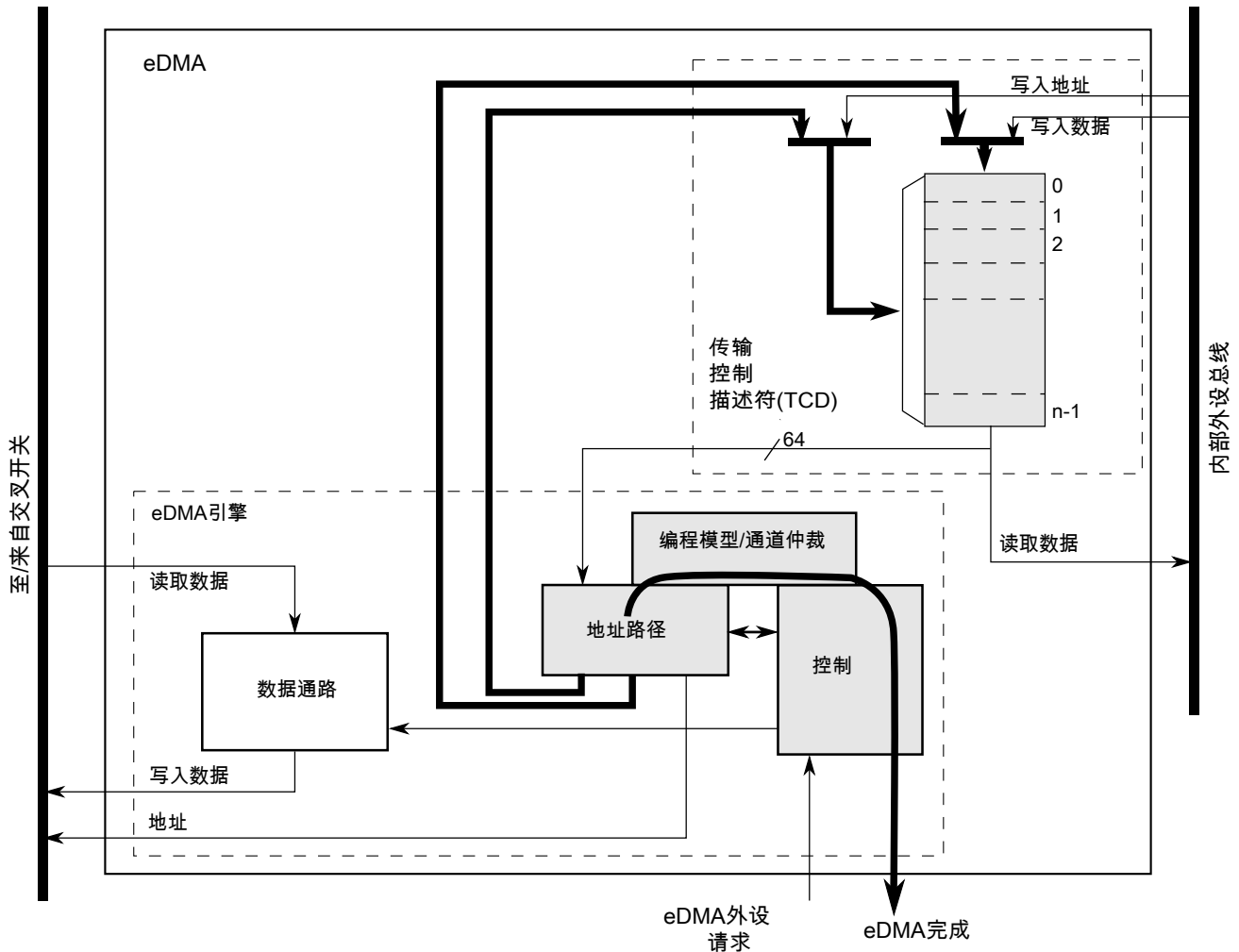


图 23-292. eDMA 操作，第 3 部分

23.4.2 故障报告和处理

通道错误在错误状态寄存器 (DMAx_ES) 中报告，导致这些错误的原因如下：

- 配置错误，它在传输控制描述符中是非法设置，或者在固定仲裁模式中是非法优先级设置。
- 总线主机读取或写入周期出现错误终止

如果起始源或目标地址、源或目标偏移量、次循环字节计数或传输大小存在不一致的状态，则会报告配置错误。下面详细介绍各种可能的原因：

- 地址和偏移量必须在传输大小边界上对齐。
- 次循环字节计数必须是源和目标传输大小的倍数。
- 所有源读取和目标写入地址必须和已编程的传输大小边界对齐。
- 在固定仲裁模式下，配置错误由任意两个同等通道优先级导致。启用固定仲裁模式时，所有通道优先级水平必须是唯一的。

注

如果两个通道具有相同的优先级，则存在通道优先级错误，并且会在错误状态寄存器中报告该错误。但是，不会在错误状态寄存器中报告通道编号。如果组内的所有通道优先级并非都是唯一的，将无法确定由仲裁选择的通道编号。

为协助执行通道优先级错误 (CPE) 调试，请在 DMA 的控制寄存器中置位停止错误位。如果组内的所有通道优先级并非都是唯一的，则在记录 CPE 错误后将停止 DMA。DMA 将保持停止状态，并且不会处理任何通道服务请求。一旦将所有通道优先级置位为唯一编号，可通过清除停止位来重新使能 DMA。

- 如果在通道完成后已使能分散/集中操作，则如果分散/集中地址 (DLAST_SGA) 未在 32 字节边界上对齐，系统会报告配置错误。
- 如果在通道完成后已使能次循环通道链接，则在尝试链接时会报告配置错误 (如果 TCDn_CITER[E_LINK]位与 TCDn_BITER[E_LINK]位不相等)。

如果使能，所有配置错误将会在通道激活时被报告，并且产生一个错误中断请求 (分散/集中和次循环链接错误除外)。如果分散/集中操作在主循环完成时开始 (若已正确使能)，则会报告分散/集中配置错误。如果链接操作在次循环完成时执行，则会报告次循环通道链接配置错误。

如果系统总线读取或写入因错误终止，则数据传输将停止，并且会置位相应的总线错误标志位。在这种情况下，DMA 引擎将使用当前源地址、目标地址和故障点的当前迭代计数来更新通道传输控制描述符的状态。如果出现系统总线错误，则该通道会在下一次传输后终止。由于管道效应，当 eDMA 收到总线错误时下一次传输已在进行中。如果在开始写入序列之前，上一次读取出现总线错误，则会使用在总线错误期间捕获的数据来执行写入。如果在切换到下一次读取序列之前，上一次写入出现总线错误，则因目标总线错误，在通道终止之前会执行读取序列。

软件可能会使用 CR[CX] 位来取消传输。当识别到取消传输请求时，DMA 引擎将停止处理通道。系统允许完成当前读取-写入序列。如果在主循环或次循环的上一个读取-写入序列中出现取消请求，则系统会放弃该取消请求，通道正常停用。

错误取消传输与取消传输相同，不同之处在于错误状态寄存器 (DMAx_ES) 将使用取消的通道编号进行更新并且会置位 ECX。已取消通道的 TCD 中包含 TCD 所保存的上一次传输的源和目标地址。如果需要重新启动该通道，用户必须重新初始化 TCD，因为上述字段不再显示原始参数。如果错误取消传输机制取消了传输，则通道编号将加载到 DMA_ES[ERRCHN]，并且会置位 ECX 和 VLD。此外，如果已使能，可能会生成错误中断。

注

如果不再需要传输完整数据，用户可通过取消传输请求停止较大数据的传输。取消传输位不会强制关闭通道。它只会停止传输数据并通过正常关闭序列停用通道。应用软件必须处理取消的上下文。如果需要（或不需要）中断，则应在取消请求之前使能（或禁用）中断。由于并未进行完整传输，应用软件必须清除传输控制描述符。

出现的任何错误都会导致 eDMA 引擎立即停止对活动通道的正常处理（它将进入错误处理状态，并且系统总线事务仍具有管道效应），并且 eDMA 错误寄存器中的相应通道位将被置位。同时，错误情况的详细信息将加载到错误状态寄存器 (DMAx_ES)。当检测到错误时，用于置位传输控制描述符 DONE 标志位并且可能产生中断请求的主循环完成指示器将不受影响。在更新错误状态后，eDMA 引擎将通过服务下一个相应通道来继续工作。系统不会自动禁用遇到错误状况的通道。如果某个通道因错误终止并在修复该错误之前发出另一个服务请求，则该通道将执行并因同一错误状况终止。

23.4.3 通道抢占

可通过置位 DCHPRIn[ECP]位来为每个通道使能通道抢占。通道抢占允许临时挂起正在执行的通道数据传输，以便启动高优先级通道。当抢占通道完成所有次循环数据传输后，被抢占的通道将恢复，并且会恢复执行。当恢复的通道完成一次读取/写入序列后，将可能重新被抢占。如果任何高优先级通道正在请求服务，则恢复的通道将挂起，并为高优先级通道服务。不支持嵌套抢占，即尝试对抢占通道进行抢占。当抢占通道开始执行后，不能对其进行抢占。只有针对选择固定仲裁时，抢占功能才可用。

可通过置位 DCHPRIn[DPA]来禁用某个通道抢占其他通道的功能。如果已禁用某个通道的抢占功能，则该通道不能挂起低优先级通道的数据传输，无论该低优先级通道的 ECP 设置如何。这样可以为低优先级、大量数据移动通道定义一个池。可将这些低优先级通道配置为互不抢占，以防止低优先级通道占用真正高优先级通道可用的抢占机会。

23.4.4 性能

本章节讲述 eDMA 模块的性能，重点介绍两种指标：

- 在传统的数据传输环境下，最能体现性能的是使用 eDMA 时达到的数据传送速率峰值。在大多数实际操作中，这种传送速率受到源速度和目标地址空间大小的限制。
- 在第二种环境中，按设备节奏将单个数据值输入外设/从外设输出是主要指标，而在固定时间内可服务的请求则是更贴切的指标。在此环境中，源速度及目标地址空间仍然非常重要。但是，eDMA 的微体系结构对相应的指标也有重要影响。

23.4.4.1 传输速率峰值

下表显示了几个不同源和目标传输的传输速率峰值。这些表假定：

- 从系统总线数据阶段来看，可在零等待状态下访问内部 SRAM
- 从系统总线数据阶段来看时，所有的内部外设总线读取需要两种等待状态，写入需要三种等待状态
- 所有内部外设总线访问的大小都为 32 位

注

所有架构并不符合上述假设。有关更多信息，请参见 SRAM 配置章节。

此表基于各种可行的系统速度，对传输速率峰值进行比较。特定芯片/器件可能并不支持列出的所有系统速度。

表 23-293. eDMA 传输速率峰值 (Mbytes/sec)

系统速度, 宽度	内部 SRAM-至- 内部 SRAM	32 位内部外设总线至内部 SRAM	内部 SRAM-至-32 位 内部外设总线
66.7 MHz, 32 位	133.3	66.7	53.3
83.3 MHz, 32 位	166.7	83.3	66.7
100.0 MHz, 32 位	200.0	100.0	80.0
133.3 MHz, 32 位	266.7	133.3	106.7
150.0 MHz, 32 位	300.0	150.0	120.0

在内核的数据通路宽度发生内部-SRAM-至-内部 1-SRAM 传输。所有涉及内部外设总线的传输均采用 32 位传输。在所有情况下，传输速率包含源读取时间+写入目标时间。

23.4.4.2 请求速率峰值

第二个性能指标是在给定时间内可以服务的 DMA 请求数。对于此指标，假定外设请求导致通道向/从内部 SRAM 移动一个内部外设总线映射操作数。在之前示例中使用的相同时间假设也适用于此计算。此外，此指标尤其反映了激活通道所需的时间。

eDMA 设计支持下列硬件服务请求序列。注意，从周期 7 开始的确切时间是通道读写访问功能的响应时间。内部外设总线读取、内部 SRAM 写入的请求流程中，数据阶段需要 4 个周期。对于 SRAM 读取和内部外设总线写入的情形，则为 5 个周期。

表 23-294. 硬件服务请求流程

周期		说明
内部外设总线读取、内部 SRAM 写入	SRAM 读取、内部外设总线写入	
	1	eDMA 外设请求有效。
	2	eDMA 外设请求在 eDMA 模块中进行本地注册并认证。TCDn_CSR[START] 位在此刻开始发起请求。同时用户写入 TCDn 字组 7 注册。
	3	通道仲裁开始
	4	通道仲裁完成开始读取传输控制描述符局部存储器。
	56	激活通道 TCD 的前两部分从本地存储器读取。eDMA 引擎的存储器宽度为 64 位，因此可在四个周期内访问整个描述符
	7	从本地存储器读取通道 TCD 的第三部分时，第一个系统总线读取周期开始。依据交叉开关的状态，在系统总线上的仲裁可在此额外插入一个延时周期。
[8:11]	8–12	正在读取 TCD 的最后一部分。此周期代表读取的第一个数据阶段及目标写入的地址阶段。
12	13	此周期代表最后一个目标写入的数据阶段。
13	14	eDMA 引擎已完成内部次循环执行操作，并准备将所需的 TCDn 字段回写到本地存储器。已读取和检查 TCDn 字组 7，以确认通道连接或分散/聚集请求。
14	15	TCDn 第一部分的相应字段已回写到本地存储器。
15	16	TCDn 第二部分的字段已回写到本地存储器。此周期进行时，下个通道仲裁周期正好开始。
16	17	待激活的下一个通道从本地存储器读取其 TCD 的第一部分。这相当于第一个通道服务请求的周期 4。

假设系统总线为零等待状态，则每隔 9 个周期即可处理一次 DMA 请求。假设与内部外设总线-至-SRAM 相关联的访问时间平均为 4 个周期，SRAM-至-内部外设总线为 5 个周期，则每隔 11.5 个周期 ($4 + (4+5)/2 + 3$)即可处理一次 DMA 请求。这是从周期 4 至周期 x+5 的时间。与系统频率呈函数关系的相应请求速率峰值如下表所示。

表 23-295. eDMA 请求速率峰值(MReq/sec)

系统频率(MHz)	请求速率 处于零等待状态	请求速率 处于等待状态
66.6	7.4	5.8
83.3	9.2	7.2
100.0	11.1	8.7
133.3	14.8	11.6
150.0	16.6	13.0

利用重叠请求计算请求速率峰值的常用公式:

$$\text{PEAKreq} = \text{freq} / [\text{entry} + (1 + \text{read_ws}) + (1 + \text{write_ws}) + \text{exit}]$$

其中:

表 23-296. 峰值请求公式操作数

操作数	说明
PEAKreq	请求速率峰值
freq	系统频率
entry	通道启动 (4 个周期)
read_ws	在系统总线读取数据阶段, 等待状态可见
write_ws	在系统总线写入数据阶段, 等待状态可见
exit	通道关闭 (3 个周期)

23.4.4.3 eDMA 性能示例

考虑具备以下特征的系统:

- 从系统总线数据阶段来看, 访问内部 SRAM 需要一个等待状态
- 从系统总线数据阶段查看时, 所有的内部外设总线读取需要两种等待状态, 写入需要三种等待状态
- 系统运行在 150 MHz 下

对于 SRAM 至内部外设总线的传输,

$$\text{PEAKreq} = 150 \text{ MHz} / [4 + (1 + 1) + (1 + 3) + 3] \text{ cycles} = 11.5 \text{ Mreq/sec}$$

对于内部外设总线至 SRAM 的传输,

$$\text{PEAKreq} = 150 \text{ MHz} / [4 + (1 + 2) + (1 + 1) + 3] \text{ cycles} = 12.5 \text{ Mreq/sec}$$

假设两种传输类型均匀分布，则请求速率平均峰值为：

$$\text{PEAKreq} = (11.5 \text{ Mreq/sec} + 12.5 \text{ Mreq/sec}) / 2 = 12.0 \text{ Mreq/sec}$$

在无任何通道处于执行且 eDMA 闲置状态下，冷启动到系统在总线上执行单一读/写（零等待状态）所需的最少周期数为：

- 软件为 11 个周期，也就是说，一个 TCD_n_CSR[START] 位，请求
- 硬件为 12 个周期，也就是说，一个 eDMA 外设请求信号，请求

仲裁流程需两个周期，另外因 eDMA 外设请求信号内部注册造成的一个硬件请求也需要一个周期。对于上述的请求速率峰值计算，仲裁和请求注册会计算入之前的正在执行的通道时间，他们是重叠的。

注

当通道连接或分散/聚集使能时，将会强制对下一个通道选择和启动执行两个周期延时。这样，连接通道或分散/聚集通道会包含在仲裁池中，可参与下次通道选择。

23.5 初始化/应用信息

以下章节探讨 eDMA 的初始化和编程的考虑因素。

23.5.1 eDMA 初始化

要初始化 eDMA：

1. 如果需要除默认设置以外的配置，请写入 CR。
2. 如果需要除默认设置以外的配置，请将通道优先级写入 DCHPRI_n 寄存器。
3. 如果需要，请在 EEI 寄存器中使能错误中断。
4. 为每个可能请求服务的通道写入 32 字节 TCD。
5. 通过 ERQ 寄存器使能任何硬件服务请求。
6. 通过以下方式请求通道服务：
 - 软件：置位 TCD_n_CSR[START]
 - 硬件：从设备使其 eDMA 外设请求信号有效

在任何通道请求服务后，将根据写入编程模型的仲裁和优先级来选择要执行的通道。eDMA 引擎会将所选通道的完整 TCD（包括 TCD 控制和状态字段，如下表所示）读取到其内部地址路径模块。

当 TCD 读取后，内部 总线上将启动第一次传输，除非检测到配置错误。从源（由 TCDn_SADDR 定义）到目标（由 TCDn_DADDR 定义）的传输将继续，直到传输完由 TCDn_NBYTES 指定的字节数。

当传输完成后，eDMA 引擎的本地 TCDn_SADDR、TCDn_DADDR 和 TCDn_CITER 将回写到主 TCD 存储器，并且会执行任何次循环通道链接（如果已使能）。如果主循环耗尽，将进一步执行后处理，例如中断、主循环通道链接以及分散/集中操作（如果已使能）。

表 23-297. TCD 控制和状态字段

TCDn_CSR 字段名称	说明
START	在使用软件启动 DMA 服务时用于明确启动通道的控制位（由硬件自动清除）
ACTIVE	用于指示通道当前正在执行的状态位
DONE	用于指示主循环完成的状态位（当使用软件启动 DMA 服务时由软件清除）
D_REQ	在使用硬件启动 DMA 服务时用于在主循环完成后禁用 DMA 请求的控制位
BWC	用于调节通道带宽控制的控制位
E_SG	用于使能分散-集中功能的控制位
INT_HALF	用于在主循环完成一半时使能中断的控制位
INT_MAJ	用于在主循环完成时使能中断的控制位

下图显示每个 DMA 请求如何在无 CPU 干预的情况下启动一个次循环传输或迭代。在每个次循环后会出现 DMA 仲裁，并且允许一级次循环 DMA 抢占。主循环中的次循环数由开始迭代计数(BITER)指定。

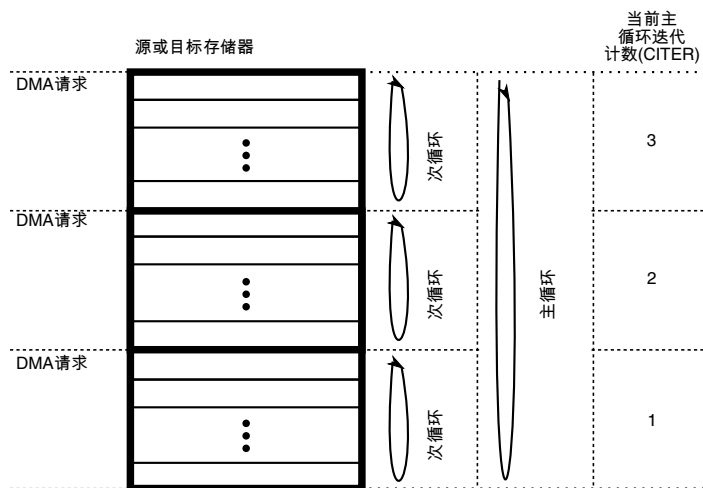


图 23-293. 多循环迭代示例

下图列出了存储器阵列术语以及 TCD 设置的关联方式。

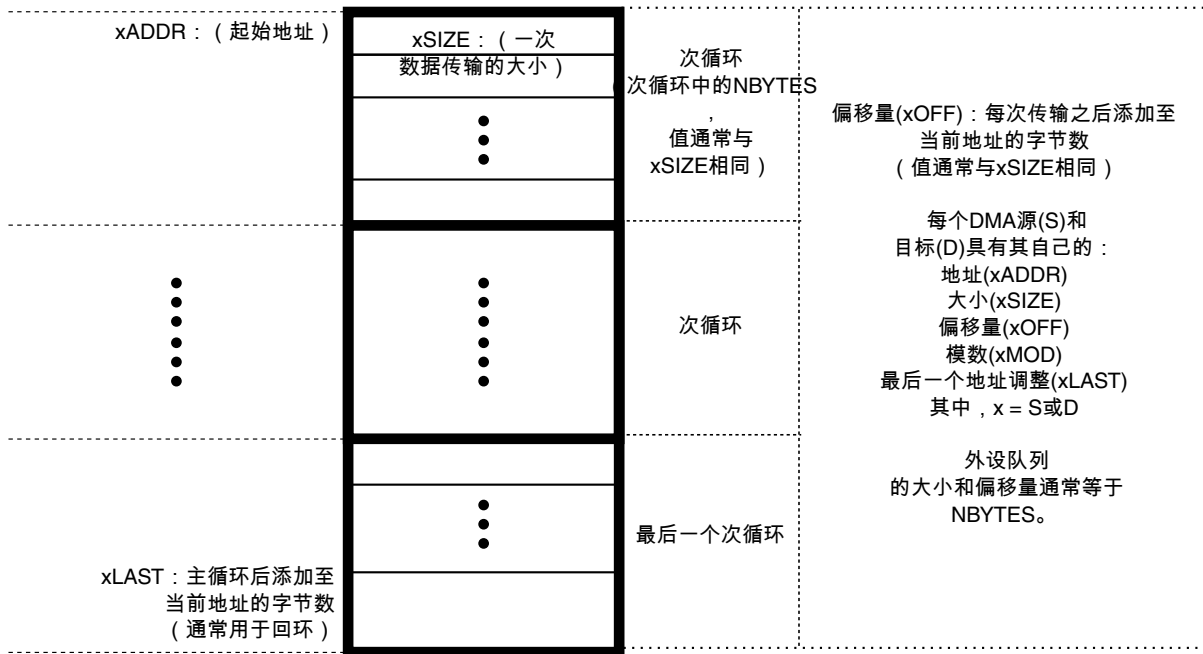


图 23-294. 存储器阵列术语

23.5.2 编程错误

eDMA 负责对传输控制描述符执行各种测试以验证描述符数据的一致性。大多数编程错误都按通道进行报告，除两种通道优先级错误 (ES[CPE]) 外。

对于除通道优先级错误以外的所有错误类型，将在错误状态寄存器 (DMAx_ES) 中记录导致错误的通道编号。如果在下一次激活问题通道之前未清除错误源，则系统会再次检测到该错误并记录下来。

如果优先级并非唯一，当任何通道请求服务时，系统都会报告通道优先级错误。已选择具有活动请求的最高通道优先级，但具有该优先级的最低编号通道通过仲裁选择并由 eDMA 引擎执行。硬件服务请求握手信号、错误中断和错误报告与选定的通道相关。

23.5.3 仲裁模式注意事项

本节介绍 eDMA 的仲裁模式注意事项。

23.5.3.1 固定通道仲裁

在此模式下，选定执行最高优先级的服务请求。

23.5.3.2 循环通道仲裁

C 通道，而不会考虑组内分配的。

23.5.4 执行 DMA 传输

本章节给出示例，演示如何利用 eDMA 执行 DMA 传输。

23.5.4.1 单个请求

要通过一次激活操作传输 n 字节数据，则需将主循环置位为一 ($TCDn_CITER = TCDn_BITER = 1$)。确认通道服务请求并选择通道开始执行之后，开始数据传输。传输完成后， $TCDn_CSR[DONE]$ 位置位，且如果中断正常使能则生成一个中断。

例如，配置以下 TCD 条目可传输 16 字节数据。对 eDMA 编程实现主循环（每个迭代传输 16 字节）的一个迭代。源存储器有一个字节位宽的存储器端口，位于 $0x1000$ 。此目标存储器具有一个 32 位端口，位于 $0x2000$ 。对地址偏移量进行增量编程，以匹配传输大小：源的增量为 1 个字节，目标的增量为 4 个字节。调节最终的源和目标地址，以还原为初始值。

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA = -16
TCDn_CSR[INT_MAJ] = 1
TCDn_CSR[START] = 1 (Should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

此操作会生成以下事件序列：

1. 用户写入 $TCDn_CSR[START]$ 位请求通道服务。
2. 通过仲裁选择通道进行服务。
3. eDMA 引擎写入： $TCDn_CSR[DONE] = 0$ 、 $TCDn_CSR[START] = 0$ 、 $TCDn_CSR[ACTIVE] = 1$ 。
4. eDMA 引擎读取：通道 TCD 数据，从本地存储器到内部寄存器文件。
5. 按照以下方式执行源-目标传输：

- a. 从 0x1000 位置读取字节，从 0x1001 位置读取字节，从 0x1002 读取字节，从 0x1003 读取字节。
 - b. 向 0x2000 位置写入 32 位→次循环的首个迭代。
 - c. 从 0x1004 位置读取字节，从 0x1005 位置读取字节，从 0x1006 读取字节，从 0x1007 读取字节。
 - d. 向 0x2004 位置写入 32 位→次循环的第 2 个迭代。
 - e. 从 0x1008 位置读取字节，从 0x1009 位置读取字节，从 0x100A 读取字节，从 0x100B 读取字节。
 - f. 向 0x2008 位置写入 32 位→次循环的第 3 个迭代。
 - g. 从 0x100C 位置读取字节，从 0x100D 位置读取字节，从 0x100E 读取字节，从 0x100F 读取字节。
 - h. 向 0x200C 位置写入 32 位→次循环的最后一个迭代→主循环完成。
6. eDMA 引擎写入：TCDn_SADDR = 0x1000、TCDn_DADDR = 0x2000、TCDn_CITER = 1 (TCDn_BITER)。
 7. eDMA 引擎写入：TCDn_CSR[ACTIVE] = 0、TCDn_CSR[DONE] = 1、INT[n] = 1。
 8. 通道停用，eDMA 进入空闲状态或服务下一个通道。

23.5.4.2 多次请求

以下示例通过两种硬件请求传输 32 字节，其他与之前的示例相同。唯一的字段变化是主循环迭代数目和最终地址的偏移量。对 eDMA 编程实现主循环（每个迭代传输 16 字节）的两个迭代。当通道的硬件请求在 ERQ 寄存器中使能之后，从设备发起通道服务请求。

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

此操作会生成以下事件序列：

1. 第一个硬件，即 eDMA 外设，请求通道服务。
2. 通过仲裁选择通道进行服务。
3. eDMA 引擎写入：TCDn_CSR[DONE] = 0、TCDn_CSR[START] = 0、TCDn_CSR[ACTIVE] = 1。

4. eDMA 引擎读取：通道 TCD n 数据，从本地存储器到内部寄存器文件。
5. 按照以下方式执行源-目标传输：
 - a. 从 0x1000 位置读取字节，从 0x1001 位置读取字节，从 0x1002 读取字节，从 0x1003 读取字节。
 - b. 向 0x2000 位置写入 32 位→次循环的首个迭代。
 - c. 从 0x1004 位置读取字节，从 0x1005 位置读取字节，从 0x1006 读取字节，从 0x1007 读取字节。
 - d. 向 0x2004 位置写入 32 位→次循环的第 2 个迭代。
 - e. 从 0x1008 位置读取字节，从 0x1009 位置读取字节，从 0x100A 读取字节，从 0x100B 读取字节。
 - f. 向 0x2008 位置写入 32 位→次循环的第 3 个迭代。
 - g. 从 0x100C 位置读取字节，从 0x100D 位置读取字节，从 0x100E 读取字节，从 0x100F 读取字节。
 - h. 向 0x200C 位置写入 32 位→次循环的最后一个迭代。
6. eDMA 引擎写入：TCD n _SADDR = 0x1010、TCD n _DADDR = 0x2010、TCD n _CITER = 1。
7. eDMA 引擎写入：TCD n _CSR[ACTIVE] = 0。
8. 通道停用 → 主循环的一个迭代。eDMA 进入空闲状态或服务下一个通道。
9. 第二个硬件，即 eDMA 外设，请求通道服务。
10. 通过仲裁选择通道进行服务。
11. eDMA 引擎写入：TCD n _CSR[DONE] = 0、TCD n _CSR[START] = 0、TCD n _CSR[ACTIVE] = 1。
12. eDMA 引擎读取：通道 TCD 数据，从本地存储器到内部寄存器文件。
13. 按照以下方式执行源-目标传输：
 - a. 从 0x1010 位置读取字节，从 0x1011 位置读取字节，从 0x1012 读取字节，从 0x1013 读取字节。
 - b. 向 0x2010 位置写入 32 位→次循环的首个迭代。
 - c. 从 0x1014 位置读取字节，从 0x1015 位置读取字节，从 0x1016 读取字节，从 0x1017 读取字节。
 - d. 向 0x2014 位置写入 32 位→次循环的第 2 个迭代。

- e. 从 0x1018 位置读取字节，从 0x1019 位置读取字节，从 0x101A 读取字节，从 0x101B 读取字节。
 - f. 向 0x2018 位置写入 32 位→次循环的第 3 个迭代。
 - g. 从 0x101C 位置读取字节，从 0x101D 位置读取字节，从 0x101E 读取字节，从 0x101F 读取字节。
 - h. 向 0x201C 位置写入 32 位→次循环的最后一个迭代→主循环完成。
14. eDMA 引擎写入：TCD_n_SADDR = 0x1000、TCD_n_DADDR = 0x2000、TCD_n_CITER = 2 (TCD_n_BITER)。
15. eDMA 引擎写入：TCD_n_CSR[ACTIVE] = 0、TCD_n_CSR[DONE] = 1、INT[n] = 1。
16. 通道停用 → 主循环完成。eDMA 进入空闲状态或服务下一个通道。

23.5.4.3 使用取模特性

利用 eDMA 的模数特性可以实施一个环形数据队列，其中队列的大小为 2 的幂数。MOD 是 TCD 中的一个 5 位源和目标字段，它指出在进行地址+偏移计算之后，哪个低位地址位相比原值递增。所有高位地址位保持原值不变。如果此字段置位为 0，则禁用模数特性。

下表显示根据 MOD 字段设置指定传输地址的方式。当地址覆盖原数值，而 28 个高位地址位(0x1234567x)保持原值不变时，即产生循环缓冲区。在本示例中，源地址设置为 0x12345670，偏移量设置为 4 个字节且 MOD 字段设置为 4，因此可出现 2⁴ 字节 (16 字节) 大小的队列。

表 23-298. 模数示例

传输编号	地址
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C
5	0x12345670
6	0x12345674

23.5.5 监控传输描述符状态

本节讨论如何监控 eDMA 状态。

23.5.5.1 测试次循环是否完成

使用软件启动服务请求时，可通过两种方法测试次循环是否完成。第一种是读取 `TCDn_CITER` 字段并测试是否更改。另一种方法可参考以下流程。第二种方法是测试 `TCDn_CSR[START]`位和 `TCDn_CSR[ACTIVE]`位。次循环完成情况可通过在 `TCDn_CSR[START]`置位之后两个位均为 0 表示。轮询 `TCDn_CSR[ACTIVE]`位可能无结果，因为当通道执行的持续时间较短时可能会错过 `ACTIVE` 状态。

TCD 状态位为软件激活的通道执行以下序列：

阶段	TCDn_CSR 位			状态
	开始	激活	完成	
1	1	0	0	通过软件发出的通道服务请求
2	0	1	0	通道正在执行
3a	0	0	0	通道已完成次循环并处于空闲状态
3b	0	0	1	通道已完成主循环并处于空闲状态

使用硬件（外设发起的服务请求）时，测试是否完成次循环的最佳方法是读取 `TCDn_CITER` 字段并测试是否存在更改。硬件请求和确认握手信号在编程模型中不可见。

TCD 状态位针对硬件激活的通道执行以下序列：

阶段	TCDn_CSR 位			状态
	开始	激活	完成	
1	0	0	0	通过硬件发出的通道服务请求（外设请求已断言）
2	0	1	0	通道正在执行
3a	0	0	0	通道已完成次循环并处于空闲状态
3b	0	0	1	通道已完成主循环并处于空闲状态

对于两种激活类型，主循环完成状态通过 `TCDn_CSR[DONE]`位显式表示。

当通道开始执行时，无论通道是否激活，`TCDn_CSR[START]`位均会自动清零。

23.5.5.2 读取活动通道的传输描述符

如果 eDMA 在通道执行时读取，则可回读 TCD_n_SADDR、TCD_n_DADDR 及 TCD_n_NBYTES 的真实值。SADDR、DADDR 及 NBYTES 的真实值指的是 eDMA 引擎当前在其内部寄存器文件中使用的值，而不是该通道在 TCD 本地存储器中使用的值。地址、SADDR 和 DADDR 及 NBYTES 随传输进程降低到 0，可指示传输的进程。所有其他数值均从 TCD 本地存储器读回。

23.5.5.3 检查通道抢占状态

只有针对选择固定仲裁作为通道仲裁模式时，抢占功能才可用。可抢占状况：在该种情况下，抢占使能通道运行且较高优先级请求激活。当 eDMA 引擎未在固定通道仲裁模式下运行时，则不能明确决定有效运行的相应优先级是否完成请求。通道优先级平等对待，也就是说，当选择循环仲裁模式时，会不断循环。

在整个抢占过程中，被抢占通道的 TCD_n_CSR[ACTIVE] 位保持有效。被抢占通道暂时处于挂起状态，而抢占通道则执行一次主循环迭代。如果在全局 TCD 映射中，有两个 TCD_n_CSR[ACTIVE] 位同时置位，则优先级较高的通道会主动抢占优先级较低的通道。

23.5.6 通道链接

通道链接（或链）是一种机制，其中一个通道置位另一个通道（或自身）的 TCD_n_CSR[START]位，从而发起该通道的服务请求。如果已正确使能，EDMA 引擎将自动在主循环或次循环完成时执行此操作。

次循环通道链接在次循环（或主循环的一个迭代）完成时出现。

TCD_n_CITER[E_LINK]字段确定是否请求次循环链接。如果已使能，在主循环的每次迭代（最后一次迭代除外）后建立通道链接。当主循环耗尽时，系统仅使用主循环通道链接字段确定是否应建立通道链接。例如，初始字段：

```
TCDn_CITER[E_LINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJOR_E_LINK] = 1
TCDn_CSR[MAJOR_LINKCH] = 0x7
```

执行为：

1. 次循环完成 → 置位 TCD12_CSR[START]位
2. 次循环完成 → 置位 TCD12_CSR[START]位
3. 次循环完成 → 置位 TCD12_CSR[START]位

4. 次循环完成、主循环完成 → 置位 TCD7_CSR[START]位

如果次循环链接已使能($TCDn_CITER[E_LINK] = 1$), 则 $TCDn_CITER[CITER]$ 字段使用一个 9 位向量来构成当前迭代计数。如果次循环链接已禁用 ($TCDn_CITER[E_LINK] = 0$), 则 $TCDn_CITER[CITER]$ 字段使用一个 15 位向量来构成当前迭代计数。与 $TCDn_CITER[LINKCH]$ 字段相关联的位将级联至 CITER 值以增加 CITER 的范围。

注

$TCDn_CITER[E_LINK]$ 位和 $TCDn_BITER[E_LINK]$ 位必须相等, 否则系统会报告配置错误。CITER 和 BITER 向量宽度必须相等, 这样才能计算主循环在完成过程中的中断点。

下表概述了在循环结束时 DMA 通道如何链接至另一个 DMA 通道, 即用另一个通道的 TCD。

表 23-299. 通道链接参数

所需链接行为	TCD 控制字段名称	说明
在次循环结束时链接	E_LINKCITER	在次循环 (当前迭代) 完成时使能通道至通道链接
	CITER[LINKCH]	在次循环 (当前迭代) 完成时链接的通道编号
在主循环结束时链接	CSR[MAJOR_E_LINK]	在主循环完成时使能通道至通道链接
	CSR[MAJOR_LINKCH]	在主循环完成时链接的通道编号

23.5.7 动态编程

本章节提供建议在执行通道期间用于更改编程模型的方法。

23.5.7.1 动态更改通道优先级

推荐采用以下两种选项来动态地更改通道的优先级:

1. 切换至循环通道仲裁模式, 更改通道的优先级, 然后切回到固定仲裁模式,
2. 禁用的所有通道, 仅更改通道优先级, 然后使能相应的通道。

23.5.7.2 动态通道链接

动态通道链接指的是在通道执行期间设置 TCD.major.e_link 位的过程 (参见 [TCD 结构](#) 中的图)。此位在通道执行结束的时候从 TCD 本地存储器中读取, 因此用户可以在通道执行过程中使能此功能。

因为用户可以在执行过程中更改此配置, 所以需要一致性模型。试想这样的场景: 用户试图在 eDMA 引擎退出通道时通过使能 TCD.major.e_link 位来执行动态通道链接。TCD.major.e_link 会在编程模式下置位, 但并不明确是否在通道停用前实施实际链接。

在执行动态通道链接请求时, 建议采用下列一致性模型。

1. 向 TCD.major.e_link 位写入 1。
2. 回读 TCD.major.e_link 位。
3. 测试 TCD.major.e_link 请求状态:
 - 如果 TCD.主.e_link = 1, 则动态链接尝试成功。
 - 如果 TCD.major.e_link = 0, 则尝试动态链接未成功 (通道已停用)。

对于此请求, TCD 本地存储器控制器迫使 TCD.major.e_link 位在某个通道的 TCD.done 位置位之后, 在向该通道的 TCD.word7 写入时归零, 表示主循环已完成。

注

用户在写入 TCD.major.e_link 位之前必须先清除 TCD.done 位。在通道开始执行之后, eMDA 引擎会自动清除 TCD.done 位。

23.5.7.3 动态分散/聚集

分散/聚集指的是自动将新的 TCD 加载至通道的过程。它使得 DMA 通道可使用多个 TCD; 这使得 DMA 通道可将 DMA 数据分散至多个目标, 或从多个源聚集数据。当分散/聚集已使能且通道已完成其主循环, eDMA 从系统存储器中取回一个新的 TCD, 并在 eDMA 编程模式下加载至通道的描述符位置中, 从而替换现有的描述符。

因为用户可以在执行过程中更改此配置, 所以需要一致性模型。试想这样的场景: 用户试图在 eDMA 引擎退出通道时通过使能 TCD.e_sg 位来执行动态分散/聚集操作。TCD.e_sg 会在编程模式下置位, 但并不明确是否在通道停用前兑现实际的分散/聚集请求。

以下子章节中显示了此一致性模型的两种方法。方法 1 的优势在于可一次读取 major.linkch 字段和 e_sg 位。对于动态通道链接和分散/聚集请求, TCD 本地存储器控制器迫使 TCD.major.e_link 位和 TCD.e_sg 位在某个通道的 TCD.done 位置位 (表示主循环已完成之后), 在向该通道的 TCD.word7 写入时归零。

注

用户在写入 TCD.major.e_link 和 TCD.e_sg 位之前必须先清除 TCD.done 位。在通道开始执行之后，eMDA 引擎会自动清除 TCD.done 位。

23.5.7.3.1 方法 1（不使用主循环通道链接的通道）

对于不使用主循环通道链接的通道，此处所述的一致性模型可用于动态分散/集中请求。

当 TCD.major.e_link 位为 0 时，eDMA 不会使用 TCD.major.linkch 字段。在这种情况下，TCD.major.linkch 位可用作其他用途。此方法将 TCD.major.linkch 字段用作 TCD 标识(ID)。

1. 构建描述符之后，利用动态分散/集中，针对每个与通道关联的 TCD，在 TCD.major.linkch 字段中写入一个唯一的 TCD ID。
2. 将 1b 写入 TCD.d_req 位。

如果动态分散/集中尝试失败，则置位 TCD.d_req 位未来将无法激活此通道硬件。这将使通道无法执行使用分散/集中地址（在下一步写入）替代最终偏移量值(dlast)计算得出的目标地址(daddr)。

3. 将分散/集中地址写入 TCD.dlast_sga 字段。
4. 将 1b 写入 TCD.e_sg 位。
5. 回读 16 位 TCD 控制/状态字段。
6. 测试 TCD.e_sg 请求状态和 TCD.major.linkch 值：

如果 e_sg = 1b，则动态链接尝试成功。

如果 e_sg = 0b 且 major.linkch (ID) 未更改，则尝试的动态链接未成功（通道已停用）。

如果 e_sg = 0b 且 major.linkch (ID)更改，则动态链接尝试成功（新的 TCDe_sg 值清除了 e_sg 位）。

23.5.7.3.2 方法 2（使用主循环通道链接的通道）

对于使用主循环通道链接的通道，此处所述的一致性模型可用于动态分散/集中请求。此方法将 TCD.dlast_sga 字段用作 TCD 标识(ID)。

1. 将 1b 写入 TCD.d_req 位。

如果动态分散/集中尝试失败，则置位 d_req 位未来将无法激活此通道硬件。这将使通道无法执行使用分散/集中地址 (在下一步写入) 替代最终偏移量值(dlast) 计算得出的目标地址(daddr)。

2. 将分散/集中地址写入 TCD.dlast_sga 字段。
3. 将 1b 写入 TCD.e_sg 位。
4. 回读 TCD.e_sg 位。
5. 测试 TCD.e_sg 请求状态:

如果 e_sg = 1b，则动态链接尝试成功。

如果 e_sg = 0b，则读取 32 位 TCD dlast_sga 字段。

如果 e_sg = 0b 且 dlast_sga 未更改，则尝试的动态链接未成功 (通道已停用)。

如果 e_sg = 0b 且 dlast_sga 更改，则动态链接尝试成功 (新的 TCDe_sg 值清除了 e_sg 位)。

第 24 章 外部看门狗监控器 (EWM)

24.1 此模块的芯片实现细节

24.1.1 EWM 时钟

此表显示了 EWM 时钟和相应的芯片时钟。

表 24-1. EWM 时钟连接

模块时钟	芯片时钟
低功耗时钟	1 kHz LPO 时钟

24.1.2 EWM 低功耗模式

此表显示 EWM 低功耗模式和对应的芯片低功耗模式。

表 24-2. EWM 低功耗模式

模块模式	芯片模式
等待	等待、VLPW
停止	停止、VLPS、LLS

24.1.3 EWM_OUT 低功耗模式下的引脚状态

当 CPU 退出等待或停止模式进入运行模式时, 引脚会恢复进入等待或停止模式之前的状态。当 CPU 上电后进入运行模式时, 引脚重新回到复位状态。

24.2 简介

为安全起见，冗余看门狗系统（外部看门狗监控器(EWM)）设计用于监控外部电路以及 MCU 软件流。这可为复位 MCU CPU 和外设的内部看门狗提供备份机制。

看门狗通常用于监控 MCU 内嵌入式软件的流程和执行。看门狗包含一个计数器，它（如果允许溢出）将强制所有片上外设进行内部复位（异步）并可选择性地使 $\overline{\text{RESET}}$ 引脚有效以复位外部器件/电路。如果软件代码工作正常并且响应看门狗重新计数，则不会发生看门狗计数器溢出。

EWM 与内部看门狗的不同之处在于它无法复位 MCU CPU 和外设。如果复位或将外部电路置于安全模式，则 EWM 可提供独立的 $\overline{\text{EWM_out}}$ 信号。EWM 计数器超时时使 $\overline{\text{EWM_out}}$ 信号有效。提供一个可选外部输入 EWM_in 实现对 $\overline{\text{EWM_out}}$ 信号的额外控制。

24.2.1 特性

EWM 模块的特性包括：

- 独立的 LPO_CLK 时钟源
- 可编程超时周期，以 EWMLPO_CLK 时钟周期数指定。
- 窗口刷新选项
 - 提供程序流程快于预期的鲁棒性检查。
 - 可编程窗口。
 - 刷新外部窗口将导致 $\overline{\text{EWM_out}}$ 有效。
- 可靠刷新机制
 - 在 15 (*EWM_refresh_time*) 个外设总线时钟周期内将值 0xB4 和 0x2C 写入 EWM 刷新寄存器。
- 一个使用于复位或将外部电路置于安全模式有效的输出端口 $\overline{\text{EWM_out}}$ 。
- 一个允许外部电路控制使 $\overline{\text{EWM_out}}$ 信号有效的输入端口 $\overline{\text{EWM_in}}$ 。

24.2.2 工作模式

本章节描述模块的工作模式。

24.2.2.1 停止模式

当 EWM 处于停止模式时，CPU 无法刷新至 EWM。进入停止模式时，EWM 的计数器冻结。

有两种退出停止模式的方法：

- 通过复位退出停止模式，此时 EWM 保持禁用。
- 通过中断退出停止模式，此时 EWM 重新使能，而计数器则从进入停止模式前的相同数值开始继续计时。

如果 EWM 在 CPU 刷新期间进入了停止模式，则需注意以下事项：如果通过中断退出停止模式，则刷新机制状态机从前一状态开始，也就是说，如果首个刷新命令正确写入且 EWM 立即进入停止模式，则下个命令必须在退出停止模式后的下个 15 (*EWM_refresh_time*) 个外设总线时钟内写入。在执行 EWM 刷新指令之前，用户必须屏蔽所有中断。

24.2.2.2 等待模式

EWM 模块以相同的方式对待停止和等待模式。在这些模式下，EWM 的功能均保持相同。

24.2.2.3 调试模式

进入调试模式不会对 EWM 产生影响。

- 如果 EWM 在进入调试模式前使能，则它保持使能。
- 如果 EWM 在进入调试模式前禁用，则它保持禁用。

24.2.3 结构框图

此图显示的是 EWM 的结构框图。

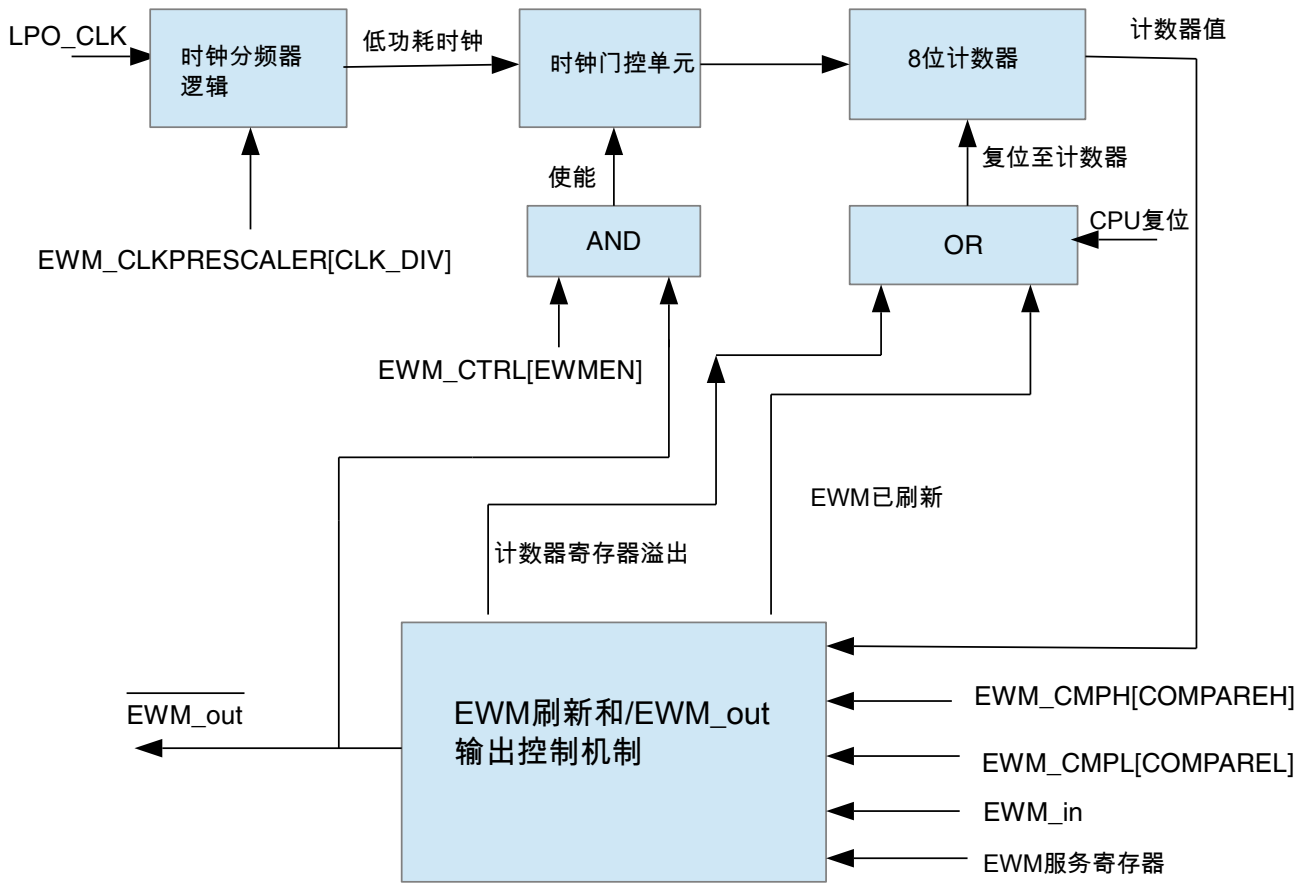


图 24-1. EWM 结构框图

24.3 EWM 信号说明

EWM 具有两个外部信号，如下表所示。

表 24-3. EWM 信号说明

信号	说明	I/O
EWM_in	用于外部安全电路安全状态的 EWM 输入。EWM_in 的极性可通过 EWM_CTRL[ASSIN]位进行编程。默认极性为低电平有效。	I
EWM_out	EWM 复位输出信号	O

24.4 存储器映射/寄存器定义

此章节介绍模块存储器映射和寄存器。

EWM 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_1000	控制寄存器 (EWM_CTRL)	8	R/W	00h	24.4.1/443
4006_1001	服务寄存器 (EWM_SERV)	8	W (始终读 0)	00h	24.4.2/444
4006_1002	比较低电平寄存器 (EWM_CMPL)	8	R/W	00h	24.4.3/444
4006_1003	比较高电平寄存器 (EWM_CMPH)	8	R/W	FFh	24.4.4/445
4006_1005	时钟预分频器寄存器 (EWM_CLKPRESCALER)	8	R/W	00h	24.4.5/445

24.4.1 控制寄存器 (EWM_CTRL)

任何复位均会清除 CTRL 寄存器。

注

在 CPU 复位之后, 可一次写入 INEN、ASSIN 和 EWMEN 位。如果对这些位的更改多于一次, 则会生成一个总线传输错误。

地址: 4006_1000h 基准 + 0h 偏移 = 4006_1000h

位	7	6	5	4	3	2	1	0
读	0				INTEN	INEN	ASSIN	EWMEN
写	0				0	0	0	0
复位	0	0	0	0	0	0	0	0

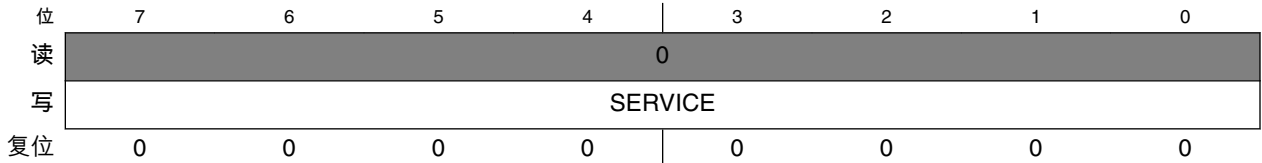
EWM_CTRL 字段描述

字段	描述
7-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 INTEN	中断使能。 此位置位且 $\overline{\text{EWM_out}}$ 被置为有效时, 会生成一个中断请求。如果要解除对此中断请求, 用户应通过写入 0 来清除此位。
2 INEN	输入使能 此位置位时, 会使能 EWM_in 端口。
1 ASSIN	EWM_in 的有效状态选择。 EWM_in 信号的默认有效状态为逻辑 0。置位 ASSIN 位会使得 EWM_in 信号的有效状态反相变为逻辑 1。
0 EWMEN	EWM 使能。 此位置位时, 会使能 EWM 模块。因此 EWM 计数器复位为 0, 并解除对 $\overline{\text{EWM_out}}$ 信号的认定。此位未置位时, 保持 EWM 模块被禁用。因为此位具备一次写入特性, 因此直到下次复位前, 都无法重新对其使能。

24.4.2 服务寄存器 (EWM_SERV)

SERV 寄存器提供从 CPU 至 EWM 模块的接口。它是只写接口，读取此寄存器会返回 0。

地址: 4006_1000h 基准 + 1h 偏移 = 4006_1001h



EWM_SERV 字段描述

字段	描述
SERVICE	EWM 刷新机制要求 CPU 向 SERV 寄存器写入两个值：首个数据字节 0xB4，后接第二个数据字节 0x2C。如果以下任意条件成立，则 EWM 刷新无效。 <ul style="list-style-type: none"> 第一或第二个数据字节未正确写入。 第二个数据字节未在第一个数据字节的固定外设总线周期数内写入。固定周期数称为 <i>EWM_refresh_time</i>。

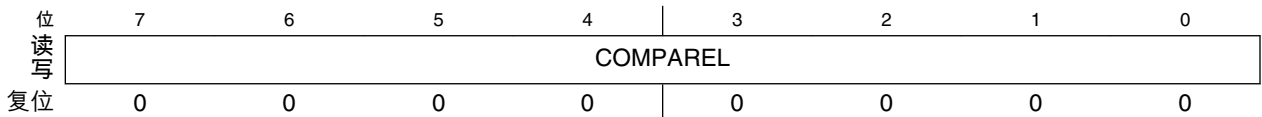
24.4.3 比较低电平寄存器 (EWM_CMPL)

CMPL 寄存器在 CPU 复位后复位为 0。这不会提供 CPU 刷新 EWM 计数器的最短时间。

注

仅可在 CPU 复位之后一次写入此寄存器。如果对此寄存器的写入多于一次，则会生成一个总线传输错误。

地址: 4006_1000h 基准 + 2h 偏移 = 4006_1002h



EWM_CMPL 字段描述

字段	描述
COMPAREL	为了防止失控代码更改此字段，软件应在 CPU 复位后对此字段进行写入，即使在需要（默认）最短的刷新时间时也是如此。

24.4.4 比较高电平寄存器 (EWM_CMPH)

CMPH 寄存器在 CPU 复位后复位为 0xFF。这提供 CPU 刷新 EWM 计数器的最长 256 个时钟时间。

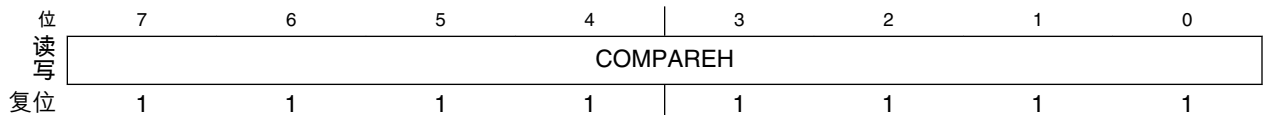
注

仅可在 CPU 复位之后一次写入此寄存器。如果对此寄存器的写入多于一次，则会生成一个总线传输错误。

注

CMPH 的有效值高达 0xFE，这是因为 EWM 计数器在 CMPH = 0xFF 时绝不会过期。只有当 EWM 计数器超过 CMPH 时，才会过期。

地址: 4006_1000h 基准 + 3h 偏移 = 4006_1003h



EWM_CMPH 字段描述

字段	描述
COMPAREH	为了防止失控代码更改此字段，软件应在 CPU 复位后对此字段写入，即使在需要（默认）最大的刷新时间时也是如此。

24.4.5 时钟预分频器寄存器 (EWM_CLKPRESCALER)

CLKPRESCALER 寄存器在 CPU 复位后复位为 0x00。

注

仅可在 CPU 复位之后一次写入此寄存器。如果对此寄存器的写入多于一次，则会生成一个总线传输错误。

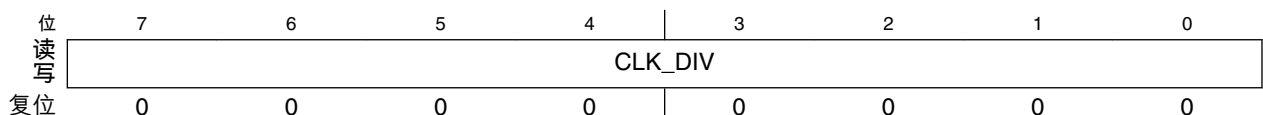
注

使能 EWM 之前，先写入所需的预分频器值。

注

此寄存器的实施取决于具体芯片。详情请参见芯片配置信息。

地址: 4006_1000h 基准 + 5h 偏移 = 4006_1005h



EWM_CLKPRESCALER 字段描述

字段	描述
CLK_DIV	选定用于运行 EWM 计数器的低功耗时钟源可按如下方式预分频。 <ul style="list-style-type: none"> 预分频时钟频率 = 低功耗时钟源频率 / (1 + CLK_DIV)

24.5 功能说明

以下各节描述 EWM 模块的详细功能。

注

当 BUS_CLK 丢失时，EWM 模块不会生成 $\overline{\text{EWM_out}}$ 信号，且不可执行刷新操作。

24.5.1 $\overline{\text{EWM_out}}$ 信号

$\overline{\text{EWM_out}}$ 是数字输出信号，用于对控制关键安全功能的外部电路（因应用而异）进行门控。例如，可以将 $\overline{\text{EWM_out}}$ 连接至控制大型家电的交流电机的高压晶体管电路。

当 EWM 由 CPU 在可编程刷新窗口中定期刷新时， $\overline{\text{EWM_out}}$ 信号将保持解除置位状态，表示正常执行应用代码。

$\overline{\text{EWM_out}}$ 信号在任意下列条件下置位：

- EWM 刷新发生于计数器值小于 CMPL 值时。
- EWM 计数器值达到 CMPH 值并且未发生 EWM 刷新。
- 如果在刷新 EWM 时，EWM_in 引脚的功能被使能且 EWM_in 引脚置位。
- 任何复位之后（通过 $\overline{\text{EWM_out}}$ 引脚上的外部下拉机制实现）

$\overline{\text{EWM_out}}$ 在通过 $\overline{\text{EWM_out}}$ 信号上的外部下拉机制实现的任何复位之后置位。然后，要解除置位 $\overline{\text{EWM_out}}$ 信号，需置位 CTRL 寄存器中的 EWMEN 位，以使能 EWM。

如果 $\overline{\text{EWM_out}}$ 信号与数字 I/O 引脚共用管脚，则在复位时，此管脚将顺势成为输入信号。只有当通过 CTRL 寄存器中的 EWMEN 位使能 EWM 之后，管脚状态才能由 $\overline{\text{EWM_out}}$ 信号控制。

注

当使用 EWM 功能时且 EWM 处于复位状态时， $\overline{\text{EWM_out}}$ 管脚必须处于下拉状态。

24.5.2 EWM_in 信号

$\overline{\text{EWM_in}}$ 是外部安全电路的安全状态数字输入信号，使得外部电路能够控制对 $\overline{\text{EWM_out}}$ 信号的认定。例如，在应用中，外部电路监测一个关键的安全功能，如果此安全功能出现故障，则外部电路能够自主发起控制门电路的 $\overline{\text{EWM_out}}$ 信号。

在复位之后，如果 EWM 禁用，或者 CTRL 寄存器的 INEN 位被清除，则 $\overline{\text{EWM_in}}$ 信号会被忽略。

在使能 EWM (设置 CTRL[EWMEN]位) 和使能 $\overline{\text{EWM_in}}$ 功能性 (设置 CTRL[INEN]位) 时，在 CPU 开始刷新 EWM 之前， $\overline{\text{EWM_in}}$ 信号必须处于被解除认定的状态。这可以确保 $\overline{\text{EWM_out}}$ 保持被解除认定状态；否则， $\overline{\text{EWM_out}}$ 输出信号会被认定。

注

在使能 EWM 之前，用户必须更新 CMPH 和 CMPL 寄存器。使能 EWM 之后，计数器复位为 0，因此，在对外部监测电路做上电复位之后，用户应提供一个合理的稳定时间。此外，用户还应确保 $\overline{\text{EWM_in}}$ 引脚已被解除认定。

24.5.3 EWM 计数器

它是一个 8 位纹波计数器，从独立于外设总线时钟源的时钟源计数。如果偏好的超时时间在 1 ms 和 100 ms 之间，则实际的时钟源应在 kHz 范围。

在 CPU 复位之后，或 EWM 刷新完成之后，或在计数器溢出时，计数器复位为 0。CPU 无法访问计数器数值。

24.5.4 EWM 比较寄存器

比较寄存器 CMPL 和 CMPH 在 CPU 复位后一次写入，且直到再次发生 CPU 复位之前，都无法更改。

EWM 比较寄存器用于创建刷新窗口，以便刷新 EWM 模块。

将 CMPL 和 CMPH 设定为相同的值是不合法的。在这种情况下，只要计数器达到 (CMPL + 1)， $\overline{\text{EWM_out}}$ 即被置为有效。

24.5.5 EWM 刷新机制

除了 EWM 的初始配置以外，CPU 只能通过 EWM 服务寄存器访问 EWM。CPU 必须在 CMPL 和 CMPH 寄存器指定的窗口时间帧以内正确写入唯一的数据，以此来访问 EWM 服务寄存器，从而实现正确的 EWM 刷新操作。因此，可能发生三种情况：

表 24-10. EWM 刷新机制

条件	机制
在以下情况下，EWM 刷新操作完成： CMPL < 计数器 < CMPH。	软件照常工作，EWM 计数器复位为 0。在 EWM 刷新操作期间，如果 EWM_in 输入处于解除认定状态，则 EWM_out 输出信号保持解除认定状态..
当计数器 < CMPL，EWM 刷新操作完成	软件先于窗口时间帧刷新 EWM，计数器复位为 0 且 EWM_out 输出信号被认定（无关输入 EWM_in）。
在 EWM 刷新操作完成之前，计数器数值达到 CMPH。	软件并未刷新 EWM。EWM 计数器复位为 0 且 EWM_out 输出信号被认定（无关输入 EWM_in）。

24.5.6 EWM 中断

当 $\overline{\text{EWM_out}}$ 被认定后，会生成一个中断请求，以表示认定 EWM 复位输出信号。此中断在 CTRL[INTEN]置位时使能。清除此位会清除中断请求，但不会影响 $\overline{\text{EWM_out}}$ 。 $\overline{\text{EWM_out}}$ 只能通过迫使系统复位来解除对此信号的认定。

24.5.7 计数器时钟预分频器

EWM 计数器时钟源由时钟分频器通过编程 CLKPRESCALER[CLK_DIV]预分频。此分频时钟用于运行 EWM 计数器。

注

用于运行 EWM 计数器的分频时钟的频率不得超过总线时钟频率的一半。

第 25 章 看门狗定时器 (WDOG)

25.1 此模块的芯片实现细节

25.1.1 WDOG 时钟

此表显示了 WDOG 模块时钟和相应的芯片时钟。

表 25-1. WDOG 时钟连接

模块时钟	芯片时钟
LPO 振荡器	1 kHz LPO 时钟
Alt 时钟	总线时钟
快速测试时钟	总线时钟
系统总线时钟	总线时钟

25.1.2 WDOG 低功耗模式

此表显示 WDOG 低功耗模式和对应的芯片低功耗模式。

表 25-2. WDOG 低功耗模式

模块模式	芯片模式
等待	等待、VLPW
停止	停止、VLPS
休眠	LLS、VLLSx

25.2 简介

看门狗定时器 (WDOG) 可监控系统功能, 并在系统出现故障时进行重置。故障原因包括: 软件代码跑飞和系统时钟停止, 这会导致安全关键系统出现严重后果。在此情况下, 看门狗将使系统进入安全操作状态。看门狗通过预期定期的软件通讯来对系统操作进行监控, 通常被称为服务或刷新看门狗。如果此周期性刷新没有发生, 则看门狗将复位系统。

25.3 特性

看门狗定时器 (WDOG) 的特性包括:

- 独立于 CPU/总线时钟的时钟源输入。在两个时钟源之间选择:
 - 低功耗振荡器(LPO)
 - 外部系统时钟
- 允许对仅可一次写入 WDOG 控制/配置位更新的解锁顺序。
- 在解锁的 256 个总线时钟周期内, 所有 WDOG 控制/配置位仅可写入一次。
 - 在 256 个总线时钟周期内解锁之后, 用户需要始终更新这些位。未更新这些位将会使系统复位。
- 可编程超时周期, 以 WDOG 时钟周期数指定
- 可以测试 WDOG 定时器并通过表明看门狗测试的标志位复位
 - 快速测试—为快速测试设定的短超时值。
 - 字节测试—一次一个测试定时器的单个字节。
 - 只读访问 WDOG 定时器—允许动态检查 WDOG 定时器是否运行。

注

当看门狗在总线时钟上运行时, 读取看门狗定时器计数器得出的计数器值可能不准确。

- 窗口刷新选项
 - 提供程序流程快于预期的鲁棒性检查。
 - 可编程窗口
 - 刷新外部窗口将导致复位。

- 可靠刷新机制
 - 在 20 个总线时钟周期内将值 0xA602 和 0xB480 写入 WDOG 刷新寄存器。
- 发生复位时的 WDOG 复位计数
- 用于调试 Breadcrumb 的可配置超时中断。随后在 256 个总线时钟周期之后复位。

25.4 功能概述

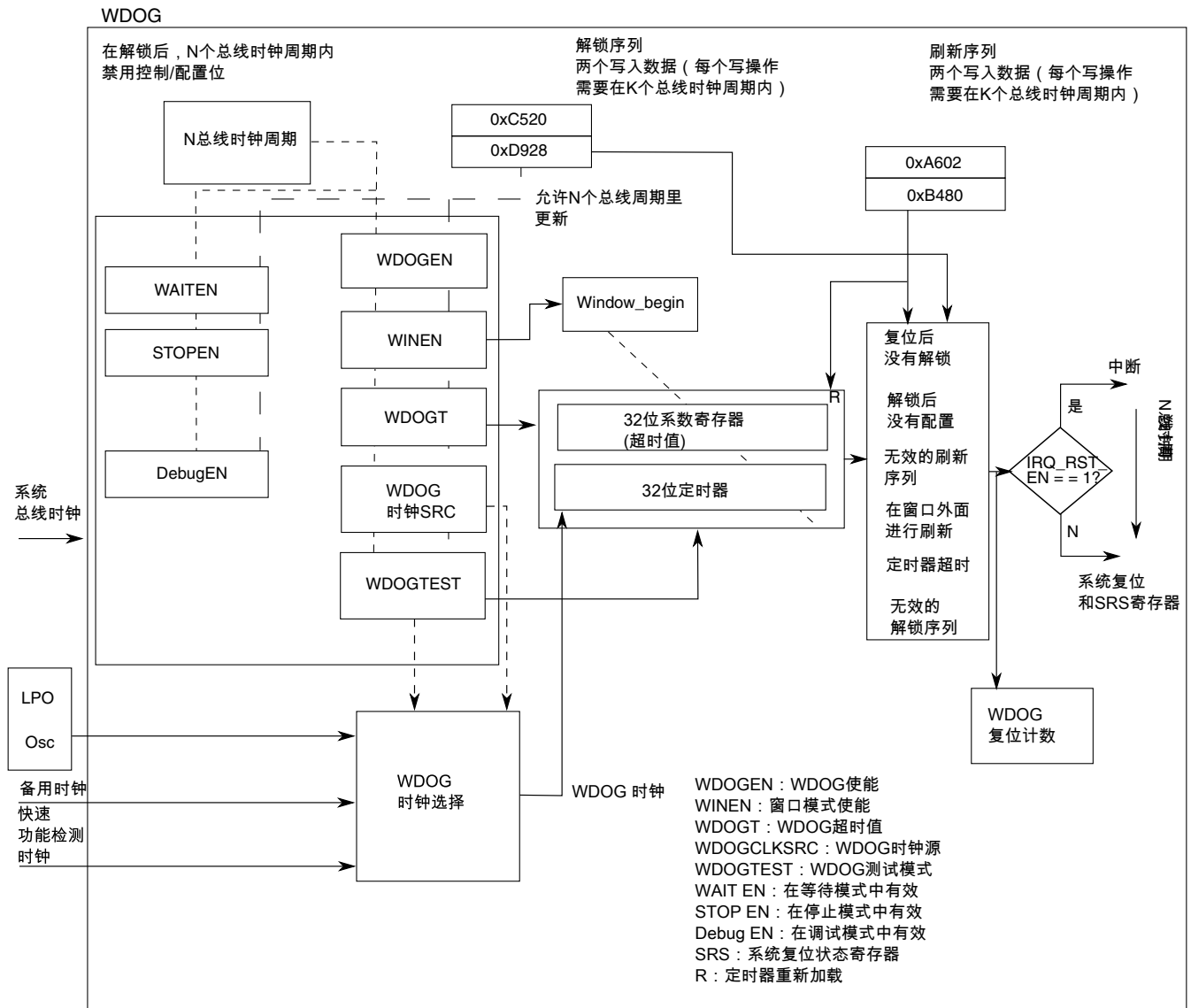


图 25-1. WDOG 操作

上图所示为看门狗的操作。N 和 K 的值如下：

- N = 256
- K = 20

WDOG 模块提供故障安全机制，确保系统在出现故障时（比如 CPU 时钟停摆或软件代码出现跑飞）能够复位至已知状态。看门狗计数器采用选定的时钟源工作，并判断是否得到定期的更新服务。若非如此，它便会复位系统。这确保软件可以正确执行并且不会使软件跑飞。软件可以调整服务周期或看门狗定时器的超时值，以满足应用需求。

用户可以选择窗口化操作模式，在此模式下，应仅在特定超时周期窗口内进行服务。在此窗口外尝试使用看门狗将导致复位。在此模式下工作时，用户可以得到代码是否以高于预期速度运行的指示。用户也可以编程窗口长度。

如果系统因未知和持续原因而无法更新/刷新看门狗，那么看门狗定时器会一直产生复位。为了分析这种情况的原因，可以将看门狗编程为先发起中断，然后再复位。在中断服务程序中，软件可分析系统协议栈，以帮助调试。

为了提高看门狗对系统的独立性，它运行于独立的 LPO 振荡器时钟。用户还可以通过控制寄存器位（如需要）切换至备选时钟源。

25.4.1 解锁和更新看门狗

只要置位看门狗控制寄存器中的 `ALLOW_UPDATE`，用户就可以解锁并修改仅一次写入控制和配置寄存器：

1. 在 20 总线时钟周期内将 `0xC520`（后跟 `0xD928`）写入特定解锁寄存器 (`WDOG_UNLOCK`)。
2. 等待一个总线时钟周期。用户不能在写入解锁序列后立即在总线时钟周期内更新寄存器。
3. 与看门狗配置时间(WCT)等长的更新窗口将会打开。在此窗口内，用户可以更新配置和控制寄存器位。

解锁后，只能对这些寄存器位进行一次修改。

如果在更新窗口内未更新任何配置和控制寄存器，则看门狗将向系统发出复位（即中断并复位）。在初始解锁后尝试在 WCT 内解锁看门狗无任何作用。在更新操作过程中，看门狗定时器将不会暂停，并继续在后台运行。当更新窗口关闭后，看门狗定时器将重新启动并且看门狗将根据新配置运行。

更新功能对于具有初始、非安全关键部分的应用程序很有用，其中看门狗将保持禁用或具有较长的超时周期。这意味着应用程序编码器无须经常为看门狗服务。当应用程序的关键部分开始后，可以根据需要重新配置看门狗。

对于任何无效解锁序列，看门狗将向系统发出复位（即中断并复位，如果已使能）：

- 将除 0xC520 或 0xD928 以外的任何值写入解锁寄存器。
- ALLOW_UPDATE 将置位并且在写入解锁序列值之间插入超过 20 个总线时钟周期的间隔。

在两次写入解锁序列之间以及在成功解锁后的 WCT 时间内尝试的刷新操作将不会被检测到。另请参阅[通过 8 位访问的看门狗操作](#)，了解对解锁寄存器的 8 位访问的相关指南。

注

解锁和刷新过程中的环境切换可能导致看门狗复位。

25.4.2 看门狗配置时间(WCT)

为了防止意外修改看门狗的控制和配置寄存器位，系统只允许在解锁后的 256 总线时钟周期内更新它们。此周期称为看门狗配置时间(WCT)。此外，在解锁进行编辑甚至是复位后，只能对这些寄存器位进行一次修改。

用户必须在系统复位后在 WCT 内解锁寄存器，在此之后 WDOG 将向系统发起复位。换言之，用户必须在复位后在 WCT 内写入至少第一个解锁序列字。完成此操作后，用户将拥有另一个 20 总线时钟周期（解锁序列字之间允许的最大间隔）来完成解锁操作。在此之后，请确保不要忘记配置看门狗，如果在解锁后未在 WCT 内更新 WDOG 控制和配置寄存器，看门狗将发起复位。在关闭此窗口或首次写入后，这些寄存器位将锁定以防止任何后续更改。

在解锁和更新操作期间看门狗定时器将根据其默认配置保持运行，该配置可扩展至最多 2 个 WCT + 20 总线时钟周期。因此，必须确保看门狗的超时值始终大于 2 个 WCT 时间 + 20 总线时钟周期。

一次写入寄存器中的更新仅在 WCT 窗口关闭后生效，但在以下情况下更改将立即生效：

- 停止、等待和调试模式使能
- IRQ_RST_EN

在 WCT 期间未检测到看门狗刷新操作。

25.4.3 刷新看门狗

已为看门狗选择强大的刷新机制。有效刷新是指在 20 总线时钟周期内向看门狗刷新寄存器写入 0xA602 后再接着写入 0xB480。如果在超过 20 个总线周期内分别写入这两个值或者向寄存器写入除这两个值以外的其他值，则会向系统发起看门狗复位或中断后复位（如已使能）。有效刷新可以使看门狗定时器在下一个总线时钟重启。此外，在两次写入刷新序列之间尝试进行解锁操作将不会被检测到。请参阅[进行 8 位访问的看门狗操作](#)，了解对刷新寄存器进行 8 位访问的相关指南。

25.4.4 窗口化操作模式

在此操作模式下，对超时周期内的时间点有限制，在此周期内，可刷新看门狗。仅当看门狗定时器递增到超出看门狗窗口寄存器规定的特定计数时，刷新才被视为有效。这称之为在总超时周期窗口内刷新看门狗。如果在定时器达到窗口值之前尝试刷新，则看门狗将会生成复位或中断后再复位（如已使能）。如果完全没有刷新，则看门狗将会超时并生成复位或中断后复位（如已使能）。

25.4.5 看门狗禁用操作模式

当看门狗通过看门狗状态和控制寄存器中的 WDOG_EN 位禁用时，看门狗定时器复位为 0 并停止计数，直到用户将其使能或通过系统复位使其再次使能。在此模式下，无法刷新看门狗定时器—定时器禁用时无需进行此操作。但是，如已使能，则在非超时异常时看门狗仍将生成复位或中断后再复位。请参见[生成的复位和中断](#)。使能之前，用户需要解锁看门狗。系统复位将使看门狗退出禁用模式。

25.4.6 调试运行模式

用户可以通过看门狗控制寄存器中的 DBG_EN，将看门狗设为在调试模式下禁用。这将导致看门狗定时器在此模式下暂停。此时仍允许读取/写入寄存器，这意味着允许诸如刷新、解锁之类的操作。退出此模式时，定时器将从暂停处恢复其运行。

解锁之后，系统进入以上所述 必须在 WCT 内强制配置看门狗，除非系统总线时钟关闭，在此情况下，状态机也暂停。未能进行此操作也将导致系统复位、中断后复位（如已使能）。此外，导致系统复位的所有异常条件（如[生成的复位和中断](#)所述）在模式下仍有效。因此，如果发生异常且系统总线时钟开启，则会发生复位或中断后复位（如已使能）。

复位之后, 在 WCT 内进入调试模式的处理方式也不相同。WDOG 定时器保持复位为 0 且无需在 WCT 内对其进行解锁和配置。在此状态下, 用户不得尝试刷新或解锁 WDOG, 否则可能会导致出现未知行为。退出模式时, 看门狗定时器将重新启动, 并且必须在 WCT 内解锁和配置 WDOG。

25.5 测试看门狗

对于 IEC 60730 和其他安全标准, 预计用于监控安全功能的所有部件均需经过测试, 且此测试需具有容错能力。要测试看门狗, 必须测试其主定时器及其关联比较和复位逻辑。目前为止, 可以为看门狗执行两种测试, 如[快速测试](#)和[字节测试](#)中所述。可通过控制位将看门狗置于功能测试模式。此外, 还提供有一个覆盖测试禁用控制位, 用于永久禁用功能测试模式。置位后, 此测试禁用位仅可通过复位清除。

这两种测试均可达到测试计数器功能以及比较和复位逻辑的总体目标。

注

请勿在这些测试过程中使能看门狗中断。如果需要, 用户必须确保有效的超时值大于 WCT 时间。详情请参见[生成的复位和中断](#)。

要运行特定测试:

1. 请选择快速测试或字节测试。
2. 置位某一测试模式位, 将看门狗置于功能测试模式。自动置位此位会将看门狗定时器切换至快速时钟源。切换时钟源可实现更快超时, 从而缩短测试时间。

如果测试成功, 则定时器将在达到设定的超时值之后超时并生成系统复位。

注

因看门狗测试退出复位之后, 请解锁和配置看门狗。在测试模式下, 不会自动禁用刷新和解锁操作以及中断。

25.5.1 快速测试

在此测试下, 看门狗定时器的超时值被设为非常低的值, 以实现快速超时。看门狗快速测试和普通模式的唯一区别在于在快速模式下 TESTWDOG 置位。这允许更快地测试看门狗复位机制。

25.5.2 字节测试

字节测试是一种更彻底的看门狗定时器测试。在此测试下，定时器将被拆分成字节宽度的各级，这些级独立运行并依据对应的超时值寄存器字节测试它们是否超时。下图说解释了拆分概念：

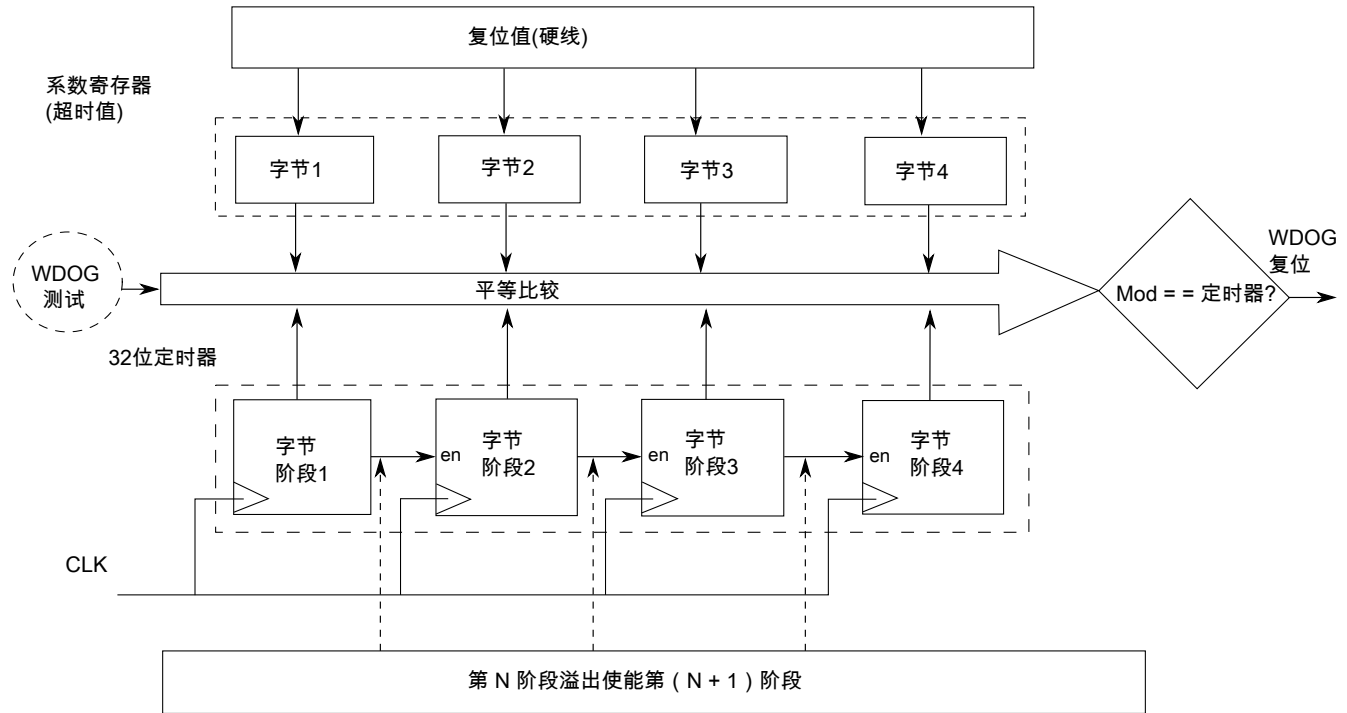


图 25-2. 看门狗定时器字节拆分

每级均为 8 位同步计数器后跟生成溢出信号的组合逻辑。溢出信号用作使能 $N + 1$ 级。

在测试模式下，当测试单个字节 N 时，字节 $N - 1$ 将被强制加载为 $0xFF$ ，并且允许这些字节运行时钟源。这样一来，将会立即生成 $N - 1$ 级的溢出信号，以使能计数器级 N 。第 N 级运行并比较超时值寄存器的第 N 个字节。通过这种方式，也可将字节 N 及其与前一级之间的链路一起测试。字节 N 上的测试未使能任何其他级 $N - 2$ 、 $N - 3$... 和 $N + 1$ 、 $N + 2$...。除了最重要的计数器级之外，这些禁用级将被加载值 $0xFF$ 。

25.6 备份复位生成器

备份复位生成器用于生成系统外的最终复位。它具有备份机制，该机制可确保在总线时钟停止时防止主状态机复位异常/中断，看门狗定时器超时将单独路由输出为系统复位。在无强制系统复位的情况下出现两次连续定时器超时将会导致备份复位生成器将超时信号路由输出为系统复位。

25.7 生成的复位和中断

看门狗在以下事件中生成复位，这些事件也称为例外：

- 看门狗超时
- 系统复位解除置位后，未在 WCT 时间内解锁看门狗
- 解锁后，控制和配置寄存器在 WCT 窗口内无更新。在 WCT 窗口内必须写入以下寄存器中的至少一个，才能避免复位：
 - WDOG_ST_CTRL_H, WDOG_ST_CTRL_L
 - WDOG_TO_VAL_H, WDOG_TO_VAL_L
 - WDOG_WIN_H, WDOG_WIN_L
 - WDOG_PRESCALER
- 将解锁序列或刷新序列以外的值分别写入解锁和/或刷新寄存器。
- 解锁序列两个值的写入之间存在 20 个总线周期以上的间隙。
- 刷新序列两个值的写入之间存在 20 个总线周期以上的间隙。

看门狗也可生成中断。如果 IRQ_RST_EN 置位，则在上述事件中 WDOG_ST_CTRL_L[INT_FLG] 置位，从而生成中断。看门狗复位也在 WCT 时间后生产，以确保看门狗的故障容限。可通过将 1 写入 INT_FLG 来清除中断。

中断与复位之间的 WCT 间隙意味着 WDOG 超时值必须大于 WCT。否则，如果因超时而生成中断，该 WCT 间隙内将发生第二次连续超时。这将触发备份复位生成器，对系统生成复位，提前结束中断服务程序执行。另外，计算看门狗复位次数的任务也无法完成。

25.8 存储器映射和寄存器定义

本节包含存储器映射和寄存器说明。

WDOG 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4005_2000	看门狗状态和控制寄存器高位 (WDOG_STCTRLH)	16	R/W	01D3h	25.8.1/458
4005_2002	看门狗状态和控制寄存器低位 (WDOG_STCTRL_L)	16	R/W	0001h	25.8.2/460

下一页继续介绍此表...

WDOG 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4005_2004	看门狗超时值寄存器高位 (WDOG_TOVALH)	16	R/W	004Ch	25.8.3/460
4005_2006	看门狗超时值寄存器低位 (WDOG_TOVALL)	16	R/W	4B4Ch	25.8.4/460
4005_2008	靠门狗窗口寄存器高位 (WDOG_WINH)	16	R/W	0000h	25.8.5/461
4005_200A	看门狗窗口寄存器低位 (WDOG_WINL)	16	R/W	0010h	25.8.6/461
4005_200C	看门狗刷新寄存器 (WDOG_REFRESH)	16	R/W	B480h	25.8.7/462
4005_200E	看门狗解锁寄存器 (WDOG_UNLOCK)	16	R/W	D928h	25.8.8/462
4005_2010	看门狗定时器输出寄存器高位 (WDOG_TMROUTH)	16	R/W	0000h	25.8.9/462
4005_2012	看门狗定时器输出寄存器低位 (WDOG_TMROUTL)	16	R/W	0000h	25.8.10/463
4005_2014	看门狗复位计数寄存器 (WDOG_RSTCNT)	16	R/W	0000h	25.8.11/463
4005_2016	看门狗预分频器寄存器 (WDOG_PRESC)	16	R/W	0400h	25.8.12/463

25.8.1 看门狗状态和控制寄存器高位 (WDOG_STCTRLH)

地址: 4005_2000h 基准 + 0h 偏移 = 4005_2000h

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读	0	DISTESTWDOG	BYTESEL[1:0]		TESTSEL	TESTWDOG	0	Reserved	WAITEN	STOPEN	DBGEN	ALLOWUPDATE	WINEN	IRQRSTEN	CLKSFC	WDOGEN
写																
复位	0	0	0	0	0	0	0	1	1	1	0	1	0	0	1	1

WDOG_STCTRLH 字段描述

字段	描述
15 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
14 DISTESTWDOG	允许永久禁用 WDOG 的功能测试模式。置位后，它只能通过复位清除。置位后，它不能解锁进行编辑。 0 WDOG 功能测试模式未禁用。 1 WDOG 功能测试模式将永久禁用，直到复位为止。
13-12 BYTESEL[1:0]	当看门狗处于字节测试模式时，此 2 位字段将选择要测试的字节。 00 已选择字节 0 01 已选择字节 1 10 已选择字节 2 11 已选择字节 3
11 TESTSEL	仅当置位 TESTWDOG 时才有效。选择要在看门狗定时器上运行的测试。 0 快速测试。定时器在正常操作下运行。用户可以加载小超时值以执行快速测试。 1 字节测试。将定时器置于字节测试模式，其中可对定时器的各个字节进行操作，并针对超时与对应字节的已编程超时值进行比较。通过 BYTESEL[1:0]选择字节以进行测试。

下一页继续介绍此表...

WDOG_STCTRLH 字段描述 (继续)

字段	描述
10 TESTWDOG	将看门狗置于功能测试模式。在此模式下，将测试看门狗定时器和相关联的比较及复位生成逻辑，以便实现正确操作。定时器时钟将从主看门狗时钟切换到看门狗功能测试的快速时钟输入。TESTSEL 位将选择要运行的测试。
9 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
8 Reserved	此字段为保留字段。
7 WAITEN	在待机模式下使能或禁用 WDOG。 0 WDOG 在 CPU 待机模式下禁用。 1 WDOG 在 CPU 待机模式下使能。
6 STOPEN	在停止模式下使能或禁用 WDOG。 0 WDOG 在 CPU 停止模式下禁用。 1 WDOG 在 CPU 停止模式下使能。
5 DBGEN	在调试模式下使能或禁用 WDOG。 0 WDOG 在 CPU 调试模式下禁用。 1 WDOG 在 CPU 调试模式下使能。
4 ALLOWUPDATE	在复位触发的初始配置窗口(WCT)关闭后，支持通过解锁顺序对看门狗一次写入寄存器进行更新。 0 不允许对 WDOG 一次写入寄存器进行更新。 1 可以对 WDOG 一次写入寄存器解锁以进行更新。
3 WINEN	使能窗口模式。 0 窗口模式禁用。 1 窗口模式使能。
2 IRQRSTEN	用于使能调试 Breadcrumb 功能。对此位所做的更改将自动更新，而不是在 WCT 之后更新。 0 WDOG 超时仅生成复位。 1 WDOG 超时最初生成中断。在 WCT 之后，它将生成复位。
1 CLKSRC	为 WDOG 定时器和其他内部定时操作选择时钟源。 0 WDOG 时钟源自 LPO。 1 WDOG 时钟源自备选时钟源。
0 WDOGEN	使能或禁用 WDOG 的操作。在禁用状态下，看门狗定时器将保持在复位状态，但其他异常情形仍会触发复位/中断。对此位值的更改必须保留超过一个 WDOG_CLK 周期，这样才能使能或禁用 WDOG。 0 WDOG 禁用。 1 WDOG 使能。

25.8.2 看门狗状态和控制寄存器低位 (WDOG_STCTRL)

地址: 4005_2000h 基准 + 2h 偏移 = 4005_2002h

位	15	14	13	12	11	10	9	8
读写	INTFLG	Reserved						
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
读写	Reserved							
复位	0	0	0	0	0	0	0	1

WDOG_STCTRL 字段描述

字段	描述
15 INTFLG	中断标志位。它在出现异常时置位。IRQRSTEN = 1 是设置此标志位的前提条件。INTFLG = 1 会导致中断，随后进行复位、WCT。可通过将 1 写入此位来清除中断。也可通过系统复位进行清除。
Reserved	此字段为保留字段。 注：请勿修改此字段值。

25.8.3 看门狗超时值寄存器高位 (WDOG_TOVALH)

地址: 4005_2000h 基准 + 4h 偏移 = 4005_2004h

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读写	TOVALHIGH															
复位	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0

WDOG_TOVALH 字段描述

字段	描述
TOVALHIGH	定义看门狗定时器的 32 位超时值的高 16 位。它以看门狗时钟周期进行定义。

25.8.4 看门狗超时值寄存器低位 (WDOG_TOVALL)

必须将看门狗的超时值设置为至少四个看门狗时钟周期。在看门狗时钟域内生效的新设置中的延迟将会考虑在内。

地址: 4005_2000h 基准 + 6h 偏移 = 4005_2006h

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读写	TOVALLOW															
复位	0	1	0	0	1	0	1	1	0	1	0	0	1	1	0	0

WDOG_TOVALL 字段描述

字段	描述
TOVALLOW	定义看门狗定时器的 32 位超时值的低 16 位。它以看门狗时钟周期进行定义。

25.8.5 靠门狗窗口寄存器高位 (WDOG_WINH)

注

用户必须将窗口寄存器值设置为低于超时值寄存器。

地址: 4005_2000h 基准 + 8h 偏移 = 4005_2008h

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读写	WINHIGH															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_WINH 字段描述

字段	描述
WINHIGH	定义看门狗窗口操作模式 32 位窗口的高 16 位。它以看门狗时钟周期进行定义。在此模式下, 仅当定时器达到大于或等于此窗口长度的值时, 看门狗才会刷新。在此窗口外部的刷新将会复位系统, 或者如果已置位 IRQRSTEN, 它将中断并复位系统。

25.8.6 看门狗窗口寄存器低位 (WDOG_WINL)

注

用户必须将窗口寄存器值设置为低于超时值寄存器。

地址: 4005_2000h 基准 + Ah 偏移 = 4005_200Ah

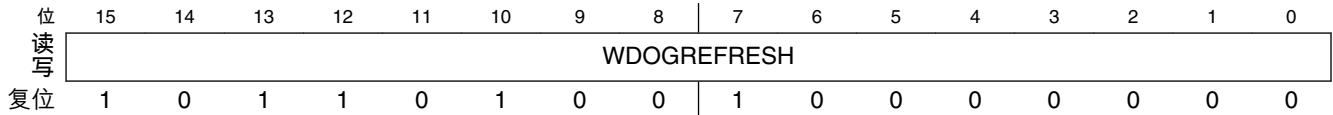
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读写	WINLOW															
复位	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

WDOG_WINL 字段描述

字段	描述
WINLOW	定义看门狗窗口操作模式 32 位窗口的低 16 位。它以预分频看门狗时钟的周期进行定义。在此模式下, 仅当定时器达到大于或等于此窗口长度的值时, 看门狗才会刷新。在此窗口外部的刷新将会复位系统, 或者如果已置位 IRQRSTEN, 它将中断并复位系统。

25.8.7 看门狗刷新寄存器 (WDOG_REFRESH)

地址: 4005_2000h 基准 + Ch 偏移 = 4005_200Ch

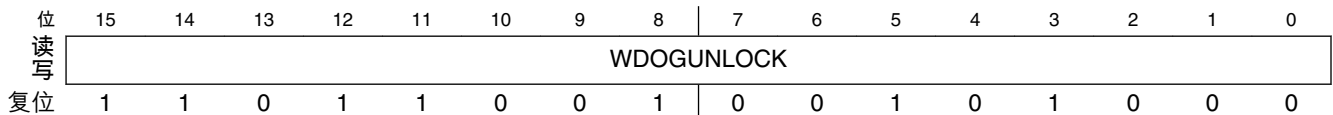


WDOG_REFRESH 字段描述

字段	描述
WDOGREFRESH	看门狗刷新寄存器。写入此寄存器的 20 个总线时钟周期内的 0xA602 序列 (后跟 0xB480) 将刷新 WDOG 并阻止它复位系统。写入除上述序列之外的其他值, 或者如果序列超过 20 个总线周期, 则复位系统, 或如果已置位 IRQRSTEN, 它将中断并复位系统。

25.8.8 看门狗解锁寄存器 (WDOG_UNLOCK)

地址: 4005_2000h 基准 + Eh 偏移 = 4005_200Eh

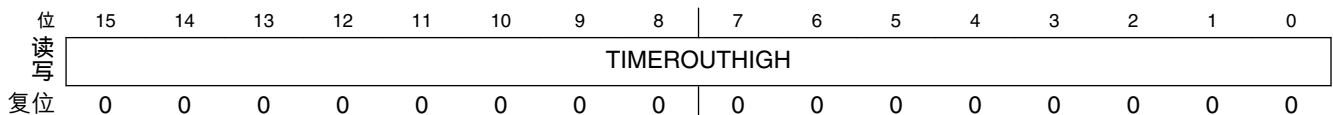


WDOG_UNLOCK 字段描述

字段	描述
WDOGUNLOCK	将解锁序列值写入此寄存器, 使看门狗一次写入寄存器可再次写入。在 20 个总线时钟周期内所需解锁序列为 0xC520 (后跟 0xD928)。有效解锁序列将打开与 WCT 等长的窗口, 可在其中更新寄存器。写入除上述序列之外的其他值, 或者如果序列超过 20 个总线周期, 则复位系统, 或如果已置位 IRQRSTEN, 它将中断并复位系统。仅当置位 ALLOWUPDATE 时, 解锁序列才有效。

25.8.9 看门狗定时器输出寄存器高位 (WDOG_TMROUTH)

地址: 4005_2000h 基准 + 10h 偏移 = 4005_2010h



WDOG_TMROUTH 字段描述

字段	描述
TIMEROUTHIGH	显示看门狗定时器的高 16 位值。

25.8.10 看门狗定时器输出寄存器低位 (WDOG_TMROUTL)

在停止模式下，将在看门狗定时器的预停止值捕获 WDOG_TIMER_OUT。在退出停止模式后，在 WDOG_TIMER_OUT 启动以下看门狗定时器之前，将出现 1 个 WDOG_CLK 周期 + 3 个总线时钟周期的最大延迟。

地址: 4005_2000h 基准 + 12h 偏移 = 4005_2012h

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读	TIMEROUTLOW															
写																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_TMROUTL 字段描述

字段	描述
TIMEROUTLOW	显示看门狗定时器的低 16 位值。

25.8.11 看门狗复位计数寄存器 (WDOG_RSTCNT)

地址: 4005_2000h 基准 + 14h 偏移 = 4005_2014h

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读	RSTCNT															
写																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_RSTCNT 字段描述

字段	描述
RSTCNT	对看门狗复位系统的次数进行计数。该寄存器只会在收到 POR 时复位。通过将 1 写入要清除的位，用户可以清除此寄存器的内容。

25.8.12 看门狗预分频器寄存器 (WDOG_PRESC)

地址: 4005_2000h 基准 + 16h 偏移 = 4005_2016h

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
读	0				PRESCVAL				0							
写																
复位	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

WDOG_PRESC 字段描述

字段	描述
15-11 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
10-8 PRESCVAL	看门狗时钟源的 3 位预分频器。零值表示未分频输入 WDOG 时钟。看门狗时钟将由(PRESCVAL + 1)分频以提供预分频 WDOG_CLK。
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

25.9 通过 8 位访问的看门狗操作

25.9.1 通用准则

对看门狗 16 位寄存器执行 8 位访问（旨在访问两个寄存器字节）时，在代码中相继置放两个 8 位访问。

25.9.2 通过 8 位访问的刷新和解锁操作

生成系统复位的异常情形是向对应的刷新和解锁寄存器写入除合法刷新/更新数序所需值之外的任何值。

对于这些寄存器的 8 位访问，写入正确的值需要至少两个总线时钟周期，从而导致一个周期内的寄存器中存在无效值。因此，即便目的是将正确的值写入刷新/解锁寄存器，系统也会复位。请记住，已对 8 位访问的异常情形稍微做了修改。

虽然刷新/解锁序列的正确值匹配依据的是原始定义，但是，不正确值的匹配则是在刷新/解锁时逐字节进行，而不是整个 16 位值。这意味着，如果刷新/解锁寄存器的高位字节包含的值不同于组成序列的两个值的高位字节值，则它将被作为异常情形对待，从而导致复位或中断后复位。对于刷新或解锁寄存器的低位字节，以上说法同样成立。下面以预期向刷新寄存器写入 0xA602 后再写入 0xB480 的刷新操作为例。

表 25-16. 8 位访问刷新

	WDOG_REFRESH1.58	WDOG_REFRESH[7:0]	序列值 1 或值 2 匹配	不匹配异常
当前值	0xB4	0x80	值 2 匹配	无
写入 1	0xB4	0x02	无匹配	无
写入 2	0xA6	0x02	值 1 匹配	无
写入 3	0xB4	0x02	无匹配	无

下一页继续介绍此表...

表 25-16. 8 位访问刷新 (继续)

	WDOG_REFRESH1.58	WDOG_REFRESH[7:0]	序列值 1 或值 2 匹配	不匹配异常
写入 4	0xB4	0x80	值 2 匹配。序列完成。	无
写入 5	0x02	0x80	无匹配	有

如上表所示，刷新寄存器最初保留其复位值。其后，将在寄存器上执行两次 8 位访问，以写入刷新序列的第一个值。中间写入（写入 1）上将不会记录任何不匹配异常。通过执行另外两次 8 位访问来完成序列，以便写入序列的第二个值，从而实现成功刷新。请务必注意，仅当正确地写入完整的 16 位值（写入 4）时才会发生值 2 匹配。因此，需要通过测量写入 2 与写入 4 之间的间隔来检查是否符合在值 1 的 20 总线时钟周期内写入序列的值 2 这一要求。

重申一下，刷新或解锁序列的值 1 和 2 的匹配情况保持不变。8 位访问的不同之处在于检测不匹配情况的标准没有那么严格。任何 16 位访问仍需符合[刷新看门狗](#)章节中提到的原有准则。

25.10 看门狗操作限制

本节介绍某些用户难以察觉的看门狗操作异常情形。

- 解锁/刷新操作限制 — 在解锁后 WCT 窗口关闭与看门狗定时器实际重载之间的时间内，请不要尝试解锁和刷新操作。
- 在成功配置解锁之后，看门狗定时器更新和重载发生在 WCT 窗口关闭之后的 2-3 个看门狗时钟内。
- 时钟切换延迟 — 看门狗在两个位置使用无毛刺多路复用器 – 一个在 LPO 振荡器输入与备用时钟输入之间选择；另一个在看门狗功能时钟和看门狗功能测试的快速时钟输入之间选择。从请求切换到发生实际时钟切换经过了~2 个时钟 A 周期加上~2 个时钟 B 周期的最大时间周期，其中时钟 A 和 B 为时钟多路复用器的两个输入时钟。
- 对于窗口化模式，在看门狗计数器穿过窗口值与寄存在总线时钟域之间存在 2-3 个总线时钟延迟。
- 为了使看门狗正常工作，看门狗时钟的速度必须始终至少比系统总线时钟的速度慢 5 倍。特殊情况是当看门狗时钟与总线时钟同步时，在此情况下，看门狗时钟的速度可与总线时钟速度一样快。

- WCT 必须等于至少 3 个看门狗时钟周期。如果不能确保，则意味着即便在关闭 WCT 窗口之后，用户也必须在配置更新生效之前等待同步系统复位以取消认定看门狗时钟域。
- 应将看门狗的超时值设置为至少四个看门狗时钟周期。在看门狗时钟域内生效的新设置中的延迟将会考虑在内。
- 用户不仅需要在看门狗定时器的实际超时周期内刷新看门狗，而且还必须提供足够的时间，供看门狗时钟上的看门狗定时器检测到刷新序列。
- 用户不能在写入解锁序列后立即在总线时钟周期内进行更新，但可以在一个总线时钟周期后进行更新。
- 必须确保看门狗的超时值始终大于 2 个 WCT 时间 + 20 总线时钟周期。
- 在两次写入解锁序列之间以及在成功解锁后的 WCT 时间内尝试的刷新操作将不会被检测到。
- 在初始解锁后尝试在 WCT 时间内解锁看门狗无任何作用。
- 在看门狗功能测试模式下，不会自动禁用刷新和解锁操作以及中断。
- 退出因看门狗功能测试导致的复位之后，用户仍需要执行解锁和配置看门狗这些必要的步骤。看门狗将继续处于功能测试模式，因此，用户将在复位 WCT 时间内使看门狗退出功能测试模式。
- 退出因看门狗功能测试导致的复位之后，用户仍需要执行解锁和配置看门狗这些必要的步骤。
- 用户必须确保在时钟切换过程中无毛刺时钟多路复用器的两个时钟输入均有效。否则，可能会导致输出时钟丢失。
- 因同步问题，从进入停止模式到看门狗定时器实际停止之间存在 2-3 个看门狗时钟周期间隔。退出停止模式时同样如此，此时间导致定时器重启时存在 2-3 个看门狗时钟周期延迟。如果停止模式的持续时间少于一个看门狗时钟周期，则无法保证看门狗定时器停止。
- 考虑写入第一个刷新值时的情况，在此之后，系统将进入停止模式且系统总线时钟仍运行。如果在第一个值的 20 总线周期内未写入第二个刷新值，则系统将会复位或中断后复位（如已使能）。

第 26 章 多用途时钟产生器 (MCG)

26.1 此模块的芯片实现细节

26.1.1 MCG 振荡器时钟输入选项

MCG 有多个振荡器输入时钟源。对 MCG 而言，它们均被称为外部参考时钟，通过 MCG_C7[OSCSEL] 位域加以选择。下表列出了针对该位域的具体芯片时钟分配情况。

表 26-1. MCG 振荡器参考选项

MCG_C7[OSCSEL]	MCG 定义选择	芯片时钟
00	OSCCLK0 - 系统振荡器	OSCCLK - 未分频的系统振荡器输出。源自外部晶振电路或直接源自 EXTAL。
01	OSC2/RTC 振荡器	RTC 32kHz 振荡器输出。RTC 时钟源自与 RTC 相关的外部晶振电路。
10	OSCCLK1 - 振荡器	IRC48MCLK. 源自内部 48 MHz 振荡器。
11	保留	—

有关这些时钟的详情请参见[时钟分配](#)部分。

26.1.2 MCG 实例化信息

注

对于本芯片，存储器映射中的寄存器 MCG_C12/S2/T3 均为保留位。

26.2 简介

多用途时钟产生器 (MCG) 模块为 MCU 提供了多个时钟源选择。

该模块包括一个锁频环 (FLL) 和一个锁相环 (PLL)。FLL 由内部或外部参考时钟控制。PLL 由外部参考时钟控制。该模块可选择 FLL 或 PLL 输出时钟，或一个参考时钟 (内部或外部) 作为 MCU 系统的时钟源。MCG 与晶体振荡器结合工作，以允许外部晶振、陶瓷谐振器或其他外部时钟源生成外部参考时钟。

26.2.1 特性

MCU 模块的主要特性有：

- 锁频环 (FLL):
 - 数控振荡器 (DCO)
 - DCO 频率范围可以编程，最多支持 4 个不同频率范围。
 - 针对低频外部参考时钟源提供 DCO 输出频率可编程和最大化选项。
 - 在不改变 FLL 参考频率时，提供选项可以防止 FLL 在切换时钟模式时重置其当前锁定频率。
 - 可使用内部或外部参考时钟作为 FLL 源。
 - 可以用作其他片上外设的时钟源。
- 锁相环 (PLL):
 - 电压控制振荡器 (VCO)
 - 用外部参考时钟作为 PLL 源。
 - 模数 VCO 分频器。
 - 相位/频率检测器。
 - 集成环路滤波器。
 - 可以用作其他片上外设的时钟源。
- 内部参考时钟产生器:
 - 慢速时钟 (带 9 个调整位以确保准确性)
 - 快速时钟 (带 4 个调整位)

- 可用作 FLL 的源时钟。在 FEI 模式下，只有慢速内部参考时钟 (IRC) 可以用作 FLL 源。
- 可以选择慢速或快速时钟作为 MCU 的时钟源。
- 可以用作其他片上外设的时钟源。
- 提供用于 MCG 外部参考低功耗振荡器时钟产生器的控制信号：
 - HGO、RANGE、EREFS
- 来自晶体振荡器 的外部时钟：
 - 可用作 FLL 和/或 PLL 的时钟源。
 - 可以选择作为 MCU 的时钟源。
- 来自实时计数器的外部时钟 (RTC):
 - 只能用作 FLL 的时钟源。
 - 可以选择作为 MCU 的时钟源。
- 带复位和中断请求功能的外部时钟监控器，用于检测运行 FBE、PEE、BLPE 或 FEE 模式时的外部时钟故障
- 与 PLL 配合使用的锁定检测器，带中断请求功能
- 内部参考时钟自动调整机 (ATM) 功能，以外部时钟为参考
- 提供针对 FLL 和 PLL 的参考分频器。
- 提供针对快速内部参考时钟的参考分频器。
- 提供 MCG PLL 时钟 (MCGPLLCLK) 作为其他片上外设的时钟源。
- 提供 MCG FLL 时钟 (MCGFLLCLK) 作为其他片上外设的时钟源。
- 提供 MCG 固定频率时钟 (MCGFFCLK) 作为其他片上外设的时钟源。
- 提供 MCG 内部参考时钟 (MCGIRCLK) 作为其他片上外设的时钟源。

下图是 MCG 模块的方框图。

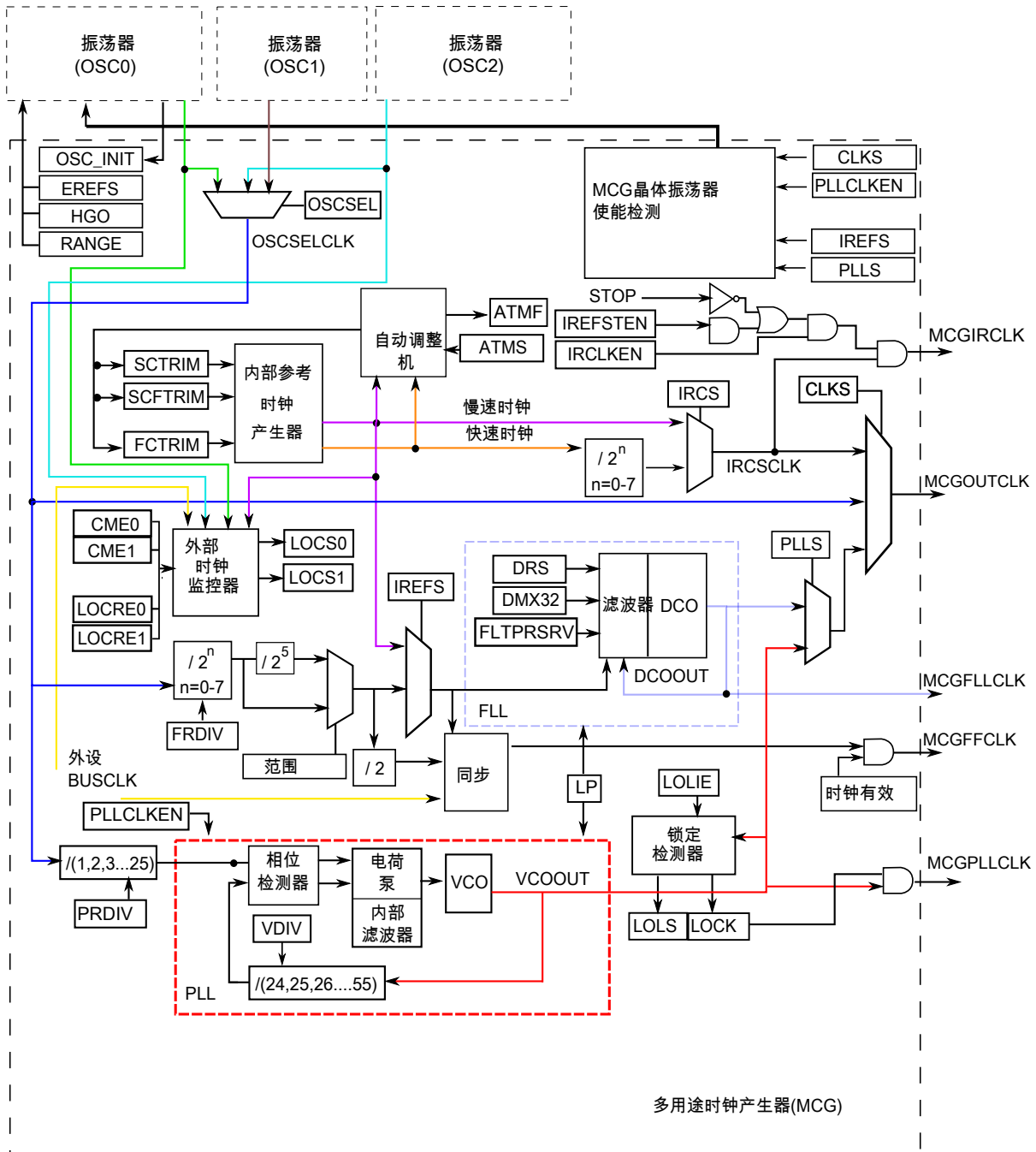


图 26-1. 多用途时钟产生器 (MCG) 框图

26.2.2 操作模式

MCG 具有以下操作模式：FEI、FEE、FBI、FBE、PBE、PEE、BLPI、BLPE、和停止。详情请参见 [MCG 工作模式](#)。

26.3 外部信号说明

没有任何 MCG 信号被连接到芯片外部。

26.4 存储器映射/寄存器定义

本节包括存储器映射和寄存器定义。

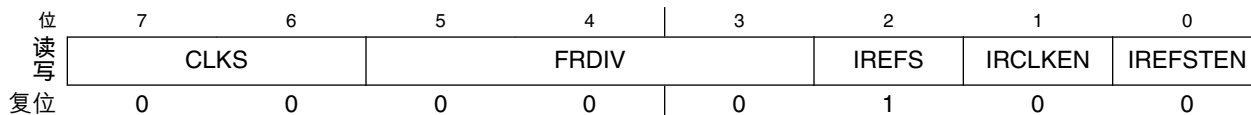
这些 MCG 寄存器仅可在管理员模式下写入。在用户模式下进行写入访问将导致总线错误。可在管理模式和用户模式下执行读取访问。

MCG 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_4000	MCG 控制 1 寄存器 (MCG_C1)	8	R/W	04h	26.4.1/472
4006_4001	MCG 控制 2 寄存器 (MCG_C2)	8	R/W	80h	26.4.2/473
4006_4002	MCG 控制 3 寄存器 (MCG_C3)	8	R/W	未定义	26.4.3/474
4006_4003	MCG 控制 4 寄存器 (MCG_C4)	8	R/W	未定义	26.4.4/474
4006_4004	MCG 控制 5 寄存器 (MCG_C5)	8	R/W	00h	26.4.5/475
4006_4005	MCG 控制 6 寄存器 (MCG_C6)	8	R/W	00h	26.4.6/476
4006_4006	MCG 状态寄存器 (MCG_S)	8	R	10h	26.4.7/478
4006_4008	MCG 状态和控制寄存器 (MCG_SC)	8	R/W	02h	26.4.8/479
4006_400A	MCG 自动调整比较值高电平寄存器 (MCG_ATCVH)	8	R/W	00h	26.4.9/480
4006_400B	MCG 自动调整比较值低电平寄存器 (MCG_ATCVL)	8	R/W	00h	26.4.10/481
4006_400C	MCG 控制 7 寄存器 (MCG_C7)	8	R/W	00h	26.4.11/481
4006_400D	MCG 控制 8 寄存器 (MCG_C8)	8	R/W	80h	26.4.12/482
4006_4011	MCG 控制 12 寄存器 (MCG_C12)	8	R/W	00h	26.4.13/482
4006_4012	MCG 状态 2 寄存器 (MCG_S2)	8	R/W	00h	26.4.13/483
4006_4013	MCG 测试 3 寄存器 (MCG_T3)	8	R/W	00h	26.4.13/483

26.4.1 MCG 控制 1 寄存器 (MCG_C1)

地址: 4006_4000h 基准 + 0h 偏移 = 4006_4000h



MCG_C1 字段描述

字段	描述
7-6 CLKS	<p>时钟源选择</p> <p>选择 MCGOUTCLK 的时钟源。</p> <p>00 编码 0 — 选择 FLL 或 PLL 输出 (取决于 PLLS 控制位)。</p> <p>01 编码 1 — 选择内部参考时钟。</p> <p>10 编码 2 — 选择外部参考时钟。</p> <p>11 编码 3 — 保留。</p>
5-3 FRDIV	<p>FLL 外部参考分频器</p> <p>选择 FLL 外部参考时钟的降频参数。得出的频率必须在范围 31.25 kHz 至 39.0625 kHz 之间 (当 FLL/DCO 为 MCGOUTCLK 的时钟源时, 这是必须的。在 FBE 模式下, 无需满足此范围要求, 但建议在尝试从 FBE 进入 FLL 模式时满足此范围要求)。</p> <p>000 如果 RANGE = 0 或 OSCSEL=1, 则分频系数为 1; 对于所有其他 RANGE 值, 分频系数为 32。</p> <p>001 如果 RANGE = 0 或 OSCSEL=1, 则分频系数为 2; 对于所有其他 RANGE 值, 分频系数为 64。</p> <p>010 如果 RANGE = 0 或 OSCSEL=1, 则分频系数为 4; 对于所有其他 RANGE 值, 分频系数为 128。</p> <p>011 如果 RANGE = 0 或 OSCSEL=1, 则分频系数为 8; 对于所有其他 RANGE 值, 分频系数为 256。</p> <p>100 如果 RANGE = 0 或 OSCSEL=1, 则分频系数为 16; 对于所有其他 RANGE 值, 分频系数为 512。</p> <p>101 如果 RANGE = 0 或 OSCSEL=1, 则分频系数为 32; 对于所有其他 RANGE 值, 分频系数为 1024。</p> <p>110 如果 RANGE = 0 或 OSCSEL=1, 则分频系数为 64; 对于所有其他 RANGE 值, 分频系数为 1280。</p> <p>111 如果 RANGE = 0 或 OSCSEL=1, 则分频系数为 128; 对于所有其他 RANGE 值, 分频系数为 1536。</p>
2 IREFS	<p>内部参考选择</p> <p>选择 FLL 的参考时钟源。</p> <p>0 选择外部参考时钟。</p> <p>1 选择内部慢速参考时钟。</p>
1 IRCLKEN	<p>内部参考时钟使能</p> <p>使能内部参考时钟用作 MCGIRCLK。</p> <p>0 MCGIRCLK 无效。</p> <p>1 MCGIRCLK 有效。</p>
0 IREFSTEN	<p>内部参考停止使能</p> <p>控制内部参考时钟在 MCG 进入停止模式时是否保持使能。</p> <p>0 内部参考时钟在停止模式下禁用。</p> <p>1 如果 IRCLKEN 置位, 或者 MCG 在进入停止模式之前处于 FEI、FBI 或 BLPI 模式, 则内部参考时钟在停止模式下保持使能。</p>

26.4.2 MCG 控制 2 寄存器 (MCG_C2)

地址: 4006_4000h 基准 + 1h 偏移 = 4006_4001h

位 读 写	7	6	5	4	3	2	1	0
	LOCRE0	FCFTRIM	RANGE		HGO	EREFS	LP	IRCS
复位	1	0	0	0	0	0	0	0

MCG_C2 字段描述

字段	描述
7 LOCRE0	<p>时钟丢失复位使能</p> <p>决定在 OSC0 外部参考时钟丢失后是否生成中断或复位请求。只有当 CME0 置位时 LOCRE0 才有效。</p> <p>0 在 OSC0 外部参考时钟丢失时生成中断请求。 1 在 OSC0 外部参考时钟丢失时生成复位请求</p>
6 FCFTRIM	<p>快速内部参考时钟精密调整</p> <p>FCFTRIM 控制快速内部参考时钟频率的微调。FCFTRIM 置位将加长参考时钟周期，FCFTRIM 清零将缩短周期，调整量为最小可能的量。如果 FCFTRIM 值存储在非易失性存储器中，用户需要将存储器上的值拷贝到该寄存器位中。</p>
5-4 RANGE	<p>频率范围选择</p> <p>选择晶体振荡器或外部时钟源的频率范围。更多细节请参考振荡器 (OSC) 章节，各频率范围的具体数值请查阅数据手册。</p> <p>00 编码 0 — 为晶体振荡器选择的低频率范围。 01 编码 1 — 为晶体振荡器选择的高频率范围。 1X 编码 2 — 为晶体振荡器选择的超高频率范围。</p>
3 HGO	<p>高增益振荡器选择</p> <p>控制晶体振荡器的操作模式。更多详情，请参见振荡器(OSC)章节。</p> <p>0 将晶体振荡器配置为低功耗操作。 1 将晶体振荡器配置为高增益操作。</p>
2 EREFS	<p>外部参考选择</p> <p>选择外部参考时钟的源。更多详情，请参见振荡器(OSC)章节。</p> <p>0 选择外部参考时钟。 1 选择振荡器。</p>
1 LP	<p>低功耗选择</p> <p>控制 FLL 或 PLL 在 BLPI 和 BLPE 模式下是否禁用。在 FBE 或 PBE 模式下，将此位置位为 1 将使 MCG 转换为 BLPE 模式；在 FBI 模式下，将此位置位为 1 将使 MCG 转换为 BLPI 模式。在任何其他 MCG 模式下，LP 位不起作用。</p> <p>0 FLL 或 PLL 在旁路模式下不禁用。 1 FLL 或 PLL 在旁路模式下禁用 (低功耗)，</p>
0 IRCS	<p>内部参考时钟选择</p> <p>在快速或慢速内部参考时钟源之间选择。</p>

下一页继续介绍此表...

MCG_C2 字段描述 (继续)

字段	描述
0	选择慢速内部参考时钟。
1	选择快速内部参考时钟。

26.4.3 MCG 控制 3 寄存器 (MCG_C3)

地址: 4006_4000h 基准 + 2h 偏移 = 4006_4002h

位	7	6	5	4	3	2	1	0
读	SCTRIM							
写	SCTRIM							
复位	x*	x*	x*	x*	x*	x*	x*	x*

* 注:

- x = 复位时未定义。

MCG_C3 字段描述

字段	描述
SCTRIM	<p>慢速内部参考时钟调整设置</p> <p>SCTRIM¹ 通过控制内部慢速参考时钟周期来控制慢速内部参考时钟频率。STRIM 位采用二进制加权，也就是说，位 1 的调整量是位 0 的两倍。提高二进制值会加长周期，降低该值会缩短周期。</p> <p>C4 寄存器中的 SCFTRIM 位可用作附加微调位。复位时，此值将加载为出厂调整值。</p> <p>如果 SCTRIM 值存储在非易失性存储器中，用户需要将存储器上的值拷贝到该寄存器位中。</p>

1. 在复位过程中，调整值将从一个工厂编程位置载入到 SCTRIM 寄存器中。

26.4.4 MCG 控制 4 寄存器 (MCG_C4)

地址: 4006_4000h 基准 + 3h 偏移 = 4006_4003h

位	7	6	5	4	3	2	1	0
读	DMX32	DRST_DRS		FCTRIM			SCFTRIM	
写	DMX32	DRST_DRS		FCTRIM			SCFTRIM	
复位	0	0	0	x*	x*	x*	x*	x*

* 注:

- x = 复位时未定义。
- x = 复位时未定义。

MCG_C4 字段描述

字段	描述
7 DMX32	<p>DCO 最大频率 (带 32.768 kHz 参考)</p> <p>DMX32 位控制是否将 DCO 频率范围缩小到其最大频率 (带 32.768 kHz 参考)。</p> <p>下表所示为 DCO 频率范围的设置。</p> <p>注: 来自此时钟源的系统时钟不得超过其指定的最大值。</p>

下一页继续介绍此表...

MCG_C4 字段描述 (继续)

字段	描述					
	DRST_DRS	DMX32	参考范围	FLL 因子	DCO 范围	
	00	0	31.25–39.0625 kHz	640	20–25 MHz	
		1	32.768 kHz	732	24 MHz	
	01	0	31.25–39.0625 kHz	1280	40–50 MHz	
		1	32.768 kHz	1464	48 MHz	
	10	0	31.25–39.0625 kHz	1920	60–75 MHz	
		1	32.768 kHz	2197	72 MHz	
	11	0	31.25–39.0625 kHz	2560	80–100 MHz	
		1	32.768 kHz	2929	96 MHz	
	0 DCO 的默认范围为 25%。 1 DCO 经过微调以达到最大频率 (带 32.768 kHz 参考)。					
	6–5 DRST_DRS	DCO 范围选择 DRS 位用于选择 FLL 输出 DCOOUT 的频率范围。当 LP 位置位时，对 DRS 位进行的写入操作将被忽略。DRST 读取字段表示 DCOOUT 当前的频率范围。由于时钟域之间的内部同步，DRST 字段不会在写入 DRS 后立即更新。更多详情，请参阅 DCO 频率范围表。 00 编码 0 — 低频范围 (复位默认值)。 01 编码 1 — 中频范围。 10 编码 2 — 中高频范围。 11 编码 3 — 高频范围。				
	4–1 FCTRIM	快速内部参考时钟调整设置 FCTRIM ¹ 通过控制内部快速参考时钟周期来控制快速内部参考时钟频率。FCTRIM 位采用二进制加权，也就是说，位 1 的调整量是位 0 的两倍。提高二进制值会加长周期，降低该值会缩短周期。 如果 FCTRIM[3:0] 值存储在非易失性存储器中，用户需要将存储器上的值拷贝到该寄存器位中。				
0 SCFTRIM	慢速内部参考时钟精密调整 SCFTRIM ² 对内部慢速参考时钟频率进行微小调整。SCFTRIM 置位将加长周期，SCFTRIM 清零将缩短周期，调整量为最小可能的量。 如果 SCFTRIM 值存储在非易失性存储器中，则用户需要将存储器上的值拷贝到该寄存器位中。					

- 在复位过程中，调整值将从一个工厂编程位置载入到 FCTRIM 寄存器中。
- 在复位过程中，调整值将从一个工厂编程位置载入到 SCFTRIM 寄存器中。

26.4.5 MCG 控制 5 寄存器 (MCG_C5)

地址: 4006_4000h 基准 + 4h 偏移 = 4006_4004h

位	7	6	5	4	3	2	1	0
读	0	PLLCLKEN0	PLLSTEN0		PRDIV0			
写	0							
复位	0	0	0	0	0	0	0	0

MCG_C5 字段描述

字段	描述																																																																																																			
7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。																																																																																																			
6 PLLCLKEN0	<p>PLL 时钟使能</p> <p>使能 PLL (无需设置 PLLS) 并使能 PLL 时钟作为 MCGPLLCLK。(置位 PLLCLKEN 0 位之前, 需要正确设置分频 PRDIV 0 , 使生成的 PLL 参考时钟频率在 2 - 4 MHz 范围内。) 置位 PLLCLKEN 0 将使能外部振荡器 (如尚未使能)。任何时候, 如果 PLL 通过 PLLCLKEN 0 位使能且外部振荡器用作参考时钟, 则应检查 OSCINIT 0 位以确保其已置位。</p> <p>0 MCGPLLCLK 无效。 1 MCGPLLCLK 有效。</p>																																																																																																			
5 PLLSTEN0	<p>PLL 停止使能</p> <p>在正常停止模式下使能 PLL 时钟。在低功耗停止模式下, 即使 PLLSTEN 0 =1, PLL 时钟也会被禁用。在所有其他低功耗模式下, 如果写入 1, 则 PLLSTEN 0 位不起作用且不会使能 PLL 时钟运行。</p> <p>0 MCGPLLCLK 在任何停止模式下均禁用。 1 如果系统处于正常停止模式, 则 MCGPLLCLK 使能。</p>																																																																																																			
PRDIV0	<p>PLL 外部参考分频器</p> <p>选择 PLL 外部参考时钟的降频参数。得出的频率范围必须在 2 MHz 至 4 MHz 之间。使能 PLL 之后(通过置位 PLLCLKEN 0 或 PLLS), 当 LOCK 0 为零时, 不得更改 PRDIV 0 值。</p> <p style="text-align: center;">表 26-8. PLL 外部参考分频系数</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>PRDIV 0</th> <th>分频系数</th> <th></th> <th>PRDIV 0</th> <th>分频系数</th> <th></th> <th>PRDIV 0</th> <th>分频系数</th> <th></th> <th>PRDIV 0</th> <th>分频系数</th> </tr> </thead> <tbody> <tr> <td>00000</td> <td>1</td> <td></td> <td>01000</td> <td>9</td> <td></td> <td>10000</td> <td>17</td> <td></td> <td>11000</td> <td>25</td> </tr> <tr> <td>00001</td> <td>2</td> <td></td> <td>01001</td> <td>10</td> <td></td> <td>10001</td> <td>18</td> <td></td> <td>11001</td> <td>保留</td> </tr> <tr> <td>00010</td> <td>3</td> <td></td> <td>01010</td> <td>11</td> <td></td> <td>10010</td> <td>19</td> <td></td> <td>11010</td> <td>保留</td> </tr> <tr> <td>00011</td> <td>4</td> <td></td> <td>01011</td> <td>12</td> <td></td> <td>10011</td> <td>20</td> <td></td> <td>11011</td> <td>保留</td> </tr> <tr> <td>00100</td> <td>5</td> <td></td> <td>01100</td> <td>13</td> <td></td> <td>10100</td> <td>21</td> <td></td> <td>11100</td> <td>保留</td> </tr> <tr> <td>00101</td> <td>6</td> <td></td> <td>01101</td> <td>14</td> <td></td> <td>10101</td> <td>22</td> <td></td> <td>11101</td> <td>保留</td> </tr> <tr> <td>00110</td> <td>7</td> <td></td> <td>01110</td> <td>15</td> <td></td> <td>10110</td> <td>23</td> <td></td> <td>11110</td> <td>保留</td> </tr> <tr> <td>00111</td> <td>8</td> <td></td> <td>01111</td> <td>16</td> <td></td> <td>10111</td> <td>24</td> <td></td> <td>11111</td> <td>保留</td> </tr> </tbody> </table>	PRDIV 0	分频系数		PRDIV 0	分频系数		PRDIV 0	分频系数		PRDIV 0	分频系数	00000	1		01000	9		10000	17		11000	25	00001	2		01001	10		10001	18		11001	保留	00010	3		01010	11		10010	19		11010	保留	00011	4		01011	12		10011	20		11011	保留	00100	5		01100	13		10100	21		11100	保留	00101	6		01101	14		10101	22		11101	保留	00110	7		01110	15		10110	23		11110	保留	00111	8		01111	16		10111	24		11111	保留
PRDIV 0	分频系数		PRDIV 0	分频系数		PRDIV 0	分频系数		PRDIV 0	分频系数																																																																																										
00000	1		01000	9		10000	17		11000	25																																																																																										
00001	2		01001	10		10001	18		11001	保留																																																																																										
00010	3		01010	11		10010	19		11010	保留																																																																																										
00011	4		01011	12		10011	20		11011	保留																																																																																										
00100	5		01100	13		10100	21		11100	保留																																																																																										
00101	6		01101	14		10101	22		11101	保留																																																																																										
00110	7		01110	15		10110	23		11110	保留																																																																																										
00111	8		01111	16		10111	24		11111	保留																																																																																										

26.4.6 MCG 控制 6 寄存器 (MCG_C6)

地址: 4006_4000h 基准 + 5h 偏移 = 4006_4005h

	7	6	5	4	3	2	1	0
位 读 写	LOLIE0	PLLS	CME0	VDIV0				
复位	0	0	0	0	0	0	0	0

MCG_C6 字段描述

字段	描述																																																																																																			
7 LOLIE0	<p>失锁中断使能</p> <p>决定在失锁指示出现后是否生成中断请求。该位仅在 LOLS 0 置位时有效。</p> <p>0 失锁时不生成中断请求。 1 失锁时生成中断请求。</p>																																																																																																			
6 PLLS	<p>PLL 选择</p> <p>控制当 CLK[S1:0]=00 时是将 PLL 还是 FLL 选作 MCG 源。如果已清除 PLLS 位且未置位 PLLCLKEN 0，则 PLL 在所有模式下均禁用。如果已置位 PLLS，则 FLL 在所有模式下均禁用。</p> <p>0 选择 FLL。 1 选择 PLL (在设置 PLLS 位之前，需要正确设置分频 PRDIV 0，使生成的 PLL 参考时钟频率在 2-4 MHz 范围内)。</p>																																																																																																			
5 CME0	<p>时钟监控器使能</p> <p>使能 OSC0 外部参考多路复用器选择的时钟丢失监控电路。LOCRE0 位决定 OSC0 时钟丢失指示之后是否生成中断或复位请求。仅当 MCG 处于使用外部时钟的工作模式 (FEE、FBE、PEE、PBE 或 BLPE) 时，才必须将 CME0 位设置为逻辑 1。当 CME0 位设置为逻辑 1 时，不得更改 C2 寄存器中的 RANGE0 位的值。在 MCG 进入任何停止模式之前，应向 CME0 位写入逻辑 0。否则，可能会在停止模式下发生复位请求。如果 MCG 处于 BLPE 模式，那么在进入 VLPR 或 VLPW 前，CME0 也应设为逻辑 0。</p> <p>0 OSC0 的外部时钟监控器禁用。 1 OSC0 的外部时钟监控器使能。</p>																																																																																																			
VDIV0	<p>VCO 0 分频器</p> <p>选择 PLL VCO 输出的分频比。VDIV 0 位用于确定应用到参考时钟频率的乘法因数(M)。使能 PLL 之后 (通过置位 PLLCLKEN 0 或 PLLS)，当 LOCK 0 为零时，不得更改 VDIV 0 值。</p> <p style="text-align: center;">表 26-10. PLL VCO 分频系数</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>VDIV 0</th> <th>乘法系数</th> <th></th> <th>VDIV 0</th> <th>乘法系数</th> <th></th> <th>VDIV 0</th> <th>乘法系数</th> <th></th> <th>VDIV 0</th> <th>乘法系数</th> </tr> </thead> <tbody> <tr> <td>00000</td> <td>24</td> <td></td> <td>01000</td> <td>32</td> <td></td> <td>10000</td> <td>40</td> <td></td> <td>11000</td> <td>48</td> </tr> <tr> <td>00001</td> <td>25</td> <td></td> <td>01001</td> <td>33</td> <td></td> <td>10001</td> <td>41</td> <td></td> <td>11001</td> <td>49</td> </tr> <tr> <td>00010</td> <td>26</td> <td></td> <td>01010</td> <td>34</td> <td></td> <td>10010</td> <td>42</td> <td></td> <td>11010</td> <td>50</td> </tr> <tr> <td>00011</td> <td>27</td> <td></td> <td>01011</td> <td>35</td> <td></td> <td>10011</td> <td>43</td> <td></td> <td>11011</td> <td>51</td> </tr> <tr> <td>00100</td> <td>28</td> <td></td> <td>01100</td> <td>36</td> <td></td> <td>10100</td> <td>44</td> <td></td> <td>11100</td> <td>52</td> </tr> <tr> <td>00101</td> <td>29</td> <td></td> <td>01101</td> <td>37</td> <td></td> <td>10101</td> <td>45</td> <td></td> <td>11101</td> <td>53</td> </tr> <tr> <td>00110</td> <td>30</td> <td></td> <td>01110</td> <td>38</td> <td></td> <td>10110</td> <td>46</td> <td></td> <td>11110</td> <td>54</td> </tr> <tr> <td>00111</td> <td>31</td> <td></td> <td>01111</td> <td>39</td> <td></td> <td>10111</td> <td>47</td> <td></td> <td>11111</td> <td>55</td> </tr> </tbody> </table>	VDIV 0	乘法系数		VDIV 0	乘法系数		VDIV 0	乘法系数		VDIV 0	乘法系数	00000	24		01000	32		10000	40		11000	48	00001	25		01001	33		10001	41		11001	49	00010	26		01010	34		10010	42		11010	50	00011	27		01011	35		10011	43		11011	51	00100	28		01100	36		10100	44		11100	52	00101	29		01101	37		10101	45		11101	53	00110	30		01110	38		10110	46		11110	54	00111	31		01111	39		10111	47		11111	55
VDIV 0	乘法系数		VDIV 0	乘法系数		VDIV 0	乘法系数		VDIV 0	乘法系数																																																																																										
00000	24		01000	32		10000	40		11000	48																																																																																										
00001	25		01001	33		10001	41		11001	49																																																																																										
00010	26		01010	34		10010	42		11010	50																																																																																										
00011	27		01011	35		10011	43		11011	51																																																																																										
00100	28		01100	36		10100	44		11100	52																																																																																										
00101	29		01101	37		10101	45		11101	53																																																																																										
00110	30		01110	38		10110	46		11110	54																																																																																										
00111	31		01111	39		10111	47		11111	55																																																																																										

26.4.7 MCG 状态寄存器 (MCG_S)

地址: 4006_4000h 基准 + 6h 偏移 = 4006_4006h

位	7	6	5	4	3	2	1	0
读	LOLS0	LOCK0	PLLST	IREFST	CLKST		OSCINIT0	IRCST
写								
复位	0	0	0	1	0	0	0	0

MCG_S 字段描述

字段	描述
7 LOLS0	<p>失锁状态</p> <p>此位为粘滞位，用于指示 PLL 的锁定状态。如果获取锁定之后 LOLS 置位，则 PLL 输出频率落在失锁频率容差范围 D_{unl} 之外。LOLIE 决定 LOLS 置位时是否生成中断请求。LOLRE 决定 LOLS 置位时是否生成复位请求。此位在置位时通过复位或写入逻辑 1 清零。向此位写入逻辑 0 不起作用。</p> <p>0 PLL 自上次 LOLS 0 清零以来未失锁。 1 PLL 自上次 LOLS 0 清零以来已失锁。</p>
6 LOCK0	<p>锁定状态</p> <p>此位指示 PLL 是否已获取锁定。只有在 PLL 使能之后，锁定检测才会使能（通过锁定模式选择或 PLLCLKEN0=1 置位）。当 PLL 时钟锁定为所需频率时，MCG PLL 时钟(MCGPLLCLK) 将关断，直到锁定位有效。如果已置位锁定状态位，则更改 C5 寄存器中 PRDIV 0 [4:0]位或 C6 寄存器中 VDIV0[4:0]位的值将导致锁定状态位清零，并且保持清零，直到 PLL 获取锁定。PLL 参考时钟丢失也会导致 LOCK0 位清零，直到 PLL 重新获取锁定。进入 LLS、VLPS 或常规 STOP 模式且 PLLSTEN=0 也会导致锁定状态位清零，并且保持清零，直到退出 STOP 模式且 PLL 重新获取锁定。只要 PLL 使能且 LOCK0 位清零，MCGPLLCLK 将关断，直到 LOCK0 位再次有效。</p> <p>0 PLL 当前已解锁。 1 PLL 当前已锁定。</p>
5 PLLST	<p>PLL 选择状态</p> <p>此位指示 PLLS 选择的时钟源。由于时钟域之间的内部同步，PLLST 位不会在写入 PLLS 位后立即更新。</p> <p>0 PLLS 时钟来源于 FLL 时钟。 1 PLLS 时钟来源于 PLL 输出时钟。</p>
4 IREFST	<p>内部参考时钟状态</p> <p>此位指示 FLL 参考时钟的当前来源。由于时钟域之间的内部同步，IREFST 位不会在写入 IREFS 位后立即更新。</p> <p>0 FLL 参考时钟来源于外部参考时钟。 1 FLL 参考时钟来源于内部参考时钟。</p>
3-2 CLKST	<p>时钟模式状态</p> <p>此位指示当前时钟模式。由于时钟域之间的内部同步，CLKST 位不会在写入 CLKS 位后立即更新。</p> <p>00 编码 0 — 选择 FLL 输出（复位默认值）。 01 编码 1 — 选择内部参考时钟。</p>

下一页继续介绍此表...

MCG_S 字段描述 (继续)

字段	描述
	10 编码 2 — 选择外部参考时钟。 11 编码 3 — 选择 PLL 输出。
1 OSCINIT0	OSC 初始化 晶体 振荡器的初始化周期完成之后, 此位 (复位值为 0) 被置位为 1。置位后, 如果随后禁用 OSC, 则此位将清零。更多信息, 请参阅 OSC 模块详细说明。
0 IRCST	内部参考时钟状态 IRCST 位表示内部参考时钟选择时钟(IRCSCCLK)的当前源。由于时钟域之间的内部同步, IRCST 位不会在写入 IRCST 位后立即更新。仅当内部参考时钟使能时才会更新 IRCST 位, 方法是使 MCG 处于使用 IRC 的模式或置位 C1[IRCLKEN]位。 0 内部参考时钟源为慢速时钟(32 kHz IRC)。 1 内部参考时钟源为快速时钟(4 MHz IRC)。

26.4.8 MCG 状态和控制寄存器 (MCG_SC)

地址: 4006_4000h 基准 + 8h 偏移 = 4006_4008h

位	7	6	5	4	3	2	1	0
读	ATME	ATMS	ATMF	FLTPRSRV	FCRDIV			LOCS0
写			w1c					w1c
复位	0	0	0	0	0	0	1	0

MCG_SC 字段描述

字段	描述
7 ATME	自动调整机使能 使能自动调整机, 以自动开始调整选定的内部参考时钟。 注: 自动调整机完成 ARMS 位选择的所有 IRCS 时钟调整位调整之后, ATME 将撤销。写入 C1、C3、C4 和 SC 寄存器或进入 STOP 模式将中止自动调整机工作并将此位清零。 0 自动调整机已禁用。 1 自动调整机已使能。
6 ATMS	自动调整机选择 为自动调整测试选择 IRCS 时钟。 0 已选择 32 kHz 内部参考时钟。 1 已选择 4 MHz 内部参考时钟。
5 ATMF	自动调整机故障标志 自动调整机(ATM)故障标志。当自动调整机使能, ATME=1, 并且检测到 C1、C3、C4 和 SC 寄存器的写入操作, 或者 MCG 进入任何 STOP 模式时此位置位。写入 ATMF 将清除此标志。

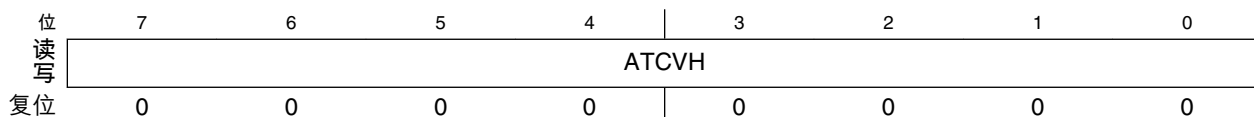
下一页继续介绍此表...

MCG_SC 字段描述 (继续)

字段	描述
	0 自动调整机已正常完成。 1 自动调整机故障。
4 FLTPRSRV	FLL 滤波器保留使能 此位将防止 FLL 滤波器值复位，从而使 FLL 输出频率在时钟模式更改过程（其中，FLL/DCO 输出仍有效）中保持相同。（注：这需要 FLL 参考频率在新的时钟模式切换之前保持相同。否则，FLL 滤波器和频率值将更改。） 0 FLL 滤波器和 FLL 频率将在更改为当前时钟模式时复位。 1 FLL 滤波器和 FLL 频率在新的时钟模式更改过程中保留其先前的值。
3-1 FCRDIV	快速内部参考时钟分频器 选择快速内部参考时钟的分频参数。得出的频率将在范围 31.25 kHz 至 4 MHz 之间（注：不支持在使能快速 IRC 时更改分频器）。 000 分频系数为 1 001 分频系数为 2。 010 分频系数为 4。 011 分频系数为 8。 100 分频系数为 16 101 分频系数为 32 110 分频系数为 64 111 分频系数为 128。
0 LOCS0	OSC0 时钟丢失状态 LOCS0 指示发生了 OSC0 参考时钟丢失。LOCS0 位仅在 CME0 置位时有效。此位在置位时通过写入逻辑 1 清零。 0 未发生 OSC0 丢失。 1 已发生 OSC0 丢失。

26.4.9 MCG 自动调整比较值高电平寄存器 (MCG_ATCVH)

地址: 4006_4000h 基准 + Ah 偏移 = 4006_400Ah

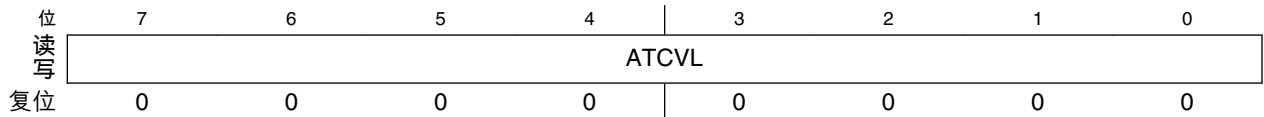


MCG_ATCVH 字段描述

字段	描述
ATCVH	ATM 比较值高电平 自动调整机使用这些值比较和调整 ATM SAR 转换过程中的内部参考时钟调整值。

26.4.10 MCG 自动调整比较值低电平寄存器 (MCG_ATCVL)

地址: 4006_4000h 基准 + Bh 偏移 = 4006_400Bh

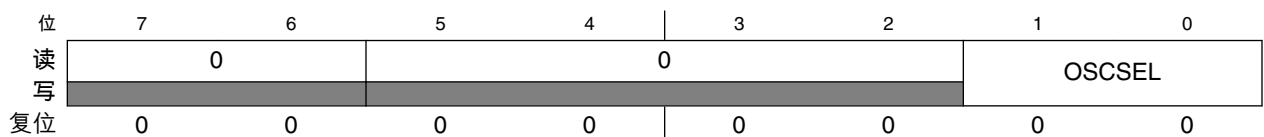


MCG_ATCVL 字段描述

字段	描述
ATCVL	ATM 比较值低电平 自动调整机使用这些值比较和调整 ATM SAR 转换过程中的内部参考时钟调整值。

26.4.11 MCG 控制 7 寄存器 (MCG_C7)

地址: 4006_4000h 基准 + Ch 偏移 = 4006_400Ch

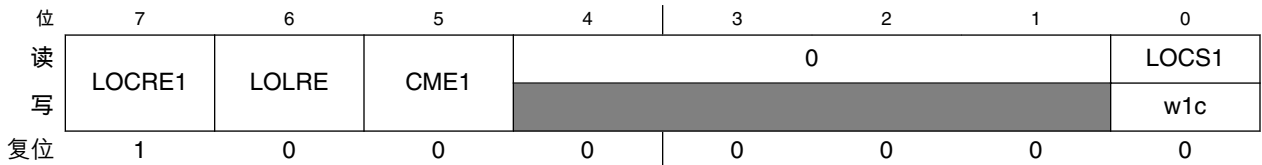


MCG_C7 字段描述

字段	描述
7-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5-2 保留	保留 此字段为保留字段。 此只读字段为保留字段且值始终为 0。
OSCSEL	MCG OSC 时钟选择 选择 MCG FLL 外部参考时钟 注: 当外部时钟用作 MCG 的时钟源时, OSCSEL 字段在 MCG 模式 (比如 PBE) 下不能更改。 00 选择振荡器 (OSCCLK0)。 01 选择 32 kHz RTC 振荡器。 10 选择振荡器 (OSCCLK1)。 11 保留

26.4.12 MCG 控制 8 寄存器 (MCG_C8)

地址: 4006_4000h 基准 + Dh 偏移 = 4006_400Dh



MCG_C8 字段描述

字段	描述
7 LOCRE1	<p>时钟丢失复位使能</p> <p>决定在 RTC 外部参考时钟丢失后是否生成中断或复位请求。仅当置位 CME1 时 LOCRE1 才有效。</p> <p>0 在 RTC 外部参考时钟丢失时生成中断请求。 1 在 RTC 外部参考时钟丢失时生成复位请求</p>
6 LLORE	<p>PLL 失锁复位使能</p> <p>决定在 PLL 失锁出现后是否生成中断或复位请求。</p> <p>0 PLL 失锁指示时生成中断请求。必须置位 PLL 失锁中断使能位才能生成中断请求。 1 PLL 失锁指示时生成复位请求。</p>
5 CME1	<p>时钟监控器使能 1</p> <p>使能 RTC 外部参考时钟的时钟丢失监控电路。LOCRE1 位决定 RTC 时钟丢失指示之后是否生成中断或复位请求。当 MCG 处于将此 RTC 用作外部参考时钟的工作模式或当 RTC 处于工作状态时，应将 CME1 位设为逻辑 1。在 MCG 进入任何停止模式之前，必须向 CME1 位写入逻辑 0。否则，可能会在停止模式下发生复位请求。进入 VLPR 或 VLPW 前，CME1 也应设为逻辑 0。</p> <p>0 RTC 时钟的外部时钟监控器禁用。 1 RTC 时钟的外部时钟监控器使能。</p>
4-1 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
0 LOCS1	<p>RTC 时钟丢失状态</p> <p>此位指示何时发生时钟丢失。此位在置位时通过写入逻辑 1 清零。</p> <p>0 未发生 RTC 丢失。 1 已发生 RTC 丢失</p>

26.4.13 MCG 控制 12 寄存器 (MCG_C12)

地址: 4006_4000h 基准 + 11h 偏移 = 4006_4011h



MCG_C12 字段描述

字段	描述
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

26.4.13 MCG 状态 2 寄存器 (MCG_S2)

地址: 4006_4000h 基准 + 12h 偏移 = 4006_4012h

位	7	6	5	4	3	2	1	0
读	0							
写								
复位	0	0	0	0	0	0	0	0

MCG_S2 字段描述

字段	描述
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

26.4.13 MCG 测试 3 寄存器 (MCG_T3)

地址: 4006_4000h 基准 + 13h 偏移 = 4006_4013h

位	7	6	5	4	3	2	1	0
读	0							
写								
复位	0	0	0	0	0	0	0	0

MCG_T3 字段描述

字段	描述
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

26.5 功能说明**26.5.1 MCG 模式状态图**

MCG 的 9 种状态如下图所示且在 [表 26-20](#) 中进行了描述。箭头表示允许的 MCG 模式转换。

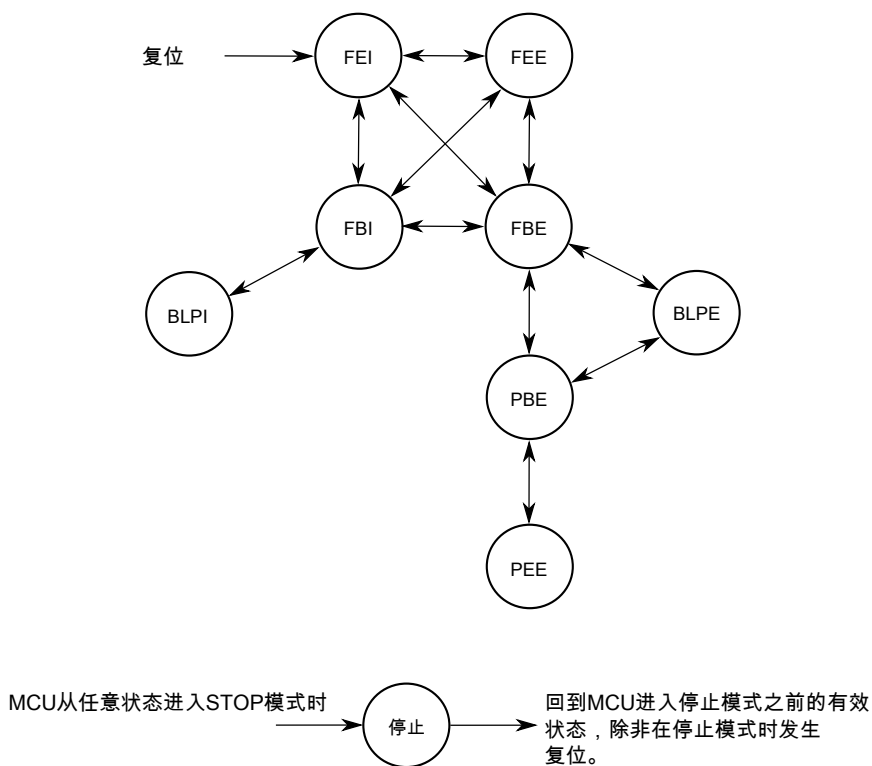


图 26-17. MCG 模式状态图

注

- MCG 处于 PEE 模式时，在退出 LLS 或 VLPS 的过程中，MCG 将复位至 PBE 时钟模式且 C1[CLKS]和 S[CLKST]将自动置位为 2'b10。
- MCG 处于 PEE 模式且 PLLSTEN=0 时，如果进入正常停止模式，MCG 将复位至 PBE 时钟模式且 C1[CLKS]和 S[CLKST]将自动置位为 2'b10。

26.5.1.1 MCG 工作模式

MCG 以下列模式之一运行。

注

MCG 限制模式之间的转换。有关允许的转换，请参见图 26-17。

表 26-20. MCG 工作模式

模式	说明
FLL 内部启用 (FEI)	FLL 内部启用 (FEI) 为默认工作模式，在出现全部下列条件时将进入：

下一页继续介绍此表...

表 26-20. MCG 工作模式 (继续)

模式	说明
	<ul style="list-style-type: none"> • 00 被写入 C1[CLKS]。 • 1 被写入 C1[IREFS]。 • 0 被写入 C6[PLLS]。 <p>在 FEI 模式下, MCGOUTCLK 源自 FLL 时钟 (DCOCLK), 后者由 32 kHz 内部参考时钟 (IRC) 控制。FLL 环将 DCO 频率锁定为 C4[DRST_DRS]和 C4[DMX32]位选定的 FLL 因子乘以内部参考频率。更多详情, 请参见 C4[DMX32]位描述。在 FEI 模式下, 除非置位 C5[PLLCLKEN], PLL 在低功耗状态下均禁用。</p>
FLL 外部启用 (FEE)	<p>出现以下所有条件时, 进入 FLL 外部启用 (FEE) 模式:</p> <ul style="list-style-type: none"> • 00 被写入 C1[CLKS]。 • 0 被写入 C1[IREFS]。 • 对 C1[FRDIV]进行写操作可将外部参考时钟分频到 31.25 kHz 至 39.0625 kHz 范围内。 • 0 被写入 C6[PLLS]。 <p>在 FEE 模式下, MCGOUTCLK 源自 FLL 时钟 (DCOCLK), 后者由外部参考时钟控制。FLL 环将 DCO 频率锁定为 C4[DRST_DRS]和 C4[DMX32]位选定的 FLL 因子乘以外部参考频率, 由 C1[FRDIV]和 C2[RANGE]指定。更多详情, 请参见 C4[DMX32]位描述。在 FEE 模式下, 除非置位 C5[PLLCLKEN], PLL 在低功耗状态下均禁用。</p>
FLL 旁路内部 (FBI)	<p>出现以下所有条件时, 进入 FLL 内部旁路 (FBI) 模式:</p> <ul style="list-style-type: none"> • 01 被写入 C1[CLKS]。 • 1 被写入 C1[IREFS]。 • 0 被写入 C6[PLLS]。 • 0 被写入 C2[LP]。 <p>在 FBI 模式下, MCGOUTCLK 源自 C2[IRCS]位选择的慢速 (32 kHz IRC) 或快速 (4 MHz IRC) 内部参考时钟。FLL 运行但其输出未被使用。此模式非常有用, 用于在 MCGOUTCLK 源自 C2[IRCS]选定的内部参考时钟时允许 FLL 获取其目标频率。FLL 时钟(DCOCLK)由慢速内部参考时钟控制, 并且 DCO 时钟频率将锁定为 C4[DRST_DRS]和 C4[DMX32]位选择的乘数因子乘以内部参考频率。更多详情, 请参见 C4[DMX32]位描述。在 FBI 模式下, 除非置位 C5[PLLCLKEN], PLL 在低功耗状态下禁用。</p>
FLL 外部旁路(FBE)	<p>出现以下所有条件时, 进入 FLL 外部旁路 (FBE) 模式:</p> <ul style="list-style-type: none"> • 10 被写入 C1[CLKS]。 • 0 被写入 C1[IREFS]。 • 对 C1[FRDIV]进行写操作可将外部参考时钟分频到 31.25 kHz 至 39.0625 kHz 范围内。 • 0 被写入 C6[PLLS]。 • 0 被写入 C2[LP]。 <p>在 FBE 模式下, MCGOUTCLK 源自 OSCSEL 外部参考时钟。FLL 运行但其输出未被使用。此模式非常有用, 用于在 MCGOUTCLK 源自外部参考时钟时允许 FLL 获取其目标频率。FLL 时钟(DCOCLK)由外部参考时钟控制, 并且 DCO 时钟频率将锁定为 C4[DRST_DRS]和 C4[DMX32]位选择的乘数因子乘以分频的外部参考时钟。更多详情, 请参见 C4[DMX32]位描述在 FBE 模式下, 除非置位 C5[PLLCLKEN], PLL 在低功耗状态下禁用。</p>
PLL 外部启用 (PEE)	<p>出现以下所有条件时, 进入 PLL 外部启用 (PEE) 模式:</p> <ul style="list-style-type: none"> • 00 被写入 C1[CLKS]。

下一页继续介绍此表...

表 26-20. MCG 工作模式 (继续)

模式	说明
	<ul style="list-style-type: none"> • 0 被写入 C1[IREFS]。 • 1 被写入 C6[PLLS]。 <p>在 PEE 模式下, MCGOUTCLK 源自 PLL, 它由 外部参考时钟控制。PLL 时钟频率锁定为乘数因子 (由其相应的 VDIV 指定) 乘以选定的 PLL 参考频率 (由相应的 PRDIV 指定)。必须配置 PLL 的可编程参考分频器才能生成有效的 PLL 参考时钟。FLL 在低功耗状态下禁用。</p>
PLL 外部旁通 (PBE)	<p>出现以下所有条件时, 进入 PLL 外部旁路 (PBE) 模式:</p> <ul style="list-style-type: none"> • 10 被写入 C1[CLKS]。 • 0 被写入 C1[IREFS]。 • 1 被写入 C6[PLLS]。 • 0 被写入 C2[LP]。 <p>在 PBE 模式下, MCGOUTCLK 源自 OSCSEL 外部参考时钟; PLL 可正常工作, 但其输出时钟未被使用。此模式非常有用, 用于在 MCGOUTCLK 通过外部参考时钟驱动时允许 PLL 获取其目标频率。PLL 时钟频率锁定为其[VDIV]指定的乘数因子乘以由其 PRDIV 选定的 PLL 参考频率。在准备向 PEE 转换时, 必须配置 PLL 的可编程参考分频器才能生成有效的 PLL 参考时钟。FLL 在低功耗状态下禁用。</p>
内部旁通低功耗 (BLPI)	<p>出现以下所有条件时, 将进入内部旁通低功耗 (BLPI) 模式:</p> <ul style="list-style-type: none"> • 01 被写入 C1[CLKS]。 • 1 被写入 C1[IREFS]。 • 0 被写入 C6[PLLS]。 • 1 被写入 C2[LP]。 <p>在 BLPI 模式下, MCGOUTCLK 源自内部参考时钟。即使 C5[PLLCLKEN]设为 1, FLL 和 PLL 也会被禁用。</p>
外部旁通低功耗 (BLPE)	<p>出现以下所有条件时, 将进入外部旁通低功耗 (BLPE) 模式:</p> <ul style="list-style-type: none"> • 10 被写入 C1[CLKS]。 • 0 被写入 C1[IREFS]。 • 1 被写入 C2[LP]。 <p>在 BLPE 模式下, MCGOUTCLK 源自 OSCSEL 外部参考时钟。即使 C5[PLLCLKEN] 设为 1, FLL 和 PLL 也会禁用。</p>
停止	<p>一旦 MCU 进入停止状态则进入。功耗模式取决于芯片。有关功耗模式分配, 请参见描述如何配置模块以及从停止退出期间 MCG 行为的章节。进入停止模式之后, FLL 禁用, 并且所有 MCG 时钟信号为静态, 除了以下情况:</p> <p>当 PLLSTEN=1 时, MCGPLLCLK 在正常停止模式下激活。</p> <p>在以下所有条件为真时, MCGIRCLK 将在常规停止模式下激活:</p> <ul style="list-style-type: none"> • C1[IRCLKEN] = 1 • C1[IREFSTEN] = 1

表 26-20. MCG 工作模式

模式	说明
	注: <ul style="list-style-type: none"> 当进入低功耗停止模式 (LLS 或 VLPS), MCG 时钟模式强制退出 PEE 时钟模式进入 PBE 时钟模式。C1[CLKS]和 S[CLKST]将配置为 2'b10,S[LOCK]将被清除, S[LOLS]不会置位。 当进入普通停止模式, 如果 C5[PLLSTEN]为 0, MCG 时钟模式强制退出 PEE 时钟模式进入 PBE 时钟模式, C1[CLKS]和 S[CLKST]将配置为 2'b10,S[LOCK]将被清除, S[LOLS]不会置位。如果 C5[PLLSTEN]为 1, S[LOCK]不会清除, 且 MCG 继续在 PEE 模式下运行。

注

有关芯片特性的工作模式, 请参见此 MCU 的电源管理章节。

26.5.1.2 MCG 模式切换

C1[IREFS]可随时进行更改, 但 S[IREFST]会显示实际最新选择的参考时钟。在内部使用模式和外部使用模式间切换完成后, FLL 将再次开始锁定。

C1[CLKS]也可随时进行更改, 但 S[CLKST]会显示实际最新选择的时钟。如果新选择的时钟不可用, 将继续选择上一个时钟。

除了当 C2[LP]位为 1 之外, 可随时更改 C4[DRST_DRS]写入位。如果在 FLL 内部使用(FEI)或 FLL 外部使用(FEE)模式更改 C4[DRST_DRS]写入位, 则 MCGOUTCLK 将在三个 DCO 时钟内切换至新选择 DCO 范围。切换至新的 DCO 范围 (由更新的 C4[DRST_DRS]读位表示) 之后, FLL 将在若干个参考周期内保持解锁。FLL 锁定时间会显示在器件数据手册中, 如 $t_{fll_acquire}$ 。

26.5.2 低功耗位使用

C2[LP]用于允许禁用 FLL 或者 PLL, 从而在不使用这些系统时节省功耗。C2[LP]为 1 时, C4[DRST_DRS]无法写入。但在某些应用程序中, 可能需要使能 FLL 或 PLL, 且在切换至启用模式之前, 允许其锁定以获得最大精度。通过向 C2[LP]写入 0 来实现。

26.5.3 MCG 内部参考时钟

此模块支持两个标称频率为 32 kHz (慢速 IRC) 和 4 MHz (快速 IRC) 的内部参考时钟。快速 IRC 频率可通过对 FCRDIV 进行编程降频, 以使频率范围在 32 kHz 至 4 MHz 之间。

26.5.3.1 MCG 内部参考时钟

MCG 内部参考时钟 (MCGIRCLK) 可作为其他片上外设的时钟源, 并且在 C1[IRCLKEN]=1 时使能。使能后, MCGIRCLK 通过快速内部参考时钟 (4 MHz IRC, 可通过 FRDIV 降频) 或慢速内部参考时钟(32 kHz IRC)驱动。可通过调整 IRCS 选定的内部参考时钟周期来重新设定 IRCS 时钟频率。这可通过向 C3[SCTRIM]:C4[SCFTRIM] (如果选择慢速 IRC 时钟) 写入新值或向 C4[FCTRIM]:C2[FCFTRIM] (如果选择快速 IRC 时钟) 写入新值来实现。内部参考时钟周期与写入的调整值成比例。如果 MCG 处于 FBI 或 BLPI 模式, 则 C3[SCTRIM]:C4[SCFTRIM] (如果 C2[IRCS]=0) 和 C4[FCTRIM]:C2[FCFTRIM] (如果 C2[IRCS]=1) 位影响 MCGOUTCLK 频率。如果 MCG 处于 FEI 模式, 则 C3[SCTRIM]:C4[SCFTRIM] (如果 C2[IRCS]=0) 位也会影响 MCGOUTCLK 频率。

此外, 可以通过设置 C1[IRCLKEN]和 C1[IREFSTEN]在停止模式下使能此时钟, 否则, 此时钟将在停止模式下禁用。

26.5.4 外部参考时钟

MCG 模块在所有模式下均支持外部参考时钟。有关外部参考频率范围, 请参见器件数据手册。置位 C1[IREFS]时, FLL 或 PLL 不会使用外部参考时钟。在这些模式下, 频率能达到芯片时序规格支持的最大频率。

如果置位任何 CME 位, 则会使能慢速内部参考时钟以及外部时钟监视器。当 C6[CME0]=1 时, 如果 OSC0 外部参考频率低于最低频率 (f_{loc_high} 或 f_{loc_low} , 具体取决于 C2[RANGE0]), 则会检测到时钟丢失。当 C8[CME1]=1 时, 如果 RTC 外部参考频率降至低于最低频率 (f_{loc_low}), 则会检测到时钟丢失。

注

所有时钟监控器必须在进入这些低功耗模式前禁用: 停止、VLPS、VLPR、VLPW、LLS 和 VLLSx。

检测到时钟丢失事件时, 如果已置位对应的 LOCRE 位, 则 MCU 将生成系统复位。否则, MCU 将置位对应的 LOCS 位, 并且 MCG 将生成 LOCS 中断请求。如果已检测到 OSC 时钟丢失, PLL 的 LOCK 状态位会清零

26.5.5 MCG 固定频率时钟

MCG 固定频率时钟 (MCGFFCLK) 可作为其他片上外设的固定频率时钟源; 请查看结构框图。此时钟通过来自内部参考时钟发生器的慢速时钟或晶体振荡器的外部参考时钟驱动, 由 FLL 参考时钟分频器分频。MCGFFCLK 源由 C1[IREFS]选择。

此时钟与外设总线时钟同步，并且仅当其频率不超过 MCGOUTCLK 频率的 1/8 时才有效。当无效时，它将禁用并保持高电平。当 MCU 处于 BLPI 模式时，MCGFFCLK 不可用。此时钟在停止模式下也禁用。FLL 参考时钟的频率必须在 MCGFFCLK 的有效频率范围内。

26.5.6 MCG PLL 时钟

MCG PLL 时钟 (MCGPLLCLK) 为可用，具体取决于器件 MCG 模块。有关详细信息，请参见此 MCU 的时钟分布章节。在使能且置位 S[LOCK0] 之前，MCGPLLCLK 无法从 MCG 发出。

26.5.7 MCG 自动调整 (ATM)

MCG 自动调整 (ATM) 是一种 MCG 功能，使能后，它使用外部时钟作为参考自动调整 MCG 内部参考时钟。测试和使能 MCG IRC 时钟的选择由 ATC[ATMS] 控制位控制 (如果 ATC[ATMS]=0，则选择 32 kHz IRC，如果 ATC[ATMS]=1，则选择 4 MHz IRC)。如果 ATM 选择 4 MHz IRC，则将通过 128 分频将 4 MHz IRC 降频为 31.250 kHz 范围。

当通过向 ATC[ATME] 位写入 1 使能 MCG ATM 时，ATM 机将开始自动调整选定的 IRC 时钟。在自动调整过程中，ATC[ATME] 保持置位，并且将在 ATM 完成或发生中止之后解除置位。如果检测到对任何以下控制寄存器 (C1、C3、C4 或 ATC) 进行了写操作或者如果已进入停止模式，则会中止 MCG ATM。如果发生中止，则会置位 ATC[ATMF] 故障标志。

ATM 机将总线时钟用作外部参考时钟，以对 IRC 自动调整。因此，需要在某一时钟模式下配置 MCG，在此模式下，用于生成系统时钟的参考时钟为外部参考时钟，例如 FBE 时钟模式。MCG 不能配置成某一种时钟模式，在此模式下，选定的 IRC ATM 时钟用于生成系统时钟。总线时钟也需要在 8–16 MHz 频率范围运行。

为了在选定 IRC 上执行 ATM，ATM 机使用逐次逼近技术来调整 IRC 调整位，以生成所需的 IRC 调整频率。ATM SAR 每个 ATM IRC 调整位都以 MSB 开头。对于每个调整位测试，ATM 使用由 ATM 选定 IRC 时钟生成的脉冲，以使能用于计数 ATM 外部时钟数的计数器。每个调整位结束时，ATM 外部计数器值将与 ATCV[15:0] 寄存器值比较。接受测试的 ATM 调整位将根据比较结果被清零或保持置位。这将在 ATM SAR 机测试所有调整位后完成。

使能 ATM 之前，需要推导 ATM 预期计数并将其存储到 ATCV 寄存器中。ATCV 预期计数根据所需的目标内部参考时钟 (IRC) 频率和外部参考时钟频率得出，公式如下：

$$\text{ATCV ExpectedCount Value} = 21 * (\text{Fe} / \text{Fr})$$

- Fr = 目标内部参考时钟 (IRC) 调整频率
- Fe = 外部时钟频率

如果在 4 MHz IRC 上执行自动调整，则计算得出的预期计数值必须先乘以 128，然后再存储到 ATCV 寄存器中。因此，使用以下公式计算 ATCV 预期计数值，以调整 4 MHz IRC。

$$\text{ExpectedCount Value} = (\text{Fe}/\text{Fr}) * 21 * (128)$$

26.6 初始化/应用信息

本节介绍了如何在应用中初始化和配置 MCG 模块。

以下章节包括如何初始化 MCG 以及在各种可用模式之间正确切换的示例。

26.6.1 MCG 模块初始化序列

MCG 退出复位后处于 FEI 模式。

内部参考将在 FLL 获取锁定之前在 t_{irefst} 微秒内稳定。一旦内部参考稳定，FLL 将在 $t_{\text{fill_acquire}}$ 微秒内获取锁定。

26.6.1.1 初始化 MCG

MCG 复位后处于 FEI 模式，可直接切换的 MCG 模式为 FEE，FBE 和 FBI 模式 (参见图 26-17)。要想进入任何其他模式，需要先将 MCG 配置为这三种中间模式之一。务必检查 MCG 状态寄存器中的相关状态位是否反映每种模式中的所有配置更改。

要从 FEI 模式更改为 FEE 或 FBE 模式，请执行以下操作：

1. 通过设置 C2 寄存器中相应的位来使能外部时钟源。
2. 对 C1 寄存器执行写操作以选择时钟模式。
 - 如果进入 FEE 模式，则相应地设置 C1[FRDIV]，清除 C1[IREFS] 位以切换至外部参考，并让 C1[CLKS] 位保持为 2'b00，以便选择 FLL 输出作为系统时钟源。

- 如果进入 FBE 模式，则清除 C1[IREFS] 以切换至外部参考，并将 C1[CLKS] 位更改为 2'b10，以便选择外部参考时钟作为系统时钟源。此时，还必须根据外部参考频率相应地设置 C1[FRDIV] 位，从而使 FLL 参考时钟的频率保持在 31.25 kHz 至 39.0625 kHz 范围内。尽管 FLL 被旁通，但其仍处于 FBE 模式。
 - 可通过设置 C1[IRCLKEN] 选择性地保持内部参考运行。当应用在内部和外部模式之间来回切换时，这非常有用。为了最小化功耗，处于外部时钟模式时，请保持内部参考禁用。
3. 设置正确的配置位之后，等待 MCG 寄存器中受影响的位进行相应的更改，以反映 MCG 已转换至正确的模式。
- 如果 MCG 处于 FEE、FBE、PEE、PBE 或 BLPE 模式，且已在第一步中设置 C2[EREFS]，请等待 S[OSCINIT0] 位置位，以指示外部时钟源已完成其初始化周期并且已稳定。
 - 如果处于 FEE 模式，请检查以确保在转换之前已清除 S[IREFST]。
 - 如果处于 FBE 模式，请检查以确保已清除 S[IREFST] 并且 S[CLKST] 位已更改为 2'b10，指示已正确选择外部参考时钟。尽管 FLL 被旁通，但其仍处于 FBE 模式。
4. 对 C4 寄存器执行写操作，以确定 DCO 输出 (MCGFLLCLK) 频率范围。
- 默认情况下，如果 C4[DMX32] 已清零，则 DCO 输出的 FLL 乘数为 640。为了获得更大的灵活性，如果需要的是中低范围 FLL 乘数为 1280，请将 C4[DRST_DRS] 位设置为 2'b01，使 DCO 输出频率为 40 MHz。如果需要的是中高范围 FLL 乘数为 1920，请将 C4[DRST_DRS] 位设置为 2'b10，使 DCO 输出频率为 60 MHz。如果需要的是高范围 FLL 乘数为 2560，请将 C4[DRST_DRS] 位设置为 2'b11，使 DCO 输出频率为 80 MHz。
 - 使用 32.768 kHz 外部参考时，如果需要通过 32.768 kHz 参考实现最大低范围 DCO 频率，则将 C4[DRST_DRS] 位设置为 2'b00 并将 C4[DMX32] 位置 1。通过新的乘数 732 得出的 DCO 输出(MCGOUTCLK)频率为 24 MHz。
 - 使用 32.768 kHz 外部参考时，如果需要通过 32.768 kHz 参考实现最大中范围 DCO 频率，则将 C4[DRST_DRS] 位设置为 2'b01 并将 C4[DMX32] 位置 1。通过新的乘数 1464 得出的 DCO 输出(MCGOUTCLK)频率为 48 MHz。

- 使用 32.768 kHz 外部参考时，如果需要通过 32.768 kHz 参考实现最大高范围 DCO 频率，则将 C4[DRST_DRS]位设置为 2'b10 并将 C4[DMX32]位置 1。通过新的乘数 2197 得出的 DCO 输出(MCGOUTCLK)频率为 72 MHz。
 - 使用 32.768 kHz 外部参考时，如果需要通过 32.768 kHz 参考实现最大高范围 DCO 频率，则将 C4[DRST_DRS]位设置为 2'b11 并将 C4[DMX32]位置 1。通过新的乘数 2929 得出的 DCO 输出(MCGOUTCLK)频率为 96 MHz。
5. 等待 FLL 锁定时间，已确保 FLL 以新的 C4[DRST_DRS]和 C4[DMX32]设定的频率运行。

要从 FEI 时钟模式更改为 FBI 时钟模式，请执行以下操作：

1. 将 C1 寄存器中的 C1[CLKS] 位更改为 2'b01，从而选择内部参考时钟作为系统时钟源。
2. 等待 MCG 状态寄存器中的 S[CLKST] 位更改为 2'b01，表示已选择相应的内部参考时钟。
3. 对 C2 寄存器执行写操作，以确定 IRCS 输出 (IRCSCLK) 频率范围。
 - 默认情况下，如果 C2[IRCS] 清零，则 IRCS 选择的输出为慢速内部参考时钟 (32 kHz IRC)。如果需要的是更快的 IRC，通过 C2[IRCS] 置 1，IRCS 时钟源自 4MHz IRC 源。

26.6.2 使用 32.768 kHz 参考

在 FEE 和 FBE 模式下，如果使用 32.768 kHz 外部参考，当默认 FLL 乘数因子为 640 时，DCO 输出 (MCGFLLCLK) 频率为 20.97 MHz (低频)。

如果 C4[DRST_DRS] 位置位为 2'b01，则乘数因子翻倍至 1280，并且得出的 DCO 输出频率为 41.94 MHz (中低频)。如果 C4[DRST_DRS]位置位为 2'b10，则乘数因子变为 1920，并且得出的 DCO 输出频率为 62.91 MHz (中高频)。如果 C4[DRST_DRS] 位置位为 2'b11，则乘数因子变为 2560，并且得出的 DCO 输出频率为 83.89 MHz (高频)。

在 FBI 和 FEI 模式下，不建议置位 C4[DMX32] 位。如果将内部参考调整为高于 32.768 kHz 的频率，则更高的 FLL 乘数因子可能使微处理器系统时钟超出规格并且会损坏部件。

26.6.3 MCG 模式切换

在 MCG 运行模式之间切换时，必须更改某些配置位，以便从一种模式正确转换至另一种模式。

每次更改任意位 (C6[PLLS]、C1[IREFS]、C1[CLKS]、C2[IRCS]或 C2[EREFS])时，必须先检查 MCG 状态寄存器中对应的位 (PLLST、IREFST、CLKST、IRCST 或 OSCINIT)，然后再转至应用软件。

此外，务必确保根据要切换至的模式正确置位参考时钟分频器 (C1[FRDIV] 和 C5[PRDIV0])。例如，在 PEE 模式中，如果使用一个 4 MHz 晶体，则 C5[PRDIV0] 必须被置位为 5'b000 (除 1) 或 5'b001 (除 2)，以将外部参考分频至所需的频率 (介于 2 和 4 MHz 之间)。

在 FBE、FEE、FBI 和 FEI 模式下，应用可随时通过 C4[DRST_DRS]位将 FLL 乘数因子切在 640, 1280, 1920 和 2560 之间切换。如果 C2[LP]=1，则会忽略对 C4[DRST_DRS]位的写入操作。

下表介绍了在每种时钟模式使用 C1[FRDIV]、C5[PRDIV0] 和 C6[VDIV0] 设置时如何计算 MCGOUTCLK 频率。

表 26-21. MCGOUTCLK 频率计算选项

时钟模式	$f_{\text{MCGOUTCLK}}^1$	注释
FEI (FLL 内部启用)	$f_{\text{int}} \times F$	刚复位之后，典型 $f_{\text{MCGOUTCLK}} = 21$ MHz。
FEE (FLL 外部启用)	$(f_{\text{ext}} / \text{FLL_R}) \times F$	$f_{\text{ext}} / \text{FLL_R}$ 必须在 31.25 kHz 至 39.0625 kHz 范围内。
FBE (FLL 外部旁路)	OSCCLK	OSCCLK / FLL_R 必须在 31.25 kHz 至 39.0625 kHz 范围内。
FBI (FLL 内部旁路)	MCGIRCLK	可在慢速和快速 IRC 之间选择。
PEE (PLL 外部启用)	$(\text{OSCCLK} / \text{PLL_R}) \times M / 2$	OSCCLK / PLL_R 必须在 2 – 4 MHz 范围内。
PBE (PLL 外部旁路)	OSCCLK	OSCCLK / PLL_R 必须在 2 – 4 MHz 范围内。
BLPI (内部旁通低功耗)	MCGIRCLK	可在慢速和快速 IRC 之间选择。
BLPE (外部旁通低功耗)	OSCCLK	

1. FLL_R 是 C1[FRDIV] 位选择的参考分频器，F 是 C4[DRST_DRS]和 C4[DMX32] 位选择的 FLL 因数，PLL_R 是 C5[PRDIV0] 位选择的参考分频器，M 是 C6[VDIV0] 位选择的乘数。

本节介绍了使用 4 MHz 外部晶振的几种模式切换示例。如果使用频率低于 2 MHz，则 MCG 不得配置用于任意 PLL 模式 (PEE 和 PBE)。

26.6.3.1 示例 1: 从 FEI 转换至 PEE 模式: 外部晶体 = 4 MHz, MCGOUTCLK 频率 = 48 MHz

在此示例中, MCG 从 FEI 正确转换至 PEE 操作模式, 以从 4 MHz 外部晶体参考实现 48 MHz MCGOUTCLK 频率。首先将介绍代码顺序。然后将通过流程图演示该顺序。

1. 首先, FEI 必须转换至 FBE 模式:

a. $C2 = 0x2C$

- $C2[RANGE]$ 设置为 $2'b01$, 因为 4MHz 的频率在高频范围内。
- $C2[HGO]$ 置 1, 以针对高增益操作配置晶体振荡器。
- $C2[EREFS]$ 置 1, 因为使用的是晶体。

b. $C1 = 0x90$

- $C1[CLKS]$ 设置为 $2'b10$, 以便将外部参考时钟选择系统时钟源
- $C1[FRDIV]$ 设置为 $3'b010$, 或除以 128, 因为 $4\text{ MHz} / 128 = 31.25\text{ kHz}$, 这在 FLL 所需的范围 31.25 kHz 至 39.0625 kHz 范围内
- $C1[IREFS]$ 清零, 以便选择外部参考时钟并使能外部振荡器。

c. 循环直到 $S[OSCINIT0]$ 为 1, 表示已初始化 $C2[EREFS0]$ 选择的晶体。

d. 循环直到 $S[IREFST]$ 为 0, 表示外部参考为参考时钟的当前来源。

e. 循环直到 $S[CLKST]$ 为 $2'b10$, 表示选择外部参考时钟作为 MCGOUTCLK。

2. 然后配置 $C5[PRDIV0]$, 以生成正确的 PLL 参考频率。

a. $C5 = 0x01$

- $C5[PRDIV]$ 设置为 $5'b00001$, 或除以 2, 从而使 PLL 参考频率为 $4\text{MHz} / 2 = 2\text{ MHz}$ 。

3. 然后, FBE 必须先直接转换至 PBE 模式, 或先转换为 BLPE 模式然后再转换为 PBE 模式:

a. BLPE: 如果需要通过 BLPE 模式转换, 则先将 $C2[LP]$ 置 1。

b. BLPE/PBE: $C6 = 0x40$

- $C6[PLLS]$ 置 1, 选择 PLL。此时, 当 $C1[PRDIV]$ 值为 $2'b001$, PLL 参考分频器为 2 (请参见 PLL 外部参考分频因数表), 从而使参考频率 $4\text{MHz} / 2 = 2\text{ MHz}$ 。在 BLPE 模式下, 更改 $C6[PLLS]$ 位仅决定 PBE 模式下 MCG 使用 PLL。

C6[VDIV]设置为 5'b00000，或乘以 24，因为 2MHz 参考 * 24 = 48 MHz。在 BLPE 模式下，配置 VDIV 位并不会造成影响，因为 PLL 已禁用。更改它们仅会设定 PLL 在 PBE 模式下使用的乘数值。

- c. BLPE: 如果从 BLPE 模式转换，则将 C2[LP]清零以切换至 PBE 模式。
- d. PBE: 循环直到置位 S[PLLST]，表示 PLLS 当前的时钟源为 PLL。
- e. PBE: 然后循环直到置位 S[LOCK0]，表示 PLL 已获取锁定。

4. 最后，PBE 模式转换至 PEE 模式：

- a. C1 = 0x10
 - C1[CLKS]设置为 2'b00，以便将 PLL 输出选作系统时钟源。
- b. 循环直到 S[CLKST]为 2'b11，表示在当前时钟模式下选择 PLL 输出作为 MCGOUTCLK。
 - 现在，当 PRDIV 为除以 2，且 C6[VDIV]为乘以 24 时，MCGOUTCLK = $[(4 \text{ MHz} / 2) * 24] = 48 \text{ MHz}$ 。

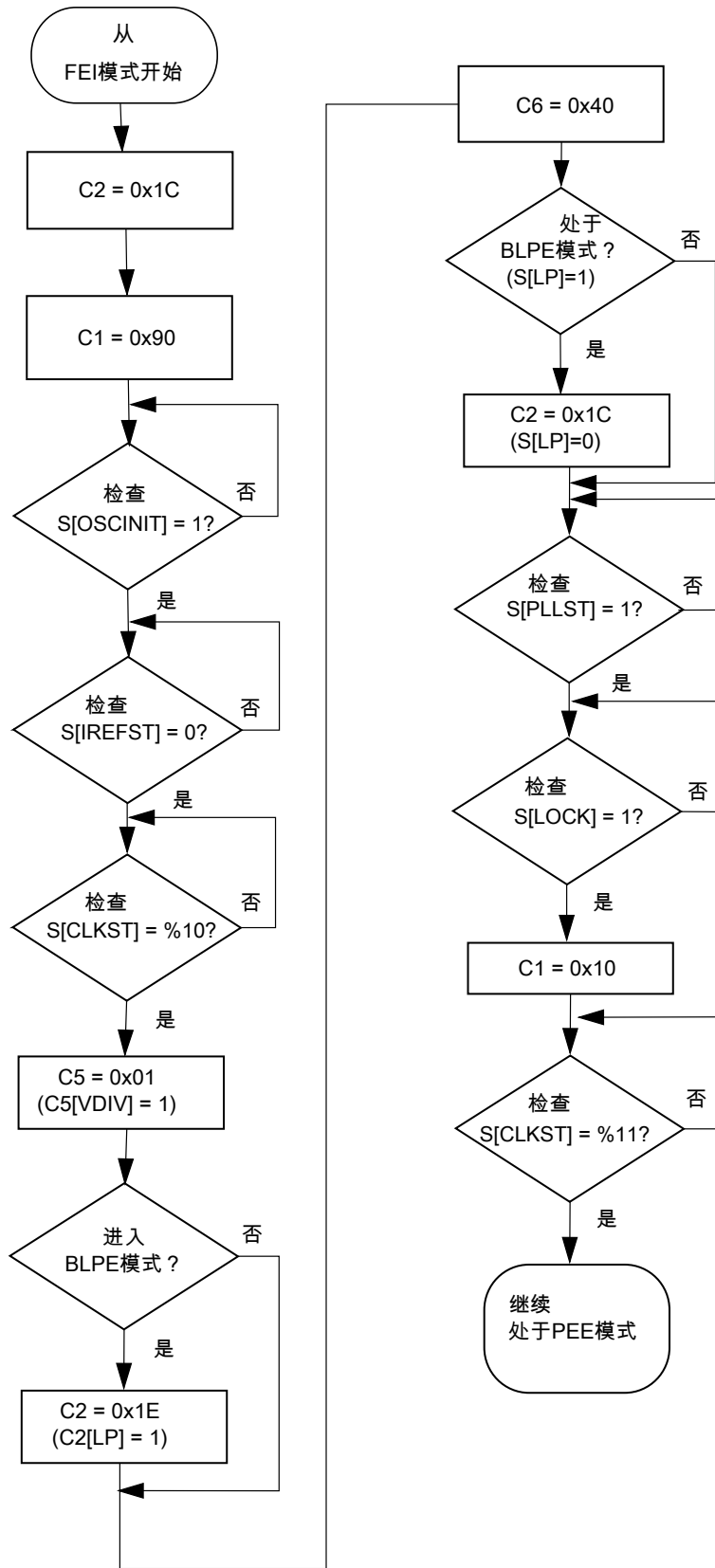


图 26-18. 使用 4 MHz 晶体从 FEI 至 PEE 模式转换的流程图

26.6.3.2 示例 2: 从 PEE 移至 BLPI 模式: MCGOUTCLK 频率 =32 kHz

在此示例中, MCG 将经历从 PEE 模式至 BLPI 模式的正确运行模式, PEE 模式使用 4 MHz 晶体生成 48 MHz MCGOUTCLK 频率 (请参见上一示例), BLPI 模式使用 32 kHz MCGOUTCLK 频率。首先将介绍代码顺序。然后将通过流程图演示该顺序。

1. 首先, PEE 必须转换至 PBE 模式:
 - a. $C1 = 0x90$
 - $C1[CLKS]$ 设置为 $2'b10$, 以便将系统时钟源切换为外部参考时钟。
 - b. 循环至 $S[CLKST]$ 为 $2'b10$, 表示选择外部参考时钟作为 MCGOUTCLK。
2. 然后, PBE 必须先直接转换至 FBE 模式, 或先转换为 BLPE 模式然后再转换为 FBE 模式:
 - a. BLPE: 如果需要通过 BLPE 模式转换, 则先将 $C2[LP]$ 置 1。
 - b. BLPE/FBE: $C6 = 0x00$
 - $C6[PLLS]$ 清零, 以选择 FLL。此时, 如果 $C1[FRDIV]$ 的值为 $3'b010$, 则将 FLL 分频器设为 128, 从而使参考频率 $4\text{ MHz} / 128 = 31.25\text{ kHz}$ 。如果先前未将 $C1[FRDIV]$ 设置为 $3'b010$ (当外部时钟源频率为 4 MHz 时, 实现所需的 31.25–39.06 kHz FLL 参考频率的必要条件), 则在将 $C6[PLLS]$ 位清零之前必须进行更改。在 BLPE 模式下, 更改此位将仅决定 MCG 在 FBE 模式下使用 FLL。当 $C6[PLLS] = 0$ 时, $C6[VDIV]$ 值不起作用。
 - c. BLPE: 如果从 BLPE 模式转换, 则将 $C2[LP]$ 清零以切换至 FBE 模式。
 - d. FBE: 循环至清除 $S[PLLST]$, 表示 PLLS 当前的时钟源为 FLL。
3. 接着, FBE 模式转换至 FBI 模式:
 - a. $C1 = 0x54$
 - $C1[CLKS]$ 设置为 $2'b01$, 以便将系统时钟切换为内部参考时钟。
 - $C1[IREFS]$ 置 1, 以便将内部参考时钟选择为参考时钟源。
 - $C1[FRDIV]$ 保持不变, 因为参考分频器对内部参考没有影响。
 - b. 循环至 $S[IREFST]$ 为 1, 表示已将内部参考时钟选作参考时钟源。
 - c. 循环至 $S[CLKST]$ 为 $2'b01$, 表示选择内部参考时钟作为 MCGOUTCLK。
4. 最后, FBI 转换至 BLPI 模式。

a. C2 = 0x02

- C2[LP]为 1
- 当 C1[IREFS]位置位时，C2[RANGE]、C2[HGO]、C2[EREFS]、C1[IRCLKEN]和 C1[IREFSTEN]位将被忽略。此时，它们可保持置位或被清零。

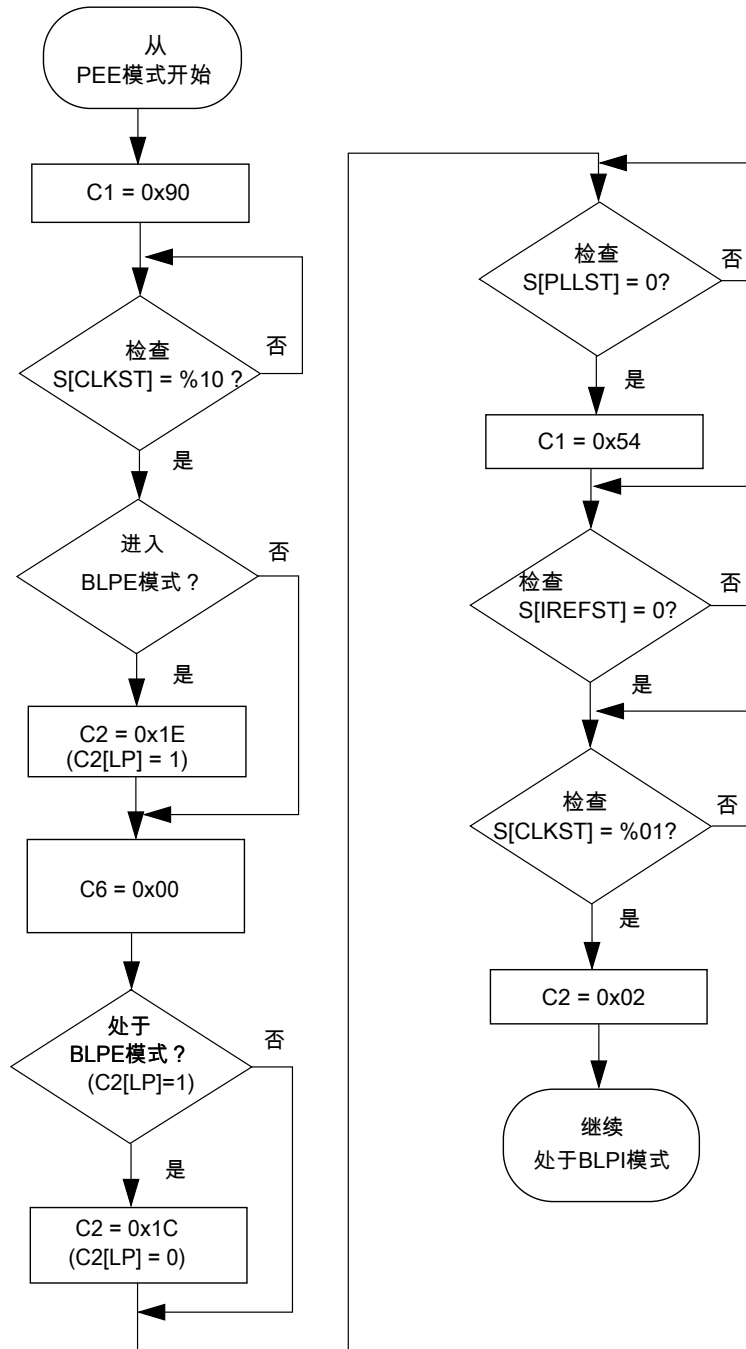


图 26-19. 使用 4 MHz 晶体从 PEE 至 BLPI 模式转换的流程图

26.6.3.3 示例 3: 从 BLPI 转移至 FEE 模式

在此示例中，MCG 将经历从 BLPI 模式至 FEE 模式的正确运行模式，BLPI 模式使用运行内部参考时钟（请参见上一示例）提供 32 kHz MCGOUTCLK 频率，FEE 模式使用 4 MHz 晶体生成 20 MHz MCGOUTCLK 频率。首先将介绍代码顺序。然后将通过流程图演示该顺序。

1. 首先，BLPI 必须转换至 FBI 模式。
 - a. $C2 = 0x00$
 - $C2[LP]$ 为 0
2. 接着，FBI 将转换至 FEE 模式。
 - a. $C2 = 0x1C$
 - $C2[RANGE]$ 设置为 2'b01，因为 4 MHz 的频率在高频范围内。
 - $C2[HGO]$ 置 1，以针对高增益操作配置晶体振荡器。
 - $C2[EREFS]$ 置 1，因为使用的是晶振。
 - b. $C1 = 0x10$
 - $C1[CLKS]$ 设置为 2'b00，以便将 FLL 输出选作系统时钟源。
 - $C1[FRDIV]$ 保持在 3'b010 或除以 128，以实现 4 MHz / 128 的参考 = 31.25 kHz。
 - $C1[IREFS]$ 清零，以便选择外部参考时钟。
 - c. 循环至 $S[OSCINIT]$ 为 1，表示 $C2[EREFS]$ 位选择的晶振已初始化。
 - d. 循环至 $S[IREFST]$ 为 0，表示外部参考时钟为参考时钟当前的来源。
 - e. 循环至 $S[CLKST]$ 为 2'b00，表示选择 FLL 输出作为 MCGOUTCLK。
 - f. 现在，在参考频率为 31.25 kHz 时，固定 DCO 乘法器为 640， $MCGOUTCLK = 31.25 \text{ kHz} * 640 / 1 = 20 \text{ MHz}$ 。
 - g. 此时，默认情况下， $C4[DRST_DRS]$ 位设置为 2'b00 且 $C4[DMX32]$ 清零。如果 MCGOUTCLK 所需的频率为 40 MHz，则将 $C4[DRST_DRS]$ 位置位为 0x01，以便将 FLL 乘数因子从 640 切换为 1280。要将 MCGOUTCLK 频率返回至 20 MHz，请再次将 $C4[DRST_DRS]$ 位设置为 2'b00，并且 FLL 乘数因子将切换回 640。

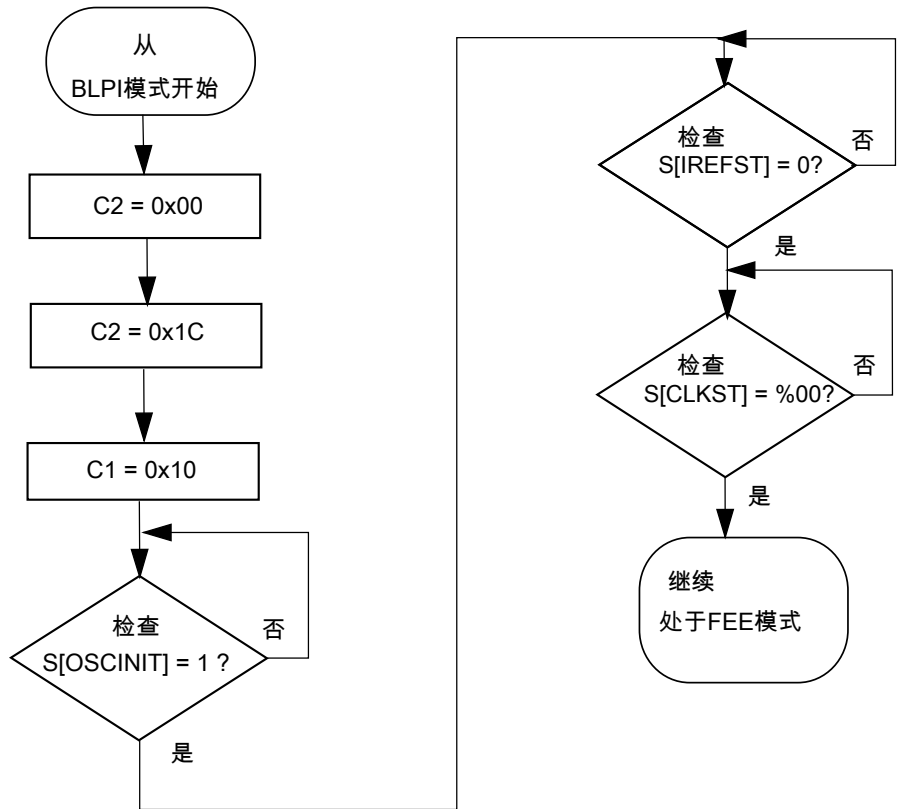


图 26-20. 使用 4 MHz 晶体从 BLPI 至 FEE 模式的流程图

第 27 章 振荡器 (OSC)

27.1 此模块的芯片实现细节

27.1.1 OSC 在 MCG 下的操作模式

MCG 的 C2 寄存器位配置振荡器频率范围。详情请参见 OSC 和 MCG 章节。

27.2 简介

OSC 模块是一个晶体振荡器。该模块与外部晶体或谐振器一同生成 MCU 的参考时钟。

27.3 特性和模式

模块的主要特性如下。

- 支持 32 kHz 晶振 (低范围模式)
- 支持 3–8 MHz、8–32 MHz 晶振和谐振器 (高范围模式)
- 自动增益控制 (AGC)，可优化高频率范围 (3–8 MHz、8–32 MHz) 低功耗模式下的功耗
- 以下频率范围内可配置高增益：32 kHz、3–8 MHz 和 8–32 MHz
- 电压和频率滤波功能，确保时钟频率和稳定性
- 可选择直接从 EXTAL 信号输入旁路时钟

- 一个时钟用于 MCU 时钟系统
- 两个时钟用于可工作于停止模式的片上外设

功能说明 更加详细地描述了模块的工作情况。

27.4 结构框图

OSC 模块用一个晶振或谐振器来生成三个经滤波的振荡器时钟信号。从 OSC 模块输出三个时钟：用于 MCU 系统的 OSCCLK、用于片上外设的 OSCERCLK 和 OSC32KCLK。OSCCLK 只能工作于运行模式。OSCERCLK 和 OSC32KCLK 可以工作于低功耗模式。有关时钟源分配的信息，请参见本 MCU 的时钟分布信息。

有关本 MCU 中外部参考时钟源的详情，请参见芯片配置信息。

本图展示了 OSC 模块的结构框图。

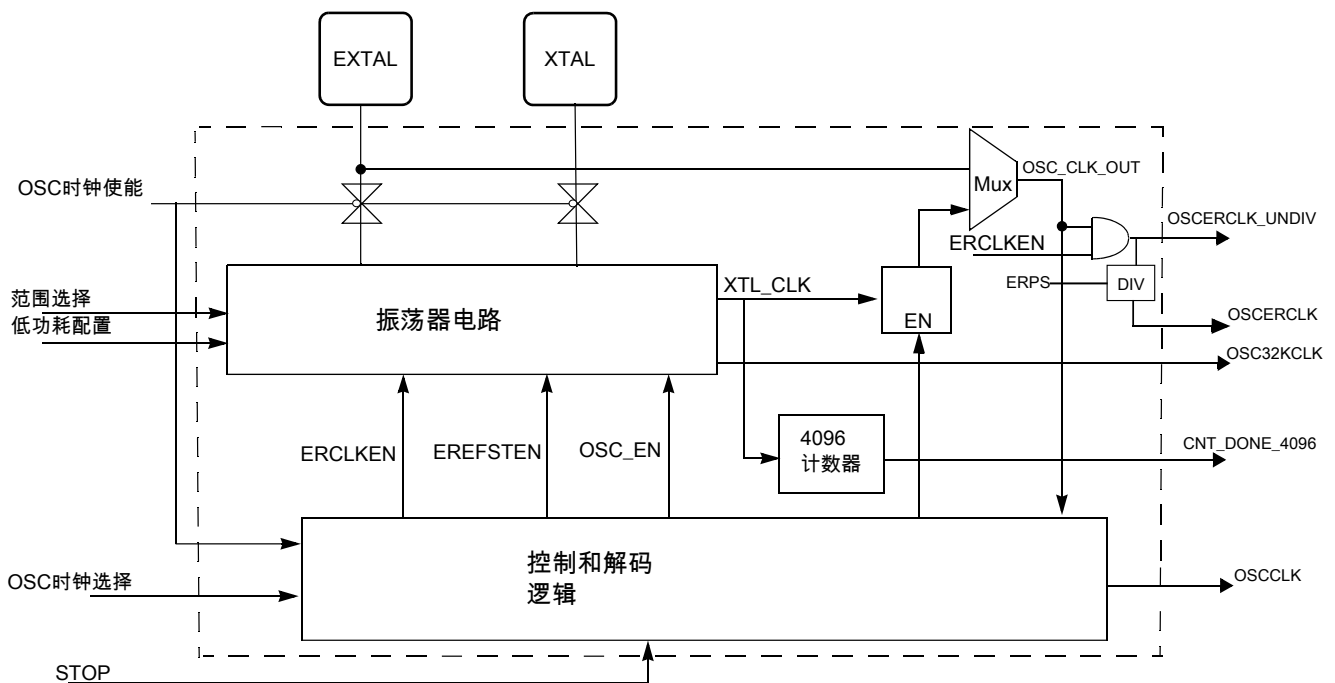


图 27-1. OSC 模块结构框图

27.5 OSC 信号说明

此表介绍用户可访问的 OSC 模块信号。

详情请参见此 MCU 的信号复用信息。

表 27-1. OSC 信号说明

信号	说明	I/O
EXTAL	外部时钟/振荡器输入	I
XTAL	振荡器输出	O

27.6 外部晶振/谐振器连接

晶振/谐振器频率基准的连接如这里的示意图所示。

在使用低频、低功耗模式时，唯一的外部组件就是晶振或陶瓷谐振器本身。在其他振荡器模式下，需要使用负载电容器(C_x , C_y)和反馈电阻器 (R_F)。下表展示了所有可能的连接。

表 27-2. 外部晶振/谐振器连接

振荡器模式	连接
低频 (32 kHz)、低功耗	连接 1
低频 (32 kHz)、高增益	连接 2/连接 3
高频 (3~32 MHz)、低功耗	连接 1/连接 3 ^{2,2}
高频 (3~32 MHz)、高增益	连接 2/连接 3 ²

1. 当负载电容 (C_x , C_y) 大于 30 pF 时，使用连接 3。
2. 在低功耗模式下，振荡器有内部反馈电阻 R_F 。因此，低功耗模式下要去掉连接 3 中的外部 R_F 。

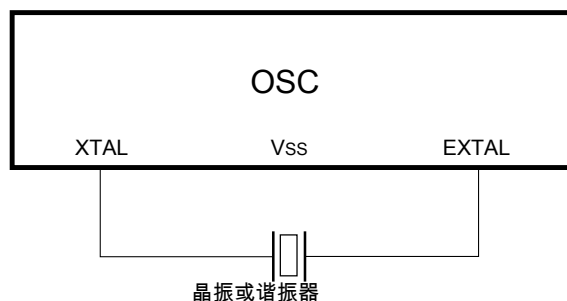


图 27-2. 晶振/陶瓷谐振器连接 - 连接 1

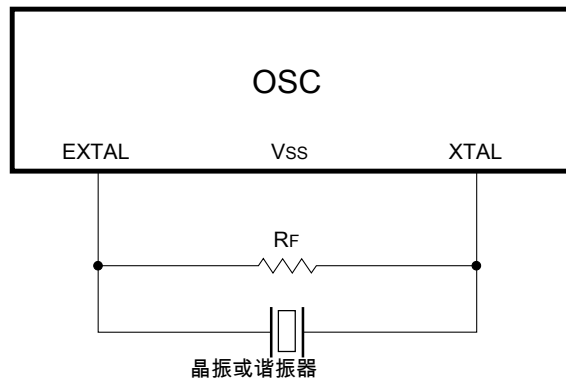


图 27-3. 晶振/陶瓷谐振器连接 - 连接 2

注

连接 1 和连接 2 应通过配置 CR[SCxP]位，用内部电容作为振荡器负载。

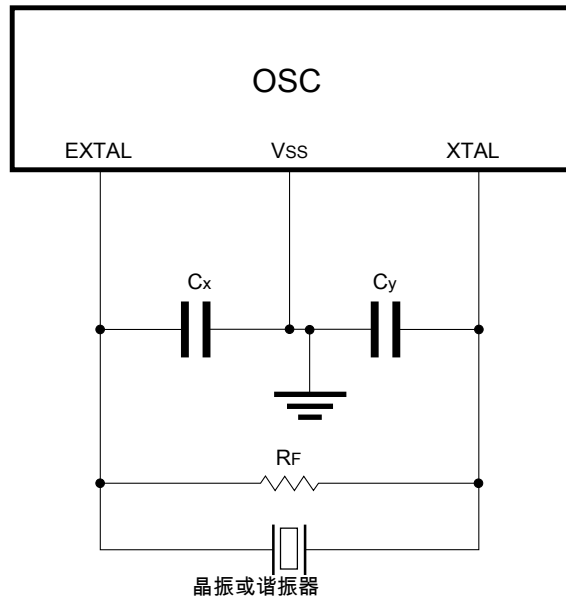


图 27-4. 晶振/陶瓷谐振器连接 - 连接 3

27.7 外部时钟连接

在外部时钟模式下，引脚可如下图所示连接。

注

当将 XTAL 引脚配置为 GPIO 功能时，可将 XTAL 用作 GPIO。

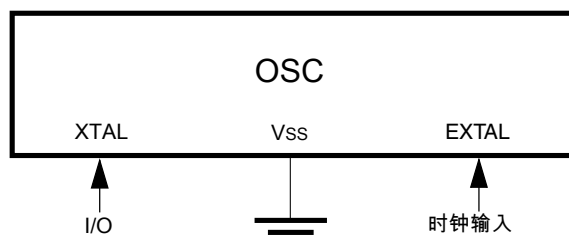


图 27-5. 外部时钟连接

27.8 存储器映射/寄存器定义

有些振荡器模块寄存器位一般会集成到其他外设中，比如 MCG 或 SIM 等。

27.8.1 OSC 存储器映射/寄存器定义

OSC 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_5000	OSC 控制寄存器 (OSC_CR)	8	R/W	00h	27.8.1.1/505
4006_5002	OSC_DIV (OSC_OSC_DIV)	8	R/W	00h	27.8.1.2/506

27.8.1.1 OSC 控制寄存器 (OSC_CR)

注

OSC 使能并开始生成时钟后,低功耗和频率范围等配置不得更改。

地址: 4006_5000h 基准 + 0h 偏移 = 4006_5000h

位	7	6	5	4	3	2	1	0
读写	ERCLKEN	0	EREFSTEN	0	SC2P	SC4P	SC8P	SC16P
复位	0	0	0	0	0	0	0	0

OSC_CR 字段描述

字段	描述
7 ERCLKEN	外部参考使能 使能外部参考时钟(OSCERCLK)。

下一页继续介绍此表...

OSC_CR 字段描述 (继续)

字段	描述
	0 外部参考时钟无效。 1 外部参考时钟使能。
6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 EREFSTEN	外部参考停止使能 控制外部参考时钟(OSCERCLK)在 MCG 进入停止模式时是否保持使能。 0 外部参考时钟在停止模式下停用。 1 如果 ERCLKEN 在进入停止模式之前置位，则外部参考时钟在停止模式下保持使能。
4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 SC2P	振荡器 2pF 电容负载配置 配置振荡器负载。 0 停用选择。 1 向振荡器负载添加 2 pF 电容。
2 SC4P	振荡器 4pF 电容负载配置 配置振荡器负载。 0 停用选择。 1 向振荡器负载添加 4 pF 电容。
1 SC8P	振荡器 8pF 电容负载配置 配置振荡器负载。 0 停用选择。 1 向振荡器负载添加 8 pF 电容。
0 SC16P	振荡器 16pF 电容负载配置 配置振荡器负载。 0 停用选择。 1 向振荡器负载添加 16 pF 电容。

27.8.1.2 OSC_DIV (OSC_OSC_DIV)

OSC 时钟分频器寄存器。

地址: 4006_5000h 基准 + 2h 偏移 = 4006_5002h

位	7	6	5	4	3	2	1	0
读	ERPS		0	0	0	0	0	0
写								
复位	0	0	0	0	0	0	0	0

OSC_OSC_DIV 字段描述

字段	描述
7-6 ERPS	ERCLK 预分频器。这两位用于配置 ERCLK 的分频比。未分频的 ERCLK 输出不受这两位影响。 00 分频比为 1。 01 分频比为 2。 10 分频比为 4。 11 分频比为 8。
5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
0 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

27.9 功能说明

模块的详细功能可在此找到。

27.9.1 OSC 模块状态

OSC 模块的状态转换如下图所示。本节具体说明各状态以及状态间的相互转换。

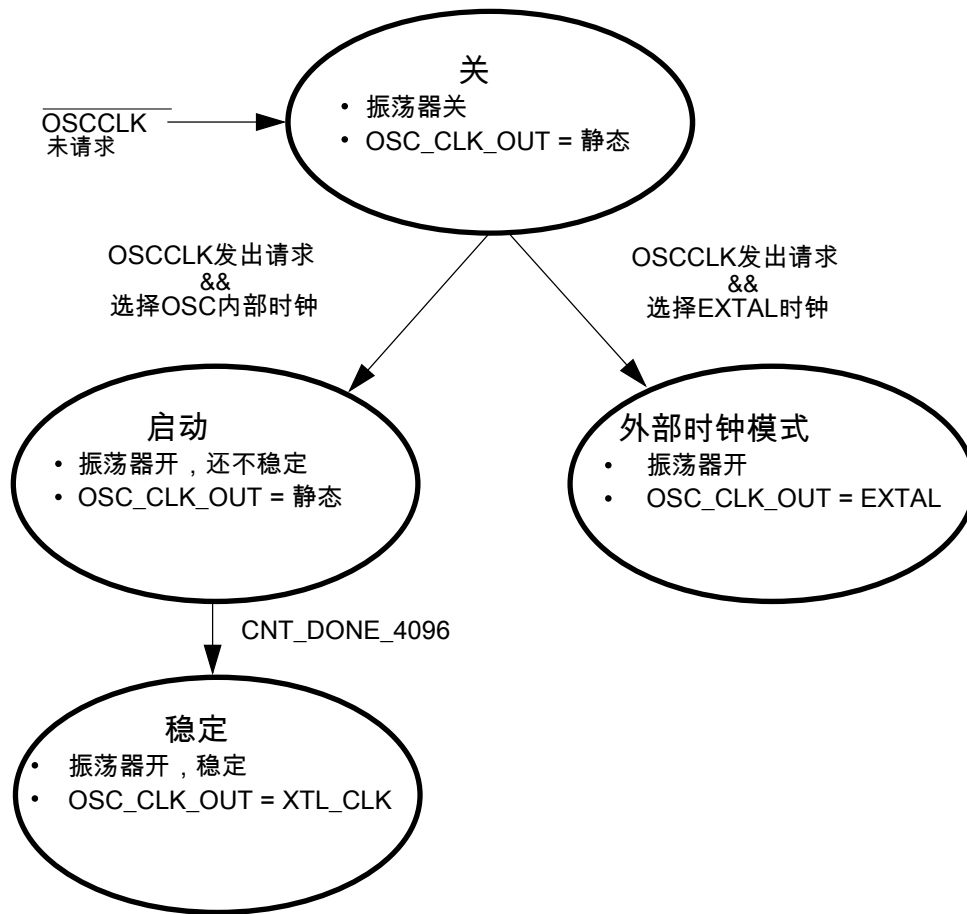


图 27-8. OSC 模块状态图

注

XTL_CLK 为从 OSC 电路内部生成的时钟。

27.9.1.1 关

OSC 在系统不需要 OSC 时钟时进入关闭状态。进入此状态后，除非已通过清除外部参考时钟选择位将 OSC 配置为 EXTAL，否则 XTL_CLK 为静态。有关本 MCU 中外部参考时钟源的详情，请参阅芯片配置信息。在此状态下，EXTAL 和 XTAL 引脚还会与其他所有振荡器电路解除耦合。在此模式下，OSC 模块消耗少许电流。

27.9.1.2 振荡器启动

通过置位外部参考时钟选择位来选择使用内部振荡器电路, 将 OSC 配置成生成时钟 (OSC_EN 拉高), OSC 进入启动状态。在此状态下, OSC 模块使能, 振荡开始, 但是还不稳定。当振荡幅度达到可以通过输入缓冲区时, XTL_CLK 开始为计数器提供时钟源。当计数器达到 4096 个 XTL_CLK 周期时, 振荡器可视为稳定, XTL_CLK 被传送到输出时钟 OSC_CLK_OUT。

27.9.1.3 振荡器稳定

振荡器稳定后, OSC 模块在 OSC_CLK_OUT 上产生稳定的输出时钟。其频率由所使用的外部组件确定。

27.9.1.4 外部时钟模式

OSC 在使能且外部参考时钟选择位清零时进入外部时钟状态。有关本 MCU 中外部参考时钟源的详情, 请参阅芯片配置信息。此状态下, OSC 模块设置为将来自 EXTAL 的时钟缓冲 (有迟滞) 至 OSC_CLK_OUT。其频率由提供的外部时钟确定。

27.9.2 OSC 模块模式

此 OSC 是 Pierce 类型的振荡器, 支持外部晶振或谐振器以表 27-6 中显示的频率范围运行。这些模式假定以下条件成立: OSC 使能以生成时钟 (OSC_EN=1), 配置为从内部生成时钟 (MCG_C2[EREFS] = 1), 一些或一个其他外设 (MCG、定时器等) 配置为用振荡器输出时钟 (OSC_CLK_OUT)。

表 27-6. 振荡器模式

模式	频率范围
低频、高增益	f_{osc_lo} (32.768 kHz) 到 f_{osc_lo} (39.0625 kHz)
高频模式 1、高增益	$f_{osc_hi_1}$ (3 MHz) 到 $f_{osc_hi_1}$ (8 MHz)
高频模式 1、低功耗	
高频模式 2、高增益	$f_{osc_hi_2}$ (8 MHz) 到 $f_{osc_hi_2}$ (32 MHz)
高频模式 2、低功耗	

注

有关本芯片所用低功耗工作模式及其与某些 OSC 模式的匹配情况, 详见芯片电源管理信息。

27.9.2.1 低频、高增益模式

在低频、高增益模式下，振荡器使用简单的逆变器类放大器。设置增益以实现轨到轨的振幅。

此模式下的振荡器输入缓冲区是单端缓冲区。它可提供低通滤波功能以及电压过滤迟滞，并将输出转换为逻辑电平。在此模式下，可使用内部电容。

27.9.2.2 低频、低功耗模式

在低频、低功耗模式下，振荡器使用增益控制环路来最大程度地降低功耗。当振幅增大时，放大器电流减小。依此持续下去，直到所需的振幅达到稳定状态。此模式可提供低通滤波功能以及电压过滤迟滞，并将输出转换为逻辑电平。在此模式下，可使用内部电容，并连接了内部反馈电阻，不能使用外部电阻。

在此模式下，放大器输入、增益控制输入以及输入缓冲区的输入都是通过电容耦合的方式，以承受漏电流（对 EXTAL 的直流电平不敏感）。

另外，此模式下的所有外部组件（除了谐振器本身）都集成在一起，包括负载电容器以及偏置 EXTAL 的反馈电阻器。

27.9.2.3 高频、高增益模式

在高频、高增益模式下，振荡器使用简单的逆变器类放大器。设置增益以实现轨到轨的振幅。此模式可提供低通滤波功能以及电压过滤迟滞，并将输出转换为逻辑电平。在此模式下，可使用内部电容。

27.9.2.4 高频、低功耗模式

在高频、低功耗模式下，振荡器使用增益控制环路来最大程度地降低功耗。当振幅增大时，放大器电流减小。依此持续下去，直到所需的振幅达到稳定状态。在此模式下，可使用内部电容，并连接了内部反馈电阻，不能使用外部电阻。

此模式下的振荡器输入缓冲区是差分缓冲区。它可提供低通滤波功能以及电压过滤迟滞，并将输出转换为逻辑电平。

27.9.3 计数器

振荡器输出时钟(OSC_CLK_OUT)将被门控关闭,直到计数器检测到 4096 个输入时钟周期(XTL_CLK)。在 4096 个周期完成后,计数器会将 XTL_CLK 传递到 OSC_CLK_OUT 上。此计数定时溢出可用于确保输出时钟的稳定性。

27.9.4 参考时钟的引脚要求

在振荡器模式下,OSC 模块需要同时使用 EXTAL 和 XTAL 引脚来生成输出时钟,但在外部时钟模式下,只需要使用 EXTAL 引脚。EXTAL 和 XTAL 引脚也可用作 I/O。关于这些引脚在此器件上的实现方式,请参见《信号复用和信号说明》一章。

27.10 复位

不存在与 OSC 模块相关联的复位状态。当 OSC 未配置为生成时钟时,计数器处于复位状态。

没有针对 OSC 模块的复位请求源。

27.11 低功耗模式下的操作

当 MCU 进入停止模式时,OSC 的工作状态取决于 CR[ERCLKEN]和 CR[EREFSETN]位设置。如果这两位均置位,则 OSC 处于工作状态。

在低漏电停止(LLS)模式下,OSC 保存所有寄存器设置。如果 CR[ERCLKEN]和 CR[EREFSTEN]均在进入低漏电停止模式之前置位,则在这些模式下 OSC 仍然工作。从超低漏电停止(VLLSx)模式唤醒之后,所有 OSC 寄存器位均复位,并需要通过软件进行初始化。

27.12 中断

OSC 模块不生成任何中断。

第 28 章

RTC 振荡器(OSC32K)

28.1 简介

RTC 振荡器模块为 RTC 提供时钟源。RTC 振荡器模块与外部晶体一同生成 RTC 参考时钟。

28.1.1 特性和模式

RTC 振荡器的主要特性如下：

- 支持 32 kHz 晶振（极低功耗）
- 包含内部反馈电阻
- 包含内部可编程电容器，可作为振荡器的 C_{load}
- 自动增益控制 (AGC)，以优化功耗

RTC 振荡器操作的详细描述请见 [功能说明](#)。

28.1.2 结构框图

以下是 RTC 振荡器的框图。

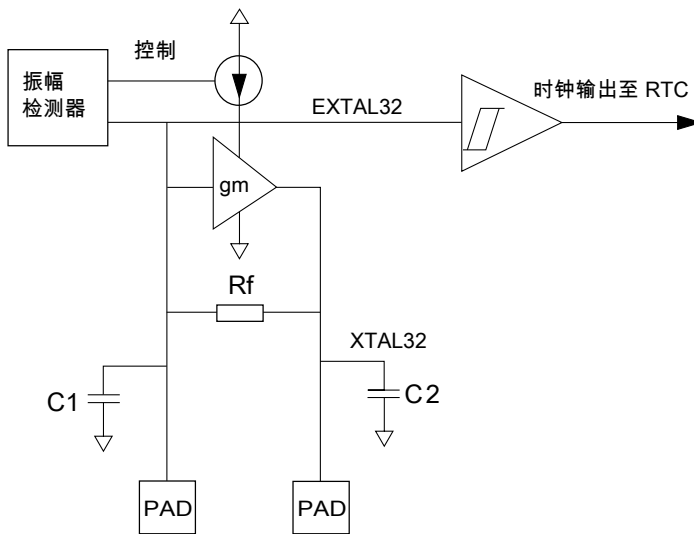


图 28-1. RTC 振荡器框图

28.2 RTC 信号说明

下表介绍用户可访问的 RTC 振荡器信号。参见芯片规格来了解哪些信号连接到了外部引脚上。

表 28-1. RTC 信号说明

信号	说明	I/O
EXTAL32	振荡器输入	I
XTAL32	振荡器输出	O

28.2.1 EXTAL32 — 振荡器输入

此信号是 RTC 振荡器的模拟输入。

28.2.2 XTAL32 — 振荡器输出

此信号是 RTC 振荡器模块的模拟输出。

28.3 外部晶振连接

下图显示了与晶振的连接。不需要外部负载电容器和反馈电阻。

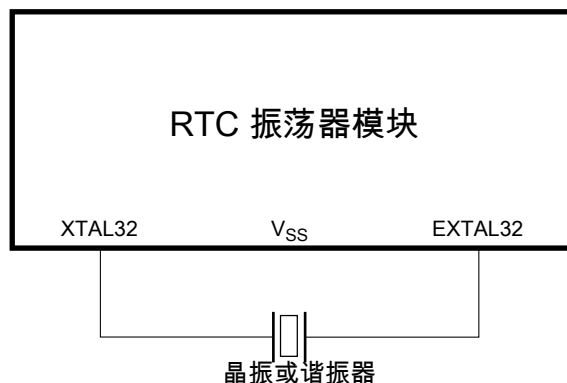


图 28-2. 晶振连接

28.4 存储器映射/寄存器说明

RTC 振荡器控制位是 RTC 寄存器的组成部分。有关详情，请参见 RTC 控制寄存器 (RTC_CR) 或具体芯片章节中的 RTC_GP_DATA_REG。

28.5 功能说明

如图 28-1 所示，该模块包含一个放大器，用于为 RTC 振荡器提供所需的负电阻。放大器增益由振幅检测器控制，这样做可优化功耗。采用一个施密特触发器来将振荡器生成的正弦波转换为脉冲时钟输出，它是 RTC 数字内核的参考时钟。

该振荡器包含一个约为 100 MΩ 的内部反馈电阻（位于 EXTAL32 与 XTAL32 之间）。

此外，此振荡器还有两个可编程电容器，可用作振荡器的 C_{load} 。可编程范围为 0pF 至 30pF。

28.6 复位概述

没有与 RTC 振荡器关联的复位状态。

28.7 中断

RTC 振荡器不会生成任何中断。

第 29 章

Flash 存储器控制器(FMC)

29.1 简介

Flash 控制器 (FMC) 是一种内存加速单元，可提供以下项目：

- 器件与非易失性存储器之间的接口。
- 可加速 Flash 存储器传输的缓存区。

29.1.1 概述

Flash 存储器控制器管理着器件与 flash 存储器之间的接口。FMC 接收详细配置存储器的状态信息，并利用此信息确保接口合适。下表显示了受支持的读/写操作。

Flash 存储器类型	读取	写入
程序 flash 存储器	8 位、16 位和 32 位读取	—

此外，对于存储阵列 (Bank) 0 和存储阵列 (Bank) 1，FMC 还提供三套独立的机制，用于加速器件与 flash 存储器之间的接口。64 位预测缓冲器可预取下一个 64 位 flash 存储器位置，而 4 路 8 组高速缓存和一个单入口 64 位缓存可存储之前访问过的 flash 存储器数据，以便快速访问。

29.1.2 特性

FMC 的功能包括：

- 器件与 flash 存储器之间的接口：
 - 对程序 flash 存储器进行 8 位、16 位及 32 位读写操作。

- 对于存储阵列 (Bank) 0 和存储阵列 (Bank) 1: 读取访问存储器中的连续 32 位空间会同时返回第二个读取数据, 且无等待状态。也就是说, 存储器将通过 32 位总线访问返回 64 位的数据。
- 针对每个 Crossbar 模块的主设备设置不可访问、只读访问、只写访问或的读/写访问提供交叉主机访问保护。
- 对于存储阵列 (Bank) 0 和存储阵列 (Bank) 1: 加速从程序 flash 存储器至器件的数据传输:
 - 对于每个主设备和存储阵列 (Bank), 提供一个 64 位预取预测缓存, 用户可控制该缓存的
 - 对每一个存储阵列 (Bank), 提供 4 路、8 组、64 位线大小高速缓存。这样总计实现 32 路 64 位输入, 且提供对的每路替换算法和锁定控制
 - 每个存储阵列 (Bank) 均有一个单入口缓存
 - 对预测缓存和单入口缓存的失效控制

29.2 工作模式

FMC 只在某个总线主机访问 flash 存储器时工作。

对于任何无法访问 flash 存储器的器件功耗模式, FMC 都将被禁用。

29.3 外部信号说明

FMC 上不使用外部信号。

29.4 存储器映射和寄存器说明

编程模型包含 FMC 控制寄存器和程序可视高速缓存 (数据和标签/有效条目)。

注

仅当 flash 控制器空闲时 (如通过 RAM 执行) 对其寄存器进行编程。在存取 Flash 时更改 Flash 设置会导致不确定的行为。

表 29-2. FMC 寄存器访问

寄存器	读取访问		写入访问	
	模式	长度	模式	长度
控制寄存器: PFAPR、PFB0CR、PFB1CR	管理员 (特权) 模式或用户模式	32 位	仅管理员 (特权) 模式	32 位
高速缓存寄存器	管理员 (特权) 模式或用户模式	32 位	仅管理员 (特权) 模式	32 位

注

访问 FMC 4 KB 地址空间内的没有列出地址的寄存器将会返回总线错误。

可随时读取高速缓存条目 (数据和标签/有效条目)。

注

对 flash 高速缓存的任何段进行写操作时, 系统软件一定要维持存储器的一致性。例如, 与被重新写入的 flash 相关的所有缓冲器数据都应失效。相应地, 在完成写或擦除事件之后, 访问新存储器映射之前, 相关数据必须显式写入高速缓存。

此高速缓存为 4 路, 8 组的组相联高速缓存。路编号为 0-3, 组编号为 0-7。下表详述了标签/有效条目和数据条目。

表 29-3. 编程可视高速缓存寄存器

高速缓存存储器	基于偏移量	32 位读取内容	命名法	命名法示例
标签	100h	13'h0, 标签[18:5], 4'h0, 有效 d	在 TAGVDWxSy 中, x 表示路, y 表示组。	TAGVDW2S0 为高速缓存条目 2 路 0 组的 14 位标签位和 1 位有效位。
数据	200h	上部或下部数据长字	在 DATAWxSyU 和 DATAWxSyL 中, x 表示路, y 表示组, U 和 L 分别代表上字和下字。	DATAW1S0U 数据条目 1 路 0 组的位[63:32], DATAW1S0L 数据条目 1 路 0 组的位[31:0]。

FMC 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4001_F000	Flash 访问保护寄存器 (FMC_PFAPR)	32	R/W	00F8_003Fh	29.4.1/523
4001_F004	Flash 存储阵列 0 控制寄存器 (FMC_PFB0CR)	32	R/W	3002_001Fh	29.4.2/526
4001_F008	Reserved (FMC_Reserved)	32	R/W	3000_0000h	29.4.3/528
4001_F100	高速缓存标签存储器 (FMC_TAGVDW0S0)	32	R/W	0000_0000h	29.4.4/529
4001_F104	高速缓存标签存储器 (FMC_TAGVDW0S1)	32	R/W	0000_0000h	29.4.4/529
4001_F108	高速缓存标签存储器 (FMC_TAGVDW0S2)	32	R/W	0000_0000h	29.4.4/529

下一页继续介绍此表...

FMC 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4001_F10C	高速缓存标签存储器 (FMC_TAGVDW0S3)	32	R/W	0000_0000h	29.4.4/529
4001_F110	高速缓存标签存储器 (FMC_TAGVDW0S4)	32	R/W	0000_0000h	29.4.4/529
4001_F114	高速缓存标签存储器 (FMC_TAGVDW0S5)	32	R/W	0000_0000h	29.4.4/529
4001_F118	高速缓存标签存储器 (FMC_TAGVDW0S6)	32	R/W	0000_0000h	29.4.4/529
4001_F11C	高速缓存标签存储器 (FMC_TAGVDW0S7)	32	R/W	0000_0000h	29.4.4/529
4001_F120	高速缓存标签存储器 (FMC_TAGVDW1S0)	32	R/W	0000_0000h	29.4.5/530
4001_F124	高速缓存标签存储器 (FMC_TAGVDW1S1)	32	R/W	0000_0000h	29.4.5/530
4001_F128	高速缓存标签存储器 (FMC_TAGVDW1S2)	32	R/W	0000_0000h	29.4.5/530
4001_F12C	高速缓存标签存储器 (FMC_TAGVDW1S3)	32	R/W	0000_0000h	29.4.5/530
4001_F130	高速缓存标签存储器 (FMC_TAGVDW1S4)	32	R/W	0000_0000h	29.4.5/530
4001_F134	高速缓存标签存储器 (FMC_TAGVDW1S5)	32	R/W	0000_0000h	29.4.5/530
4001_F138	高速缓存标签存储器 (FMC_TAGVDW1S6)	32	R/W	0000_0000h	29.4.5/530
4001_F13C	高速缓存标签存储器 (FMC_TAGVDW1S7)	32	R/W	0000_0000h	29.4.5/530
4001_F140	高速缓存标签存储器 (FMC_TAGVDW2S0)	32	R/W	0000_0000h	29.4.6/531
4001_F144	高速缓存标签存储器 (FMC_TAGVDW2S1)	32	R/W	0000_0000h	29.4.6/531
4001_F148	高速缓存标签存储器 (FMC_TAGVDW2S2)	32	R/W	0000_0000h	29.4.6/531
4001_F14C	高速缓存标签存储器 (FMC_TAGVDW2S3)	32	R/W	0000_0000h	29.4.6/531
4001_F150	高速缓存标签存储器 (FMC_TAGVDW2S4)	32	R/W	0000_0000h	29.4.6/531
4001_F154	高速缓存标签存储器 (FMC_TAGVDW2S5)	32	R/W	0000_0000h	29.4.6/531
4001_F158	高速缓存标签存储器 (FMC_TAGVDW2S6)	32	R/W	0000_0000h	29.4.6/531
4001_F15C	高速缓存标签存储器 (FMC_TAGVDW2S7)	32	R/W	0000_0000h	29.4.6/531
4001_F160	高速缓存标签存储器 (FMC_TAGVDW3S0)	32	R/W	0000_0000h	29.4.7/532
4001_F164	高速缓存标签存储器 (FMC_TAGVDW3S1)	32	R/W	0000_0000h	29.4.7/532
4001_F168	高速缓存标签存储器 (FMC_TAGVDW3S2)	32	R/W	0000_0000h	29.4.7/532
4001_F16C	高速缓存标签存储器 (FMC_TAGVDW3S3)	32	R/W	0000_0000h	29.4.7/532
4001_F170	高速缓存标签存储器 (FMC_TAGVDW3S4)	32	R/W	0000_0000h	29.4.7/532
4001_F174	高速缓存标签存储器 (FMC_TAGVDW3S5)	32	R/W	0000_0000h	29.4.7/532
4001_F178	高速缓存标签存储器 (FMC_TAGVDW3S6)	32	R/W	0000_0000h	29.4.7/532
4001_F17C	高速缓存标签存储器 (FMC_TAGVDW3S7)	32	R/W	0000_0000h	29.4.7/532
4001_F200	高速缓存数据存储器 (上字) (FMC_DATAW0S0U)	32	R/W	0000_0000h	29.4.8/532
4001_F204	高速缓存数据存储器 (下字) (FMC_DATAW0S0L)	32	R/W	0000_0000h	29.4.9/533
4001_F208	高速缓存数据存储器 (上字) (FMC_DATAW0S1U)	32	R/W	0000_0000h	29.4.8/532
4001_F20C	高速缓存数据存储器 (下字) (FMC_DATAW0S1L)	32	R/W	0000_0000h	29.4.9/533
4001_F210	高速缓存数据存储器 (上字) (FMC_DATAW0S2U)	32	R/W	0000_0000h	29.4.8/532
4001_F214	高速缓存数据存储器 (下字) (FMC_DATAW0S2L)	32	R/W	0000_0000h	29.4.9/533
4001_F218	高速缓存数据存储器 (上字) (FMC_DATAW0S3U)	32	R/W	0000_0000h	29.4.8/532
4001_F21C	高速缓存数据存储器 (下字) (FMC_DATAW0S3L)	32	R/W	0000_0000h	29.4.9/533
4001_F220	高速缓存数据存储器 (上字) (FMC_DATAW0S4U)	32	R/W	0000_0000h	29.4.8/532
4001_F224	高速缓存数据存储器 (下字) (FMC_DATAW0S4L)	32	R/W	0000_0000h	29.4.9/533

下一页继续介绍此表...

FMC 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4001_F228	高速缓存数据存储器 (上字) (FMC_DATAW0S5U)	32	R/W	0000_0000h	29.4.8/532
4001_F22C	高速缓存数据存储器 (下字) (FMC_DATAW0S5L)	32	R/W	0000_0000h	29.4.9/533
4001_F230	高速缓存数据存储器 (上字) (FMC_DATAW0S6U)	32	R/W	0000_0000h	29.4.8/532
4001_F234	高速缓存数据存储器 (下字) (FMC_DATAW0S6L)	32	R/W	0000_0000h	29.4.9/533
4001_F238	高速缓存数据存储器 (上字) (FMC_DATAW0S7U)	32	R/W	0000_0000h	29.4.8/532
4001_F23C	高速缓存数据存储器 (下字) (FMC_DATAW0S7L)	32	R/W	0000_0000h	29.4.9/533
4001_F240	高速缓存数据存储器 (上字) (FMC_DATAW1S0U)	32	R/W	0000_0000h	29.4.10/533
4001_F244	高速缓存数据存储器 (下字) (FMC_DATAW1S0L)	32	R/W	0000_0000h	29.4.11/534
4001_F248	高速缓存数据存储器 (上字) (FMC_DATAW1S1U)	32	R/W	0000_0000h	29.4.10/533
4001_F24C	高速缓存数据存储器 (下字) (FMC_DATAW1S1L)	32	R/W	0000_0000h	29.4.11/534
4001_F250	高速缓存数据存储器 (上字) (FMC_DATAW1S2U)	32	R/W	0000_0000h	29.4.10/533
4001_F254	高速缓存数据存储器 (下字) (FMC_DATAW1S2L)	32	R/W	0000_0000h	29.4.11/534
4001_F258	高速缓存数据存储器 (上字) (FMC_DATAW1S3U)	32	R/W	0000_0000h	29.4.10/533
4001_F25C	高速缓存数据存储器 (下字) (FMC_DATAW1S3L)	32	R/W	0000_0000h	29.4.11/534
4001_F260	高速缓存数据存储器 (上字) (FMC_DATAW1S4U)	32	R/W	0000_0000h	29.4.10/533
4001_F264	高速缓存数据存储器 (下字) (FMC_DATAW1S4L)	32	R/W	0000_0000h	29.4.11/534
4001_F268	高速缓存数据存储器 (上字) (FMC_DATAW1S5U)	32	R/W	0000_0000h	29.4.10/533
4001_F26C	高速缓存数据存储器 (下字) (FMC_DATAW1S5L)	32	R/W	0000_0000h	29.4.11/534
4001_F270	高速缓存数据存储器 (上字) (FMC_DATAW1S6U)	32	R/W	0000_0000h	29.4.10/533
4001_F274	高速缓存数据存储器 (下字) (FMC_DATAW1S6L)	32	R/W	0000_0000h	29.4.11/534
4001_F278	高速缓存数据存储器 (上字) (FMC_DATAW1S7U)	32	R/W	0000_0000h	29.4.10/533
4001_F27C	高速缓存数据存储器 (下字) (FMC_DATAW1S7L)	32	R/W	0000_0000h	29.4.11/534
4001_F280	高速缓存数据存储器 (上字) (FMC_DATAW2S0U)	32	R/W	0000_0000h	29.4.12/534
4001_F284	高速缓存数据存储器 (下字) (FMC_DATAW2S0L)	32	R/W	0000_0000h	29.4.13/535
4001_F288	高速缓存数据存储器 (上字) (FMC_DATAW2S1U)	32	R/W	0000_0000h	29.4.12/534

下一页继续介绍此表...

FMC 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4001_F28C	高速缓存数据存储器 (下字) (FMC_DATAW2S1L)	32	R/W	0000_0000h	29.4.13/535
4001_F290	高速缓存数据存储器 (上字) (FMC_DATAW2S2U)	32	R/W	0000_0000h	29.4.12/534
4001_F294	高速缓存数据存储器 (下字) (FMC_DATAW2S2L)	32	R/W	0000_0000h	29.4.13/535
4001_F298	高速缓存数据存储器 (上字) (FMC_DATAW2S3U)	32	R/W	0000_0000h	29.4.12/534
4001_F29C	高速缓存数据存储器 (下字) (FMC_DATAW2S3L)	32	R/W	0000_0000h	29.4.13/535
4001_F2A0	高速缓存数据存储器 (上字) (FMC_DATAW2S4U)	32	R/W	0000_0000h	29.4.12/534
4001_F2A4	高速缓存数据存储器 (下字) (FMC_DATAW2S4L)	32	R/W	0000_0000h	29.4.13/535
4001_F2A8	高速缓存数据存储器 (上字) (FMC_DATAW2S5U)	32	R/W	0000_0000h	29.4.12/534
4001_F2AC	高速缓存数据存储器 (下字) (FMC_DATAW2S5L)	32	R/W	0000_0000h	29.4.13/535
4001_F2B0	高速缓存数据存储器 (上字) (FMC_DATAW2S6U)	32	R/W	0000_0000h	29.4.12/534
4001_F2B4	高速缓存数据存储器 (下字) (FMC_DATAW2S6L)	32	R/W	0000_0000h	29.4.13/535
4001_F2B8	高速缓存数据存储器 (上字) (FMC_DATAW2S7U)	32	R/W	0000_0000h	29.4.12/534
4001_F2BC	高速缓存数据存储器 (下字) (FMC_DATAW2S7L)	32	R/W	0000_0000h	29.4.13/535
4001_F2C0	高速缓存数据存储器 (上字) (FMC_DATAW3S0U)	32	R/W	0000_0000h	29.4.14/535
4001_F2C4	高速缓存数据存储器 (下字) (FMC_DATAW3S0L)	32	R/W	0000_0000h	29.4.15/536
4001_F2C8	高速缓存数据存储器 (上字) (FMC_DATAW3S1U)	32	R/W	0000_0000h	29.4.14/535
4001_F2CC	高速缓存数据存储器 (下字) (FMC_DATAW3S1L)	32	R/W	0000_0000h	29.4.15/536
4001_F2D0	高速缓存数据存储器 (上字) (FMC_DATAW3S2U)	32	R/W	0000_0000h	29.4.14/535
4001_F2D4	高速缓存数据存储器 (下字) (FMC_DATAW3S2L)	32	R/W	0000_0000h	29.4.15/536
4001_F2D8	高速缓存数据存储器 (上字) (FMC_DATAW3S3U)	32	R/W	0000_0000h	29.4.14/535
4001_F2DC	高速缓存数据存储器 (下字) (FMC_DATAW3S3L)	32	R/W	0000_0000h	29.4.15/536
4001_F2E0	高速缓存数据存储器 (上字) (FMC_DATAW3S4U)	32	R/W	0000_0000h	29.4.14/535

下一页继续介绍此表...

FMC 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4001_F2E4	高速缓存数据存储器 (下字) (FMC_DATAW3S4L)	32	R/W	0000_0000h	29.4.15/536
4001_F2E8	高速缓存数据存储器 (上字) (FMC_DATAW3S5U)	32	R/W	0000_0000h	29.4.14/535
4001_F2EC	高速缓存数据存储器 (下字) (FMC_DATAW3S5L)	32	R/W	0000_0000h	29.4.15/536
4001_F2F0	高速缓存数据存储器 (上字) (FMC_DATAW3S6U)	32	R/W	0000_0000h	29.4.14/535
4001_F2F4	高速缓存数据存储器 (下字) (FMC_DATAW3S6L)	32	R/W	0000_0000h	29.4.15/536
4001_F2F8	高速缓存数据存储器 (上字) (FMC_DATAW3S7U)	32	R/W	0000_0000h	29.4.14/535
4001_F2FC	高速缓存数据存储器 (下字) (FMC_DATAW3S7L)	32	R/W	0000_0000h	29.4.15/536

29.4.1 Flash 访问保护寄存器 (FMC_PFAPR)

地址: 4001_F000h 基准 + 0h 偏移 = 4001_F000h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	保留								M7PFD	M6PFD	M5PFD	M4PFD	M3PFD	M2PFD	M1PFD	M0PFD
W																
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M7AP[1:0]	M6AP[1:0]	M5AP[1:0]	M4AP[1:0]	M3AP[1:0]	M2AP[1:0]	M1AP[1:0]	M0AP[1:0]								
W																
复位	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

FMC_PFAPR 字段描述

字段	描述
31-24 Reserved	此字段为保留字段。
23 M7PFD	<p>主机 7 预取禁用</p> <p>这些位根据请求交叉开关主机的逻辑号控制是否使能预取。此字段将通过 PFBnCR[BxDPE,BxIPE]位进一步限定。</p> <p>0 此主机已使能预取。 1 此主机已禁用预取。</p>

下一页继续介绍此表...

FMC_PFAPR 字段描述 (继续)

字段	描述
22 M6PFD	<p>主机 6 预取禁用</p> <p>这些位根据请求交叉开关主机的逻辑号控制是否使能预取。此字段将通过 PFBnCR[BxDPE,BxIPE]位进一步限定。</p> <p>0 此主机已使能预取。 1 此主机已禁用预取。</p>
21 M5PFD	<p>主机 5 预取禁用</p> <p>这些位根据请求交叉开关主机的逻辑号控制是否使能预取。此字段将通过 PFBnCR[BxDPE,BxIPE]位进一步限定。</p> <p>0 此主机已使能预取。 1 此主机已禁用预取。</p>
20 M4PFD	<p>主机 4 预取禁用</p> <p>这些位根据请求交叉开关主机的逻辑号控制是否使能预取。此字段将通过 PFBnCR[BxDPE,BxIPE]位进一步限定。</p> <p>0 此主机已使能预取。 1 此主机已禁用预取。</p>
19 M3PFD	<p>主机 3 预取禁用</p> <p>这些位根据请求交叉开关主机的逻辑号控制是否使能预取。此字段将通过 PFBnCR[BxDPE,BxIPE]位进一步限定。</p> <p>0 此主机已使能预取。 1 此主机已禁用预取。</p>
18 M2PFD	<p>主机 2 预取禁用</p> <p>这些位根据请求交叉开关主机的逻辑号控制是否使能预取。此字段将通过 PFBnCR[BxDPE,BxIPE]位进一步限定。</p> <p>0 此主机已使能预取。 1 此主机已禁用预取。</p>
17 M1PFD	<p>主机 1 预取禁用</p> <p>这些位根据请求交叉开关主机的逻辑号控制是否使能预取。此字段将通过 PFBnCR[BxDPE,BxIPE]位进一步限定。</p> <p>0 此主机已使能预取。 1 此主机已禁用预取。</p>
16 M0PFD	<p>主机 0 预取禁用</p> <p>这些位根据请求交叉开关主机的逻辑号控制是否使能预取。此字段将通过 PFBnCR[BxDPE,BxIPE]位进一步限定。</p> <p>0 此主机已使能预取。 1 此主机已禁用预取。</p>
15-14 M7AP[1:0]	<p>主机 7 访问保护</p> <p>此字段根据请求交叉开关主机的逻辑主机号，控制是否允许对 flash 进行读取和写入访问。</p>

下一页继续介绍此表...

FMC_PFAPR 字段描述 (继续)

字段	描述
	00 此主机不可执行访问。 01 此主机仅可执行读取访问。 10 此主机仅可执行写入访问。 11 此主机可执行读取和写入访问。
13-12 M6AP[1:0]	主机 6 访问保护 此字段根据请求交叉开关主机的逻辑主机号，控制是否允许对 flash 进行读取和写入访问。 00 此主机不可执行访问 01 此主机仅可执行读取访问 10 此主机仅可执行写入访问 11 此主机可执行读取和写入访问
11-10 M5AP[1:0]	主机 5 访问保护 此字段根据请求交叉开关主机的逻辑主机号，控制是否允许对 flash 进行读取和写入访问。 00 此主机不可执行访问 01 此主机仅可执行读取访问 10 此主机仅可执行写入访问 11 此主机可执行读取和写入访问
9-8 M4AP[1:0]	主机 4 访问保护 此字段根据请求交叉开关主机的逻辑主机号，控制是否允许对 flash 进行读取和写入访问。 00 此主机不可执行访问 01 此主机仅可执行读取访问 10 此主机仅可执行写入访问 11 此主机可执行读取和写入访问
7-6 M3AP[1:0]	主机 3 访问保护 此字段根据请求交叉开关主机的逻辑主机号，控制是否允许对 flash 进行读取和写入访问。 00 此主机不可执行访问 01 此主机仅可执行读取访问 10 此主机仅可执行写入访问 11 此主机可执行读取和写入访问
5-4 M2AP[1:0]	主机 2 访问保护 此字段根据请求交叉开关主机的逻辑主机号，控制是否允许对 flash 进行读取和写入访问。 00 此主机不可执行访问 01 此主机仅可执行读取访问 10 此主机仅可执行写入访问 11 此主机可执行读取和写入访问
3-2 M1AP[1:0]	主机 1 访问保护 此字段根据请求交叉开关主机的逻辑主机号，控制是否允许对 flash 进行读取和写入访问。 00 此主机不可执行访问 01 此主机仅可执行读取访问

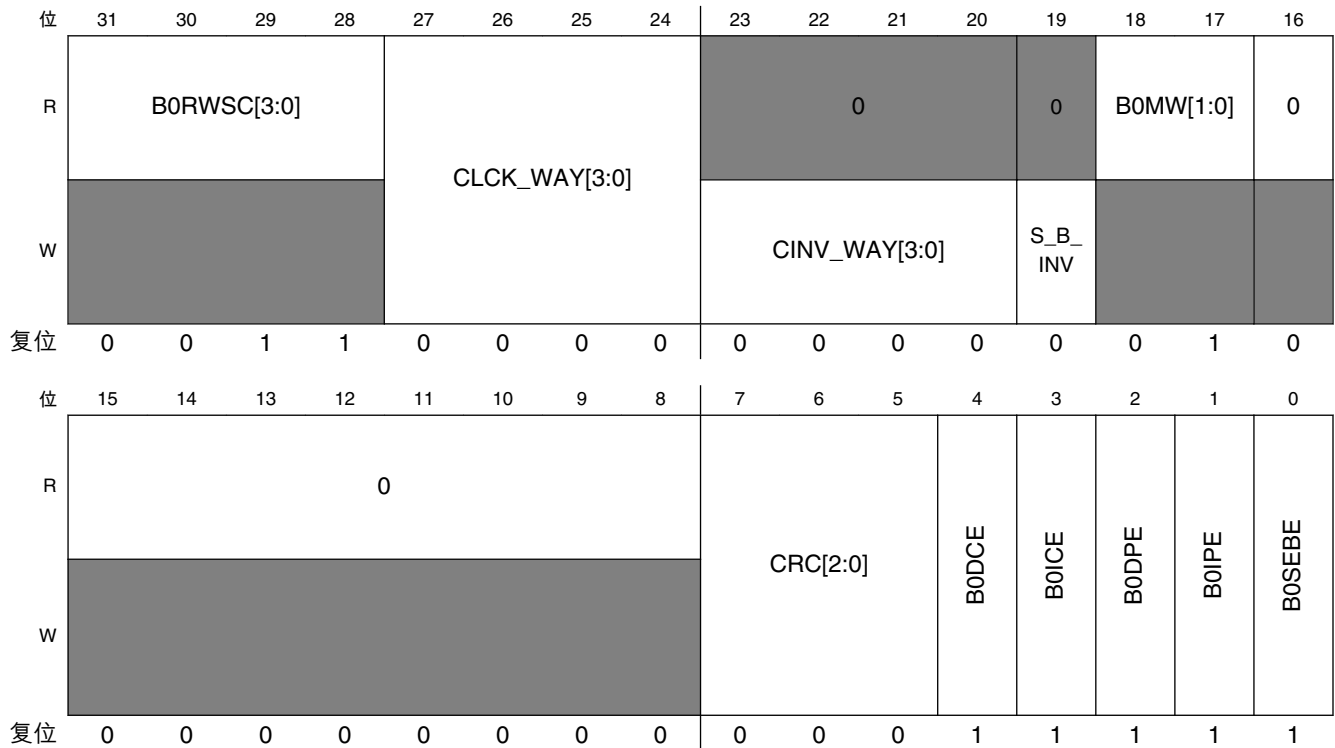
下一页继续介绍此表...

FMC_PFAPR 字段描述 (继续)

字段	描述
	10 此主机仅可执行写入访问 11 此主机可执行读取和写入访问
M0AP[1:0]	主机 0 访问保护 此字段根据请求交叉开关主机的逻辑主机号，控制是否允许对 flash 进行读取和写入访问。 00 此主机不可执行访问 01 此主机仅可执行读取访问 10 此主机仅可执行写入访问 11 此主机可执行读取和写入访问

29.4.2 Flash 存储阵列 0 控制寄存器 (FMC_PFB0CR)

地址: 4001_F000h 基准 + 4h 偏移 = 4001_F004h



FMC_PFB0CR 字段描述

字段	描述
31-28 B0RWSC[3:0]	存储阵列 0 读取等待状态控制 此只读字段定义访问存储阵列 0 flash 存储器所需的等待状态数。 Flash 阵列和 RWSC 读取访问时间之间的关系 (以系统时钟周期表示) 如下所示： Flash 阵列的访问时间[系统时钟] = RWSC + 1

下一页继续介绍此表...

FMC_PFB0CR 字段描述 (继续)

字段	描述
	FMC 将根据系统时钟速度与 flash 时钟速度之比自动计算此值。例如，当比率为 4:1，此字段值为 3h。
27-24 CLK_WAY[3:0]	<p>高速缓存 x 路锁定</p> <p>这些位确定是否锁定给定的高速缓存路以使其内容不会被后续未命中内容替代。</p> <p>此位设置定义针对字段中的每个位。</p> <p>0 高速缓存路已解锁并且可能被替代 1 高速缓存路已锁定并且其内容未被替代</p>
23-20 CINV_WAY[3:0]	<p>高速缓存 x 路失效</p> <p>这些位确定给定高速缓存路是否会失效 (被清除)。在此字段写入位之后，对应的高速缓存路立即失效：将清除该路的标签、数据和有效内容。此字段始终读为 0。</p> <p>高速缓存失效优先于锁定。此高速缓存器因系统复位而失效。对 Flash 存储器的任意段进行编程或擦除操作时，需要系统软件维持存储器一致性。相应地，缓存器必须在完成编程或擦除事件之后，访问新存储器映射之前失效。</p> <p>此位设置定义针对字段中的每个位。</p> <p>0 对应高速缓存中无高速缓存路失效 1 对应高速缓存的高速缓存路失效：清除选定路的标签、数据和 vld 位。</p>
19 S_B_INV	<p>使预取推测缓冲器失效</p> <p>此位确定 FMC 的预取推测缓冲器和单条目页面缓冲器是否会失效 (被清除)。写入此位时，将会立即清除推测缓冲器和单条目缓冲器。此位始终读为 0。</p> <p>0 推测缓冲器和单条目缓冲器不受影响。 1 (清除) 推测缓冲器和单条目缓冲器失效</p>
18-17 BOMW[1:0]	<p>存储阵列 0 存储器宽度</p> <p>此只读字段定义存储阵列 0 存储器的宽度。</p> <p>00 32 位 01 64 bits 10 保留 11 保留</p>
16 保留	<p>此字段为保留字段。</p> <p>此只读字段为保留字段且值始终为 0。</p>
15-8 保留	<p>此字段为保留字段。</p> <p>此只读字段为保留字段且值始终为 0。</p>
7-5 CRC[2:0]	<p>高速缓存替换控制</p> <p>此 3 位字段定义高速缓存访问的替换算法。</p> <p>000 所有 4 路中每个组的 LRU 替换算法 001 保留 010 独立的 LRU，[0-1]路用于 ifetches，[2-3]路用于数据 011 独立的 LRU，[0-2]路用于 ifetches，[3]路用于数据 1xx 保留</p>
4 B0DCE	<p>存储阵列 0 数据高速缓存使能</p> <p>此位控制是否将数据参考加载到高速缓存中。</p>

下一页继续介绍此表...

FMC_PFB0CR 字段描述 (继续)

字段	描述
	0 不高速缓存数据参考。 1 高速缓存数据参考。
3 B0ICE	存储阵列 0 指令高速缓存使能 此位控制是否将指令提取加载到高速缓存中。 0 不高速缓存指令提取。 1 高速缓存指令提取。
2 B0DPE	存储阵列 0 数据预取使能 此位控制是否发起预取 (或推测式访问) 以响应数据参考。 0 不预取以响应数据参考。 1 使能预取以响应数据参考。
1 B0IPE	存储阵列 0 指令预取使能 此位控制是否发起预取 (或推测式访问) 以响应指令提取。 0 不预取以响应指令提取。 1 使能预取以响应指令提取。
0 B0SEBE	存储阵列 0 单条目缓冲器使能 此位控制是否使能单条目页面缓冲器, 以响应 flash 读取访问。 此位从高电平跃进到低电平将强制页面缓冲器无效。 0 单条目缓冲器已禁用。 1 单条目缓冲器已使能。

29.4.3 Reserved (FMC_Reserved)

此寄存器地址已保留。

地址: 4001_F000h 基准 + 8h 偏移 = 4001_F008h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R	保留								0						保留		0
W	保留																
复位	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0								0				0				
W	保留																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

FMC_Reserved 字段描述

字段	描述
31-28 Reserved	此字段为保留字段。
27-19 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
18-17 Reserved	此字段为保留字段。
16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7-5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

29.4.4 高速缓存标签存储器 (FMC_TAGVDW0Sn)

此高速缓存为 4 路 8 组的组相联高速缓存。这些路编号为 0-3，组编号为 0-7。在 TAGVDW_xS_y 中，x 表示路，y 表示组。本节介绍了所示路中所有组的标签/vld 信息。

Address: 4001_F000h base + 100h offset + (4d × i), where i=0d to 7d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													tag[18:5]		
W	0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tag[18:5]											0				有效
W	0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

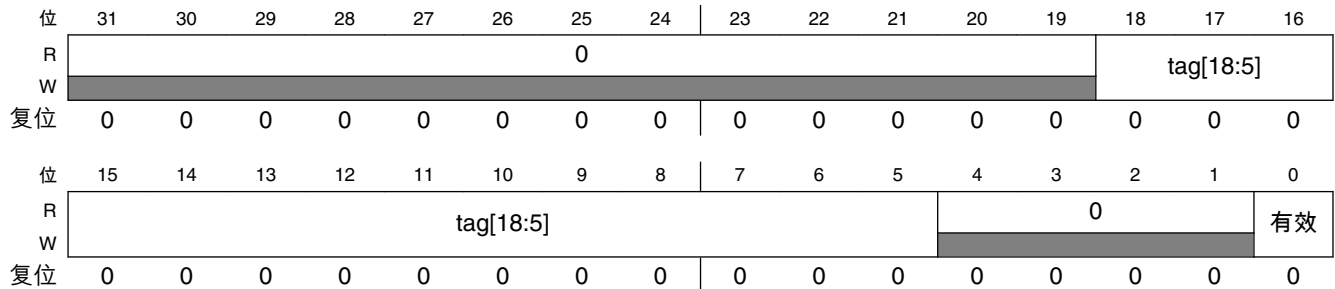
FMC_TAGVDW0Sn 字段描述

字段	描述
31-19 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
18-5 tag[18:5]	用于高速缓存条目的 14 位标签
4-1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
0 有效	1 位，对高速缓存条目有效

29.4.5 高速缓存标签存储器 (FMC_TAGVDW1Sn)

此高速缓存为 4 路 8 组的组相联高速缓存。这些路编号为 0-3，组编号为 0-7。在 TAGVDW_xS_y 中，x 表示路，y 表示组。本节介绍了所示路中所有组的标签/vld 信息。

Address: 4001_F000h base + 120h offset + (4d × i), where i=0d to 7d



FMC_TAGVDW1Sn 字段描述

字段	描述
31-19 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
18-5 tag[18:5]	用于高速缓存条目的 14 位标签
4-1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
0 有效	1 位，对高速缓存条目有效

29.4.6 高速缓存标签存储器 (FMC_TAGVDW2Sn)

此高速缓存为 4 路 8 组的组相联高速缓存。这些路编号为 0-3，组编号为 0-7。在 TAGVDW_xS_y 中，x 表示路，y 表示组。本节介绍了所示路中所有组的标签/vld 信息。

Address: 4001_F000h base + 140h offset + (4d × i), where i=0d to 7d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														tag[18:5]	
W	0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tag[18:5]											0				有效
W	0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

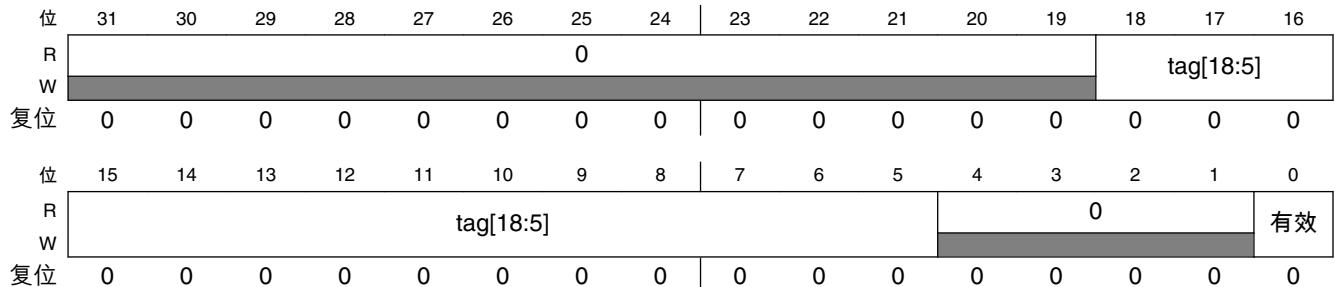
FMC_TAGVDW2Sn 字段描述

字段	描述
31-19 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
18-5 tag[18:5]	用于高速缓存条目的 14 位标签
4-1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
0 有效	1 位，对高速缓存条目有效

29.4.7 高速缓存标签存储器 (FMC_TAGVDW3Sn)

此高速缓存为 4 路 8 组的组相联高速缓存。这些路编号为 0-3，组编号为 0-7。在 TAGVDW_xS_y 中，x 表示路，y 表示组。本节介绍了所示路中所有组的标签/vld 信息。

Address: 4001_F000h base + 160h offset + (4d × i), where i=0d to 7d



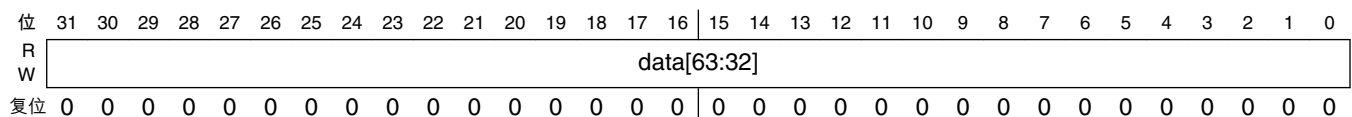
FMC_TAGVDW3Sn 字段描述

字段	描述
31-19 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
18-5 tag[18:5]	用于高速缓存条目的 14 位标签
4-1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
0 有效	1 位，对高速缓存条目有效

29.4.8 高速缓存数据存储器（上字） (FMC_DATAW0SnU)

此 64 位条目高速缓存为 4 路组相联高速缓存，具有 8 个组。这些路编号为 0-3，组编号为 0-7。在 DATAW_xS_yU 和 DATAW_xS_yL 中，x 表示路，y 表示组，U 和 L 分别代表上字和下字。本节介绍了所示路中所有组的上字（位[63:32]）数据。

Address: 4001_F000h base + 200h offset + (8d × i), where i=0d to 7d



FMC_DATAW0SnU 字段描述

字段	描述
data[63:32]	数据条目的位[63:32]

29.4.9 高速缓存数据存储器（下字）(FMC_DATAW0SnL)

此 64 位条目高速缓存为 4 路组相联高速缓存，具有 8 个组。这些路编号为 0-3，组编号为 0-7。在 DATAWxSyU 和 DATAWxSyL 中，x 表示路，y 表示组，U 和 L 分别代表上字和下字。本节介绍了所示路中所有组的下字（位[31:0]）数据。

Address: 4001_F000h base + 204h offset + (8d × i), where i=0d to 7d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW0SnL 字段描述

字段	描述
data[31:0]	数据条目的位[31:0]

29.4.10 高速缓存数据存储器（上字）(FMC_DATAW1SnU)

此 64 位条目高速缓存为 4 路组相联高速缓存，具有 8 个组。这些路编号为 0-3，组编号为 0-7。在 DATAWxSyU 和 DATAWxSyL 中，x 表示路，y 表示组，U 和 L 分别代表上字和下字。本节介绍了所示路中所有组的上字（位[63:32]）数据。

Address: 4001_F000h base + 240h offset + (8d × i), where i=0d to 7d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

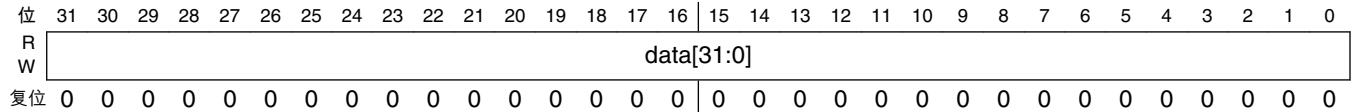
FMC_DATAW1SnU 字段描述

字段	描述
data[63:32]	数据条目的位[63:32]

29.4.11 高速缓存数据存储器（下字） (FMC_DATAW1SnL)

此 64 位条目高速缓存为 4 路组相联高速缓存，具有 8 个组。这些路编号为 0-3，组编号为 0-7。在 DATAWxSyU 和 DATAWxSyL 中，x 表示路，y 表示组，U 和 L 分别代表上字和下字。本节介绍了所示路中所有组的下字（位[31:0]）数据。

Address: 4001_F000h base + 244h offset + (8d × i), where i=0d to 7d



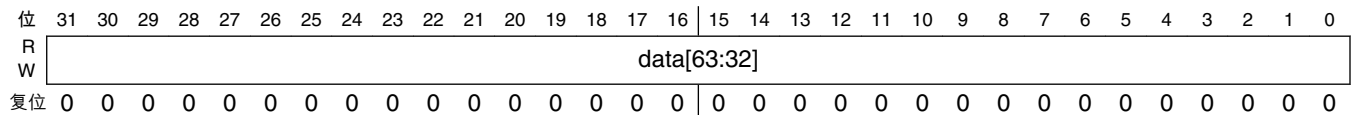
FMC_DATAW1SnL 字段描述

字段	描述
data[31:0]	数据条目的位[63:32]

29.4.12 高速缓存数据存储器（上字） (FMC_DATAW2SnU)

此 64 位条目高速缓存为 4 路组相联高速缓存，具有 8 个组。这些路编号为 0-3，组编号为 0-7。在 DATAWxSyU 和 DATAWxSyL 中，x 表示路，y 表示组，U 和 L 分别代表上字和下字。本节介绍了所示路中所有组的上字（位[63:32]）数据。

Address: 4001_F000h base + 280h offset + (8d × i), where i=0d to 7d



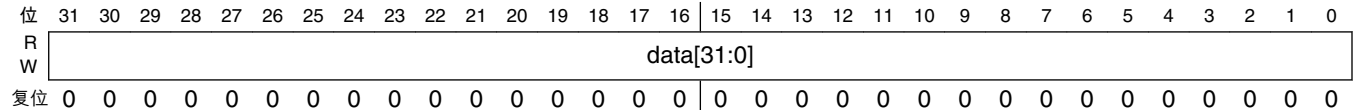
FMC_DATAW2SnU 字段描述

字段	描述
data[63:32]	数据条目的位[63:32]

29.4.13 高速缓存数据存储器（下字）(FMC_DATAW2SnL)

此 64 位条目高速缓存为 4 路组相联高速缓存，具有 8 个组。这些路编号为 0-3，组编号为 0-7。在 DATAW_xSyU 和 DATAW_xSyL 中，x 表示路，y 表示组，U 和 L 分别代表上字和下字。本节介绍了所示路中所有组的下字（位[31:0]）数据。

Address: 4001_F000h base + 284h offset + (8d × i), where i=0d to 7d



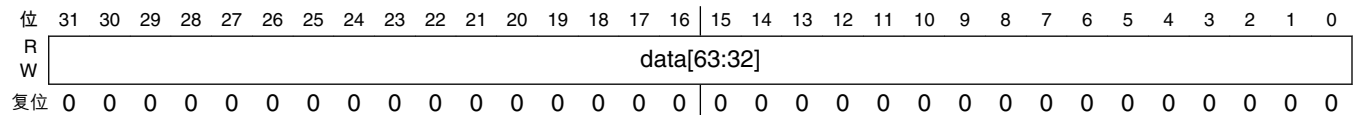
FMC_DATAW2SnL 字段描述

字段	描述
data[31:0]	数据条目的位[63:32]

29.4.14 高速缓存数据存储器（上字）(FMC_DATAW3SnU)

此 64 位条目高速缓存为 4 路组相联高速缓存，具有 8 个组。这些路编号为 0-3，组编号为 0-7。在 DATAW_xSyU 和 DATAW_xSyL 中，x 表示路，y 表示组，U 和 L 分别代表上字和下字。本节介绍了所示路中所有组的上字（位[63:32]）数据。

Address: 4001_F000h base + 2C0h offset + (8d × i), where i=0d to 7d



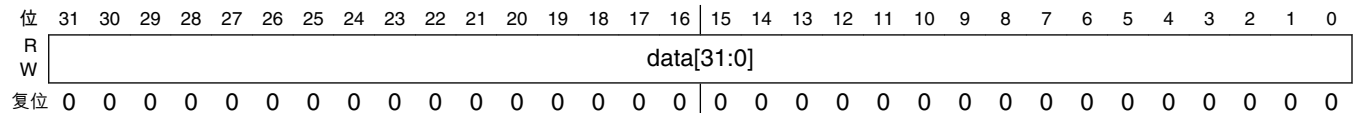
FMC_DATAW3SnU 字段描述

字段	描述
data[63:32]	数据条目的位[63:32]

29.4.15 高速缓存数据存储器（下字） (FMC_DATAW3SnL)

此 64 位条目高速缓存为 4 路组相联高速缓存，具有 8 个组。这些路编号为 0-3，组编号为 0-7。在 DATAW_xS_yU 和 DATAW_xS_yL 中，x 表示路，y 表示组，U 和 L 分别代表上字和下字。本节介绍了所示路中所有组的下字（位[31:0]）数据。

Address: 4001_F000h base + 2C4h offset + (8d × i), where i=0d to 7d



FMC_DATAW3SnL 字段描述

字段	描述
data[31:0]	数据条目的位[63:32]

29.5 功能说明

FMC 是一个 Flash 加速单元，拥有可供用户配置的灵活缓存区。除了管理器件与 flash 存储器之间的接口以外，FMC 还可用于限制从交叉开关主机访问以及定制高速缓存和缓存区，以便提供单周期系统时钟数据访问时间。无论何时，只要命中预取推测缓存区、高速缓存或单式缓冲器，请求的数据就会在单个系统时钟周期内完成传输。

29.5.1 默认配置

系统复位时，可配置 FMC，以便对从 flash 存储器进行传输提供高水平缓冲。

- 交叉主机 0、1、2 可读取访问存储阵列 0。
- 对于存储阵列 0:
 - 对于交叉主机 0、1、2，已使能数据和指令的预取支持。
 - 针对最近最少使用(LRU)的所有 4 路替换，配置高速缓存。
 - 针对数据或指令替换，配置高速缓存。
 - 已使能单式缓冲器。

29.5.2 配置选项

尽管默认配置提供高等级 Flash 加速，但高级用户可能想要定制 FMC 缓冲器配置，以最大程度提高其使用案例的吞吐量。针对自定义用例重新配置 FMC 时，请勿在访问 flash 存储器的同时对 FMC 的控制寄存器进行编程。相反，应在管理模式下从 RAM 执行例程，由此更改控制寄存器。

可通过 PFB0CR 内的 FMC 高速缓存和缓冲控制调整资源，以适合特定应用需求。高速缓存和缓冲单独受控。寄存器控制可按照访问类型（指令提取或数据引用）使能缓冲和预取。高速缓存还支持 3 类 LRU 替换算法：

- 所有 4 路中每个组的 LRU，
- LRU，[0-1]路用于指令提取，[2-3]路用于数据读取，以及
- LRU，[0-2]路用于指令提取，[3]路用于数据提取。

作为应用示例：如果指令提取和数据引用正在访问 flash 存储器，则控制可用于将指令提取、数据引用或两者发送到高速缓存或单式缓冲器。同样，可以为任一类型的访问使能或禁用预测。如果已缓存指令提取和数据引用，则可能会通过多种方式在指令提取与数据引用之间划分高速缓存的路径资源。

29.5.3 预测式读取

FMC 拥有单个缓存区，如有闲置周期，则将提前读取 Flash 存储器中的下一个字。对于使用 PFB0CR 的 B1DPE 和 B1IPE 字段的指令和/或数据访问的每个存储阵列，可对推测预取进行编程。由于许多代码访问都是连续的，在大多数情况下，使用推测预取缓冲器可提高性能。

如果已使能推测式读取，在完成读取后，FMC 将立即请求下一个连续地址。当访问连续代码和/或数据时，通过立即请求下一个字，推测式读取可帮助减少甚至消除等待状态。

例如，考虑以下场景：

- 假设某个系统的内核与 flash 时钟比为 4:1，并且已使能推测式读取。
- 内核以一个接一个的请求的方式，请求四个连续的长字，这意味着除了停止等待返回 flash 内存器数据之外，没有内核周期延迟。
- 缓存器或推测缓冲器内尚未存储任何数据。

在此场景中，访问四个长字的事件顺序如下所示：

1. 第一个长字读取需要 4 至 7 个内核时钟。更多详情，请参阅[等待状态](#)。
2. 由于 flash 存储器采用 64 位数据总线，FMC 内已有可用数据，第二个长字读取只需 1 个内核时钟。当第二个长字数据返回内核时，FMC 也将开始从 flash 存储器读取第三个和第四个长字。
3. 访问第三个长字需要 3 个内核时钟周期。flash 存储器读取自身需要 4 个时钟，但第一个时钟将与第二个长字读取重叠。

4. 由于采用 64 位 flash 存储器数据总线，读取第四个长字（与第二个长字一样）只需要 1 个时钟。

29.5.4 Flash 访问控制(FAC)功能

Flash 访问控制(FAC)是一个可配置的存储器优化保护方案，其可让终端客户，在所提供的库有可编程限制的时候，也可以使用软件库。Flash 存储器分为大小相等的区段，可为专有软件库提供保护。这些区段的保护类型是可控的：针对每一次片上 flash 存储器的数据传输，FAC 会按周期评估其访问权限。两级供应商可以将其专有软件添加到器件；每个级别的 FAC 区段保护都使用 PGMONCE 命令定义一次。

Flash 访问控制符合受 ARM Cortex-M 系列产品支持的 3 种特权级别要求。

- 大多数安全状态均为管理员/特权安全：允许仅执行并提供仅管理员访问控制。
- 中级状态为仅执行。
- 在非安全状态下，未置位任何访问控制状态。

特性：

- 片上 flash 存储器的轻量级访问控制逻辑
- Flash 地址空间分为 (32 或 64 个) 大小相等的区段 (区段大小定义为 flash_size [字节]/ (32 或 64))
- 每个区段的仅管理员访问和仅执行访问的单独控制位
- 在每个总线周期内评估的路由至 flash 的访问控制
- 访问违规错误将终止总线周期并将读取数据返回为零
- 编程模型允许 2 级受保护区段

29.5.4.1 存储器映射和寄存器定义

下表显示 FAC 寄存器映射情况。随后是各寄存器的描述及其位分配情况。

- Flash 管理单元(FMU)支持通过 32 位从机外设总线连接访问其 FAC 编程模型。
- 未生效的寄存器位读为零。
- 对于仅支持 32 个段的实施方案，只实现 32 位“低”寄存器。
- 对任何只读或保留寄存器的写操作均被忽略；如果试图访问偏移量“2B”以上的 Flash 寄存器空间，会产生传输错误。
- 术语管理员和用户模式等效于特权和非特权模式。
- 在关于 FAC 的本节中， n 指段编号， x 为寄存器所在模块（有时因器件而异）的缩略语。

x 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
18	只执行访问寄存器高位 (x_XACCH)	32	不适用	参见章节	29.5.4.1.1/ 539
1C	只执行访问寄存器低位 (x_XACCL)	32	不适用	参见章节	29.5.4.1.2/ 540
20	仅管理员访问寄存器高位 (x_SACCH)	32	不适用	参见章节	29.5.4.1.3/ 540
24	仅管理员访问寄存器低位 (x_SACCL)	32	不适用	参见章节	29.5.4.1.4/ 541
28	配置寄存器 (x_CR)	32	不适用	参见章节	29.5.4.1.5/ 541

29.5.4.1.1 只执行访问寄存器高位 (x_XACCH)

只执行访问寄存器是一种实现为两个 32 位寄存器的 64 位寄存器。

- 只执行访问高位 (63-32 段) 包含于 x_XACCH 中。
- 只执行访问低位 (31-0 段) 包含于 x_XACCL 中。

x_XACC{H,L} 寄存器为 flash 段提供位映射, 以便对各个关联段进行数据读取或只执行操作或者同时进行数据和指令获取两种操作。根据定义, 只执行访问包括指令获取或者来自处理器的 PC 相关数据的加载。

在复位序列中, XACC 寄存器中将加载来自 Flash 中非易失性空间的一个预配置值。有关 NVM 特性的更多信息, 请参见功能描述。NVM 位置的任何变动都会在下一次系统复位时生效。Flash 的这些值的基础值在复位值中用 x 表示。

地址: 0h 基准 + 18h 偏移 = 18h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	XA[63:32]																															
W	XA[63:32]																															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	

* 注:

- 预配置 flash 值 x = 复位时未定义。

x_XACCH 字段描述

字段	描述
XA[63:32]	63-32 段的只执行访问控制
0	只能在执行模式下访问相关段 (作为指令获取)
1	作为数据或在执行模式下访问相关段

29.5.4.1.2 只执行访问寄存器低位 (x_XACCL)

XACC{H,L}寄存器为 flash 段提供位映射，以便对各个关联段进行数据读取或只执行操作或者同时进行数据和指令获取两种操作。XACCCH 覆盖 63-32 段，XACCL 覆盖 31-0 段。根据定义，只执行访问包括指令获取或者来自处理器的 PC 相关数据的加载。

在复位序列中，XACC 寄存器中将加载来自 Flash 中非易失性空间的一个预配置值。有关 NVM 特性的更多信息，请参见功能描述。NVM 位置的任何变动都会在下一次系统复位时生效。Flash 的这些值的基础值在复位值中用 x 表示。

地址: 0h 基准 + 1Ch 偏移 = 1Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	XA[31:0]																															
W	XA[31:0]																															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

*注:

- 预配置 flash 值 x = 复位时未定义。

x_XACCL 字段描述

字段	描述
XA[31:0]	31-0 段的只执行访问控制
0	只能在执行模式下访问相关段 (作为指令获取)
1	作为数据或在执行模式下访问相关段

29.5.4.1.3 仅管理员访问寄存器高位 (x_SACCH)

仅管理员访问寄存器是一种实现为两个 32 位寄存器的 64 位寄存器。

- 仅管理员访问高位 (63-32 段) 包含于 x_SACCH 中。
- 仅管理员访问低位 (31-0 段) 包含于 x_SACCL 中。

x_SACC{H,L}为 flash 段提供位映射，以便对关联段进行仅管理员或用户以及管理员访问操作。

在复位序列中，SACC 寄存器中将加载来自 Flash 中非易失性空间的一个预配置值。有关 NVM 特性的更多信息，请参见功能描述。NVM 位置的任何变动都会在下一次系统复位时生效。Flash 的这些值的基础值在复位值中用 x 表示。

地址: 0h 基准 + 20h 偏移 = 20h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SA[63:32]																															
W	SA[63:32]																															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	

*注:

- 预配置 flash 值 x = 复位时未定义。

x_SACCH 字段描述

字段	描述
SA[63:32]	63-32 段的管理员访问控制 0 只能在管理员模式下访问相关段 1 可在用户或管理员模式下访问相关段

29.5.4.1.4 仅管理员访问寄存器低位 (x_SACCL)

SACC{H,L}寄存器为 flash 段提供位映射，以便对关联段进行仅管理员或用户以及管理员访问操作。SACCH 覆盖 63-32 段，SACCL 覆盖 31-0 段。

在复位序列中，SACC 寄存器中将加载来自 Flash 中非易失性空间的一个预配置值。有关 NVM 特性的更多信息，请参见功能描述。NVM 位置的任何变动都会在下一次系统复位时生效。Flash 的这些值的基础值在复位值中用 x 表示。

地址: 0h 基准 + 24h 偏移 = 24h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SA[31:0]																															
W	SA[31:0]																															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	

* 注:

- 预配置 flash 值 x = 复位时未定义。

x_SACCL 字段描述

字段	描述
SA[31:0]	31-0 段的管理员访问 0 只能在管理员模式下访问相关段 1 可在用户或管理员模式下访问相关段

29.5.4.1.5 配置寄存器 (x_CR)

FAC 配置寄存器提供基本配置信息，包括 flash 段大小和段划分指示器。

器件的 NUMSG 和 SGSIZE 值是固定的。这些值的芯片特性基础在复位值中用*表示。

地址: 0h 基准 + 28h 偏移 = 28h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NUMSG								0								SGSIZE															
W	NUMSG																SGSIZE															
复位	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*

* 注:

- NUMSG 字段: 器件指定值。
- SGSIZE 字段: 器件指定值。

x_CR 字段描述

字段	描述																																	
31-24 NUMSG	<p>段数量指示器</p> <p>NUMSG 字段指示 Flash 中大小相等段的数量。</p> <p>0x20 Flash 被分为 32 个段 0x40 Flash 被分为 64 个段</p>																																	
23-8 保留	<p>此字段为保留字段。</p> <p>此只读字段为保留字段且值始终为 0。</p>																																	
SGSIZE	<p>段大小</p> <p>段大小是一个固定值，为可用 flash 大小（舍入至最近似的 2 的幂）除以 32 或 64 所得的商，具体取决于可用程序 flash 量。此字段确定要使用地址中的哪些位来索引至 x_SACC 和 x_XACC 位图，以便获相应的权限标志位。段大小由等式 $2^{(8 + SGSIZE[7:0])}$ 定义。下表所示为可能设置示例。</p> <table border="1"> <thead> <tr> <th>Flash 大小</th> <th>段大小</th> <th>段大小编码</th> </tr> </thead> <tbody> <tr> <td colspan="3" style="text-align: center;">32 段编码</td> </tr> <tr> <td>16 KB</td> <td>16 KB/32 = 512 B</td> <td>0x1</td> </tr> <tr> <td>32 KB</td> <td>32 KB/32 = 1 KB</td> <td>0x2</td> </tr> <tr> <td>64 KB</td> <td>64 KB/32 = 2 KB</td> <td>0x3</td> </tr> <tr> <td>128 KB</td> <td>128 KB/32 = 4 KB</td> <td>0x4</td> </tr> <tr> <td colspan="3" style="text-align: center;">64 段编码</td> </tr> <tr> <td>256 KB</td> <td>256 KB/64 = 4 KB</td> <td>0x4</td> </tr> <tr> <td>512 KB</td> <td>512 KB/64 = 8 KB</td> <td>0x5</td> </tr> <tr> <td>1 MB</td> <td>1 MB/64 = 16 KB</td> <td>0x6</td> </tr> <tr> <td>2 MBytes</td> <td>2 MBytes/64 = 32 KBytes</td> <td>0x7</td> </tr> </tbody> </table>	Flash 大小	段大小	段大小编码	32 段编码			16 KB	16 KB/32 = 512 B	0x1	32 KB	32 KB/32 = 1 KB	0x2	64 KB	64 KB/32 = 2 KB	0x3	128 KB	128 KB/32 = 4 KB	0x4	64 段编码			256 KB	256 KB/64 = 4 KB	0x4	512 KB	512 KB/64 = 8 KB	0x5	1 MB	1 MB/64 = 16 KB	0x6	2 MBytes	2 MBytes/64 = 32 KBytes	0x7
Flash 大小	段大小	段大小编码																																
32 段编码																																		
16 KB	16 KB/32 = 512 B	0x1																																
32 KB	32 KB/32 = 1 KB	0x2																																
64 KB	64 KB/32 = 2 KB	0x3																																
128 KB	128 KB/32 = 4 KB	0x4																																
64 段编码																																		
256 KB	256 KB/64 = 4 KB	0x4																																
512 KB	512 KB/64 = 8 KB	0x5																																
1 MB	1 MB/64 = 16 KB	0x6																																
2 MBytes	2 MBytes/64 = 32 KBytes	0x7																																

29.5.4.2 FAC 功能说明

在 SoC 中，访问控制功能通过 2 个独立的数据块实现。Flash 管理单元(FMU)包括非易失性配置信息，这些信息在复位过程中取回并发回平台，以便控制正常工作过程中对 Flash 阵列的访问。

有(4)个 64 位 NVM 存储位置，用于支持访问控制功能。这些 NVM 位置的情况总结如下表。

表 29-121. NVM 位置

NVM 位置	说明	
NVSACC1, NVSACC2	两个位置通过 AND 运算相结合，并在复位过程中载入 x_SACC 寄存器，以提供访问配置功能。	对于管理员和用户访问控制，该段提供了基于段的仅管理员访问
NVXACC1, NVXACC2	两个位置通过 AND 运算相结合，并在复位过程中载入 x_XACC 寄存器，以提供访问配置功能。	对于数据和执行，该段提供了基于段的仅执行方式

这些 NVM 位置中的每一个都可通过一次性写入 Flash 命令进行写入，且只能写入一次。这些 NVM 位置不受“擦除所有块”的 Flash 命令和调试接口启动的批量擦除操作影响。由于这 2 个 NVXACCx 字段已经过 AND 运算，所以，访问保护只能增加。段的访问控制可从数据读和执行($XAn = 1$)更改为仅执行($XAn = 0$)，或者从管理员和用户模式($SAn = 1$)更改为仅管理员模式($SAn = 0$)。

在复位过程中，flash 会更早地退出复位状态，而此时，内核还继续保持在复位状态。FMU 捕捉内部寄存器中的 NVM 访问控制信息。FMU 对多个仅执行字段进行 AND 运算，以创建单一的仅执行字段。驱动至平台的该仅执行字段在内核退出复位之前不变。仅管理员字段的处理方式与此相同。

FMU 包括为 Flash 地址空间提供控制访问的 FAC 寄存器。在每次尝试进行 Flash 传输的寻址阶段，会检查管理员访问位(SAn)和执行访问位(XAn)以决定允许还是拒绝访问。如果拒绝访问，则会中止访问并返回总线错误；读取的数据为全 0。

下表所示为相对于 Flash 位置的段分配情况。

表 29-122. Flash 保护范围

SAn 和 XAn 位	受保护的段地址范围	段大小(总 Flash 的小数)
64 段编码		
0	$0x0_0000_0000 - (Flash_size/64-1)$	1/64
1	$(Flash_size/64) - 2*(Flash_size/64-1)$	1/64
.....		
63	$63*(Flash_size/64) - 62*(Flash_size/64-1)$	1/64
32 段编码		
0	$0x0_0000_0000 - (Flash_size/32-1)$	1/32
1	$(Flash_size/32) - 2*(Flash_size/32-1)$	1/32
.....		
31	$31*(Flash_size/32) - 30*(Flash_size/32-1)$	1/32

Flash 存储器中的各段可以独立保护，防止用户访问和数据访问。保护由 x_SACC 和 x_XACC 寄存器中的各个位控制，如下图所示。

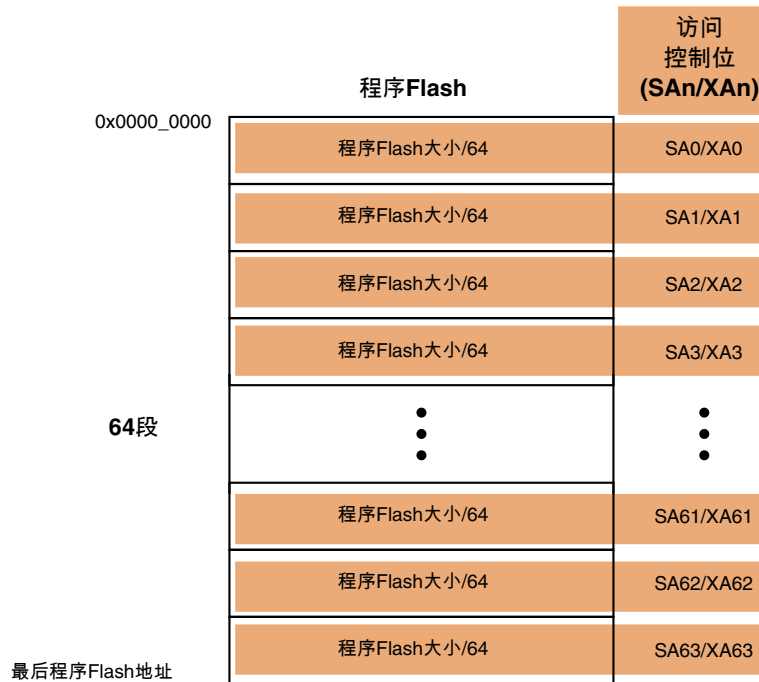


图 29-117. 程序 Flash 保护（64 段）

29.5.4.2.1 接口信号

表 29-123. 接口信号

信号	宽度	从	到	说明
xacc	64 或 32	FMU	平台	直接 xacc（仅执行访问控制）寄存器
sacc	64 或 32	FMU	平台	直接 sacc（管理员访问控制）寄存器
numsg	8	FMU	平台	NUMSG 位字段——段的二进制编码数 64 个段为 0x40 32 个段为 0x20
fac_enable	1	SIM	FMU	SIM 选项位——源于 IFR 位且在 SIM_SOPTx 中捕捉。针对无此功能的虚拟设备，这是禁用 Flash 访问控制的一种方式。 fac_enable==1——访问控制功能已使能 fac_enable==0——访问控制功能已禁用 <ul style="list-style-type: none"> 在复位序列中，XACC 寄存器写为全"1"。 在复位序列中，SACC 寄存器写为全"1"。 关闭基于 XACC 寄存器的隐性保护功能。

29.5.4.2.2 Flash 命令影响

写长字/字段/节	如果目标 Flash 位于只执行保护段内，除非完成执行“读取 1s 所有数据块”命令，并返回成功标志（表示该部分已完全擦除），则这些写命令是被禁止的。在完成执行“读取 1s 所有数据块”命令并返回成功标志之后，受保护段可接受写命令。要再次关闭只执行空间的写能力，则必须复位器件否则执行“读取 1s 所有数据块”命令，结果返回失败。如果试图去对不接受写命令的受保护段进行写操作，则会导致保护违规标志。
PGMCHK	FMU 不会对已配置为只执行的段码执行 PGMCHK。如果尝试对只执行地址执行 PGMCHK 命令，则 Flash 保护违规标志会置位。
擦除 Flash 扇区	如果目标 flash 扇区处于只执行保护段之内，则“擦除 Flash 扇区”命令会不被允许执行，并置保护违规标志。擦除受保护空间的唯一方式是执行“擦除所有”操作。
ERSALL	“擦除所有数据块”命令不受访问控制的影响。“擦除所有数据块”命令会擦除已写进任何只执行段的任何库。“擦除所有数据块”命令执行时，不会擦除配置的只执行分配，访问控制区保持以前配置的状态不变。 注：ERSALL 命令可用于字段更新。访问控制状态保持配置状态不变。软件必须进行相应的规划，可能腾出额外的空间以供将来使用。
SWAP	向 SWAP 命令添加了一条新的控制，以禁用 SWAP 特性。 IFR SWAP 字段和 SWAP 指示器在“擦除所有数据块”命令执行时被擦除，导致 SWAP 系统处于未初始化状态。必须使用初始化代码来运行 SWAP 命令，这样才能置位 SWAP 指示器地址并初始化 SWAP 系统。 禁用之后，如果不执行“擦除所有”命令，则无法使能 SWAP。“擦除所有”命令会擦除 flash 阵列中的预加载代码和库，并将 SWAP 系统复位至未初始化状态，但会使访问控制保持以前配置的状态不变。如果要在访问控制保护中禁用 SWAP，则必须在“擦除所有数据块”命令执行之后恢复已禁用的设置。
ERSXA	命令的目标区域由 FAC 控制。对于 ERSXA，如果 flash 扇区处于只执行受保护段中且不受 flash 保护（FPROT），则会被擦除。
RD1XA	命令的目标区域由 FAC 控制。对于 RD1XA，如果 flash 扇区处于只执行受保护段中，则会进行擦除验证。

29.5.4.2.3 内核平台影响

平台内核高速缓存（Flash 和 LMEM 高速缓存）	如果任何段被标记为仅执行，则该高速缓存对用户隐藏。标签为只读且无法写入，并且数据高速缓存也无法读写。针对标签和数据阵列的写操作将会被忽略，针对数据阵列的读操作返回零。这些会影响调试断点。详见调试部分。
调试	调试器是一个非处理器总线主设备，在仅执行区域不能步进、跟踪或断点。在仅管理员模式下，调试器被限制不能更改模式。对标记为仅执行的 flash 空间的任何段进行调试访问，会导致总线错误。
PC 相关寻址	对于 PC 相关寻址问题，我们目前的了解尚不深入，本节将来会适时更新。 将允许 PC 相关重新进入仅执行段..... 将就 PC 相关寻址对软件进行限制，因为硬件不能确定 PC 相关数据引用是否跨越了段边界。 <ul style="list-style-type: none"> • 如果 ifetch 在受保护段中执行，则允许数据引用。 • 硬件无法跨边界跟踪推测式 ifetch。
中断	如果通过函数调用来移进仅执行段，则在使用典型的软件控制时（即在执行新代码之前保存寄存器和状态），硬件可以进行跟踪。
复位向量	在 ARM 内核中，复位向量获取为超级管理员模式的操作，如果复位向量位于标记为仅执行段中，这会导致问题。在复位后、第一次有效指令获取之前，用额外的逻辑来支持超级管理员模式的代码对仅可执行代码空间的读取操作。在第一次有效指令获取之后，FAC 逻辑遵循正常检查规则。

29.5.4.2.4 软件影响

随着实现、检验和验证的继续，我们需要把更多的软件影响告诉工具和库供应商。硬件无法看到 ARM 内核的所有状态，对于健壮的方案所需要的硬件以及其驱动软件的限制，硬件也无法跟踪其软件流程。

- 任何标志为仅可执行的代码段都能看到系统中的所有代码。这意味着，一个仅可执行段的代码能读取另一个段中的仅执行代码。因此，如果恩智浦把预加载的代码发给另一家供应商，则该供应商将获得恩智浦代码的访问权限。我们可能会用保密协议和法律协议来处理这个问题。
- 对于单次预载情形（例如，如果恩智浦针对一个 GP 市场预载代码，或者一家供应商向空白器件中预加载专有代码），必须编写两级访问控制，以保护预加载的代码。
- 如果受保护段有任何部分未被预加载代码使用，则应在编程时为其（受保护段中未被预加载代码使用的任何部分）加上 NOP 标识，以防止黑客在该段中写入其他代码。

29.5.4.2.5 访问检查评估

针对每一个和片上 Flash 存储器相关的数据传输，Flash 控制器(FAC)提供了以周期评估的访问权限

整个 flash 存储容量分为大小相等的段。对于每段，两个寄存器包括一个仅限管理员访问控制指示器和一个仅执行访问控制指示器。

FAC 逻辑用来自总线周期的参考地址和一个 2 位属性（或“保护”字节）作为输入，加上编程模型寄存器的内容，执行所需访问控制评估。

下列代码示例展示了用于 FAC 评估的 C 代码：

```
unsigned long long sacc; // supervisor-only map
unsigned long long xacc; // execute-only map
unsigned int seg_size; // 8-bit segment size
unsigned int fac_error;

fac_evaluation (addr, prot)
    unsigned int addr; // access address
    unsigned int hprot; // encoded 2-bit "protection" field {supv, data}
{
    unsigned int sacc_flag; // sacc flag for this segment
    unsigned int xacc_flag; // xacc flag for this segment
    unsigned int i; // segment index

    i = (addr >> (8 + seg_size & 0x0f)) & 0x3f; // form 6-bit segment index
    sacc_flag = (sacc >> i) & 1; // extract sacc bit for this segment
    xacc_flag = (xacc >> i) & 1; // extract xacc bit for this segment

    // create a 4-tuple concatenating the 2-bit protection field + {sacc, xacc} flags

    switch ((hprot & 3) << 2 | (sacc_flag << 1) | xacc_flag) {
        // all these combinations are allowed accesses
        case 0x2: // {user, ifetch} && {supv+user, ifetch-only}
        case 0x3: // {user, ifetch} && {supv+user, ifetch+data}
```

```

case 0x7: // {user, data}    && {supv+user, ifetch+data}
case 0x8: // {supv, ifetch} && {supv-only, ifetch-only}
case 0x9: // {supv, ifetch} && {supv-only, ifetch+data}
case 0xa: // {supv, ifetch} && {supv+user, ifetch-only}
case 0xb: // {supv, ifetch} && {supv+user, ifetch+data}
case 0xd: // {supv, data}   && {supv-only, ifetch+data}
case 0xf: // {supv, data}   && {supv+user, ifetch+data}
    fac_error = 00;
    break;

// all these combinations are unallowed, that is, errored accesses
case 0x0: // {user, ifetch} && {supv-only, ifetch-only}
case 0x1: // {user, ifetch} && {supv-only, ifetch+data}
case 0x4: // {user, data}   && {supv-only, ifetch-only}
case 0x5: // {user, data}   && {supv-only, ifetch+data}
case 0x6: // {user, data}   && {supv+user, ifetch-only}
case 0xc: // {supv, data}   && {supv-only, ifetch-only}
case 0xe: // {supv, data}   && {supv+user, ifetch-only}
    fac_error = 1;
    break;

} // switch()
} // fac_evaluation()

```

29.5.4.2.6 FAC 应用技巧

在一种使用方式中，NVSACC1 和 NVXACC1 位置由恩智浦写入，它们负责保护已写入器件中关联 flash 段的恩智浦库。其后，第三方供应商如果希望编写其专有软件，扩展受保护的 flash 段以纳入其软件库，然后再将其提供给客户，就可以选择对 NVSACC2 和 NVXACC2 NVM 位置进行写入。

在此基础上，他们的客户可以用现有库开发自己的代码，并把这些代码编进剩余的可用片上 flash 空间中。器件将以标准安全特性继续为最终用户提供支持，进一步限制对 flash 资源的外部访问。

SWAP: 如果 flash 阵列的两半中均有仅可执行的代码的镜像，则可安全地使能 SWAP；否则，SWAP 应禁用，因为在 SWAP 期间，硬件不会跟踪访问控制寻址。

29.6 初始化和应用信息

FMC 无需用户初始化。Flash 加速功能默认已使能。

FMC 无法查看 flash 存储器擦除和编程周期，因为 Flash 存储器模块对其进行直接管理。因此，如果应用程序正在执行 flash 存储器命令，则需要禁用和/或刷新 FMC 高速缓存，以防止返回过期的数据。通过这种方式，使用 PFB0CR[CINV_WAY]字段使缓存无效。

第 30 章

Flash 存储器模块(FTFA)

30.1 简介

Flash 存储器模块包括以下可访问的存储器区域：

- 用于向量空间和代码存储的可程序 Flash 存储器

Flash 存储器非常适合单电源应用，允许字段中擦除和重编程操作，无需任何外部高电压电源。

Flash 存储器模块包含一个存储器控制器，用于执行命令以修改 Flash 存储器内容。已擦除的位读取为“1”，已编程的位读取为“0”。编程操作是单向的；它只能将位从“1”（已擦除）状态移至“0”状态（已编程）。只有擦除操作能将位从“0”恢复到“1”；不能将位从“0”编程为“1”。

警告

在编程之前，Flash 存储器位置必须处于已擦除状态。不允许在某个 Flash 存储器位置进行位的累计写操作（中间没有擦除操作的连续写操作）。不允许将现有的 0 重新编程为 0，因为这样会加重器件负担。

Flash 存储器的标准发货条件已被擦除且安全性已禁用。由于已擦除（“1”）状态和/或已编程（“0”）状态的降级，可能会随着时间的推移出现数据丢失现象。因此，建议在出厂编程之前立即对每个 Flash 数据块或扇区进行重新擦除，以确保实现充分的数据保留能力。

30.1.1 特性

Flash 存储器模块具有以下特性。

注

有关用户器件上的准确 flash 存储器数量，请参见器件的芯片配置详细信息。

30.1.1.1 程序 Flash 存储器功能

- 扇区大小为 2 KB
- 程序 Flash 保护方案可防止意外编程或擦除存储的数据
- 程序 flash 访问控制方案可防止未经授权访问选定代码段
- 内置带有校验功能的自动化编程和擦除算法

30.1.1.2 其他 Flash 存储器模块特性

- 用于 flash 存储器编程和擦除操作的内部高压生成器
- 可选择在 Flash 指令执行完成时产生中断
- 支持 MCU 加密机制，可阻止未经授权访问 Flash 存储器中的内容

30.1.2 框图

下图是 flash 存储器模块的框图。

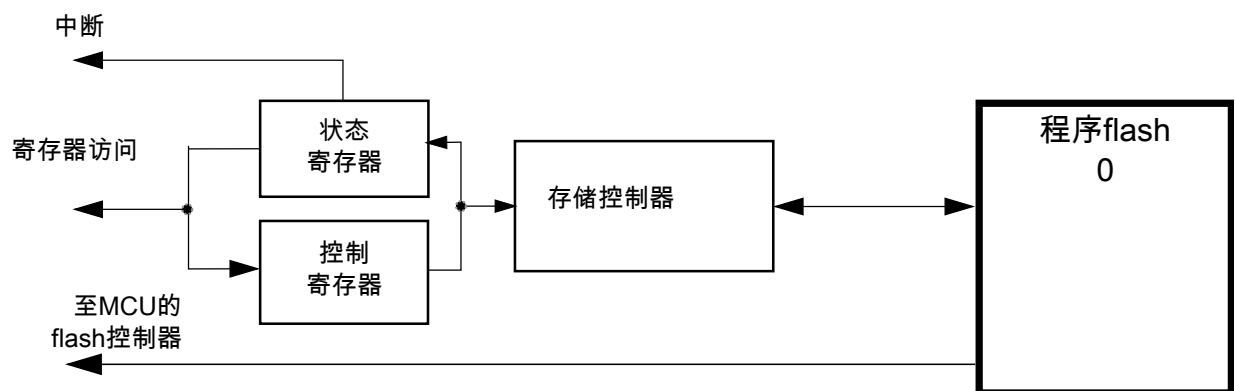


图 30-1. Flash 框图

30.1.3 术语表

命令写入序列 — 一系列 MCU 对 flash FCCOB 寄存器组执行写入操作，后者用于发起和控制内置于 flash 存储器模块的 flash 算法执行。

耐受能力 — 可对 flash 存储器位置进行擦除和重新编程的次数。

FCCOB (Flash 通用命令对象) — 一组 flash 寄存器，用于将命令、地址、数据和任何相关参数传递至 flash 存储器模块中的存储器控制器。

Flash 块 — flash 存储器模块内的大块空间，用于提供非易失性存储器存储。

Flash 存储器模块 — 所有 flash 块加上一个 flash 管理单元可提供高级控制、MCU 总线连接。

HSRUN — 一种 MCU 功耗模式，它允许高速访问 flash 模块中的存储器资源。如果 MCU 处于 HSRUN 模式，则用户无权访问 flash 命令集。

IFR — 位于每个 flash 块内的非易失性信息寄存器，它独立于主存储器阵列。

长字 — 具有对齐长字的 32 位数据，其字节地址为[1:0] = 00。

NVM — 非易失性存储器。一种可在关机期间保留存储数据的存储器技术。flash 阵列是一种使用 NOR 型 flash 存储器技术的 NVM。

NVM 正常模式 — 一种 NVM 模式，它可提供对 flash 存储器模块资源的基本用户访问。CPU 或其他总线主机通过写入 flash 存储器模块中的 FCCOB 寄存器组，发起 flash 编程和擦除操作（或其他 flash 命令）。

NVM 特殊模式 — 一种 NVM 模式，它允许对 flash 存储器模块中的存储器资源进行外部、片外访问。如果 MCU 已加密，则可用简化的 flash 命令集。有关何时使用此模式，请参见芯片配置详细信息。

字段 — 具有已对齐字段的 64 位数据，其字节地址为[2:0] = 000。

程序 flash — 程序 flash 存储器提供用于存储向量和代码的非易失性存储。

程序 flash 扇区 — 可擦除的程序 flash 存储器（连续地址）的最小单元。

保留 — 可在 NVM 中保留数据而读出时不会出错的时长。由于擦除(1)状态受降级影响（与编程(0)状态一样），因此上一次擦除操作（而不是编程时间）可能会达到数据保留限制。

RWW — 写入时读取。当命令操作在某个存储器资源中处于活动状态时能够同时从另一个存储器资源进行读取。

加密 — 一种 MCU 状态，它可传送到 flash 存储器模块，如此器件的芯片配置详细信息中所述。在加密状态下，读取和更改 NVM 内容将会受到限制。

字 — 具有对齐字的 16 位数据，其字节地址为[0] = 0。

30.2 外部信号说明

Flash 存储器模块不包含连接终端用户片外的。

30.3 存储器映射和寄存器

本节将介绍 flash 存储器模块的存储器映射和寄存器。

未定义从 flash 存储器模块中的未生效存储器空间数据读取。对 flash 存储器模块内的未生效或保留的存储器空间（寄存器）执行写入操作将被忽略。

30.3.1 Flash 配置字段说明

该程序 flash 存储器包含一个 16 字节 flash 配置字段，用于存储默认保护设置（复位时加载）和允许 MCU 限制访问 flash 存储器模块的安全信息。

Flash 配置字段偏移地址	大小（字节）	字段说明
0x0_0400–0x0_0407	8	后门比较密钥。请参阅 验证后门访问密钥命令 和 使用后门密钥解密芯片 。
0x0_0408–0x0_040B	4	程序 flash 保护字节。请参阅程序 Flash 保护寄存器(FPROT0-3)说明。
0x0_040F	1	保留
0x0_040E	1	保留
0x0_040D	1	Flash 非易失性选项字节。请参阅 Flash 选项寄存器(FOPT)说明。
0x0_040C	1	Flash 安全字节。请参阅 Flash 安全寄存器(FSEC)说明。

30.3.2 程序 Flash IFR 映射

程序 flash IFR 是一种非易失性信息存储器，用户可自由读取，但没有擦除功能，并且编程功能受限（参见[一次性读取命令](#)、[一次性编程命令](#)和[读取资源命令](#)中的一次性读取、一次性编程和读取资源命令）。

下表概述了程序 flash IFR 的内容，在后续章节中会进一步介绍。

程序 flash IFR 位于只包含程序 flash 器件的程序 flash 0 存储器块

地址范围	大小 (字节)	字段说明
0x00 – 0x9F	160	保留
0xA0 – 0xA3	4	一次性编程 XACCH-1 字段 (索引 = 0x10)
0xA4 – 0xA7	4	一次性编程 XACCL-1 字段 (索引 = 0x10)
0xAB	4	一次性编程 XACCH-2 字段 (索引 = 0x11)
0xAC – 0xAF	4	一次性编程 XACCL-2 字段 (索引 = 0x11)
0xB0 – 0xB3	4	一次性编程 SACCH-1 字段 (索引 = 0x12)
0xB4 – 0xB7	4	一次性编程 SACCL-1 字段 (索引 = 0x12)
0xB8 – 0xBB	4	一次性编程 SACCH-2 字段 (索引 = 0x13)
0xBC – 0xBF	4	一次性编程 SACCL-2 字段 (索引 = 0x13)
0xFF	64	一次性编程 ID 字段 (索引 = 0x00 - 0x0F)

30.3.2.1 一次性编程 字段

程序 flash IFR 中的一次性编程字段可提供独立于程序 flash 主阵列的 96 个字节的用户数据存储。用户可以只编程一次性编程字段一次，因为对用户来说无程序 flash IFR 擦除机制。一次性编程字段可读取任意次数。可使用一次性读取和一次性编程命令在 4 字节或 8 字节记录中访问本部分程序 flash IFR (参见[一次性读取命令](#)和[一次性编程命令](#))。

30.3.3 寄存器说明

flash 存储器模块包含一组存储器映射控制和状态寄存器。

注

当命令正在运行(FSTAT[CCIF]=0)时，除 FCNFG 和 FSTAT 以外的任何寄存器都不接受寄存器写入。在启动复位序列期间、在 CCIF 首次上升之前会放宽无写入规则。在此初始化期间，用户可以写入任何寄存器。当擦除挂起请求激活

(FCNFG[ERSSUSP]=1)时，所有寄存器写入都会禁用（寄存器 FCNFG 和 FSTAT 除外）。

FTFA 存储器映射

绝对地址（十六进制）	寄存器名称	宽度（单位：位）	访问	复位值	小节/页
4002_0000	Flash 状态寄存器 (FTFA_FSTAT)	8	R/W	00h	30.3.3.1/555
4002_0001	Flash 配置寄存器 (FTFA_FCNFG)	8	R/W	00h	30.3.3.2/557
4002_0002	Flash 安全寄存器 (FTFA_FSEC)	8	R	未定义	30.3.3.3/558
4002_0003	Flash 选项寄存器 (FTFA_FOPT)	8	R	未定义	30.3.3.4/559
4002_0004	Flash 通用命令对象寄存器 (FTFA_FCCOB3)	8	R/W	00h	30.3.3.5/560
4002_0005	Flash 通用命令对象寄存器 (FTFA_FCCOB2)	8	R/W	00h	30.3.3.5/560
4002_0006	Flash 通用命令对象寄存器 (FTFA_FCCOB1)	8	R/W	00h	30.3.3.5/560
4002_0007	Flash 通用命令对象寄存器 (FTFA_FCCOB0)	8	R/W	00h	30.3.3.5/560
4002_0008	Flash 通用命令对象寄存器 (FTFA_FCCOB7)	8	R/W	00h	30.3.3.5/560
4002_0009	Flash 通用命令对象寄存器 (FTFA_FCCOB6)	8	R/W	00h	30.3.3.5/560
4002_000A	Flash 通用命令对象寄存器 (FTFA_FCCOB5)	8	R/W	00h	30.3.3.5/560
4002_000B	Flash 通用命令对象寄存器 (FTFA_FCCOB4)	8	R/W	00h	30.3.3.5/560
4002_000C	Flash 通用命令对象寄存器 (FTFA_FCCOBB)	8	R/W	00h	30.3.3.5/560
4002_000D	Flash 通用命令对象寄存器 (FTFA_FCCOBA)	8	R/W	00h	30.3.3.5/560
4002_000E	Flash 通用命令对象寄存器 (FTFA_FCCOB9)	8	R/W	00h	30.3.3.5/560
4002_000F	Flash 通用命令对象寄存器 (FTFA_FCCOB8)	8	R/W	00h	30.3.3.5/560
4002_0010	程序 Flash 保护寄存器 (FTFA_FPROT3)	8	R/W	未定义	30.3.3.6/561
4002_0011	程序 Flash 保护寄存器 (FTFA_FPROT2)	8	R/W	未定义	30.3.3.6/561
4002_0012	程序 Flash 保护寄存器 (FTFA_FPROT1)	8	R/W	未定义	30.3.3.6/561
4002_0013	程序 Flash 保护寄存器 (FTFA_FPROT0)	8	R/W	未定义	30.3.3.6/561
4002_0018	只执行访问寄存器 (FTFA_XACCH3)	8	R	未定义	30.3.3.7/562

下一页继续介绍此表...

FTFA 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4002_0019	只执行访问寄存器 (FTFA_XACCH2)	8	R	未定义	30.3.3.7/ 562
4002_001A	只执行访问寄存器 (FTFA_XACCH1)	8	R	未定义	30.3.3.7/ 562
4002_001B	只执行访问寄存器 (FTFA_XACCH0)	8	R	未定义	30.3.3.7/ 562
4002_001C	只执行访问寄存器 (FTFA_XACCL3)	8	R	未定义	30.3.3.7/ 562
4002_001D	只执行访问寄存器 (FTFA_XACCL2)	8	R	未定义	30.3.3.7/ 562
4002_001E	只执行访问寄存器 (FTFA_XACCL1)	8	R	未定义	30.3.3.7/ 562
4002_001F	只执行访问寄存器 (FTFA_XACCL0)	8	R	未定义	30.3.3.7/ 562
4002_0020	仅管理员访问寄存器 (FTFA_SACCH3)	8	R	未定义	30.3.3.8/ 563
4002_0021	仅管理员访问寄存器 (FTFA_SACCH2)	8	R	未定义	30.3.3.8/ 563
4002_0022	仅管理员访问寄存器 (FTFA_SACCH1)	8	R	未定义	30.3.3.8/ 563
4002_0023	仅管理员访问寄存器 (FTFA_SACCH0)	8	R	未定义	30.3.3.8/ 563
4002_0024	仅管理员访问寄存器 (FTFA_SACCL3)	8	R	未定义	30.3.3.8/ 563
4002_0025	仅管理员访问寄存器 (FTFA_SACCL2)	8	R	未定义	30.3.3.8/ 563
4002_0026	仅管理员访问寄存器 (FTFA_SACCL1)	8	R	未定义	30.3.3.8/ 563
4002_0027	仅管理员访问寄存器 (FTFA_SACCL0)	8	R	未定义	30.3.3.8/ 563
4002_0028	Flash 访问段大小寄存器 (FTFA_FACSS)	8	R	未定义	30.3.3.9/ 565
4002_002B	Flash 访问段数量寄存器 (FTFA_FACSN)	8	R	未定义	30.3.3.10/ 565

30.3.3.1 Flash 状态寄存器 (FTFA_FSTAT)

FSTAT 寄存器报告 flash 存储器模块的操作状态。

可对 CCIF、RDCOLERR、ACCERR 和 FPVIOL 位执行读取和写入操作。MGSTAT0 位为只读。未指定位读为 0 并且不可写入。

注

置位后，此寄存器中的访问错误(ACCERR)和 Flash 保护违反(FPVIOL)位将阻止启用任何其他命令，直到标志位清除(通过向其写入 1)。

地址: 4002_0000h 基准 + 0h 偏移 = 4002_0000h

位	7	6	5	4	3	2	1	0
读	CCIF	RDCOLERR	ACCERR	FPVIOL	0			MGSTAT0
写	w1c	w1c	w1c	w1c				
复位	0	0	0	0	0	0	0	0

FTFA_FSTAT 字段描述

字段	描述
7 CCIF	<p>命令完成中断标志</p> <p>指明 flash 命令已完成。将 1 写入 CCIF 可将 CCIF 标志清零以启动某个命令，CCIF 在命令完成或命令出错之前会一直保持低电平。</p> <p>CCIF 复位为 0，但在复位初始化序列结束时存储器控制器会将其置位为 1。用户可能会或无法看到 0 硬件复位值，具体取决于在复位释放后的读取速度。</p> <p>0 Flash 命令正在进行 1 Flash 命令已完成</p>
6 RDCOLERR	<p>Flash 读取冲突错误标志位</p> <p>指明 MCU 已尝试从正由 flash 命令(CCIF=0)操作的 flash 存储器资源读取。块仲裁逻辑将任何同步访问检测为冲突错误。在这种情况下，无法保证能够读取数据。通过向 RDCOLERR 位写入 1 可将其清除。向 RDCOLERR 写入 0 没有影响。</p> <p>0 未检测到冲突错误 1 检测到冲突错误</p>
5 ACCERR	<p>Flash 访问错误标志位</p> <p>指明 Flash 存储器资源遭遇了非法访问，其原因要么是违反了命令写入序列，要么是发出了非法 Flash 命令。ACCERR 置位时，无法通过清除 CCIF 标志来启动命令。CCIF 置位时，通过向 ACCERR 写入 1 可清除 ACCERR 位。向 ACCERR 位写入 0 没有影响。</p> <p>0 未检测到访问错误 1 检测到访问错误</p>
4 FPVIOL	<p>Flash 保护违反标志</p> <p>指明在命令写入序列期间，某个受保护的程序 Flash 存储器区域中的地址遭遇了试图对其进行编程或擦除的操作。FPVIOL 置位时，无法通过清除 CCIF 标志来启动命令。CCIF 置位时，通过向 FPVIOL 写入 1 可清除 FPVIOL 位。向 FPVIOL 位写入 0 没有影响。</p> <p>0 没有检测到保护违规情况 1 检测到保护违规情况</p>
3-1 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
0 MGSTAT0	<p>存储器控制器命令完成状态标志</p> <p>执行某个 Flash 命令期间或 Flash 复位序列期间，如果检测到错误，MGSTAT0 状态标志位将置位。作为状态标志位，此字段无法像此寄存器中的其他错误标志位一样由用户清除，也不需要清除。</p>

下一页继续介绍此表...

FTFA_FSTAT 字段描述 (继续)

字段	描述
	"command-N"的 MGSTAT0 位值仅在"command-N"执行结束 (如果 CCIF=1) 后并在下一条命令启动之前有效。在执行"command-N+1"的某个时间点, 先前结果将被放弃并且先前任何错误都将被清除。

30.3.3.2 Flash 配置寄存器 (FTFA_FCNFG)

此寄存器提供与 flash 存储器模块的当前功能状态相关的信息。

擦除控制位 (ERSAREQ 和 ERSSUSP) 具有写入限制。未指定位按注释读取并且不可写入。

地址: 4002_0000h 基准 + 1h 偏移 = 4002_0001h

位	7	6	5	4	3	2	1	0
读	CCIE	RDCOLLIE	ERSAREQ	ERSSUSP	0	0	0	0
写								
复位	0	0	0	0	0	0	0	0

FTFA_FCNFG 字段描述

字段	描述
7 CCIE	命令完成中断使能位 完成某个 flash 命令时控制中断生成。 0 命令完成中断已禁用 1 命令完成中断已使能当 FSTAT[CCIF]标志位置位时, 生成中断请求。
6 RDCOLLIE	读取冲突错误中断使能位 发生 flash 存储器读取冲突错误时控制中断生成。 0 读取冲突错误中断已禁用 1 读取冲突错误中断已使能。当检测到 flash 存储器读取冲突错误时, 生成中断请求 (参见 FSTAT[RDCOLERR]的说明)。
5 ERSAREQ	擦除所有请求 请求存储器控制器执行擦除所有块命令并解除加密状态。无法直接对 ERSAREQ 进行写操作, 但可由用户间接控制。有关如何请求此命令的信息, 请参阅器件的芯片配置详细信息。 当从 flash 存储器模块外部触发擦除所有请求时, ERSAREQ 将置位, 且 CCIF 置位(当前未执行任何命令)。当操作完成后, flash 存储器模块将清除 ERSAREQ。 0 无请求或请求完成 1 请求: 1. 运行擦除所有块模块命令, 2. 验证已擦除状态 3. 将 Flash 配置字段中的安全字节编程为解密状态, 并 4. 通过将 FSEC[SEC]字段设置为解密状态以解除 MCU 加密状态。
4 ERSSUSP	擦除暂停

下一页继续介绍此表...

FTFA_FCNFG 字段描述 (继续)

字段	描述
	允许用户在执行时暂停 (中断) 擦除 Flash 扇区命令。 0 未请求暂停 1 暂停执行当前的擦除 Flash 扇区命令。
3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
0 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

30.3.3.3 Flash 安全寄存器 (FTFA_FSEC)

此只读寄存器保留与 MCU 和 Flash 存储器模块的安全性相关的所有位。

在复位序列期间，会将程序 Flash 存储器中 Flash 配置字段内 Flash 安全字节的内容加载到此寄存器。Flash 的这些基础值在复位值中用 X 表示。

地址: 4002_0000h 基准 + 2h 偏移 = 4002_0002h

位	7	6	5	4	3	2	1	0
读	KEYEN		MEEN		FSLACC		SEC	
写								
复位	x*	x*	x*	x*	x*	x*	x*	x*

* 注:

- x = 复位时未定义。

FTFA_FSEC 字段描述

字段	描述
7-6 KEYEN	后门密钥安全使能 使能或禁止后门密钥对 flash 存储器模块的访问。 00 后门密钥访问已禁止 01 后门密钥访问已禁止 (禁止后门密钥访问的首选 KEYEN 状态) 10 后门密钥访问已使能 11 后门密钥访问已禁止
5-4 MEEN	批量擦除使能 在所有 NVM 模式下，始终使能和禁用 flash 存储器模块的批量擦除功能。在正常 NVM 以外的模式下，仅当 SEC 设为加密时，此字段状态才相关。当 SEC 设为解密时，MEEN 设置无任何影响。

下一页继续介绍此表...

FTFA_FSEC 字段描述 (继续)

字段	描述
	00 批量擦除已使能 01 批量擦除已使能 10 批量擦除已禁用 11 批量擦除已使能
3-2 FSLACC	工厂安全等级访问代码 允许或禁止在恩智浦返回器件故障分析中访问 flash 存储器内容。当 SEC 为加密且 FSLACC 遭拒时，访问程序 Flash 内容将遭拒，并且恩智浦工厂测试执行的任何故障分析必须先全部擦除，然后才可解密器件。 允许访问(SEC 解密或 SEC 加密且允许 FSLACC)时，恩智浦工厂测试可以查看当前 flash 内容。仅当 SEC 设为加密时，FSLACC 位的状态才相关。当 SEC 设为解密时，FSLACC 设置无任何影响。 00 允许恩智浦工厂访问 01 拒绝恩智浦工厂访问 10 拒绝恩智浦工厂访问 11 允许恩智浦工厂访问
SEC	Flash 安全 定义 MCU 的加密状态。在加密状态下，MCU 限制访问 flash 存储器模块资源。这些限制因器件而异，详情参见器件芯片配置详细信息。如果 Flash 存储器模块未利用后门密钥访问权限进行加密保护，则 SEC 字段强制为 10b。 00 MCU 加密状态为已加密。 01 MCU 加密状态为已加密。 10 MCU 加密状态为未加密。(Flash 存储器模块的标准运输条件为未加密。) 11 MCU 加密状态为已加密。

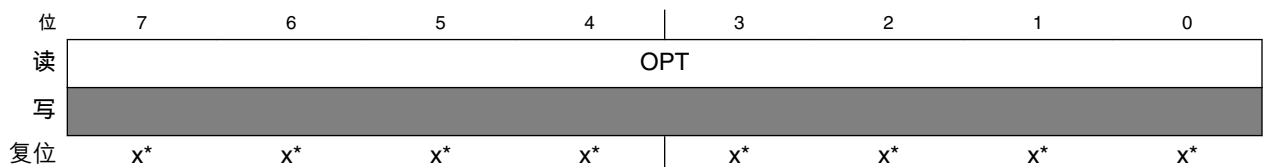
30.3.3.4 Flash 选项寄存器 (FTFA_FOPT)

Flash 选项寄存器允许 MCU 自定义其操作，方法是检查在复位时从 NVM 加载的只读位的状态。这些位的功能如器件芯片配置详细信息中所定义。

在正常 NVM 模式下，寄存器中的所有位均为只读，寄存器中的所有位均可读取和写入。

在复位序列期间，会从程序 Flash 存储器中 Flash 配置字段内的 Flash 非易失选项字节加载此寄存器。Flash 的这些基础值在复位值中用 X 表示。但是，如果 flash 非易失性选项字节为 0x00，则会向此寄存器写入 0xFF。

地址: 4002_0000h 基准 + 3h 偏移 = 4002_0003h



* 注:

- x = 复位时未定义。

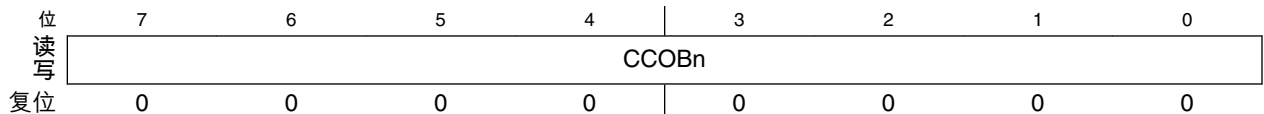
FTFA_FOPT 字段描述

字段	描述
OPT	非易失性选项 这些位将在复位时从 flash 加载到此寄存器。有关这些位的定义和使用，请参阅器件芯片配置详细信息。

30.3.3.5 Flash 通用命令对象寄存器 (FTFA_FCCOBn)

FCCOB 寄存器组为命令代码和参数提供 12 个字节。数据集内的各个字节都将 0-B 十六进制标识符附加于 FCCOB 寄存器名称：FCCOB0, FCCOB1, ..., FCCOBB。

Address: 4002_0000h base + 4h offset + (1d × i), where i=0d to 11d



FTFA_FCCOBn 字段描述

字段	描述																		
CCOBn	<p>FCCOB 寄存器为存储器控制器提供命令代码和相关参数。可按任何顺序写入组成 FCCOB 数据集的各个寄存器，但必须提供不同命令的所有必需值。首先，设置所有必填 FCCOB 字段，然后通过将 1 写入 STAT[CCIF]位来启动命令的执行。这会清除 CCIF 位，从而锁定所有 FCCOB 参数字段，用户无法对其进行更改，直至命令完成（CCIF 返回为 1）。不提供任何命令缓冲或队列；当前命令完成后才能加载下一条命令。</p> <p>一些命令将信息返回到 FCCOB 寄存器中。FSTAT[CCIF]标志由存储器控制器恢复为 1 之后，即可读取返回到 FCCOB 的任何值。</p> <p>下表显示通用 flash 命令格式。第一个 FCCOB 寄存器 FCCOB0 始终包含命令代码。此 8 位值定义要执行的命令。命令代码后跟此特定 flash 命令所需的参数，通常是地址和/或数据值。</p> <p>注：命令参数表根据 FCCOB 编号（它等同于字节数）进行编写。此编号是对 FCCOB 寄存器名称而不是寄存器地址的引用。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">FCCOB 编号</th> <th>典型命令参数内容[7:0]</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>FCMD（用于定义 flash 命令的代码）</td> </tr> <tr> <td>1</td> <td>Flash 地址[23:16]</td> </tr> <tr> <td>2</td> <td>Flash 地址[15:8]</td> </tr> <tr> <td>3</td> <td>Flash 地址[7:0]</td> </tr> <tr> <td>4</td> <td>数据字节 0</td> </tr> <tr> <td>5</td> <td>数据字节 1</td> </tr> <tr> <td>6</td> <td>数据字节 2</td> </tr> <tr> <td>7</td> <td>数据字节 3</td> </tr> </tbody> </table>	FCCOB 编号	典型命令参数内容[7:0]	0	FCMD（用于定义 flash 命令的代码）	1	Flash 地址[23:16]	2	Flash 地址[15:8]	3	Flash 地址[7:0]	4	数据字节 0	5	数据字节 1	6	数据字节 2	7	数据字节 3
FCCOB 编号	典型命令参数内容[7:0]																		
0	FCMD（用于定义 flash 命令的代码）																		
1	Flash 地址[23:16]																		
2	Flash 地址[15:8]																		
3	Flash 地址[7:0]																		
4	数据字节 0																		
5	数据字节 1																		
6	数据字节 2																		
7	数据字节 3																		

FTFA_FCCOBn 字段描述 (继续)

字段	描述	
	FCCOB 编号	典型命令参数内容[7:0]
	8	数据字节 4
	9	数据字节 5
	A	数据字节 6
	B	数据字节 7
<p>FCCOB 端字节序和多字节访问：</p> <p>FCCOB 寄存器组使用高字节序寻址约定。对于所有大于 1 字节的命令参数字段，最重要的数据位于具有最低编号的 FCCOB 寄存器。FCCOB 寄存器组可以单个字节、对齐字（2 字节）或对齐长字（4 字节）进行读取和写入。</p>		

30.3.3.6 程序 Flash 保护寄存器 (FTFA_FPROTn)

FPROT 寄存器定义了针对编程和擦除操作，为哪些 Flash 区域提供保护。不能更改受保护 flash 区域的内容；即，不能对这些区域进行编程，也不能使用任何 flash 命令进行擦除。可以通过编程和擦除操作更改未受保护的区域。

四个 FPROT 寄存器允许存在多达 32 个可保护区域。每个位可保护程序 Flash 存储器的 1/32 区域，程序 Flash 低于以下大小的存储器配置除外：64 KB 程序 Flash，其中每个分配的位保护 2 KB。对于程序 Flash 存储器为 48 KB 或更低的配置，不使用 FPROT0。对于程序 Flash 存储器为 32 KB 或更低的配置，不使用 FPROT1。对于程序 Flash 存储器为 16 KB 或更低的配置，不使用 FPROT2。位域在每个寄存器中定义，如下所示：

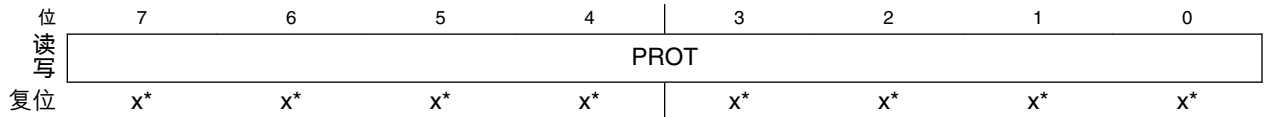
程序 Flash 保护寄存器	程序 Flash 保护位
FPROT0	PROT[31:24]
FPROT1	PROT[23:16]
FPROT2	PROT[15:8]
FPROT3	PROT[7:0]

在复位序列期间，会将 Flash 配置字段中的程序 Flash 保护字节内容加载到 FPROT 寄存器，如下表中所示。

程序 Flash 保护寄存器	Flash 配置字段偏移地址
FPROT0	0x000B
FPROT1	0x000A
FPROT2	0x0009
FPROT3	0x0008

要更改在复位序列期间加载的程序 Flash 保护, 请取消保护包含 Flash 配置字段的程序 Flash 存储器的扇区。然后, 重新对程序 Flash 保护字节进行编程。

Address: 4002_0000h base + 10h offset + (1d × i), where i=0d to 3d



* 注:

- x = 复位时未定义。

FTFA_FPROTn 字段描述

字段	描述
PROT	<p>程序 Flash 区域保护</p> <p>每个程序 Flash 区域都可以通过置位相关 PROT 位提供保护, 避免进行编程和擦除操作。</p> <p>在正常 NVM 模式下: 仅可增加保护, 这意味着可以保护当前不受保护的存储器, 但当前已受保护的存储器无法不受保护。由于不受保护的区域标记为 1 并且受保护的区域标记为 0, 因此只接受从 1 更改为 0 的写入。将逐位执行此 1 至 0 的转换检查。接受具有 1 至 0 转换的 FPROT 位, 而具有 0 至 1 转换的所有位都将被忽略。</p> <p>在 NVM 特殊模式: 所有 FPROT 位均可写入, 而无任何限制。可以保护未受保护的区域, 也可以取消保护已保护的区域。</p> <p>限制: 当命令正在运行(CCIF=0)时, 用户不得写入任何 FPROT 寄存器。</p> <p>尝试改变程序 Flash 存储器中任何受保护区域中的数据会导致保护违规错误, 并将置位 FSTAT[FPVIOL]位。如果它包含任何受保护的区域, 则无法整块擦除程序 Flash 块。</p> <p>32 位保护存储器中的每个位均代表整个程序 Flash 的 1/32 区域, 程序 Flash 低于以下大小的存储器配置除外: 64 KB 程序 Flash, 其中每个分配的位保护 2 KB。</p> <p>0 已保护程序 Flash 区域。 1 未保护程序 Flash 区域</p>

30.3.3.7 只执行访问寄存器 (FTFA_XACCn)

XACC 寄存器定义要将哪些程序 flash 段限制为只可进行数据读取或执行, 或者可获取数据和指令。

8 个 XACC 寄存器允许具有多达 64 个存储器大小相等的受限段。

只执行访问寄存器	程序 flash 只执行访问位
XACCH0	XA[63:56]
XACCH1	XA[55:48]
XACCH2	XA[47:40]
XACCH3	XA[39:32]
XACCL0	XA[31:24]

下一页继续介绍此表...

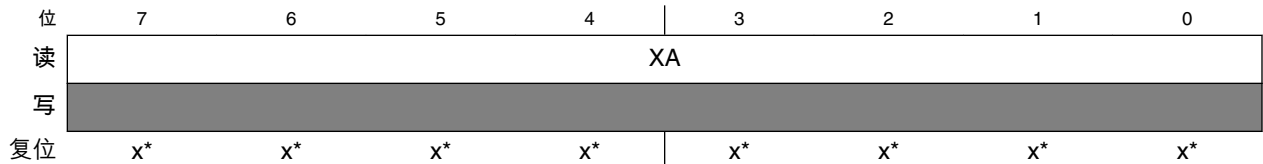
只执行访问寄存器	程序 flash 只执行访问位
XACCL1	XA[23:16]
XACCL2	XA[15:8]
XACCL3	XA[7:0]

在复位序列期间，将向 XACC 寄存器加载程序 Flash IFR 地址 A 和 B 的逻辑 AND，如下表中所示。

只执行访问寄存器	程序 Flash IFR 地址 A	程序 Flash IFR 地址 B
XACCH0	0xA3	0xAB
XACCH1	0xA2	0xAA
XACCH2	0xA1	0xA9
XACCH3	0xA0	0xA8
XACCL0	0xA7	0xAF
XACCL1	0xA6	0xAE
XACCL2	0xA5	0xAD
XACCL3	0xA4	0xAC

使用一次性编程命令对在复位序列期间加载的只执行访问控制字段进行编程。

Address: 4002_0000h base + 18h offset + (1d × i), where i=0d to 7d



* 注:

- x = 复位时未定义。

FTFA_XACCn 字段描述

字段	描述
XA	只执行访问控制 0 只能在执行模式下访问相关段 (作为指令获取) 1 作为数据或在执行模式下访问相关段

30.3.3.8 仅管理员访问寄存器 (FTFA_SACCn)

SACC 寄存器定义要将哪些程序 flash 段限制为仅管理员访问或者用户和管理员访问。

8 个 SACC 寄存器允许具有多达 64 个存储器大小相等的受限段。

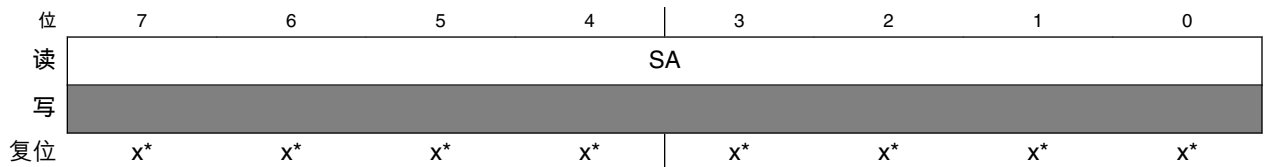
仅管理员访问寄存器	程序 flash 仅管理员访问位
SACCH0	SA[63:56]
SACCH1	SA[55:48]
SACCH2	SA[47:40]
SACCH3	SA[39:32]
SACCL0	SA[31:24]
SACCL1	SA[23:16]
SACCL2	SA[15:8]
SACCL3	SA[7:0]

在复位序列期间，将向 SACC 寄存器加载程序 Flash IFR 地址 A 和 B 的逻辑 AND，如下表中所示。

仅管理员访问寄存器	程序 Flash IFR 地址 A	程序 Flash IFR 地址 B
SACCH0	0xB3	0xBB
SACCH1	0xB2	0xBA
SACCH2	0xB1	0xB9
SACCH3	0xB0	0xB8
SACCL0	0xB7	0xBF
SACCL1	0xB6	0xBE
SACCL2	0xB5	0xBD
SACCL3	0xB4	0xBC

使用一次性编程命令对在复位序列期间加载的仅管理员访问控制字段进行编程。

Address: 4002_0000h base + 20h offset + (1d × i), where i=0d to 7d



* 注:

- x = 复位时未定义。

FTFA_SACCN 字段描述

字段	描述
SA	仅管理员访问控制
0	只能在管理员模式下访问相关段
1	可在用户或管理员模式下访问相关段

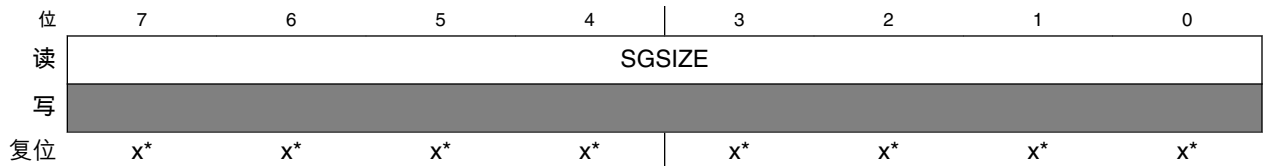
30.3.3.9 Flash 访问段大小寄存器 (FTFA_FACSS)

flash 访问段大小寄存器确定要使用地址中的哪些位来索引至 SACC 和 XACC 位图，以便获相应的权限标志位。

寄存器中的所有位均为只读。

此寄存器的内容在复位序列期间加载。

地址: 4002_0000h 基准 + 28h 偏移 = 4002_0028h



*注:

- x = 复位时未定义。

FTFA_FACSS 字段描述

字段	描述		
SGSIZE	段大小		
	段大小是一个固定值，它等于可用程序 Flash 大小与 NUMSG 之商。		
	程序 Flash 大小	段大小	段大小编码
	64 kB	2 KB	0x3
	128 KB	4 KB	0x4
	160 KB	4 KB	0x4
	256 KB	4 KB	0x4
512 KB	8 KB	0x5	

30.3.3.10 Flash 访问段数量寄存器 (FTFA_FACSN)

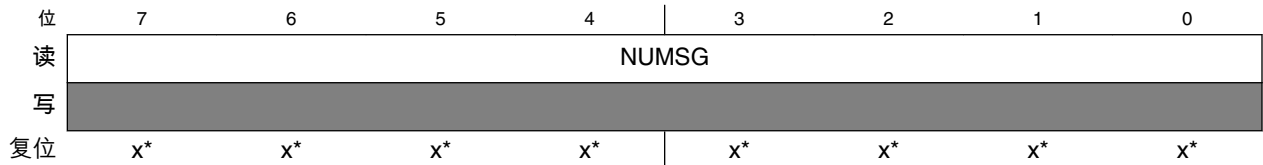
flash 访问段数量寄存器提供 XACC 和 SACC 权限可用的程序 Flash 段数量。

寄存器中的所有位均为只读。

此寄存器的内容在复位序列期间加载。

功能说明

地址: 4002_0000h 基准 + 2Bh 偏移 = 4002_002Bh



*注:

- x = 复位时未定义。

FTFA_FACSN 字段描述

字段	描述
NUMMSG	<p>段数量指示器</p> <p>NUMMSG 字段指明程序 Flash 中大小相等的段数量。</p> <p>0x20 程序 Flash 存储器分为 32 个段 (64 KB , 128 KB)</p> <p>0x28 程序 Flash 存储器分为 40 个段 (160 KB)</p> <p>0x4x 程序 Flash 存储器分为 64 个段 (256 KB , 512 KB)</p>

30.4 功能说明

此部分详细介绍 flash 存储器模块的功能。

30.4.1 Flash 保护

可以对 Flash 存储器内的单个区域进行保护，防止进行编程和擦除操作。

保护功能由以下寄存器控制:

- FPROT_n —
 - 对于 2ⁿ 程序 flash 大小，通常情况下由 4 个寄存器保护程序 flash 存储器的 32 个区域，如下图所示

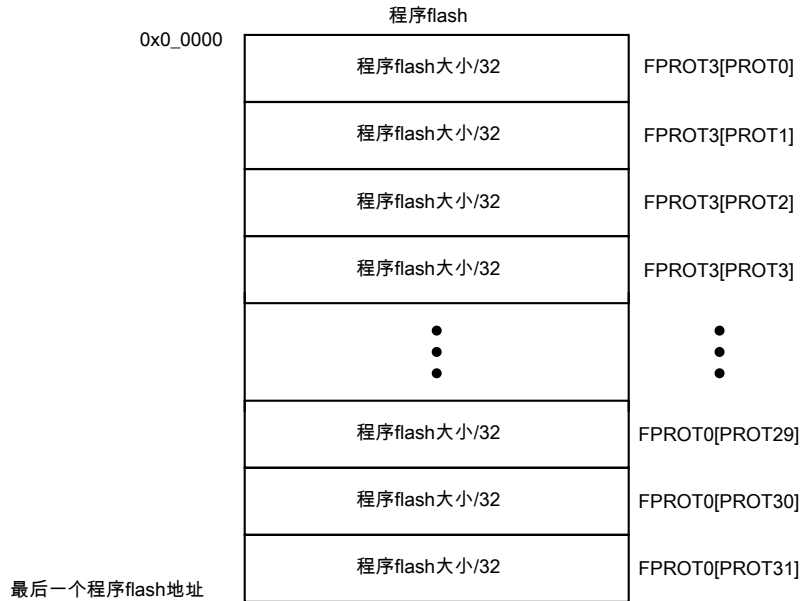


图 30-44. 程序 flash 保护

注

在 [AN4507: 使用 Kinetis 加密和 Flash 保护功能](#) 中将会进一步讨论 flash 保护功能。并非此应用笔记中介绍的所有功能在此器件上均可用。

30.4.2 Flash 访问保护

可以将程序 flash 存储器内的单个区段指定为受限访问。特定 flash 命令 (编程检查、写长字、擦除 Flash 扇区) 可监控 FXACC 内容, 以保护 flash 存储器, 但 FSACC 内容不会影响 flash 命令操作。

更多详情, 请参见 [AN5112: Using the Kinetis Flash Execute-Only Access Control Feature](#)。

访问操作受以下寄存器控制:

- FTFA_XACC
 - 对于大于 128KB 的 2ⁿ 程序 flash 大小, 8 个寄存器控制程序 flash 存储器的 64 个区段, 如下图所示

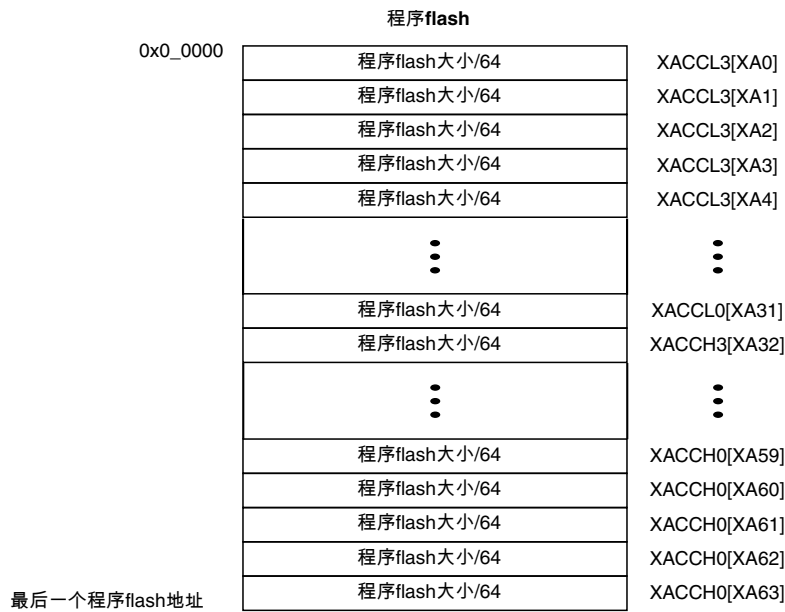


图 30-45. 程序 flash 仅执行访问控制 (256 KB 或 512 KB 程序 flash)

• —FTFA_SACC

- 对于大于 128KB 的 2ⁿ 程序 flash 大小，8 个寄存器控制程序 flash 存储器的 64 个区段，如下图所示

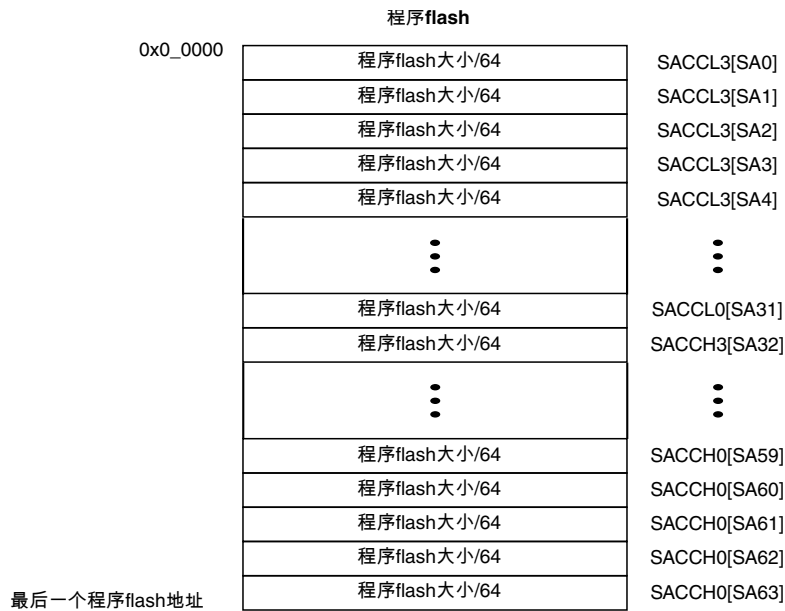


图 30-46. 程序 flash 管理员访问控制 (256 KB 或 512 KB 程序 flash)

30.4.3 中断

Flash 存储器模块可在发生各种 flash 事件时生成 MCU 中断请求。

这些中断事件及其相关的状态和控制位如下表所示。

表 30-44. Flash 中断源

Flash 事件	可读取 状态位	中断 使能位
Flash 命令完成	FSTAT[CCIF]	FCNFG[CCIE]
Flash 读取冲突错误	FSTAT[RDCOLERR]	FCNFG[RDCOLLIE]

注

将在 MCU 级别确定向量地址及其相应中断优先级。

某些器件还可因读取冲突错误事件生成总线错误响应。请参见芯片配置信息，以确定是否也支持生成总线错误响应。

30.4.4 低功耗模式下的 Flash 操作

30.4.4.1 等待模式

如果 MCU 进入等待模式，flash 存储器模块不受影响。Flash 存储器模块可通过命令完成中断从等待模式恢复 MCU（请参见[中断](#)）。

30.4.4.2 停止模式

当 MCU 请求停止模式时，如果 flash 命令处于活动状态(CCIF = 0)，则在允许 MCU 进入停止模式之前会先执行完成命令。

警告

在任何 flash 命令运行(CCIF = 0)时，MCU 均不应进入停止模式。

注

当 MCU 处于超低功耗模式 (VLPR、VLPW、VLPS) 时，flash 存储器模块不会接受 flash 命令。

30.4.5 功能操作模式

Flash 存储器模块具有两种操作模式：正常 NVM 和特殊 NVM。

操作模式影响命令集可获性（参见表 30-45）。请参阅此器件的芯片配置详细信息，了解如何启动每种模式。

30.4.6 Flash 读取和忽略的写入操作

Flash 存储器模块仅需 flash 地址即可执行 flash 存储器读取操作。

在该块上运行命令（由 CCIF=0 表示）时，MCU 不得通过 flash 存储器读取。由块内的任何命令时，无法确保通过 flash 块读取数据。块仲裁逻辑将检测任何同时访问，并将此报告为读取冲突错误（参见 FSTAT[RDCOLERR]位）。

30.4.7 写入时读取（RWW）

对于仅带程序 Flash 的器件，不允许：

- 当 Flash 命令处于活动状态（CCIF=0）时从程序 Flash 存储器空间中读取。

30.4.8 Flash 程序和擦除

除读取之外的所有 flash 功能均需要用户通过一系列外设总线写入来设置和启动 flash 命令。

在通知当前命令已完成之前，用户无法发起任何其他 flash 命令。Flash 命令结构和操作详情如 [Flash 命令操作](#) 所述。

30.4.9 Flash 命令操作

Flash 命令操作一般用于修改 Flash 存储器内容。

以下章节将介绍：

- 用于设置 Flash 命令参数和启动执行的命令写入序列。
- 所有可用 Flash 命令说明

30.4.9.1 命令写入序列

Flash 命令使用命令写入序列进行指定，如 [图 30-47](#) 中所示。flash 存储器模块对命令 (FCCOB) 内容执行各种检查，并在满足所有要求的情况下继续执行命令。

在启动命令之前，FSTAT 寄存器中的 ACCERR 位和 FPVIOL 位必须为零，CCIF 标志位必须读为 1，以验证之前执行的命令已完成。如果 CCIF 为 0，表示之前的命令执行仍处于激活状态，此时无法启动新的命令写入序列，对 FCCOB 寄存器的所有写入操作都将被忽略。

在 VLP 模式下启动 flash 命令的尝试将被忽略。在 HSRUN 模式下启动 flash 命令的尝试将受置位的 ACCERR 标志位的限制。

30.4.9.1.1 加载 FCCOB 寄存器

用户必须向 FCCOB 寄存器加载所需的 Flash 命令要求的所有参数。对于组成 FCCOB 数据集的各独立寄存器，可以任何顺序写入。

30.4.9.1.2 通过清除 CCIF 启用命令

加载所有相关命令参数后，用户通过清零 FSTAT[CCIF] 启用命令，向 FSTAT[CCIF] 位写 1 可以清除该位。FSTAT[CCIF] 保持为零，直到完成 flash 命令。

如果之前的命令导致访问错误(FSTAT[ACCERR]=1)或误写入保护区(FSTAT[FPVIOL]=1)，则 FSTAT 寄存器包含的阻止机制将会阻止启用新命令（无法清零 FSTAT[CCIF]）。在错误情形下，需要两次写 FSTAT 以启用下一个命令：第一次写入清除错误标志，第二次写入清除 CCIF。

30.4.9.1.3 命令执行和错误报告

命令处理步骤如下：

1. Flash 存储器模块可读取命令代码并执行一系列各命令独有的参数检查和保护检查（如适用）。

如果参数检查失败，FSTAT[ACCERR]（访问错误）标志置位。FSTAT[ACCERR] 报告指令代码无效且地址超过边界。通常，访问错误表明命令在 FCCOB 寄存器组中未设置有效参数。

编程和擦除命令还检查地址，以确定是否要求在受保护区域执行操作。如果保护检查失败，FSTAT[FPVIOL]（保护错误）标志置位。

当参数或保护步骤失败时，不会继续执行命令处理。而是在置位 FSTAT[CCIF] 之后终止命令处理。

2. 如果参数和保护检查通过，则会继续执行命令。在执行过程中可能会发生运行时错误，如擦除验证失败。运行时错误将在 FSTAT[MGSTAT0] 中报告。命令可能出现访问错误、保护错误和运行时错误，但在纠正所有访问和保护错误之前不会出现运行时错误。
3. 命令执行结果（如适用）会通过 FCCOB 和 FSTAT 寄存器报告给用户。

4. Flash 存储器模块置位 FSTAT[CCIF], 这意味着已完成命令。

通用命令写入序列的流程图如下图所示。

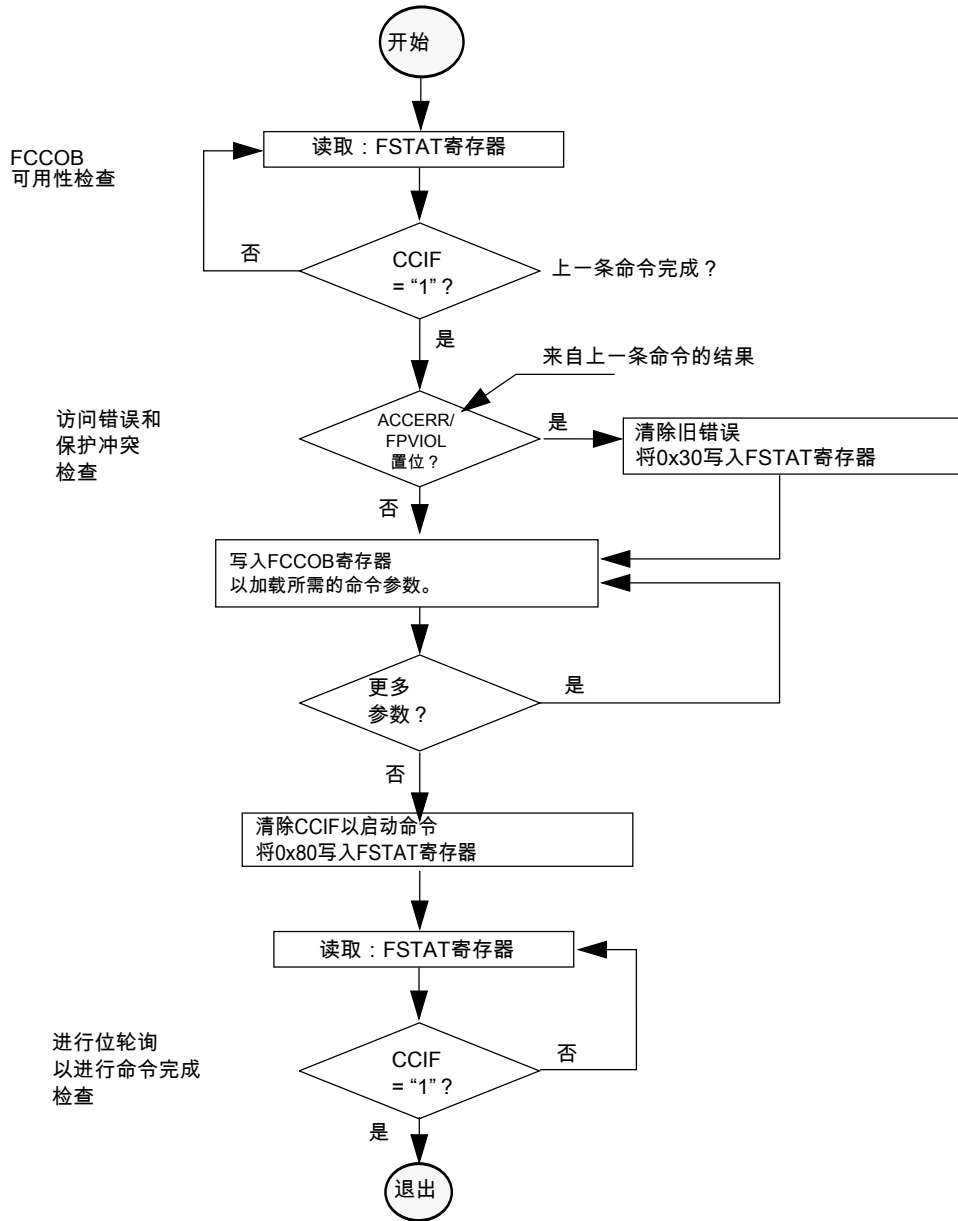


图 30-47. Flash 命令写入序列通用流程图

30.4.9.2 Flash 命令

下表概述了所有 flash 命令的功能。

FCMD	命令	程序 flash	功能
0x01	指定区域空白验证(Read 1s Section)	×	验证从起始地址开始的给定数量的程序 flash 位置是否已擦除。
0x02	编程检查	×	测试处于裕量读取级别的先前已编程的位置。
0x03	读取资源	IFR, ID	从程序 flash IFR 或版本 ID 中读取 4 个字节。
0x06	写长字	×	在程序 flash 块中编程 4 个字节。
0x09	擦除 Flash 扇区	×	擦除某个程序 flash 扇区中的所有字节。
0x40	读取 1s 所有模块	×	验证程序 flash 块是否已擦除, 然后释放 MCU 安全。
0x41	一次性读取	IFR	在程序 flash 0 IFR 中读取专用 64 字节字段的 4 个字节。
0x43	一次性编程	IFR	对程序 flash 0 IFR 中专用 64 字节字段的 4 个字节进行一次性编程。
0x44	擦除所有模块	×	擦除程序 flash 块, 验证擦除操作, 然后释放 MCU 安全。 注: 仅当所有存储器位置都未受保护时, 才能执行擦除操作。
0x45	验证后门访问密钥	×	将一组用户提供的安全密钥与程序 flash 中存储的密钥进行比较后, 释放 MCU 安全。
0x49	擦除所有解密模块	×	擦除程序 flash 块, 验证擦除操作, 将安全字节编程为非安全状态, 然后释放 MCU 安全。
0x4A	读取 1s 所有仅执行区段	×	验证所有程序 flash 仅执行(XA)段是否已擦除, 然后释放 flash 访问控制。
0x4B	擦除所有仅执行区段	×	擦除所有程序 flash 仅执行(XA)段, 然后释放 flash 访问控制。

30.4.9.3 Flash 命令（按模式分类）

下表显示了可在各 flash 操作模式下执行的 flash 命令。

表 30-45. Flash 命令（按模式分类）

FCMD	命令	正常 NVM			特殊 NVM		
		不安全	安全	MEEN=10	不安全	安全	MEEN=10
0x01	指定区域空白验证(Read 1s Section)	×	×	×	×	—	—
0x02	编程检查	×	×	×	×	—	—

下一页继续介绍此表...

表 30-45. Flash 命令（按模式分类）（继续）

FCMD	命令	正常 NVM			特殊 NVM		
		不安全	安全	MEEN=10	不安全	安全	MEEN=10
0x03	读取资源	×	×	×	×	—	—
0x06	写长字	×	×	×	×	—	—
0x09	擦除 Flash 扇区	×	×	×	×	—	—
0x40	读取 1s 所有模块	×	×	×	×	×	—
0x40	读取 1s 所有模块	×	×	—	×	×	—
0x41	一次性读取	×	×	×	×	—	—
0x43	一次性编程	×	×	×	×	—	—
0x44	擦除所有模块	×	×	×	×	×	—
0x44	擦除所有模块	×	×	—	×	×	—
0x45	验证后门访问密钥	×	×	×	×	—	—
0x49	擦除所有解密模块	×	×	—	×	×	—
0x4A	读取 1s 所有仅执行区段	×	×	×	×	—	—
0x4B	擦除所有仅执行区段	×	×	×	×	—	—

30.4.10 裕量读取命令

Read-1 命令（读取 1s 所有模块、指定区域空白验证(Read 1s Section)、读取 1s 所有仅执行区段）和编程检查命令具有裕量选择参数，允许用户将非标准读取参考级别应用于这些命令执行的程序 flash 阵列读取。使用预设的“用户”和“出厂”裕量级别，这些命令可以严于“正常”读取的容差执行相关的读取操作。仅在命令执行过程中应用这些非标准读取级别。基本的 flash 阵列读取使用标准、非裕量的读取参考级别。

在正常 flash 使用过程中，应仅使用“正常”读取级别。仅在特殊情况下使用非标准、“用户”和“出厂”裕量级别。它们可用于特殊诊断例程中，以增强器件不会遭受 flash 存储器使用寿命终止导致数据丢失的信心。

已擦除（“1”）和已编程（“0”）位状态会因时间流逝和数据循环（位被擦除和重新编程的次数）而降级。已擦除状态的生存期与最后一次擦除操作相关。已编程状态的生存期从最后一次编程时间开始算起。

“用户”和“出厂”级别实际上是最低安全裕量，即，如果读取以严于“用户”和“出厂”裕量的容差通过，则在遭遇数据丢失之前，“正常”读取至少具有这么多安全裕量。

“用户”裕量略高于正常读取参考级别。“用户”裕量级别可用于检查 flash 存储器内容是否拥有足够的裕量，以便执行正常级别的读取操作。如果在以“用户”裕量级别检查 flash 存储器内容时遇到意外读取结果，则在“正常”读取过程中可能很快出现信息丢失的情况。

“出厂”裕量是一个大于正常标准的更大偏差，它是一个更严格的读取标准，应在循环周期早期完成擦除或编程命令之后立即（或尽快）使用。“出厂”裕量级别用于检查 flash 存储器内容是否具有足够的裕量，以便以正常级别设置实现长期数据保留。如果在以“出厂”裕量级别检查 Flash 存储器内容时遇到意外结果，则应擦除 Flash 存储器内容并对其重新编程。

警告

只有在检验初始出厂程序的过程中，才能使用出厂裕量级别。

30.4.11 Flash 命令说明

本节介绍由命令写入序列启动的所有 Flash 命令。

如果出现以下任意非法条件，则 Flash 存储器模块将置位 FSTAT[ACCERR]位并中止执行命令：

- FCCOB FCMD 字段中存在未被识别的命令代码。
- 特定命令的 FCCOB 字段中存在错误。请参阅针对各个命令提供的错误处理表。

启动命令写入序列之前，请确保 FSTAT[ACCERR]和 FSTAT[FPVIOL]已清零。如[通过清除 CCIF 启用命令](#)所述，当置位这些错误标志位时，无法启动新命令。

当 flash 存储器模块在块上运行命令(FSTAT[CCIF] = 0)时，请勿尝试读取该 flash 块。Flash 存储器模块可能会向 MCU 返回无效数据，并置位冲突错误标志位(FSTAT[RDCOLERR])。

警告

Flash 数据在编程之前必须处于已擦除状态。不允许对位进行累计编程（添加更多 0）。

30.4.11.1 指定区域空白验证(Read 1s Section)命令

指定区域空白验证(Read 1s Section)命令检查程序 flash 存储器的某个部分是否擦除为指定的读取裕量级别。指定区域空白验证(Read 1s Section)命令定义要验证的起始地址以及字段的数量。

表 30-46. 指定区域空白验证(Read 1s Section)命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x01 (RD1SEC)
1	要验证的首个字段的 Flash 地址[23:16]
2	要验证的首个字段的 Flash 地址[15:8]
3	要验证的首个字段的 flash 地址[7:0] ¹
4	要验证的字段数量[15:8]
5	要验证的字段数量[7:0]
6	读取-1 裕量选择

1. 必须字段对齐 (Flash 地址[2:0] = 000)。

在清除 CCIF 以启动指定区域空白验证(Read 1s Section)命令后, flash 存储器模块将根据表 30-47 将读取裕量置位为 1s, 然后读取 flash 存储器特定部分内的所有位置。如果 flash 存储器模块无法读取所有 1s (即 flash 部分未擦除), 将置位 FSTAT[MGSTAT0]。FSTAT[CCIF]在指定区域空白验证(Read 1s Section)操作完成后进行置位。

表 30-47. 指定区域空白验证(Read 1s Section)的裕量级别选择

读取裕量选择	裕量级说明
0x00	使用 1s 的“正常”读取级
0x01	将“用户”裕量应用到正常读取-1 级
0x02	将“出厂”裕量应用到正常读取-1 级

表 30-48. 指定区域空白验证(Read 1s Section)命令错误处理

错误情况	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
提供的裕度代码无效。	FSTAT[ACCERR]
提供的 Flash 地址无效。	FSTAT[ACCERR]
Flash 地址不是字段对齐。	FSTAT[ACCERR]
请求的区域越过了 Flash 存储器块边界。	FSTAT[ACCERR]
请求的字段数量为 0。	FSTAT[ACCERR]
读取 1s 失败。	FSTAT[MGSTAT0]

30.4.11.2 编程检查命令

编程检查命令对之前写入 flash 的长字进行测试, 查看在指定的裕量级别是否可以正确地读取。

表 30-49. 编程检查命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x02 (PGMCHK)
1	Flash 地址[23:16]
2	Flash 地址[15:8]
3	Flash 地址[7:0] ¹
4	裕度选择
8	PDATA[31:24]
9	PDATA[23:16]
A	PDATA[15:8]
B	PDATA[7:0]

1. 必须长字对齐 (Flash 地址[1:0] = 00)。

清除 CCIF 以启用编程检查命令时，flash 存储器模块将根据表 30-50 将读取裕量设为 1s，读取指定的长字并将实际读取数据与 FCCOB 提供的预期数据对比。如果在裕量-1 比较失败，则会置位 FSTAT[MGSTAT0]。

Flash 存储器模块随后会将读取裕量设为 0s，重新读取并再次对比。如果在裕量-0 比较失败，则会置位 FSTAT[MGSTAT0]。FSTAT[CCIF]在编程检查操作完成后置位。

提供的地址必须为长字对齐 (字节地址的最低两位必须为 00):

- 字节 3 数据将写入提供的字节地址 (“起始”),
- 字节 2 数据将编程为字节地址起始+0b01,
- 字节 1 数据将编程为字节地址起始+0b10,
- 字节 0 数据将编程为字节地址起始+0b11。

注

请参见裕量读取说明，[裕量读取命令](#)

表 30-50. 编程检查的裕量级选择

读取裕量选择	裕量级说明
0x01	“用户”裕量-1 和“用户”裕量-0 的读数
0x02	“工厂”裕量-1 和“工厂”裕量-0 的读数

表 30-51. 编程检查命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
提供的 Flash 地址无效	FSTAT[ACCERR]
Flash 地址不是长字对齐	FSTAT[ACCERR]
提供的裕量选择无效	FSTAT[ACCERR]

下一页继续介绍此表...

表 30-51. 编程检查命令错误处理 (继续)

错误状态	错误位
Flash 地址位于 XA 控制的段，自上次复位以来非安全擦除块命令或者读取 1s 所有模块命令未成功执行过	FSTAT[FPVIOL]
其中一个裕量读数与预期数据不匹配	FSTAT[MGSTAT0]

30.4.11.3 读取资源命令

读取资源命令允许用户读取位于 flash 存储器模块内专用内存资源中的数据。用户可用的专用内存资源包括程序 flash IFR 空间和版本 ID 字段。对每个资源均会分配一个选择代码，如表 30-53 所示。

表 30-52. 读取资源命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x03 (RDRSRC)
1	Flash 地址[23:16]
2	Flash 地址[15:8]
3	Flash 地址[7:0] ¹
返回值	
4	读取数据[31:24]
5	读取数据[23:16]
6	读取数据[15:8]
7	读取数据[7:0]
用户提供的值	
8	资源选择代码 (参见表 30-53)

1. 必须长字对齐 (Flash 地址[1:0] = 00)。

表 30-53. 读取资源选择代码

资源选择代码	说明	资源大小	本地地址范围
0x00	程序 Flash 0 IFR	256 字节	0x00_0000–0x00_00FF
0x01 ¹	版本 ID	8 字节	0x00_0000–0x00_0007

1. 位于程序 flash 0 保留空间内。

清除 CCIF 以启用读取资源命令之后，可以从提供的相对地址处的选定资源内读取四个连续字节，并将其存储在 FCCOB 寄存器中。完成读取资源操作之后，CCIF 标志位将置位。如果提供的是无效资源代码或者如果适用区域的地址超出范围，则读取资源命令将退出并返回访问错误。

表 30-54. 读取资源命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
输入的资源代码无效	FSTAT[ACCERR]
目标资源的 Flash 地址超出范围。	FSTAT[ACCERR]
Flash 地址不是长字对齐	FSTAT[ACCERR]

30.4.11.4 写长字命令

写长字命令使用嵌入式算法对程序 flash 存储器中之前已擦除的 4 字节进行编程。

警告

在编程之前，Flash 存储器位置必须处于已擦除状态。不允许在某个 Flash 存储器位置进行位的累计写操作（中间没有擦除操作的连续写操作）。不允许将现有的 0 重新编程为 0，因为这样会加重器件负担。

表 30-55. 写长字命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x06 (PGM4)
1	Flash 地址[23:16]
2	Flash 地址[15:8]
3	Flash 地址[7:0] ¹
4	字节 0 编程值
5	字节 1 编程值
6	字节 2 编程值
7	字节 3 编程值

1. 必须长字对齐 (Flash 地址[1:0] = 00)。

在清除 CCIF 以启动写长字命令时，flash 存储器模块将使用提供的地址将数据字节编程到 flash 中。目标 flash 位置当前必须处于未受保护状态(请参阅 FPROT 寄存器说明)，这样才能执行写长字操作。

编程操作为单向操作。它只能将 NVM 位从已擦除状态('1')移至已编程状态('0')。无法编程为'0'状态的已擦除位将在 FSTAT[MGSTAT0]中标记为错误。完成写长字操作之后，CCIF 标志位将置位。

提供的地址必须为长字对齐 (flash 地址[1:0] = 00):

- 字节 3 数据将写入提供的字节地址 (“起始”),
- 字节 2 数据将编程为字节地址起始+0b01,

- 字节 1 数据将编程为字节地址起始+0b10，并且
- 字节 0 数据将编程为字节地址起始+0b11。

表 30-56. 写长字命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
提供的 Flash 地址无效	FSTAT[ACCERR]
Flash 地址不是长字对齐	FSTAT[ACCERR]
在 NVM 正常模式或 NVM 特殊模式下，Flash 地址指向受保护的区域	FSTAT[FPVIOL]
Flash 地址位于 XA 控制的段，自上次复位以来非安全擦除块命令或者读取 1s 所有模块命令未成功执行过	FSTAT[FPVIOL]
在验证操作期间遇到任何错误	FSTAT[MGSTAT0]

30.4.11.5 擦除 Flash 扇区命令

擦除 Flash 扇区操作将擦除 flash 扇区中的所有地址。

表 30-57. 擦除 Flash 扇区命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x09 (ERSSCR)
1	要擦除 flash 扇区中的 Flash 地址[23:16]
2	要擦除 flash 扇区中的 Flash 地址[15:8]
3	Flash 地址[7:0] ¹ 要擦除的 flash 扇区中

1. 必须字段对齐 (flash 地址[2:0] = 000)。

清除 CCIF 以启用擦除 Flash 扇区命令时，flash 存储器模块将擦除选定的程序 flash 扇区并验证其已被擦除。如果选定扇区受到保护，则擦除 Flash 扇区命令将中止 (请参见 FPROT 寄存器说明)。如果擦除验证失败，FSTAT[MGSTAT0]位将会置位。完成擦除 Flash 扇区操作之后，CCIF 标志位将置位。擦除 Flash 扇区命令可挂起 (请参见 FCNFG[ERSSUSP]位和图 30-48)。

表 30-58. 擦除 Flash 扇区命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
提供的 flash 地址无效	FSTAT[ACCERR]
Flash 地址不是字段对齐。	FSTAT[ACCERR]
选定的程序 flash 扇区受到保护	FSTAT[FPVIOL]
选定的程序 flash 扇区位于 XA 控制段中和擦除所有模块，自上次复位以来非安全擦除块命令或者读取 1s 所有模块命令 未成功执行过	FSTAT[FPVIOL]
验证期间发生的任何错误 ¹	FSTAT[MGSTAT0]

1. 可以使用指定区域空白验证(Read 1s Section)命令运行用户裕量读取，以验证所有位是否已擦除。

30.4.11.5.1 挂起擦除 Flash 扇区操作

要挂起擦除 Flash 扇区操作，需要在 CCIF、ACCERR 和 FPVIOL 位为零时，置位 FCNFG[ERSSUSP]位，并在 CCOB 命令字段填入擦除 Flash 扇区命令的代码。在擦除 Flash 扇区操作过程中（参见[擦除 Flash 扇区命令](#)），flash 存储器模块将在方便时对 ERSSUSP 位的状态进行采样。如果 flash 存储器模块检测到 ERSSUSP 位置位，擦除 Flash 扇区操作将挂起且 flash 存储器模块将置位 CCIF。置位 ERSSUSP 时，除了对 FSTAT 和 FCNFG 寄存器的写入操作之外，对 flash 寄存器的所有写入操作将被忽略。

在 flash 存储器模块检测到已挂起请求之前，如果擦除 Flash 扇区操作有效完成，则 flash 存储器模块将在置位 CCIF 之前清除 ERSSUSP 位。如果擦除 Flash 扇区操作已成功挂起，则 flash 存储器模块将置位 CCIF 并使 ERSSUSP 位保持置位。CCIF 置位时，仅可清除 ERSSUSP 位以防止在 flash 存储器模块应答之前撤销挂起请求。

30.4.11.5.2 恢复挂起的擦除 Flash 扇区操作

如果在清除 CCIF 以启动下一条命令时 ERSSUSP 位仍置位，则上一个擦除 Flash 扇区操作将恢复。flash 存储器模块通过清除 ERSSUSP 位来确认恢复挂起操作的请求。随后，可通过置位 ERSSUSP 来发起新的挂起请求。单个擦除 Flash 扇区操作可进行多次挂起和恢复。

请求恢复擦除 Flash 扇区操作（清除 CCIF）与请求重新挂起操作（置位 ERSSUSP）之间的最短间隔时间限制为 4.3 毫秒。必须满足此最短时间周期要求以确保擦除 Flash 扇区操作将最终完成。如果不断违反此最短周期（即反复和过快地发起挂起请求），则擦除 Flash 扇区算法不会继续向前处理。恢复/挂起序列操作将反复进行，而不能完成擦除操作。

30.4.11.5.3 中止挂起的擦除 Flash 扇区操作

用户可以选择中止挂起的擦除 Flash 扇区操作，方法是在清除 CCIF 以便启动下一条命令之前先清除 ERSSUSP 位。当中止挂起的操作后，flash 存储器模块将使用新的 FCCOB 内容启动新命令。

注

中止擦除会使位单元处于不确定的部分擦除状态。在新擦除命令全部完成之前，此扇区内的数据不可靠。

下图显示如何挂起和恢复擦除 Flash 扇区操作。

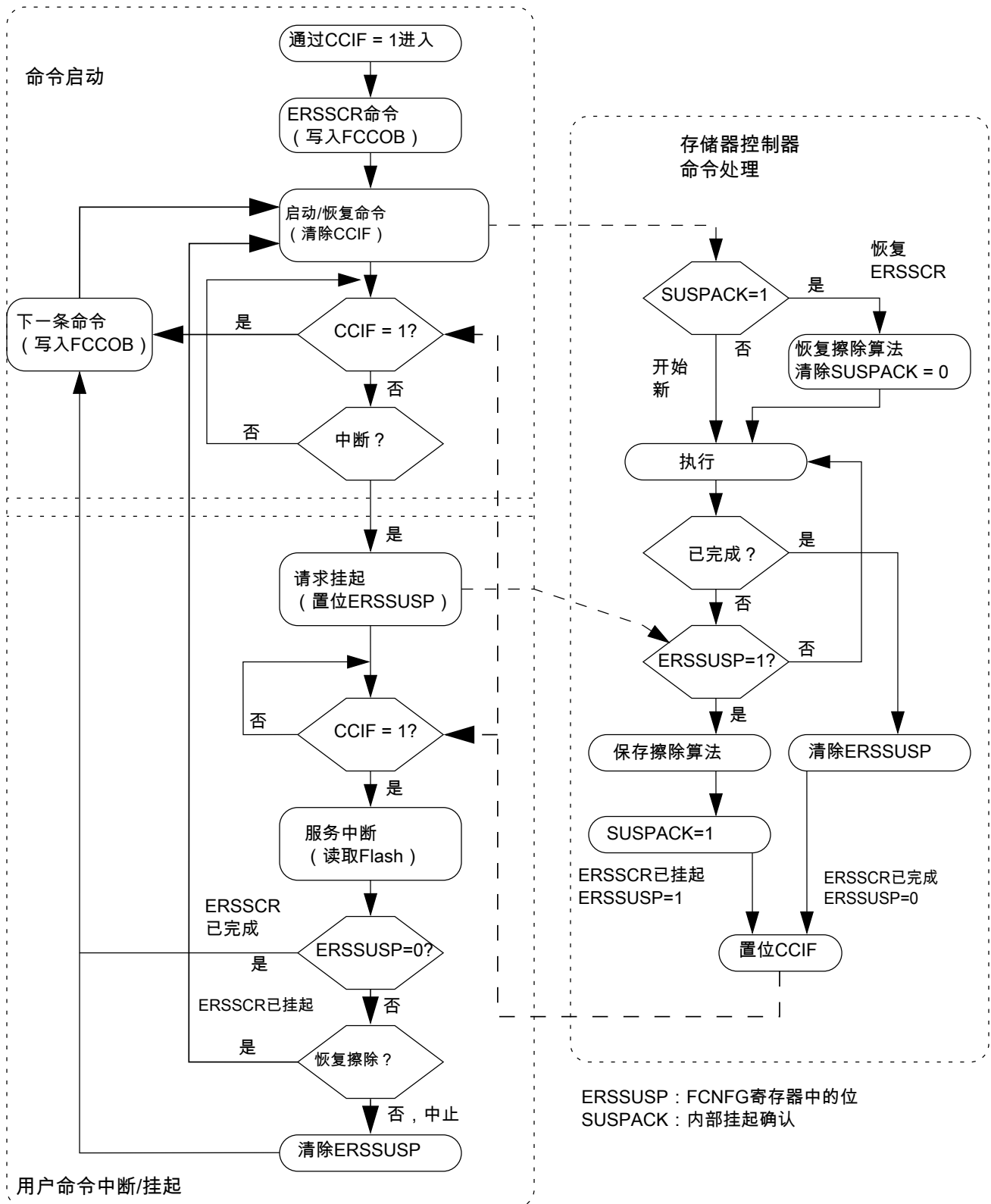


图 30-48. 挂起和恢复擦除 Flash 扇区操作

30.4.11.6 读取 1s 所有模块命令

读取 1s 所有模块命令检查程序 flash 块是否已擦除至指定读取裕量级别, 如果适用, 在读数通过时将释放安全, 即所有数据读取为'1'。

表 30-59. 读取 1s 所有模块命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x40 (RD1ALL)
1	读取-1 裕量选择

在清除 CCIF 以启动读取 1s 所有模块命令后, flash 存储器模块:

- 根据表 30-60 将读取裕量置位为 1s,
- 检查程序 flash 的内容是否处于已擦除状态。

如果 flash 存储器模块确认这些存储器资源已被擦除, 则访问控制将禁用, 并且系统会通过将 FSEC[SEC]字段置位为非安全状态来释放安全。flash 配置字段中的安全字节 (请参阅 [Flash 配置字段说明](#)) 将不受读取 1s 所有模块命令的影响。如果读取失败, 即所有存储器资源并非都处于完全擦除状态, 则 FSTAT[MGSTAT0]位将置位。

完成读取 1s 所有模块操作之后, CCIF 标志位将置位。完成操作之后, MMBSEL 寄存器将被清除。

表 30-60. 读取 1s 所有模块的裕量级别选择

读取裕量选择	裕量级说明
0x00	使用 1s 的“正常”读取级
0x01	将“用户”裕量应用到正常读取-1 级
0x02	将“出厂”裕量应用到正常读取-1 级

表 30-61. 读取 1s 所有模块命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
指定了无效的裕量选择	FSTAT[ACCERR]
读取-1s 失败	FSTAT[MGSTAT0]

30.4.11.7 一次性读取命令

一次性读取命令提供对程序 flash 0 IFR 中特殊 96 字节字段的读取访问权限(参见程序 [Flash IFR 映射](#) 和 [一次性编程 字段](#))。通过 16 条记录 (索引值为 0x00 - 0x0F) 对一次性编程 ID 字段进行访问, 每条记录的长度为 4 字节。通过 4 条记录 (索引值为 0x10 - 0x13) 对一次性编程 XACC 和 SACC 字段进行访问, 每条记录的长度为 8 字节。可使用 [一次性编程命令](#) 中介绍的一次性编程命令对这些字段进行编程。

表 30-62. 一次性读取命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x41 (RDONCE)
1	一次性编程记录索引(0x00 - 0x13)
2	不使用
3	不使用
返回值	
4	的一次性编程字节 0 值
5	的一次性编程字节 1 值
6	的一次性编程字节 2 值
7	的一次性编程字节 3 值
8	一次性编程字节 4 值 (索引为 0x10 - 0x13)
9	一次性编程字节 5 值 (索引为 0x10 - 0x13)
10	一次性编程字节 6 值 (索引为 0x10 - 0x13)
11	一次性编程字节 7 值 (索引为 0x10 - 0x13)

在清除 CCIF 以启动一次性读取命令后, 将读取 4 字节或 8 字节一次性编程记录并存储在 FCCOB 寄存器中。完成一次性读取操作之后, CCIF 标志位将置位。一次性读取命令的有效记录索引值范围为 0x00 - 0x13。在一次性读取命令的执行过程中, 对包含所选记录索引的程序 flash 块内地址的任何读取尝试都将返回无效数据。一次性读取命令可以执行任意次数。

表 30-63. 一次性读取命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
提供了无效的记录索引	FSTAT[ACCERR]

30.4.11.8 一次性编程命令

一次性编程命令允许对程序 flash 0 IFR 中的特殊 96 字节字段进行编程 (参见程序 [Flash IFR 映射](#) 和 [一次性编程 字段](#))。通过 16 条记录 (索引值为 0x00 - 0x0F) 访问一次性编程 ID 字段, 每条记录的长度为 4 字节。通过 4 条记录 (索引值为 0x10 -

0x13) 对一次性编程 XACC 和 SACC 字段进行访问, 每条记录的长度为 8 字节。可以使用一次性读取命令 (参见[一次性读取命令](#)) 或读取资源命令 (参见[读取资源命令](#)) 来读取这些记录。由于程序 flash 0 IFR 无法由擦除, 因此只能对这些记录编程一次。

表 30-64. 一次性编程命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	tPGMONCE0x430x43
1	一次性编程记录索引(0x00 - 0x13)
2	未使用
3	未使用
4	一次性编程字节 0 值
5	一次性编程字节 1 值
6	一次性编程字节 2 值
7	一次性编程字节 3 值
8	一次性编程字节 4 值 (索引为 0x10 - 0x13)
9	一次性编程字节 5 值 (索引为 0x10 - 0x13)
10	一次性编程字节 6 值 (索引为 0x10 - 0x13)
11	一次性编程字节 7 值 (索引为 0x10 - 0x13)

在清除 CCIF 以启动一次性编程命令后, flash 存储器模块首先验证所选记录是否已被擦除。如果已擦除, 则系统会使用提供的值对所选记录进行编程。一次性编程命令还会验证是否可以正确回读已编程的值。完成一次性编程操作之后, CCIF 标志位将置位。

如果某条记录的值不全是 0xF(表示已擦除), 则不允许有对其的任何写入尝试。一次性编程命令的有效记录索引值范围为 0x00 - 0x13。在一次性编程命令的执行过程中, 对包含所选记录索引的被写入 flash 块内地址的任何读取尝试都将返回无效数据。

表 30-65. 一次性编程命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
提供了无效的记录索引	FSTAT[ACCERR]
请求的记录已编程为非 FFFF 值 ¹	FSTAT[ACCERR]
验证操作期间发生的任何错误	FSTAT[MGSTAT0]

1. 如果一次性编程记录最初编程为 0xFFFF_FFFF (0xFFFF_FFFF_FFFF_FFFF 表示索引 0x10 - 0x13), 则允许重新对同一记录执行一次性编程命令。

30.4.11.9 擦除所有模块命令

擦除所有模块操作用于擦除所有 flash 存储器，验证所有存储器内容和解除 MCU 加密。

表 30-66. 擦除所有模块命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x44 (ERSALL)

清除 CCIF 以启用擦除所有模块命令后，flash 存储器模块将擦除所有程序 flash 存储器，然后再验证是否已全部擦除。

如果 flash 存储器模块验证所有 flash 存储器已正确擦除，则访问控制将禁用，并且系统会通过将 FSEC[SEC] 字段置位于非安全状态来解除加密。如果在正常 NVM 或特殊 NVM 模式下，任何 flash 区域受到保护 Flash 配置字段的加密字节和所有其他内容（参见 [Flash 配置字段说明](#)）均可通过擦除所有模块命令擦除。如果擦除验证失败，则 FSTAT[MGSTAT0] 位将置位。完成擦除所有模块操作之后，CCIF 标志位将置位。

由 FXACC 寄存器内容决定的访问控制不会阻止擦除所有模块命令的执行。擦除大部分 Flash 存储器后，包含 Program Once XACC 和 SACC 字段的程序 flash IFR 空间将不会擦除，因此，Program Once XACC 和 SACC 字段的内容不会更改。FXACC 和 FSACC 寄存器的内容不会受擦除所有模块命令执行的影响。完成擦除所有模块命令之后，访问控制将禁用，直到下一次复位 flash 模块或执行读取 1s 所有模块命令并且失败（FSTAT[MGSTAT0] 置位）。

表 30-67. 擦除所有模块命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
在正常 NVM 模式或特殊 NVM 模式下，程序 flash 存储器的任何区域均受到保护	FSTAT[FPVIOL]
验证期间发生的任何错误 ¹	FSTAT[MGSTAT0]

1. 可以使用读取 1s 所有模块命令运行用户裕量读取，以验证所有位是否已擦除。

30.4.11.9.1 对 Flash 存储器模块触发“外部全部擦除”

擦除所有模块/擦除所有解密模块命令的功能也适用于 flash 存储器模块的输入引脚外的非命令方式。请参阅器件的芯片配置详细信息，以了解有关此功能的信息。

在调用外部全部擦除功能之前，必须清除 FSTAT[ACCERR 和 PVIOL] 标志位，并且 FCCOB0 寄存器不得包含 0x44。无论当前有怎样的保护设置，当调用时，全部擦除功能将擦除所有程序 flash 存储器。如果擦除后检验通过，将禁用于 FXACC 寄存器内容确定的访问控制，并且该流程随后会通过将 FSEC[SEC] 字段寄存器置位为解密

状态。“Flash 配置”字段中的安全字节也将设置为解密状态。FCNFG[ERSAREQ]位反映全部擦除请求的状态。该操作完成之后，FCNFG[ERSAREQ]位将清零，且正常 FSTAT 错误报告将可用（FPVIOL 除外），如[擦除所有模块命令/擦除所有解密模块命令](#)中所述。

30.4.11.10 验证后门访问密钥命令

仅当满足模式和加密条件时才会执行验证后门访问密钥命令（参见[Flash 命令（按模式分类）](#)）。验证后门访问密钥命令的执行由 FSEC[KEYEN]位进一步确定。如果 FCCOB 中用户提供的密钥与 Flash 配置字段的后门比较密钥字节中存储的密钥匹配，则验证后门访问密钥命令将会解除加密状态（参见[Flash 配置字段说明](#)）。标记 Flash 配置字段偏移地址的列显示了 flash 配置字段中匹配字节的位置。

表 30-68. 验证后门访问密钥命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]	Flash 配置字段偏移地址
0	0x45 (VFYKEY)	
1-3	未使用	
4	密钥字节 0	0x0_0003
5	密钥字节 1	0x0_0002
6	密钥字节 2	0x0_0001
7	密钥字节 3	0x0_0000
8	密钥字节 4	0x0_0007
9	密钥字节 5	0x0_0006
A	密钥字节 6	0x0_0005
B	密钥字节 7	0x0_0004

清除 CCIF 以启用验证后门访问密钥命令后，flash 存储器模块将检查 FSEC[KEYEN]位，以验证是否使能此命令。如果未使能，则 flash 存储器模块将置位 FSTAT[ACCERR]位并终止。如果已使能该命令，则 flash 存储器模块会将 FCCOB 中提供的密钥与 Flash 配置字段中的后门比较密钥进行对比。如果后门密钥匹配，则 FSEC[SEC]字段将更改为解密状态并解除加密。如果后门密钥不匹配，则不会解除加密且所有执行验证后门访问密钥命令的其他尝试将会立即中止，并且 FSTAT[ACCERR]位将（再次）置位为 1，直到 flash 存储器模块发生复位。如果整个 8 字节密钥为全 0 或全 1，则验证后门访问密钥命令将会因访问错误而失败。完成验证后门访问密钥操作之后，CCIF 标志位将置位。

表 30-69. 验证后门访问密钥命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
提供的密钥为全 0 或全 F	FSTAT[ACCERR]

下一页继续介绍此表...

表 30-69. 验证后门访问密钥命令错误处理 (继续)

错误状态	错误位
提供的后门密钥不正确	FSTAT[ACCERR]
后门密钥访问未使能 (参见 FSEC 寄存器说明)	FSTAT[ACCERR]
自上次掉电复位之后, 此命令已启用并且后门密钥不匹配	FSTAT[ACCERR]

30.4.11.11 擦除所有解密模块命令

擦除所有解密模块操作用于擦除所有 flash 存储器, 验证所有存储器内容, 将 Flash 配置字段中的加密字节设为解密状态以及解除 MCU 加密。

表 30-70. 擦除所有解密模块命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x49 (ERSALLU)

清除 CCIF 以启用擦除所有解密模块命令后, flash 存储器模块将擦除所有程序 flash 存储器, 然后再验证是否已全部擦除。

如果 flash 存储器模块验证已正确擦除所有程序 flash 存储器, 则访问控制禁用, 并且可通过将 FSEC[SEC]字段置位于解密状态来解除加密, 并且可通过擦除所有解密模块命令将加密字节 (参见 [Flash 配置字段说明](#)) 设为解密状态。如果擦除或编程验证失败, 则 FSTAT[MGSTAT0]位将会置位。完成擦除所有解密模块操作之后, CCIF 标志位将置位。

由 FXACC 寄存器内容决定的访问控制不会阻止擦除所有解密模块命令的执行。擦除大部分 Flash 存储器后, 包含 Program Once XACC 和 SACC 字段的程序 flash IFR 空间将不会擦除, 因此, Program Once XACC 和 SACC 字段的内容不会更改。FXACC 和 FSACC 寄存器的内容不会受擦除所有解密模块命令执行的影响。完成擦除所有解密模块命令之后, 访问控制将禁用, 直到下一次复位 flash 模块或执行读取 1s 所有模块命令并且失败 (FSTAT[MGSTAT0]置位)。

表 30-71. 擦除所有解密模块命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
擦除或编程验证操作期间发生任何错误	FSTAT[MGSTAT0]

30.4.11.12 读取 1s 所有仅执行区段命令

读取 1s 所有仅执行区段命令检查由 FXACC 寄存器定义的程序 flash 只执行段是否已擦除至指定读取裕量级别，如果适用，在读数通过时将释放 flash 访问控制，即所有数据读取为'1'。

表 30-72. 读取 1s 所有仅执行区段命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x4A (RD1XA)
1	读取-1 裕量选择

在清除 CCIF 以启动读取 1s 所有仅执行区段命令后，flash 存储器模块：

- 根据表 30-73 将读取裕量置位为 1s，
- 检查程序 flash 只执行段的内容是否处于已擦除状态。

如果 flash 存储器模块确认这些段已被擦除，则在下一次复位之前 flash 访问控制将禁用，或者在编程任何只执行段后，读取 1s 所有仅执行区段命令将执行并失败，并且 FSTAT[MGSTAT0]位将会置位。如果读取失败，即所有段并非都处于完全擦除状态，则 FSTAT[MGSTAT0]位将置位。

完成读取 1s 所有仅执行区段操作之后，CCIF 标志位将置位。完成操作之后，MMBSEL 寄存器将被清除。

表 30-73. 读取 1s 所有仅执行区段的裕量级别选择

读取裕量选择	裕量级说明
0x00	使用 1s 的“正常”读取级
0x01	将“用户”裕量应用到正常读取-1 级
0x02	将“出厂”裕量应用到正常读取-1 级

表 30-74. 读取 1s 所有仅执行区段命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
指定了无效的裕量选择	FSTAT[ACCERR]
扇区大于段区	FSTAT[ACCERR]
读取-1s 失败	FSTAT[MGSTAT0]

30.4.11.13 擦除所有仅执行区段命令

擦除所有仅执行区段操作用于擦除 FXACC 寄存器所定义的所有程序 flash 只执行段，验证是否已擦除全部段并解除 flash 访问控制。

表 30-75. 擦除所有仅执行区段命令 FCCOB 要求

FCCOB 编号	FCCOB 内容[7:0]
0	0x4B (ERSXA)

清除 CCIF 以启用擦除所有仅执行区段命令后，flash 存储器模块将擦除所有程序 flash 只执行段，然后再验证是否已擦除全部段。

如果 flash 存储器模块验证所有段已正确擦除，则在下一次复位之前 flash 访问控制将禁用，或者在写入任何只执行段后，读取 1s 所有仅执行区段命令将执行并失败，并且 FSTAT[MGSTAT0]位将会置位。如果在正常 NVM 或特殊 NVM 模式下，任何 XA 受控段受到保护，则擦除所有仅执行区段命令将中止。如果擦除验证失败，则 FSTAT[MGSTAT0]位将置位。完成擦除所有仅执行区段操作之后，CCIF 标志位将置位。

由 FXACC 寄存器内容决定的访问控制不会阻止擦除所有仅执行区段命令的执行。擦除所有 XA 受控段后，包含 Program Once XACC 字段的程序 flash IFR 空间将不会擦除，因此，Program Once XACC 字段的内容不会更改。FXACC 寄存器的内容不会受擦除所有仅执行区段命令执行的影响。

表 30-76. 擦除所有仅执行区段命令错误处理

错误状态	错误位
命令在当前模式/安全状态下不可用	FSTAT[ACCERR]
扇区大于段区	FSTAT[ACCERR]
任何写入的 flash 存储器的，且受 XA 控制的段都处于被保护状态。	FSTAT[FPVIOL]
验证期间发生的任何错误	FSTAT[MGSTAT0]

30.4.12 加密

Flash 存储器模块可根据 FSEC 加密寄存器的内容为 MCU 提供加密信息。

MCU 随后将按照器件的芯片配置详细信息中的定义限制对 flash 存储器资源的访问。复位期间，flash 存储器模块使用从 Flash 配置字段加密字节读取的数据初始化 FSEC（参见 [Flash 配置字段说明](#)）。

FSEC 寄存器中提供以下字段。设置如 [Flash 安全寄存器 \(FTFA_FSEC\)](#) 详细信息中所述。

在 [AN4507: 使用 Kinetis 加密和 Flash 保护功能](#) 中将会进一步讨论 flash 加密功能。请注意，并非此应用笔记中介绍的所有功能在此器件上均可用。

表 30-77. FSEC 寄存器字段

FSEC 字段	说明
KEYEN	后门密钥访问
MEEN	批量擦除功能
FSLACC	Factory Security Level Access
SEC	MCU 加密

30.4.12.1 按模式和加密列出的 Flash 存储器访问

下表概述了对 flash 存储器模块的访问如何受加密和操作模式的影响。

表 30-78. Flash 存储器访问概述

操作模式	芯片加密状态	
	解密	加密
NVM 正常	完整命令集	
NVM 特殊	完整命令集	仅限擦除所有模块、擦除所有模块解密和读取 1s 所有模块命令。

30.4.12.2 更改加密状态

可以通过程序 flash 配置字段的加密字节来永久更改复位外的加密状态。假定开始的模式可用必要的程序 flash 擦除和程序命令，并且包含 flash 配置字段的程序 flash 区域不受保护。如果成功地对 flash 安全字节进行了编程，其新值将在下次芯片复位后生效。

30.4.12.2.1 使用后门密钥解密芯片

可使用后门密钥访问功能解密芯片，该功能需要使用存储在 flash 配置字段中 8 字节后门密钥值的内容（参见 [Flash 配置字段说明](#)）。如果 FSEC[KEYEN]位处于已使能状态，则可运行验证后门访问密钥命令（参见 [验证后门访问密钥命令](#)）；它允许用户提供预期密钥，以便与存储的密钥比较。如果密钥匹配，则 FSEC[SEC]位将会更改以解密此芯片。整个 8 字节密钥不得是全 0 或全 1；也就是说，验证后门访问密钥命令不会接受 0000_0000_0000_0000h 和 FFFF_FFFF_FFFF_FFFFh 为有效比较值。当验证后门访问密钥命令处于活动状态时，则程序 flash 存储器不可用于读取访问，并返回无效数据。

程序 flash 存储器中存储的用户代码必须能够从外部刺激中接收后门密钥。此外部刺激通常是通过其中一个片上串行端口获取的。

如果 KEYEN 位处于已使能状态，可通过下述后门密钥访问序列解密此芯片：

1. 遵循验证后门访问密钥命令的命令序列，如在[验证后门访问密钥命令](#)中的解释
2. 如果验证后门访问密钥命令成功，则会解密此芯片，并且 FSEC[SEC]位将强制为解密状态

提供给验证后门访问密钥命令的所有非法密钥可禁止进一步使用验证后门访问密钥命令。比较失败之后，复位芯片是重新使能验证后门访问密钥命令的唯一方法。

后门密钥正确匹配之后，可通过更改 FSEC[SEC]位解密此芯片。成功执行验证后门访问密钥命令只会更改 FSEC 寄存器中的加密状态。它不会更改 Flash 配置字段 ([Flash 配置字段说明](#)) 中存储的加密字节或密钥。下次复位芯片之后，flash 存储器模块的加密状态将恢复为 Flash 配置字段中的 flash 加密字节。验证后门访问密钥命令序列对程序 flash 保护寄存器中定义的编程和擦除保护没有影响。

如果后门密钥匹配成功，则此解密芯片已完全控制 Flash 配置字段的内容。此芯片可擦除包含 Flash 配置字段的扇区并将 flash 加密字节重新设为解密状态，并将后门密钥更改为任何需要的值。

30.4.13 复位序列

每次系统复位后，flash 存储器模块都会执行一个序列，此序列将为 flash 块配置参数、FPROT、FOPTFSEC、FXACC、FSACC 和 FACNFG 寄存器建立初始值。

FSTAT[CCIF]在整个复位序列期间会被清零。在复位序列期间，flash 存储器模块将暂时停止 CPU 访问。CPU 访问恢复之后，允许执行 Flash 读取。复位序列的完成由 CCIF 置位来标记，这将使能 flash 用户命令。

如果在执行任何 flash 命令时发生复位，那么此命令将立即被中止。此时无法保证所编程字或所擦除扇区/模块的状态。在退出复位后，命令和操作不会自动恢复。

第 31 章 循环冗余校验(CRC)

31.1 简介

循环冗余校验(CRC)模块生成 16/32 位 CRC 码以便进行 CRC 校验。

CRC 模块可以根据提供的 16 位或 32 位多项式和其他参数，来计算 CRC 校验值。

对于 32 位数据，每次都计算该 16/32 位代码。

31.1.1 特性

CRC 模块特性包括：

- 硬件电路 CRC 生成器，包含一个 16 位或者 32 位可编程移位寄存器
- 可编程初始种子和多项式
- 输入数据或输出数据(CRC 结果)可以逐位或逐字节取反。某些 CRC 标准要求提供该选项。以 8 位读取操作访问 CRC 数据寄存器时，无法执行逐字节转置操作。这种情况下，用户软件必须执行逐字节转置功能。
- 可以将最终 CRC 结果反转
- 32 位 CPU 寄存器编程接口

31.1.2 结构框图

下图为 CRC 结构框图。

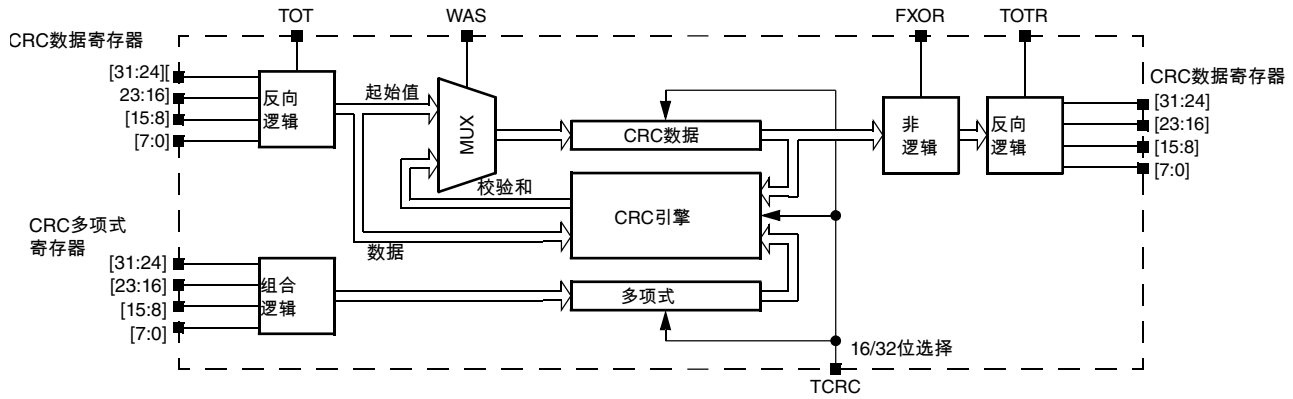


图 31-1. 可编程循环冗余校验(CRC)结构框图

31.1.3 工作模式

不同 MCU 模式会影响 CRC 模块的功能。

31.1.3.1 运行模式

这是基本工作模式。

31.1.3.2 低功耗模式（等待或停止）

MCU 进入低功耗模式后禁用模块时钟，任何进行中的 CRC 计算都会停止。CRC 计算在时钟使能后或通过系统复位退出低功耗模式后恢复。该模块的时钟选通取决于 MCU。

31.2 存储器映射和寄存器说明

CRC 存储器映射

绝对地址（十六进制）	寄存器名称	宽度（单位：位）	访问	复位值	小节/页
4003_2000	CRC 数据寄存器 (CRC_DATA)	32	R/W	FFFF_FFFFh	31.2.1/595
4003_2004	CRC 多项式寄存器 (CRC_GPOLY)	32	R/W	0000_1021h	31.2.2/596
4003_2008	CRC 控制寄存器 (CRC_CTRL)	32	R/W	0000_0000h	31.2.3/596

31.2.1 CRC 数据寄存器 (CRC_DATA)

CRC 数据寄存器包含种子值、数据及校验和。如果设置了 CTRL[WAS]，则对数据寄存器执行的任何写操作都视为种子值。如果 CTRL[WAS]清零，则对数据寄存器进行的任何写操作都被视为用于一般 CRC 计算的数据。

在 16 位 CRC 模式下，不使用 HU 字段和 HL 字段来设定种子值，对这些字段进行读操作会返回不确定的值。在 32 位 CRC 模式中，所有字段均用于设定种子值。

当写入数据时，如果所有字节都是连续的，那么可一次写入 8 位、16 位或 32 位的数值；首先写入的是 MSB 数据数值。

写入所有数据数值后，可从该数据寄存器中读取 CRC 结果。在 16 位 CRC 模式下，LU 字段和 LL 字段提供 CRC 结果。在 32 位 CRC 模式下，所有字段均包含此结果。在任意时间读取此寄存器会返回中间 CRC 值。

地址: 4003_2000h 基准 + 0h 偏移 = 4003_2000h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HU								HL								LU								LL							
W																																
复位	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

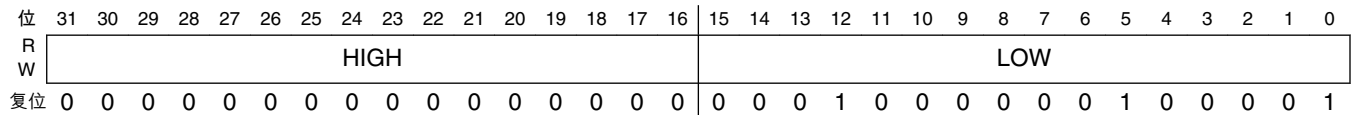
CRC_DATA 字段描述

字段	描述
31–24 HU	CRC 高字段高位字节 在 16 位 CRC 模式 (CTRL[TCRC]为 0) 下，该字段并不用于种子值的编程。在 32 位 CRC 模式 (CTRL[TCRC]为 1) 下，当 CTRL[WAS]为 1 时，写入该字段的值是种子值的一部分。当 CTRL[WAS]为 0 时，写入该字段的数据用于在 16 位 CRC 模式和 32 位 CRC 模式下生成 CRC 校验和。
23–16 HL	CRC 高字段低位字节 在 16 位 CRC 模式(CTRL[TCRC]为 0)下，该字段并不用于设定种子值。在 32 位 CRC 模式(CTRL[TCRC]为 1) 下，当 CTRL[WAS]为 1 时，写入该字段的值是种子值的一部分。当 CTRL[WAS]为 0 时，写入该字段的数据用于在 16 位 CRC 模式和 32 位 CRC 模式下生成 CRC 校验和。
15–8 LU	CRC 低字段高位字节 当 CTRL[WAS]为 1 时，写入该字段的数值是种子值的一部分。当 CTRL[WAS]为 0 时，写入该字段的数据用于生成 CRC 校验和。
LL	CRC 低字段低位字节 当 CTRL[WAS]为 1 时，写入该字段的数值是种子值的一部分。当 CTRL[WAS]为 0 时，写入该字段的数据用于生成 CRC 校验和。

31.2.2 CRC 多项式寄存器 (CRC_GPOLY)

该寄存器含有 CRC 计算所需的多项式值。HIGH 字段为 CRC 多项式的高 16 位，仅在 32 位 CRC 模式下使用。在 16 位 CRC 模式下会忽略对 HIGH 字段的写操作。LOW 字段包含在 16 位 CRC 模式和 32 位 CRC 模式下都使用的 CRC 多项式的低 16 位。

地址: 4003_2000h 基准 + 4h 偏移 = 4003_2004h



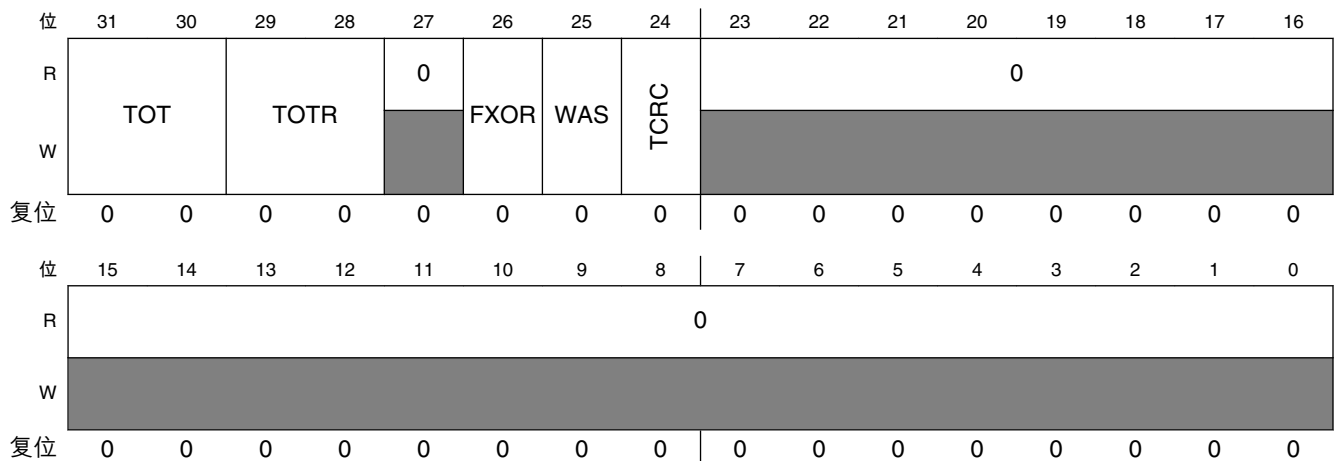
CRC_GPOLY 字段描述

字段	描述
31-16 HIGH	多项式高 16 位 32 位 CRC 模式下可读写 (CTRL[TCRC]为 1)。该字段在 16 位 CRC 模式下不可写 (CTRL[TCRC]为 0)。
LOW	多项式低 16 位 在 32 位 CRC 模式和 16 位 CRC 模式下都可读写。

31.2.3 CRC 控制寄存器 (CRC_CTRL)

该寄存器控制 CRC 模块的配置和工作。开始进行新的 CRC 计算前，相应位必须置位。初始化新的 CRC 计算的方法是：置位 CTRL[WAS]，然后将种子值写入 CRC 数据寄存器。

地址: 4003_2000h 基准 + 8h 偏移 = 4003_2008h



CRC_CTRL 字段描述

字段	描述
31–30 TOT	<p>写入的转置类型</p> <p>定义写入 CRC 数据寄存器的数据的转置配置。有关可用的转置选项，请参见转置特性说明。</p> <p>00 无转置。 01 字节中的位转置；字节不转置。 10 字节中的位和字节均转置。 11 仅字节转置；字节中的位不转置。</p>
29–28 TOTR	<p>读取的转置类型</p> <p>识别从 CRC 数据寄存器读取的数值的转置配置。有关可用的转置选项，请参见转置特性说明。</p> <p>00 无转置。 01 字节中的位转置；字节不转置。 10 字节中的位和字节均转置。 11 仅字节转置；字节中的位不转置。</p>
27 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
26 FXOR	<p>CRC 数据寄存器的补码读取</p> <p>某些 CRC 协议要求最终校验和与 0xFFFFFFFF 或 0xFFFF 进行异或运算。置位该位可能使能对最终 CRC 结果的异或操作。</p> <p>0 读取时不执行异或运算。 1 将最终结果进行异或操作。</p>
25 WAS	<p>作为种子值写入 CRC 数据寄存器</p> <p>置位后，写入 CRC 数据寄存器的值被视为种子值。解除置位后，写入 CRC 数据寄存器的值用作 CRC 计算中的数据。</p> <p>0 写入 CRC 数据寄存器的是数据值。 1 写入 CRC 数据寄存器的是种子值。</p>
24 TCRC	<p>CRC 协议宽度。</p> <p>0 16 位 CRC 协议。 1 32 位 CRC 协议。</p>
保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>

31.3 功能说明

31.3.1 CRC 初始化/重新初始化

要使能 CRC 计算，用户必须对 CRC_CTRL[WAS]、CRC_GPOLY、转置所必需的参数以及 CRC 结果反转进行编程。置位 CRC_CTRL[WAS] 可将种子值编入 CRC_DATA 寄存器。

完成 CRC 计算后，重新置位 CRC_CTRL[WAS]并编入新的或已用过的种子值，可重新初始化模块以进行新的 CRC 计算。所有其他参数必须在编入种子值和随后数据值前置位。

31.3.2 CRC 计算

在 16 位 CRC 模式和 32 位 CRC 模式下，如果所有字节都是连续的，可对数据值进行 8 位、16 位或 32 位的一次性编程。非连续字节可能导致 CRC 计算错误。

31.3.2.1 16 位 CRC

如需计算 16 位 CRC:

1. 清零 CRC_CTRL[TCRC]以使能 16 位 CRC 模式。
2. 按 CRC 计算要求对转置进行编程，并在 CTRL 寄存器中补全选项位。详情请参见[转置特性](#)和[CRC 结果补码](#)。
3. 将 16 位多项式写入 CRC_GPOLY[LOW]字段。CRC_GPOLY[HIGH]字段在 16 位 CRC 模式下不可用。
4. 置位 CRC_CTRL[WAS]以进行起始值的编程。
5. 将 16 位起始值写入 CRC_DATA[LU:LL]。CRC_DATA[HU:HL]不使用。
6. 清零 CRC_CTRL[WAS]以开始写入数据值。
7. 将数据值写入 CRC_DATA[HU:HL:LU:LL]。每次写入数据值时计算 CRC，并将 CRC 中间值结果存回至 CRC_DATA[LU:LL]。
8. 写入所有数值后，从 CRC_DATA[LU:LL]读取最终的 CRC 结果。

转置和补码操作在读取或写入数值的同时执行。详情请参见[转置特性](#)和[CRC 结果补码](#)。

31.3.2.2 32 位 CRC

如需计算 32 位 CRC:

1. 置位 CRC_CTRL[TCRC]以使能 32 位 CRC 模式。
2. 按 CRC 计算要求对转置进行编程，并在 CTRL 寄存器中补全选项位。详情请参见[转置特性](#)和[CRC 结果补码](#)。
3. 将 32 位多项式写入 CRC_GPOLY[HIGH:LOW]。
4. 置位 CRC_CTRL[WAS]以进行种子值的编程。
5. 将 32 位种子值写入 CRC_DATA[HU:HL:LU:LL]。
6. 清零 CRC_CTRL[WAS]以开始写入数据值。

7. 将数据值写入 CRC_DATA[HU:HL:LU:LL]。每次写入数据值时计算 CRC, 并将 CRC 中间值结果存回至 CRC_DATA[HU:HL:LU:LL]。
8. 完成所有数值的写入操作后, 从 CRC_DATA[HU:HL:LU:LL]读取最终的 CRC 结果。CRC 计算逐字节执行, 且需要两个时钟周期完成一次 CRC 计算。

转置和补码操作在读取或写入数值的同时执行。详情请参见[转置特性](#)和[CRC 结果补码](#)。

31.3.3 转置特性

默认情况下, 转置特性未使能。然而, 某些 CRC 标准要求对输入数据"与"或者"或"最终校验和进行转置。用户软件可根据 CRC 标准需要选择单独配置每个转置操作。数据在读写的同时进行转置。

某些协议计算 CRC 时对数据流采用低字节序格式。在这种情况下, 转置特性非常有用, 可以翻转位。该转置选项是 CRC 模块支持的功能类型之一。

31.3.3.1 转置类型

CRC 模块提供可翻转位"与"或者"或"字节的多种转置功能类型, 以便根据采用的 CRC 计算方法分别使用 CTRL[TOT]或 CTRL[TOTR]字段写入输入数据和读取 CRC 结果。

下列转置功能类型可用于对 CRC 数据寄存器进行读写操作:

1. CTRL[TOT]或 CTRL[TOTR]为 00。

不发生转置。

2. CTRL[TOT]或 CTRL[TOTR]为 01。

字节中的位转置, 而字节不转置。

reg[31:0]变为{reg[24:31], reg[16:23], reg[8:15], reg[0:7]}

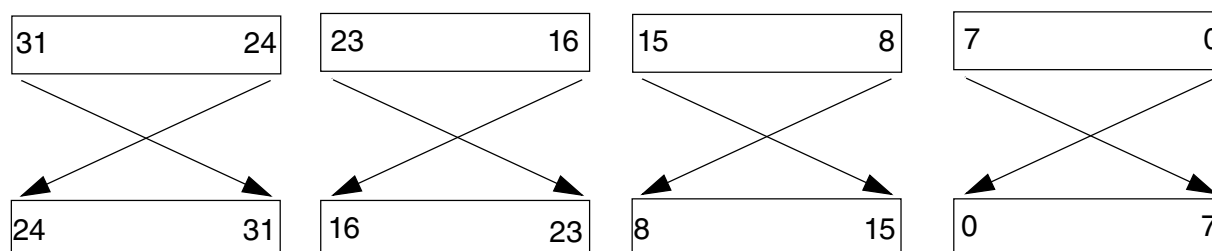


图 31-5. 转置类型 01

3. CTRL[TOT]或 CTRL[TOTR]为 10。

字节中的位和字节均转置。

reg[31:0]变为 = {reg[0:7], reg[8:15], reg[16:23], reg[24:31]}

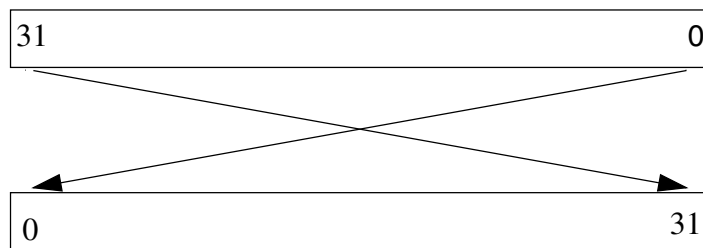


图 31-6. 转置类型 10

4. CTRL[TOT]或 CTRL[TOTR]为 11。

字节转置，但位不转置。

reg[31:0]变为 {reg[7:0], reg[15:8], reg[23:16], reg[31:24]}

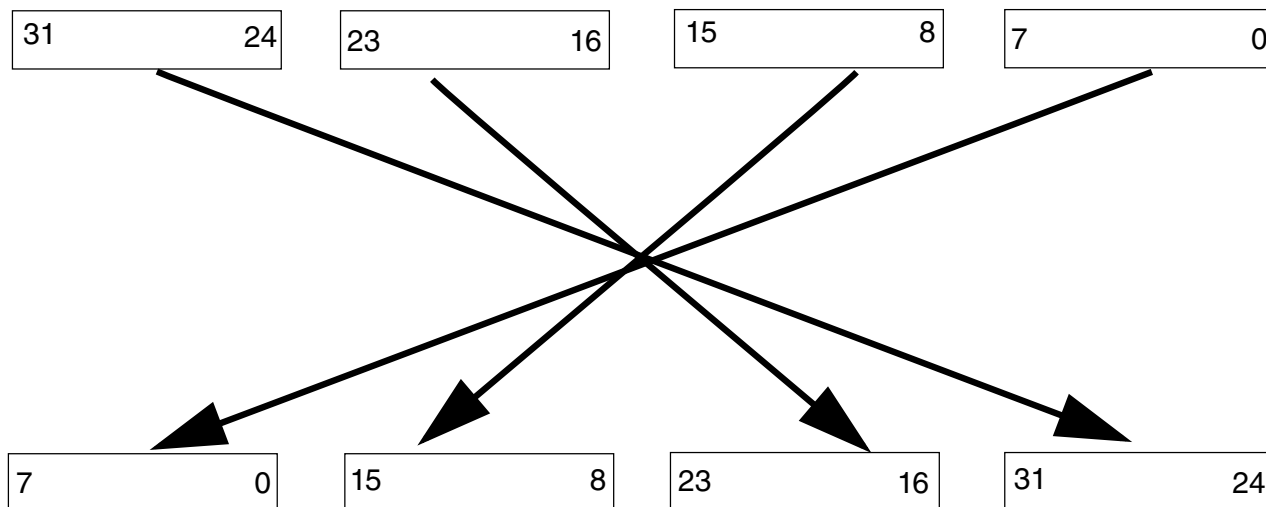


图 31-7. 转置类型 11

注

- 为了对 CRC 数据寄存器进行 8 位或 16 位的写访问，对不使用的字节该数据转置为零(将 32 位作为一个整体)，但对于有效字节仅计算 CRC。
- 读取 CRC 数据寄存器中的 16 位 CRC 结果并使用转置选项 10 和 11 时，转置后的结果数值位于 CRC[HU:HL] 字段。读取 16 位 CRC 结果时，用户软件必须将该情形考虑在内，因此建议优先读取 32 位。

31.3.4 CRC 结果补码

CTRL[FXOR]置位后，对校验和进行补码。CRC 结果补码功能在每次读取 CRC 数据寄存器时将 CRC 数据寄存器中的值进行补码操作。CTRL[FXOR]清零后，对 CRC 数据寄存器进行读操作会访问原始校验和值。

第 32 章

随机数生成加速器 (RNGA)

32.1 简介

本章节介绍随机数生成加速器 RNGA，包括编程模型、功能说明和应用程序信息。在整个章节中，“RNG”和“RNGA”表示相同意义。

32.1.1 概述

RNGA 是一种数字集成电路，可以生成 32 位的随机数。随机位是用移位寄存器生成的，其时钟源于两个自由运行的独立环形振荡器。移位寄存器的配置确保数据具有统计学意义，即数据看起来是随机的。振荡器（其频率不明，相位独立）提供了生成创建随机数据所需熵的方式。RNGA 生成的随机字将被载入一个输出寄存器 (OR) 中。当输出寄存器 (OR) 中的数值无效且 OR 被读取时，会产生一个错误中断 (如果中断使能的话)。如果状态寄存器 (SR) 中的 LVL 字段为 1，则 OR 含有有效的随机数据。

需要注意的是，目前并无已知加密技术可证明这是一种生成随机数据的安全方法。事实上，如果在加密应用中直接使用 RNGA 的输出，则可能存在针对该随机数生成器的攻击。该攻击基于内部移位寄存器的线性度。因此，强烈建议将此模块生成的随机数据作为熵源，向经 NIST 批准（基于 DES 或 SHA-1）的伪随机数生成器提供输入种子，该伪随机数生成器在 *NIST Fips Pub 186-2* 附录 3 和 *NIST Fips Pub SP 800-90* 中定义。

为了使此输入种子的熵最大化。在从 RNGA 中以硬件允许的速度抽取数据时，每个 32 位字都有约一个或两个额外的熵位。该字的任一位都含有该熵。因此，用作熵源时，应为需要的每个熵位生成一个随机数，并保留每个字的最低有效位（任何等效位）。然后应丢失每个随机数的剩余部分。这样，即使完全掌握了 RNGA 的内部状态以及以前的所有随机数，攻击者也不可能预测抽取位的值。

另外还可以将 RNGA 与其他熵源一起配合使用，以生成伪随机算法所需要的种子。组合起来以创建种子的随机源越多，越好。以下列出了可与此模块输出容易组合起来的源：

- 当前时间（采用可能的最高精度）
- 可视为“随机”的实时系统输入
- 用户直接提供的其他熵

32.2 工作模式

RNGA 支持下列工作模式。

表 32-1. RNGA 支持的工作模式

模式	说明
正常	环形振荡器时钟有效；RNGA 从时钟生成熵（随机）并将其存储在移位寄存器中。
睡眠	环形振荡器时钟无效；RNGA 不生成熵。

32.2.1 进入正常模式

要进入正常模式，请向 CR[SLP]写入 0。

32.2.2 进入睡眠模式

要进入睡眠模式，请向 CR[SLP]写入 1。

32.3 存储器映射和寄存器定义

本节描述各 RNGA 寄存器。

RNG 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4002_9000	RNGA 控制寄存器 (RNG_CR)	32	R/W	0000_0000h	32.3.1/605
4002_9004	RNGA 状态寄存器 (RNG_SR)	32	R	0001_0000h	32.3.2/607
4002_9008	RNGA 熵寄存器 (RNG_ER)	32	W (始终读 0)	0000_0000h	32.3.3/609
4002_900C	RNGA 输出寄存器 (RNG_OR)	32	R	0000_0000h	32.3.4/609

32.3.1 RNGA 控制寄存器 (RNG_CR)

控制 RNGA 的工作。

地址: 4002_9000h 基准 + 0h 偏移 = 4002_9000h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	[阴影区域]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SLP	0	INTM	HA	GO
W	[阴影区域]											CLRI	[阴影区域]	[阴影区域]	[阴影区域]	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RNG_CR 字段描述

字段	描述
31-5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
4 SLP	睡眠 指定 RNGA 是处于睡眠模式还是正常模式。 注: 也可通过读取 DOZE 信号进入睡眠模式。 0 正常模式 1 睡眠 (低功耗) 模式
3 CLRI	清零中断 通过复位错误中断指示器(SR[ERRI])清零中断。 0 不清零中断。 1 清零中断。当将 1 写入此字段时, RNGA 会复位错误中断指示器(SR[ERRI])。此位始终读为 0。
2 INTM	中断掩码 当发生 OR 下溢条件时, 屏蔽中断控制器错误中断的触发。

下一页继续介绍此表...

RNG_CR 字段描述 (继续)

字段	描述
	<p>当读取 OR[RANDOUT]且 SR[OREG_LVL]=0 时，就会出现 OR 下溢条件。参见输出寄存器(OR)说明。</p> <p>0 未屏蔽 1 已屏蔽</p>
1 HA	<p>高可靠</p> <p>使能安全违规通知 (通过 SR[SECV])。</p> <p>当读取 OR[RANDOUT]且 SR[OREG_LVL]=0 时，就会出现安全违规。</p> <p>注：此字段具有粘滞性。在使能安全违规之后，必须复位 RNGA 以再次禁用它们。</p> <p>0 已禁用 1 已使能</p>
0 GO	<p>GO</p> <p>指定是否使能随机数据生成和加载 (进 OR[RANDOUT])。</p> <p>注：此字段具有粘滞性。必须复位 RNGA，才从阻止 RNGA 将数据载入 OR[RANDOUT]。</p> <p>0 已禁用 1 已使能</p>

32.3.2 RNGA 状态寄存器 (RNG_SR)

指示 RNGA 的状态。此寄存器为只读。

地址: 4002_9000h 基准 + 4h 偏移 = 4002_9004h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								OREG_SIZE							
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OREG_LVL								0		SLP	ERRI	ORU	LRS	SECV	
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RNG_SR 字段描述

字段	描述
31-24 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
23-16 OREG_SIZE	输出寄存器大小 以其可以保存的 32 位随机数据字数量，指示输出(OR)寄存器的大小。 1 一个字 (此值固定)
15-8 OREG_LVL	输出寄存器电平 指示处于 OR[RANDOUT]的随机数据数，该数指示 OR[RANDOUT]是否有效。 注: 如果在 SR[OREG_LVL]不为 0 时读取 OR[RANDOUT]，则会返回 OR[RANDOUT]中所含随机数的内容，并且 RNGA 会将 0 同时写入 OR[RANDOUT]和 SR[OREG_LVL]。 0 无字 (空) 1 一个字 (有效)
7-5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

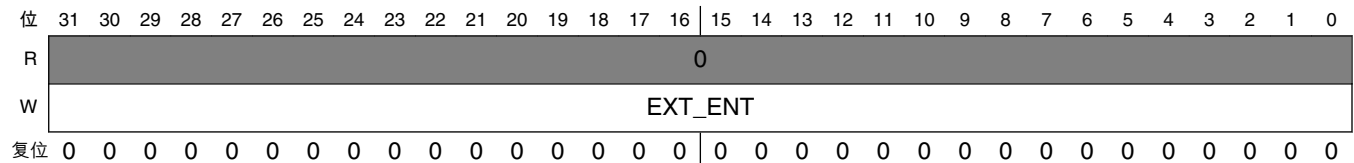
RNG_SR 字段描述 (继续)

字段	描述
4 SLP	<p>睡眠</p> <p>指定 RNGA 是处于睡眠模式还是正常模式。</p> <p>注：也可通过读取 DOZE 信号进入睡眠模式。</p> <p>0 正常模式 1 睡眠 (低功耗) 模式</p>
3 ERRI	<p>错误中断</p> <p>指示自最后清零错误中断(CR[CLRI])或 RNGA 复位以来，是否发生了 OR 下溢条件，无论错误中断是否屏蔽(CR[INTM])。</p> <p>当读取 OR[RANDOUT]且 SR[OREG_LVL]=0 时，就会出现 OR 下溢条件。</p> <p>注：复位错误中断指示器 (通过 CR[CLRI]) 之后，RNGA 将把 0 写入此寄存器。</p> <p>0 无下溢 1 下溢</p>
2 ORU	<p>输出寄存器下溢</p> <p>指示自最后读取此寄存器(SR)或 RNGA 复位以来，是否发生了 OR 下溢条件，无论错误中断是否屏蔽(CR[INTM])。</p> <p>当读取 OR[RANDOUT]且 SR[OREG_LVL]=0 时，就会出现 OR 下溢条件。</p> <p>注：读取此寄存器之后，RNGA 会将 0 写入此字段。</p> <p>0 无下溢 1 下溢</p>
1 LRS	<p>上次读取状态</p> <p>指示对 OR[RANDOUT]的最近读取操作是否导致了 OR 下溢条件，无论错误中断是否屏蔽(CR[INTM])。</p> <p>当读取 OR[RANDOUT]且 SR[OREG_LVL]=0 时，就会出现 OR 下溢条件。</p> <p>注：读取此寄存器之后，RNGA 会将 0 写入此字段。</p> <p>0 无下溢 1 下溢</p>
0 SECV	<p>安全违规</p> <p>仅用于高可靠性使能时(CR[HA])。指示发生了安全违规。</p> <p>注：此字段具有粘滞性。要清零 SR[SECV]，必须复位 RNGA。</p> <p>0 无安全违规 1 安全违规</p>

32.3.3 RNGA 熵寄存器 (RNG_ER)

指定一个熵值，RNGA 用此值与其环形振荡器一起为其伪随机算法提供种子。这是一个只写寄存器；读取操作会返回全零。

地址: 4002_9000h 基准 + 8h 偏移 = 4002_9008h



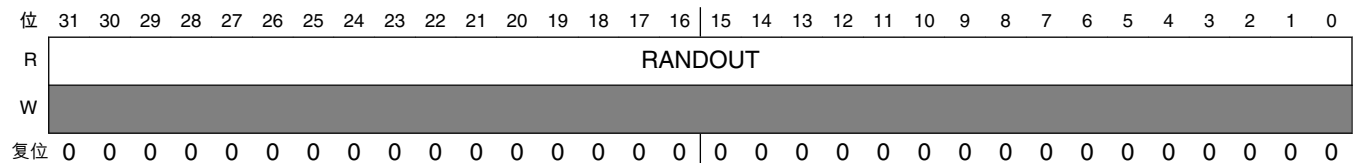
RNG_ER 字段描述

字段	描述
EXT_ENT	外部熵 指定一个熵值，RNGA 用此值与其环形振荡器一起为其伪随机算法提供种子。 注：为此字段指定值为可选操作，但建议这样做。可以在操作过程中随时写入此字段。

32.3.4 RNGA 输出寄存器 (RNG_OR)

存储 RNGA 生成的随机数据字。

地址: 4002_9000h 基准 + Ch 偏移 = 4002_900Ch



RNG_OR 字段描述

字段	描述
RANDOUT	随机输出 存储 RNGA 生成的随机数据字。这是一个只读字段。 注：在读取 RANDOUT 之前，要确保其是有效的(SR[OREG_LVL]=1)。

RNG_OR 字段描述 (继续)

字段	描述
0	无效数据 (如果在其为 0 且 SR[OREG_LVL]为 0 时读取此寄存器, 则 RNGA 会将 1 写入 SR[ERRI]、SR[ORU]和 SR[LRS]; 当错误中断未屏蔽(CR[INTM]=0)时, RNGA 也会认定中断控制器的错误中断请求)。
所有其他值	有效数据 (如果在 SR[OREG_LVL]不为 0 时读取此字段, RNGA 会返回 RANDOUT, 然后将 0 写入此字段和 SR[OREG_LVL])。

32.4 功能说明

此为 RNGA 的结构框图。

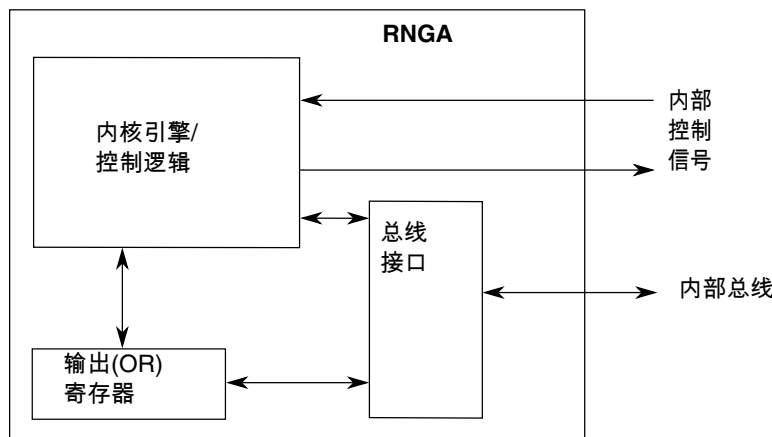


图 32-5. RNG-A 结构框图

32.4.1 输出(OR)寄存器

输出寄存器 (OR) 为内核引擎/控制逻辑生成的随机数据提供临时存储空间。状态寄存器 (SR) 允许用户通过 SR[OREG_LVL]监控 OR 中是否存在有效随机数据。

如果在 OR 中含有有效随机数据 (标志为 SR[OREG_LVL] = 1) 时读取 OR, 则 OR 和 SR[OREG_LVL]均清零。如果用户在 OR 为空时读取 OR, 则 RNGA 返回全零, 并且如果中断使能, RNGA 会向中断控制器发出请求。轮询 SR[OREG_LVL]对于确保在读取 OR 之前存在随机值非常重要。

32.4.2 内核引擎/控制逻辑

此数据块含有 RNGA 的控制逻辑以及用于生成随机数据的内核引擎。

32.4.2.1 控制逻辑

控制逻辑含有 RNGA 的地址解码器、所有可寻址寄存器和控制状态机。此数据块负责与外设接口和输出(OR)寄存器接口的通信,同时控制用于生成随机数据的内核引擎。此数据块的通用功能如下:

复位后, RNGA 工作于正常模式, 具体如下:

1. 内核引擎生成熵, 并将其存储在移位寄存器中。
2. 通过加载 CR[GO]使能随机数据生成之后, 内核引擎每 256 个时钟周期都会生成一个新的随机数据字。如果 SR[OREG_LVL] = 0, 则控制块将新的随机数据载入 OR 并使 SR[OREG_LVL] = 1; 否则, 新数据将被丢弃。

32.4.2.2 内核引擎

内核引擎块含有用于生成随机数据的逻辑。内核引擎中的逻辑含有内部移位寄存器以及用于生成两个基于振荡器的时钟的逻辑。控制逻辑决定移位寄存器的配置方式以及振荡器时钟的开启时间。

32.5 初始化/应用信息

RNGA 一般按以下预期方式工作:

1. 复位/初始化
2. 将 1 写入 CR[INTM]、CR[HA]和 CR[GO]。
3. 轮询 SR[OREG_LVL], 直到其不为 0 为止。
4. 当 SR[OREG_LVL]不为 0 时, 从 OR[RANDOUT]读取可用的随机数据。
5. 根据需要重复步骤 3 和 4。

有关应用信息, 请参阅[概述](#)。

第 33 章 模数转换器 (ADC)

33.1 此模块的芯片实现细节

33.1.1 ADC 实例化信息

此器件包含一个 ADC。

33.1.1.1 ADC 通道数

器件上的 ADC 通道数取决于特定器件封装的引脚配置。有关特定封装可用 ADC 通道数的详细信息，请参考本 MCU 的信号复用章节。

33.1.2 ADC 的 DMA 支持

应用可能需要对 ADC(4K 样本/秒) 持续取样，会给 CPU 带来较大的负载。尽管使用 PDB 来触发 ADC 可能减少 CPU 的部分负载，但在 ADC 采样速度非常高或者在 PDB 被绕过时，ADC 支持 DMA 请求功能可获得更高性能。在转换完成时，ADC 可触发 DMA (通过 DMA 请求)。

33.1.3 ADCx 连接/通道分配

注

如以下各节所示，每个 ADCx_DPx 输入和某些 ADCx_DMx 输入可以在单端模式下以单端 ADC 通道的方式运行。

33.1.3.1 ADC0 通道分配

对于 100 引脚封装:

表 33-1. ADC0 分配

ADC 通道 (SC1n[ADCH])	通道	输入信号 (SC1n[DIFF]= 1)	输入信号 (SC1n[DIFF]= 0)
00000	DAD0	ADC0_DP0 和 ADC0_DM0	ADC0_DP0
00001	DAD1	ADC0_DP1 和 ADC0_DM1	ADC0_DP1
00010	DAD2	ADC0_DP2 和 ADC0_DM2	ADC0_DP2
00011	DAD3	ADC0_DP3 和 ADC0_DM3	ADC0_DP3
00100	AD4a	保留	ADC0_SE4a
00101 ¹	AD5a	保留	ADC0_SE5a
00110 ¹	AD6a	保留	ADC0_SE6a
00111 ¹	AD7a	保留	ADC0_SE7a
00100 ¹	AD4b	保留	ADC0_SE4b
00101 ¹	AD5b	保留	ADC0_SE5b
00110 ¹	AD6b	保留	ADC0_SE6b
00111 ¹	AD7b	保留	ADC0_SE7b
01000	AD8	保留	ADC0_SE8
01001	AD9	保留	ADC0_SE9
01010	AD10	保留	保留
01011	AD11	保留	保留
01100	AD12	保留	ADC0_SE12
01101	AD13	保留	ADC0_SE13
01110	AD14	保留	ADC0_SE14
01111	AD15	保留	ADC0_SE15
10000	AD16	保留	VBAT
10001	AD17	保留	ADC0_SE17
10010	AD18	保留	ADC0_SE18
10011	AD19	保留	ADC0_DM0
10100	AD20	保留	ADC0_DM1
10101	AD21	保留	保留
10110	AD22	保留	保留
10111	AD23	保留	12 位 DAC0 输出/ADC0_SE23
11000	AD24	保留	保留
11001	AD25	保留	保留
11010	AD26	温度传感器(Diff)	温度传感器(S.E)
11011	AD27	带隙基准电压	带隙(S.E) ²
11100	AD28	保留	保留
11101	AD29	-VREFH (Diff)	VREFH (S.E)

下一页继续介绍此表...

表 33-1. ADC0 分配 (继续)

ADC 通道 (SC1n[ADCH])	通道	输入信号 (SC1n[DIFF]= 1)	输入信号 (SC1n[DIFF]= 0)
11110	AD30	保留	VREFL
11111	AD31	模块已禁用	模块已禁用

1. ADCx_CFG2[MUXSEL]位在 ADCx_SEn 通道 a 和 b 之间选择。详情请参见 ADC 一章中关于 MUXSEL 的说明。
2. 在从该 ADC 通道读取之前,要确保通过置位 PMC_REGSC[BGBE]位来使能带隙缓冲器。有关带隙电压(V_{BG})规格,请参见器件数据手册。

对于 64 引脚封装:

表 33-2. ADC0 分配

ADC 通道 (SC1n[ADCH])	通道	输入信号 (SC1n[DIFF]= 1)	输入信号 (SC1n[DIFF]= 0)
00000	DAD0	ADC0_DP0 和 ADC0_DM0	ADC0_DP0
00001	DAD1	保留	ADC0_DP1
00010	DAD2	保留	保留
00011	DAD3	ADC0_DP3 和 ADC0_DM3	ADC0_DP3
00100	AD4a	保留	ADC0_SE4a
00101 ¹	AD5a	保留	ADC0_SE5a
00110 ¹	AD6a	保留	保留
00111 ¹	AD7a	保留	保留
00100 ¹	AD4b	保留	ADC0_SE4b
00101 ¹	AD5b	保留	ADC0_SE5b
00110 ¹	AD6b	保留	ADC0_SE6b
00111 ¹	AD7b	保留	ADC0_SE7b
01000	AD8	保留	ADC0_SE8
01001	AD9	保留	ADC0_SE9
01010	AD10	保留	保留
01011	AD11	保留	保留
01100	AD12	保留	ADC0_SE12
01101	AD13	保留	ADC0_SE13
01110	AD14	保留	ADC0_SE14
01111	AD15	保留	ADC0_SE15
10000	AD16	保留	VBAT
10001	AD17	保留	保留
10010	AD18	保留	保留
10011	AD19	保留	ADC0_DM0
10100	AD20	保留	保留
10101	AD21	保留	保留
10110	AD22	保留	保留

下一页继续介绍此表...

表 33-2. ADC0 分配 (继续)

ADC 通道 (SC1n[ADCH])	通道	输入信号 (SC1n[DIFF]= 1)	输入信号 (SC1n[DIFF]= 0)
10111	AD23	保留	12 位 DAC0 输出/ADC0_SE23
11000	AD24	保留	保留
11001	AD25	保留	保留
11010	AD26	温度传感器(Diff)	温度传感器(S.E)
11011	AD27	带隙基准电压	带隙(S.E) ²
11100	AD28	保留	保留
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	保留	VREFL
11111	AD31	模块已禁用	模块已禁用

1. ADCx_CFG2[MUXSEL]位在 ADCx_SEn 通道 a 和 b 之间选择。详情请参见 ADC 一章中关于 MUXSEL 的说明。
2. 在从该 ADC 通道读取之前，要确保通过置位 PMC_REGSC[BGBE]位来使能带隙缓冲器。有关带隙电压(V_{BG})规格，请参见器件数据手册。

33.1.4 ADC 通道 MUX 选择

下图显示了如何通过 MUX 多路选择器来实现 ADCx_SEn 通道 ADCx_SEn a 和 b 的分配。要在可选通道集之间选择，请参考 ADCx_CFG2[MUXSEL]位设置了解详情。

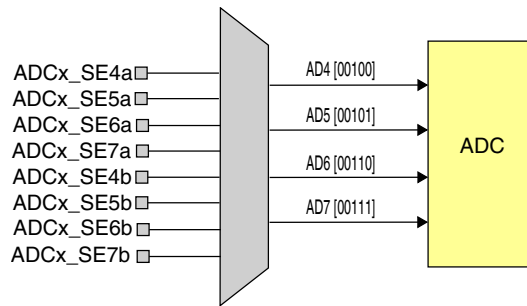


图 33-1. ADCx_SEn 通道 a 和 b 的选择

33.1.5 ADC 参考电压选项

ADC 支持以下 参考电压选项:

- VREFH/VREFL - 主要参考电压选项

ADCx_SC2[REFSEL] 为 ADC 参考电压选择位。更多详情参见 ADC 章节的 REFSEL 说明。

33.1.6 VBAT 与 ADC 输入通道的连接

VBAT 电源输入可以转换，作为 ADC0 通道 16 的单端输入。当 VBAT 大于所选参考电压时，转换结果将显示满量程结果（16 位工作模式下为~0xFFFF）。在测量 VBAT 电压电平时，ADC 应配置为长采样时间（ADC0_CFG1[ADLSMP]=1，ADC0_CFG2[ADLSTS]=00）。

33.1.7 ADC 触发器

ADC 支持软件和硬件触发器。用于触发 ADC 的主要硬件机制为 PDB。PDB 本身可由其他外设触发。例如：RTC（警报，秒）信号连接至 PDB。PDB 输入触发器可以接收 RTC（警报/秒）触发器，强制 ADC 转换进入运行模式（此处 PDB 必须处于使能状态）。另一方面，ADC 可以在低功耗模式下进行转换，不由 PDB 触发。这样就允许 ADC 在低功耗模式下进行转换，并把输出存入结果寄存器中。在将系统从低功耗模式唤醒的结果寄存器中，当数据准备就绪时，ADC 会生成中断。也可利用 SIM_SOPT7 寄存器中的 ADCxTRGSEL 位绕过 PDB。

表 33-3. ADC 可选触发器选项

SIM_SOPT7[ADCxTRGSEL]	触发源
0000	PDB 外部触发器引脚输入 (PDB0_EXTRG)
0001	高速比较器 0 输出
0010	保留
0011	保留
0100	PIT 触发器 0
0101	PIT 触发器 1
0110	PIT 触发器 2
0111	PIT 触发器 3
1000	TPM0 溢出
1001	TPM1 溢出
1010	TPM2 溢出
1011	保留
1100	RTC 警报
1101	RTC 秒数
1110	LPTMR 触发器
1111	TPM1 通道 0 (A 预触发器) 和通道 1 (B 预触发器)

要了解触发器在不同模式的工作情况，请参见《电源管理》一章。

33.1.8 ADC 转换时钟选项

ADC 有多个输入时钟源。具体选择由 ADCx_CFG1[ADICLK]位域决定。下表列出了针对该位域的具体芯片时钟分配情况。

注

ALTCLK 选项只能在 OSCERCLK 处于 MHz 范围之内时才可使用。OSCERCLK 处于 kHz 范围的系统有低于最小 ADC 时钟工作频率的可选时钟源。

表 33-4. ADC 转换时钟选项

ADCx_CFG1[ADICLK]	ADC 定义选择	芯片时钟	注释
00	总线时钟	总线时钟	
01	ALTCLK2	IRC48MCLKIRC48MCLK	注意 ¹
10	ALTCLK	OSCERCLK	注释 ¹
11	异步时钟(ADACK)	N/A—源于 ADC 模块内部	注释

1. 对于仅限计算模式下的 ADC 操作，PSTOP1、停止和 VLPS、ADACK 以及备用时钟源均是允许的时钟源。但请注意，ALTCLK2 被强制禁用，因此在 VLPS 下不可用。

33.1.9 ADC 低功耗模式

此表所示为 ADC 低功耗模式和对应的芯片低功耗模式。

表 33-5. ADC 低功耗模式

模块模式	芯片模式
等待	WAIT, VLPW
常规停止模式	STOP, VLPS
低功耗停止模式	LLS, VLLS3, VLLS2, VLLS1, VLLS0

33.2 简介

16 位模数转换器 (ADC) 是一种设计用于集成式微控制器片上系统操作的逐次逼近型 ADC。

注

有关特定芯片的工作模式，请参见器件的电源管理信息。

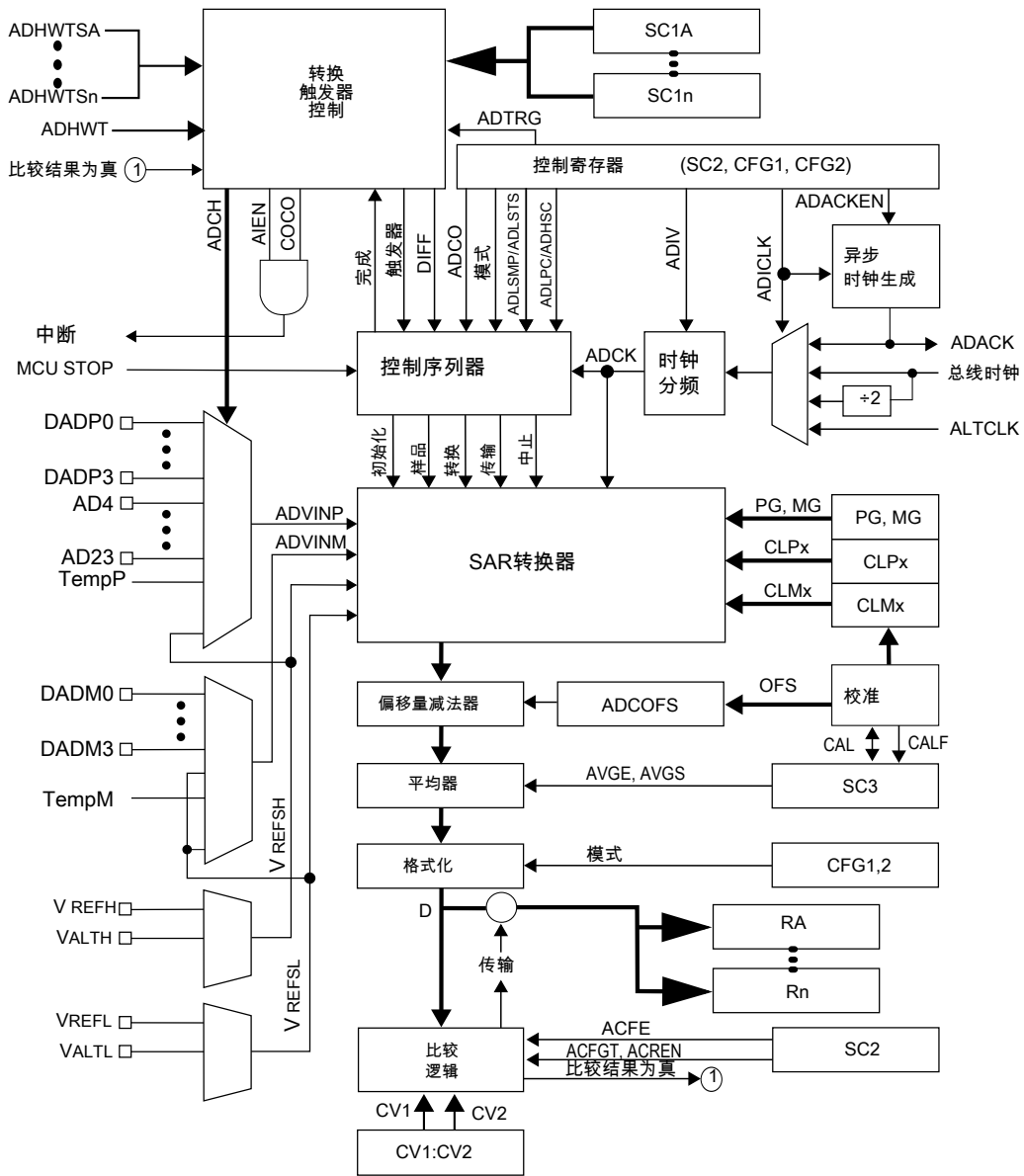
33.2.1 特性

ADC 模块的特性如下:

- 采用最高 16 位分辨率的线性逐次逼近算法
- 最多 4 对差分和 24 个单端外部模拟输入
- 输出模式:
 - 16 位、13 位、11 位和 9 位差分模式
 - 16 位、12 位、10 位和 8 位单端模式
- 差分模式输出 2 的补码 16 位符号扩展格式
- 单端模式输出右对齐的无符号格式
- 单次或连续转换 (单次转换后自动返回到空闲状态)
- 可配置采样时间和转换速度/功耗
- 转换完成/硬件均值完成标志和中断
- 可从最多 4 个来源中选择输入时钟
- 低功耗工作模式可降低噪声
- 异步时钟源, 可降低工作噪声 (带时钟输出选项)
- 带硬件通道选择的可选硬件转换触发器
- 自动与范围内或范围外的设定值进行比较 (小于、大于或等于), 根据结果产生中断
- 温度传感器
- 硬件平均功能
- 可选电压基准: 外部或备用
- 自校准模式

33.2.2 结构框图

下图为 ADC 模块的结构框图。



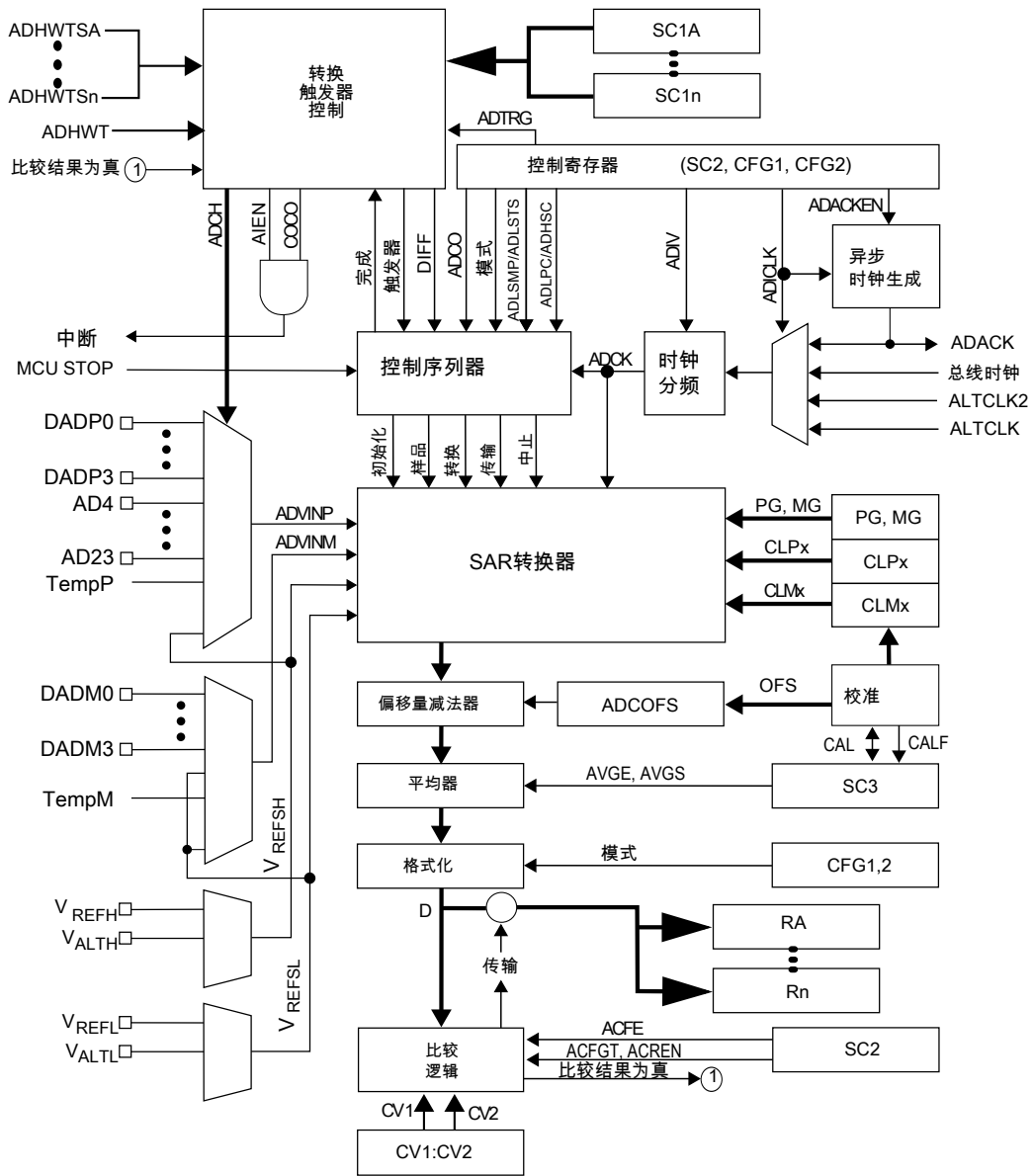


图 33-2. ADC 结构框图

33.3 ADC 信号说明

该 ADC 模块支持高达 4 对差分输入和高达 24 个单端输入。

每个差分对要求两个输入 DADPx 和 DADMx。ADC 还要求四个电源/基准/接地连接。

注

有关此器件支持的通道数量以及 ADC 模块中其他芯片特性输入的相关信息，请参见芯片特性的 ADC 配置信息。

表 33-6. ADC 信号说明

信号	说明	I/O
DADP3–DADP0	差分模拟通道输入	I
DADM3–DADM0	差分模拟通道输入	I
AD n	单端模拟通道输入	I
V _{REFSH}	参考电压（高）	I
V _{REFSL}	参考电压（低）	I
V _{DDA}	模拟电源	I
V _{SSA}	模拟接地	I

33.3.1 模拟电源(V_{DDA})

ADC 模拟部分将 V_{DDA} 用作其电源连接。在某些封装中，V_{DDA} 内部连接到 V_{DD}。如果可以外部连接，则将 V_{DDA} 引脚连接到与 V_{DD} 相同的电势。可能必须进行外部滤波确保 V_{DDA} 干净以便获得良好的结果。

33.3.2 模拟接地(V_{SSA})

ADC 模拟部分将 V_{SSA} 用作其接地连接。在某些封装中，V_{SSA} 内部连接到 V_{SS}。如果可以外部连接，则将 V_{SSA} 引脚连接到与 V_{SS} 相同的电势。

33.3.3 参考电压选择

V_{REFSH} 和 V_{REFSL} 为 ADC 模块的高电平参考电压和低电平参考电压。

可将 ADC 配置为接受 V_{REFSH} 和 V_{REFSL} 的两个参考电压对之一。每对含有一个正参考电压，该电压必须在最小高电平参考电压和 V_{DDA} 之间，以及一个接地参考电压，其电位必须与 V_{SSA} 相同。两对分别为外部 (V_{REFH} 和 V_{REFL}) 以及备用 (V_{ALTH} 和 V_{ALTTL}) 电压对。这些参考电压是通过 SC2[REFSEL]选择的。备用 V_{ALTH} 和 V_{ALTTL} 参考电压对可以选择额外的外部引脚或内部源，具体取决于 MCU 配置。有关此 MCU 特定的参考电压，请参见芯片配置信息。

在部分封装中，V_{REFH} 连接至 V_{DDA}，V_{REFL} 连接至 V_{SSA}。如果采用外部源，则正参考电压可以连接至与 V_{DDA} 相同的电位，也可以由外部源驱动至最小高电平参考电压与 V_{DDA} 电位之间的某个电平。V_{REFH} 不得超过 V_{DDA}。将接地参考电压连接至与 V_{SSA} 相同的电压电位。

33.3.4 模拟通道输入(ADx)

ADC 模块最多支持 24 个单端模拟输入。当 SC1n[DIFF]为低电平时，通过 SC1[ADCH]通道选择位选择单端输入进行转换。

33.3.5 差分模拟通道输入(DADx)

ADC 模块最多支持 4 个差分模拟通道输入。每个差分模拟输入为一对外部引脚，DADPx 和 DADMx，二者互为参考，以提供最精确的模拟转数字读数。当 SC1n[DIFF]为高时，选择一个差分输入以通过 SC1[ADCH]进行转换。如果 SC1n[DIFF]为低，所有 DADPx 输入均可用作单端输入。在某些 MCU 配置中，如果 SC1n[DIFF]为低电平，有些 DADMx 输入也可用作单端输入。有关此器件特有的 ADC 连接，请参见芯片特性的 ADC 信息。

33.4 存储器映射和寄存器定义

本节描述各 ADC 寄存器。

ADC 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4003_B000	ADC 状态和控制寄存器 1 (ADC0_SC1A)	32	R/W	0000_001Fh	33.4.1/624
4003_B004	ADC 状态和控制寄存器 1 (ADC0_SC1B)	32	R/W	0000_001Fh	33.4.1/624
4003_B008	ADC 配置寄存器 1 (ADC0_CFG1)	32	R/W	0000_0000h	33.4.2/627
4003_B00C	ADC 配置寄存器 2 (ADC0_CFG2)	32	R/W	0000_0000h	33.4.3/628
4003_B010	ADC 数据结果寄存器 (ADC0_RA)	32	R	0000_0000h	33.4.4/629
4003_B014	ADC 数据结果寄存器 (ADC0_RB)	32	R	0000_0000h	33.4.4/629
4003_B018	比较值寄存器 (ADC0_CV1)	32	R/W	0000_0000h	33.4.5/630
4003_B01C	比较值寄存器 (ADC0_CV2)	32	R/W	0000_0000h	33.4.5/630
4003_B020	状态和控制寄存器 2 (ADC0_SC2)	32	R/W	0000_0000h	33.4.6/631
4003_B024	状态和控制寄存器 3 (ADC0_SC3)	32	R/W	0000_0000h	33.4.7/633
4003_B028	ADC 偏移校正寄存器 (ADC0_OFS)	32	R/W	0000_0004h	33.4.8/634
4003_B02C	ADC 同相端增益寄存器 (ADC0_PG)	32	R/W	0000_8200h	33.4.9/634
4003_B030	ADC 反相端增益寄存器 (ADC0_MG)	32	R/W	0000_8200h	33.4.10/635
4003_B034	ADC 同相端通用校准值寄存器 (ADC0_CLPD)	32	R/W	0000_000Ah	33.4.11/636
4003_B038	ADC 同相端通用校准值寄存器 (ADC0_CLPS)	32	R/W	0000_0020h	33.4.12/636
4003_B03C	ADC 同相端通用校准值寄存器 (ADC0_CLP4)	32	R/W	0000_0200h	33.4.13/637

下一页继续介绍此表...

ADC 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4003_B040	ADC 同相端通用校准值寄存器 (ADC0_CLP3)	32	R/W	0000_0100h	33.4.14/ 637
4003_B044	ADC 同相端通用校准值寄存器 (ADC0_CLP2)	32	R/W	0000_0080h	33.4.15/ 638
4003_B048	ADC 同相端通用校准值寄存器 (ADC0_CLP1)	32	R/W	0000_0040h	33.4.16/ 638
4003_B04C	ADC 同相端通用校准值寄存器 (ADC0_CLP0)	32	R/W	0000_0020h	33.4.17/ 639
4003_B054	ADC 反相端通用校准值寄存器 (ADC0_CLMD)	32	R/W	0000_000Ah	33.4.18/ 639
4003_B058	ADC 反相端通用校准值寄存器 (ADC0_CLMS)	32	R/W	0000_0020h	33.4.19/ 640
4003_B05C	ADC 反相端通用校准值寄存器 (ADC0_CLM4)	32	R/W	0000_0200h	33.4.20/ 640
4003_B060	ADC 反相端通用校准值寄存器 (ADC0_CLM3)	32	R/W	0000_0100h	33.4.21/ 641
4003_B064	ADC 反相端通用校准值寄存器 (ADC0_CLM2)	32	R/W	0000_0080h	33.4.22/ 641
4003_B068	ADC 反相端通用校准值寄存器 (ADC0_CLM1)	32	R/W	0000_0040h	33.4.23/ 642
4003_B06C	ADC 反相端通用校准值寄存器 (ADC0_CLM0)	32	R/W	0000_0020h	33.4.24/ 642

33.4.1 ADC 状态和控制寄存器 1 (ADCx_SC1n)

SC1A 用于软件和硬件触发的 ADC 操作。

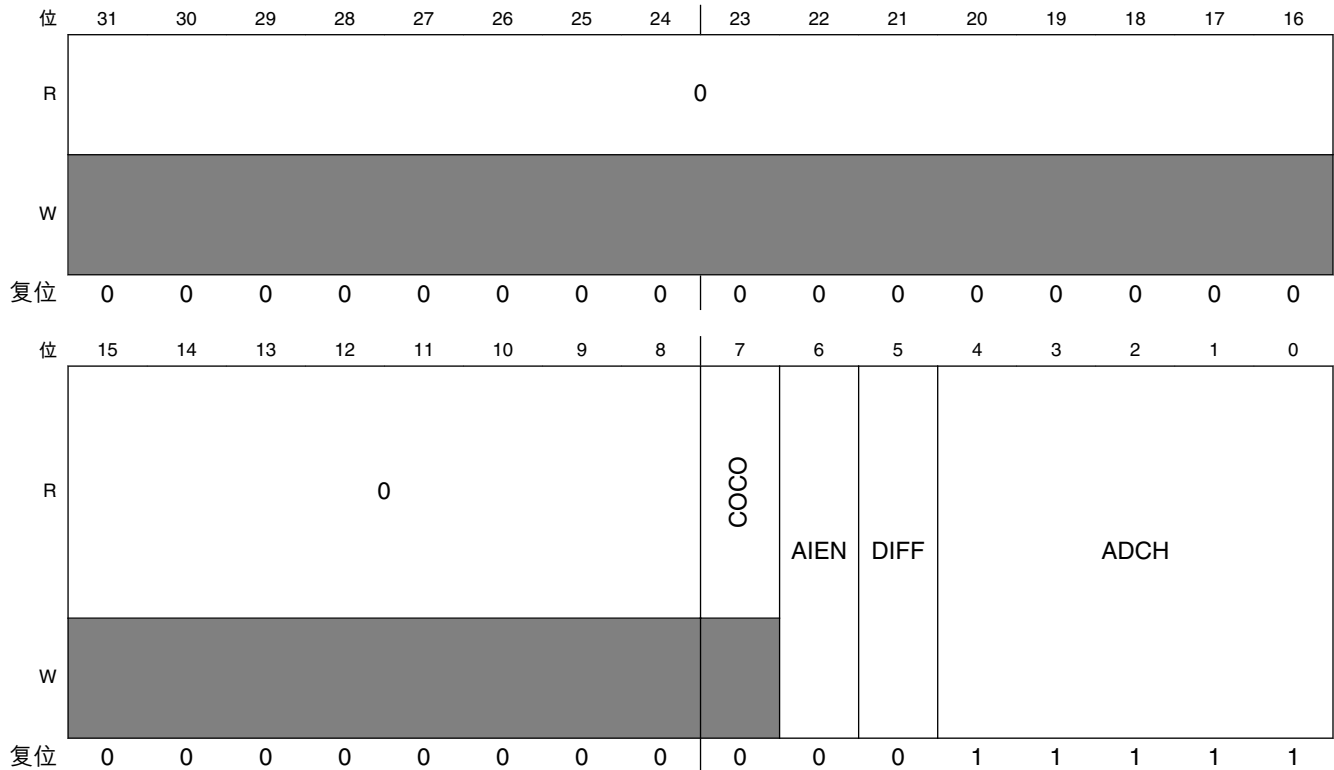
为了允许内部外设触发 ADC 的顺序转换功能，ADC 可以有多个状态和控制寄存器：每次转换一个 SC1B–SC1n 寄存器表示仅硬件触发模式中可用的多路 SC1 寄存器。有关该器件指定的 SC1n 寄存器数量，请参见芯片配置信息。SC1n 寄存器字段相同，用于控制 ADC 操作的“乒乓”方式中。

某个时间点上，仅一个 SC1n 寄存器有效控制 ADC 转换。对于该 MCU 特有的任意 SC1n 寄存器，允许在 SC1n 有效控制转换时更新 SC1A。

在 SC1A 有效控制转换时写入 SC1A 终止当前转换。在软件触发模式下，SC2[ADTRG]=0 时，若 SC1[ADCH]包含的值不为 1s (模块禁用)，则写入 SC1A 立即启动 ADC 转换。

在特定 SC1n 寄存器有效控制时写入 SC1n 终止当前转换。SC1B–SC1n 寄存器均不用于软件触发操作，因此写入 SC1B–SC1n 寄存器不会启动转换。

地址: 4003_B000h 基准 + 0h 偏移 + (4d × i), 其中 i=0d 到 1d



ADCx_SC1n 字段描述

字段	描述
31–8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 COCO	转换完成标志 此为只读字段，在比较功能禁用或 SC2[ACFE]=0 和硬件平均功能禁用，或 SC3[AVGE]=0 时，每完成一次转换，该字段都会置位。比较功能使能，或 SC2[ACFE]=1 时，只有在比较结果成立的情况下，COCO 才会在转换完成时置位。硬件平均功能使能，或 SC3[AVGE]=1 时，COCO 将在完成选定次数的转换之后置位（取决于 AVGS）。SC1A 中的 COCO 也会在校准序列完成时置位。当写入相应的 SC1n 寄存器或读取相应的 Rn 寄存器时，COCO 会清零。 0 转换已完成。 1 转换已完成。
6 AIEN	中断使能 使能转换完成中断。如果在 COCO 置位的同时相应 AIEN 为高电平，那么会产生中断。 0 转换完成中断已禁用。 1 转换完成中断使能。
5 DIFF	差分模式使能 配置 ADC，使其在差分模式下运行。使能时，此模式会自动从差分通道中选择，并更改转换算法和完成一次转换所需要的周期数。 0 选择单端转换和输入通道。 1 选择差分转换和输入通道。

下一页继续介绍此表...

ADCx_SC1n 字段描述 (继续)

字段	描述
ADCH	<p>输入通道选择</p> <p>选择一个输入通道。输入通道解码取决于 DIFF 值 DAD0-DAD3 与输入引脚对 DADPx 和 DADMx 关联。</p> <p>注：位字段-设置描述中的部分输入通道选项在用户的器件上不可用。有关器件的实际 ADC 通道分配，请参见芯片配置详情。</p> <p>当通道选择位全部置位，即 ADCHn = 11111 时，逐次渐进转换器子系统关闭。此特性允许显性禁用 ADC，并将输入通道与所有源相隔离。以这种方式终止连续转换将避免执行一次额外的单一转换。当未使能连续转换时，不必将通道选择位设为全 1 也可使 ADC 进入低功耗状态，因为模块会在转换完成时自动进入低功耗状态。</p> <p>00000 DIFF=0 时，选择 DADP0 作为输入；DIFF=1 时，选择 DAD0 作为输入。</p> <p>00001 DIFF=0 时，选择 DADP1 作为输入；DIFF=1 时，选择 DAD1 作为输入。</p> <p>00010 DIFF=0 时，选择 DADP2 作为输入；DIFF=1 时，选择 DAD2 作为输入。</p> <p>00011 DIFF=0 时，选择 DADP3 作为输入；DIFF=1 时，选择 DAD3 作为输入。</p> <p>00100 DIFF=0 时，选择 AD4 作为输入；DIFF=1 时，将其保留。</p> <p>00101 DIFF=0 时，选择 AD5 作为输入；DIFF=1 时，将其保留。</p> <p>00110 DIFF=0 时，选择 AD6 作为输入；DIFF=1 时，将其保留。</p> <p>00111 DIFF=0 时，选择 AD7 作为输入；DIFF=1 时，将其保留。</p> <p>01000 DIFF=0 时，选择 AD8 作为输入；DIFF=1 时，将其保留。</p> <p>01001 DIFF=0 时，选择 AD9 作为输入；DIFF=1 时，将其保留。</p> <p>01010 DIFF=0 时，选择 AD10 作为输入；DIFF=1 时，将其保留。</p> <p>01011 DIFF=0 时，选择 AD11 作为输入；DIFF=1 时，将其保留。</p> <p>01100 DIFF=0 时，选择 AD12 作为输入；DIFF=1 时，将其保留。</p> <p>01101 DIFF=0 时，选择 AD13 作为输入；DIFF=1 时，将其保留。</p> <p>01110 DIFF=0 时，选择 AD14 作为输入；DIFF=1 时，将其保留。</p> <p>01111 DIFF=0 时，选择 AD15 作为输入；DIFF=1 时，将其保留。</p> <p>10000 DIFF=0 时，选择 AD16 作为输入；DIFF=1 时，将其保留。</p> <p>10001 DIFF=0 时，选择 AD17 作为输入；DIFF=1 时，将其保留。</p> <p>10010 DIFF=0 时，选择 AD18 作为输入；DIFF=1 时，将其保留。</p> <p>10011 DIFF=0 时，选择 AD19 作为输入；DIFF=1 时，将其保留。</p> <p>10100 DIFF=0 时，选择 AD20 作为输入；DIFF=1 时，将其保留。</p> <p>10101 DIFF=0 时，选择 AD21 作为输入；DIFF=1 时，将其保留。</p> <p>10110 DIFF=0 时，选择 AD22 作为输入；DIFF=1 时，将其保留。</p> <p>10111 DIFF=0 时，选择 AD23 作为输入；DIFF=1 时，将其保留。</p> <p>11000 保留。</p> <p>11001 保留。</p> <p>11010 DIFF=0 时，选择温度传感器（单端）作为输入；DIFF=1 时，选择温度传感器（差分）作为输入。</p> <p>11011 DIFF=0 时，选择带隙（单端）作为输入；DIFF=1 时，选择带隙（差分）作为输入。</p> <p>11100 保留。</p> <p>11101 DIFF=0 时，选择 V_{REFSH} 作为输入；DIFF=1 时，选择 -V_{REFSH}（差分）作为输入。所选参考电压取决于 SC2[REFSEL]。</p> <p>11110 DIFF=0 时，选择 V_{REFSL} 作为输入；DIFF=1 时，将其保留。所选参考电压取决于 SC2[REFSEL]。</p> <p>11111 模块已禁用。</p>

33.4.2 ADC 配置寄存器 1 (ADCx_CFG1)

配置寄存器 1 (CFG1) 用于选择工作模式、时钟源、时钟分频，并配置低功耗或长采样时间。

地址: 4003_B000h 基准 + 8h 偏移 = 4003_B008h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADLPC	ADIV		ADLSMP	MODE		ADICLK	
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_CFG1 字段描述

字段	描述
31–8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 ADLPC	低功耗配置 控制逐次渐进转换器的功耗配置。在无需更高采样率时，这能优化功耗。 0 正常功耗配置。 1 低功耗配置。减小功耗的同时会降低时钟频率所允许的最大值。
6–5 ADIV	时钟分频选择 选择 ADC 用于生成内部时钟 ADCK 的分频比。 00 分频比为 1，时钟速率为输入时钟。 01 分频比为 2，时钟速率为 (输入时钟)/2。 10 分频比为 4，时钟速率为 (输入时钟)/4。 11 分频比为 8，时钟速率为 (输入时钟)/8。
4 ADLSMP	采样时间配置 根据选定的转换模式，在不同的采样时间之间进行选择。字段调整采样周期可以使高阻抗输入的采样更精确或使低阻抗输入的转换速度更快。当连续转换使能且无需高转换速率时，使用长采样时间还能降低整体功耗。当 ADLSMP=1 时，长采样时间选择位 (ADLSTS[1:0]) 可以选择长采样时间的程度。 0 短采样时间。 1 长采样时间。
3–2 MODE	转换模式选择

下一页继续介绍此表...

ADCx_CFG1 字段描述 (继续)

字段	描述
	选择 ADC 分辨率模式。 00 DIFF=0 时：为单端 8 位转换；DIFF=1 时，为差分 9 位转换，输出 2 的补码。 01 DIFF=0 时：为单端 12 位转换；DIFF=1 时，为差分 13 位转换，输出 2 的补码。 10 DIFF=0 时：为单端 10 位转换；DIFF=1 时，为差分 13 位转换，输出 2 的补码。 11 DIFF=0 时：为单端 10 位转换..；DIFF=1 时，为差分 16 位转换，输出 2 的补码
ADICLK	输入时钟选择 选择用于生成内部时钟 ADCK 的输入时钟源 ADCK。需要注意的是，当选择 ADACK 时钟源时，在转换开始前其不必处于有效状态。当选择了 ADACK 并且 ADACK 在转换开始前未处于有效状态，当 CFG2[ADACKEN]=0 时，异步时钟将在转换开始时激活，并在转换终止时失效。在这种情况下，每次重新激活时钟源时，都存在一个相关联的时钟启动延迟。 00 总线时钟 01 备选时钟 2 (ALTCLK2) 10 备用时钟(ALTCLK) 11 异步时钟(ADACK)

33.4.3 ADC 配置寄存器 2 (ADCx_CFG2)

配置寄存器 2 (CFG2) 选择针对超高速转换的特别高速配置，并在长采样模式下选择长采样时间的持续时间。

地址: 4003_B000h 基准 + Ch 偏移 = 4003_B00Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	[Shaded]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0			MUXSEL	ADACKEN	ADHSC	ADLSTS	
W	[Shaded]								[Shaded]			MUXSEL	ADACKEN	ADHSC	ADLSTS	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_CFG2 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7-5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

ADCx_CFG2 字段描述 (继续)

字段	描述
4 MUXSEL	<p>ADC 多路复用器选择</p> <p>更改 ADC 多路复用器置位以在 ADC 通道的备用置位中进行选择。</p> <p>0 已选择 ADxxa 通道。 1 已选择 ADxxb 通道。</p>
3 ADACKEN	<p>异步时钟输出使能</p> <p>使能异步时钟源和时钟源输出，不受 CFG1 转换和状态的影响[ADICLK]。基于 MCU 配置，异步时钟可由其他模块使用。请参见芯片配置信息。置位该字段，即可允许使用该时钟，即使在 ADC 空闲或者以不同时钟源工作时，也可如此。另外，以所选异步时钟启动单次或首次连续转换时的延迟会减少，因为 ADACK 时钟已经开始工作。</p> <p>0 异步时钟输出已禁用；异步时钟只会由 ADICLK 选择且有一个转换有效时才能使能。 1 使能异步时钟和时钟输出，不受 ADC 状态的影响。</p>
2 ADHSC	<p>高速配置</p> <p>将 ADC 配置为超高速工作模式。更改了转换序列(转换时间增加了 2 个 ADCK 周期)，以允许速度更快的转换时钟。</p> <p>0 选择了正常转换序列。 1 选择了高速转换序列(总转换时间另外增加了 2 个 ADCK 周期)。</p>
ADLSTS	<p>长采样时间选择</p> <p>选择长采样时间 (CFG1[ADLSMP]=1) 时，在扩展采样时间之间进行选择。这样可以使高阻抗输入的采样更精确或使低阻抗输入的转换速度更快。当连续转换使能且无需高转换速率时，使用长采样时间还能降低整体功耗。</p> <p>00 默认的最长采样时间 (20 个额外的 ADCK 周期；总共 24 个 ADCK 周期)。 01 12 个额外的 ADCK 周期；采样总时间为 16 个 ADCK 周期。 10 6 个额外的 ADCK 周期；采样总时间为 10 个 ADCK 周期。 11 2 个额外的 ADCK 周期；采样总时间为 6 个 ADCK 周期。</p>

33.4.4 ADC 数据结果寄存器 (ADCx_Rn)

数据结果寄存器 (Rn) 含有相应状态和通道控制寄存器 (SC1A:SC1n) 所选通道 ADC 转换的结果。每个状态和通道控制寄存器都有一个对应的数据结果寄存器。

在无符号右对齐模式下，Rn 寄存器中未使用的位将清零，并且在符号扩展 2 二进制补码模式下，携带符号位(MSB)。例如，当配置为 10 位单端模式时，D[15:10]将清零。当配置为 11 位差分模式时，D[15:10]将携带符号位 (第 10 位通过第 15 位扩展)。

下表描述了数据结果寄存器在不同工作模式下的行为。

表 33-50. 数据结果寄存器描述

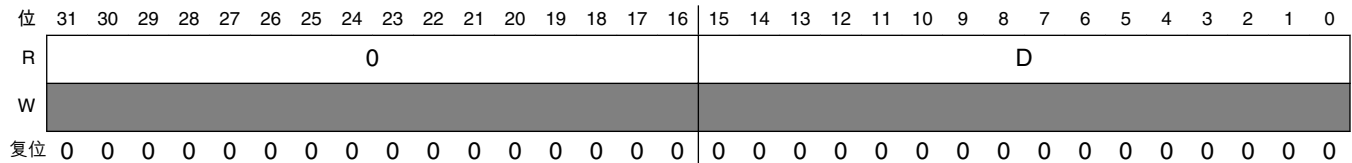
转换模式	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	格式
16 位差分	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	带符号的 2 二进制补码
16 位单端	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	无符号右对齐
13 位差分	S	S	S	S	D	D	D	D	D	D	D	D	D	D	D	D	符号扩展二进制补码
12 位单端	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	无符号右对齐
11 位差分	S	S	S	S	S	S	D	D	D	D	D	D	D	D	D	D	符号扩展二进制补码
10 位单端	0	0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	无符号右对齐
9 位差分	S	S	S	S	S	S	S	S	D	D	D	D	D	D	D	D	符号扩展二进制补码
8 位单端	0	0	0	0	0	0	0	0	D	D	D	D	D	D	D	D	无符号右对齐

注

S: 符号位或符号位扩展;

D: 数据 (指出时为二进制补码数据)

地址: 4003_B000h 基准 + 10h 偏移 + (4d × i), 其中 i=0d 到 1d



ADCx_Rn 字段描述

字段	描述
31-16 保留	此字段为保留字段。此只读字段为保留字段且值始终为 0。
D	数据结果

33.4.5 比较值寄存器 (ADCx_CVn)

比较值寄存器 (CV1 和 CV2) 含有一个比较值, 用于在比较功能使能 (SC2[ACFE]=1) 时与转换结果进行比较。此寄存器在位位置定义和值格式 (无符号或符号扩展二进制补码) 两个方面均采用与 Rn 寄存器在不同工作模式下相同的格式。因此, 比较功能只会使用与 ADC 工作模式相关的 CVn 字段。

只有在比较范围功能使能 (SC2[ACREN]=1) 时才使用比较值 2 寄存器 (CV2)。

地址: 4003_B000h 基准 + 18h 偏移 + (4d × i), 其中 i=0d 到 1d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CV															
W	0																0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_CVn 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CV	比较值。

33.4.6 状态和控制寄存器 2 (ADCx_SC2)

状态和控制寄存器 2 (SC2) 含有 ADC 模块的有效转换、硬件/软件触发器选择、比较功能和参考电压选择。

地址: 4003_B000h 基准 + 20h 偏移 = 4003_B020h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADACT	ADTRG	ACFE	ACFGT	ACREN	DMAEN	REFSEL	
W	0								0	0	0	0	0	0	0	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_SC2 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 ADACT	转换有效 指示转换或硬件平均正在进行。ADACT 在转换启动时置位，在转换完成或中止时清零。 0 转换未在进行。 1 转换正在进行。
6 ADTRG	转换触发选择 选择用于启动转换的触发类型。有两种类型触发可选： <ul style="list-style-type: none"> • 软件触发：选择软件触发时，对 SC1A 进行写操作后启动转换。 • 硬件触发：选择硬件触发时，ADHWT 输入在 ADHWTSn 输入脉冲之后变为有效值，然后启动转换。 0 选择软件触发。 1 选择硬件触发。
5 ACFE	比较功能使能 使能比较功能。 0 比较功能禁用。 1 比较功能使能。
4 ACFGT	比较功能大于使能 配置比较功能，以根据 ACREN 的值，检查转换结果相对于 CV1 和 CV2 的情况。ACFE 位必须置位，ACFGT 才能发挥作用。 0 根据 CV1 和 CV2 中的值，配置小于阈值、超范围（不含）和范围内（不含）功能。 1 根据 CV1 和 CV2 中的值，配置大于或等于阈值、超范围（含）和范围内（含）功能。
3 ACREN	比较功能范围使能 配置比较功能，以检查输入的转换结果是在 CV1 和 CV2 形成的范围之间还是之外，具体取决于 ACFGT 的值。ACFE 位必须置位，ACFGT 才能发挥作用。 0 范围功能禁用。仅比较 CV1。 1 范围功能使能。CV1 和 CV2 均进行比较。
2 DMAEN	DMA 使能 0 DMA 已禁用。 1 DMA 使能，并会在任意 SC1n[COCO]标志变为有效时注意到的 ADC 转换完成事件过程中，使 ADC DMA 请求变为有效。
REFSEL	基准电压源选择 选择用于转换的基准电压源。 00 默认参考电压引脚对（外部引脚 V_{REFH} 和 V_{REFL} ） 01 备选参考对（ V_{ALTH} 和 V_{ALTLL} ），此对可以选择额外的外部引脚或内部源，具体取决于 MCU 配置。有关此 MCU 的具体详情，请参见芯片配置信息。 10 保留 11 保留

33.4.7 状态和控制寄存器 3 (ADCx_SC3)

状态和控制寄存器 3 (SC3) 控制 ADC 模块的校准、连续转换和硬件平均功能。

地址: 4003_B000h 基准 + 24h 偏移 = 4003_B024h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	[Shaded]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CAL	CALF	0		ADCO	AVGE	AVGS	
W	[Shaded]								w1c	[Shaded]						
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_SC3 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 CAL	校准 置位时开始校准序列。此字段在校准进行时保持置位，在校准序列完成时清零。必须检查 CALF，以确定校准序列的结果。一旦开始，就不能通过写入 ADC 寄存器来中断校准例程，否则，结果将无效且 CALF 会置位。置位 CAL 会中止当前的任何转换。
6 CALF	校准失败标志 显示校准序列的结果。如果 SC2[ADTRG] = 1，任何 ADC 寄存器被写入，或者在校准序列完成之前进入任何停止模式，校准序列将失败。将 1 写入 CALF 可将其清零。 0 校准正常完成。 1 校准失败。不保证 ADC 精度规格。
5-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 ADCO	连续转换使能

下一页继续介绍此表...

ADCx_SC3 字段描述 (继续)

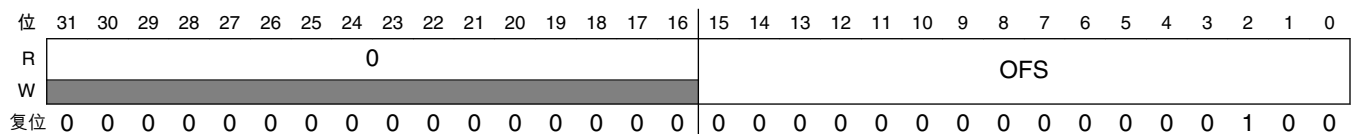
字段	描述
	使能连续转换。 0 如果在启动转换之后使能硬件平均值功能 (AVGE=1), 则为一次转换或一组转换。 1 如果在启动转换之后使能硬件平均值功能 (AVGE=1), 则为连续转换或多组转换。
2 AVGE	硬件平均值使能 使能 ADC 的硬件平均值功能。 0 硬件平均值功能已禁用。 1 硬件平均值功能使能。
AVGS	硬件平均值选择 决定要计算多少次 ADC 转换平均值以形成 ADC 平均值结果。 00 计算了 4 个样本的平均值。 01 计算了 8 个样本的平均值。 10 计算了 16 个样本的平均值。 11 计算了 32 个样本的平均值。

33.4.8 ADC 偏移校正寄存器 (ADCx_OFS)

ADC 偏移校正寄存器 (OFS) 含有用户选择的或校准生成的偏移误差校正。此寄存器为二进制补码、左对齐的 16 位值，级联而成。从转换中减去 OFS 中的值，把结果传输至结果寄存器 Rn。如果结果大于最大结果值或小于最小结果值，则会强制应用当前操作模式的相应限值。

有关校准程序的更多信息，请参见[校准功能](#) 章节。

地址: 4003_B000h 基准 + 28h 偏移 = 4003_B028h



ADCx_OFS 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
OFS	偏移误差校正

33.4.9 ADC 同相端增益寄存器 (ADCx_PG)

同相端增益寄存器 (PG) 含有差分模式下的同相端输入增益误差校正因子，或单端模式下的整个转换。16 位实数二进制格式的 PG 为增益调整因子，其基数点固定在 PG[15]和 PG[14]之间。用户必须在此寄存器中写入校准程序中描述的值。否则，可能达不到增益误差规格的要求。

有关校准程序的更多信息，请参见[校准功能](#) 章节。

地址: 4003_B000h 基准 + 2Ch 偏移 = 4003_B02Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
R	0																PG																	
W	0																0																	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

ADCx_PG 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
PG	同相端增益

33.4.10 ADC 反相端增益寄存器 (ADCx_MG)

反相端增益低电平寄存器 (MG) 含有差分模式下的反相端输入增益误差校正因子。在单端模式下，此寄存器将被忽略。16 位实数二进制格式的 MG 为增益调整因子，其基数点固定在 MG[15]和 MG[14]之间。用户必须在此寄存器中写入校准程序中描述的值。否则，可能达不到增益误差规格的要求。

有关校准程序的更多信息，请参见[校准功能](#) 章节。

地址: 4003_B000h 基准 + 30h 偏移 = 4003_B030h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
R	0																MG																	
W	0																0																	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

ADCx_MG 字段描述

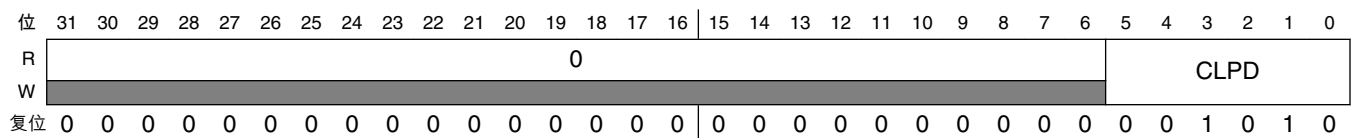
字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
MG	反相端增益

33.4.11 ADC 同相端通用校准值寄存器 (ADCx_CLPD)

同相端通用校准值寄存器 (CLPx) 含有校准功能生成的校准信息。这些寄存器含有宽度不同的七个校准值：CLP0[5:0]、CLP1[6:0]、CLP2[7:0]、CLP3[8:0]、CLP4[9:0]、CLPS[5:0]和 CLPD[5:0]。CLPx 寄存器将在自校准序列完成 (CAL 清零) 时自动置位。如果用户在校准之后对这些寄存器执行写操作，则可能无法达到线性度误差规格的要求。

有关校准规程的更多信息，请参见[校准功能](#) 章节。

地址: 4003_B000h 基准 + 34h 偏移 = 4003_B034h



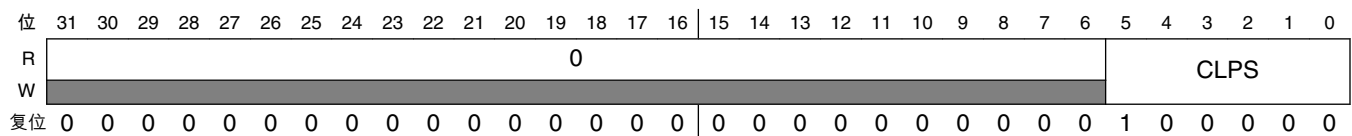
ADCx_CLPD 字段描述

字段	描述
31-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLPD	校准值 校准值

33.4.12 ADC 同相端通用校准值寄存器 (ADCx_CLPS)

有关更多信息，请参见 CLPD 寄存器说明。

地址: 4003_B000h 基准 + 38h 偏移 = 4003_B038h



ADCx_CLPS 字段描述

字段	描述
31-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLPS	校准值 校准值

33.4.13 ADC 同相端通用校准值寄存器 (ADCx_CLP4)

有关更多信息，请参见 CLPD 寄存器说明。

地址: 4003_B000h 基准 + 3Ch 偏移 = 4003_B03Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0																CLP4																
W																																	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_CLP4 字段描述

字段	描述
31-10 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLP4	校准值 校准值

33.4.14 ADC 同相端通用校准值寄存器 (ADCx_CLP3)

有关更多信息，请参见 CLPD 寄存器说明。

地址: 4003_B000h 基准 + 40h 偏移 = 4003_B040h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0																CLP3																
W																																	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

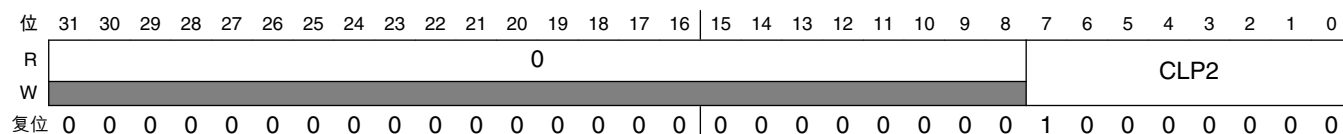
ADCx_CLP3 字段描述

字段	描述
31-9 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLP3	校准值 校准值

33.4.15 ADC 同相端通用校准值寄存器 (ADCx_CLP2)

有关更多信息，请参见 CLPD 寄存器说明。

地址: 4003_B000h 基准 + 44h 偏移 = 4003_B044h



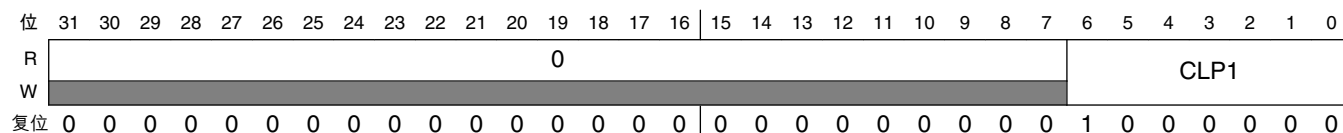
ADCx_CLP2 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLP2	校准值 校准值

33.4.16 ADC 同相端通用校准值寄存器 (ADCx_CLP1)

有关更多信息，请参见 CLPD 寄存器说明。

地址: 4003_B000h 基准 + 48h 偏移 = 4003_B048h



ADCx_CLP1 字段描述

字段	描述
31-7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLP1	校准值 校准值

33.4.17 ADC 同相端通用校准值寄存器 (ADCx_CLP0)

有关更多信息，请参见 CLPD 寄存器说明。

地址: 4003_B000h 基准 + 4Ch 偏移 = 4003_B04Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0																CLP0																
W																																	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLP0 字段描述

字段	描述
31-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLP0	校准值 校准值

33.4.18 ADC 反相端通用校准值寄存器 (ADCx_CLMD)

反相端通用校准值 (CLM_x) 寄存器含有校准功能生成的校准信息。这些寄存器含有宽度不同的七个校准值: CLM0[5:0]、CLM1[6:0]、CLM2[7:0]、CLM3[8:0]、CLM4[9:0]、CLMS[5:0]和 CLMD[5:0]。CLM_x 寄存器将在自校准序列完成 (CAL 清零) 时自动置位。如果用户在校准之后对这些寄存器执行写操作，则可能无法达到线性度误差规格的要求。

有关校准程序的更多信息，请参见[校准功能](#) 章节。

地址: 4003_B000h 基准 + 54h 偏移 = 4003_B054h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0																CLMD																
W																																	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

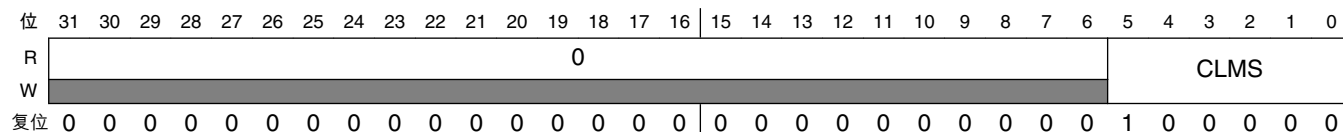
ADCx_CLMD 字段描述

字段	描述
31-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLMD	校准值 校准值

33.4.19 ADC 反相端通用校准值寄存器 (ADCx_CLMS)

有关更多信息，请参见 CLMD 寄存器说明。

地址: 4003_B000h 基准 + 58h 偏移 = 4003_B058h



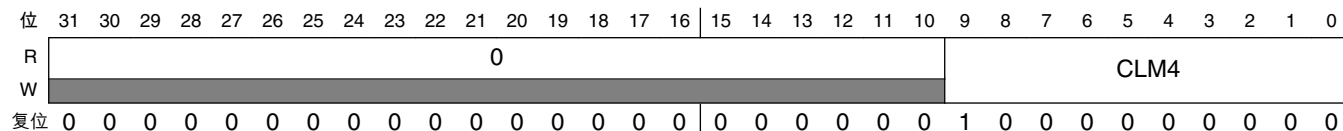
ADCx_CLMS 字段描述

字段	描述
31-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLMS	校准值 校准值

33.4.20 ADC 反相端通用校准值寄存器 (ADCx_CLM4)

有关更多信息，请参见 CLMD 寄存器说明。

地址: 4003_B000h 基准 + 5Ch 偏移 = 4003_B05Ch



ADCx_CLM4 字段描述

字段	描述
31-10 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLM4	校准值 校准值

33.4.21 ADC 反相端通用校准值寄存器 (ADCx_CLM3)

有关更多信息，请参见 CLMD 寄存器说明。

地址: 4003_B000h 基准 + 60h 偏移 = 4003_B060h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0																CLM3																
W																																	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

ADCx_CLM3 字段描述

字段	描述
31-9 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLM3	校准值 校准值

33.4.22 ADC 反相端通用校准值寄存器 (ADCx_CLM2)

有关更多信息，请参见 CLMD 寄存器说明。

地址: 4003_B000h 基准 + 64h 偏移 = 4003_B064h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0																CLM2																
W																																	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

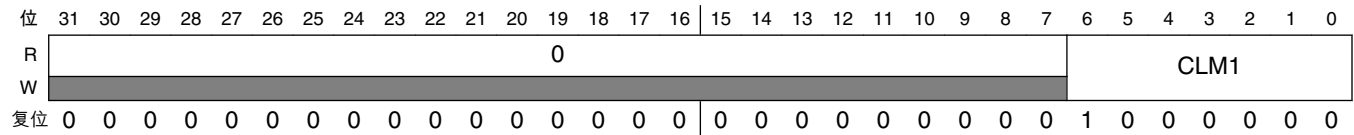
ADCx_CLM2 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLM2	校准值 校准值

33.4.23 ADC 反相端通用校准值寄存器 (ADCx_CLM1)

有关更多信息，请参见 CLMD 寄存器说明。

地址: 4003_B000h 基准 + 68h 偏移 = 4003_B068h



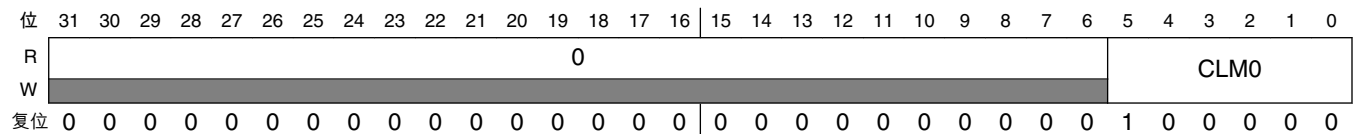
ADCx_CLM1 字段描述

字段	描述
31-7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLM1	校准值 校准值

33.4.24 ADC 反相端通用校准值寄存器 (ADCx_CLM0)

有关更多信息，请参见 CLMD 寄存器说明。

地址: 4003_B000h 基准 + 6Ch 偏移 = 4003_B06Ch



ADCx_CLM0 字段描述

字段	描述
31-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLM0	校准值 校准值

33.5 功能说明

ADC 模块在复位过程中、在低功耗停止模式下或者在 SC1n[ADCH]全部为高电平时禁用；详见电源管理信息。该模块在转换完成后以及另一个转换还未发起前处于空闲状态。当它为空闲且异步时钟输出使能功能被禁用（或 CFG2[ADACKEN]=0）时，模块处于最低功耗状态。ADC 可以在任意软件可选通道上执行模数转换。所有模式都会通过一种逐次逼近算法执行转换。

为了达到精度规格要求，必须用片上校准功能校准 ADC 模块。

请参阅[校准功能](#)，了解如何执行校准的详情。

转换完成后，结果置于 Rn 数据寄存器中。相应的 SC1n[COCO]随后置位并且在相应转换完成中断被使能（SC1n[AIEN]=1）的情况下会生成一个中断。

ADC 模块能够自动将转换结果与 CV1 和 CV2 寄存器的内容作比较。比较功能在 SC2[ACFE]置位的情况下使能，而且能在任何转换模式和配置下工作。

ADC 模块能自动求多次转换结果的平均值。硬件平均值功能在 SC3[AVGE]置位的情况下使能，而且能在任何转换模式和配置下工作。

注

有关特定芯片的工作模式，请参见此 MCU 的电源管理信息。

33.5.1 时钟选择和分频控制

ADC 模块的时钟源可以从四个时钟源中选择。

然后用一个可配置的值将该时钟源分频，生成模块的输入时钟 ADCK。利用 CFG1[ADICLK]从下列时钟源选择时钟：

- 总线时钟。这些复位后的默认选择。
- ALTCLK2 定义见此 MCU。请参见芯片配置信息。在 MCU 处于正常停止模式时，可以将 ALTCLK2 作为输入时钟源进行转换。
- ALTCLK：定义见此 MCU。请参见芯片配置信息。在 MCU 处于正常停止模式时，可以将 ALTCLK 作为输入时钟源进行转换。
- 异步时钟(ADACK)：该时钟从 ADC 模块内部的时钟源生成。当选择 ADACK 时钟源时，在转换开始前其不必处于有效状态。当选择了 ADACK 并且 ADACK 在转换开始前未处于有效状态 CFG2[ADACKEN]=0 时，ADACK 将在

转换开始时激活，并在转换终止时失效。在这种情况下，每次重新激活时钟源时，都存在一个相关联的时钟启动延迟。为了避免出现与 ADACK 时钟启动相关的转换时间变化和延迟，使 $CFG2[ADACKEN]=1$ ，并在用 ADACK 时钟源启动任何转换之前，等待最长条件启动时间，即 $5\ \mu s$ 。在 MCU 处于正常停止模式时，可以将 ADACK 作为输入时钟源进行转换。有关更多信息，请参阅[电源控制](#)。

无论选择哪个时钟，其频率必须在 ADCK 的指定频率范围内。如果可用时钟太慢，ADC 可能无法以额定性能工作。如果可用时钟太快，必须将其分频到合适的频率。该分频器由 $CFG1[ADIV]$ 指定，可执行 1、2、4 或 8 分频。

33.5.2 参考电压选择

可将 ADC 配置为接受两个参考电压对之一，将其作为用于转换的参考电压 (V_{REFSH} 和 V_{REFSL})。

每对含有一个正参考电压，该电压必须在最小高电平参考电压和 V_{DDA} 之间，以及一个接地参考电压，其电位必须与 V_{SSA} 相同。两对分别为外部 (V_{REFH} 和 V_{REFL}) 以及备用 (V_{ALTH} 和 V_{ALTTL}) 电压对。这些参考电压是通过 $SC2[REFSEL]$ 选择的。备用 (V_{ALTH} 和 V_{ALTTL}) 参考电压对可以选择额外的外部引脚或内部源，具体取决于 MCU 配置。有关此 MCU 特定的参考电压，请参见芯片配置信息。

33.5.3 硬件触发和通道选择

ADC 模块具有一个可选的异步硬件转换触发 ADHWT，该触发器将在 $SC2[ADTRG]$ 置位且发生硬件触发选择事件 ADHWTSn 时使能。

此触发源不适用于所有 MCU。有关此 MCU 专用的 ADHWT 源和 ADHWTSn 配置的信息，请参见芯片专用 ADC 信息。

如果 ADHWT 存在有效的触发源且使能硬件触发 $SC2[ADTRG]=1$ ，则在发生硬件触发选择事件 ADHWTSn 后会在 ADHWT 的上升沿发起转换。如果出现触发上升沿时转换正在进行中，则该上升沿会被忽略。在持续的转换配置中，仅观测到用初始化上升沿来发布持续转换，直至转换中止，ADC 在启动转换的 SCn 寄存器上继续转换。硬件触发器功能可在任何转换模式及配置下运行。

在接收 ADHWT 信号之前，硬件触发选择事件 ADHWTSn 必须置位。若无法满足这些条件，则转换器会忽略触发或使用不正确的配置。若硬件触发选择事件在转换过程中置位，则必须保持置位至当前转换结束并且触发信号必须能保持生效一直到被 ADHWT 接受启动新的 ADC 转换。根据有效触发选择信号来选择用于转换的通道和状态字段：

- ADHWTS_A 有效选择 SC_{1A}。
- ADHWTS_n 有效选择 SC_{1n}。

注

同时置位一个以上硬件触发选择信号(ADHWTS_n)会导致未知的结果。为避免这种情况,在下一个目标转换前仅选择一个硬件触发选择信号(ADHWTS_n)。

转换完成后,结果置于与接收到的 ADHWTS_n 关联的 R_n 寄存器中。例如:

- ADHWTS_A 有效选择 RA 寄存器
- ADHWTS_n 有效选择 R_n 寄存器

转换完成标志 SC_{1n}[COCO]与接收到的 ADHWTS_n 关联,随后置位并在相应转换完成中断已使能的情况下(SC₁[AIEN]=1)生成一个中断。

33.5.4 转换控制

CFG₁[MODE]和 SC_{1n}[DIFF]控制 ADC 转换模式。

转换可由软件或硬件触发发起。

此外,可配置 ADC 模块以用于:

- 低功耗运行
- 长采样时间
- 持续转换
- 硬件平均值
- 自动比较转换结果与软件预设的比较值

33.5.4.1 启动转换

启动转换:

- 在写入 SC_{1A} 后,且 SC_{1n}[ADCH]不全为 1 的情况下,软件触发模式下 (SC₂[ADTRG]=0)。
- 硬件触发或 ADHWT 事件后,在硬件触发模式下 (SC₂[ADTRG]=1) 且发生硬件触发选择事件 ADHWTS_n。根据有效触发选择信号来选择通道和状态字段:
 - ADHWTS_A 有效选择 SC_{1A}。
 - ADHWTS_n 有效选择 SC_{1n}。
 - 若两种皆无效,则选择关闭

注

在转换完成前选择一个以上的 ADHWTSn 会导致未知的结果。为避免这种情况，在转换完成前仅选择一个 ADHWTSn。

- 在连续转换使能的情况下(即 SC3[ADCO]=1)，当上次转换结束，结果存入数据寄存器后。

如果连续转换使能，则在当前转换完成后会自动发起新转换。在软件触发操作中 (SC2[ADTRG] = 0)，持续转换在写入 SC1A 后开始，并一直继续直到中止。在硬件触发操作中 (SC2[ADTRG] = 1)，且发生了一件 ADHWTSn 事件，持续转换在硬件触发事件后开始，并一直继续直到中止。

若硬件平均使能，则在当前转换完成后会自动发起新转换，直到完成了正确的转换数。在软件触发操作中，转换在写入 SC1A 后开始。在硬件触发操作中，转换在硬件触发后开始。若持续转换使能，在上一次转换和选定转换数后，开始求新转换集的平均值。

33.5.4.2 完成转换

当转换结果存储到转换结果寄存器 Rn 时即完成转换。在禁用比较功能的情况下，SC1n[COCO]置位，表明转换结束。若硬件平均使能，相应的 SC1n[COCO]仅在转换完成最后一个选定数时置位。若比较功能使能，仅在比较条件为真的情况下，相应的 SC1n[COCO]置位，并且存储转换结果到数据寄存器。若硬件平均和比较功能均使能，则相应的 SC1n[COCO]仅在转换完成最后一个选定数且比较条件为真的情况下置位。如果相应 SC1n[AIEN]在 SC1n[COCO]置位时为高电平，则会生成中断。

33.5.4.3 中止转换

正在进行的任何转换在下列情况下中止：

- 在 SC1A 启动转换生效时，写入 SC1A，中止现有转换。在软件触发模式下，SC2[ADTRG]=0 时，若 SC1A[ADCH]的值不全是 1，则写入 SC1A 启动新转换。在特定 SC1B-SC1n 寄存器启动转换生效时，写入任意 SC1B-SC1n 寄存器，即可中断现有转换。SC1(B-n)寄存器无法用于软件触发模式，因此写入 SC1(B-n)寄存器并不会启动新转换。
- 将写入除 SC1A-SC1n 寄存器以外的任意 ADC 寄存器。这表明工作模式已发生改变，因此当前转换无效。

- MCU 复位或进入低功耗停止模式。
- MCU 进入正常停止模式，在 ADACK 或备用时钟源未使能的情况下，MCU 进入正常停止模式。

转换中止后，数据寄存器 Rn 的内容保持不变。数据寄存器仍然是完成最后一次成功转换之后传输的值。如果转换因复位或低功耗停止模式而中止，RA 和 Rn 会恢复到复位状态。

33.5.4.4 功耗控制

ADC 模块在启动转换前一直保持在其空闲状态。如果将 ADACK 选为转换时钟源，但异步时钟输出被禁用，即 CFG2[ADACKEN]=0，ADACK 时钟发生器也会保持闲置状态（禁用），直到启动转换为止。如果异步时钟输出使能，即 CFG2[ADACKEN]=1，无论 ADC 或 MCU 功率模式处于何种状态，也会保持有效。ADC 为活动状态时，其功耗可通过设置 CFG1[ADLPC]来降低。这会降低 f_{ADCK} 。

33.5.4.5 采样时间和总转换时间

对于短采样，即当 CFG1[ADLSMP]=0 时，首次转换时会在四个 ADCK 周期的基本采样时间的基础上增加 2 个周期。对于高速转换，即当 CFG2[ADHSC]=1 时，任何转换都会额外增加 2 个周期。下表总结了可能的 ADC 配置的采样时间。

ADC 配置			采样时间 (ADCK 周期)	
CFG1[ADLSMP]	CFG2[ADLSTS]	CFG2[ADHSC]	首次或单次	后续
0	X	0	6	4
1	00	0	24	
1	01	0	16	
1	10	0	10	
1	11	0	6	
0	X	1	8	6
1	00	1	26	
1	01	1	18	
1	10	1	12	
1	11	1	8	

总转换时间取决于：

- 由 CFG1[ADLSMP]和 CFG2[ADLSTS]决定的采样时间
- MCU 总线频率

- 由 CFG1[MODE] 和 SC1n[DIFF] 决定的转换模式
- 高速配置，即 CFG2[ADHSC]
- 转换时钟的频率，即 f_{ADCK} 。

CFG2[ADHSC]用于配置较高的时钟输入频率。这样可以缩短总转换时间。为了满足内部 ADC 时序要求，CFG2[ADHSC]会添加额外的 ADCK 周期。以 CFG2[ADHSC]=1 进行转换会多花两个 ADCK 周期。当 ADCLK 超过 CFG2[ADHSC]=0 的限值时，必须使用 CFG2[ADHSC]。

该模块激活后，便开始对输入进行采样。

1. CFG1[ADLSMP]和 CFG2[ADLSTS]基于选择的转换模式在采样时间之间进行选择。
2. 采样完成时，转换器与输入通道隔离开来，用逐次逼近算法以确定模拟信号的数字值。
3. 转换算法完成时，转换结果转移到 Rn。

如果总线频率低于 f_{ADCK} 频率，使能短采样时间时(CFG13[ADLSMP] = 0)，就无法保证连续转换的精确采样时间。

最长总转换时间由选定的时钟源和分频比决定。时钟源可通过 CFG1[ADICLK]选择，分频比通过 CFG1[ADIV]指定。

下式总结了所有配置下的最大总转换时间。等式中使用的各种变量，请参见下面的表格。

$$\text{ConversionTime} = \text{SFCAdder} + \text{AverageNum} \times (\text{BCT} + \text{LSTAdder} + \text{HSCAdder})$$

等式 1. 转换时间等式

表 33-75. 单次或首次连续时间加法器(SFCAdder)

CFG1[ADLSMP]	CFG2[ADACKEN]	CFG1[ADICLK]	单次或首次连续时间加法器(SFCAdder)
1	x	0x, 10	3 个 ADCK 周期 + 5 个总线时钟周期
1	1	11	3 个 ADCK 周期 + 5 个总线时钟周期 ¹
1	0	11	5 μ s + 3 个 ADCK 周期 + 5 个总线时间周期
0	x	0x, 10	5 个 ADCK 周期 + 5 个总线时钟周期
0	1	11	5 个 ADCK 周期 + 5 个总线时钟周期 ¹
0	0	11	5 μ s + 5 个 ADCK 周期 + 5 个总线时钟周期

1. 为了实现此时间目标，在启动转换之前，CFG2[ADACKEN]必须为 1 且至少保持 5 μ s。

表 33-76. 平均数据因子(AverageNum)

SC3[AVGE]	SC3[AVGS]	平均数据因子(AverageNum)
0	xx	1

下一页继续介绍此表...

表 33-76. 平均数据因子(AverageNum) (继续)

SC3[AVGE]	SC3[AVGS]	平均数据因子(AverageNum)
1	00	4
1	01	8
1	10	16
1	11	32

表 33-77. 基本转换时间(BCT)

模式	基本转换时间(BCT)
8 位单端	17 个 ADCK 周期
9 位差分	27 个 ADCK 周期
10 位单端	20 个 ADCK 周期
11 位差分	30 个 ADCK 周期
12 位单端	20 个 ADCK 周期
13 位差分	30 个 ADCK 周期
16 位单端	25 个 ADCK 周期
16 位差分	34 个 ADCK 周期

表 33-78. 长采样时间加法器(LSTAdder)

CFG1[ADLSMP]	CFG2[ADLSTS]	长采样时间加法器(LSTAdder)
0	xx	0 个 ADCK 周期
1	00	20 个 ADCK 周期
1	01	12 个 ADCK 周期
1	10	6 个 ADCK 周期
1	11	2 个 ADCK 周期

表 33-79. 高速转换时间加法器(HSCAdder)

CFG2[ADHSC]	高速转换时间加法器(HSCAdder)
0	0 个 ADCK 周期
1	2 个 ADCK 周期

注

ADCK 频率必须在 f_{ADCK} 最小值和 f_{ADCK} 最大值之间, 以满足 ADC 规格要求。

33.5.4.6 转换时间示例

以下示例使用等式 1 第 页上 648，以及表 33-75 至表 33-79 中提供的信息。

33.5.4.6.1 典型转换时间配置

ADC 转换的一种典型配置：

- 10 位模式，其中，总线时钟被选为输入时钟源
- 选择的比率是输入时钟除以 1
- 总线频率为 8 MHz
- 长采样时间被禁用
- 高速转换被禁用

单次转换的转换时间通过等式 1 第 页上 648 以及表 33-75 和表 33-79 中提供的信息计算。下表列出了等式 1 第 页上 648 的变量。

表 33-80. 典型转换时间

变量	时间
SFCAdder	5 个 ADCK 周期 + 5 个总线时钟周期
AverageNum	1
BCT	20 个 ADCK 周期
LSTAdder	0
HSCAdder	0

最终转换时间使用上表中的参数产生。因此，如果总线时钟和 ADCK 频率等于 8 MHz，则最终转换时间为 3.75 μ s。

33.5.4.6.2 长转换时间配置

长 ADC 转换的一种配置：

- 16 位差分模式，其中，总线时钟被选为输入时钟源
- 选择的比率是输入时钟除以 8
- 总线频率为 8 MHz
- 长采样时间使能
- 配置为最长加法器
- 高速转换被禁用
- 平均使能 32 次转换

此次转换的转换时间通过等式 1 第 页上 648 以及表 33-75 和表 33-79 中提供的信息计算。下表列出了等式 1 第 页上 648 的变量。

表 33-81. 典型转换时间

变量	时间
SFCAdder	3 个 ADCK 周期 + 5 个总线时钟周期
AverageNum	32
BCT	34 个 ADCK 周期
LSTAdder	20 个 ADCK 周期
HSCAdder	0

最终转换时间使用上表中的参数产生。因此，如果总线时钟等于 8 MHz 且 ADCK 等于 1 MHz，最终转换时间为 57.625 μ s，即 AverageNum。结果，总转换时间为 1.844 ms。

33.5.4.6.3 短转换时间配置

短 ADC 转换的一种配置：

- 8 位单端模式，其中，总线时钟被选为输入时钟源
- 选择的比率是输入时钟除以 1
- 总线频率为 20 MHz
- 长采样时间被禁用
- 高速转换使能

此次转换的转换时间通过等式 1 第 页上 648 以及表 33-75 和表 33-79 中提供的信息计算。下表列出了等式 1 第 页上 648 的变量。

表 33-82. 典型转换时间

变量	时间
SFCAdder	5 个 ADCK 周期 + 5 个总线时钟周期
AverageNum	1
BCT	17 个 ADCK 周期
LSTAdder	0 个 ADCK 周期
HSCAdder	2

最终转换时间使用上表中的参数产生。因此，如果总线时钟和 ADCK 频率等于 20 MHz，则最终转换时间为 1.45 μ s。

33.5.4.7 硬件平均功能

设置 SC3[AVGE]=1，可使能硬件平均功能，以计算多次转换的硬件均值。转换次数取决于 AVGS[1:0]位，可能选择 4、8、16 或 32 次转换来求均值。当硬件平均功能在执行时，SC2[ADACT]会置位。

在所选输入被采样和转换时，结果将置于累加器中，当所选数量的转换完成时，则会从累加器计算平均值。当硬件平均功能被选中时，单次转换完成不会使 SC1n[COCO]置位。

如果比较功能被禁用，或者其值为真，则在所选转换次数完成后，平均转换结果将被传至数据结果寄存器 Rn 中，SC1n[COCO]置位。如果相应的 ADC 中断使能，即 [AIEN]=1，则在 SC1n[COCO]置位时会生成一个 ADC 中断。

注

当 MCU 处于等待或正常停止模式下时，硬件平均功能可以对通道执行转换。如果 SC1n[AIEN]置位，当硬件平均功能完成时，此 ADC 中断会唤醒 MCU。

33.5.5 自动比较功能

可以配置比较功能，检查结果是小于、大小还是等于单个比较值，或者，结果是处于两个比较值确定的范围之内，还是在该范围之外。

比较模式取决于 SC2[ACFGT]、SC2[ACREN]以及比较值寄存器 CV1 和 CV2 中的值。在对输入采样和转换后，CV1 和 CV2 中的比较值被用于下表描述的用途。共有 6 个比较模式，如下表所示。

表 33-83. 比较模式

SC2[ACFGT]	SC2[ACREN]	ADCCV1 (相对于 ADCCV2)	功能	比较模式描述
0	0	—	小于阈值	如果结果小于 CV1 寄存器中的值，则比较结果为真。
1	0	—	大于或等于阈值	如果结果大于或等于 CV1 寄存器中的值，则比较结果为真。
0	1	小于或等于	范围外 (不含)	如果结果小于 CV1 或者结果大于 CV2，则比较结果为真。
0	1	大于	范围内 (不含)	如果结果小于 CV1 且结果大于 CV2，则比较结果为真。
1	1	小于或等于	范围内 (含)	如果结果大于或等于 CV1 且结果小于或等于 CV2，则比较结果为真。
1	1	大于	范围外 (含)	如果结果大于或等于 CV1 或结果小于或等于 CV2，则比较结果为真。

当 SC2[ACREN] =1 时，如果 CV1 的值小于或等于 CV2 的值，则置位 SC2[ACFGT] 会选择“若在比较范围内 (含端点) 则触发”(trigger-if-inside-compare-range inclusive-of-endpoints)功能。对 SC2[ACFGT]清零会选择“若不在比较范围内 (不含端点) 则触发”(trigger-if-outside-compare-range, not-inclusive-of-endpoints)功能。

如果 CV1 大于 CV2 的值，则置位 SC2[ACFGT]清零会选择“若不在比较范围内（含端点）则触发”(trigger-if-outside-compare-range, inclusive-of-endpoints)功能。对 SC2[ACFGT]清零会选择“若在比较范围内（不含端点）则触发”(trigger-if-inside-compare-range, not-inclusive-of-endpoints)功能。

如果所选条件成立，则 SC1n[COCO]置位。

比较功能使能，且完成转换后，如果比较条件不成立，则 SC1n[COCO]不置位且转换结果数据不会传输到结果寄存器 Rn。如果硬件平均功能使能，则比较功能会将平均结果与比较值进行比较。适用相同的比较函数定义。当 SC1n[COCO]置位时且相应 ADC 中断使能，即 SC1n[AIEN]=1，就会生成一个 ADC 中断。

注

在 MCU 处于等待或正常停止模式时，比较功能可监测通道上的电压。在比较条件得到满足时，ADC 中断会唤醒 MCU。

33.5.6 校准功能

ADC 含有一个实现指定精度需要的自校准功能。

必须在任何复位之后且转换开始之前运行校准，或写入有效校准值。校准功能设置偏移校准值、反相端校准值和同相端校准值。偏移校准值自动存储在 ADC 偏移校正寄存器 (OFS) 中，同相端和反相端校准值自动存储在 ADC 同相端和反相端校准寄存器 CLPx 和 CLMx 中。用户必须在校准前正确配置 ADC，还必须生成同相端和反相端增益校准结果，并在校准功能完成后将其存储在 ADC 同相端增益寄存器 (PG) 中。

校准前，用户必须根据应用的时钟源可用性和需求，配置 ADC 的时钟源和频率、低功耗配置、电压基准选择、采样时间和高速配置。如果应用要在多种配置下使用 ADC，则应选择可以取得最高精度的配置，也可以针对不同配置进行多次校准。要获得最佳校准结果：

- 将硬件均值设为最大值，即是说，使 SC3[AVGE] = 1 且 SC3[AVGS] = 11，可得均值 32
- 将 ADC 时钟频率 f_{ADCK} 设为小于或等于 4 MHz
- $V_{REFH} = V_{DDA}$
- 在额定电压和温度下校准

在校准功能执行过程中，输入通道、转换模式连续功能、比较功能、分辨率模式和差分/单端模式都将被忽略。

要启动校准，用户需使 SC3[CAL]置位，如果 SC2[ADTRG]为 0，校准将自动开始。如果 SC2[ADTRG]为 1，SC3[CAL]不会置位，SC3[CALF]会置位。当校准有效时，不能对 ADC 寄存器执行写操作，并且不能进入停止模式，否则校准例程将终止，导致 SC3[CAL]清零且 SC3[CALF]置位。在校准序列结束时，SC1n[COCO]将置位。可以使用 SC1n[AIEN]以允许在校准序列结束时发生中断。在校准例程结束时，如果 SC3[CALF]未置位，则自动校准例程将成功执行完毕。

要完成校准，用户必须通过以下程序生成增益校准值：

1. 初始化或清除 RAM 中的一个 16 位变量。
2. 将同相端校准结果 CLP0、CLP1、CLP2、CLP3、CLP4 和 CLPS 添加到变量中。
3. 将该变量除以 2。
4. 将该变量的 MSB 置位。
5. 要实现上述两步，可以将进位位置位，并先后通过高字节和低字节上的进位位旋转至右侧。
6. 把值存储在同相端增益校准寄存器 PG 中。
7. 对反相端增益校准值重复以上程序。

当校准完成时，用户可以根据需要重新配置并使用 ADC。如果需要，可以通过清除并再次置位 SC3[CAL]，进行二次校准。

整体而言，校准例程可能需要多达 14k 个 ADCK 周期和 100 个总线周期，具体取决于结果和所选时钟源。对于 8 MHz 时钟源，这相当于约 1.7 ms。为了减小该延迟，经过偏移的校准值、同相端和反相端增益以及同相端和反相端校准值可在初始校准之后存储在 flash 存储器中，并在首次 ADC 转换之前恢复。通过这种方法，对于后续的所有加电、复位或低功耗停止模式恢复操作，均可将校准延迟缩短至 20 次寄存器存储操作。

有关校准程序的更多信息，请参见《[AN3949: ADC16 校准程序与可编程延迟模块同步](#)》的“校准”部分。

33.5.7 用户自定义偏移功能

OFS 含有 s 用户选择的或者校准生成的偏移误差校正值。

此寄存器为二进制补码、左对齐级联而成。从转换中减去 OFS 中的值，把结果传输至结果寄存器 Rn 中。如果结果大于最大结果值或小于最小结果值，则会强制应用当前操作模式的相应限值。

OFS 的格式与数据结果寄存器 Rn 不同，以便保存校准值的分辨率，并且不受所选转换模式的影响。在较低分辨率模式下，将忽略低阶位。例如，在 8 位单端模式下，将从 D[7:0] 中减去 OFS[14:7]；OFS[15] 表示符号（负数将与结果相加），OFS[6:0] 则会被忽略。相同的位用于 9 位差分模式，因为 OFS[15] 表示符号位，映射至 D[8]。对于 16 位差分模式，将直接从转换结果数据 D[15:0] 中减去 OFS[15:0]。在 16 位单端模式下，OFS 中没有与最小有效结果 D[0] 相对应的字段，因此，无法从结果中减去奇数值（-1 或 +1）。

自校准序列一旦完成 (SC3[CAL] 清零)，OFS 将根据校准要求自动置位。如有必要，用户可以写入 OFS 以覆盖校准结果。如果用户向 OFS 写入一个不同于校准值的值，则可能达不到 ADC 误差规格的要求。建议在以用户指定值覆盖之前，将校准功能生成的值存储在存储器中。

注

用户可以设定的偏移值存在有效限制。如果偏移量过高，则转换结果将以限值为最大值。

偏移校准功能可能会被用户用来删除应用偏移或 DC 偏置值。OFS 以数字 2 的补码格式写入，且该偏移将从结果或硬件平均值中减去。为添加偏移，将负偏移以 2 的补码格式存储，将成为有效的添加。结果值在范围外的偏移纠正将会强制变为最小或最大值。单端转换的最小值为 0x0000；差分转换时为 0x8000。

为保持准确性，最初存储在 OFS 中的校准偏移值必须添加至用户定义偏移。对运行过程中可能重复更改偏移的应用，将初始偏移校准值存储在 flash 中，以便恢复或添加至任何用户偏移调整值和 OFS 中存储的总和。

33.5.8 温度传感器

ADC 模块具有一个温度传感器，其输出与某一个特定的 ADC 模拟通道输入相连。下面的等式是温度传感器的近似传送函数。

$$\text{Temp} = 25 - \left(\left(V_{\text{TEMP}} - V_{\text{TEMP}25} \right) \div m \right)$$

等式 2. 温度传感器的近似传送函数

其中：

- V_{TEMP} 表示温度传感器通道在环境温度下的电压。
- $V_{\text{TEMP}25}$ 是温度传感器通道在 25°C 时的电压。
- m 为器件数据手册中的温度传感器斜率。它表示在热或冷环境下，电压与温度的斜率，单位是 $V/^\circ\text{C}$ 。

计算温度时，使用“ADC 电气”表中的 V_{TEMP25} 和温度传感器斜率值。

在应用模式下，用户可以对温度传感器通道执行读操作，计算 V_{TEMP} 并与 V_{TEMP25} 进行比较。如果 V_{TEMP} 大于 V_{TEMP25} ，则在上述等式中应用冷斜率值。如果 V_{TEMP} 小于 V_{TEMP25} ，则在上述等式中应用热斜率值。“ADC 电气”表可能仅指定一个温度传感器斜率值。这种情况下，用户可以在运行温度范围计算中使用相同的斜率。

有关温度传感器使用的更多信息，请参见名为 *HCS08* 微控制器系列使用的温度传感器的应用笔记（文档 AN3031）。

33.5.9 MCU 等待模式下的操作

等待模式是低功耗的待机模式，通过该模式可迅速恢复，因为时钟源保持有效。

如果转换在 MCU 进入等待模式时正在进行，则它会一直继续，直到完成。MCU 因硬件触发而处于等待模式或连续转换已使能的情况下，可发起转换。

在等待模式下，可选择总线时钟、ADACK 和备选时钟源作为转换时钟源。在等待模式下用 ALTCLK 作为转换时钟源取决于针对此 MCU 的 ALTCLK 定义。有关此 MCU 特定的 ALTCLK，请参见芯片配置信息。

如果比较和硬件平均功能被禁用，则一个转换完成事件将置位 $SC1n[COCO]$ ，如果相应的 ADC 中断使能（即当 $SC1n[AIEN]=1$ 时），同时还会生成一个 ADC 中断，以从等待模式唤醒 MCU。如果硬件平均功能使能，则 $SC1n[COCO]$ 会置位，并且当完成所选数量的转换时，会在使能的情况下产生一个中断。如果比较功能使能，则 $SC1n[COCO]$ 会置位，并且仅当符合比较条件时才会使能的情况下产生一个中断。如果选择的是单次转换并且不符合比较触发条件，则 ADC 会返回空闲状态并且不能从等待模式唤醒 MCU，硬件触发器启动了新的转换时除外。

33.5.10 MCU 在普通停止模式下的操作

停止模式是低功耗待机模式，在此模式下，MCU 上的大多数甚至全部时钟源都被禁用。

33.5.10.1 在总线时钟被选中条件下的正常停止模式

如果没有选择总线时钟作为转换时钟，则执行停止指令会中止当前转换并使 ADC 处于空闲状态。ADC 寄存器的内容（包括 Rn ）不受正常停止模式影响。退出正常停止模式后，需要一次软件或硬件触发来使转换继续。

33.5.10.2 在 ADACK 或备用时钟源被使能条件下的正常停止模式

如果选择 ADACK 或备用时钟源作为转换时钟，则 ADC 在正常停止模式期间会继续运行。有关此器件的配置信息，请参见芯片特性的 ADC 信息。

如果转换在 MCU 进入正常停止模式时正在进行，则它会一直继续，直到完成。MCU 因硬件触发而处于正常停止模式或连续转换已使能的情况下，可发起转换。

如果比较和硬件平均功能被禁用，则一个转换完成事件将置位 SC1n[COCO]，如果相应的 ADC 中断使能（即当 SC1n[AIEN]=1 时），同时还会生成一个 ADC 中断，以从正常停止模式唤醒 MCU。结果寄存器 Rn 会含有来自正常停止模式期间首次完成的转换的数据。如果硬件平均功能使能，则 SC1n[COCO]会置位，并且当完成所选数量的转换时，会在使能的情况下产生一个中断。如果比较功能使能，则 SC1n[COCO]会置位，并且仅当符合比较条件时才会使能的情况下产生一个中断。如果选择的是单次转换并且不符合比较条件，则 ADC 会返回空闲状态并且不能从正常停止模式唤醒 MCU，另一硬件触发器启动了新的转换时除外。

33.5.11 MCU 低功耗停止模式操作

当 MCU 进入低功耗停止模式时，ADC 模块自动禁用。

退出低功耗停止模式后，所有模块寄存器均包含其复位值。因此，退出低功耗停止模式后，必须重新使能和配置该模块。

注

有关特定芯片的工作模式，请参见有关此器件的电源管理信息。

33.6 初始化信息

本节将举例说明 ADC 模块的初始化和配置步骤。

用户可以将模块配置为 16 位、12 位、10 位或 8 位单端分辨率或 16 位、13 位、11 位或 9 位差分分辨率单一或连续转换以及轮询或中断方法，还有其他许多选项可供选择。有关本例中所使用的信息，请参见表 33-78、表 33-79 和表 33-80。

注

十六进制值的前缀是 0x，二进制值的前缀是%，十进制值无前缀字符。

33.6.1 ADC 模块初始化示例

33.6.1.1 初始化序列

在使用 ADC 模块完成转换前，首先必须执行初始化过程。常见序列为：

1. 根据[校准功能](#)中的校准说明，校准 ADC。
2. 更新 CFG 以选择输入时钟源以及生成 ADCK 所使用的分频比。此寄存器还可用于选择采样时间和低功耗配置。
3. 更新 SC2 以选择硬件或软件转换触发、比较功能选项（如果使能）。
4. 更新 SC3 以选择转换为连续还是仅完成一次(ADCO)，以及是否执行硬件平均功能。
5. 更新 SC1:SC1n 寄存器以选择转换为单端还是差分，以及是使能还是禁用转换完成中断。另外，选择可用于执行转换的输入通道。

33.6.1.2 伪码示例

在本例中，ADC 模块设置为启用中断，并以低功率、长采样时间在输入通道 1 上执行单一的 10 位转换，其中 ADCK 是总线时钟的 1 分频。

CFG1 = 0x98 (%10011000)

```

Bit 7   ADLPC   1   Configures for low power, lowers maximum clock speed.
Bit 6:5 ADIV   00   Sets the ADCK to the input clock ÷ 1.
Bit 4   ADLSMP  1   Configures for long sample time.
Bit 3:2  MODE   10   Selects the single-ended 10-bit conversion, differential 11-
bit conversion.
Bit 1:0  ADICLK 00   Selects the bus clock.

```

SC2 = 0x00 (%00000000)

```

Bit 7   ADOACT  0   Flag indicates if a conversion is in progress.
Bit 6   ADTRG  0   Software trigger selected.
Bit 5   ACFE   0   Compare function disabled.
Bit 4   ACFG   0   Not used in this example.
Bit 3   ACREN  0   Compare range disabled.
Bit 2   DMAEN  0   DMA request disabled.
Bit 1:0 REFSEL 00   Selects default voltage reference pin pair (External pins
VREFH and VREFL).

```

SC1A = 0x41 (%01000001)

```

Bit 7   COCO   0   Read-only flag which is set when a conversion completes.
Bit 6   AIEN   1   Conversion complete interrupt enabled.
Bit 5   DIFF  0   Single-ended conversion selected.
Bit 4:0 ADCH   00001 Input channel 1 selected as ADC input channel.

```

RA = 0xxx

Holds results of conversion.

CV = 0xxx

Holds compare value when compare function enabled.

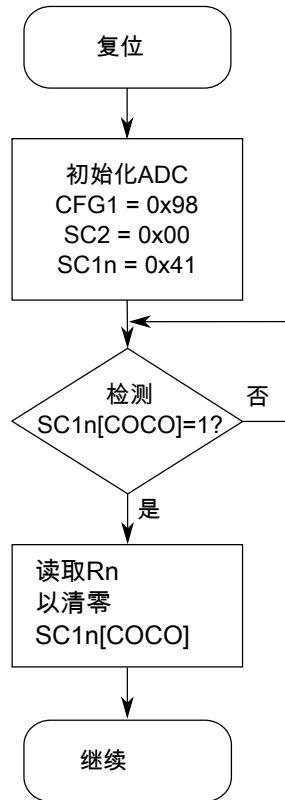


图 33-63. 初始化流程图示例

33.7 应用信息

ADC 设计成集成到微控制器中，用于需要 ADC 的嵌入式控制应用。

有关如何选择最佳外部组件值和转换器参数的指导意见，请参见《AN4373: SAR ADC 测量操作手册》。

33.7.1 外部引脚和布线

33.7.1.1 模拟电源引脚

根据器件的不同，ADC 模块的模拟电源和接地电源 (V_{DDA} 和 V_{SSA}) 如下：

- V_{DDA} 和 V_{SSA} 为独立引脚——如果用在单独的引脚上， V_{DDA} 和 V_{SSA} 必须连接到与其对应的 MCU 数字电源 (V_{DD} 和 V_{SS}) 相同的电位上，并且布线时必须非常小心，以最大限度提高抗噪性并且旁路电容必须尽可能靠近封装放置。
- V_{SSA} 与 MCU 数字 V_{SS} 共用同一引脚。
- V_{SSA} 和 V_{DDA} 共享 MCU 数字电源引脚——在这些情况下，在引脚上（与对应的数字电源相同）为模拟电源粘合单独的芯片，以确保模拟电源和数字电源间保持一定的隔离。

如果分别使用单独的模拟电源和数字电源，则两者之间的接地连接必须位于 V_{SSA} 引脚上。这必须是两个电源之间唯一的接地连接（如果可能）。 V_{SSA} 引脚是一个很好的单点接地位置。

33.7.1.2 模拟参考电压引脚

除模拟电源之外，ADC 模块还具有用于供转换器使用的两个参考电压输入的连接：

- V_{REFSH} 是转换器的高参考电压。
- V_{REFSL} 是转换器的低参考电压。

可将 ADC 配置为接受 V_{REFSH} 和 V_{REFSL} 的两个参考电压对之一。每对含有一个正参考电压和一个接地参考电压。两个参考电压对分别为外部参考电压对 V_{REFH} 和 V_{REFL} ，以及备用参考电压对 V_{ALTH} 和 V_{ALTTL} 。这些参考电压可用 $SC2[REFSEL]$ 来选择。备用参考电压对 V_{ALTH} 和 V_{ALTTL} 可以根据 MCU 的配置选择额外的外部引脚或内部源。有关此 MCU 特有的参考电压，请参见芯片配置信息。

在有些封装中，外部或备用对均在封装中分别连接至 V_{DDA} 和 V_{SSA} 。在有些器件上，其中一个正参考电压可与 V_{DDA} 共用一个引脚。在有些器件上，其中一个接地参考电压可与 V_{SSA} 共用一个引脚。

如果采用外部源，则正参考电压可以连接至与 V_{DDA} 相同的电位，也可以由外部源驱动至最小高电平参考电压与 V_{DDA} 电位之间的某个电平。正参考电压不得超过 V_{DDA} 。如果采用外部源，则接地参考电压必须连接至与 V_{SSA} 相同的电位。参考电压对在布线时必须非常小心，以最大限度提高抗噪性并且旁路电容必须尽可能靠近封装放置。

需要在每个逐次逼近步骤将电荷供应给电容阵列的交流电流（采用电流尖脉冲的形式）通过 V_{REFH} 和 V_{REFL} 环路消耗。满足这一电流要求的最佳外部元件为具有良好高频特性的 $0.1 \mu\text{F}$ 电容。该电容在 V_{REFH} 和 V_{REFL} 之间连接，并且必须尽量靠近封装引脚放置。该路径中不建议采用电阻，因为电流会引起电压降，这可能导致转换错误。该路径中的电感必须为最小值，即仅为寄生。

33.7.1.3 模拟输入引脚

外部模拟输入通常由 MCU 器件上的数字 I/O 引脚共用。

经验数据表明，当存在噪声或者源阻抗较高时，模拟输入上的电容可以改善性能。使用具有良好高频特性的 0.01 μF 电容就足够了。在某些情况下并不一定要使用这类电容，但是一旦使用，则必须将其尽可能地靠近封装引脚放置并将 V_{SSA} 作为基准电压。

为确保转换正确完成，输入电压必须介于 V_{REFH} 和 V_{REFL} 之间。如果输入电压等于或大于 V_{REFH} ，转换器电路会将信号转换为 0xFFF (满刻度 12 位表示法)、0x3FF (满刻度 10 位表示法) 或 0xFF (满刻度 8 位表示法)。如果输入电压等于或小于 V_{REFL} ，则转换器电路会将其转换为 0x000。在 V_{REFH} 和 V_{REFL} 之间的输入电压属于直线线性转换。采用电容进行充电时，存在与 V_{REFL} 有关的短时电流。

为了尽量减小电流注入造成的准确度损失，转换过程中不得对与模拟输入引脚相邻的引脚进行电平转换操作。

33.7.2 误差来源

33.7.2.1 采样误差

为正确转换，输入的采样时间必须长到能实现正确的精度。

$$\text{RAS} + \text{RADIN} = \text{SC} / (\text{FMAX} * \text{NUMTAU} * \text{CADIN})$$

图 33-64. 采样等式

其中：

RAS = 外部模拟源电阻

SC = 采样窗口中使用的 ADCK 周期数

CADIN = 内部 ADC 输入电容

NUMTAU = $-\ln(\text{LSBERR} / 2^N)$

LSBERR = 可接受采样错误值 (单位: LSB)

N = 8 (8 位模式)、10 (10 位模式)、12 (12 位模式) 或 16 (16 位模式)

通过设置 CFG1[ADLSMP]和更改 CFG2[ADLSTS], 以扩大采样窗口, 或减小 ADCK 频率以增加采样时间, 可提高源电阻或采样精度。

33.7.2.2 引脚漏电误差

如果外部模拟源电阻 R_{AS} 偏高, I/O 引脚上的漏电流可能会导致转换出错。如果应用无法接受这一误差, 则使 R_{AS} 低于 $V_{REFH} / (4 \times I_{LEAK} \times 2^N)$ 以便将漏电电流误差控制在 1/4 LSB 以外 ($N = 8$, 8 位模式; 10, 10 位模式; 12, 12 位模式; 或者 16, 16 位模式)。

33.7.2.3 噪声性误差

采样或转换过程中出现的系统噪音会影响转换的准确度。只有在满足以下条件时, 才能保证 ADC 准确度值达到预期:

- V_{REFH} 到 V_{REFL} 之间有一个 0.1 μF 的低 ESR 电容器。
- V_{DDA} 到 V_{SSA} 之间有一个 0.1 μF 的低 ESR 电容器。
- 如果主电源使用了电感隔离, 则 V_{DDA} 到 V_{SSA} 之间还要放一个 1 μF 的电容器。
- V_{SSA} 和 V_{REFL} 如果连接, 连接到接地层中某安静点处的 V_{SS} 。
- 在发起 (硬件触发转换) ADC 转换前, 或发起 (硬件或软件触发转换) ADC 转换后立即让 MCU 在等待或正常停止模式下运行。
 - 对于软件触发的转换, 对 SC1 执行写入操作后立即使用等待指令或停止指令。
 - 对于正常停止模式, 选择 ADACK 或备选时钟作为时钟源。在正常停止模式下运行可减小 V_{DD} 噪音, 但也会因停止恢复而增加有效转换时间。
- 转换期间, MCU 上无 I/O 切换、输入或输出。

在某些情况中, 外部系统的活动会引起辐射噪声或传导噪声的传播, 或使过大的 V_{DD} 噪声耦合到 ADC 中。在这类情况下, 或者无法将 MCU 置于等待或停止模式, 或者无法暂停 I/O 活动, 建议通过以下动作降低噪声对精度的影响:

- 将选定输入通道上的 0.01 μF 电阻 (C_{AS}) 置于 V_{REFL} 或 V_{SSA} 处。此举可改善噪音问题, 但会影响基于外部模拟源阻抗的采样率。

- 将模拟输入接连转换多次并除以结果总数得出结果的平均值。消除一次 1 LSB 的错误所带来的影响需要四个样本。
- 通过使用异步时钟 (ADACK) 并计算平均值来降低同步噪音的影响。与 ADCK 同步的噪音不能计算平均值。

33.7.2.4 编码宽度和量化误差

ADC 将理想的线性转换函数量化为 65536 步 (在 16 位模式下)。每一步在理想情况下都具有相同的高度 (一个编码) 和宽度。宽度定义为某个编码的转变点与下一个编码的转变点之间的差值。定义为 1LSB 的 N 位转换器的理想编码宽度 (在本例中, N 可以是 16、12、10 或 8) 是:

$$1\text{LSB} = (V_{\text{REFH}}) / 2^N$$

等式 3. IN 位转换器的理想编码宽度

结果的数字化导致存在固有的量化误差。对于 8 位、10 位或 12 位转换, 线性转换函数与实际的转换函数在一些离散点上精确重合, 编码会在电压位于这些点之间的中点时发生切换。因此, 量化误差在 8 位、10 位或 12 位模式下将为 $\pm 1/2$ LSB。并且, 第一个编码 (0x000) 宽度只有 $1/2$ LSB, 最后一个编码 (0xFF 或 0x3FF) 的宽度为 1.5LSB。

对于 16 位转换, 编码只有在电压变化了一个完整的编码宽度后才会切换, 因此量化误差为 -1 LSB 到 0 LSB, 每个振幅的编码宽度为 1 LSB。

33.7.2.5 线性误差

ADC 可能还会呈现几种非线性形式。虽然已经尽全力减少这些误差, 但是系统必须知道误差的存在, 因为它们会影响总体的准确度:

- 零刻度误差 (E_{ZS}), 有时也称为偏移量: 该误差是指第一个转换的实际代码宽度与理想代码宽度之间的差值。8 位、10 位或 12 位模式下为 $1/2$ LSB, 16 位模式下为 1 LSB。若第一个转换为 0x001, 将使用实际 0x001 代码与理想值 (1 LSB) 之间的差值。
- 满量程误差 (E_{FS}): 该误差是指最后一个转换的实际代码宽度与理想代码宽度之间的差值。8 位、10 位或 12 位模式下为 1.5 LSB, 16 位模式下为 1 LSB。若最后一个转换为 0x3FE, 将使用实际 0x3FE 代码与理想值 (1 LSB) 之间的差值。

- 差分非线性 (DNL): 该误差是指所有转换的实际编码宽度与理想编码宽度之间的最差情况差值。
- 积分非线性 (INL): 该误差是指 DNL 累加和的最高值或绝对值。更简而言之, INL 是指所有编码中, 实际切换电压与其对应的理想切换电压之间的最差情况差值。
- 未调整总误差 (TUE): 该误差是指实际转换函数与理想线性转换函数之间的差值, 包括所有形式的误差。

33.7.2.6 编码抖动、非单调性和丢码

模数转换器易受三种特殊形式错误的影响:

- 编码抖动: 编码抖动发生在给定的输入电压经过重复采样后转换为两个值中的某一个值时。理想情况下, 当输入电压小于但无限接近切换电压时, 转换器会产生较低的编码 (反之亦然)。然而, 当输入电压接近切换电压时, 即使少量的系统噪声也可能会使转换器在两个编码间变得不确定。

此错误可通过重复采样输入和计算结果平均值来减少。此外, 在[噪声性误差](#)中讨论的方法也可以减少此错误。

- 非单调性: 除了代码抖动外, 当转换器为较高的输入电压转换到较低的代码时, 会出现非单调性问题。
- 失码: 失码是指不会为任何输入值进行转换的那些值。

在 8 位或 10 位模式下, ADC 保证是单调的并且无失码。

第 34 章 比较器 (CMP)

34.1 此模块的芯片实现细节

34.1.1 CMP 输入连接

下表显示了 CMP 的固定内部连接。

表 34-1. CMP 输入连接

CMP 输入	CMP0
IN0	CMP0_IN0
IN1	CMP0_IN1
IN2	CMP0_IN2
IN3	CMP0_IN3
IN4	—
IN5	CMP0_IN5
IN6	带隙 ¹
IN7	6 位 DAC0 参考

1. 这是 PMC 带隙 1V 参考电压。使用此 CMP 通道之前，要确保通过置位 PMC_REGSC[BGBE]位来使能带隙缓冲器。带隙电压(V_{BG})规格参见器件数据手册。

34.1.2 CMP 外部参考

该 6 位 DAC 子模块仅支持以下一种参考：

- $V_{DD} - V_{in1}$ 输入
- $V_{DD} - V_{in2}$ 输入

34.1.3 外部窗口/样本输入

单个 PDB 脉冲输出信号控制每个 CMP 样本/窗口时序。

34.1.4 CMP 触发模式

当 CMPx_CR1[TRIGM]置位时，CMP 和 6 位 DAC 子数据块支持触发模式操作。在触发模式使能后，触发事件会开始执行一种比较序列：先使能 CMP 和 DAC，然后执行 CMP 操作并捕捉输出。对于此器件，由 LPTMR 来控制这两个阶段的操作顺序。当 LPTMR 使能时，LPTMR 触发输出始终保持使能。首个信号用于使能 CMP 和 DAC，且在 TCF 标志位置位的同时认定。用于触发 CMP 以捕捉比较结果的第二个信号的延迟时间由 LPTMR 配置决定。在时间计数器（预分频器已使能）模式下，延迟时间为 1/2 预分频输出周期。在时间计数器（预分频器已旁置）模式下，延迟时间为 1/2 预分频时钟周期。

LPTMR 的首个信号和 LPTMR 的第二个信号之间的延迟时间必须大于器件数据手册中指定的模拟比较器初始化延迟时间。

34.2 简介

比较器模块 (CMP) 提供一个用于比较两个模拟输入电压的电路。此比较器电路适用于在整个供电电压范围内操作（即所谓的“轨到轨操作”）。

模拟 MUX (ANMUX) 提供一个用于从 8 个通道中选择模拟输入信号的电路。其中一个通道信号可由 6 位数模转换器 (DAC) 提供。多路复用器电路设计为在整个供电电压范围内工作。

6 位 DAC 是一个 64 分接头的梯形电阻网络，可为需要电压参考的应用提供可选的电压参考。此 64 分接头的梯形电阻网络将电源参考 V_{in} 分成 64 个电压电平。6 位数字信号输入可选择输出电压电平，范围从 V_{in} 至 $V_{in}/64$ 。 V_{in} 可从两个电压源 - V_{in1} 和 V_{in2} 中选择。比较器中的 6 位 DAC 只能提供片上内部信号，不能供外部引脚使用。

34.2.1 CMP 特性

CMP 具有以下特性：

- 在整个电源电压范围内的操作
- 可以轨对轨输入
- 可编程迟滞控制

- 可选择在比较器输出上升沿、下降沿或任意沿时产生中断
- 可选择翻转比较器输出
- 能够产生多种输出，比如：
 - 采样
 - 加窗，是某些 PWM 过零检测应用的理想之选
 - 数字滤波：
 - 可以旁通滤波器
 - 可通过外部 SAMPLE 信号或成比例的总线时钟计时
- 可以在输出滤波器用于内部功能时使用外部迟滞
- 两个软件可选性能级别：
 - 短传播延迟，但高功耗
 - 低功耗，但长传播延迟
- 支持 DMA 传输
 - 可以选择比较事件以触发 DMA 传输
- 可在除 VLLS0 模式外的所有工作模式下工作
- 窗口和滤波器功能在以下模式下不可用：
 - STOP
 - VLPS
 - LLS
 - VLLSx

34.2.2 6 位 DAC 关键特性

6 位 DAC 具有以下特性：

- 6 位分辨率
- 可选参考电压源
- 掉电模式，用于在未使用时节省电力
- 可选择将输出路由至内部比较器输入

34.2.3 ANMUX 关键特性

ANMUX 具有以下特性：

- 两个 8 至 1 通道多路复用器
- 在整个电源电压范围内运行

34.2.4 CMP、DAC 和 ANMUX 结构框图

下图为高速比较器、DAC 和 ANMUX 模块的结构框图。

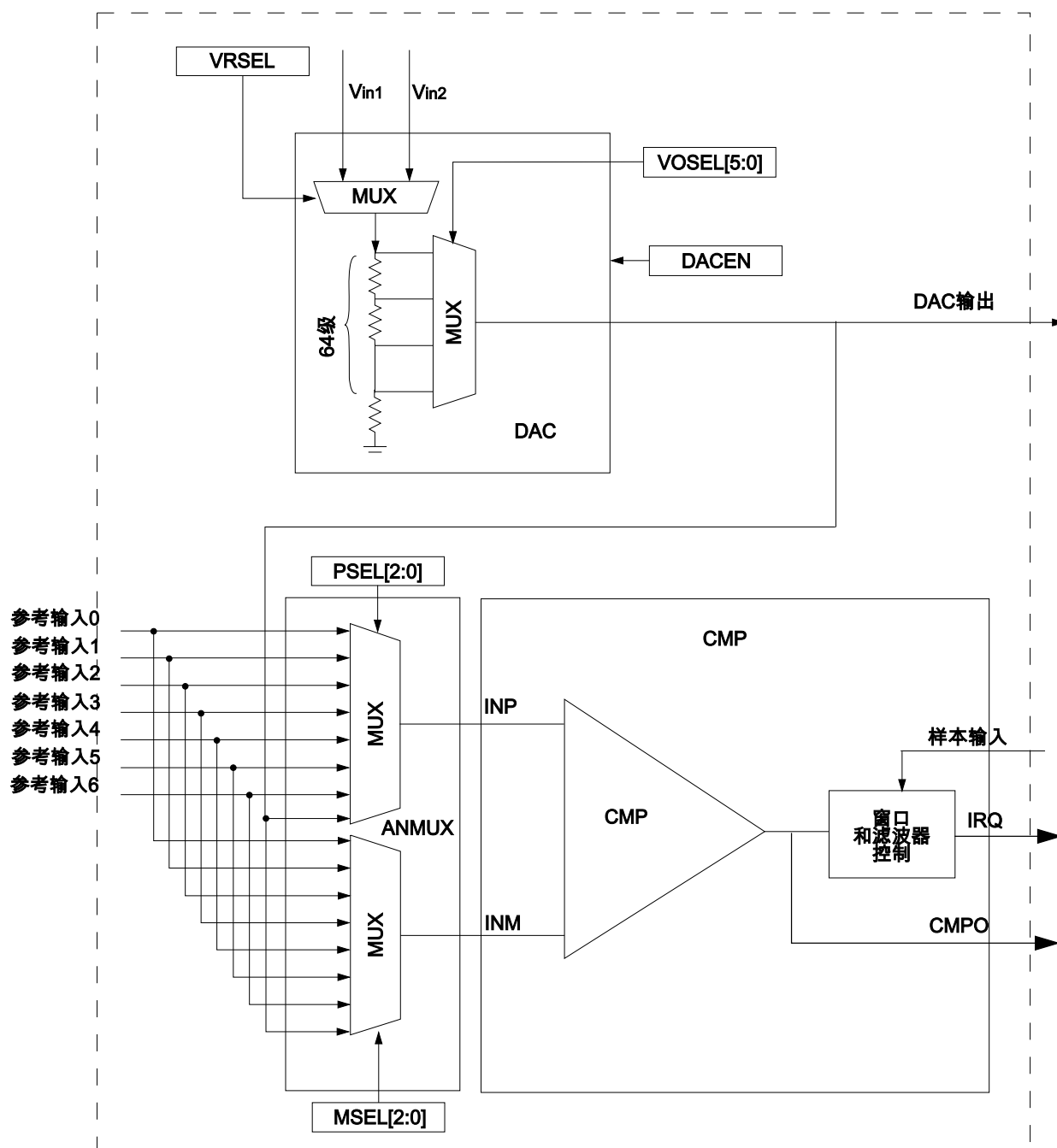


图 34-1. CMP、DAC 和 ANMUX 结构框图

34.2.5 CMP 框图

下图是 CMP 模块的结构框图。

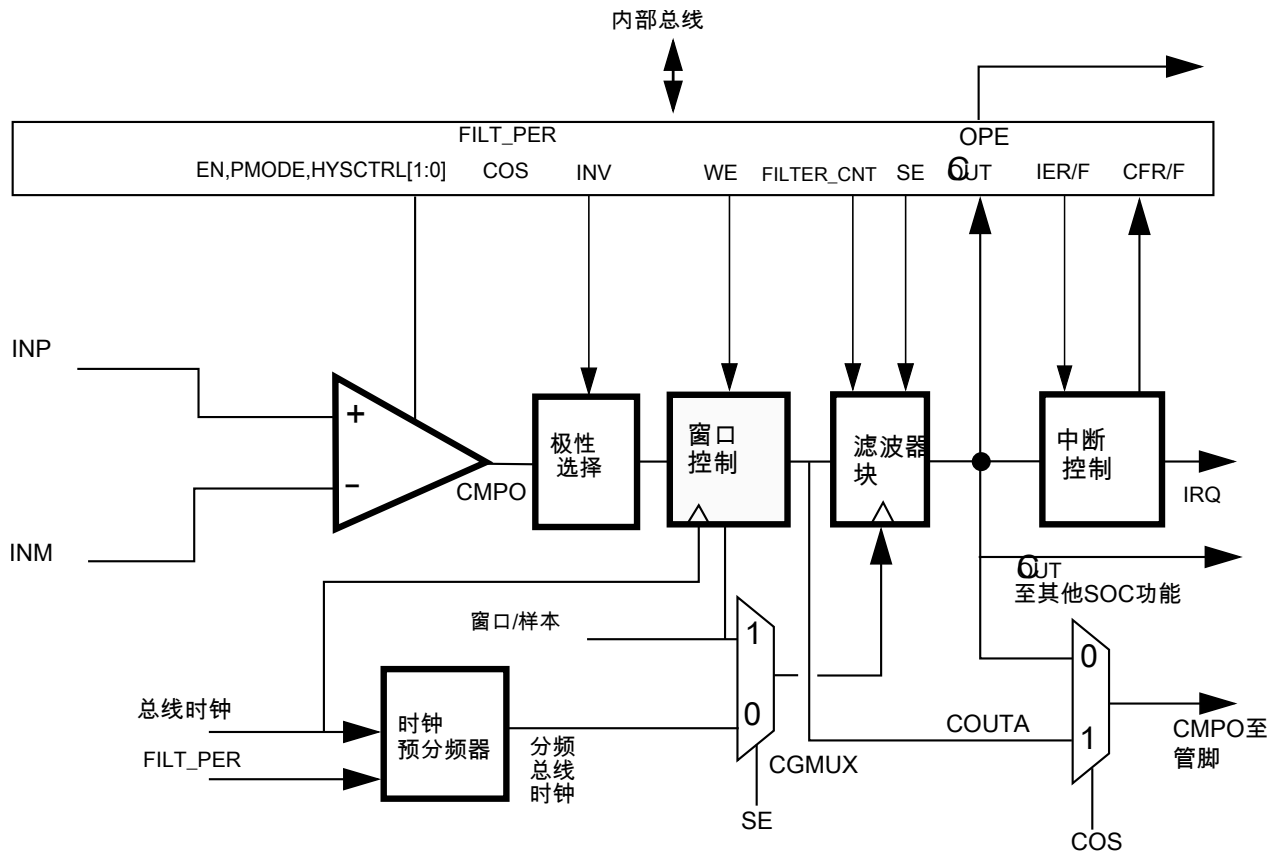


图 34-2. 比较器模块结构框图

在 CMP 框图中：

- 当 $CR1[WE] = 0$ 时，屏蔽窗口控制模块
- 若 $CR1[WE] = 1$ ，则当 $WINDOW = 1$ 时，在每个总线时钟都会对比较器输出进行采样以产生 $COUTA$ 。当 $WINDOW = 0$ 时，采样不会发生。
- 滤波器模块不用时可以旁路。
- 若旁路滤波器且 $CR0[FILTER_CNT]$ 设置为 $0x01$ ，则滤波器模块用作简单采样器。
- 当旁路滤波器且 $CR0[FILTER_CNT]$ 设置为大于 $0x01$ 的值时，滤波器模块基于多个样本进行滤波。
 - 若 $CR1[SE] = 1$ ，则外部 $SAMPLE$ 输入用作采样时钟
 - 若 $CR1[SE] = 0$ ，则分频总线时钟用作采样时钟
- 若使能，滤波器模块会给 $COUT$ 造成最多一个总线时钟的额外延迟，原因是跨越时钟域边界的 $COUT$ 必须与总线时钟重新同步。
- $CR1[WE]$ 和 $CR1[SE]$ 互斥。

34.3 存储器映射/寄存器定义

CMP 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4007_3000	CMP 控制寄存器 0 (CMP0_CR0)	8	R/W	00h	34.3.1/671
4007_3001	CMP 控制寄存器 1 (CMP0_CR1)	8	R/W	00h	34.3.2/672
4007_3002	CMP 滤波器周期寄存器 (CMP0_FPR)	8	R/W	00h	34.3.3/673
4007_3003	CMP 状态和控制寄存器 (CMP0_SCR)	8	R/W	00h	34.3.4/674
4007_3004	DAC 控制寄存器 (CMP0_DACCR)	8	R/W	00h	34.3.5/675
4007_3005	MUX 控制寄存器 (CMP0_MUXCR)	8	R/W	00h	34.3.6/675

34.3.1 CMP 控制寄存器 0 (CMPx_CR0)

地址: 4007_3000h 基准 + 0h 偏移 = 4007_3000h

位	7	6	5	4	3	2	1	0	
读	0	FILTER_CNT				0	0	HYSTCTR	
写									
复位	0	0	0	0	0	0	0	0	

CMPx_CR0 字段描述

字段	描述
7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
6-4 FILTER_CNT	滤波器样品数 表示比较器输出滤波器验收新输出状态前必须有效的连续样品数量。有关滤波器编程和延迟的信息，请参考 功能说明 。 000 滤波器已禁用。如果 SE = 1，则 COUT 为逻辑 0。这不是合法状态，我们也不推荐。如果 SE = 0，则 COUT = COUTA。 001 必须有一个有效的样品。对比较器输出进行简单采样。 010 必须有两个有效的连续样品。 011 必须有三个有效的连续样品。 100 必须有四个有效的连续样品。 101 必须有五个有效的连续样品。 110 必须有六个有效的连续样品。 111 必须有七个有效的连续样品。
3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
HYSTCTR	比较器硬模块迟滞控制

下一页继续介绍此表...

CMPx_CR0 字段描述 (继续)

字段	描述
	定义可编程迟滞级别。每个级别的迟滞值取决于具体设备。查看设备的数据手册获取准确值。
00	0 级
01	1 级
10	2 级
11	3 级

34.3.2 CMP 控制寄存器 1 (CMPx_CR1)

地址: 4007_3000h 基准 + 1h 偏移 = 4007_3001h

位	7	6	5	4	3	2	1	0
读	SE	WE	TRIGM	PMODE	INV	COS	OPE	EN
写								
复位	0	0	0	0	0	0	0	0

CMPx_CR1 字段描述

字段	描述
7 SE	<p>样品使能</p> <p>在给定的时间范围内，SE 或 WE 均可设置。若寄存器写入尝试同时设置 SE 和 WE，则两者清零。但是，避免在两个字段位置写入 1，因为“11”已保留，在后续版本中可能会更改。</p> <p>0 采样模式未选择。 1 采样模式已选择。</p>
6 WE	<p>窗口使能</p> <p>在给定的时间范围内，SE 或 WE 均可设置。若寄存器写入尝试同时设置 SE 和 WE，则两者清零。但是，避免在两个字段位置写入 1，因为“11”已保留，在后续版本中可能会更改。</p> <p>0 窗口模式未选择。 1 窗口模式已选择。</p>
5 TRIGM	<p>触发模式使能</p> <p>CMP_CR1[TRIGM]设置为 1 时，CMP 和 DAC 配置为 CMP 触发模式。此外，应使能 CMP。若 DAC 用作 CMP 的参考，也应使能。</p> <p>CMP 触发模式根据外部定时器资源定期使能 CMP 和 6 位 DAC，以生成触发比较。</p> <p>TRIGM 设置完成后，CMP 和 DAC 设为待机状态，直至收到外部定时器资源触发。</p> <p>有关外部定时器资源的详细信息，请参考芯片配置。</p> <p>0 触发模式已禁用。 1 触发模式使能。</p>
4 PMODE	<p>功耗模式选择</p> <p>参考设备数据手册中的电气规格表，以获取详细信息。</p>

下一页继续介绍此表...

CMPx_CR1 字段描述 (继续)

字段	描述
	0 已选择低速 (LS) 比较模式。在该模式中, CMP 的输出传播延迟较慢, 且电流消耗也较低。 1 已选择高速 (HS) 比较模式。在该模式中, CMP 的输出传播延迟较快, 且电流消耗也较高。
3 INV	比较器反相 允许选择模拟比较器功能的极性。若 OPE=0, 它还能驱动 COUT 输出, 在器件引脚上或作为 SCR[COUT]皆可。 0 不会反相比较器输出。 1 反相比较器输出。
2 COS	比较器输出选择 0 将过滤的比较器输出 (CMPO) 设置为等于 COUT。 1 将未过滤的比较器输出 (CMPO) 设置为等于 COUTA。
1 OPE	比较器输出引脚使能 0 CMPO 在相应的 CMPO 输出引脚上不可用。若比较器无法控制引脚, 则该字段无影响。 1 CMPO 在相关的 CMPO 输出引脚上可用。 若比较器控制引脚, 则相应 CMPO 输出引脚上的比较器输出 (CMPO) 消除。若比较器无法控制字段, 则该位无影响。
0 EN	比较器模块使能 使能模拟比较器模块。若模块未使能, 则保持在关闭状态, 不产生任何功耗。若用户选择从模拟多路复用器至正极端口或负极端口的相同输入, 则比较器自动禁用。 0 模拟比较器已禁用。 1 模拟比较器使能。

34.3.3 CMP 滤波器周期寄存器 (CMPx_FPR)

地址: 4007_3000h 基准 + 2h 偏移 = 4007_3002h

位	7	6	5	4	3	2	1	0
读	FILT_PER							
写	FILT_PER							
复位	0	0	0	0	0	0	0	0

CMPx_FPR 字段描述

字段	描述
FILT_PER	滤波器采样周期 CR1[SE]=0 时, 在比较器输出滤波器的总线时钟周期中, 指定样品周期。将 FILT_PER 设置为 0x0 以禁用滤波器。滤波器编程和延迟的详细信息请参考 功能说明 。 CR1[SE]=1 时, 该字段无效。这种情况下, 外部 SAMPLE 信号用来确定采样周期。

34.3.4 CMP 状态和控制寄存器 (CMPx_SCR)

地址: 4007_3000h 基准 + 3h 偏移 = 4007_3003h

位	7	6	5	4	3	2	1	0
读	0	DMAEN	0	IER	IEF	CFR	CFF	COUT
写						w1c	w1c	
复位	0	0	0	0	0	0	0	0

CMPx_SCR 字段描述

字段	描述
7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
6 DMAEN	DMA 使能控制 使能 CMP 模块触发的 DMA 传输。在该字段已设置的情况下，设置好 CFR 或 CFF 即可置位 DMA 请求。 0 DMA 已禁用。 1 DMA 使能。
5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
4 IER	比较器中断使能上升 使能 CMP 的 CFR 中断。该字段设置完成后，CFR 设置后将置位中断。 0 中断已禁用。 1 中断使能。
3 IEF	比较器中断使能下降 使能 CMP 的 CFF 中断。该字段设置完成后，CFF 设置后将置位中断。 0 中断已禁用。 1 中断使能。
2 CFR	模拟比较器标志上升 正常运行过程中进行设置时，在 COUT 上检测到上升沿。将 CFR 清零的方法是将 1 写入该位。在停止模式下，CFR 为电平触发来编程为边沿或电平触发。 0 COUT 上未检测到上升沿。 1 COUT 上已出现上升沿。
1 CFF	模拟比较器标志下降 正常运行过程中进行设置时，在 COUT 上检测到下降沿。将 CFF 清零的方法是将 1 写入该位。停止模式下，CFF 为电平触发来编程为边沿或电平触发。 0 COUT 上未检测到下降沿。 1 COUT 上已出现下降沿。
0 COUT	模拟比较器输出 读取时将返回模拟比较器输出的当前值。模拟比较器模块禁用后，字段重置为 0，并读取为 CR1[INV]，即 CR1[EN] = 0 时，该字段的写入被忽略。

34.3.5 DAC 控制寄存器 (CMPx_DACCR)

地址: 4007_3000h 基准 + 4h 偏移 = 4007_3004h

位	7	6	5	4	3	2	1	0
读								
写	DACEN	VRSEL	VOSEL					
复位	0	0	0	0	0	0	0	0

CMPx_DACCR 字段描述

字段	描述
7 DACEN	DAC 使能 使能 DAC。DAC 禁用后即降耗以节省能源。 0 DAC 禁用。 1 DAC 使能。
6 VRSEL	电源参考电压源选择 0 选择 $V_{中1}$ 作为梯形电阻网络电源电压参考。 1 选择 $V_{中2}$ 作为梯形电阻网络电源电压参考。
VOSEL	DAC 输出电压选择 从 64 种独特级别之一选择一种输出电压。 若 $DACO = (V_{in} / 64) * (VOSEL[5:0] + 1)$ ，则 DACO 范围为： $V_{中}/64$ 至 $V_{中}$ 。

34.3.6 MUX 控制寄存器 (CMPx_MUXCR)

地址: 4007_3000h 基准 + 5h 偏移 = 4007_3005h

位	7	6	5	4	3	2	1	0
读								
写	Reserved	0	PSEL			MSEL		
复位	0	0	0	0	0	0	0	0

CMPx_MUXCR 字段描述

字段	描述
7 Reserved	位仅能编程为 0。 此字段为保留字段。
6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5-3 PSEL	正输入多路复用器控制 确定为比较器正输入选择哪种输入。对于 INx 输入，请参考 CMP、DAC 和 ANMUX 结构框图。 注：若操作不当，为两种多路复用器选择了相同输入，则比较器自动关闭，以防止产生噪声。

下一页继续介绍此表...

CMPx_MUXCR 字段描述 (继续)

字段	描述
	000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7
MSEL	<p>负输入多路复用器控制</p> <p>确定为比较器负输入选择哪种输入。对于 INx 输入，请参考 CMP、DAC 和 ANMUX 结构框图。</p> <p>注：若操作不当，为两种多路复用器选择了相同输入，则比较器自动关闭，以防止产生噪声。</p> <p>000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7</p>

34.4 功能说明

CMP 模块可用来比较施加于 INP 和 INM 的两个模拟输入电压。

当正相输入大于反相输入时，CMPO 为高电平；当正相输入小于反相输入时，其为低电平。设置 CR1[INV] = 1 可有选择地使此信号反相。

SCR[IER]、SCR[IEF]用于选择导致 CMP 模块置位处理器中断的条件。SCR[CFF]在比较器输出的下降沿设置，SCR[CFR]在上升沿设置。可选滤波的 CMPO 可通过 SCR[COU]直接读取。

34.4.1 CMP 功能模式

CMP 模块有如下主要子模块：

- 比较器本身
- 窗口功能
- 滤波器功能

滤波器 CR0[FILTER_CNT]的时钟只能来自内部或外部时钟源。可以设置必须一致的样本数，达到该数量后，滤波器输出才会改变。最简单的情况是仅要求一个样本一致。这种情况下，滤波器用作一个简单的采样器。

外部样本输入通过 CR1[SE]使能。置位时，仅在样本输入的上升沿对比较器输出进行采样。

“窗口模式”通过 CR1[WE]使能。置位时，仅当 WINDOW=1 时对比较器输出进行采样。此特性可用于在输入电压无效时忽略比较器输出。当某些 PWM 应用实施过零检测时，此特性特别有用。

比较器的滤波器和采样特性详见下表。各种模式在下文讨论。

表 34-16. 比较器采样/滤波器控制

模式#	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	操作
1	0	X	X	X	X	已禁止 请参见禁用模式(# 1)。
2A	1	0	0	0x00	X	连续模式 请参见连续模式(#s 2A & 2B)。
2B	1	0	0	X	0x00	
3A	1	0	1	0x01	X	已采样、未滤波模式 请参见已采样、未滤波模式(#s 3A & 3B)。
3B	1	0	0	0x01	> 0x00	
4A	1	0	1	> 0x01	X	已采样、已滤波模式 请参见已采样、已滤波模式(#s 4A & 4B)。
4B	1	0	0	> 0x01	> 0x00	
5A	1	1	0	0x00	X	窗口模式 当 SAMPLE=1 时，在总线时钟的每个上升沿对比较器输出进行采样以产生 COUTA。 请参见窗口模式(#s 5A & 5B)。
5B	1	1	0	X	0x00	
6	1	1	0	0x01	0x01–0xFF	窗口/重新采样模式 当 SAMPLE=1 时，在总线时钟的每个上升沿对比较器输出进行采样以产生 COUTA，然后在 FILT_PER 所确定的间隔重新采样以产生 COUT。 请参见窗口/重新采样模式(# 6)。
7	1	1	0	> 0x01	0x01–0xFF	窗口/滤波模式 当 SAMPLE=1 时，在总线时钟的每个上升沿对比较器输出进行采样以产生 COUTA，然后重新采样并滤波以产生 COUT。 请参见加窗/过滤模式(#7)。
CR1[EN]、CR1[WE]、CR1[SE]、CR0[FILTER_CNT]和 FPR[FILT_PER]的所有其他组合皆非法。						

对于利用比较器来驱动故障输入的情况，例如：对于 FTM 等电机控制模块，必须将其配置为连续工作模式，使得外部故障可以立即通过比较器进入目标故障电路。

注

要改变滤波和采样设置，必须先设置 $CR1[SE]=0$ 且 $CR0[FILTER_CNT]=0x00$ 。这将把滤波器复位到已知状态。

34.4.1.1 禁用模式(# 1)

在禁用模式下，模拟比较器不工作，无功耗。在这种模式下，CMPO 为 0。

34.4.1.2 连续模式(#s 2A & 2B)

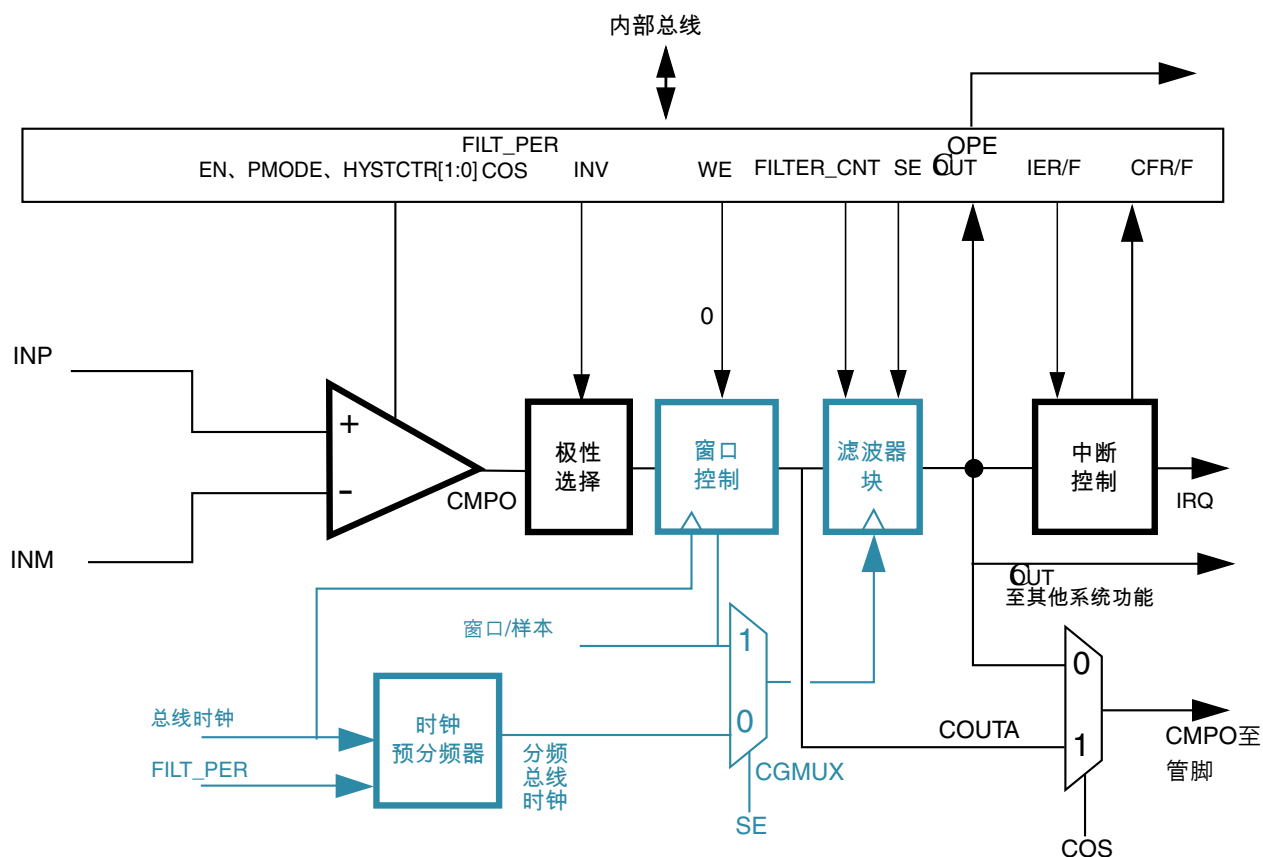


图 34-15. 比较器在连续模式下的操作

模拟比较器模块上电且有效。CMPO 可以反相，但不接受外部采样或滤波。窗口控制和滤波器模块均完全旁路。SCR[COUT]连续更新。从比较器输入引脚到输出引脚的路径以组合非时钟模式工作。COUT 和 COUTA 相同。

有关导致滤波器模块禁用的控制配置，请参见[滤波器模块旁路逻辑图](#)。

34.4.1.3 已采样、未滤波模式(#s 3A & 3B)

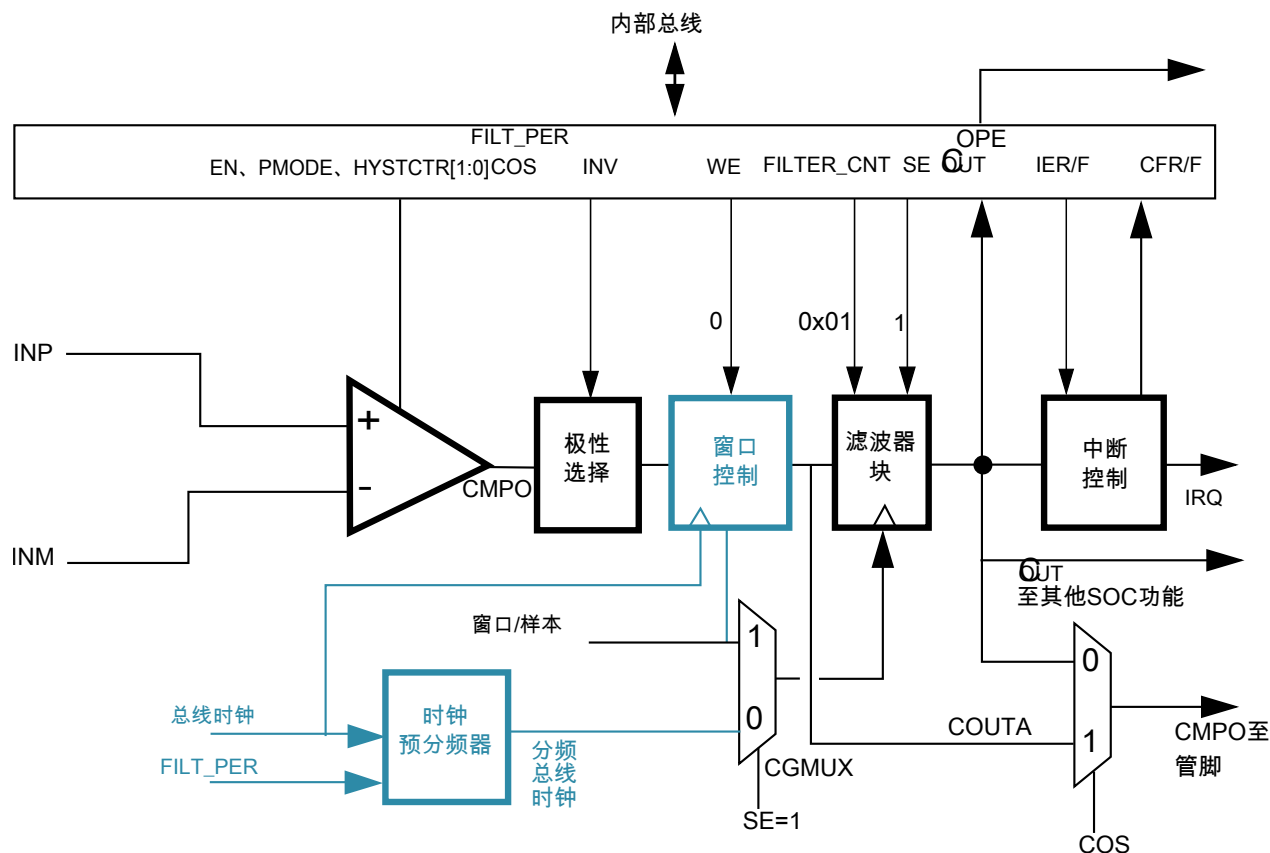


图 34-16. 已采样、未滤波(# 3A): 采样点由外部驱动

在“已采样、未滤波”模式下，模拟比较器模块上电且有效。从模拟输入到 COUTA 的路径为组合非时钟式。完全旁路窗口控制。只要在滤波器模块时钟输入上检测到上升沿，就会对 COUTA 采样。

“已采样、未滤波”(# 3A)和“已采样、未滤波”(# 3B)工作模式的唯一区别在于如何获得滤波器模块时钟。在#3A 中，滤波器模块时钟从外部获得；在#3B 中，滤波器模块时钟从内部获得。

在这种模式(# 3B)下，对比较器输出进行采样/保持是比较器滤波器的唯一功能。

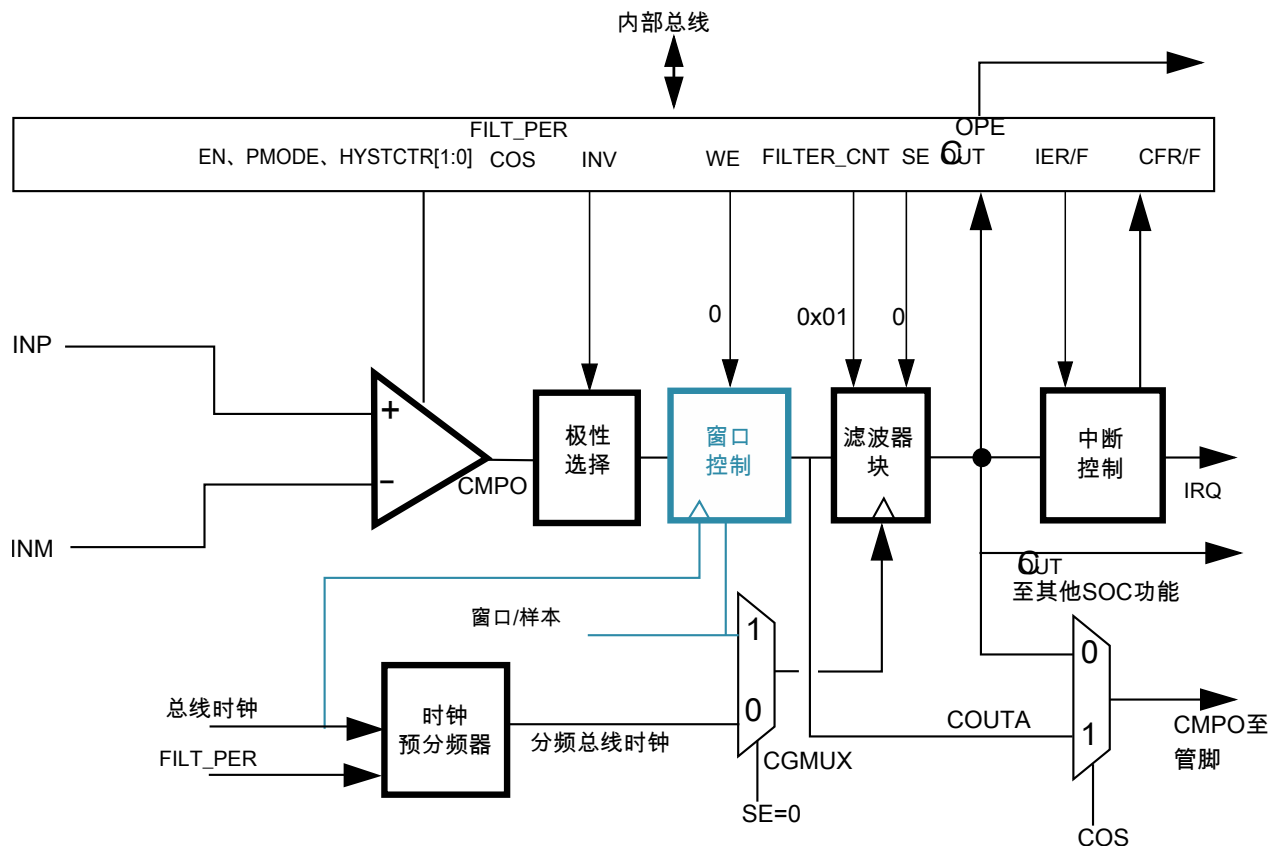


图 34-17. 已采样、未滤波(# 3B): 采样间隔从内部获得

34.4.1.4 已采样、已滤波模式(#s 4A & 4B)

在“已采样、已滤波”模式下，模拟比较器模块上电且有效。从模拟输入到 COUTA 的路径为组合非时钟式。完全旁路窗口控制。只要在滤波器模块时钟输入上检测到上升沿，就会对 COUTA 采样。

“已采样、未滤波”(# 3A)和“已采样、已滤波(# 4A)”模式的唯一区别在于 CR0[FILTER_CNT]>1，这会激活滤波器操作。

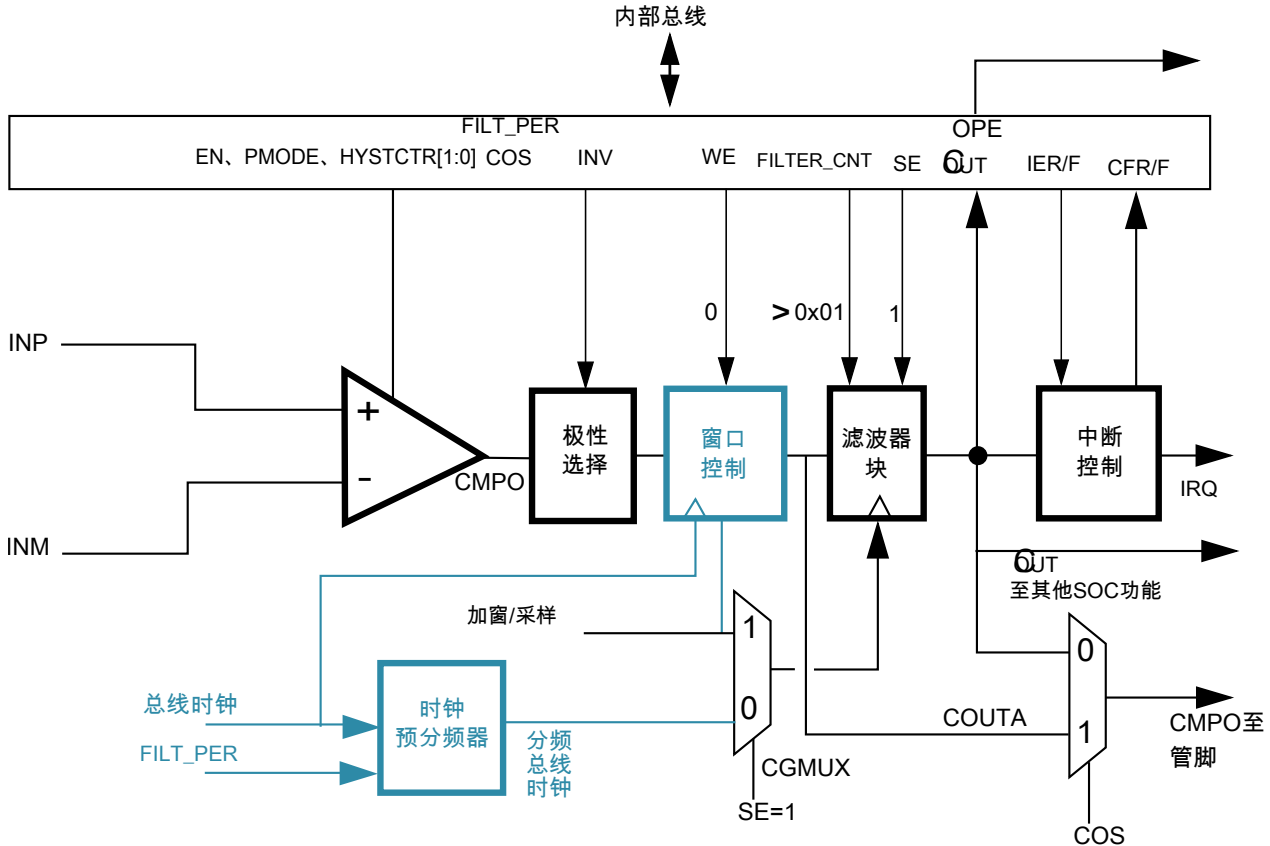


图 34-18. 已采样、已滤波(# 4A): 采样点由外部驱动

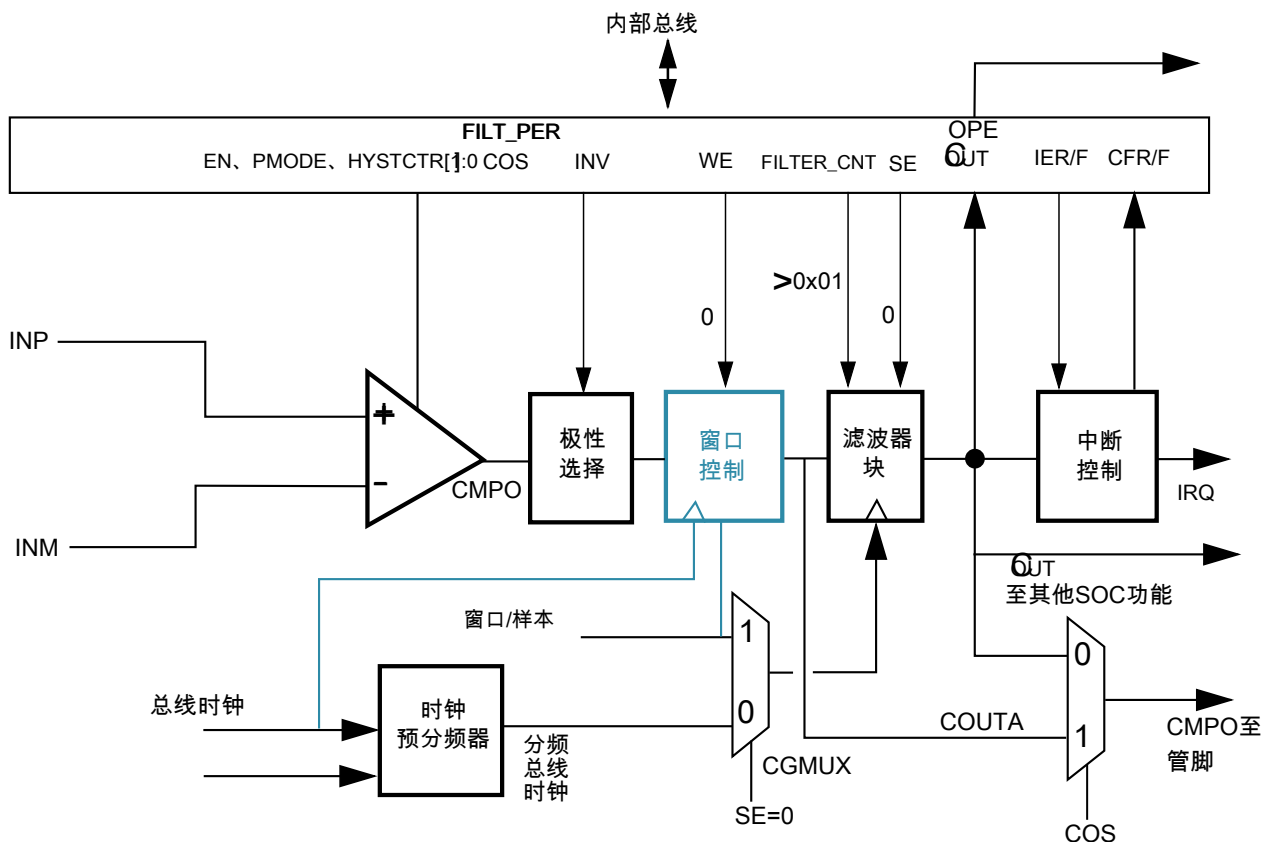


图 34-19. 已采样、已滤波(# 4B): 采样点从内部获得

“已采样、未滤波”(# 3B)和“已采样、已滤波(# 4B)”模式的唯一区别在于 $CR0[FILTER_CNT] > 1$ ，这会激活滤波器操作。

34.4.1.5 窗口模式(#s 5A & 5B)

下图显示比较器在窗口模式下的操作，忽略模拟比较器的延迟、极性选择和窗口控制模块。它还假定极性选择设置为正相状态。

注

仅当 WINDOW 信号为高电平时，模拟比较器输出才传送到 COUA。

在实际操作中，COUTA 可能滞后于模拟输入，滞后时间最长为一个总线时钟周期加上通过比较器和极性选择逻辑的组合路径延迟。

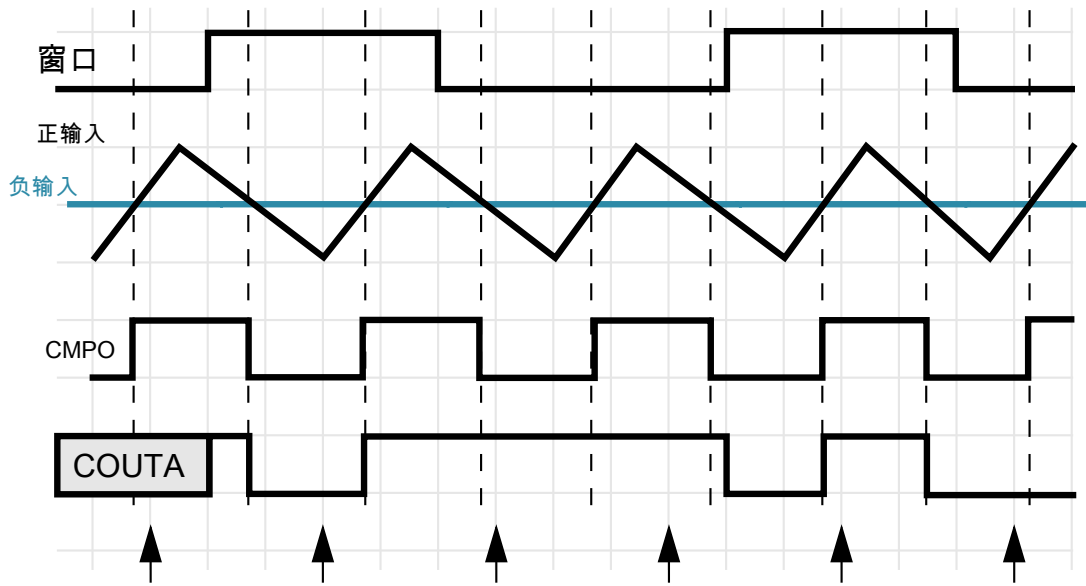


图 34-20. 窗口模式操作

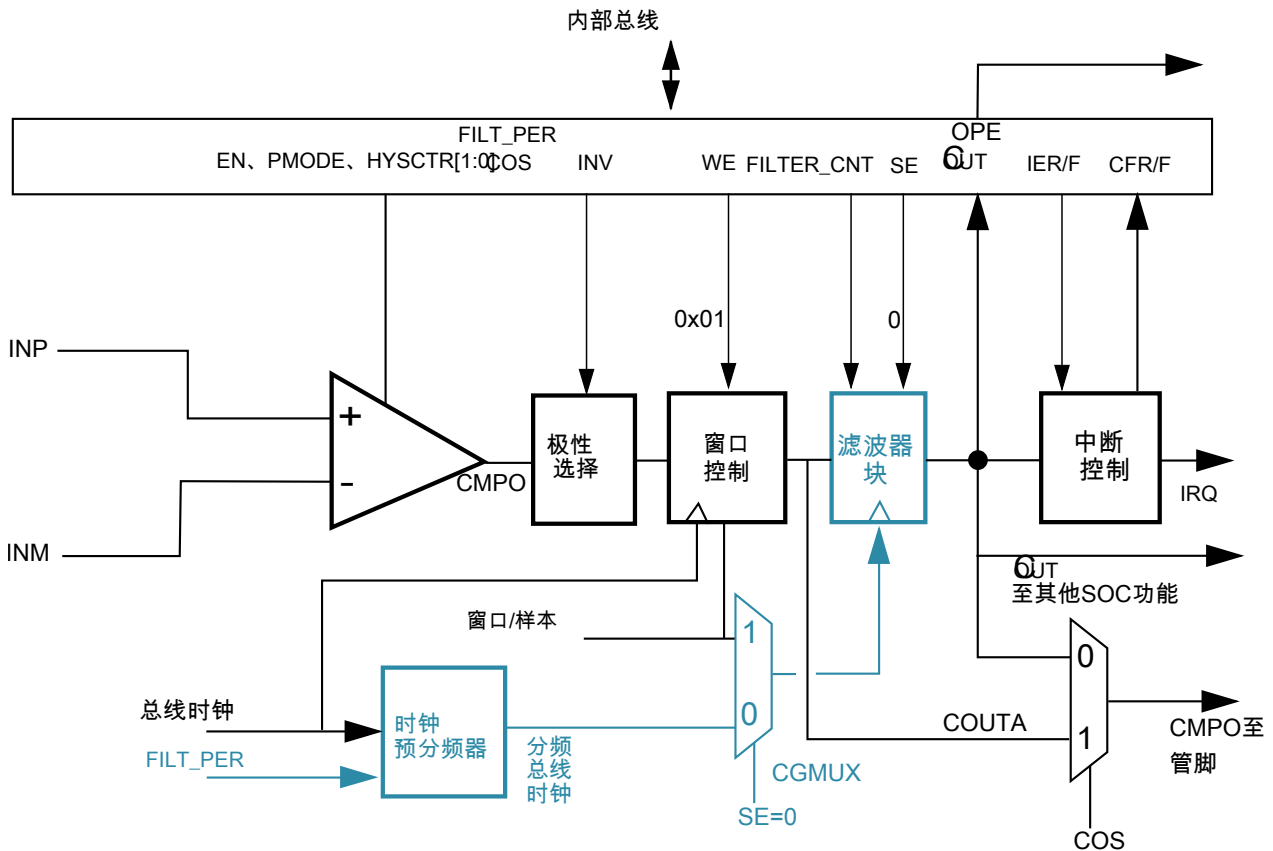


图 34-21. 窗口模式

有关导致滤波器模块禁用的控制配置，请参见[滤波器模块旁路逻辑](#)图。

如果任何窗口模式有效，则每当 $WINDOW = 1$ 时， $COUTA$ 将由总线时钟计时。当 $WINDOW = 0$ 时，将保持上一个锁存值。

34.4.1.6 窗口/重新采样模式(# 6)

下图所用输入波形与图 34-20 所示相同，并增加了对 COUTA 的重新采样以生成 COUT。采样发生于图中箭头所示时间点。同样，出于表达清楚考虑，图中忽略了传播延迟。

生成此例纯粹是为了展示比较器在加窗/重新采样模式下的工作原理，并不反应任何具体的应用。根据采样率和窗口位置，COUT 可能看不到模拟比较器检测到的过零事件。对于给定应用，必须慎重考虑采样周期"与"或者"或"窗口位置。

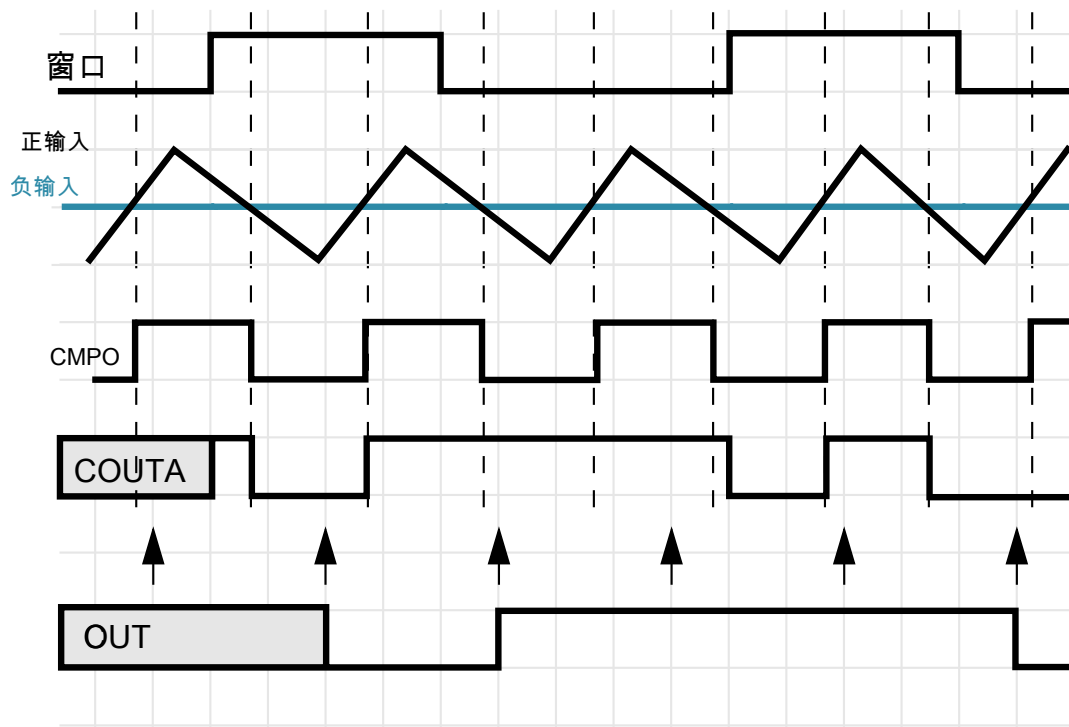


图 34-22. 加窗/重新采样工作模式

此工作模式会导致一连串未经过滤的比较器样本，其中，样本之间的间隔由 $FPR[FILT_PER]$ 和总线时钟速率决定。此模式的配置实际上与下一节所示加窗/过滤模式相同。唯一差异在于， $CR0[FILTER_CNT]$ 的值必须为 1。

34.4.1.7 加窗/过滤模式(#7)

这是比较器模块最复杂的工作模式，因为要同时用到加窗和滤波特性。另外，此模式是所有模式中延迟最高的模式。其近似值为：在窗口功能下最多 1 个总线时钟同步 + $((CR0[FILTER_CNT] * FPR[FILT_PER]) + 1) * \text{滤波器功能的总线时钟}$ 。

如果任何加窗模式有效，则每当 WINDOW = 1 时，COUTA 将由总线时钟计时。当 WINDOW = 0 时，将保持上一个锁存值。

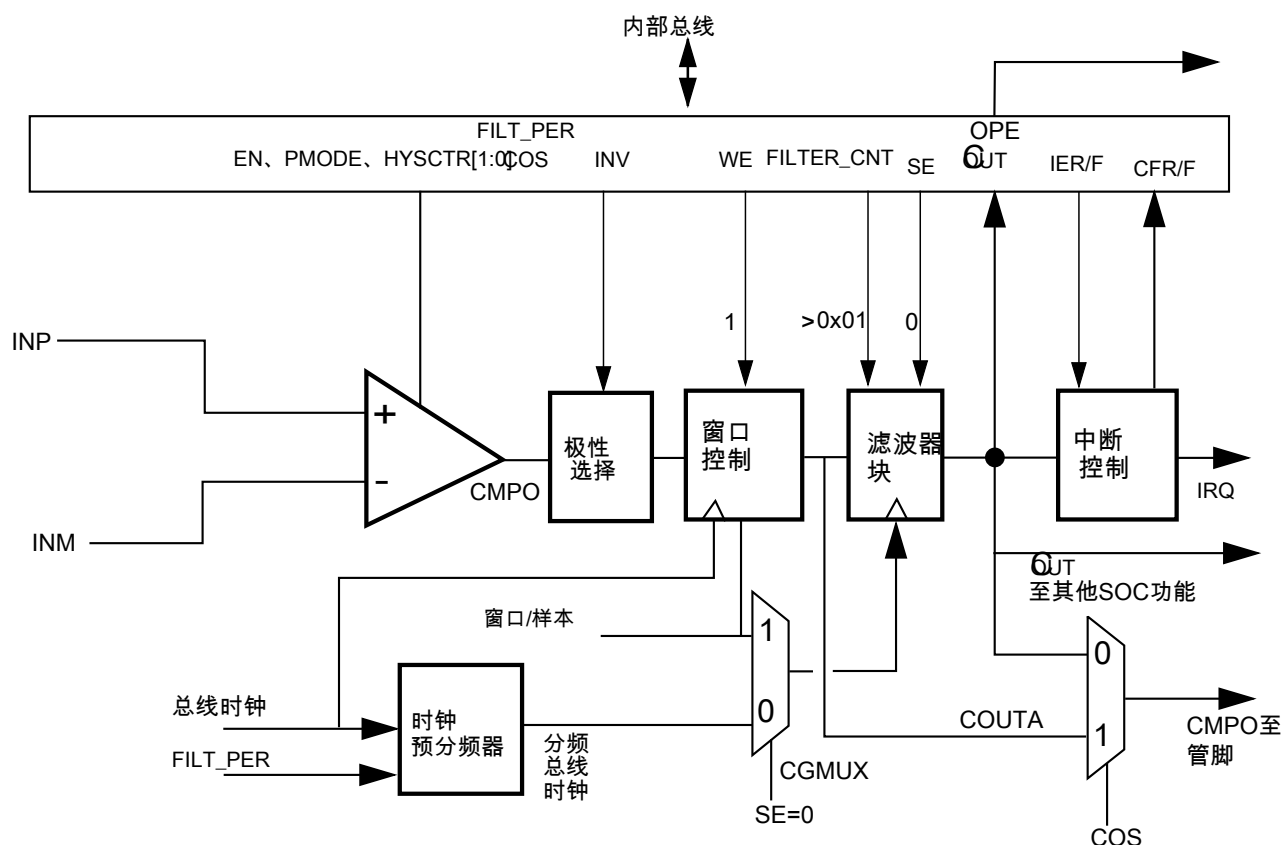


图 34-23. 加窗/过滤模式

34.4.2 电源模式

34.4.2.1 等待模式下的操作

在等待和 VLPW 模式下，CMP (若使能) 将继续正常工作，CMP 中断可以唤醒 MCU。

34.4.2.2 停止模式操作

根据与 MCU 内核或内核外设相关的时钟限制，当发生比较事件并且对应的中断使能时，MCU 将退出停止模式。类似地，如果 CR1[OPE] 被使能，比较器输出将按正常工作模式运行，比较器在外部引脚输出。在停止模式下，比较器可以运行于以下两种模式：

- 当 CR1[PMODE] = 1 时，高速(HS)比较模式
- 当 CR1[PMODE] = 0 时，低速(LS)比较模式

建议使用 LS 模式以尽量降低功耗。

如果通过复位退出停止模式，所有比较器寄存器将进入其复位状态。

34.4.2.3 低漏电模式操作

当芯片处于低漏电模式时：

- CMP 模块部分可以正常工作，但仅限于低速模式，且不受 CR1[PMODE]设置的影响
- 不支持窗口化、采样和过滤模式
- CMP 输出引脚被锁存，不反映比较输出状态。

正、负输入电压可通过外部引脚或 DAC 输出提供。如果发生比较事件并且使能 CMP 中断，可以使 MCU 结束低漏电模式。从低漏电模式唤醒后，CMP 模块将处于复位状态 (SCR[CFF]和 SCR[CFR]除外)。

34.4.3 启动和运行

这里列出一种典型的启动序列。

- 使 COUT 稳定所需要的时间为比较器的加电延迟加上从所选模拟源通过模拟比较器、加窗功能和滤波器的最大传播延迟。有关比较器加电延迟的信息，请参见数据手册。加窗功能的最大延迟为一个总线时钟周期。滤波器延迟见[低通滤波器](#)。
- 工作时，必须始终考虑所选数据路径的传播延迟。COUT 和 SCR[CFR]/SCR[CFF]要反映出输入变化或者数据路径中任一元件的配置变化，可能需要多个总线时钟周期。
- 当配置为过滤模式时，COUT 的初始值将等于 0，直到经过足够的时钟周期、填满滤波器的所有级为止。即使 COUTA 处于逻辑 1 状态，结果也会发生这种情况。

34.4.4 低通滤波器

低通滤波器作用于未过滤、未同步且可选择性反相的比较器输出 COUTA，然后生成经过滤的同步输出 COUT。

COUTA 和 COUT 均可配置为模块输出，在系统中用于不同的目的。

同步和边沿检测始终用于确定状态寄存器位值。它们同样适用于所有采样和加窗模式下的 COUT。可使用 FPR[FILT_PER]定义的内部时基实施滤波，也可使用外部采样输入来确定采样时间。

数字滤波的必要性以及滤波量取决于用户需求。在无外部迟滞电路的情况下，滤波可以起到更大的作用。若无外部迟滞，当所选 INM 和 INP 输入电压之差小于差分比较器的偏移电压时，可在 COUTA 生成高频振荡器。

34.4.4.1 使能滤波器模式

滤波器模式可通过如下方式使能：

- 设置 CR0[FILTER_CNT] > 0x01
- 并且设置 FPR[FILT_PER]为非零值或设置 CR1[SE]=1

若利用分频总线时钟驱动滤波器，它将在每个 FPR[FILT_PER]总线时钟周期对 COUTA 进行采样。

初始化时，滤波器输出为逻辑 0；随后，当所有相继 CR0[FILTER_CNT]样本都指示输出值已改变时，滤波器输出就会改变。换言之，SCR[COUT]在初始期间为 0，即便此时 COUTA 为逻辑 1。

CR1[SE]和 FPR[FILT_PER]均设置为 0 时，滤波器禁用，滤波过程相关的开关电流也就不存在。

注

更改滤波器参数之前，务必切换到此设置。这将把滤波器复位到已知状态。如果随意切换 CR0[FILTER_CNT]而不经此中间步骤，将会引起异常。

若 CR1[SE]=1，则滤波器在样本输入的每个正转换事件时对 COUTA 进行采样。当所有相继 CR0[FILTER_CNT]样本指示输出值已改变时，滤波器的输出状态就会改变。

34.4.4.2 延迟问题

FPR[FILT_PER]或 SAMPLE 周期值应设置为使采样周期比预期噪声周期稍长。这样一个噪声尖峰只会影响一次采样。选择的 CR0[FILTER_CNT]值必须降低噪声采样导致识别错误转换的可能性。错误转换的可能性定义为错误采样提升到 CR0[FILTER_CNT]功率的可能性。

FPR[FILT_PER]或 SAMPLE 周期的值和 CR0[FILTER_CNT]还必须根据所需的识别比较器输出实际转换最小延迟来调整。在标称延迟内检测到实际输出变化的可能性为正确采样提升到 CR0[FILTER_CNT]功率的可能性。

下表总结了各种工作模式在无噪声情况下的最大延迟值。每次实际输出转换被噪声屏蔽时，滤波延迟就会重新开始。

表 34-17. 比较器采样/滤波器最大延迟

模式#	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	操作	最大延迟 ¹
1	0	X	X	X	X	禁用	N/A
2A	1	0	0	0x00	X	连续模式	T _{PD}
2B	1	0	0	X	0x00		
3A	1	0	1	0x01	X	已采样、未滤波模式	T _{PD} + T _{SAMPLE} + T _{per}
3B	1	0	0	0x01	> 0x00		T _{PD} + (FPR[FILT_PER] * T _{per}) + T _{per}
4A	1	0	1	> 0x01	X	已采样、已滤波模式	T _{PD} + (CR0[FILTER_CNT] * T _{SAMPLE}) + T _{per}
4B	1	0	0	> 0x01	> 0x00		T _{PD} + (CR0[FILTER_CNT] * FPR[FILT_PER] x T _{per}) + T _{per}
5A	1	1	0	0x00	X	窗口模式	T _{PD} + T _{per}
5B	1	1	0	X	0x00		T _{PD} + T _{per}
6	1	1	0	0x01	0x01 - 0xFF	窗口/再采样模式	T _{PD} + (FPR[FILT_PER] * T _{per}) + 2T _{per}
7	1	1	0	> 0x01	0x01 - 0xFF	窗口/滤波模式	T _{PD} + (CR0[FILTER_CNT] * FPR[FILT_PER] x T _{per}) + 2T _{per}

1. T_{PD} 表示模拟元件和极性选择逻辑的内在延迟。T_{SAMPLE} 是外部采样时钟的时钟周期。T_{per} 是总线时钟周期。

34.5 CMP 中断

CMP 模块可以在比较器输出的上升沿"与"或者"或"下降沿产生中断。

下表列出了中断请求置位和解除置位的条件。

当	结果
SCR[IER]和 SCR[CFR]置位	中断请求置位
SCR[IEF]和 SCR[CFF]置位	中断请求置位
对于上升沿中断，SCR[IER]和 SCR[CFR]清零	中断请求解除置位
对于下降沿中断，SCR[IEF]和 SCR[CFF]清零	中断请求解除置位

34.6 DMA 支持

一般情况下，若 COUT 发生变化，CMP 会产生一个 CPU 中断。通过设置 SCR[DMAEN]使能 DMA 支持且通过设置 SCR[IER]"与"或者"或"SCR[IEF]使能中断时，COUT 上的对应变化将强制产生一个 DMA 传输请求，而不是 CPU 中断。当 DMA 传输完成时，它会发送一个传输完成指示，以解除置位 DMA 传输请求并清除标志，从而允许比较器输出随后发生变化并强制产生另一个 DMA 请求。

在 STOP 模式下，比较器仍可工作。

通过设置 SCR[DMAEN]使能 DMA 支持且通过设置 SCR[IER]"与"或者"或"SCR[IEF]使能中断时，COUT 上的对应变化将强制产生一个 DMA 传输请求，以便将系统从 STOP 模式唤醒。数据传输完成后，系统回到 STOP 模式。关于异步 DMA 功能的详情，请参阅《器件参考手册》中的 DMA 章节。

34.7 CMP 异步 DMA 支持

在 STOP 模式下，比较器仍然工作。

通过设置 SCR[DMAEN]使能 DMA 支持且通过设置 SCR[IER]"与"或者"或"SCR[IEF]使能中断时，COUT 上的对应变化将强制产生一个 DMA 传输请求，以便将系统从 STOP 模式唤醒。数据传输完成后，系统回到 STOP 模式。关于异步 DMA 功能的详情，请参阅《器件参考手册》中的 DMA 章节。

34.8 数模转换器

本图是 DAC 模块的结构框图。

它包含一个 64 抽头梯形电阻网络和一个 64:1 多路复用器，可从 DACO 输出的 64 个不同电平中选择一个作为输出电压。它通过 DAC 控制寄存器(DACCCR)进行控制。其参考电压源可从 V_{in1} 和 V_{in2} 这两个来源选择。不使用时，可以关断或禁用该模块。在禁用模式下，DACO 连接到模拟地。

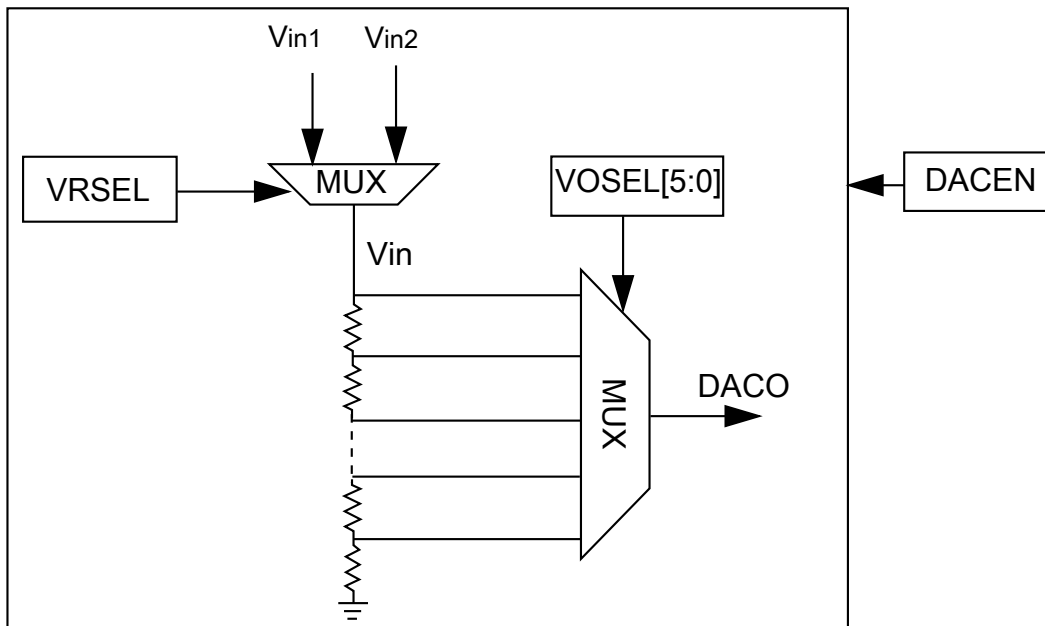


图 34-24. 6 比特 DAC 结构框图

34.9 DAC 功能说明

本部分提供 DAC 功能说明信息。

34.9.1 参考电压源选择

- V_{in1} 连接到主电压源，作为 64 抽头梯形电阻的参考电压源
- V_{in2} 连接到备用电压源

34.10 DAC 复位

该模块有一路复位输入，对应于芯片范围的外设复位。

34.11 DAC 时钟

本模块有一路时钟输入，即总线时钟。

34.12 DAC 中断

本模块没有中断。

第 35 章

12 位数模转换器 (DAC)

35.1 此模块的芯片实现细节

35.1.1 12 位 DAC 概述

本器件包含一个 12 位数字-模拟转换器 (DAC) (具备可编程参考发生器输出)。DAC 包含一个支持 DMA 的 FIFO。

35.1.2 12 位 DAC 输出

DAC 输出可置于外部引脚上或设置为模拟比较器或 ADC 的其中一个输入。

35.1.3 12 位 DAC 参考

对于此器件，只有 VDDA 可用作 DAC 参考。DACx_C0[DACRFS]控制位对此没有影响。

35.2 简介

该 12 位数模转换器 (DAC) 为低功耗通用 DAC。DAC 输出可置于外部引脚上或设置为模拟比较器、运算放大器或 ADC 的其中一个输入。

35.3 特性

DAC 模块特性包括：

- 片上可编程参考生成器输出。电压输出范围为 $1/4096V_{in}$ 至 V_{in} ，步进为 $1/4096V_{in}$ （其中 V_{in} 为输入电压）。
- V_{in} 可从两个参考电压源选择
- 在正常停止模式保持静态
- 可配置水印和多种工作模式支持 16 字长的数据缓冲器
- DMA 支持

35.4 结构框图

DAC 模块的结构框图如下：

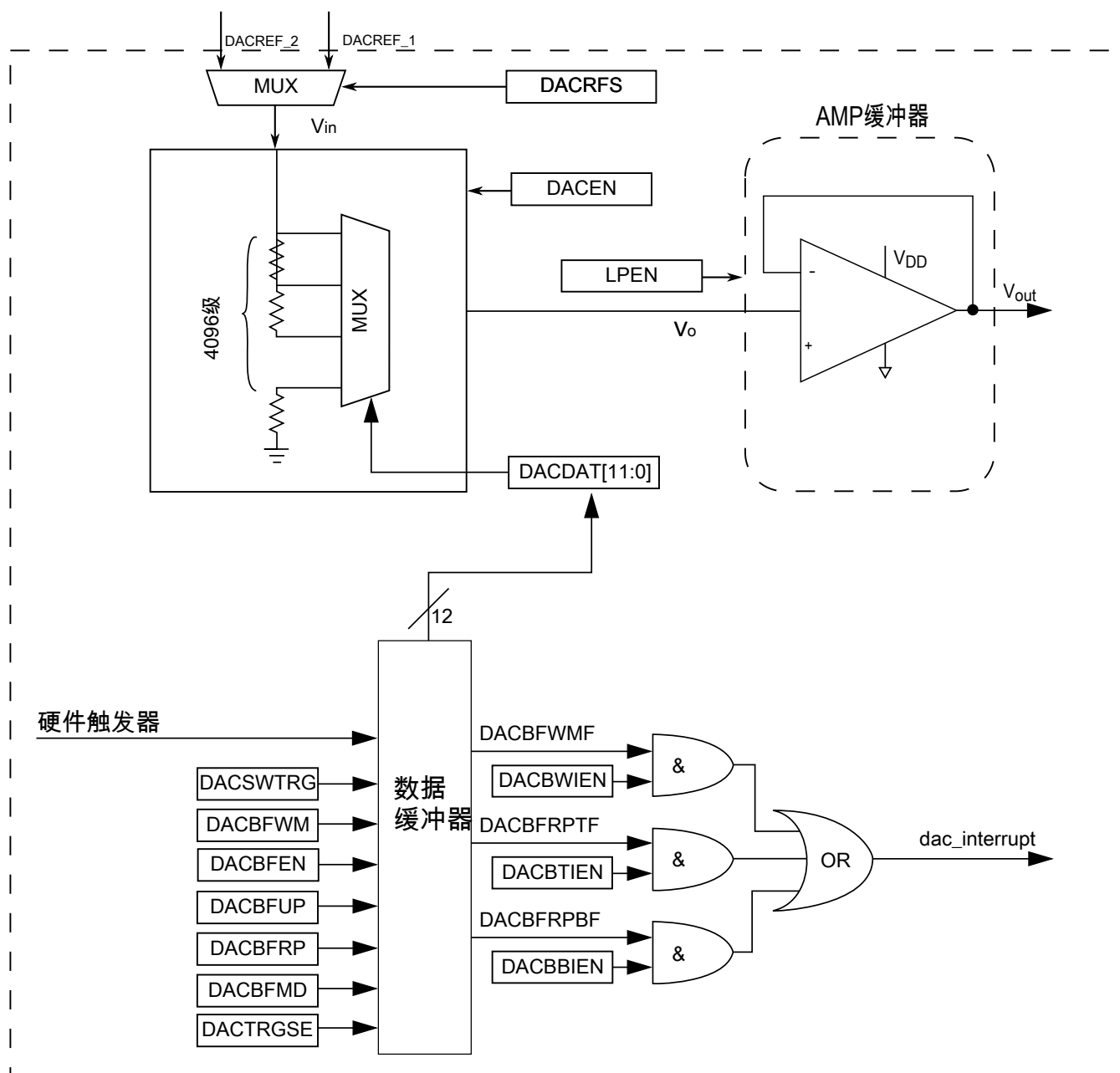


图 35-1. DAC 结构框图

35.5 存储器映射/寄存器定义

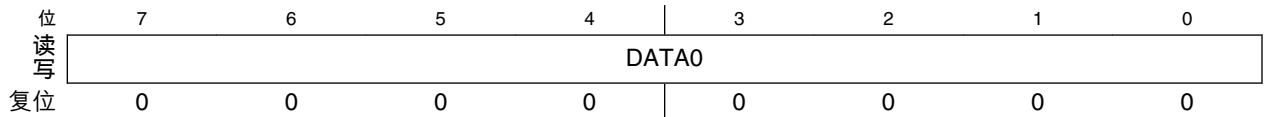
DAC 用寄存器来控制模拟比较器和可编程分压器，以执行数模转换功能。

DAC 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4003_F000	DAC 数据低电平寄存器 (DAC0_DAT0L)	8	R/W	00h	35.5.1/697
4003_F001	DAC 数据高电平寄存器 (DAC0_DAT0H)	8	R/W	00h	35.5.2/697
4003_F002	DAC 数据低电平寄存器 (DAC0_DAT1L)	8	R/W	00h	35.5.1/697
4003_F003	DAC 数据高电平寄存器 (DAC0_DAT1H)	8	R/W	00h	35.5.2/697
4003_F004	DAC 数据低电平寄存器 (DAC0_DAT2L)	8	R/W	00h	35.5.1/697
4003_F005	DAC 数据高电平寄存器 (DAC0_DAT2H)	8	R/W	00h	35.5.2/697
4003_F006	DAC 数据低电平寄存器 (DAC0_DAT3L)	8	R/W	00h	35.5.1/697
4003_F007	DAC 数据高电平寄存器 (DAC0_DAT3H)	8	R/W	00h	35.5.2/697
4003_F008	DAC 数据低电平寄存器 (DAC0_DAT4L)	8	R/W	00h	35.5.1/697
4003_F009	DAC 数据高电平寄存器 (DAC0_DAT4H)	8	R/W	00h	35.5.2/697
4003_F00A	DAC 数据低电平寄存器 (DAC0_DAT5L)	8	R/W	00h	35.5.1/697
4003_F00B	DAC 数据高电平寄存器 (DAC0_DAT5H)	8	R/W	00h	35.5.2/697
4003_F00C	DAC 数据低电平寄存器 (DAC0_DAT6L)	8	R/W	00h	35.5.1/697
4003_F00D	DAC 数据高电平寄存器 (DAC0_DAT6H)	8	R/W	00h	35.5.2/697
4003_F00E	DAC 数据低电平寄存器 (DAC0_DAT7L)	8	R/W	00h	35.5.1/697
4003_F00F	DAC 数据高电平寄存器 (DAC0_DAT7H)	8	R/W	00h	35.5.2/697
4003_F010	DAC 数据低电平寄存器 (DAC0_DAT8L)	8	R/W	00h	35.5.1/697
4003_F011	DAC 数据高电平寄存器 (DAC0_DAT8H)	8	R/W	00h	35.5.2/697
4003_F012	DAC 数据低电平寄存器 (DAC0_DAT9L)	8	R/W	00h	35.5.1/697
4003_F013	DAC 数据高电平寄存器 (DAC0_DAT9H)	8	R/W	00h	35.5.2/697
4003_F014	DAC 数据低电平寄存器 (DAC0_DAT10L)	8	R/W	00h	35.5.1/697
4003_F015	DAC 数据高电平寄存器 (DAC0_DAT10H)	8	R/W	00h	35.5.2/697
4003_F016	DAC 数据低电平寄存器 (DAC0_DAT11L)	8	R/W	00h	35.5.1/697
4003_F017	DAC 数据高电平寄存器 (DAC0_DAT11H)	8	R/W	00h	35.5.2/697
4003_F018	DAC 数据低电平寄存器 (DAC0_DAT12L)	8	R/W	00h	35.5.1/697
4003_F019	DAC 数据高电平寄存器 (DAC0_DAT12H)	8	R/W	00h	35.5.2/697
4003_F01A	DAC 数据低电平寄存器 (DAC0_DAT13L)	8	R/W	00h	35.5.1/697
4003_F01B	DAC 数据高电平寄存器 (DAC0_DAT13H)	8	R/W	00h	35.5.2/697
4003_F01C	DAC 数据低电平寄存器 (DAC0_DAT14L)	8	R/W	00h	35.5.1/697
4003_F01D	DAC 数据高电平寄存器 (DAC0_DAT14H)	8	R/W	00h	35.5.2/697
4003_F01E	DAC 数据低电平寄存器 (DAC0_DAT15L)	8	R/W	00h	35.5.1/697
4003_F01F	DAC 数据高电平寄存器 (DAC0_DAT15H)	8	R/W	00h	35.5.2/697
4003_F020	DAC 状态寄存器 (DAC0_SR)	8	R/W	02h	35.5.3/698
4003_F021	DAC 控制寄存器 (DAC0_C0)	8	R/W	00h	35.5.4/699
4003_F022	DAC 控制寄存器 1 (DAC0_C1)	8	R/W	00h	35.5.5/700
4003_F023	DAC 控制寄存器 2 (DAC0_C2)	8	R/W	0Fh	35.5.6/700

35.5.1 DAC 数据低电平寄存器 (DACx_DATnL)

地址: 4003_F000h 基准 + 0h 偏移 + (2d × i), 其中 i=0d 到 15d

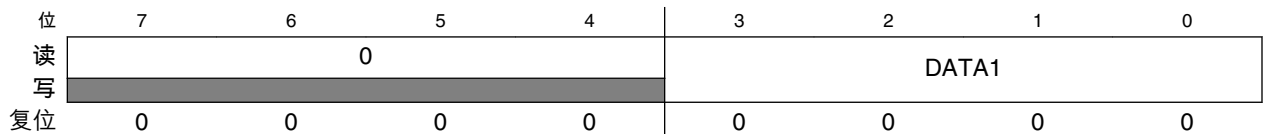


DACx_DATnL 字段描述

字段	描述
DATA0	DATA0 当 DAC 缓冲区未使能时, DATA[11:0]根据以下公式控制输出电压: $V_{out} = V_{in} * (1 + DACDAT0[11:0])/4096$ 当 DAC 缓冲区使能时, DATA 映射到 16 字缓冲区。

35.5.2 DAC 数据高电平寄存器 (DACx_DATnH)

地址: 4003_F000h 基准 + 1h 偏移 + (2d × i), 其中 i=0d 到 15d



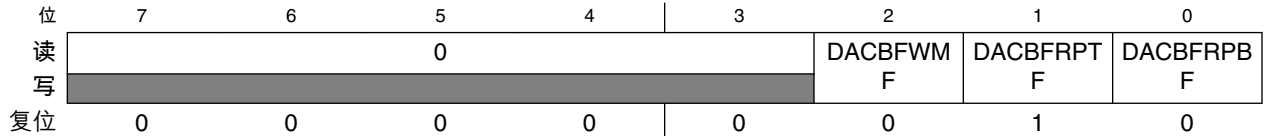
DACx_DATnH 字段描述

字段	描述
7-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
DATA1	DATA1 当 DAC 缓冲区未使能时, DATA[11:0]根据以下公式控制输出电压: $V_{out} = V_{in} * (1 + DACDAT0[11:0])/4096$ 当 DAC 缓冲区使能时, DATA[11:0]映射到 16 字缓冲区。

35.5.3 DAC 状态寄存器 (DACx_SR)

若 DMA 已使能，当 DMA 请求完成时，DMA 可自动清除状态标志。向一个字段写入 0 可将其清空，写入 1 则没有影响。复位后，DACBFRPTF 置位，需要时可通过软件将其清零。仅当数据缓冲区状态改变时，标志才会置位。

地址: 4003_F000h 基准 + 20h 偏移 = 4003_F020h



DACx_SR 字段描述

字段	描述
7-3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 DACBFWMF	DAC 缓冲区水印标志 如果剩余 FIFO 数据少于水印设置 ,此位就会置位。DMA 或 CPU 将数据写入 FIFO 时 ,它便自动清零。FIFO 模式下会忽略对此位的写操作。 0 DAC 缓冲区读取指针未到达水印位置。 1 DAC 缓冲区读取指针已到达水印位置。
1 DACBFRPTF	DAC 缓冲区读取指针顶部位置标志 在 FIFO 模式下，它是 FIFO 接近空无标志。当 FIFO 中仅有一个数据时，它置位。若设置此位以避免 DAC 输出可能发生毛刺或突然变化，则任何 DAC 触发都不会增加读取指针。若 FIFO 不为空，它会自动清零。 0 DAC 缓冲区读取指针不等于 0。 1 DAC 缓冲区读取指针等于 0。
0 DACBFRPBF	DAC 缓冲区读取指针底部位置标志 在 FIFO 模式下，它是 FIFO FULL 状态位。它意味着由于 FIFO 写入指针增加，其读取指针等于写入指针。如果该位置位，则 DAC 会忽略 DMA 或 CPU 对 FIFO 的写操作。若有 DAC 触发事件使 DAC 读取指针增加，它就会清零。FIFO 模式下会忽略对此位的写操作。 0 DAC 缓冲区读取指针不等于 C2[DACBFUP]。 1 DAC 缓冲区读取指针等于 C2[DACBFUP]。

35.5.4 DAC 控制寄存器 (DACx_C0)

地址: 4003_F000h 基准 + 21h 偏移 = 4003_F021h

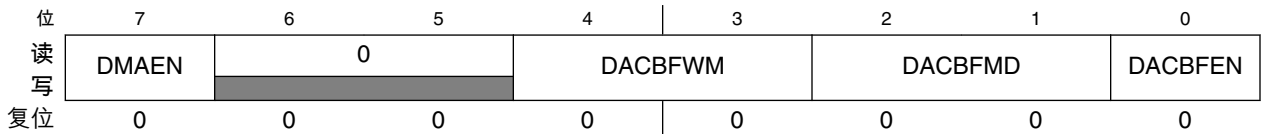
位	7	6	5	4	3	2	1	0
读	DACEN	DACRFS	DACTRGSEL	0	LPEN	DACBWIEN	DACBTIEN	DACBBIEN
写				DACSWTRG				
复位	0	0	0	0	0	0	0	0

DACx_C0 字段描述

字段	描述
7 DACEN	<p>DAC 使能</p> <p>启动可编程基准发生器操作。</p> <p>0 禁用 DAC 系统。 1 使能 DAC 系统。</p>
6 DACRFS	<p>DAC 基准选择</p> <p>0 DAC 选择 DACREF_1 作为基准电压。 1 DAC 选择 DACREF_2 作为基准电压。</p>
5 DACTRGSEL	<p>DAC 触发选择</p> <p>0 选择 DAC 硬件触发。 1 选择 DAC 软件触发。</p>
4 DACSCTR	<p>DAC 软件触发</p> <p>高电平有效。这是一个只写字段，读出值始终为 0。若选择 DAC 软件触发且使能缓冲区，则向此字段写入 1 将使缓冲区读取指针前进一次。</p> <p>0 DAC 软件触发无效。 1 DAC 软件触发有效。</p>
3 LPEN	<p>DAC 低功耗控制</p> <p>注：有关以下模式的影响的详细信息，请参见器件数据手册的 12 比特 DAC 电气特性部分。</p> <p>0 高功耗模式 1 低功耗模式</p>
2 DACBWIEN	<p>DAC 缓冲区水印中断使能</p> <p>0 禁用 DAC 缓冲区水印中断。 1 使能 DAC 缓冲区水印中断。</p>
1 DACBTIEN	<p>DAC 缓冲区读取指针顶部标志中断使能</p> <p>0 禁用 DAC 缓冲区读取指针顶部标志中断。 1 使能 DAC 缓冲区读取指针顶部标志中断。</p>
0 DACBBIEN	<p>DAC 缓冲区读取指针底部标志中断使能</p> <p>0 禁用 DAC 缓冲区读取指针底部标志中断。 1 使能 DAC 缓冲区读取指针底部标志中断。</p>

35.5.5 DAC 控制寄存器 1 (DACx_C1)

地址: 4003_F000h 基准 + 22h 偏移 = 4003_F022h

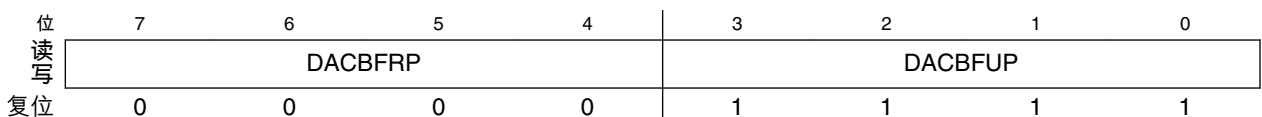


DACx_C1 字段描述

字段	描述
7 DMAEN	<p>DMA 使能选择</p> <p>0 DMA 已禁用。</p> <p>1 DMA 已使能。当 DMA 使能时，原始中断会产生 DMA 请求。这些中断不会同时出现在该模块上。</p>
6-5 保留	<p>此字段为保留字段。</p> <p>此只读字段为保留字段且值始终为 0。</p>
4-3 DACBFWM	<p>DAC 缓冲区水印选择</p> <p>在正常模式下，它控制 SR[DACBFWMF]何时置位。当 DAC 缓冲区读取指针达到此字段定义的字时，即离上限(DACBUP) 1-4 字时，SR[DACBFWMF]就会置位。它允许用户配置水印中断。在 FIFO 模式下，它是 FIFO 水印选择字段。</p> <p>00 在正常模式下，1 字。在 FIFO 模式下，FIFO 中剩余 2 个或少于 2 个数据时，水印状态位置位。</p> <p>01 在正常模式下，2 字。在 FIFO 模式下，FIFO 中剩余 Max/4 个或少于 Max/4 个数据时，水印状态位置位。</p> <p>10 在正常模式下，3 字。在 FIFO 模式下，FIFO 中剩余 Max/2 个或少于 Max/2 个数据时，水印状态位置位。</p> <p>11 在正常模式下，4 字。在 FIFO 模式下，FIFO 中剩余 Max-2 个或少于 Max-2 个数据时，水印状态位置位。</p>
2-1 DACBFMD	<p>DAC 缓冲区工作模式选择</p> <p>00 正常模式</p> <p>01 摇摆模式</p> <p>10 一次性扫描模式</p> <p>11 FIFO 模式</p>
0 DACBFEN	<p>DAC 缓冲区使能</p> <p>0 禁用缓冲区读取指针。转换的数据始终是缓冲区的第一个字。</p> <p>1 使能缓冲区读取指针。转换的数据是读取指针指向的字。这意味着转换的数据可以是缓冲区中的任何字。</p>

35.5.6 DAC 控制寄存器 2 (DACx_C2)

地址: 4003_F000h 基准 + 23h 偏移 = 4003_F023h



DACx_C2 字段描述

字段	描述
7-4 DACBFRP	DAC 缓冲区读取指针 在正常模式下，它保存缓冲区读取指针的当前值。在 FIFO 模式下，它是 FIFO 读取指针。在 FIFO 模式下，它是可写的。用户可将其配置为同一地址，以将 FIFO 复位为空状态。
DACBFUP	DAC 缓冲区上限 在正常模式下，它选择 DAC 缓冲区的上限。缓冲区读取指针不能超过它。在 FIFO 模式下，它是 FIFO 写入指针。在 FIFO 模式下，用户不能设置缓冲区上限。在正常模式下，其复位值为 MAX。当 IP 配置为 FIFO 模式时，此寄存器成为 Write Pointer，其初始值自动设置为等于 READ_POINTER，FIFO 状态为空。它是可写的，用户可将其配置为同一地址，以将 FIFO 复位为空状态。

35.6 功能说明

12 位 DAC 模块可以通过 C0 [DACRFS]，选择两个参考输入中的一个 (DACREF_1 和 DACREF_2)，将其作为 DAC 参考电压 V_{in} 。要确定 DACREF_1 和 DACREF_2 的源选项，请参阅芯片特性的 DAC 信息。

当 DAC 使能时，它会将 DACDAT0[11:0] 中的数据或者来自 DAC 数据缓冲器中的数据转换为步进模拟输出电压。输出电压范围为 V_{in} 至 $V_{in}/4096$ ，步长为 $V_{in}/4096$ 。

35.6.1 DAC 数据缓冲器操作

当 DAC 使能且缓冲器未使能时，DAC 模块始终将 DAT0 中的数据转换为模拟输出电压。

当 DAC 和缓冲器都使能时，DAC 模块将数据缓冲器中的数据转换为模拟输出电压。每当发生硬件或软件触发器事件时，数据缓冲器读指针会前进到下一个字。

数据缓冲器可以配置为正常模式、摇摆模式、一次性扫描模式或 FIFO 模式。当缓冲器从一种操作模式切换为另一种操作模式时，读指针保持不变。通过写入 C2[DACBFRP]，可将读指针设为 0 与 C2[DACBFUP] 之间的任意值。

35.6.1.1 DAC 数据缓冲器中断

可以为 DAC 缓冲器配置多种中断和相关标志。SR[DACBFRPBF] 在 DAC 缓冲器读指针达到 DAC 缓冲器上限 (即 $C2[DACBFRP] = C2[DACBFUP]$) 时置位。SR[DACBFRPTF] 在 DAC 读指针等于起始位置 0 时置位。最后，SR[DACBFWMF] 在 DAC 缓冲器读指针达到 C1[DACBFWM] 定义的位置时置位。C1[DACBFWM] 可以用于在 DAC 缓冲器读指针距离 C2[DACBFUP]1 至 4 个字时生成一个中断。

35.6.1.2 DAC 数据缓冲器操作模式

下表描述了 DAC 模块数据缓冲器的不同操作模式。

表 35-79. DAC 数据缓冲器操作模式

模式	说明
缓冲器正常模式	该模式为默认模式。缓冲器工作时充当循环缓冲器。每次触发时，读取指针增加 1。读取指针达到上限时，它将直接进入下一个触发事件中的 0。
缓冲器摇摆模式	此模式与正常模式相似。然而，当读取指针到达上限时，它不会变为 0。它会在接下来的触发事件中递减 1，直至达到 0。
缓冲器一次性扫描模式	每次触发时，读取指针增加 1。达到上限时停止。如果读取指针复位至除上限外的地址，则它会增加至上限地址，然后再次停止。 注：如果软件将读取指针设置到上限，则在这种模式下，读取指针不会前进。
FIFO 模式	在 FIFO 模式，缓冲器用作 FIFO。对任何 DACDATx 执行有效写操作时，数据输入 FIFO，写入指针自动递增。该模块内部连接到一个 32 比特接口。对于任何 16 比特或 8 比特 FIFO 访问，地址比特[1]必须为 0，否则将忽略该写操作。对于任何 32 比特 FIFO 访问，Write_Pointer 必须为偶数，否则将忽略该写操作。 注：成功的 32 比特 FIFO 写操作会将写入指针增加 2。任何导致 FIFO 上溢的写操作都会被忽略，这些情况包括：1.FIFO 已满，忽略写操作；2.FIFO 接近填满(FIFO_SIZE-1)，忽略 32 比特写操作。 注：对于 8 比特写操作，地址比特[0]根据小端字节对齐顺序，决定将哪个字节通道写入 FIFO。只有同时写入两个字节通道，写指针才会增加。用户须确保 8 比特访问成对发生，同时写入上下两个字节。对哪个字节先写入没有要求。在 FIFO 模式下，对 DACDATx 的读取访问（从正常模式）没有变化，读取 DACDATx 将把访问地址寻址的数据返回到数据缓冲器，读取访问不会改变 FIFO 模式下的写入指针和读取指针。FIFO 写操作可以在第一次数据转换未使能 DAC 时发生。但是，为使 FIFO 模式有效，DACC1[DACBFEN]须使能缓冲器。 在 FIFO 模式，DATA BUF 用作 FIFO。

35.6.2 DMA 操作

使能 DMA 时，将产生 DMA 请求而不是中断请求。“DMA 完成”信号清除 DMA 请求。

状态寄存器标志仍然置位，当 DMA 完成时自动清零。

35.6.3 复位

复位期间，DAC 会按照默认模式配置并禁用。

35.6.4 低功耗工作模式

下表显示 DAC 模块的等待模式和停止模式。

表 35-80. 工作模式

工作模式	说明
等待模式	若使能，DAC 将正常工作。
停止模式	若使能，DAC 模块将继续以常规停止模式运行，输出电压将保持停止前的值。 在低功耗停止模式下，DAC 完全关断。

注

模块模式到内核模式的分配取决于芯片。有关模块到内核的模式分配，请参见描述如何配置模块的章节。

第 36 章 可编程延时模块 (PDB)

36.1 此模块的芯片实现细节

36.1.1 PDB 实例化

36.1.1.1 PDB 输出触发器

表 36-1. PDB 输出触发器

用于 ADC 触发器的 PDB 通道数	1
每个 PDB 通道的预触发器数	2
DAC 触发器数	1
脉冲输出数	1

36.1.1.2 PDB 输入触发器连接

表 36-2. PDB 输入触发器选项

PDB 触发器	PDB 输入
0000	外部触发器
0001	CMP 0
0010	保留
0011	保留
0100	PIT 通道 0 输出
0101	PIT 通道 1 输出
0110	PIT 通道 2 输出
0111	PIT 通道 3 输出
1000	TPM0 溢出

下一页继续介绍此表...

表 36-2. PDB 输入触发器选项 (继续)

PDB 触发器	PDB 输入
1001	TPM1 溢出
1010	TPM2 溢出
1011	保留
1100	RTC 警报
1101	RTC 秒数
1110	LPTMR 输出
1111	软件触发

36.1.2 PDB 模块互连

PDB 触发器输出	连接
通道 0 触发器	ADC0 触发器
DAC 触发器	DAC0 触发器
脉冲输出	脉冲输出连到各 CMP 模块的采样/窗口输出以控制采样操作

36.1.3 背靠背应答连接

背靠背工作模式使一个 ADC 通道转换完成时，可以同时触发下一个 PDB 通道预触发器和触发器输出。

在此 MCU 中，PDB 背靠背操作应答连接按以下方式实现：

- PDB 通道 0 触发器/预触发器 0 应答输入：ADC0SC1B_COCO
- PDB 通道 0 触发器/预触发器 1 应答输入：ADC0SC1A_COCO

如此，背靠背链连接成一个环：

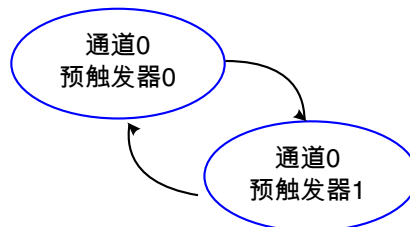


图 36-1. PDB 背靠背链

应用代码可以设置 PDBx_CHnC1[BB]位, 以将 PDB 预触发器配置为单个或多个链。

36.1.4 PDB 间隔触发器与 DAC 的连接

在本 MCU 中, PDB 间隔触发器与 DAC 的连接按如下方式实现。

- PDB 间隔触发器 0 连接 DAC0 硬件触发器输入。

36.1.5 DAC 外部触发器输入连接

在此 MCU 中, 以下 DAC 外部触发器输入。

- DAC 外部触发器输入 0: ADC0SC1A_COCO

36.1.6 脉冲输出连接

36.1.7 脉冲输出使能寄存器的实现

下表展示了脉冲输出使能寄存器在模块和芯片层次的比较。

表 36-3. PDB 脉冲输出使能寄存器

寄存器	模块实施	芯片实施
POnEN	7:0 - POEN 31:8 - 保留	

36.2 简介

可编程延迟区块 (PDB) 可为 ADC 的硬件触发输入提供来自内部或外部触发器的可控延迟或可编程间隔节拍, "与"或者"或"为 DAC 生成间隔触发器, 从而实现 ADC 转换之间的精确时序"与"或者"或"实现 DAC 更新。PDB 可选择性提供用作 CMP 模块中采样窗口的脉冲输出。

36.2.1 特性

- 最多 15 个触发器输入源和一个软件触发源
- 高达 8 个可配置 PDB 通道，用于 ADC 硬件触发器
 - 一个 PDB 通道与一个 ADC 关联
 - 每个 PDB 通道提供一个触发器输出用于 ADC 硬件触发器，多达 8 个预触发器输出用于选择 ADC 触发器
 - 可单独启用或禁用触发器输出
 - 一个 16 位延迟寄存器预触发器输出
 - 预触发器输出延迟寄存器的可选旁路
 - 在一次性或连续性模式下工作
 - 可选背靠背工作模式，此模式能够实现一个 ADC 通道转换完成触发下一个 PDB 通道
 - 一个可编程延迟中断
 - 一个序列错误中断
 - 每个预触发器一个通道标志和一个序列错误标志
 - 支持 DMA
- 高达 8 个脉冲输出
 - 可单独启用或禁用脉冲输出
 - 可编程脉冲宽度

注

PDB 输入和输出触发器的通道数因芯片而异。详情请参见芯片特性 PDB 信息。

36.2.2 实施方法

本节中，以下字母指输出触发器的数量：

- N—PDB 通道可用总数。
- n—PDB 通道编号，有效范围为 0 至 N-1。
- M—每个 PDB 通道的可用预触发器总数。

- m—预触发器编号，有效范围为 0 至 $M-1$ 。
- X—DAC 间隔触发器总数。
- x—DAC 间隔触发器输出编号，有效范围 0 至 $X-1$ 。
- Y—脉冲输出总数。
- y—脉冲输出编号，有效值范围为 0 至 $Y-1$ 。

注

模块到内核输出触发器数因芯片而异。有关模块到内核输出触发器的实施方法，请参见芯片配置信息。

36.2.3 背靠背应答连接

PDB 背靠背操作应答连接因芯片而异。有关实施方法，请参见芯片配置信息。

36.2.4 DAC 外部触发器输入连接

DAC 外部触发器输入的实施方式因芯片而异。详情请参见芯片配置信息。

36.2.5 功能框图

此图展示的是 PDB 的主要组件。

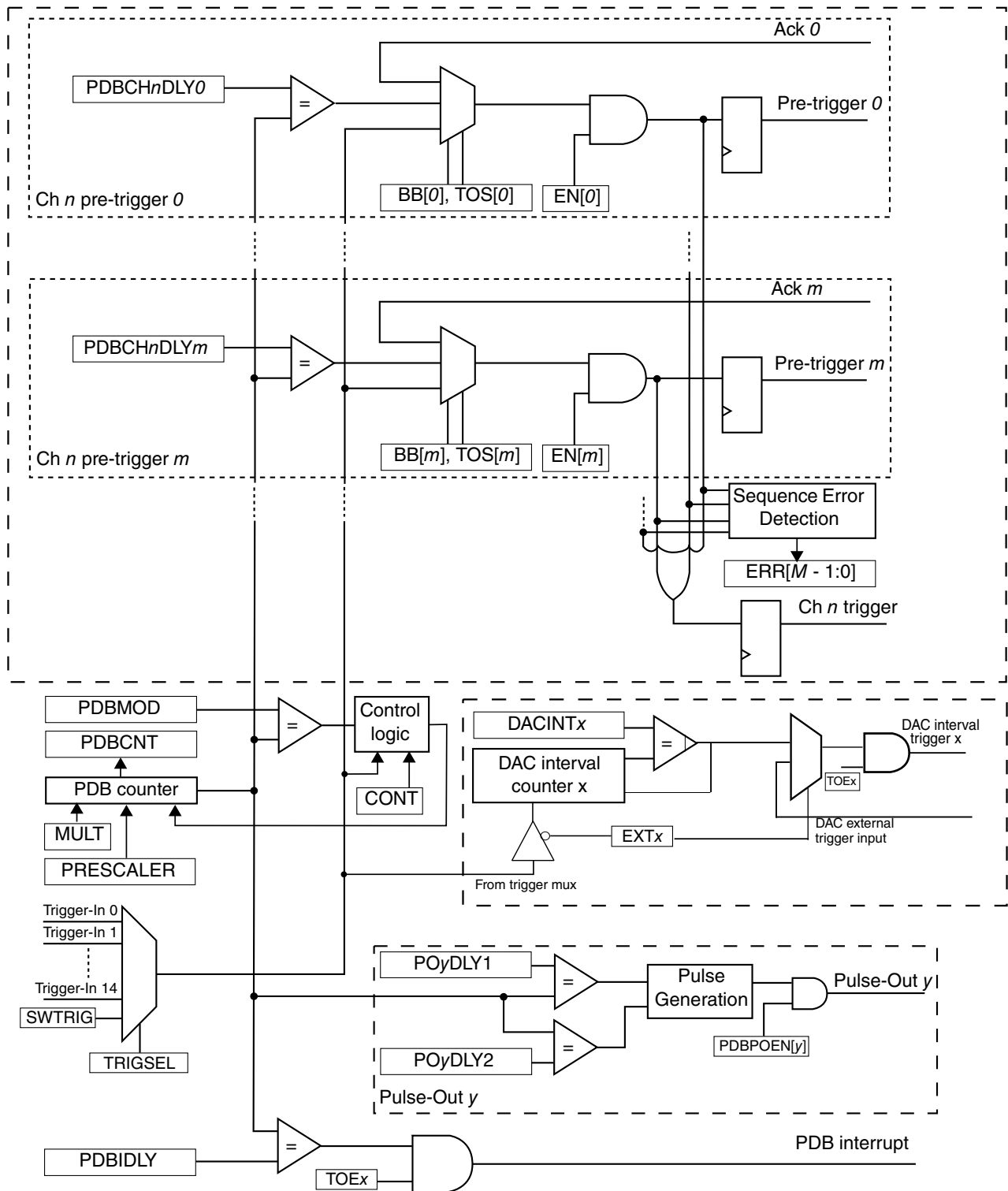


图 36-2. PDB 功能框图

在本图中，只显示了一个 PDB 通道 n 、一个 DAC 间隔触发器 x 和一个脉冲输出 y 。未显示 PDB 使能的控制逻辑和序列错误中断逻辑。

36.2.6 操作模式

PDB ADC 触发器有下列操作模式:

- 禁用—计数器关闭, 如果 PDB 未处于背靠背旁路操作模式, 则所有预触发器和触发器均输出低电平。
- 调试—当处理器处于调试模式时, 计数器暂停; DAC 触发器的计数器在调试模式下也会暂停。
- 单次使能—在所选触发器输入源上接收到上升沿时, 或者是软件触发模式 SC[SWTRIG]写入 1 时, 计数器使能, 并从零开始计数。在每个 PDB 通道中, 使能的预触发器将在每个触发器输入事件时置位一次。每当任一预触发器置位时, 触发器输出将置位。
- 连续使能—计数器使能, 并从零开始计数。当计数达到模块寄存器中指定的值时, 计数器将再次回到零重新计数。这样, 单次触发器输入事件即可导致预触发器/触发器连续输出。
- 旁路使能—预触发器和触发器输出将在所选触发器输入源上升沿立即置位, 或者选择软件触发器并向 SC[SWTRIG]写入 1; 这样就绕过了延迟寄存器。可以旁路任意一个或多个延迟寄存器; 因此, 此模式可以与单次或连续模式配合使用。

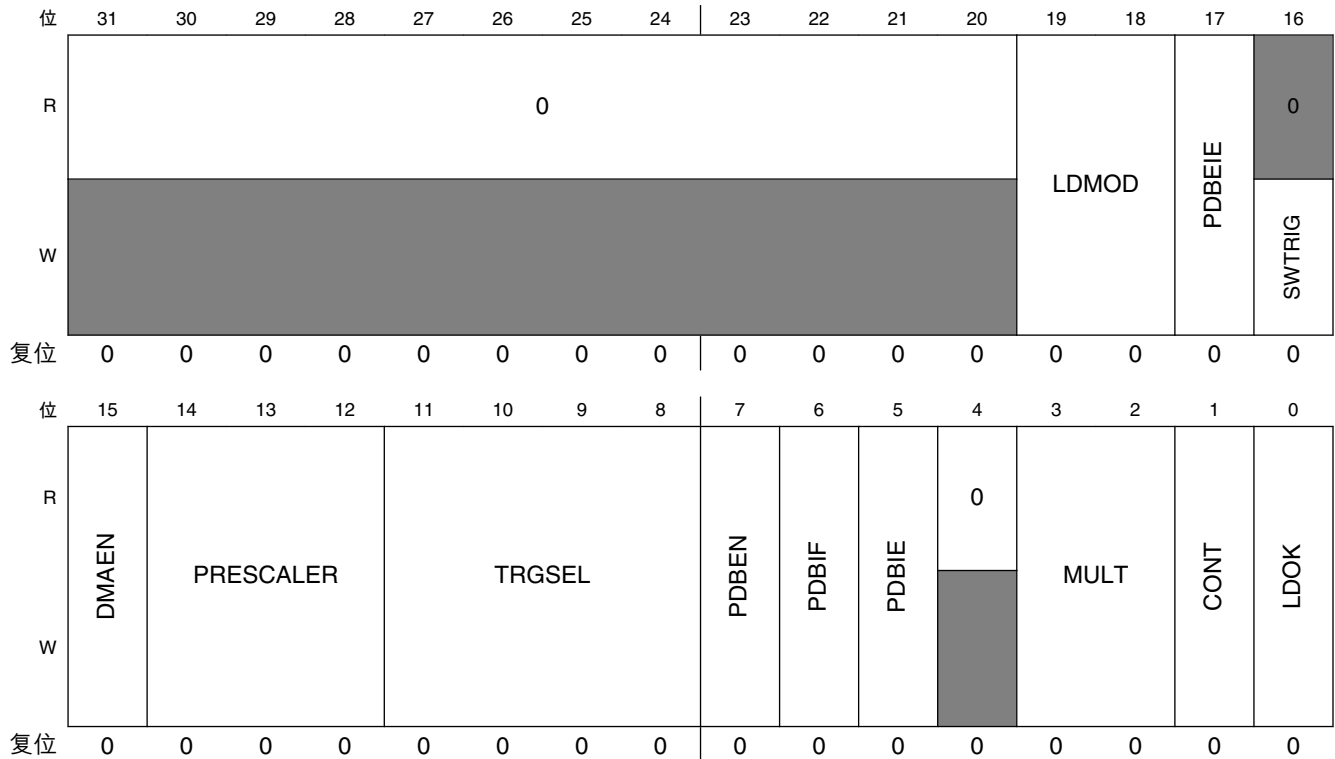
36.3 存储器映射和寄存器定义

PDB 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4003_6000	状态和控制寄存器 (PDB0_SC)	32	R/W	0000_0000h	36.3.1/712
4003_6004	系数值寄存器 (PDB0_MOD)	32	R/W	0000_FFFFh	36.3.2/714
4003_6008	计数器寄存器 (PDB0_CNT)	32	R	0000_0000h	36.3.3/715
4003_600C	中断延迟寄存器 (PDB0_IDLY)	32	R/W	0000_FFFFh	36.3.4/715
4003_6010	通道 n 控制寄存器 1 (PDB0_CH0C1)	32	R/W	0000_0000h	36.3.5/716
4003_6014	通道 n 状态寄存器 (PDB0_CH0S)	32	R/W	0000_0000h	36.3.6/717
4003_6018	通道 n 延时 0 寄存器 (PDB0_CH0DLY0)	32	R/W	0000_0000h	36.3.7/717
4003_601C	通道 n 延时 1 寄存器 (PDB0_CH0DLY1)	32	R/W	0000_0000h	36.3.8/718
4003_6150	DAC 间隔触发器 n 控制寄存器 (PDB0_DACINTC0)	32	R/W	0000_0000h	36.3.9/718
4003_6154	DAC 间隔 n 寄存器 (PDB0_DACINT0)	32	R/W	0000_0000h	36.3.10/719
4003_6158	DAC 间隔触发器 n 控制寄存器 (PDB0_DACINTC1)	32	R/W	0000_0000h	36.3.9/718
4003_6190	脉冲输出 n 使能寄存器 (PDB0_POEN)	32	R/W	0000_0000h	36.3.11/720
4003_6194	脉冲输出 n 延迟寄存器 (PDB0_PO0DLY)	32	R/W	0000_0000h	36.3.12/720

36.3.1 状态和控制寄存器 (PDBx_SC)

地址: 4003_6000h 基准 + 0h 偏移 = 4003_6000h



PDBx_SC 字段描述

字段	描述
31-20 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
19-18 LDMOD	加载模式选择 在将 1 写入 LDOK 之后，选择模式以加载 MOD、IDLY、CHnDLYm、INTx 和 POyDLY 寄存器。 00 在将 1 写入 LDOK 之后，内部寄存器将立即载入来自其缓冲器的值。 01 在将 1 写入 LDOK 之后，当 PDB 计数器达到 MOD 寄存器值时，内部寄存器将载入来自其缓冲器的值。 10 在将 1 写入 LDOK 之后，当检测到触发器输入事件时，内部寄存器将载入来自其缓冲器的值。 11 在将 1 写入 LDOK 之后，当 PDB 计数器达到 MOD 寄存器值或检测到触发器输入事件时，内部寄存器将载入来自其缓冲器的值。
17 PDBEIE	PDB 序列错误中断使能 使能 PDB 序列错误中断。此字段置位时，PDB 通道序列错误标志中的任意一个都会生成一个 PDB 序列错误中断。 0 PDB 序列错误中断已禁用。 1 PDB 序列错误中断已使能。
16 SWTRIG	软件触发

下一页继续介绍此表...

PDBx_SC 字段描述 (继续)

字段	描述
	当 PDB 使能且将软件触发器选为触发器输入源时，将 1 写入此字段会复位并重启计数器。向该字段写入 0 无效。读取此字段会返回 0。
15 DMAEN	DMA 使能 当 DMA 使能时，PDBIF 标志会生成一个 DMA 请求而非中断。 0 DMA 已禁用。 1 DMA 已使能。
14-12 PRESCALER	预分频比选择 000 计数使用的外设时钟分频比为 MULT 所选乘法因数。 001 计数使用的外设时钟分频比为 MULT 所选乘法因数的两倍。 010 计数使用的外设时钟分频比为 MULT 所选乘法因数的四倍。 011 计数使用的外设时钟分频比为 MULT 所选乘法因数的八倍。 100 计数使用的外设时钟分频比为 MULT 所选乘法因数的 16 倍。 101 计数使用的外设时钟分频比为 MULT 所选乘法因数的 32 倍。 110 计数使用的外设时钟分频比为 MULT 所选乘法因数的 64 倍。 111 计数使用的外设时钟分频比为 MULT 所选乘法因数的 128 倍。
11-8 TRGSEL	触发器输入源选择 为 PDB 选择触发器输入源。触发器输入源可以为内部或外部 (EXTRG 引脚)，也可为软件触发器。请参阅芯片配置详情，了解实际的 PDB 输入触发器连接。 0000 选择触发器-输入 0。 0001 选择触发器-输入 1。 0010 选择触发器-输入 2。 0011 选择触发器-输入 3。 0100 选择触发器-输入 4。 0101 选择触发器-输入 5。 0110 选择触发器-输入 6。 0111 选择触发器-输入 7。 1000 选择触发器-输入 8。 1001 选择触发器-输入 9。 1010 选择触发器-输入 10。 1011 选择触发器-输入 11。 1100 选择触发器-输入 12。 1101 选择触发器-输入 13。 1110 选择触发器-输入 14。 1111 软件触发。
7 PDBEN	PDB 使能 0 PDB 已禁用。计数器关闭。 1 PDB 已使能。
6 PDBIF	PDB 中断标志 当计数器值等于 IDLY 寄存器时，此字段置位。写入零会清零此字段。
5 PDBIE	PDB 中断使能 使能 PDB 中断。当此字段置位且 DMAEN 清零时，PDBIF 会生成一个 PDB 中断。

下一页继续介绍此表...

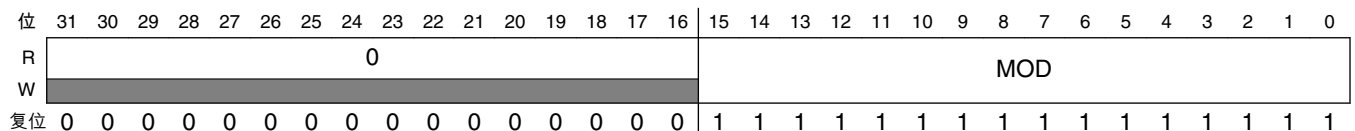
PDBx_SC 字段描述 (继续)

字段	描述
	0 PDB 中断禁用。 1 PDB 中断使能。
4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3-2 MULT	预分频比的乘法因数选择 为计数器时钟选择预分频比乘法因数。 00 乘法因数为 1。 01 乘法因数为 10。 10 乘法因数为 20。 11 乘法因数为 40。
1 CONT	连续模式使能 以连续模式使能 PDB 操作。 0 PDB 在单次模式下的操作方式 1 PDB 在连续模式下的 PDB 操作方式
0 LDOK	装载数据 OK 位 向此位写入 1，结果将以写入（并存储在）其内部缓冲器的值更新 MOD、IDLY、CHnDLYm、DACINTx 和 POyDLY 寄存器。MOD、IDLY、CHnDLYm、DACINTx 和 POyDLY 将依据 LDMOD（负载模式选择）生效。向 LDOK 字段写入 1 之后，上述寄存器的缓冲器中的值将失效，并且不能对其缓冲器进行写操作，直到缓冲器中的值被载入其内部寄存器为止。 <ul style="list-style-type: none"> LDOK 只能在 PDBEN 置位时才能写入，或者，可以在 PDBEN 被写入 1 的同时执行写入操作。 当内部缓冲器中的值被载入寄存器或者 PDBEN 位（PDB 使能）被清零时，LDOK 将自动清零。 向 LDOK 写入 0 无效。

36.3.2 系数值寄存器 (PDBx_MOD)

注意：该寄存器为内部缓冲，任何写入该寄存器的值会被写入其内部缓存器。换言之，内部器件总线不直接写入该寄存器。只有当 1 被写入 SC[LDOK] 位，该寄存器的内部缓冲器中的值会被加载到该寄存器。

地址: 4003_6000h 基准 + 4h 偏移 = 4003_6004h



PDBx_MOD 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
MOD	PDB 模数值 指定计数器的周期。当计数器达到此值时，计数器将复位至零。如果 PDB 处于连续模式，则计数重新开始。读取此字段会返回在 PDB 当前周期内有效的内部寄存器的值。

36.3.3 计数器寄存器 (PDBx_CNT)

地址: 4003_6000h 基准 + 8h 偏移 = 4003_6008h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CNT															
W	0																															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_CNT 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CNT	PDB 计数器 包含计数器的当前值。

36.3.4 中断延迟寄存器 (PDBx_IDLY)

注意: 该寄存器为内部缓冲, 任何写入该寄存器的值会被写入其内部缓存器。换言之, 内部器件总线不直接写入该寄存器。只有当 1 被写入 SC[LDOK] 位, 该寄存器的内部缓冲器中的值会被加载到该寄存器。

地址: 4003_6000h 基准 + Ch 偏移 = 4003_600Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																IDLY															
W	0																															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

PDBx_IDLY 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
IDLY	PDB 中断延迟 指定用于产生 PDB 中断的延迟值。可用于在 PDB 周期中的产生中断的某个时间点。如果使能，当计数器等于 IDLY 时，将生成一个 PDB 中断。读取此字段会返回在 PDB 当前周期内有效的内部寄存器的值。

36.3.5 通道 n 控制寄存器 1 (PDBx_CHnC1)

每个 PDB 通道都有一个控制寄存器 CHnC1。此寄存器中的字段控制各个 PDB 通道工作模式的功能。

地址: 4003_6000h 基准 + 10h 偏移 + (40d × i), 其中 i=0d 到 0d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								BB								TOS								EN							
W	0								0								0								0							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_CHnC1 字段描述

字段	描述
31-24 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
23-16 BB	PDB 通道预触发器背靠背工作模式使能 使能 PDB ADC 预触发器背靠背工作模式。此 MCU 中仅实现了低位 M 预触发器位。背靠背工作模式使得 ADC 转换完成后，可以自动触发下一个 PDB 通道预触发器和触发器输出，这样，就可以在下一组配置和结果寄存器处触发 ADC 转换。应用代码必须使能背靠背连接链的开头，仅使能 PDB 预触发器的背靠背工作模式。 0 PDB 通道的对应预触发器背靠背工作模式已禁用。 1 PDB 通道的对应预触发器背靠背工作模式已使能。
15-8 TOS	PDB 通道预触发器输出选择 这些位选择 PDB ADC 预触发器输出。此 MCU 中仅实现了低位 M 预触发器位。 0 PDB 通道的对应预触发器处于旁路模式。在所选触发器输入源上检测到上升沿后的一个外设时钟周期时，预触发器置位，或者，选择软件触发器并向 SWTRIG 写入 1。 1 当计数器达到通道延迟寄存器所设定的数值时，再加上在所选触发器输入源上检测到上升沿后的一个外设时钟周期时，预触发器置位，或者，选择软件触发器并向 SWTRIG 写入 1。
EN	PDB 通道预触发器使能 使能 PDB ADC 预触发器输出。此 MCU 中仅实现了低位 M 预触发器字段。 0 PDB 通道的对应预触发器已禁用。 1 PDB 通道的对应预触发器已使能。

36.3.6 通道 n 状态寄存器 (PDBx_CHnS)

地址: 4003_6000h 基准 + 14h 偏移 + (40d × i), 其中 i=0d 到 0d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CF								0								ERR							
W	0								0								0								0							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_CHnS 字段描述

字段	描述
31–24 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
23–16 CF	PDB 通道标志 当 PDB 计数器匹配 CHnDLYm 时, CF[m] 字段置位。写入 0 以清零这些位。
15–8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
ERR	PDB 通道序列错误标志 此 MCU 中仅实现了低位 M 位。 0 未在 PDB 通道的对应预触发器上检测到序列错误。 1 在 PDB 通道的对应预触发器上检测到序列错误。ADCn 数据块可由 PDB 通道 n 中的一个预触发器触发以进行转换。当由 PDB 通道 n 中预触发器之一触发的一次转换正在进行时, 来自 PDB 通道对应预触发器的新触发器 m 不能被 ADCn 接受, 然后, ERR[m] 置位。写入 0 以清零序列错误标志。

36.3.7 通道 n 延时 0 寄存器 (PDBx_CHnDLY0)

注意: 该寄存器为内部缓冲, 任何写入该寄存器的值会被写入其内部缓存器。换言之, 内部器件总线不直接写入该寄存器。只有当 1 被写入 SC[LDOK] 位, 该寄存器的内部缓冲器中的值会被加载到该寄存器。

地址: 4003_6000h 基准 + 18h 偏移 + (40d × i), 其中 i=0d 到 0d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																DLY															
W	0																0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_CHnDLY0 字段描述

字段	描述
31–16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

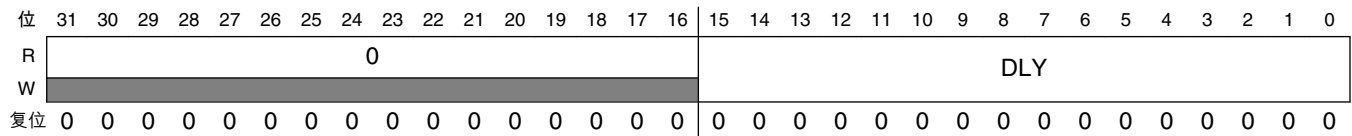
PDBx_CHnDLY0 字段描述 (继续)

字段	描述
DLY	PDB 通道延迟 指定通道对应预触发器的延迟值。预触发器在计数器等于 DLY 时置位。读取此字段会返回在当前 PDB 周期内有效的内部寄存器的值。

36.3.8 通道 n 延时 1 寄存器 (PDBx_CHnDLY1)

注意：该寄存器为内部缓冲，任何写入该寄存器的值会被写入其内部缓存器。换言之，内部器件总线不直接写入该寄存器。只有当 1 被写入 SC[LDOK] 位，该寄存器的内部缓冲器中的值会被加载到该寄存器。

地址: 4003_6000h 基准 + 1Ch 偏移 + (40d × i), 其中 i=0d 到 0d

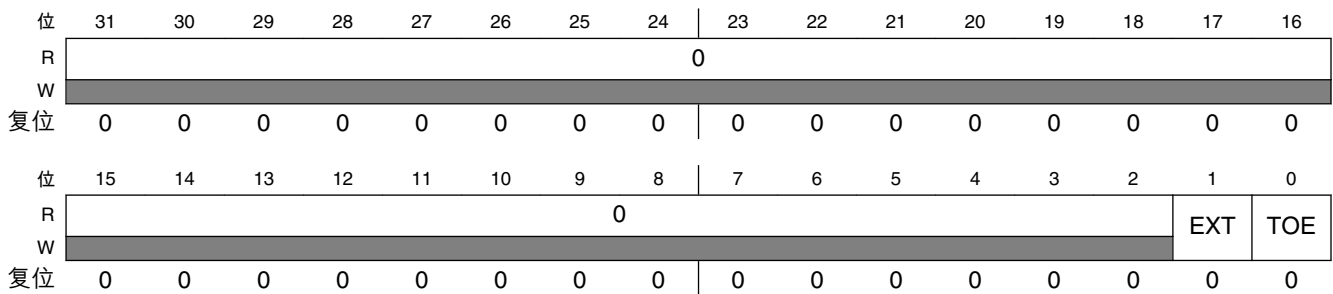


PDBx_CHnDLY1 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
DLY	PDB 通道延迟 这些位指定通道对应预触发器的延迟值。预触发器在计数器等于 DLY 时置位。读取这些位会返回在当前 PDB 周期内有效的内部寄存器的值。

36.3.9 DAC 间隔触发器 n 控制寄存器 (PDBx_DACINTCn)

地址: 4003_6000h 基准 + 150h 偏移 + (8d × i), 其中 i=0d 到 1d



PDBx_DACINTCn 字段描述

字段	描述
31-2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 EXT	DAC 外部触发器输入使能 使能 DAC 间隔计数器的外部触发器。 0 DAC 外部触发器输入已禁用。在所选触发器输入源上检测到上升沿时，DAC 间隔计数器复位并开始计数，或者，选择软件触发器并向 SWTRIG 写入 1。 1 DAC 外部触发器输入已使能。DAC 间隔计数器已旁路，DAC 外部触发器输入触发 DAC 间隔触发器。
0 TOE	DAC 间隔触发器使能 此位使能 DAC 间隔触发器。 0 DAC 间隔触发器已禁用。 1 DAC 间隔触发器已使能。

36.3.10 DAC 间隔 n 寄存器 (PDBx_DACINTn)

注意：该寄存器为内部缓冲，任何写入该寄存器的值会被写入其内部缓存器。换言之，内部器件总线不直接写入该寄存器。只有当 1 被写入 SC[LDOK] 位，该寄存器的内部缓冲器中的值会被加载到该寄存器。

地址：4003_6000h 基准 + 154h 偏移 + (8d × i)，其中 i=0d 到 0d

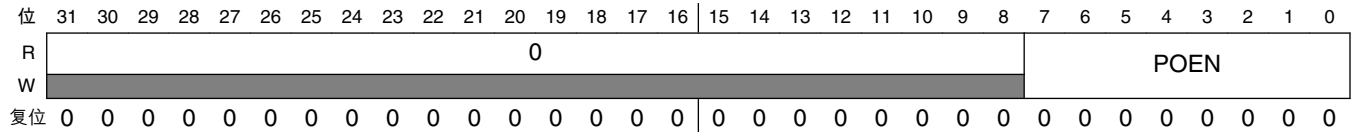
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																INT															
W	0																0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_DACINTn 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
INT	DAC 间隔 这些位指定 DAC 间隔触发器的间隔值。在 DAC 间隔计数器等于 DACINT 时，DAC 间隔触发器将触发 DAC[1:0]更新功能。读取这些位会返回在当前 PDB 周期内有效的内部寄存器的值。

36.3.11 脉冲输出 n 使能寄存器 (PDBx_POEN)

地址: 4003_6000h 基准 + 190h 偏移 = 4003_6190h



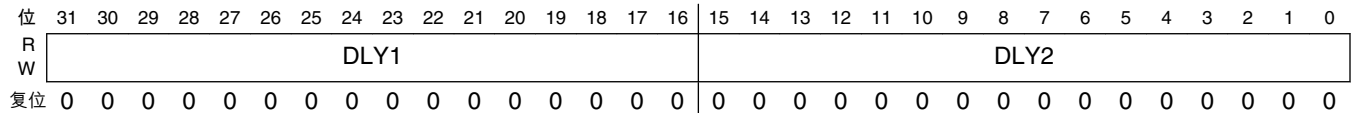
PDBx_POEN 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
POEN	PDB 脉冲输出使能 使能脉冲输出。此 MCU 中仅实现了低位 8 位。 0 PDB 脉冲输出已禁用 1 PDB 脉冲输出已使能

36.3.12 脉冲输出 n 延迟寄存器 (PDBx_POnDLY)

注意: 该寄存器为内部缓冲, 任何写入该寄存器的值会被写入其内部缓存器。换言之, 内部器件总线不直接写入该寄存器。只有当 1 被写入 SC[LDOK] 位, 该寄存器的内部缓冲器中的值会被加载到该寄存器。

地址: 4003_6000h 基准 + 194h 偏移 + (4d × i), 其中 i=0d 到 0d



PDBx_POnDLY 字段描述

字段	描述
31-16 DLY1	PDB 脉冲输出延迟 1 这些位指定 PDB 脉冲输出的延迟 1 值。脉冲输出在 PDB 计数器等于 DLY1 时变为高电平。读取这些位会返回在当前 PDB 周期内有效的内部寄存器的值。
DLY2	PDB 脉冲输出延迟 2 这些位指定 PDB 脉冲输出的延迟 2 值。脉冲输出在 PDB 计数器等于 DLY2 时变为低电平。读取这些位会返回在当前 PDB 周期内有效的内部寄存器的值。

36.4 功能说明

36.4.1 PDB 预触发器和触发器输出

PDB 含有一个计数器，其输出将与多个不同的数字值进行比较。如果 PDB 使能，则触发器输入事件会复位计数器并使其开始计数。触发器输入事件定义为在所选触发器输入源上检测到的上升沿，或者如果选择的是软件触发器，则为 SC[SWTRIG] 被写入 1 时。对于每个通道，延迟 m 决定着从触发器输入事件认定到预触发器 m 输出信号中的变更开始时的时间。此时间定义为：

- 预触发器 m 的触发器输入事件 = (预分频器 × 乘法因数 × 延迟 m) + 2 个外设时钟周期
- 加上一个额外的外设时钟周期，以确定通道触发器输出发生变化的时间。

每个通道均与一个 ADC 关联。PDB 通道 n 预触发器输出 0 至 M ；每个预触发器输出均连接至 ADC 硬件触发器选择和硬件触发器输入。预触发器用于在实际触发之前预处理 ADC 模块。当 ADC 收到触发器的上升沿时，ADC 将根据预触发器确定的前提条件启动转换。ADC 含有 M 组配置和结果寄存器，使其可以在 M 个不同模拟源之间交替进行转换(如乒乓游戏一样)。预触发器输出用于指定下一个要采样的信号。当预触发器 m 认定时，ADC 转换将与配置和结果寄存器组 m 一起触发。

下图所示波形展示了 PDB 通道 n 的预触发器和触发器输出。延迟可以用 CH n DLY m 寄存器独立设置，并且预触发器可以在 CH n C1[EN[m]]中使能或禁用。

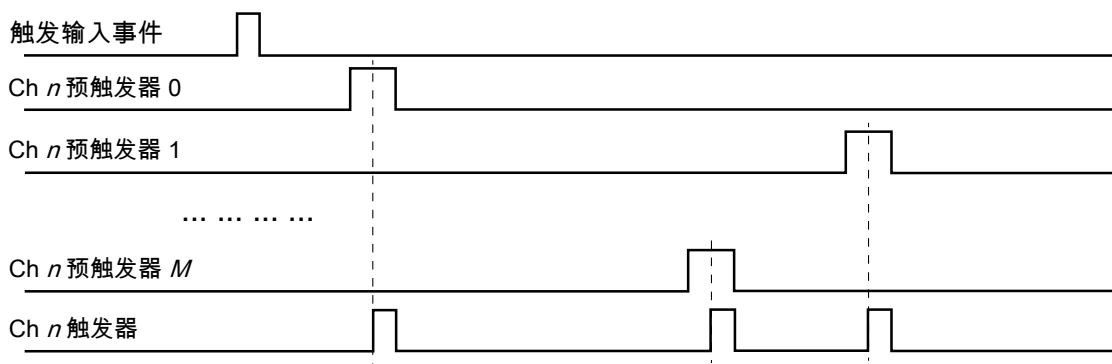


图 36-43. 预触发器和触发器输出

如果 CH n C1[TOS[m]]清零，可以选择绕过 CH n DLY m 寄存器中的延迟。这种情况下，当发生触发器输入事件时，预触发器 m 会在 2 个外设时钟周期之后认定。

PDB 可以配置为背靠背工作模式。背靠背工作模式使得 ADC 转换完成可以触发下一个 PDB 通道预触发器和触发器输出，这样，就可以在下一组配置和结果寄存器处触发 ADC 转换。当通过置位 $CHnC1[BB[m]]$ 使能背靠背工作模式时，延迟 m 将被忽略，并且预触发器 m 将在收到确认信号的 $m2$ 个外设周期之后置位。此 MCU 中的确认连接见 [背靠背应答连接](#)。

当来自 PDB 通道 n 的预触发器认定时，预触发器的关联锁变成有效。关联锁定由对应的 $ADCnSC1[COCO]$ 的上升沿释放； $ADCnSC1[COCO]$ 应在读取转换结果之后清零，以便生成 $ADCnSC1[COCO]$ 的下一个上升沿，以在随后清除锁定。锁定在下列情况下变为无效：

- 对应的 $ADCnSC1[COCO]$ 的上升沿发生时；
- 或者对应的 PDB 预触发器被禁用时；
- 或者 PDB 被禁用时。

当通道 n 中预触发器的任意锁定有效时，通道 n 触发器输出将被禁止。如果有新的预触发器 m 在 PDB 通道 n 中存在有效锁定时认定，则置位寄存器标志位 $CHnS[ERR[m]]$ （与预触发器 m 关联）。如果 $SC[PDBEIE]$ 置位，则会生成序列错误中断。出现序列错误的原因一般是设置的延迟 m 太短并且预触发器 m 在前面触发的 ADC 转换完成之前认定。

当 PDB 计数器达到 IDLY 寄存器中的设定值时， $SC[PDBIF]$ 标志置位。如果 $SC[PDBIE]$ 置位且 $SC[DMAEN]$ 清零，则可以生成 PDB 中断。如果 $SC[DMAEN]$ 置位，则 PDB 会在 $SC[PDBIF]$ 标志置位时请求 DMA 传输。

MOD 寄存器中的系数值用于在计数结束时，将计数器复位至零。如果 $SC[CONT]$ 置位，则计数器会恢复新的计数；否则，在下一个触发器输入事件发生之前，计数器会停止工作。

36.4.2 PDB 触发器输入源选择

PDB 最多可以有 15 个触发器输入源，即触发输入 0 至触发器输入 14。它们连接至片上或片外事件源。PDB 可由软件通过 $SC[SWTRIG]$ 触发。

若要了解此 MCU 中实现的触发器输入源，请参阅芯片配置信息。

36.4.3 脉冲输出

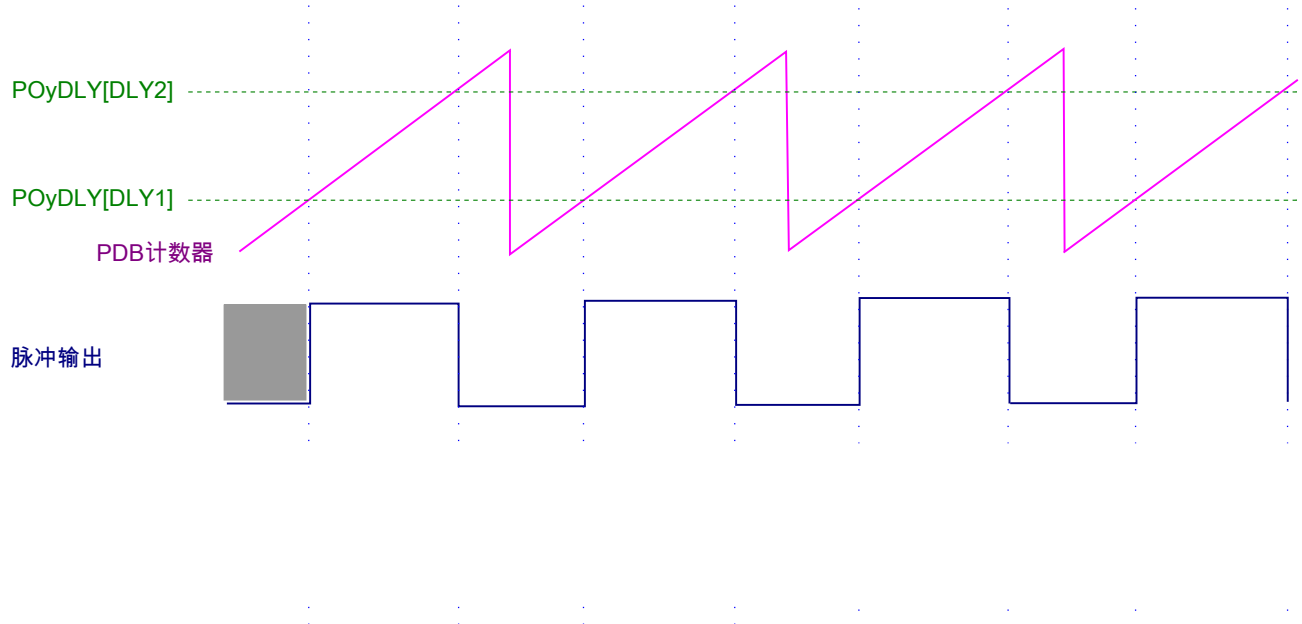
PDB 可以生成宽度可配置的脉冲输出。

- 当 PDB 计数器达到 $POyDLY[DLY1]$ 中的设定值时，脉冲输出变高。
- 当 PDB 计数器达到 $POyDLY[DLY2]$ 时，则变低。

$POyDLY[DLY2]$ 可以设为大于或小于 $POyDLY[DLY1]$ 。

ADC 预触发器/触发器输出和脉冲输出生成具有相同的时基，因为它们共用 PDB 计数器。本 MCU 中实施的脉冲输出连接详见器件芯片配置说明。

脉冲输出生成， $DLY2 > DLY1$



脉冲输出生成， $DLY1 > DLY2$

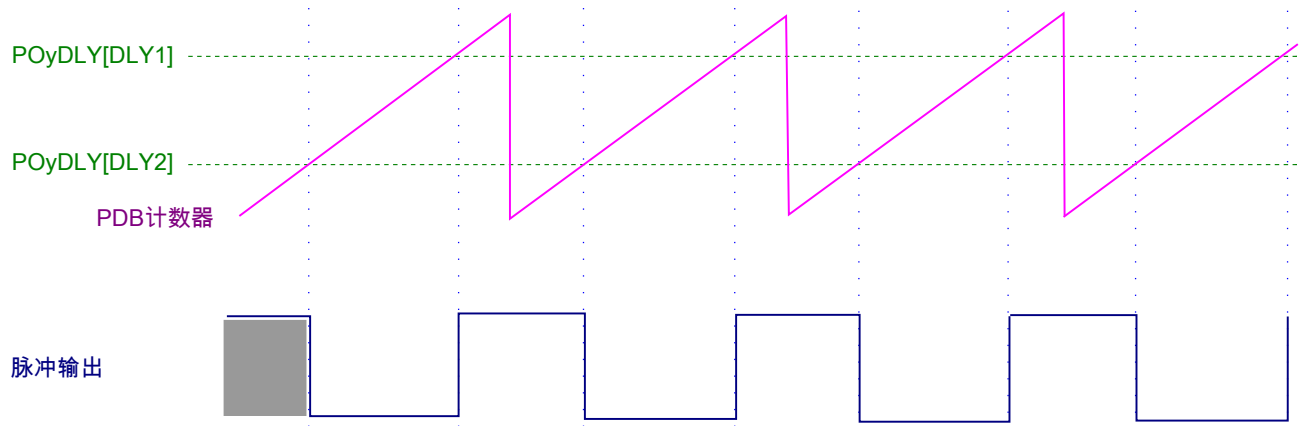


图 36-44. 脉冲输出是如何生成的

36.4.4 更新延迟寄存器

下列寄存器控制 PDB 操作的时序；在有些应用中，它们可能需要同时生效。

- PDB 系数值寄存器(MOD)

- PDB 中断延迟寄存器(IDLY)
- PDB 通道 n 延迟 m 寄存器(CH n DLY m)
- DAC 间隔 x 寄存器(DACINT x)
- PDB 脉冲输出 y 延迟寄存器(PO y DLY)

它们的内部寄存器搭载缓冲机制，写入它们的任何值都先写入它们的缓冲器。下表总结了会导致用缓冲器中的值更新它们的内部寄存器的情况。

表 36-46. 延迟寄存器更新情况

SC[LDMOD]	更新延迟寄存器
00	在将 1 写入 SC[LDOK]之后，内部寄存器将立即载入来自其缓冲器的值。
01	在将 1 写入 SC[LDOK]之后，PDB 计数器达到 MOD 寄存器值。
10	在将 1 写入 SC[LDOK]之后，检测到触发器输入事件。
11	在将 1 写入 SC[LDOK]之后，PDB 计数器达到 MOD 寄存器值，或者检测到触发器输入事件。

在将 1 写入 SC[LDOK]之后，不能对缓冲器进行写操作，直到缓冲器中的值被载入其内部寄存器为止。SC[LDOK]在内部寄存器加载时自动清零，这样，应用代码就能读取它，以确定对内部寄存器的更新。

下面的示意图展示了在 SC[LDMOD]为 00 和 x1 时更新内部寄存器的情况。

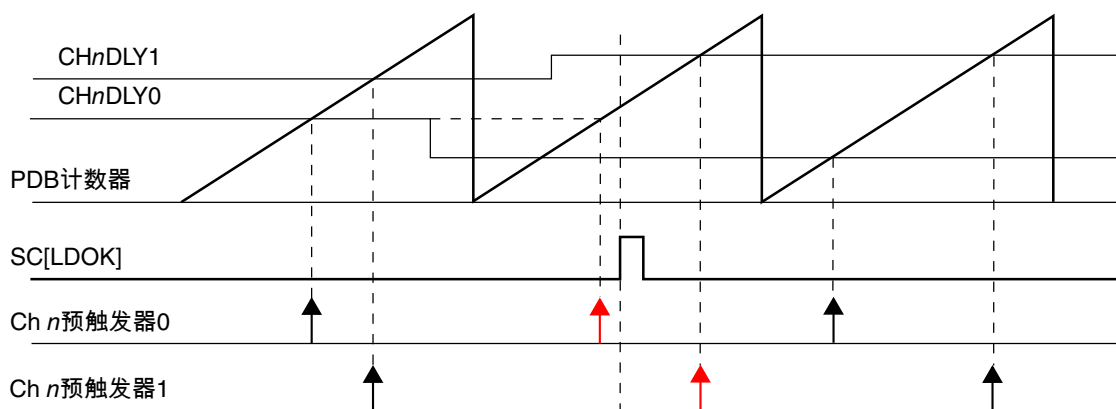


图 36-45. SC[LDMOD] = 00 时更新寄存器

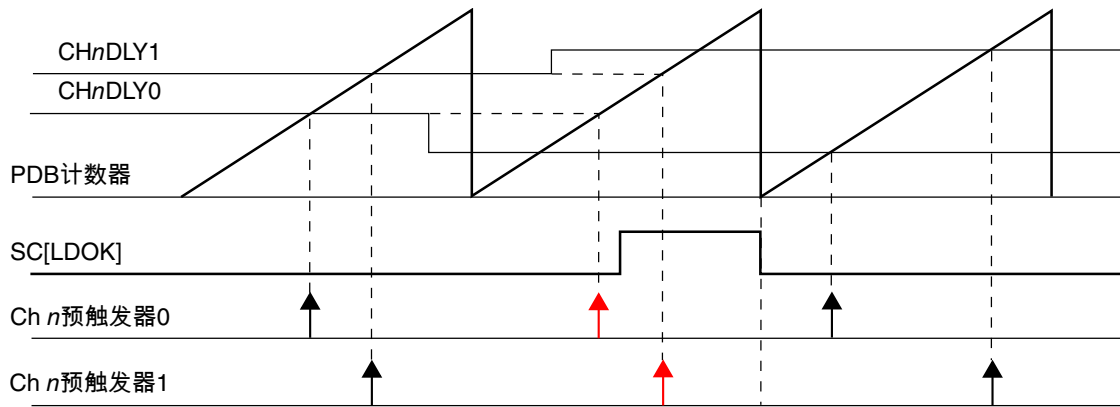


图 36-46. SC[LDMOD] = x1 时更新寄存器

36.4.5 中断

PDB 可生成两种中断：PDB 中断和 PDB 序列错误中断。下表汇总了这些中断。

表 36-47. PDB 中断汇总

中断	标志	使能位
PDB 中断	SC[PDBIF]	SC[PDBIE] = 1 且 SC[DMAEN] = 0
PDB 序列错误中断	CHnS[ERRm]	SC[PDBEIE] = 1

36.4.6 DMA

如果 SC[DMAEN]置位，PDB 会在 SC[PDBIF]置位时生成一个 DMA 传输请求。当 DMA 使能时，不会产生 PDB 中断。

36.5 应用信息

36.5.1 使用预分频器和乘法因数对时序分辨率的影响

使用预分频器和大于 1 的乘法因数会限制外设时钟周期的计数/延迟精度 (对预分频器系数值 X 乘法因数)。如果将乘法因数设为 1 且将预分频器设为 2, 则可以检测到的外设时钟总数的唯一值为偶数值; 如果将预分频器设为 4, 可以按检测方式解码的外设时钟总数的唯一值为 mod(4), 如此等等。如果应用需要真正较长的延迟值并且采用的是设为 128 的预分频器, 则分辨率会限制为 128 个外设时钟周期。

因此，对于给定应用，要尽量采用最低的预分频器和乘法因数。

第 37 章 定时器/PWM 模块 (TPM)

37.1 此模块的芯片实现细节

37.1.1 TPM 实例化信息

该器件包含三个低功耗 TPM 模块。器件中的所有 TPM 模块都配置基本 TPM 功能，带有正交解码器功能，且所有模块都可在停止/VLPS 模式中运行。停止/VLPS 模式下，时钟源可以来自外部或内部。

下表显示了这些模块的配置方式。

表 37-1. TPM 配置

TPM 实例	通道数量	功能/使用
TPM0	6	正交解码器、基本 TPM，可在停止/VLPS 模式下运行
TPM1	2	正交解码器、基本 TPM，可在停止/VLPS 模式下运行
TPM2	2	正交解码器、基本 TPM，可在停止/VLPS 模式下运行

注

由于 TPM1/TPM2 上只有 2 个可用通道 (通道 0-1)，因此与通道 2-5 相关的寄存器 (例如状态和控制寄存器、数值寄存器) 不适用于此器件的 TPM1/TPM2。

37.1.2 时钟选项

TPM 模块使用单个 TPM 时钟源，该时钟可从 OSCERCLK、MCGIRCLK、MCGPLLCLK 或 MCGFLLCLK 中选择。选定的时钟源由 SIM_SOPT2[TPMSRC] 和 SIM_SOPT2[PLLFLLSEL]控制寄存器控制。

每个 TPM 还支持外部时钟模式 (TPM_SC [CMOD]=1x)，该模式下计数器在检测到外部时钟输入的同步（与选定的 TPM 时钟源同步）上升沿之后递增。可用的外部时钟（为 TPM_CLKIN0 或 TPM_CLKIN1）由 SIM_SOPT9 [TPMxCLKSEL] 控制寄存器进行选择。为了确保有效运行，选定的外部时钟的频率必须少于选定的 TPM 时钟源频率的一半。

37.1.3 触发选项

每个 TPM 都有一个可选的触发输入源，受 TPMx_CONF[TRGSEL] 位控制，用于启动计数器"与"或者"或"重新加载计数器。可用选项如下表所示。

表 37-2. TPM 触发选项

TPMx_CONF[TRGSEL]	所选源
0000	外部触发器
0001	CMP 0
0010	保留
0011	保留
0100	PIT 通道 0 输出
0101	PIT 通道 1 输出
0110	PIT 通道 2 输出
0111	PIT 通道 3 输出
1000	TPM0 溢出
1001	TPM1 溢出
1010	TPM2 溢出
1011	保留
1100	RTC 警报
1101	RTC 秒钟
1110	LPTMR 输出
1111	软件触发

37.1.4 全局时基

每个 TPM 都具备全局时基功能，受 TPMx_CONF[GTBEEN] 位控制。当此选项使能时，TPM1 配置为全局时间。

37.1.5 TPM 中断

TPM 有多个中断源。然而，这些中断源经过 OR 运算，来对中断控制器产生单一的中断请求。当出现 TPM 中断时，读取 TPM 状态寄存器以确定确切的中断源。

37.2 简介

TPM (定时器/PWM 模块)是一种 2 至 8 通道定时器，支持输入捕捉、输出比较和生成用于控制电机和电源管理应用的 PWM 信号。

计数器、比较和捕捉寄存器的时钟源是异步时钟，则可以在低功耗模式下保持使能。异步 DMA 模式下使用 TPM 的实例可参见 [AN4631: 使用 Kinetis L 系列的异步 DMA 功能](#)。

37.2.1 TPM 理念

The TPM 是基于一个简单的定时器（即 HCS08 定时器 PWM 模块——TPM，这个定时器在恩智浦 8 位微处理器上已经使用多年）而设计的。TPM 扩展了功能以支持在低功耗模式下运行，在低功耗模式下，异步时钟作为计数器，比较和捕捉寄存器的时钟源，使得它们能够保持正常工作。

37.2.2 特性

TPM 特性包括：

- TPM 时钟模式
 - 可在异步计数器时钟的每个边沿上递增
 - 当与计数器时钟同步后，TPM 外部时钟可让计数器在每个时钟上升沿递增
- 预分频器 (1、2、4、8、16、32、64 或 128 分频)
- TPM 包括 16 位计数器
 - 它可以是自由运行计数器或模数计数器
 - 计数可以是向上计数或者先向上后向下计数
- 包括 6 条通道，均可配置为输入捕获、输出比较、边沿对齐 PWM 模式或中心对齐 PWM 模式
 - 在输入捕捉模式下，捕捉可以在上升沿、下降沿或两个边沿同时发生

- 在输出比较模式下，可以对输出信号采取置位、清除或匹配切换操作
- 所有通道均可配置为边沿对齐 PWM 模式或中心对齐 PWM 模式
- 支持根据通道生成中断"与"或者"或"DMA 请求
- 支持根据通道在计数器溢出时生成中断"与"或者"或"DMA 请求
- 支持可选择触发输入，来选择复位或者使计数器启动递增。
 - 计数器还可选择在计数器溢出时停止递增
- 支持计数器溢出和根据通道生成硬件触发信号

37.2.3 工作模式

调试模式下，TPM 可配置为临时暂停所有计数，直至内核返回至正常用户工作模式或正常工作。计数器暂停时，同时忽略触发输入和输入捕捉事件。

休眠模式下，TPM 可配置为正常工作或在休眠模式期间暂停所有计数。计数器暂停时，同时忽略触发输入和输入捕捉事件。

停止模式下，TPM 计数器时钟可正常运行，TPM 可生成异步中断使 MCU 从停止模式中退出。

37.2.4 结构框图

TPM 为每个通道使用一个输入/输出 (I/O)，CH_n (TPM 通道 (n)) 中的 n 为通道编号。

下图展示了 TPM 结构。TPM 的核心组件是一个终值可编程的 16 位计数器，该计数器可向上或者先向上后向下计数。

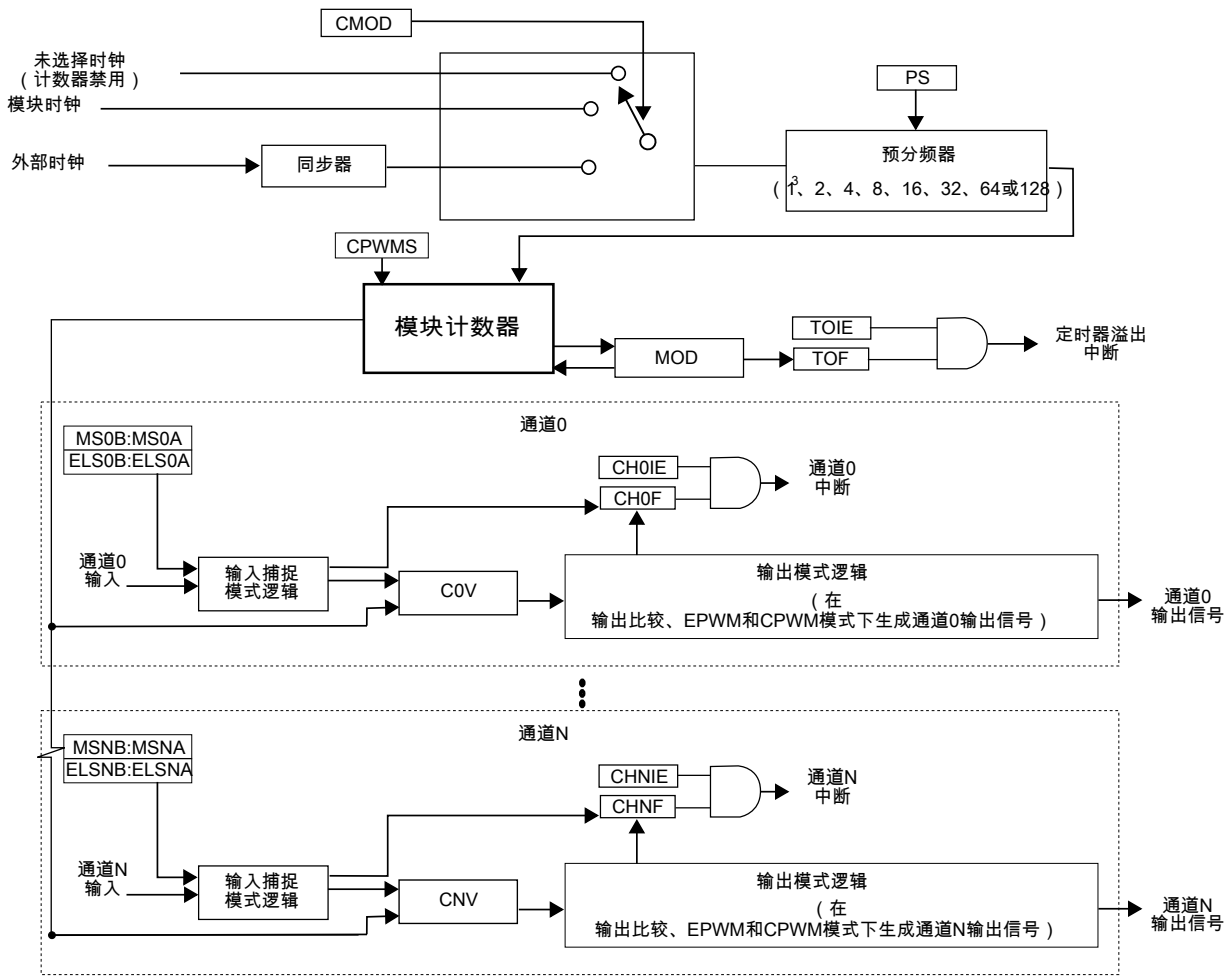


图 37-1. TPM 结构框图

37.3 TPM 信号说明

表 37-3 显示了用户可访问的 TPM 信号。

表 37-3. TPM 信号说明

信号	说明	I/O
TPM_EXTCLK	外部时钟。当与计数器时钟同步后，TPM 外部时钟可让计数器在每个时钟上升沿递增。	I
TPM_CHn	TPM 通道 (n = 5 至 0)。在输出比较或 PWM 模式下且 TPM 计数器使能时，TPM 通道引脚将配置为输出，否则 TPM 通道引脚为输入。	I/O

37.3.1 TPM_EXTCLK — TPM 外部时钟

如果通过 SC 寄存器中 CMOD[1:0]位选中外部输入信号，则该信号上升沿被用来递增 TPM 计数器。输入信号必须低于 TPM 计数器时钟频率的一半。选择了外部输入后，还会用到 TPM 计数器预分频器的选择和设置功能。

37.3.2 TPM_CHn — TPM 通道 (n) I/O 引脚

每个 TPM 通道均可配置为输入或输出。每个通道（无论是输入还是输出）关联的方向是根据分配给该通道的模式选择的。

37.4 存储器映射和寄存器定义

本节提供所有 TPM 寄存器的详细说明。

尝试访问 TPM 存储器映射中的保留寄存器位置将会导致总线错误。

TPM 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4003_8000	版本 ID 寄存器 (TPM0_VERID)	32	R	0500_0007h	37.4.1/734
4003_8004	参数寄存器 (TPM0_PARAM)	32	R	参见章节	37.4.2/735
4003_8008	TPM 全局寄存器 (TPM0_GLOBAL)	32	R/W	0000_0000h	37.4.3/736
4003_8010	状态和控制 (TPM0_SC)	32	R/W	0000_0000h	37.4.4/737
4003_8014	计数器寄存器 (TPM0_CNT)	32	R/W	0000_0000h	37.4.5/738
4003_8018	模数寄存器 (TPM0_MOD)	32	R/W	0000_FFFFh	37.4.6/739
4003_801C	捕捉和比较状态 (TPM0_STATUS)	32	R/W	0000_0000h	37.4.7/739
4003_8020	通道 (n) 状态和控制 (TPM0_C0SC)	32	R/W	0000_0000h	37.4.8/741
4003_8024	通道 (n) 值寄存器 (TPM0_C0V)	32	R/W	0000_0000h	37.4.9/743
4003_8028	通道 (n) 状态和控制 (TPM0_C1SC)	32	R/W	0000_0000h	37.4.8/741
4003_802C	通道 (n) 值寄存器 (TPM0_C1V)	32	R/W	0000_0000h	37.4.9/743
4003_8030	通道 (n) 状态和控制 (TPM0_C2SC)	32	R/W	0000_0000h	37.4.8/741
4003_8034	通道 (n) 值寄存器 (TPM0_C2V)	32	R/W	0000_0000h	37.4.9/743
4003_8038	通道 (n) 状态和控制 (TPM0_C3SC)	32	R/W	0000_0000h	37.4.8/741
4003_803C	通道 (n) 值寄存器 (TPM0_C3V)	32	R/W	0000_0000h	37.4.9/743
4003_8040	通道 (n) 状态和控制 (TPM0_C4SC)	32	R/W	0000_0000h	37.4.8/741
4003_8044	通道 (n) 值寄存器 (TPM0_C4V)	32	R/W	0000_0000h	37.4.9/743
4003_8048	通道 (n) 状态和控制 (TPM0_C5SC)	32	R/W	0000_0000h	37.4.8/741
4003_804C	通道 (n) 值寄存器 (TPM0_C5V)	32	R/W	0000_0000h	37.4.9/743

下一页继续介绍此表...

TPM 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4003_8064	组合通道寄存器 (TPM0_COMBINE)	32	R/W	0000_0000h	37.4.10/744
4003_806C	通道触发 (TPM0_TRIG)	32	R/W	0000_0000h	37.4.11/746
4003_8070	通道极性 (TPM0_POL)	32	R/W	0000_0000h	37.4.12/747
4003_8078	滤波器控制 (TPM0_FILTER)	32	R/W	0000_0000h	37.4.13/748
4003_8080	正交解码器控制和状态 (TPM0_QDCTRL)	32	R/W	0000_0000h	37.4.14/749
4003_8084	配置寄存器 (TPM0_CONF)	32	R/W	0000_0000h	37.4.15/750
4003_9000	版本 ID 寄存器 (TPM1_VERID)	32	R	0500_0007h	37.4.1/734
4003_9004	参数寄存器 (TPM1_PARAM)	32	R	参见章节	37.4.2/735
4003_9008	TPM 全局寄存器 (TPM1_GLOBAL)	32	R/W	0000_0000h	37.4.3/736
4003_9010	状态和控制 (TPM1_SC)	32	R/W	0000_0000h	37.4.4/737
4003_9014	计数器寄存器 (TPM1_CNT)	32	R/W	0000_0000h	37.4.5/738
4003_9018	模数寄存器 (TPM1_MOD)	32	R/W	0000_FFFFh	37.4.6/739
4003_901C	捕捉和比较状态 (TPM1_STATUS)	32	R/W	0000_0000h	37.4.7/739
4003_9020	通道 (n) 状态和控制 (TPM1_C0SC)	32	R/W	0000_0000h	37.4.8/741
4003_9024	通道 (n) 值寄存器 (TPM1_C0V)	32	R/W	0000_0000h	37.4.9/743
4003_9028	通道 (n) 状态和控制 (TPM1_C1SC)	32	R/W	0000_0000h	37.4.8/741
4003_902C	通道 (n) 值寄存器 (TPM1_C1V)	32	R/W	0000_0000h	37.4.9/743
4003_9030	通道 (n) 状态和控制 (TPM1_C2SC)	32	R/W	0000_0000h	37.4.8/741
4003_9034	通道 (n) 值寄存器 (TPM1_C2V)	32	R/W	0000_0000h	37.4.9/743
4003_9038	通道 (n) 状态和控制 (TPM1_C3SC)	32	R/W	0000_0000h	37.4.8/741
4003_903C	通道 (n) 值寄存器 (TPM1_C3V)	32	R/W	0000_0000h	37.4.9/743
4003_9040	通道 (n) 状态和控制 (TPM1_C4SC)	32	R/W	0000_0000h	37.4.8/741
4003_9044	通道 (n) 值寄存器 (TPM1_C4V)	32	R/W	0000_0000h	37.4.9/743
4003_9048	通道 (n) 状态和控制 (TPM1_C5SC)	32	R/W	0000_0000h	37.4.8/741
4003_904C	通道 (n) 值寄存器 (TPM1_C5V)	32	R/W	0000_0000h	37.4.9/743
4003_9064	组合通道寄存器 (TPM1_COMBINE)	32	R/W	0000_0000h	37.4.10/744
4003_906C	通道触发 (TPM1_TRIG)	32	R/W	0000_0000h	37.4.11/746
4003_9070	通道极性 (TPM1_POL)	32	R/W	0000_0000h	37.4.12/747
4003_9078	滤波器控制 (TPM1_FILTER)	32	R/W	0000_0000h	37.4.13/748
4003_9080	正交解码器控制和状态 (TPM1_QDCTRL)	32	R/W	0000_0000h	37.4.14/749

下一页继续介绍此表...

TPM 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4003_9084	配置寄存器 (TPM1_CONF)	32	R/W	0000_0000h	37.4.15/750
4003_A000	版本 ID 寄存器 (TPM2_VERID)	32	R	0500_0007h	37.4.1/734
4003_A004	参数寄存器 (TPM2_PARAM)	32	R	参见章节	37.4.2/735
4003_A008	TPM 全局寄存器 (TPM2_GLOBAL)	32	R/W	0000_0000h	37.4.3/736
4003_A010	状态和控制 (TPM2_SC)	32	R/W	0000_0000h	37.4.4/737
4003_A014	计数器寄存器 (TPM2_CNT)	32	R/W	0000_0000h	37.4.5/738
4003_A018	模数寄存器 (TPM2_MOD)	32	R/W	0000_FFFFh	37.4.6/739
4003_A01C	捕捉和比较状态 (TPM2_STATUS)	32	R/W	0000_0000h	37.4.7/739
4003_A020	通道 (n) 状态和控制 (TPM2_C0SC)	32	R/W	0000_0000h	37.4.8/741
4003_A024	通道 (n) 值寄存器 (TPM2_C0V)	32	R/W	0000_0000h	37.4.9/743
4003_A028	通道 (n) 状态和控制 (TPM2_C1SC)	32	R/W	0000_0000h	37.4.8/741
4003_A02C	通道 (n) 值寄存器 (TPM2_C1V)	32	R/W	0000_0000h	37.4.9/743
4003_A030	通道 (n) 状态和控制 (TPM2_C2SC)	32	R/W	0000_0000h	37.4.8/741
4003_A034	通道 (n) 值寄存器 (TPM2_C2V)	32	R/W	0000_0000h	37.4.9/743
4003_A038	通道 (n) 状态和控制 (TPM2_C3SC)	32	R/W	0000_0000h	37.4.8/741
4003_A03C	通道 (n) 值寄存器 (TPM2_C3V)	32	R/W	0000_0000h	37.4.9/743
4003_A040	通道 (n) 状态和控制 (TPM2_C4SC)	32	R/W	0000_0000h	37.4.8/741
4003_A044	通道 (n) 值寄存器 (TPM2_C4V)	32	R/W	0000_0000h	37.4.9/743
4003_A048	通道 (n) 状态和控制 (TPM2_C5SC)	32	R/W	0000_0000h	37.4.8/741
4003_A04C	通道 (n) 值寄存器 (TPM2_C5V)	32	R/W	0000_0000h	37.4.9/743
4003_A064	组合通道寄存器 (TPM2_COMBINE)	32	R/W	0000_0000h	37.4.10/744
4003_A06C	通道触发 (TPM2_TRIG)	32	R/W	0000_0000h	37.4.11/746
4003_A070	通道极性 (TPM2_POL)	32	R/W	0000_0000h	37.4.12/747
4003_A078	滤波器控制 (TPM2_FILTER)	32	R/W	0000_0000h	37.4.13/748
4003_A080	正交解码器控制和状态 (TPM2_QDCTRL)	32	R/W	0000_0000h	37.4.14/749
4003_A084	配置寄存器 (TPM2_CONF)	32	R/W	0000_0000h	37.4.15/750

37.4.1 版本 ID 寄存器 (TPMx_VERID)

地址: 基址 基准 + 0h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
R	MAJOR								MINOR								FEATURE																	
W	0																																	
复位	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

TPMx_VERID 字段描述

字段	描述
31-24 MAJOR	主要版本号 本只读字段返回模块规格的主要版本号。
23-16 MINOR	次要版本号 本只读字段返回模块规格的次要版本号。
FEATURE	功能标识编号 0x0001 标准 TPM 功能。 0x0003 标准 TPM 功能，并带有过滤和合并寄存器功能。 0x0007 标准 TPM 功能，并带有过滤，合并和正交寄存器功能。

37.4.2 参数寄存器 (TPMx_PARAM)

地址: 基址 基准 + 4h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
R	0								WIDTH								TRIG								CHAN																
W	0																																								
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

TPMx_PARAM 字段描述

字段	描述
31-24 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
23-16 WIDTH	计数器宽度 计数器和定时器通道宽度
15-8 TRIG	触发计数 实施的触发输入数量。
CHAN	通道计数 实施的定时器通道数量。

37.4.3 TPM 全局寄存器 (TPMx_GLOBAL)

地址: 基址 基准 + 8h 偏移

位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R	0																
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R	0															RST	0
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0

TPMx_GLOBAL 字段描述

字段	描述
31-2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 RST	软件复位 复位所有内部逻辑和寄存器，全局寄存器除外。在由软件清零之前保持置位。 0 模块未复位。 1 模块已复位。
0 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

37.4.4 状态和控制 (TPMx_SC)

状态和控制寄存器含有溢出状态标志和控制位，用来配置中断使能、模块配置和预分频系数。这些控件与此模块中的所有通道相关。

地址: 基址 基准 + 10h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	[保留]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DMA	TOF	TOIE	CPWMS	CMOD	PS		
W	[保留]								DMA	w1c	TOIE	CPWMS	CMOD	PS		
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_SC 字段描述

字段	描述
31–9 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
8 DMA	DMA 使能 使能溢出标志的 DMA 传输。 0 禁用 DMA 传输。 1 使能 DMA 传输。
7 TOF	定时器溢出标志 当 TPM 计数器达到 MOD 寄存器中的值并递增时，通过硬件置位。将 1 写入 TOF 可将其清零。将 0 写入 TOF 不起作用。 如果标志置位和清零之间出现另一个 TPM 溢出情况，则写入操作无效；因此，TOF 保持原有设置，表示发生另一个溢出。这种情况下，由于之前 TOF 的清除延迟，TOF 中断请求将不会丢失。 0 TPM 计数器未溢出。 1 TPM 计数器已溢出。

下一页继续介绍此表...

TPMx_SC 字段描述 (继续)

字段	描述
6 TOIE	<p>定时器溢出中断使能</p> <p>使能 TPM 溢出中断。</p> <p>0 禁用 TOF 中断。使用软件轮询或 DMA 请求。 1 使能 TOF 中断。TOF 等于 1 时，产生中断。</p>
5 CPWMS	<p>中心对齐 PWM 选择</p> <p>选择 CPWM 模式。该模式配置 TPM 在先增后减计数模式中的操作</p> <p>此字段受写入保护。仅在计数器禁用后可写入。</p> <p>0 TPM 计数器可工作在向上计数模式中。 1 TPM 计数器可工作在先增后减计数模式中。</p>
4-3 CMOD	<p>时钟模式选择</p> <p>选择 TPM 计数器时钟模式。禁用计数器时，字段保持置位，直至在 TPM 时钟域认可。</p> <p>00 TPM 计数器已禁用 01 TPM 在每个 TPM 计数器时钟上递增 10 TPM 计数器在同步至 TPM 计数器时钟的 TPM_EXTCLK 上升沿递增 11 TPM 计数器在选定的外部输入触发器的上升沿递增。</p>
PS	<p>预分频系数选择</p> <p>从 8 个分频系数中选择，用于 CMOD 所选择的时钟模式。</p> <p>此字段受写入保护。仅在计数器禁用后可写入。</p> <p>000 1 分频 001 2 分频 010 4 分频 011 8 分频 100 16 分频 101 32 分频 110 64 分频 111 128 分频</p>

37.4.5 计数器寄存器 (TPMx_CNT)

CNT 寄存器包含 TPM 计数器值。

复位将清除 CNT 寄存器。在 COUNT 写入任意值也会清零计数器。

调试模式下，TPM 计数器不会递增，除非另外配置。

由于同步延迟，读取 CNT 寄存器在寄存器访问方面增加了两个等待状态。

地址: 基址 基准 + 14h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_CNT 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
COUNT	计数器值

37.4.6 模数寄存器 (TPMx_MOD)

模数寄存器包含 TPM 计数器的模数值。当 TPM 计数器达到模数值并递增时，将设置溢出标记 (TOF)，且 TPM 计数器的下一个值取决于选定的计数方法（请参见[计数器寄存器](#)）。

向 MOD 寄存器写入值时，会将此值锁存到缓冲器中。将根据 [MOD 寄存器更新](#) 用 MOD 寄存器的写缓冲器值更新此 MOD 寄存器。MOD 写入缓冲区的额外写入被忽略，直至寄存器更新。

在写入到 MOD 寄存器之前，建议通过写入 CNT 以初始化 TPM 计数器，从而避免对计数器首次发生溢出的时间造成混淆。

地址: 基址 基准 + 18h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																MOD															
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TPMx_MOD 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
MOD	模数值 此字段必须写入单 16 位或 32 位访问。

37.4.7 捕捉和比较状态 (TPMx_STATUS)

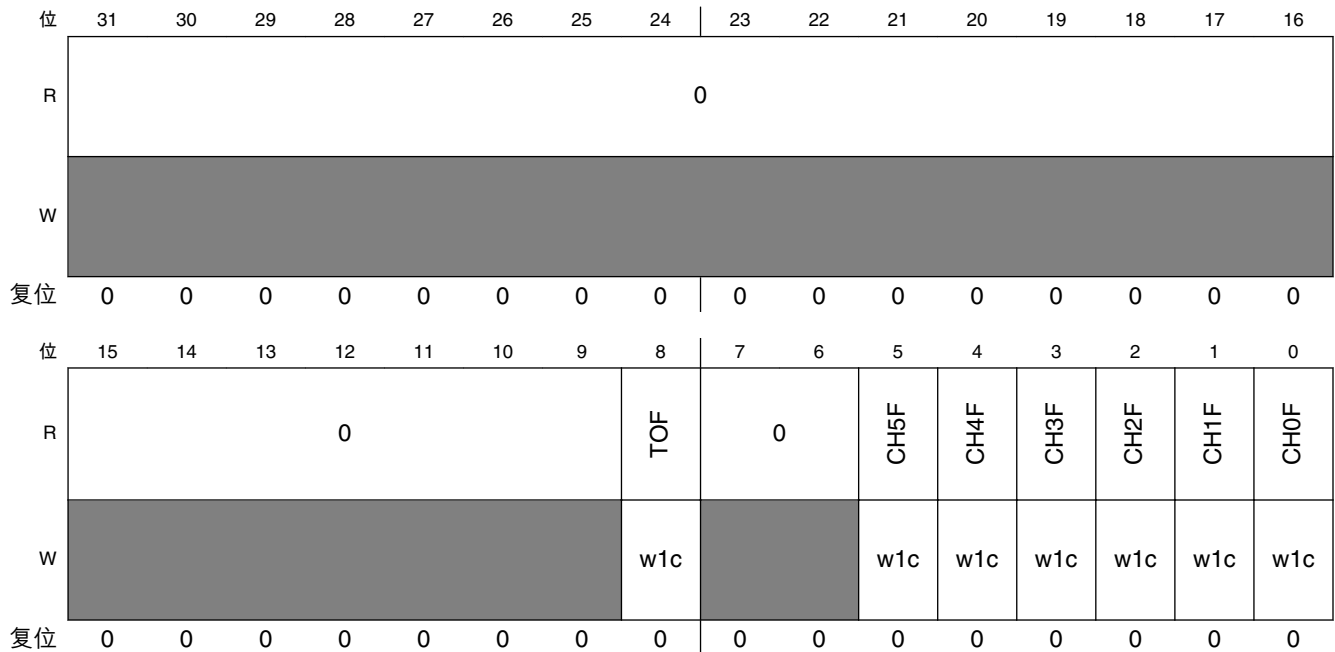
对于每个 TPM 通道，以及 SC[TOF]，STATUS 寄存器包含状态标记的副本 CnSC[CHnF]，以便于软件操作。

STATUS 中的每个 CHnF 位都是 CnSC 中的 CHnF 位的镜像。只需读取一次 STATUS 便可检查所有 CHnF 位。将所有 1 写入 STATUS 可以清零所有 CHnF 位。

当通道上发生事件时，硬件设置单独的通道标志。将 1 写入 CHF 可将其清零。向 CHF 写入 0 不起作用。

若在标志置位和写入操作之间发生另一次事件，则写入操作无效；因此，CHF 保持置位状态，表示发生了另一次事件。在这种情况下，由于之前 CHF 的清除序列，CHF 中断请求将不会丢失。

地址: 基址 基准 + 1Ch 偏移



TPMx_STATUS 字段描述

字段	描述
31-9 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
8 TOF	定时器溢出标志 参见寄存器说明 0 TPM 计数器未溢出。 1 TPM 计数器已溢出。
7-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 CH5F	通道 5 标志 参见寄存器说明。 0 未发生通道事件。 1 发生通道事件。

下一页继续介绍此表...

TPMx_STATUS 字段描述 (继续)

字段	描述
4 CH4F	通道 4 标志 参见寄存器说明。 0 未发生通道事件。 1 发生通道事件。
3 CH3F	通道 3 标志 参见寄存器说明。 0 未发生通道事件。 1 发生通道事件。
2 CH2F	通道 2 标志 参见寄存器说明。 0 未发生通道事件。 1 发生通道事件。
1 CH1F	通道 1 标志 参见寄存器说明。 0 未发生通道事件。 1 发生通道事件。
0 CH0F	通道 0 标志 参见寄存器说明 0 未发生通道事件。 1 发生通道事件。

37.4.8 通道 (n) 状态和控制 (TPMx_CnSC)

CnSC 包含用于配置中断使能、通道配置和引脚功能的通道中断状态标志和控制位。从一个通道模式切换到另一个不同的通道模式时，必须先禁用通道，且必须在 TPM 计数器时钟域中确认。

表 37-50. 模式、边沿和级别选择

CPWMS	MSnB:MSnA	ELSnB:ELSnA	模式	配置
X	00	00	无	通道禁用
X	01	00	软件比较	引脚不用于 TPM
0	00	01	输入捕捉	仅在上升沿上捕捉
		10		仅在下降沿捕捉
		11		在上升沿或下降沿上捕捉

下一页继续介绍此表...

表 37-50. 模式、边沿和级别选择 (继续)

CPWMS	MSnB:MSnA	ELSnB:ELSnA	模式	配置
	01	01	输出比较	匹配时切换输出
		10		匹配时清零输出
		11		匹配时设置输出
	10	10	边沿对齐的 PWM	高真脉冲 (匹配时清零输出, 重载时置位输出)
		X1		低真脉冲 (匹配时置位输出, 重载时清零输出)
	11		10	输出比较
01			脉冲输出高匹配	
1	10	10	中心对齐的 PWM	高真脉冲 (向上匹配时清零输出, 向下匹配时置位输出)
		01		低真脉冲 (向上匹配时置位输出, 向下匹配时清零输出)

地址: 基址 + 20h 偏移 + (8d × i), 其中 i=0d 到 5d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	[Shaded]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CHF	CHIE	MSB	MSA	ELSB	ELSA	0	DMA
W	[Shaded]								w1c						[Shaded]	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_CnSC 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 CHF	通道标志 当通道上发生事件时, 由硬件设置。通过向 CHF 位写入 1 可将 CHF 清除。向 CHF 写入 0 不起作用。 若在 CHF 置位和写入操作之间发生另一次事件, 则写入操作无效; 因此, CHF 保持置位状态, 表示发生了另一次事件。这种情况下, 由于之前 CHF 的清除延迟, CHF 中断请求将不会丢失。 0 未发生通道事件。 1 发生通道事件。
6 CHIE	通道中断使能 使能通道中断。

下一页继续介绍此表...

TPMx_CnSC 字段描述 (继续)

字段	描述
	0 禁用通道中断。 1 使能通道中断。
5 MSB	通道模式选择 用来在通道逻辑中作进一步选择。其功能取决于通道模式。通道禁用时，该字段将不会更改状态，直至 TPM 计数器时钟域中确认。
4 MSA	通道模式选择 用来在通道逻辑中作进一步选择。其功能取决于通道模式。通道禁用时，该字段将不会更改状态，直至 TPM 计数器时钟域中确认。
3 ELSB	边沿或电平选择 ELSB 和 ELSA 的功能取决于通道模式。通道禁用时，该字段将不会更改状态，直至 TPM 计数器时钟域中确认。
2 ELSA	边沿或电平选择 ELSB 和 ELSA 的功能取决于通道模式。通道禁用时，该字段将不会更改状态，直至 TPM 计数器时钟域中确认。
1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
0 DMA	DMA 使能 使能通道的 DMA 传输。 0 禁用 DMA 传输。 1 使能 DMA 传输。

37.4.9 通道 (n) 值寄存器 (TPMx_CnV)

这些寄存器包含输入模式下捕获的 TPM 计数器值或输出模式下的匹配值。

在输入捕捉模式下，将忽略对 CnV 寄存器的写入操作。

在比较模式下，向 CnV 寄存器写入值时，会将此值锁存到缓冲器中。将根据 [CnV 寄存器更新](#) 用 CnV 寄存器的写入缓冲器值更新此 CnV 寄存器。CnV 写入缓冲区的额外写入被忽略，直至寄存器更新。

地址: 基址 + 24h 偏移 + (8d × i), 其中 i=0d 到 5d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0																VAL																
W	0																0																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

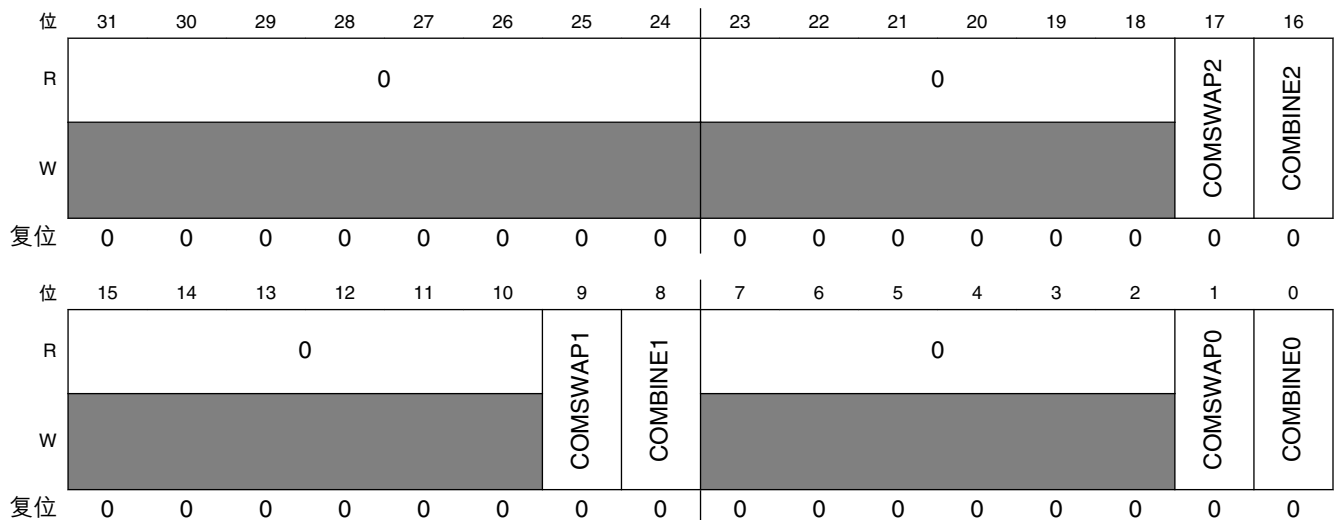
TPMx_CnV 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
VAL	通道值 输入模式下捕捉的 TPM 计数器值或用于输出模式的匹配值。此字段必须写入单 16 位或 32 位访问。

37.4.10 组合通道寄存器 (TPMx_COMBINE)

该寄存器含有用于每对通道 (n) 和 (n+1) 配置组合通道模式的控制位，其中 n 为所有通道的偶数通道值。

地址: 基址 基准 + 64h 偏移



TPMx_COMBINE 字段描述

字段	描述
31-24 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
23-18 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
17 COMSWAP2	组合通道 4 和 5 交换 在组合模式中置位时，偶数通道用于输入捕捉和第 1 次比较，奇数通道用于第 2 次比较。 0 偶数通道用于输入捕捉和第 1 次比较。 1 奇数通道用于输入捕捉和第 1 次比较。
16 COMBINE2	组合通道 4 和 5 使能通道 2 和 3 的组合功能。在输入捕捉模式下，合并的通道使用偶数通道输入。在软件比较模式下，偶数通道匹配时置位输出触发，奇数通道匹配时不置位输出触发。在 PWM 下，偶数通道用于第 1 次比较，奇数通道用于第 2 次比较。

下一页继续介绍此表...

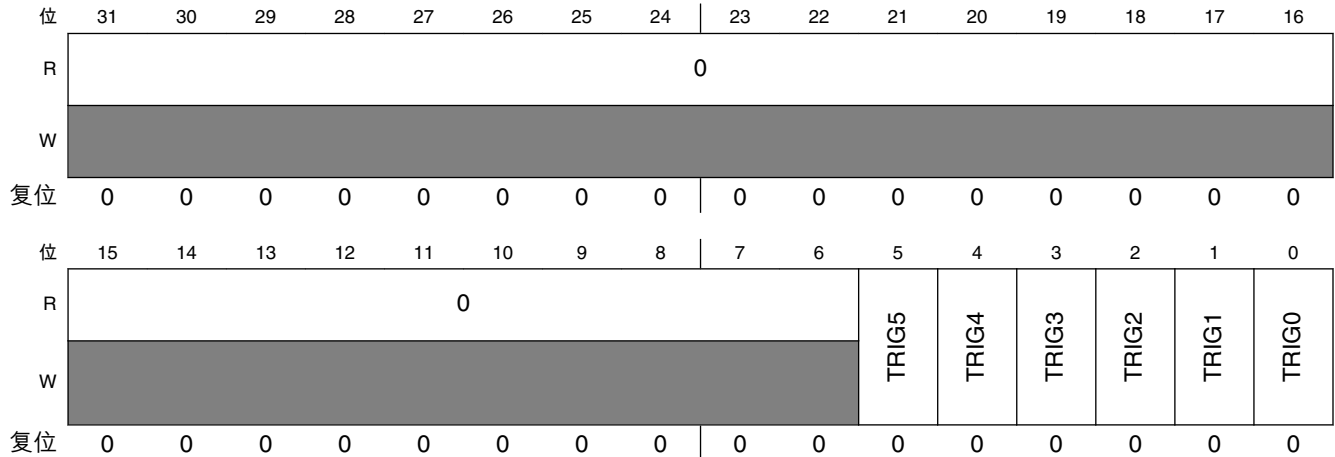
TPMx_COMBINE 字段描述 (继续)

字段	描述
	0 通道 4 和 5 独立。 1 通道 4 和 5 组合。
15-10 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
9 COMSWAP1	组合通道 2 和 3 交换。 在组合模式中置位时，奇数通道用于输入捕捉和第 1 次比较，偶数通道用于第 2 次比较。 0 偶数通道用于输入捕捉和第 1 次比较。 1 奇数通道用于输入捕捉和第 1 次比较。
8 COMBINE1	组合通道 2 和 3 使能通道 2 和 3 的组合功能。在输入捕捉模式下，合并的通道使用偶数通道输入。在软件比较模式下，偶数通道匹配时置位输出触发，奇数通道匹配时不置位输出触发。在 PWM 下，偶数通道用于第 1 次比较，奇数通道用于第 2 次比较。 0 通道 2 和 3 独立。 1 通道 2 和 3 组合。
7-2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 COMSWAP0	组合通道 0 和 1 交换。 在组合模式中置位时，偶数通道用于输入捕捉和第 1 次比较，奇数通道用于第 2 次比较。 0 偶数通道用于输入捕捉和第 1 次比较。 1 奇数通道用于输入捕捉和第 1 次比较。
0 COMBINE0	组合通道 0 和 1 使能通道 0 和 1 的组合功能。在输入捕捉模式下，合并的通道使用偶数通道输入。在软件比较模式下，偶数通道匹配置位输出触发，奇数通道匹配取反输出触发。在 PWM 下，偶数通道匹配时置位输出触发，奇数通道匹配时不置位输出触发。 0 通道 0 和 1 独立。 1 通道 0 和 1 组合。

37.4.11 通道触发 (TPMx_TRIG)

该寄存器为每个通道配置触发输入

地址: 基址 基准 + 6Ch 偏移



TPMx_TRIG 字段描述

字段	描述
31-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 TRIG5	通道 5 触发 0 无效。 1 输入触发用于输入捕捉和模数输出 (用于输出比较和 PWM)。
4 TRIG4	通道 4 触发 0 无效。 1 输入触发用于输入捕捉和模数输出 (用于输出比较和 PWM)。
3 TRIG3	通道 3 触发 0 无效。 1 输入触发用于输入捕捉和模数输出 (用于输出比较和 PWM)。
2 TRIG2	通道 2 触发 0 无效。 1 输入触发用于输入捕捉和模数输出 (用于输出比较和 PWM)。
1 TRIG1	通道 1 触发 0 无效。 1 输入触发用于输入捕捉和模数输出 (用于输出比较和 PWM)。
0 TRIG0	通道 0 触发 0 无效。 1 输入触发用于输入捕捉和模数输出 (用于输出比较和 PWM)。

37.4.12 通道极性 (TPMx_POL)

此寄存器定义每个通道的输入和输出极性。

地址: 基址 基准 + 70h 偏移

位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R	0																
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R	0										POL5	POL4	POL3	POL2	POL1	POL0	
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0

TPMx_POL 字段描述

字段	描述
31-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 POL5	通道 5 极性 0 通道极性为高电平有效。 1 通道极性为低电平有效。
4 POL4	通道 4 极性 0 通道极性为高电平有效 1 通道极性为低电平有效。
3 POL3	通道 3 极性 0 通道极性为高电平有效。 1 通道极性为低电平有效。
2 POL2	通道 2 极性 0 通道极性为高电平有效。 1 通道极性为低电平有效。
1 POL1	通道 1 极性 0 通道极性为高电平有效。 1 通道极性为低电平有效。
0 POL0	通道 0 极性 0 通道极性为高电平有效。 1 通道极性为低电平有效。

37.4.13 滤波器控制 (TPMx_FILTER)

寄存器选择通道输入的滤波器值，以及通道输出的额外输出延迟值。在 PWM 组合模式下，滤波器可以有效实施死区时间插入。

地址: 基址 基准 + 78h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0								CH5FVAL				CH4FVAL				CH3FVAL				CH2FVAL				CH1FVAL				CH0FVAL				
W	0								0				0				0				0				0								
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

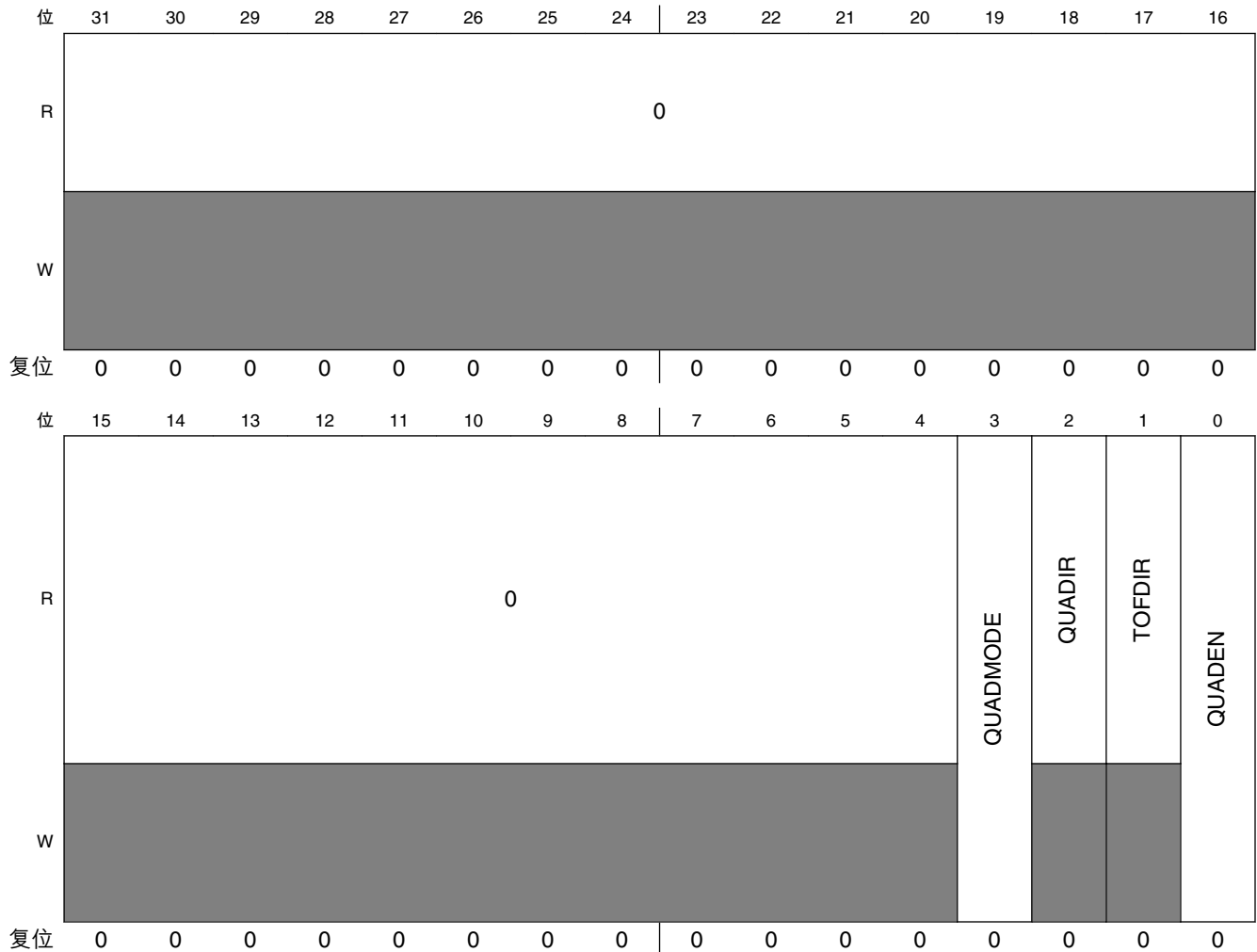
TPMx_FILTER 字段描述

字段	描述
31-24 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
23-20 CH5FVAL	通道 5 滤波器值 选择通道输入滤波器值和通道输出的延迟值。值为 0 时，滤波器/延迟禁用，否则滤波器/延迟配置为 (CH5FVAL * 4) 时钟周期。
19-16 CH4FVAL	通道 4 滤波器值 选择通道输入滤波器值和通道输出的延迟值。值为 0 时，滤波器/延迟禁用，否则滤波器/延迟配置为 (CH4FVAL * 4) 时钟周期。
15-12 CH3FVAL	通道 3 滤波器值 选择通道输入滤波器值和通道输出的延迟值。值为 0 时，滤波器/延迟禁用，否则滤波器/延迟配置为 (CH3FVAL * 4) 时钟周期。
11-8 CH2FVAL	通道 2 滤波器值 选择通道输入滤波器值和通道输出的延迟值。值为 0 时，滤波器/延迟禁用，否则滤波器/延迟配置为 (CH2FVAL * 4) 时钟周期。
7-4 CH1FVAL	通道 1 滤波器值 选择通道输入滤波器值和通道输出的延迟值。值为 0 时，滤波器/延迟禁用，否则滤波器/延迟配置为 (CH1FVAL * 4) 时钟周期。
CH0FVAL	通道 0 滤波器值 选择通道输入滤波器值和通道输出的延迟值。值为 0 时，滤波器/延迟禁用，否则滤波器/延迟配置为 (CH0FVAL * 4) 时钟周期。

37.4.14 正交解码器控制和状态 (TPMx_QDCTRL)

该寄存器含有控制和状态位，可用于正交解码器模式。

地址: 基址 基准 + 80h 偏移



TPMx_QDCTRL 字段描述

字段	描述
31–4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 QUADMODE	正交解码器模式 选择正交解码器模式中所用的编码模式。 0 相编码模式。 1 计数和方向编码模式。
2 QUADIR	正交解码器模式中的计数器方向

下一页继续介绍此表...

TPMx_QDCTRL 字段描述 (继续)

字段	描述
	指示计数方向。 0 计数方向为向下 (计数器递减)。 1 计数方向为向上 (计数器递增)。
1 TOFDIR	指示 TOF 位在最大计数值还是最小计数值时置位。 0 TOF 位在最小时计数值时置位。具有 FTM 计数器减量, FTM 计数器从最小值 (零) 变化到最大值 (MOD 寄存器)。 1 TOF 位在最大计数值时置位。具有 FTM 计数器增量, FTM 计数器从最大值 (MOD 寄存器) 变化到最小值 (零)。
0 QUADEN	使能正交解码器模式。在该模式下, 通道 0 和 1 输入控制 TPM 计数器方向并仅用于软件比较。正交解码器模式优先于其他模式。 0 正交解码器模式禁用。 1 正交解码器模式禁用使能。

37.4.15 配置寄存器 (TPMx_CONF)

寄存器选择调试行为和等待模式, 并选择使用的外部全局时基。

地址: 基址 基准 + 84h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				TRGSEL				TRGSRC	TRGPOL	0		CROT	CROT	CROT	CROT
W	■				■				■	■	■		■	■	■	■
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				GTBEEN				GTBSYNC		DBGMODE		DOZEEN		0	
W	■				■				■		■		■		■	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_CONF 字段描述

字段	描述
31-28 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
27-24 TRGSEL	触发选择 选择输入触发用来启动、重载"与"或者"或"暂停计数器。触发源 (TPM 外部或内部) 由 TRGSRC 字段配置。仅 TPM 计数器禁用时该字段才能更改。

下一页继续介绍此表...

TPMx_CONF 字段描述 (继续)

字段	描述
	<p>有关可用外部触发选项，请参见芯片配置章节。</p> <p>下方列出了可用的内部触发源。</p> <p>0001 通道 0 引脚输入捕捉 0010 通道 1 引脚输入捕捉 0011 通道 0 或 1 引脚输入捕捉 0100 通道 2 引脚输入捕捉 0101 通道 0 或 2 引脚输入捕捉 0110 通道 1 或 2 引脚输入捕捉 0111 通道 0、1 或 2 引脚输入捕捉 1000 通道 3 引脚输入捕捉 1001 通道 0 或 3 引脚输入捕捉 1010 通道 1 或 3 引脚输入捕捉 1011 通道 0、1 或 3 引脚输入捕捉 1100 通道 2 或 3 引脚输入捕捉 1101 通道 0、2 或 3 引脚输入捕捉 1110 通道 1、2 或 3 引脚输入捕捉 1111 通道 0、1、2 或 3 引脚输入捕捉</p>
23 TRGSRC	<p>触发源</p> <p>选择内部 (通道引脚输入捕捉) 或外部触发源</p> <p>选择内部触发时, 选定的通道应配置用于输入捕捉。仅上升沿输入捕捉可用来初始化启动使用 CSOT 配置的计数器; 上升沿或下降沿输入捕捉均可用来重载使用 CROT 配置的计数器; 且通道输入引脚的状态可用来暂停使用 CPOT 配置的计数器。通道极性寄存器可用来转换通道输入引脚的极性。</p> <p>仅 TPM 计数器禁用时该字段才能更改。</p> <p>0 TRGSEL 选择外部触发源。 1 TRGSEL 选择外部触发源 (通道引脚输入捕捉)。</p>
22 TRGPOL	<p>触发器极性</p> <p>选择外部触发源的极性。仅 TPM 计数器禁用时该字段才能更改。</p> <p>0 触发的有效电平高。 1 触发的有效电平低。</p>
21-20 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
19 CPOT	<p>计数器暂停触发</p> <p>使能后计数器在触发保持置位时 (电平敏感) 将暂停递增。仅 TPM 计数器禁用时该字段才能更改。</p>
18 CROT	<p>计数器重载触发</p> <p>置位后, 当在选定触发输入上检测到上升沿时, TPM 计数器将重载为 0 (并将 PWM 输出初始化为默认值)。 若 TPM 计数器在调试模式或休眠模式中暂停, 则触发输入被忽略。仅 TPM 计数器禁用时该字段才能更改。</p> <p>0 由于选定的输入触发的上升沿, 计数器没有重载。 1 由于选定的输入触发检测到上升沿, 计数器重载。</p>
17 CSOO	<p>计数器溢出停止</p>

下一页继续介绍此表...

TPMx_CONF 字段描述 (继续)

字段	描述
	<p>置位时，TPM 计数器在等于 MOD 值和递增完成后将停止递增（同样置位 TOF）。因写入计数器寄存器或触发输入而导致计数器重载为 0 并不会导致计数器停止递增。计数器一旦停止递增，直到被禁用后重新使能，或在 CSOT 置位时检测到选定触发输入的上升沿，计数器才会开始递增。</p> <p>仅 TPM 计数器禁用时该字段才能更改。</p> <p>0 TPM 溢出后继续递增或递减。 1 TPM 溢出后停止递增或递减。</p>
16 CSOT	<p>计数器触发开始</p> <p>置位时，TPM 计数器使能后不会开始递增，直至在选定触发输入上检测到上升沿。如果 TPM 计数器因为溢出而停止，则选定触发输入上的上升沿也会导致 TPM 计数器重新开始递增。</p> <p>若 TPM 计数器在调试模式或休眠模式中暂停，则触发输入被忽略。仅 TPM 计数器禁用时该字段才能更改。</p> <p>0 TPM 计数器一旦使能便立即开始递增。 1 使能或因溢出停止后，TPM 计数器仅在选定输入触发检测到上升沿时开始递增，</p>
15-10 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
9 GTBEEN	<p>启用全局时基</p> <p>配置 TPM 以使用外部生成的全局时基计数器。使用外部生成的时基时，内部 TPM 计数器不能用于通道，但可使用模数寄存器和定时器溢出标志生成周期性中断或 DMA 请求。</p> <p>0 所有通道使用内部生成的 TPM 计数器作为其时基 1 所有通道使用外部生成的全局时基作为其时基</p>
8 GTBSYNC	<p>全局时基同步</p> <p>使能后，TPM 计数器同步至全局时基。它使用全局时基使能、触发和溢出来确保 TPM 计数器与全局时基同时开始和停止递增以及复位。仅 TPM 计数器禁用时该字段才能更改。</p> <p>0 全局时基同步禁用。 1 全局时基同步使能。</p>
7-6 DBGMODE	<p>调试模式</p> <p>在调试模式中配置 TPM 行为。所有其他配置均保留。</p> <p>00 TPM 计数器在调试模式下暂停且不递增。同时忽略触发输入和输入捕捉事件。 11 TPM 计数器在调试模式中继续。</p>
5 DOZEEN	<p>使能休眠</p> <p>在等待模式中配置 TPM 行为。</p> <p>0 内部 TPM 计数器在休眠模式中继续。 1 内部 TPM 计数器在休眠模式中暂停且不递增。同时忽略触发输入和输入捕捉事件。</p>
保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>

37.5 功能说明

以下章节介绍 TPM 特性。

37.5.1 时钟域

TPM 模块支持两个时钟域。

寄存器接口使用总线时钟域，且总线时钟域用于中断和 DMA 请求。

TPM 计数器时钟域用作计数器和预分频器时钟，以及输出比较和输入捕捉逻辑。TPM 计数器时钟被认为与总线时钟异步，频率可能高于或低于总线时钟，在停止模式下也可保持运行。多个 TPM 模块均使用同一个 TPM 计数时钟源，以支持外部时钟功能。

37.5.1.1 计数器时钟模式

SC 寄存器中的 CMOD[1:0]位可以禁用 TPM 计数器或者从两种可能的时钟模式中选择一种。复位后，CMOD[1:0] = 0:0，因此 TPM 计数器已禁用。

可在任意时间对 CMOD[1:0]位进行读取或写入操作。通过在 CMOD[1:0] 位写入 0 来禁用 TPM 计数器并不会影响 TPM 计数器值或其他寄存器，但必须在读取为 0 前被 TPM 计数器时钟域确认。

外部时钟输入穿过同步器被 TPM 计数器时钟同步，以确保外部计数器转换能与计数器时钟转换正确对齐。因此，考虑到抖动因素，如需满足奈奎斯特标准，外部时钟源频率必须低于计数器时钟频率的一半。

37.5.2 预分频器

所选的计数器时钟源会经过一个预分频器，该预分频器是一个 7 位计数器。

由 PS[2:0]位选择预分频器的值。下图给出了预分频器计数器和 TPM 计数器的示例。

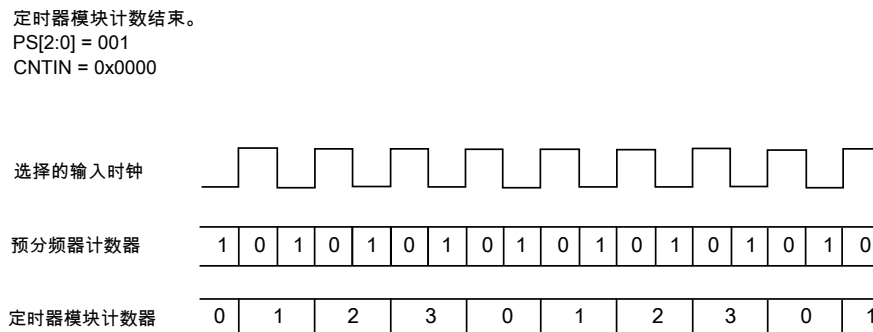


图 37-110. 预分频器计数器示例

37.5.3 计数器寄存器

TPM 有一个 16 位计数器，由通道用于输入或输出模式。

计数器由选定的时钟经过预分频器预分频进行更新。

TPM 计数器有这些工作模式：

- 向上计数（请参见[向上计数](#)）
- 向上-向下计数（请参见[向上-向下计数](#)）

37.5.3.1 向上计数

SC[CPWMS] = 0 时会选择向上计数。

零值加载到 TPM 计数器中，计数器的值递增，直至达到 MOD 的值，然后计数器将重新加载零值。

采用向上计数时的 TPM 周期为 $(MOD + 0x0001) \times$ TPM 计数器时钟的周期。

TPM 计数器从 MOD 变为 0 时，TOF 位将置位。

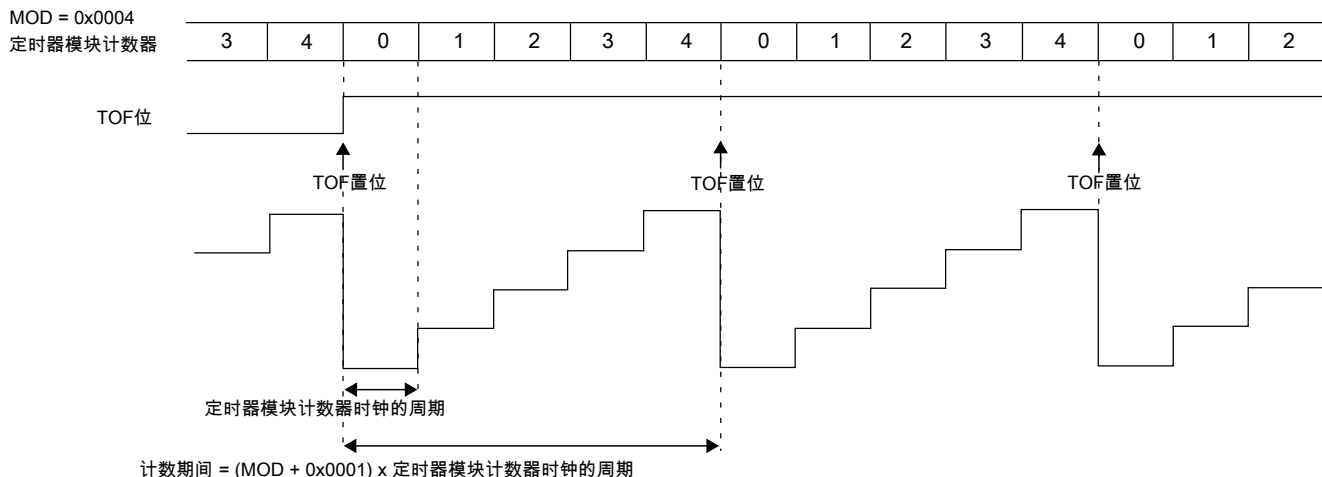


图 37-111. TPM 向上计数的示例

注

- MOD = 0000 是一个冗余条件。这种情况下，TPM 计数器始终等于 MOD，将在 TPM 计数器时钟的每一个上升沿置位 TOF 位。

37.5.3.2 向上-向下计数

SC[CPWMS] = 1 时选择向上-向下计数。在针对向上-向下计数进行配置时，不支持将 CONF[MOD]配置为 2 以下。

0 值加载到 TPM 计数器中，计数器的值一直增加，直至达到 MOD 的值，紧接着计数器的值一直减少，直至回到 0 值，然后计数器将重新开始自上而下计数。

采用向上-向下计数时的 TPM 周期为 $2 \times \text{MOD} \times \text{TPM}$ 计数时钟的周期。

TPM 计数器从 MOD 更改为 (MOD - 1) 时，TOF 位将被置位。

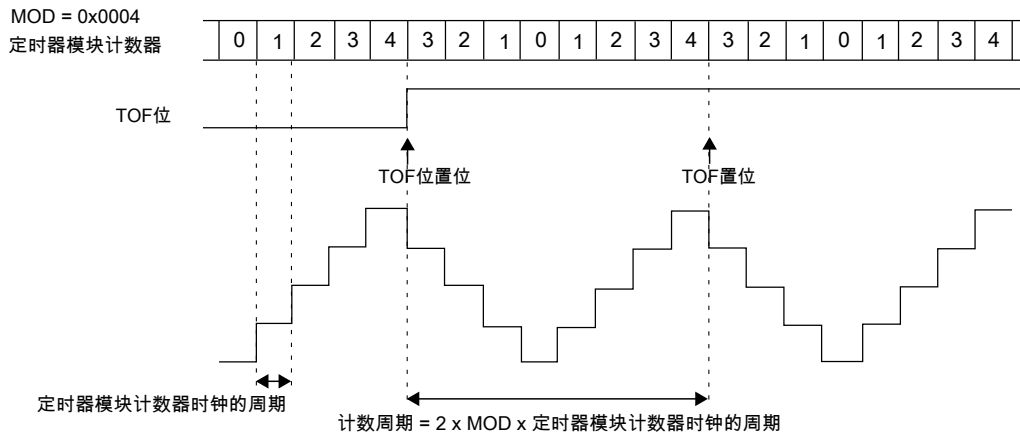


图 37-112. 向上-向下计数示例

37.5.3.3 计数器复位

对 CNT 的任意写操作都会把 TPM 计数器和通道输出复位至其初始值（输出比较模式下的通道除外）。

37.5.3.4 全局时基(GTB)

全局时基 (GTB) 是一种 TPM 功能，允许多个 TPM 模块共享相同的时基。全局时基使能时 (CONF[GTBEEN] = 1)，本地 TPM 通道使用 TPM 计数器值、计数器使能和溢出指示来生成全局时基。若本地 TPM 计数器未生成全局时基，则计数器可用作独立的计数器或脉冲累加器。

本地 TPM 计数器还可通过配置 (GTBSYNC = 1) 来同步至全局时基。同步至全局时基时，本地计数器将使用 TPM 的计数器使能和溢出指示来生成全局时基。这将使能多个 TPM 以相同的相位和不同的周期进行配置（其中全局时基必须配置最长的周期）。

37.5.3.5 计数器触发

TPM 计数器可配置为根据硬件触发输入进行启动、停止或复位。触发输入同步至异步计数器时钟，因此在触发置位和计数器响应之间有 3 个计数器时钟延迟。

- CSOT = 1 时，计数器不会开始递增，直至在触发输入上检测到上升沿。
- CSOO = 1 时，计数器将在 TOF 标志置位时停止递增。仅当计数器禁用，或 CSOT = 1 且在触发输入上检测到上升沿时，计数器才会再次递增。
- CROT = 1 时，计数器将在触发输入上检测到上升沿时复位为 0（同发生溢出的情况）。
- CPOT = 1 时，计数器将在触发输入置位时暂停递增。计数器将在触发输入取消时继续递增。

外部输入触发极性可通过 TRGPOL 寄存器位进行配置。

内部触发源选定时，触发输入从一个或多个通道输入捕捉事件中选择。输入捕捉滤波器与内部触发源一起使用，POLn 位则用来逆转输入通道的极性。需要注意的是，下列限制适用于输入捕捉通道源。

- CSOT = 1 时，假定 ELSnA = 1，计数器将在通道输入的上升沿开始递增。
- CROT = 1 时，计数器将根据 ELSnB:ELSnA 配置在通道输入的上升沿和下降沿复位为 0。
- CPOT = 1 时，计数器将在通道输入置位时暂停递增。

37.5.4 输入捕捉模式

CPWMS = 0、MSnB:MSnA = 0:0 和 ELSnB:ELSnA ≠ 0:0 时，输入捕捉模式选定。

通道输入中发生所选边沿时，TPM 计数器的当前值将会被捕捉到 CnV 寄存器中，同时还会将 CHnF 位置位并生成通道中断（如果已通过 CHnIE = 1 使能），参见下图。

如果某个通道已配置为输入捕捉模式，则 TPM_CHn 引脚为边沿敏感的输入。ELSnB:ELSnA 控制位可确定是哪个边沿（上升沿或下降沿）触发输入捕捉事件。注意，可以被正确检测到通道输入信号的最大频率为计数器时钟的四分之一，这是满足信号采样的奈奎斯特准则所必需的条件。

在输入捕捉模式下，将忽略对 CnV 寄存器的写入操作。

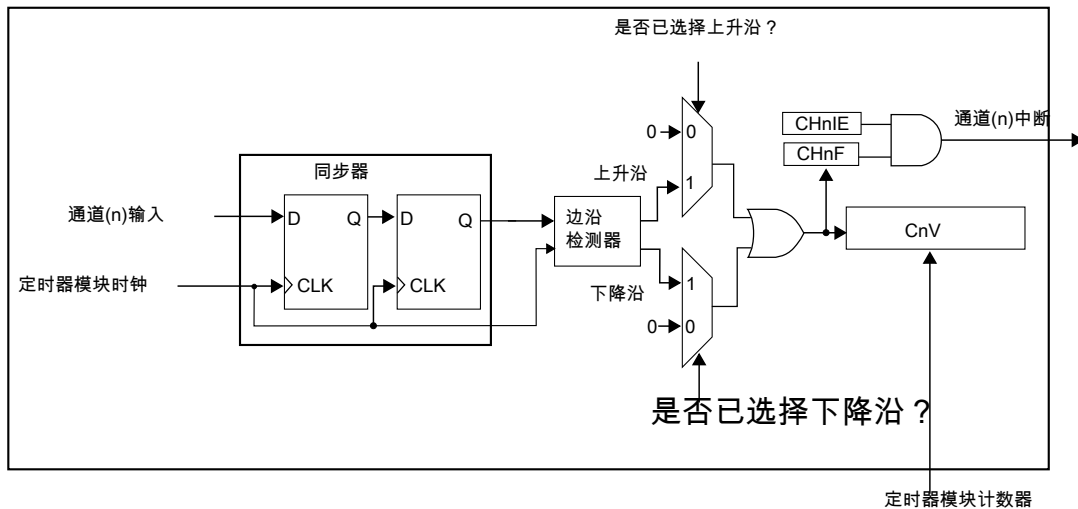


图 37-113. 输入捕捉模式

在通道输入中发生有效边沿后，CHnF 位在计数器时钟的第三个上升沿上置位。

37.5.5 输出比较模式

在 CPWMS = 0 和 MSnB:MSnA = X:1 的情况下会选择输出比较模式。

在输出比较模式下，TPM 可生成具有可编程位置、极性、持续时间和频率的定时脉冲。当计数器与某个输出比较通道的 CnV 寄存器中的值匹配时，若 MSnB 清零，可以设置、清除或翻转通道 (n) 输出。若 MSnB 置位，在计数器值与 CnV 寄存器中的值匹配时，则通道 (n) 输出脉冲为高电平或低电平。

通道最初配置为输出比较模式时，通道输出使用否定值进行更新（逻辑 0 为置位/翻转/脉冲高电平，逻辑 1 为清空/脉冲低电平）。

通道 (n) 匹配 (TPM 计数器 = CnV 时，CHnF 位将置位并生成通道 (n) 中断（如果 CHnIE = 1））。

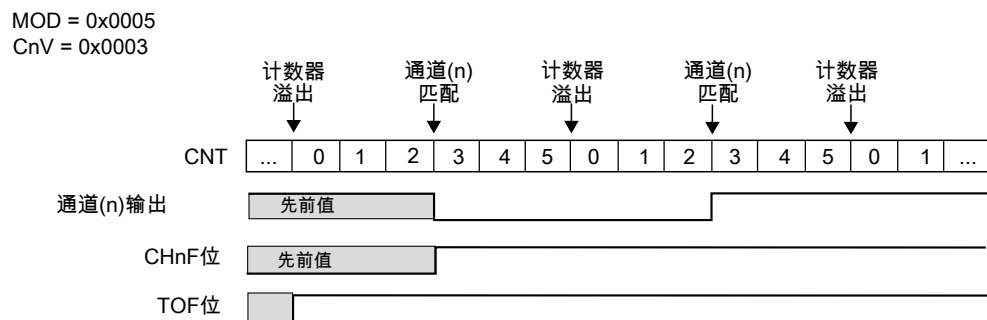


图 37-114. 匹配翻转通道输出时的输出比较模式示例

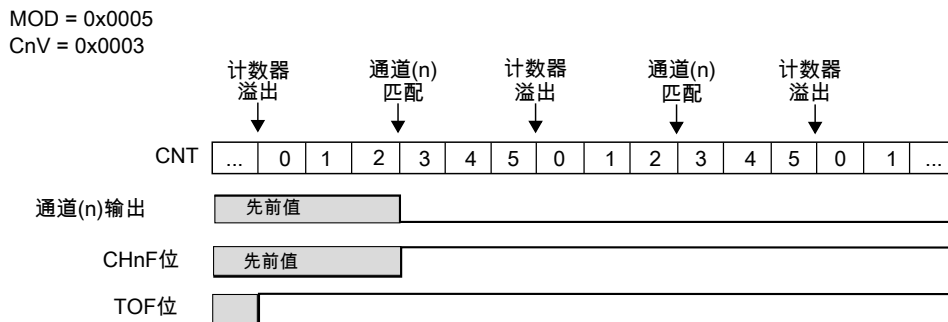


图 37-115. 匹配清空通道输出时的输出比较模式示例

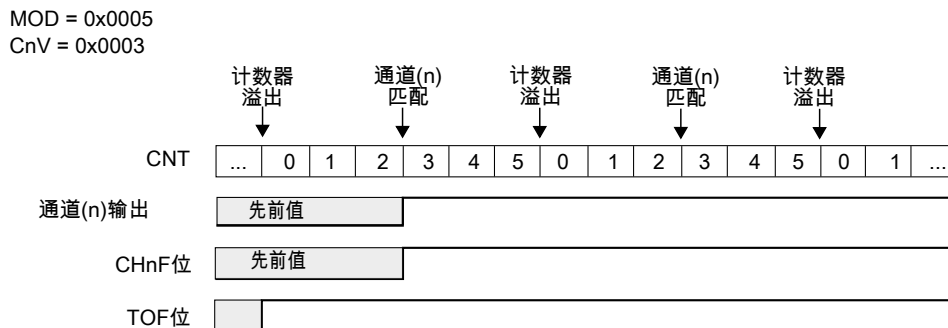


图 37-116. 匹配置位通道输出时的输出比较模式示例

可以使用输出比较模式 (ELSnB:ELSnA = 0:0)。在这种情况下，如果计数器达到 CnV 寄存器中的值，则 CHnF 位置位并生成通道 (n) 中断 (如果 CHnIE = 1)，但是，通道 (n) 输出不由 TPM 修改和控制。

37.5.6 边沿对齐 PWM (EPWM) 模式

在 CPWMS = 0 和 MSnB:MSnA = 1:0 的情况下会选择边沿对齐模式。

EPWM 周期取决于 (MOD + 0x0001)，脉宽 (占空比) 取决于 CnV。

通道 (n) 匹配 (TPM 计数器 = CnV) (即脉宽结束) 时，CHnF 位将置位并生成通道 (n) 中断 (如果 CHnIE = 1)。

这种类型的 PWM 信号称为边沿对齐，因为所有 PWM 信号的前沿都与周期的开始对齐，这对 TPM 内的所有通道都一样。

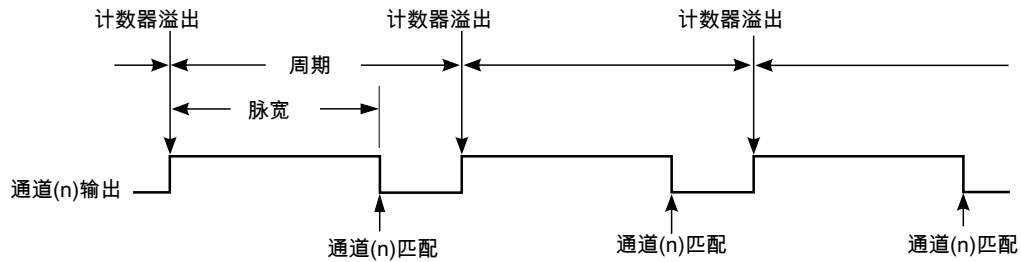


图 37-117. ELSnB:ELSnA = 1:0 条件下的 EPWM 周期和脉宽

如果计数器达到 CnV 寄存器中的值时满足 (ELSnB:ELSnA = 0:0) 条件, 则 CHnF 位置位并生成通道 (n) 中断 (如果 CHnIE = 1), 但是, 通道 (n) 输出不由 TPM 控制。

如果 (ELSnB:ELSnA = 1:0), 那么, 通道 (n) 输出会在计数器溢出 (0 加载到 TPM 计数器) 时强制为高电平, 在通道 (n) 匹配 (TPM 计数器 = CnV) 时强制为低电平 (请参见下图)。

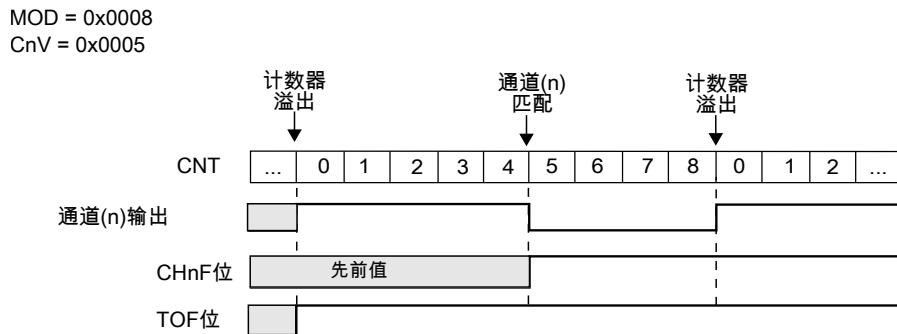


图 37-118. ELSnB:ELSnA = 1:0 条件下的 EPWM 信号

如果 (ELSnB:ELSnA = X:1), 那么, 通道 (n) 输出会在计数器溢出 (0 加载到 TPM 计数器) 时强制为低电平, 在通道 (n) 匹配 (TPM 计数器 = CnV) 时强制为高电平 (请参见下图)。

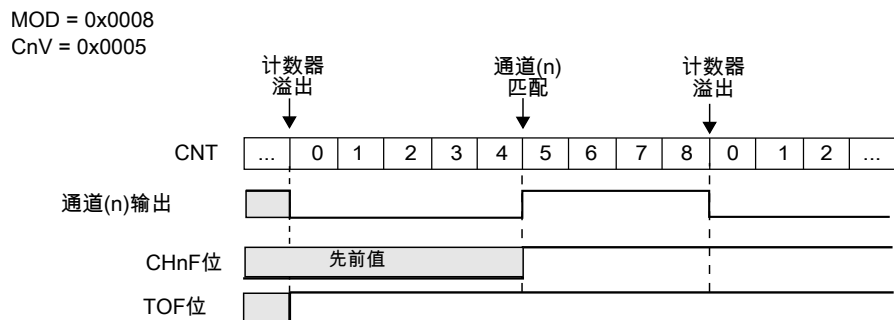


图 37-119. ELSnB:ELSnA = X:1 条件下的 EPWM 信号

如果 ($CnV = 0x0000$)，则通道 (n) 输出为 0% 占空比 EPWM 信号。如果 ($CnV > MOD$)，则通道 (n) 输出为 100% 占空比 EPWM 信号，而且 $CHnF$ 位不置位，因为不存在通道 (n) 匹配。因此，要获得 100% 占空比 EPWM 信号， MOD 必须小于 $0xFFFF$ 。

37.5.7 中心对齐 PWM (CPWM) 模式

在 ($CPWMS = 1$) 和 ($MSnB:MSnA = 1:0$) 的情况下会选择中心对齐模式。

CPWM 脉宽 (占空比) 取决于 $2 \times CnV$ ，周期取决于 $2 \times MOD$ (请参见下图)。MOD 必须保持在 $0x0001$ 到 $0x7FFF$ 范围内，因为此范围以外的值会产生不明确的结果。

在 CPWM 模式下，TPM 计数器在达到 MOD 之前会一直向上计数，然后一直向下计数，直至达到 0。

当 TPM 向下计数 (脉宽开始) 和 TPM 向上计数 (脉宽结束) 时，会在通道 (n) 匹配 (TPM 计数器 = CnV) 情况下置位 $CHnF$ 位并生成通道 (n) 中断。

这种类型的 PWM 信号称为中心对齐，因为 TPM 计数器为 0 时，所有通道的脉宽中心都对齐。

其他通道模式与自上而下的计数器不同时使用 ($CPWMS = 1$)。因此，当 ($CPWMS = 1$) 时，必须在 CPWM 模式下使用所有 TPM 通道。

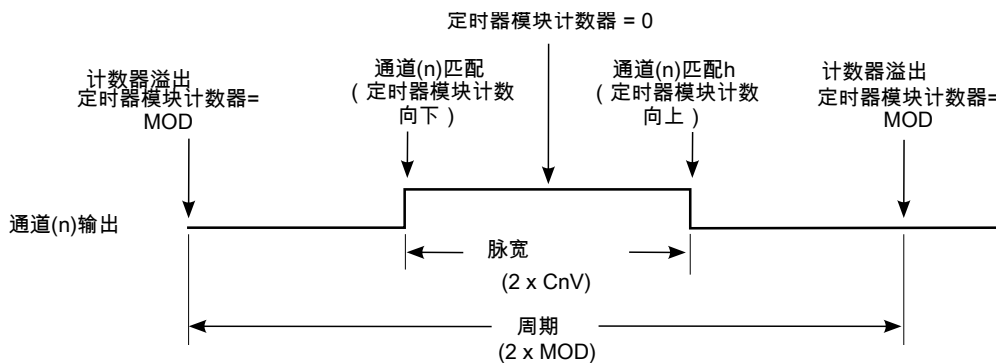


图 37-120. $ELSnB:ELSnA = 1:0$ 条件下的 CPWM 周期和脉宽

如果 TPM 计数器达到 CnV 寄存器中的值时满足 ($ELSnB:ELSnA = 0:0$) 条件，则 $CHnF$ 位置位并生成通道 (n) 中断 (如果 $CHnIE = 1$)，但是，通道 (n) 输出不由 TPM 控制。

如果 ($ELSnB:ELSnA = 1:0$)，那么，通道 (n) 输出会在向下计数时与通道 (n) 匹配 (TPM 计数器 = CnV) 的情况下强制为高电平，在向上计数时与通道 (n) 匹配的情况下强制为低电平 (请参见下图)。

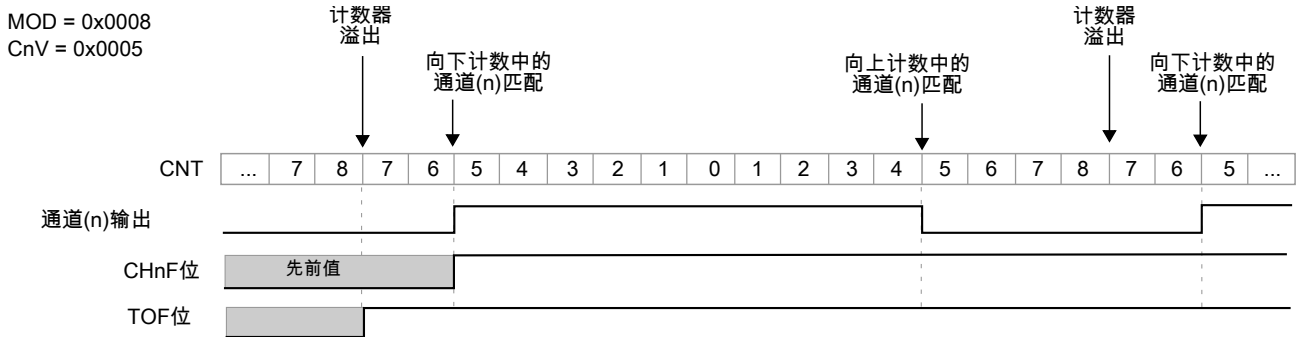


图 37-121. ELSnB:ELSnA = 1:0 条件下的 CPWM 信号

如果 (ELSnB:ELSnA = X:1)，那么，通道 (n) 输出会在向下计数时与通道 (n) 匹配 (TPM 计数器 = CnV) 的情况下强制为低电平，在向上计数时与通道 (n) 匹配的情况下强制为高电平 (请参见下图)。



图 37-122. ELSnB:ELSnA = X:1 条件下的 CPWM 信号

如果 (CnV = 0x0000)，则通道 (n) 输出为 0% 占空比 CPWM 信号。

如果 (CnV > MOD)，虽然 CHnF 位在计数器从递增变为递减时置位，但通道 (n) 输出为 100% 占空比 CPWM 信号。因此，要获得 100% 占空比 CPWM 信号，MOD 必须小于 0xFFFF。

37.5.8 组合 PWM 模式

在以下情形中会选择组合 PWM 模式：

- MSnB:MSnA = 10
- COMBINEn = 1
- QUADEN = 0，且
- CPWMS = 0

在组合 PWM 模式下，一个偶数通道 (n) 和相邻的奇数通道 (n+1) 组合起来生成一个 PWM 信号供通道 (n) 输出。

在组合模式下，PWM 周期取决于 $(MOD + 0x0001)$ ，PWM 脉宽（占空比）取决于 $(IC(n+1)V - C(n)V)$ 。

通道 (n) 匹配 (TPM 计数器 = $C(n)V$) 时， CH_nF 位将置位并生成通道 (n) 中断 (如果 $CH_nIE = 1$)。通道 (n+1) 匹配 (TPM 计数器 = $C(n+1)V$) 时， $CH_{(n+1)}F$ 位将置位并生成通道 (n+1) 中断 (如果 $CH_{(n+1)}IE = 1$)。

如果通道 (n) ($ELSnB:ELSnA = X:1$)，则在周期开始 (TPM 计数器为 0)、通道(n+1)匹配 (TPM 计数器 = $C(n+1)V$) 时强制通道 (n) 输出为低电平。在通道 (n) 匹配 (TPM 计数器 = $C(n)V$) 时强制为高电平。

如果通道 (n) ($ELSnB:ELSnA = 1:0$)，则在周期开始 (TPM 计数器为 0)、通道(n+1)匹配 (TPM 计数器 = $C(n+1)V$) 时强制通道 (n) 输出为高电平。在通道 (n) 匹配 (TPM 计数器 = $C(n)V$) 时强制为低电平。

($COMSWAP_n = 1$)时，则在周期开始 (TPM 计数器为 0)、通道 (n) 匹配 (TPM 计数器 = $C(n)V$) 时强制通道 (n) 输出为低电平或高电平。在通道(n+1)匹配 (TPM 计数器 = $C(n+1)V$) 时强制为高电平或低电平。

生成的通道(n+1)输出与通道 (n) 相同，但输出极性由通道(n+1) $ELSnB:ELSnA$ 配置控制。

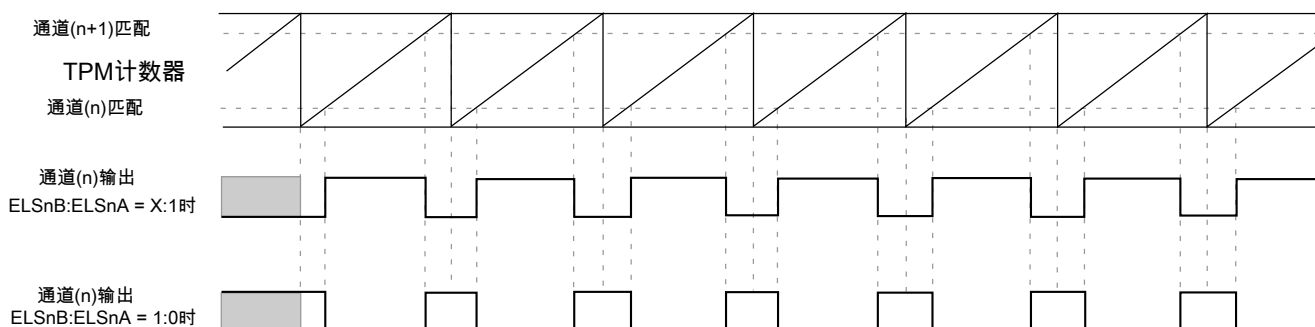


图 37-123. 组合模式

下图展示了利用组合模式生成 PWM 信号的过程。

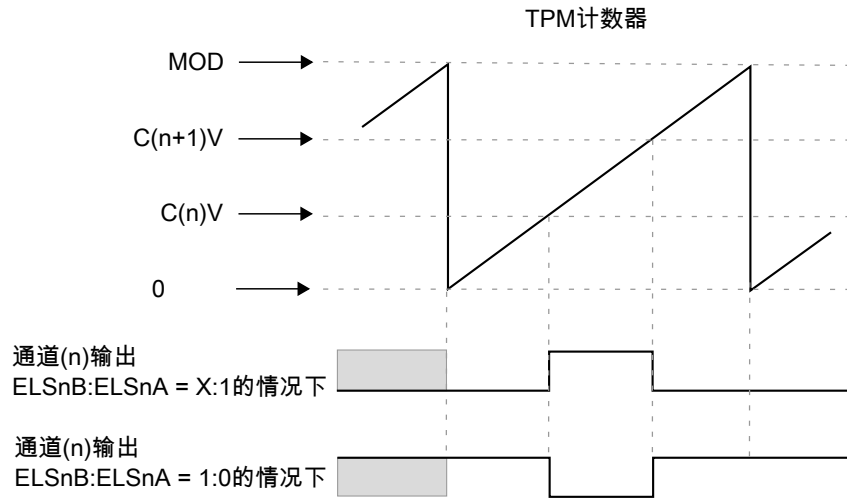


图 37-124. $(C(n)V < MOD)$ 、 $(C(n+1)V < MOD)$ 且 $(C(n)V < C(n+1)V)$ 条件下的通道 (n) 输出

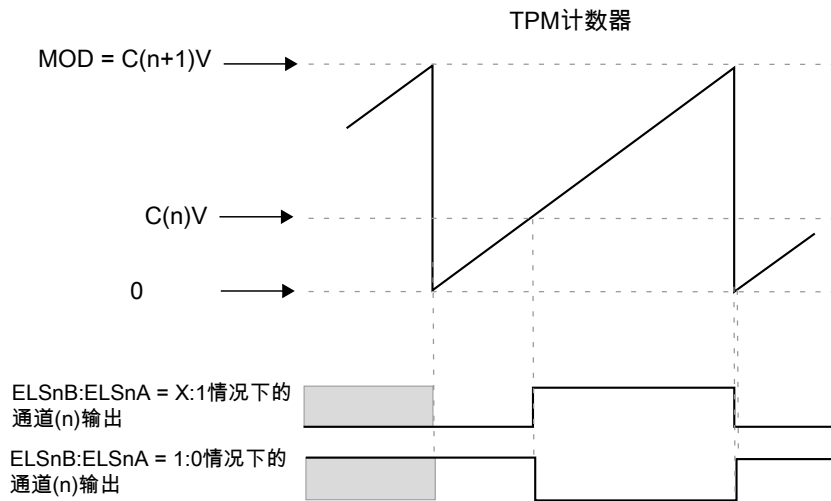


图 37-125. $(C(n)V < MOD)$ 且 $(C(n+1)V = MOD)$ 条件下的通道 (n) 输出

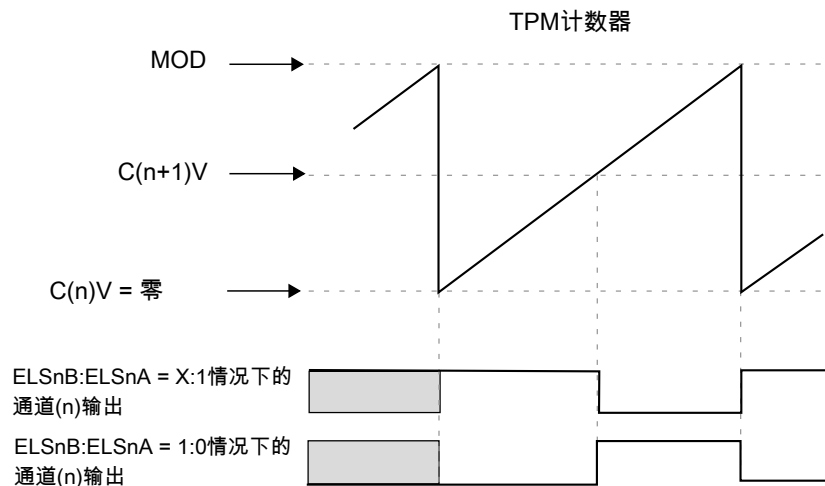


图 37-126. $(C(n)V = 0)$ 且 $(C(n+1)V < MOD)$ 条件下的通道 (n) 输出

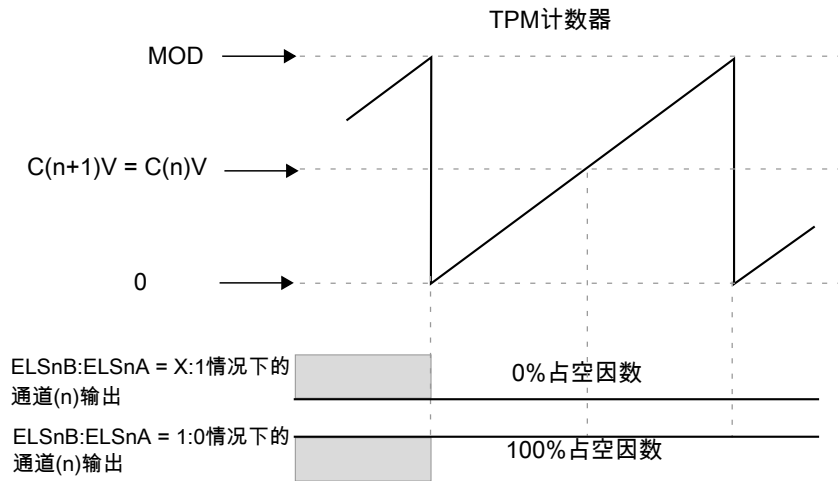


图 37-127. $(C(n)V < MOD)$ 、 $(C(n+1)V < MOD)$ 且 $(C(n)V = C(n+1)V)$ 条件下的通道 (n) 输出

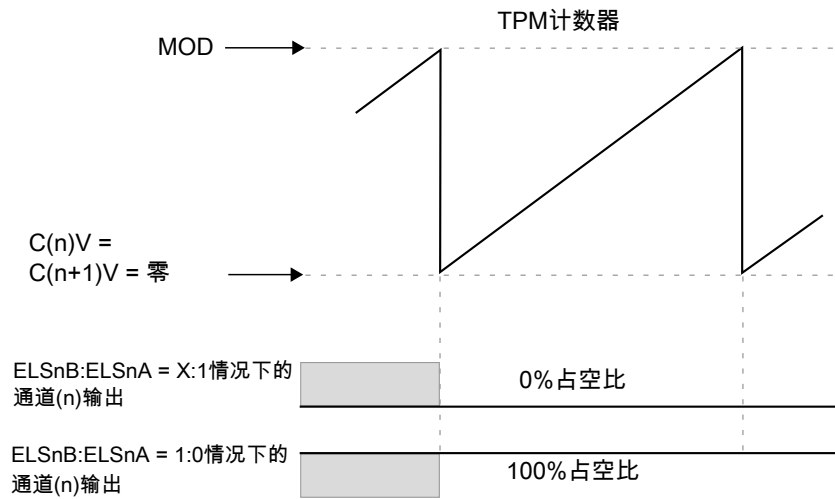


图 37-128. $(C(n)V = C(n+1)V = 0)$ 条件下的通道 (n) 输出

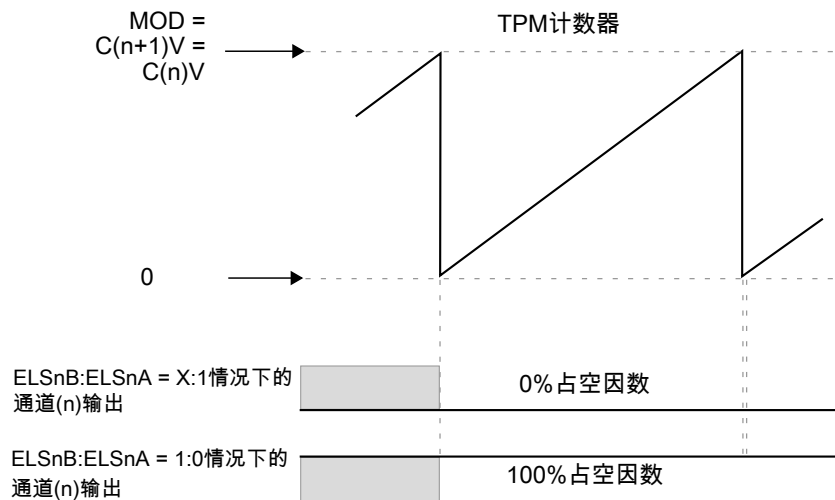


图 37-129. $(C(n)V = C(n+1)V = MOD)$ 条件下的通道 (n) 输出

37.5.9 组合输入捕捉模式

COMBINEn= 1、MSnB:MSnA = 00 且 ELSnB:ELSnA != 00 条件下选择组合输入捕捉模式。此模式允许在一对通道中通道 (n) 的输入端测量脉宽或信号。在此模式下，通道 (n) 滤波器可处于有效状态。

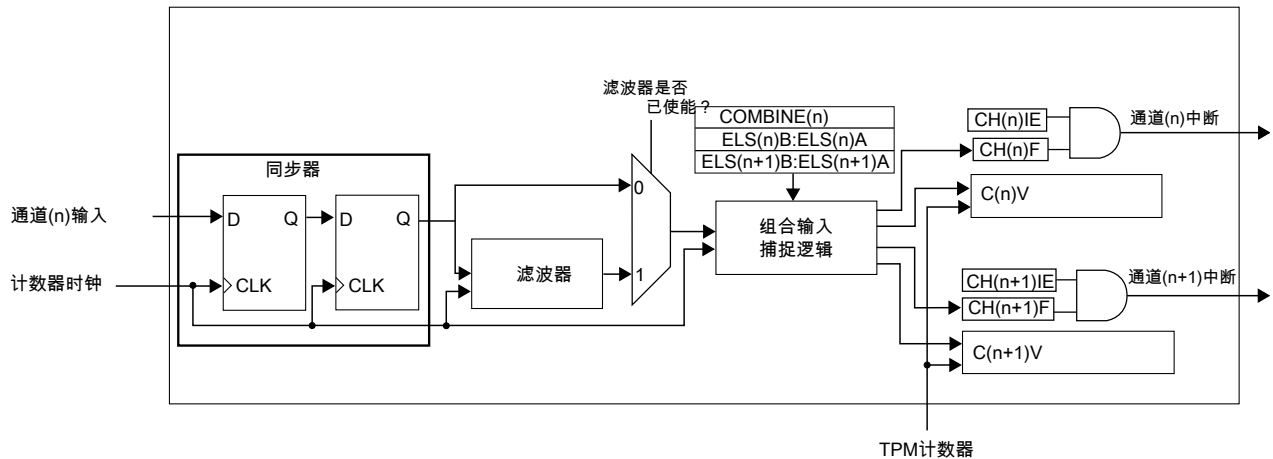


图 37-130. 组合输入捕捉模式结构框图

ELS(n)B:ELS(n)A 位选择由通道 (n) 捕捉的边沿，ELS(n+1)B:ELS(n+1)A 位选择由通道(n+1)捕捉的边沿。

在组合输入捕捉模式下，仅使用通道 (n) 输入，通道(n+1)输入被忽略。当 COMSWAPn=1 时，仅使用通道(n+1)输入，通道 (n) 输入被忽略。

如果在通道 (n) 输入端检测到由通道 (n) 位选择的边沿，则 CH(n)F 位置位并生成通道 (n) 中断（如果 CH(n)IE = 1）。如果在通道 (n) 输入端检测到由通道(n+1)位选择的边沿，则 CH(n+1)F 位置位并生成通道(n+1)中断（如果 CH(n+1)IE = 1）。

C(n)V 寄存器存储了在通道 (n) 输入端检测到通道 (n) 所选边沿时 TPM 计数器的值。C(n+1)V 寄存器存储了在通道 (n) 输入端检测到通道 (n+1) 所选边沿时 TPM 计数器的值。

注

- CH(n)F、CH(n)IE、MS(n)A、ELS(n)B 和 ELS(n)A 位是通道 (n) 位。
- CH(n+1)F、CH(n+1)IE、MS(n+1)A、ELS(n+1)B 和 ELS(n+1)A 位是通道(n+1)位。
- 必须在 ELS(n)B:ELS(n)A = 0:1 或 1:0、ELS(n+1)B:ELS(n+1)A = 0:1 或 1:0 的情况下使用组合输入捕捉模式。

37.5.10 输入捕捉滤波器

输入捕捉滤波器仅出现在输入捕捉模式，或在正交解码器模式使能的情况下出现在软件比较模式中。

首先，由计数器时钟同步输入信号。同步之后，输入信号进入滤波器区块。参见下图。

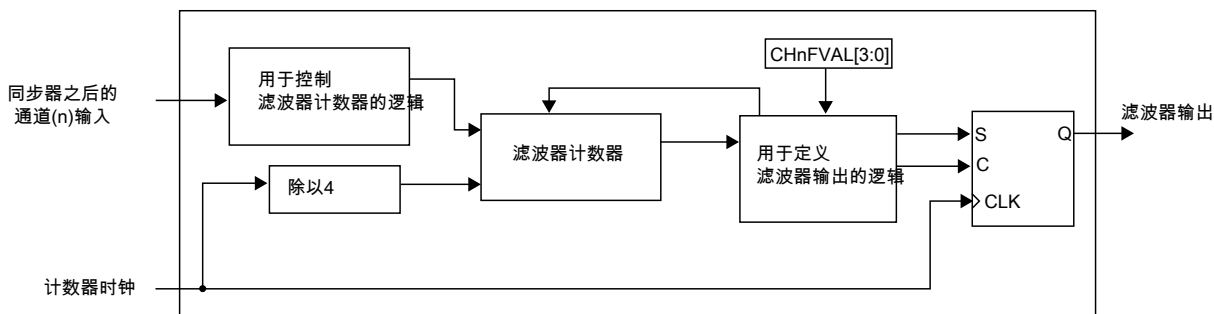


图 37-131. 通道输入滤波器

当输入信号中发生状态更改时，计数器将复位并开始向上计数。只要新状态在输入端保持稳定，计数器就会继续增加数值。当计数器等于 $(CHnFVAL[3:0] \times 4)$ 时，输入信号的状态更改得以验证。

如果在得以验证之前相对边沿出现在输入信号上，计数器就会复位。下一次输入转换时，计数器再次开始计数。任何比 $(CHnFVAL[3:0] \times 4)$ 个计数器时钟所选的最小值短的脉冲都会被认作是毛刺，不会穿过滤波器。下图为输入滤波器的时序图。

$CHnFVAL[3:0]$ 位为零时，将禁用滤波器功能。这种情况下，输入信号将按计数器时钟的 2 个上升沿延迟。如果 $(CHnFVAL[3:0] \neq 0000)$ ，则输入信号按最小脉宽 $(CHnFVAL[3:0] \times 4)$ 个系统时钟延迟，外加系统时钟的 3 个上升沿：两个上升沿到同步器，另一个到边沿检测器。换言之，有效边沿出现在通道输入端之后， $CHnF$ 将在 $(3+4 \times CHnFVAL[3:0])$ 个计数器时钟周期后被置位。

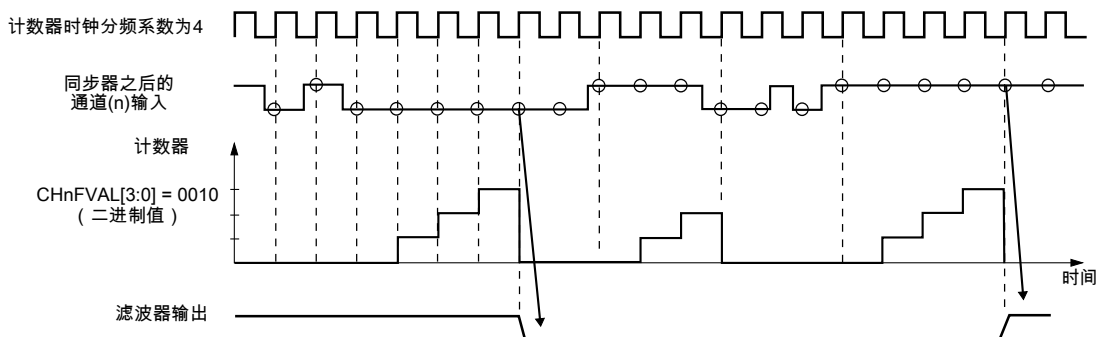


图 37-132. 通道输入滤波器示例

37.5.11 死区时间插入

当 CHnFVAL 为非零时,死区时间插入在 PWM 组合模式下使能。用于各 TPM 通道的死区延迟定义为(CHnFVAL[3:0] x 4)。

死区延迟插入可确保不会有两个互补信号 (通道(n)和(n+1)) 同时驱动有效状态。

如果 $POL(n) = 0$ 、 $POL(n+1) = 1$ 且死区处于使能状态,那么在出现通道(n)匹配 (TPM 计数器 = $C(n)V$) 情况时,通道(n)输出将保持低值,直到通道(n)输出置位、死区延迟结束时。与此类似,在出现通道(n+1)匹配 (TPM 计数器 = $C(n+1)V$) 情况时,通道(n+1)输出将保持低值,直到通道(n+1)输出置位、死区延迟结束时。参见下图。

如果 $POL(n) = 1$ 、 $POL(n+1) = 0$ 且死区处于使能状态,那么在出现通道(n)匹配 (TPM 计数器 = $C(n)V$) 情况时,通道(n)输出将保持高值,直到通道(n)输出清除、死区延迟结束时。与此类似,在出现通道(n+1)匹配 (TPM 计数器 = $C(n+1)V$) 情况时,通道(n+1)输出将保持高值,直到通道(n+1)输出清除、死区延迟结束时。

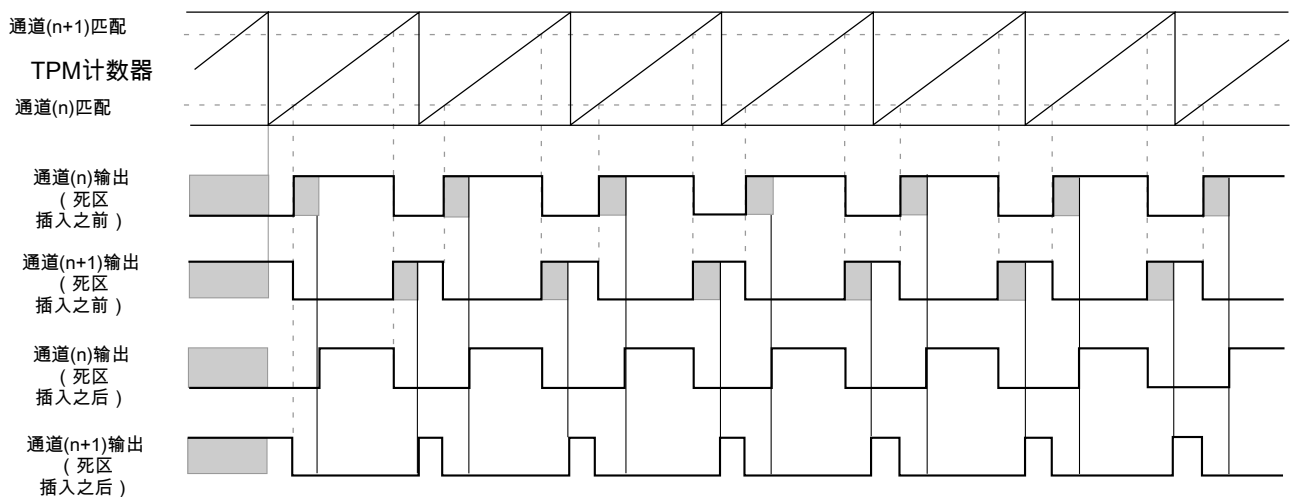


图 37-133. $ELSnB:ELSnA = X:1$ 、 $POL(n) = 0$ 且 $POL(n+1) = 1$ 条件下的死区插入

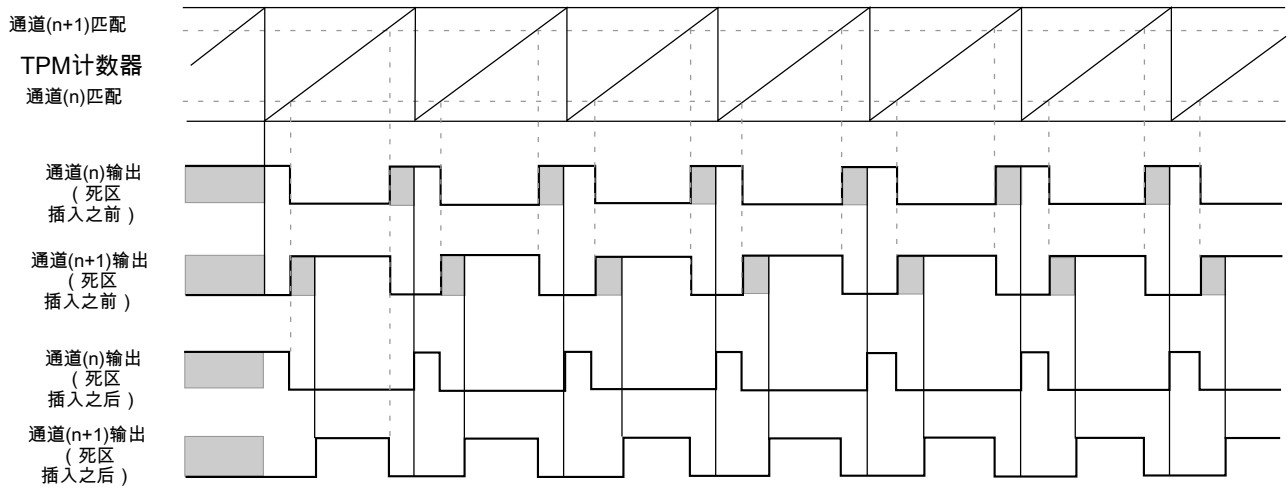


图 37-134. $ELSnB:ELSnA = 1:0$ 、 $POL(n) = 0$ 且 $POL(n+1) = 1$ 条件下的死区插入

37.5.12 正交解码器模式

如果($QUADEN = 1$)，则选择正交解码器模式。正交解码器模式利用通道 0 (A 相) 和通道 1 (B 相) 输入信号控制 TPM 计数器的增减量。以下是正交解码器功能框图。

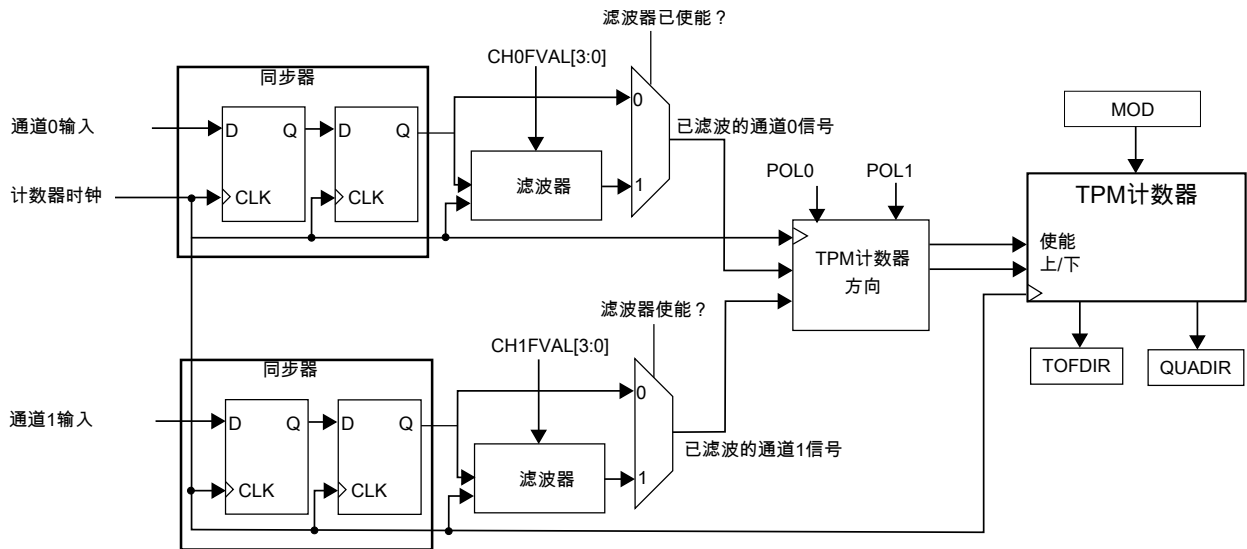


图 37-135. 正交解码器功能框图

输入捕捉滤波器和通道极性寄存器用于配置通道 0 和通道 1 输入在正交解码模式下的输入滤波器和极性。

注

注意，选择正交解码器模式后，TPM 计数器由通道 0 和通道 1 输入信号计时。因此，在正交解码器模式下，通道 0 和通道 1 只能用于软件比较模式，其他 TPM 通道只能用于输入捕捉或输出比较模式。

QUADMODE 选择正交解码器模式中所用的编码模式。如果 QUADMODE = 1，则使能计数和方向编码模式；参见下图。在此模式下，通道 1 输入值指明计数方向，通道 0 输入定义计数率。通道 0 输入信号中存在上升沿时，将更新 TPM 计数器。

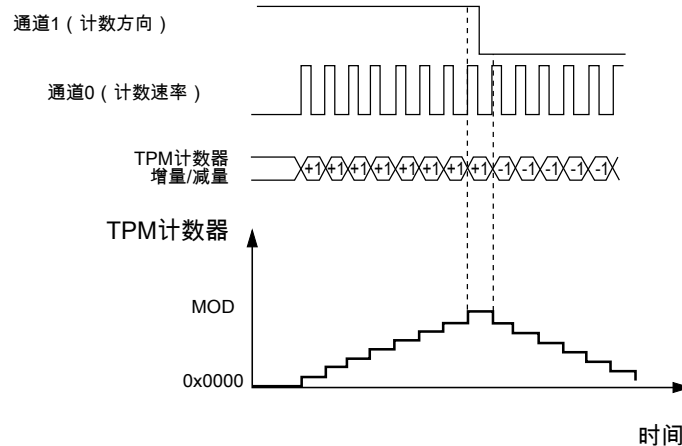


图 37-136. 正交解码器 – 计数和方向编码模式

如果 QUADMODE = 0，则使能相位编码模式；参见下图。在此模式下，通道 0 和通道 1 信号之间的关系指明计数方向，通道 0 和通道 1 信号定义计数率。当通道 0 或通道 1 信号中存在边沿时，TPM 计数器将更新。

如果 CH0POL = 0 且 CH1POL = 0，则在以下情形中出现 TPM 计数器增量：

- 通道 0 信号中存在上升沿，通道 1 信号为逻辑零；
- 通道 1 信号中存在上升沿，通道 0 信号为逻辑一；
- 通道 1 信号中存在下降沿，通道 0 信号为逻辑零；
- 通道 0 信号中存在下降沿，通道 1 信号为逻辑一；

在以下情形中出现 TPM 计数器减量：

- 通道 0 信号中存在下降沿，通道 1 信号为逻辑零；
- 通道 1 信号中存在下降沿，通道 0 信号为逻辑一；
- 通道 1 信号中存在上升沿，通道 0 信号为逻辑零；
- 通道 0 信号中存在上升沿，通道 1 信号为逻辑一；

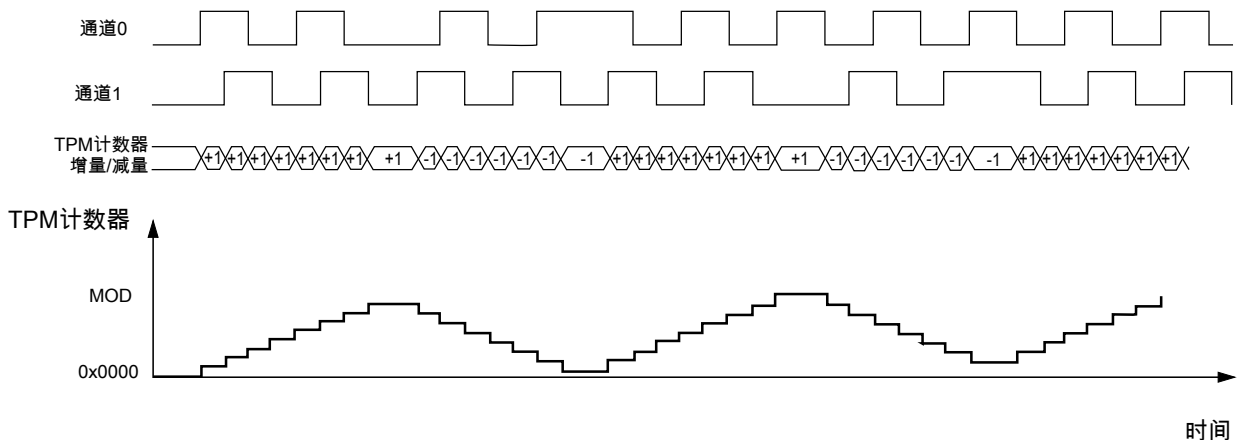


图 37-137. 正交解码器 – 相位编码模式

下图展示了向上计数时的 TPM 计数器溢出。这种情况下，TPM 计数器从 MOD 更改为零时，将置位 TOF 和 TOFDIR 位。TOF 位指明 TPM 计数器溢出已发生。TOFDIR 位指明 TPM 计数器溢出发生时计数方式为向上。

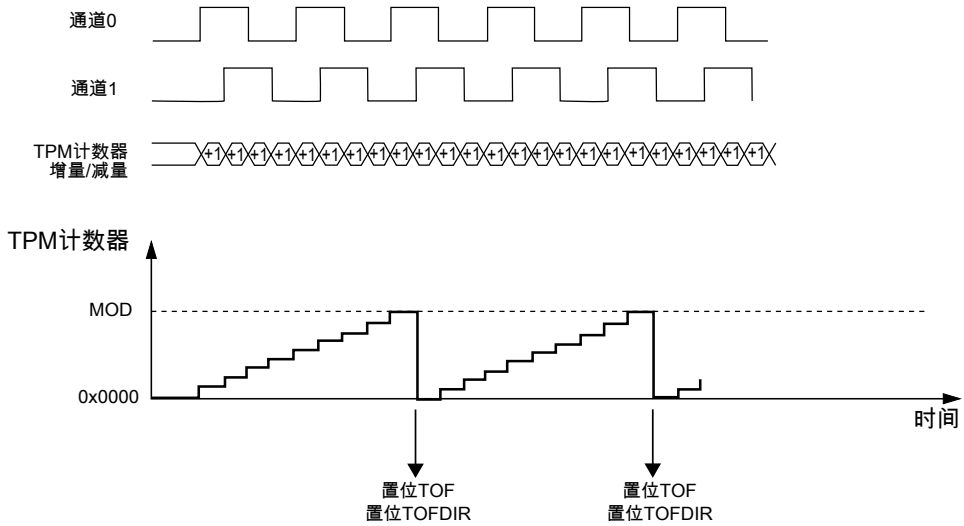


图 37-138. 正交解码器模式下向上计数过程中的 TPM 计数器溢出

下图展示了向下计数时的 TPM 计数器溢出。这种情况下，TPM 计数器从零更改为 MOD 时，将置位 TOF 位并清除 TOFDIR 位。TOF 位指明 TPM 计数器溢出已发生。TOFDIR 位指明 TPM 计数器溢出发生时计数方式为向下。

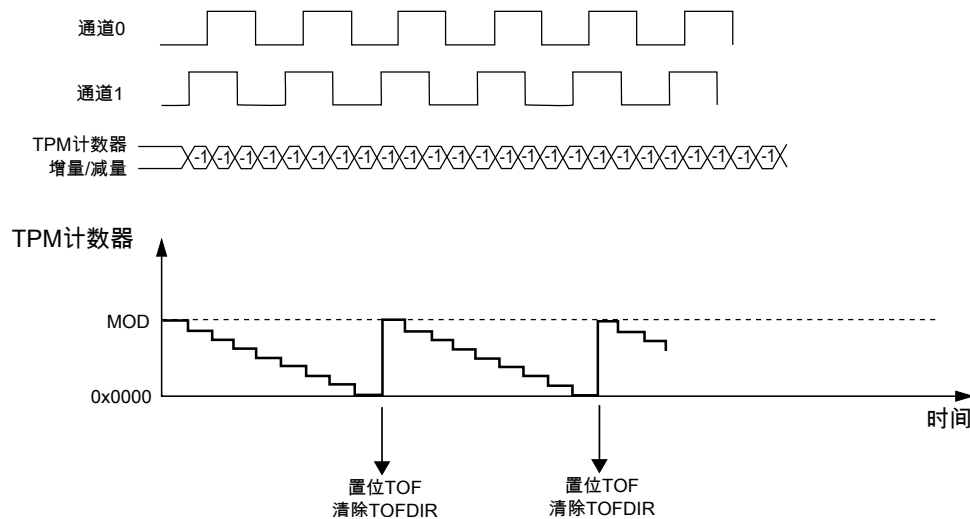


图 37-139. 正交解码器模式下向下计数过程中的 TPM 计数器溢出

37.5.13 通过写缓存更新的寄存器

37.5.13.1 MOD 寄存器更新

如果(CMOD[1:0] = 0:0), 则 MOD 将在 MOD 寄存器被写入时更新。

如果(CMOD[1:0] ≠ 0:0), 则 MOD 寄存器将根据 CPWMS 位更新, 即:

- 如果选择的模式不是 CPWM, 则 MOD 寄存器在对 MOD 寄存器采取写入操作且 TPM 计数器从 MOD 更改为零之后更新。
- 如果选择的模式为 CPWM, 则 MOD 寄存器在对 MOD 寄存器采取写入操作且 TPM 计数器从 MOD 更改为(MOD -1)之后更新。

37.5.13.2 CnV 寄存器更新

如果(CMOD[1:0] = 0:0), 则 CnV 寄存器将在 CnV 寄存器被写入时更新。

如果(CMOD[1:0] ≠ 0:0), 则 CnV 寄存器将根据所选模式更新, 即:

- 如果选择的模式为输出比较, 则 CnV 寄存器在下次 TPM 计数器增量 (预分频器计数结束) 时并且 CnV 寄存器被写入之后更新。

- 如果选择的模式为 EPWM, 则 CnV 寄存器在 CnV 寄存器被写入且 TPM 计数器从 MOD 更改为零之后更新。
- 如果选择的模式为 CPWM, 则 CnV 寄存器在 CnV 寄存器被写入且 TPM 计数器从 MOD 更改为(MOD -1)之后更新。

37.5.14 DMA

通道和溢出标志根据 DMA 和 CHnIE/TOIE 位生成 DMA 传输请求。

有关更多信息, 请参见下表。

表 37-144. DMA 传输请求

DMA	CHnIE/ TOIE	通道/溢出 DMA 传输请求	通道/溢出中断
0	0	不生成通道/溢出 DMA 传输请求。	不生成通道/溢出中断。
0	1	不生成通道/溢出 DMA 传输请求。	如果(CHnF/TOF = 1), 则生成通道/溢出中断。
1	0	如果(CHnF/TOF = 1), 则生成通道/溢出 DMA 传输请求。	不生成通道/溢出中断。
1	1	如果(CHnF/TOF = 1), 则生成通道/溢出 DMA 传输请求。	如果(CHnF/TOF = 1), 则生成通道/溢出中断。

如果 DMA = 1, CHnF/TOF 位可通过完成的 DMA 传输或者向 CHnF/TOF 位写入一来清除 (见下表)。

表 37-145. 清除 CHnF/TOF 位

DMA	如何清除 CHnF/TOF 位
0	CHnF/TOF 位通过向 CHnF/TOF 位写入 1 来清除。
1	CHnF/TOF 位在 DMA 传输完成时清除, 或者通过向 CHnF/TOF 位写入 1 清除。

37.5.15 输出触发器

TPM 产生输出触发信号给计数器以及各个通道, 来作为其他外设模块的触发事件。

每当 TOF 置位时, 计数器触发器均将置位, 并在下一次递增之前保持置位状态。

各个 TPM 通道都会同时生成预触发器输出和触发器输出。每当 CHnF 置位时, 预触发器输出都会置位; 在预触发器置位之后计数器第一次递增时, 触发器输出置位; 然后, 触发器和预触发器都会在触发器置位之后的计数器首次递增时取反。

当(COMBINEn = 1) 处于输出比较模式时，通道(n)和通道(n+1)的预触发器输出会在CH(n)F置位时置位，并在CH(n+1)F置位时取反。触发器将在预触发器置位之后、计数器第一次递增时继续置位，并在预触发器取反的同时取反。

37.5.16 复位概述

无论何时，只要有芯片复位，TPM 就会复位。

TPM 何时从复位状态退出：

- TPM 计数器和预分频器计数器为零，且已停止(CMOD[1:0] = 0:0)；
- 定时器溢出中断为零；
- 通道中断为零；
- 通道处于输入捕捉模式；
- 通道输出为零；
- 通道引脚不由 TPM (ELS(n)B:ELS(n)A = 0:0)控制。

37.5.17 TPM 中断

本节介绍 TPM 中断。

37.5.17.1 定时器溢出中断

(TOIE = 1)且(TOF = 1)时，将生成定时器溢出中断。

37.5.17.2 通道(n)中断

当(CHnIE = 1)且(CHnF = 1)时，会生成通道(n)中断。

第 38 章 周期性中断定时器 (PIT)

38.1 此模块的芯片实现细节

38.1.1 PIT/DMA 周期性触发器分配

PIT 生成 DMA 多路复用器的周期性触发事件的通道分配，如下表所示。

表 38-1. 实现 DMA 周期性触发的 PIT 通道分配

DMA 通道编号	PIT 通道
DMA 通道 0	PIT 通道 0
DMA 通道 1	PIT 通道 1
DMA 通道 2	PIT 通道 2
DMA 通道 3	PIT 通道 3

38.1.2 PIT/ADC 触发器

PIT 触发器使用 SIM_SOPT7[ADCxTRGSEL] 字段被选为 ADCx 触发器源。详情请参见 SIM 章节。

38.2 简介

PIT 模块是一组定时器，用于生成中断和触发 DMA 通道。

38.2.1 结构框图

下图是 PIT 模块的结构框图。

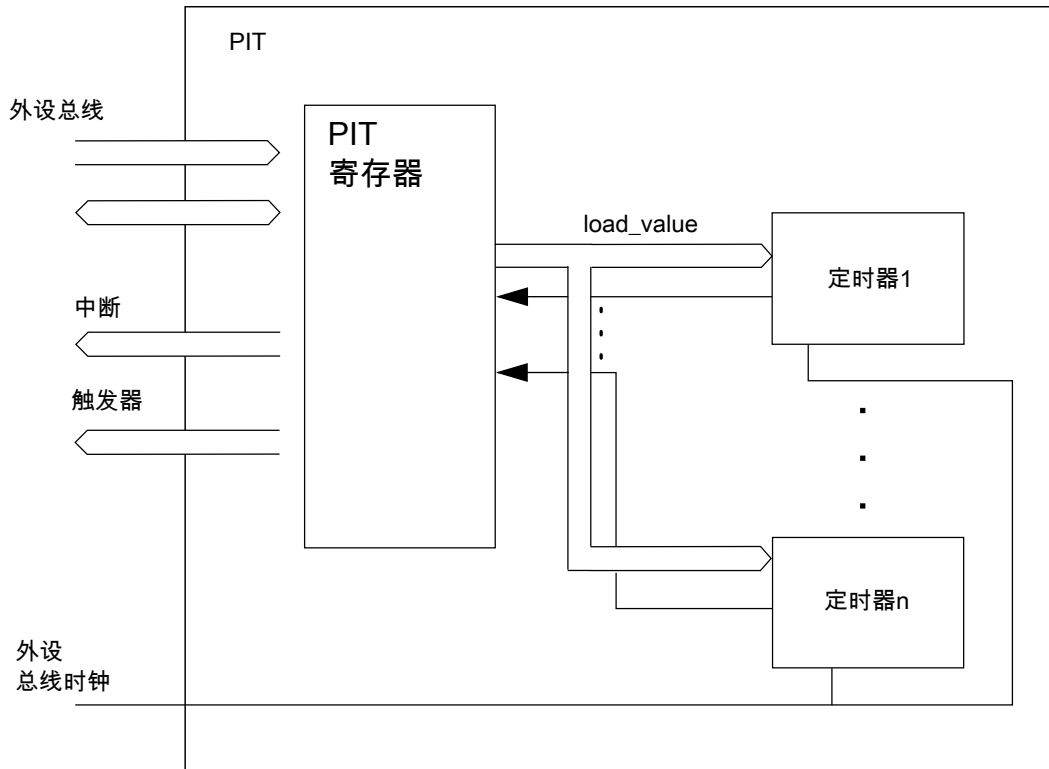


图 38-1. PIT 结构框图

注

有关该 MCU 使用的 PIT 通道数，请参见各芯片的 PIT 信息。

38.2.2 特性

该数据块具有如下主要特性：

- 定时器能够生成 DMA 触发脉冲
- 定时器能够生成中断
- 可屏蔽中断
- 每个定时器都具有独立的超时周期

38.3 信号说明

PIT 模块没有外部引脚。

38.4 存储器映射/寄存器说明

本节详细说明 PIT 模块中可访问的所有寄存器。

- 保留寄存器将读取为 0，写操作将无效。
- 有关该 MCU 使用的 PIT 通道数，请参见各芯片的 PIT 信息。

PIT 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4003_7000	PIT 模块控制寄存器 (PIT_MCR)	32	R/W	0000_0002h	38.4.1/777
4003_70E0	PIT 高位寿命定时器寄存器 (PIT_LTMR64H)	32	R	0000_0000h	38.4.2/779
4003_70E4	PIT 低位寿命定时器寄存器 (PIT_LTMR64L)	32	R	0000_0000h	38.4.3/779
4003_7100	定时器加载值寄存器 (PIT_LDVAL0)	32	R/W	0000_0000h	38.4.4/780
4003_7104	当前定时器值寄存器 (PIT_CVAL0)	32	R	0000_0000h	38.4.5/780
4003_7108	定时器控制寄存器 (PIT_TCTRL0)	32	R/W	0000_0000h	38.4.6/780
4003_710C	定时器标志寄存器 (PIT_TFLG0)	32	R/W	0000_0000h	38.4.7/781
4003_7110	定时器加载值寄存器 (PIT_LDVAL1)	32	R/W	0000_0000h	38.4.4/780
4003_7114	当前定时器值寄存器 (PIT_CVAL1)	32	R	0000_0000h	38.4.5/780
4003_7118	定时器控制寄存器 (PIT_TCTRL1)	32	R/W	0000_0000h	38.4.6/780
4003_711C	定时器标志寄存器 (PIT_TFLG1)	32	R/W	0000_0000h	38.4.7/781
4003_7120	定时器加载值寄存器 (PIT_LDVAL2)	32	R/W	0000_0000h	38.4.4/780
4003_7124	当前定时器值寄存器 (PIT_CVAL2)	32	R	0000_0000h	38.4.5/780
4003_7128	定时器控制寄存器 (PIT_TCTRL2)	32	R/W	0000_0000h	38.4.6/780
4003_712C	定时器标志寄存器 (PIT_TFLG2)	32	R/W	0000_0000h	38.4.7/781
4003_7130	定时器加载值寄存器 (PIT_LDVAL3)	32	R/W	0000_0000h	38.4.4/780
4003_7134	当前定时器值寄存器 (PIT_CVAL3)	32	R	0000_0000h	38.4.5/780
4003_7138	定时器控制寄存器 (PIT_TCTRL3)	32	R/W	0000_0000h	38.4.6/780
4003_713C	定时器标志寄存器 (PIT_TFLG3)	32	R/W	0000_0000h	38.4.7/781

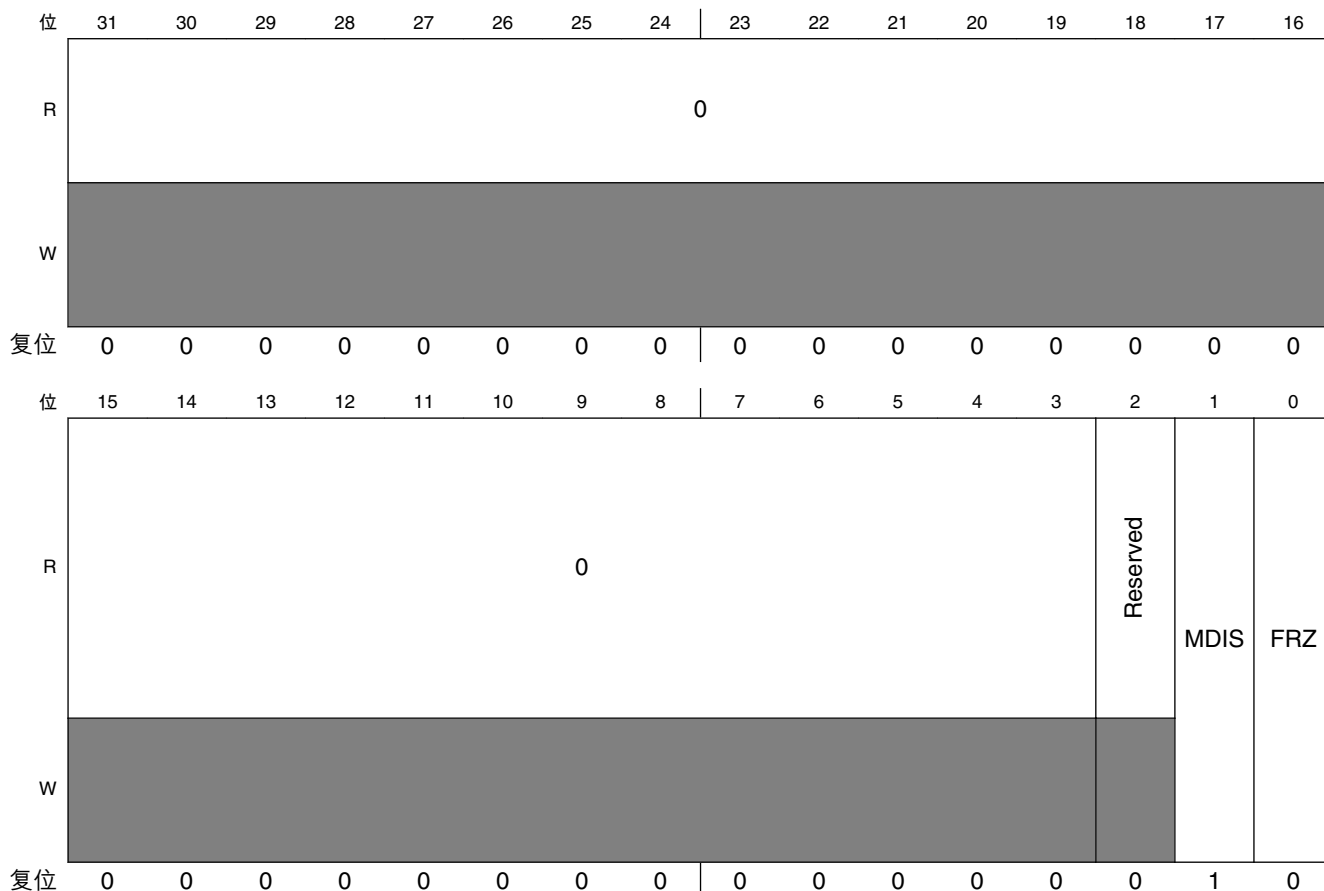
38.4.1 PIT 模块控制寄存器 (PIT_MCR)

当 PIT 进入调试模式时，此寄存器可用于使能或禁用 PIT 定时器时钟并控制定时器。

访问：用户读取/写入

存储器映射/寄存器说明

地址: 4003_7000h 基准 + 0h 偏移 = 4003_7000h



PIT_MCR 字段描述

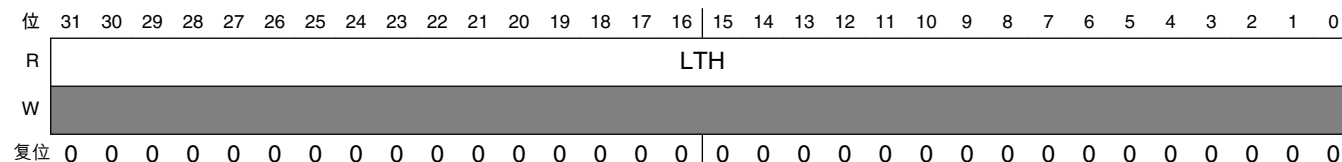
字段	描述
31-3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 Reserved	此字段为保留字段。
1 MDIS	模块禁用 - (PIT 部分) 禁用标准定时器。必须在执行任何其他设置之前使能该字段。 0 标准 PIT 定时器的时钟使能。 1 标准 PIT 定时器的时钟禁用。
0 FRZ	冻结 当器件进入调试模式时，允许停止定时器。 0 定时器在调试模式下继续运行。 1 定时器在调试模式下停止运行。

38.4.2 PIT 高位寿命定时器寄存器 (PIT_LTMR64H)

此寄存器适用于链接定时器 0 和定时器 1 以构建 64 位寿命定时器的应用。

访问：仅限用户读取

地址：4003_7000h 基准 + E0h 偏移 = 4003_70E0h



PIT_LTMR64H 字段描述

字段	描述
LTH	寿命定时器值 显示定时器 1 的定时器值。如果此寄存器读数为时间 t1，则 LTMR64L 显示定时器 0 在时间 t1 时的值。

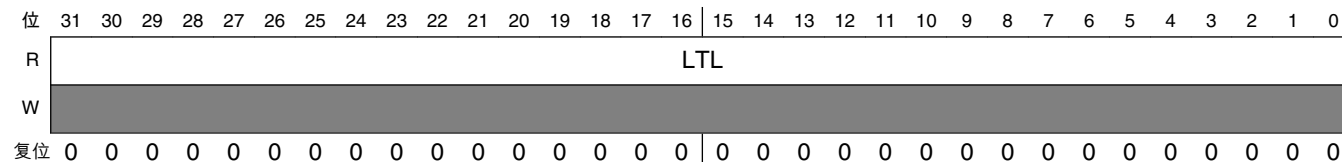
38.4.3 PIT 低位寿命定时器寄存器 (PIT_LTMR64L)

此寄存器适用于链接定时器 0 和定时器 1 以构建 64 位寿命定时器的应用。

要使用 LTMR64H 和 LTMR64L，需要链接定时器 0 和定时器 1。要获取正确值，首先读取 LTMR64H，然后读取 LTMR64L。第一次访问时 LTMR64H 将具有 CVAL1 值，第一次访问时 LTMR64L 将具有 CVAL0 值，因此此应用无需担心运行计数器的滞后效应。

访问：仅限用户读取

地址：4003_7000h 基准 + E4h 偏移 = 4003_70E4h



PIT_LTMR64L 字段描述

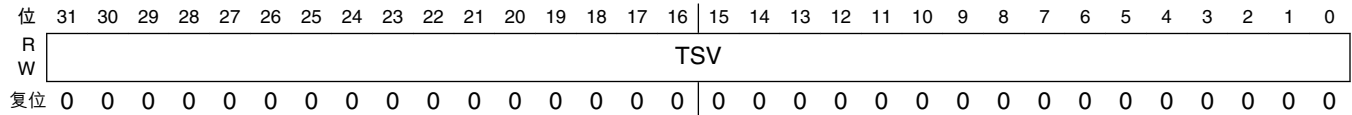
字段	描述
LTL	寿命定时器值 显示定时器 0 在上次读取 LTMR64H 时的值。仅当读取 LTMR64H 时才更新。

38.4.4 定时器加载值寄存器 (PIT_LDVALn)

这些寄存器选择定时器中断的定时溢出周期。

访问：用户读取/写入

Address: 4003_7000h base + 100h offset + (16d × i), where i=0d to 3d



PIT_LDVALn 字段描述

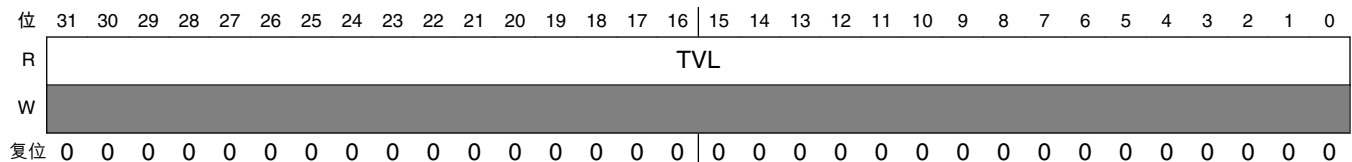
字段	描述
TSV	<p>定时器起始值</p> <p>设置定时器起始值。定时器将倒计时至 0，然后生成一个中断并再次加载该寄存器值。将新值写入该寄存器不会重启定时器，定时器到期后会加载新值。要中止当前周期并用新值开始一个定时器周期，必须先禁用该定时器然后再将其使能。</p>

38.4.5 当前定时器值寄存器 (PIT_CVALn)

这些寄存器指示当前定时器位置。

访问：仅限用户读取

Address: 4003_7000h base + 104h offset + (16d × i), where i=0d to 3d



PIT_CVALn 字段描述

字段	描述
TVL	<p>当前定时器值</p> <p>代表当前定时器值（若定时器已使能）。</p> <p>注：</p> <ul style="list-style-type: none"> 若定时器已禁用，请勿使用该字段，因为其值不可靠。 定时器使用向下计数器。如果 MCR[FRZ]置位，定时器在调试模式下会被冻结。

38.4.6 定时器控制寄存器 (PIT_TCTRLn)

这些寄存器包含每个定时器的控制位。

访问: 用户读取/写入

Address: 4003_7000h base + 108h offset + (16d × i), where i=0d to 3d

位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16	
R	0																	
W																		
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	
位	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0	
R	0															CHN	TIE	TEN
W																		
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	

PIT_TCTRLn 字段描述

字段	描述
31-3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 CHN	链模式 激活时, 定时器 n-1 需先计时结束, 定时器 n 才能递减 1。 不能链接定时器 0。 0 定时器不链接。 1 定时器链接到前一定时器。例如, 对于通道 2, 若该字段置位, 则定时器 2 链接到定时器 1。
1 TIE	定时器中断使能 某个中断等待响应或 TFLGn[TIF]置位时, 使能该中断将立即引起中断事件。为避免这种情况, 必须先清零相关的 TFLGn[TIF]。 0 定时器 n 的中断请求禁用。 1 只要 TIF 置位, 就会请求中断。
0 TEN	定时器使能 使能或禁用定时器。 0 定时器 n 禁用。 1 定时器 n 使能。

38.4.7 定时器标志寄存器 (PIT_TFLGn)

这些寄存器保存 PIT 中断标志。

访问: 用户读取/写入

Address: 4003_7000h base + 10Ch offset + (16d × i), where i=0d to 3d

位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R	0																
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															TIF
W																w1c
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_TFLGn 字段描述

字段	描述
31-1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
0 TIF	定时器中断标志 在定时器周期结束时置 1。将 1 写入该标志可将其清零。写入 0 则无效。若使能或 TCTRLn[TIE] = 1 ,TIF 将引发中断请求。 0 定时溢出尚未发生。 1 定时溢出已经发生。

38.5 功能说明

本节说明该模块的功能。

38.5.1 常规操作

本节详细说明该模块的内部操作。每个定时器都可用于生成触发脉冲和中断。每个中断都可用于单独的中断线。

38.5.1.1 定时器

定时器在使能时定期生成触发脉冲。定时器加载 LDVAL 寄存器指定的起始值，倒数至 0，然后再次加载相应的起始值。每次定时器达到 0 时，它就会生成一个触发脉冲并置位中断标志。

所有中断都可通过设置 TCTRLn[TIE]来使能或屏蔽。新的中断只能在清除上一个中断之后生成。

需要时，定时器的当前计数值可通过 CVAL 寄存器读取。

通过 TCTRLn[TEN]先禁用定时器再将其使能可重启计数周期。请参见下图。

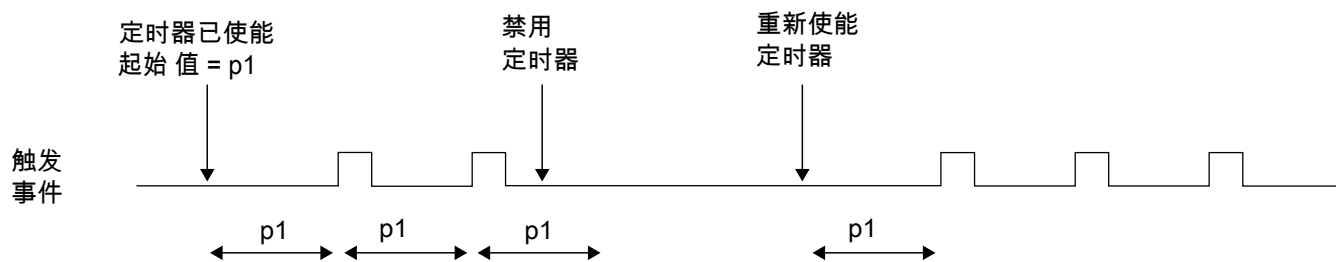


图 38-25. 停止和启动定时器

先禁用定时器，设置新加载值，然后再次使能定时器，可以修改运行中的定时器的计数周期。参见下图。

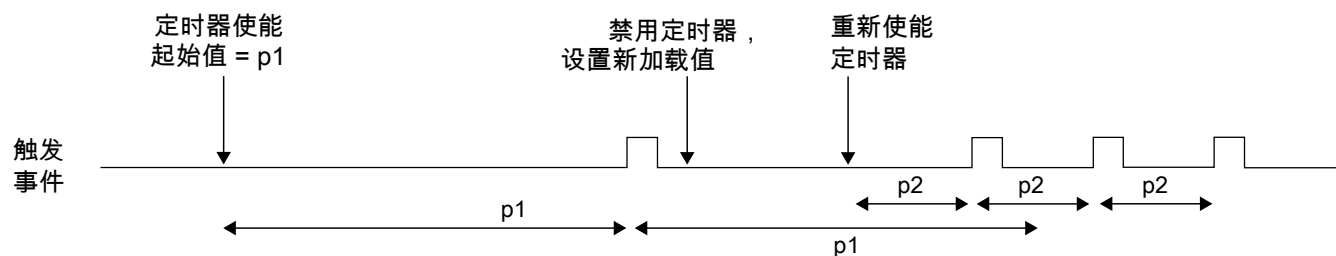


图 38-26. 修改运行中的定时器的周期

还可以在不重启定时器的情况下更改计数周期，方法是将新的加载值写入 LDVAL。该值将在下一个触发事件之后加载。参见下图。

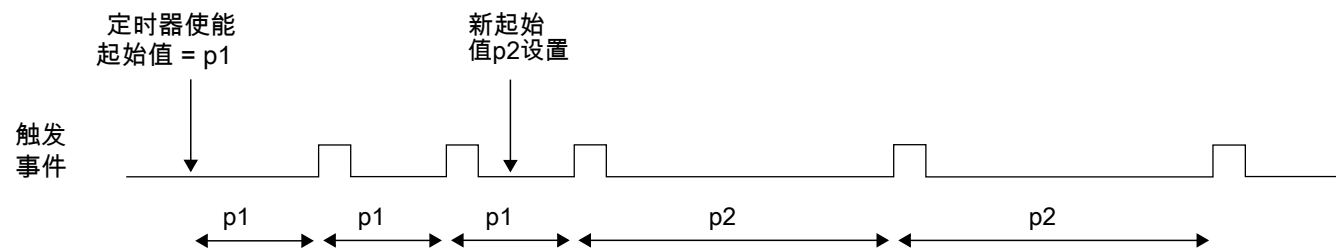


图 38-27. 动态设置新加载值

38.5.1.2 调试模式

在调试模式下，定时器将根据 MCR[FRZ] 决定是否冻结。其目的是协助软件开发，使开发人员能够暂停处理器，调查系统的当前状态，如定时器值等，然后继续运行。

38.5.2 中断

所有定时器都支持中断生成。相关的向量地址和优先级，请参见 MCU 规范。

定时器中断可通过置位 TCTRLn[TIE]来使能。相关定时器发生定时溢出时，TFLGn[TIF]置位为 1；将 1 写入对应的 TFLGn[TIF]可将其清零。

38.5.3 链接定时器

如果某个定时器链接模式处于使能状态，那么只有在上一个定时器计时结束后，它才会开始计时。因此，如果定时器 n-1 已倒数至 0，定时器 n 的值将递减 1。这样就能将某些定时器链接起来形成更长的定时器。第一个定时器（定时器 0）不能链接至任何其他定时器。

38.6 初始化和应用信息

在配置示例中：

- PIT 时钟的频率为 50 MHz。
- 定时器 1 每隔 5.12 ms 生成一个中断。
- 定时器 3 每隔 30 ms 生成一个触发事件。

PIT 模块必须通过将 0 写入 MCR[MDIS]来激活。

50 MHz 时钟频率等于时钟周期 20 ns。定时器 1 需要每隔 $5.12 \text{ ms} / 20 \text{ ns} = 256,000$ 个周期触发一次，定时器 3 每隔 $30 \text{ ms} / 20 \text{ ns} = 1,500,000$ 个周期触发一次。LDVAL 寄存器触发器的值计算如下：

LDVAL 触发器 = (周期 / 时钟周期) - 1

这意味着将 0x0003E7FF 和 0x0016E35F 分别写入 LDVAL1 和 LDVAL3。

定时器 1 的中断通过置位 TCTRL1[TIE]来使能。定时器通过将 1 写入 TCTRL1[TEN]来启动。

定时器 3 只能用于触发。因此，定时器 3 通过将 1 写入 TCTRL3[TEN]来启动。TCTRL3[TIE]保持为 0。

下面的示例代码与上述设置一致：

```
// turn on PIT
PIT_MCR = 0x00;

// Timer 1
PIT_LDVAL1 = 0x0003E7FF; // setup timer 1 for 256000 cycles
PIT_TCTRL1 = TIE; // enable Timer 1 interrupts
PIT_TCTRL1 |= TEN; // start Timer 1
```

```
// Timer 3
PIT_LDVAL3 = 0x0016E35F; // setup timer 3 for 1500000 cycles
PIT_TCTRL3 |= TEN; // start Timer 3
```

38.7 链接定时器配置示例

在配置示例中：

- PIT 时钟的频率为 100 MHz。
- 定时器 1 和定时器 2 可用。
- 每隔 1 分钟应发生一次中断。

PIT 模块需要通过将 0 写入 MCR[MDIS]来激活。

100 MHz 时钟频率相当于 10 ns 的时钟周期，因此 PIT 需要计数 60 亿个周期，这不是单个定时器能够做到的。因此，将定时器 1 设置为每 6 s (6 亿个周期) 触发一次。定时器 2 链接至定时器 1，经过编程后触发 10 次。

LDVAL 寄存器触发器的值等于周期数减 1，因此 LDVAL1 接收值 0x23C345FF，LDVAL2 接收值 0x00000009。

定时器 2 的中断通过置位 TCTRL2[TIE]来使能，链模式通过置位 TCTRL2[CHN]来激活，定时器通过将 1 写入 TCTRL2[TEN]来启动。TCTRL1[TEN]需要置位，TCTRL1[CHN]和 TCTRL1[TIE]需要清零。

下面的示例代码与上述设置一致：

```
// turn on PIT
PIT_MCR = 0x00;

// Timer 2
PIT_LDVAL2 = 0x00000009; // setup Timer 2 for 10 counts
PIT_TCTRL2 = TIE; // enable Timer 2 interrupt
PIT_TCTRL2 |= CHN; // chain Timer 2 to Timer 1
PIT_TCTRL2 |= TEN; // start Timer 2

// Timer 1
PIT_LDVAL1 = 0x23C345FF; // setup Timer 1 for 600 000 000 cycles
PIT_TCTRL1 = TEN; // start Timer 1
```

38.8 使用寿命定时器的示例配置

要配置寿命定时器，需将通道 0 和 1 链接在一起。

首先，需要通过将 0 写入 CTRL 寄存器中的 MDIS 位来激活 PIT 模块，然后需将 LDVAL 寄存器设置为最大值。

定时器是递减计数器。

下面的示例代码与上述设置一致：

```
// turn on PIT
PIT_MCR = 0x00;

// Timer 1
PIT_LDVAL1 = 0xFFFFFFFF; // setup timer 1 for maximum counting period
PIT_TCTRL1 = 0x0; // disable timer 1 interrupts
PIT_TCTRL1 |= CHN; // chain timer 1 to timer 0
PIT_TCTRL1 |= TEN; // start timer 1

// Timer 0
PIT_LDVAL0 = 0xFFFFFFFF; // setup timer 0 for maximum counting period
PIT_TCTRL0 = TEN; // start timer 0
```

要访问使用寿命，请依次阅读 LTMR64H 和 LTMR64L。

```
current_uptime = PIT_LTMR64H<<32;
current_uptime = current_uptime + PIT_LTMR64L;
```

第 39 章 低功耗定时器（LPTMR）

39.1 此模块的芯片实现细节

39.1.1 LPTMR 预分频器/毛刺滤波器时钟选项

LPTMR 模块的预分频器和毛刺滤波器可通过 LPTMR0_PSR[PCS]位域确定的四个源之一计时。下表列出了针对该位域的具体芯片时钟分布情况。

注

如果 LPTPM 将继续在所有要求的低功耗模式下运行，则选定时钟必须保持使能。

LPTMR0_PSR[PCS]	预分频器/毛刺滤波器时钟编号	芯片时钟
00	0	MCGIRCLK — 内部参考时钟 (VLPS/LLS/VLLS 模式下不可用)
01	1	LPO — 1 kHz 时钟 (VLLS0 模式下不可用)
10	2	ERCLK32K — 辅助外部参考时钟
11	3	OSCERCLK_UNDIV — 未分频外部参考时钟 (VLLS0 模式下不可用)

有关这些时钟的详情请参见[时钟分布](#)部分。

39.1.2 LPTMR 脉冲计数器输入选项

LPTMR_CSR[TPS]位域配置脉冲计数器模式下使用的输入源。下表列出了针对该位域的具体芯片输入分配情况。

LPTMR_CSR[TPS]	脉冲计数器输入编号	芯片输入
00	0	CMP0 输出
01	1	LPTMR_ALT1 引脚
10	2	LPTMR_ALT2 引脚
11	3	保留

39.2 简介

在所有功耗模式下 (包括低漏电模式), 低功耗定时器(LPTMR)可以配置为带可选预分频器的时间计数器, 或者带可选毛刺滤波器的脉冲计数器。它还可以在多数系统复位事件中继续运行, 因此可以用作当天时间计数器。

39.2.1 特性

LPTMR 模块特性包括:

- 带比较功能的 16 位时间计数器或脉冲计数器
 - 可选中断可在任何低功耗模式下产生异步唤醒
 - 硬件触发输出
 - 计数器支持自由运行模式或比较复位模式
- 可针对预分频器/毛刺滤波器配置时钟源
- 可针对脉冲计数器配置输入源
 - 上升沿或下降沿

39.2.2 操作模式

下表介绍了 LPTMR 模块在不同模式下的操作情况。

表 39-1. 操作模式

模式	说明
运行	LPTMR 正常运行。
等待	LPTMR 继续正常运行, 并且可以配置为通过生成中断请求来退出低功耗模式。
停止	LPTMR 继续正常运行, 并且可以配置为通过生成中断请求来退出低功耗模式。
低漏电	LPTMR 继续正常运行, 并且可以配置为通过生成中断请求来退出低功耗模式。
调试	LPTMR 正常运行于脉冲计数器模式下, 但计数器在时间计数器模式下不会递增。

39.3 LPTMR 信号说明

表 39-2. LPTMR 信号说明

信号	I/O	说明
LPTMR0_ALT n	I	脉冲计数器输入引脚

39.3.1 详细信号说明

表 39-3. LPTMR 接口——详细信号说明

信号	I/O	说明
LPTMR_ALT n	I	脉冲计数器输入 LPTMR 可以选择输入引脚之一，以用于脉冲计数器模式。
		状态含义 置位——如果配置为脉冲计数器模式且带高电平有效输入，则置位会导致 CNR 递增。 解除置位——如果配置为脉冲计数器模式且带低电平有效输入，则解除置位会导致 CNR 递增。
		定时 置位或解除置位可能在任何时刻发生；输入可能和总线时钟不同步置位。

39.4 存储器映射和寄存器定义

LPTMR 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4004_0000	低功耗定时器控制状态寄存器 (LPTMR0_CSR)	32	R/W	0000_0000h	39.4.1/790
4004_0004	低功耗定时器预分频寄存器 (LPTMR0_PSR)	32	R/W	0000_0000h	39.4.2/791
4004_0008	低功耗定时器比较寄存器 (LPTMR0_CMR)	32	R/W	0000_0000h	39.4.3/792
4004_000C	低功耗定时器计数器寄存器 (LPTMR0_CNR)	32	R/W	0000_0000h	39.4.4/793

39.4.1 低功耗定时器控制状态寄存器 (LPTMRx_CSR)

地址: 4004_0000h 基准 + 0h 偏移 = 4004_0000h

位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R	0																
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R	0								TCF	TIE	TPS		TPP	TFC	TMS	TEN	
W									w1c								
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0

LPTMRx_CSR 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 TCF	定时器比较标志 LPTMR 使能且 CNR 等于 CMR 和递增时，TCF 置位。LPTMR 禁用或被写入逻辑 1 时，TCF 清零。 0 CNR 值不等于 CMR 和递增。 1 CNR 值等于 CMR 和递增。
6 TIE	定时器中断启用 TIE 置位时，TCF 同时置位的任意时间生成 LPTMR 中断。 0 定时器中断禁用。 1 定时器中断使能。
5-4 TPS	定时器引脚选择 配置脉冲计数器模式下使用的输入源。仅当 LPTMR 禁用时才能修改 TPS。输入连接根据具体器件决定。请参见芯片专用 LPTMR 信息。 00 已选择脉冲计数器输入 0。 01 已选择脉冲计数器输入 1。 10 已选择脉冲计数器输入 2。 11 已选择脉冲计数器输入 3。
3 TPP	定时器引脚极性 在脉冲计数器模式下配置输入源的极性。仅当 LPTMR 禁用时才能修改 TPP。 0 脉冲计数器输入源为有效高电平，CNR 将在上升沿递增。 1 脉冲计数器输入源为有效低电平，CNR 将在下降沿递增。
2 TFC	定时器自由运行计数器 清零时，TFC 配置 CNR 以在 TCF 置位的任意时间复位。置位时，TFC 配置 CNR 以在溢出时复位。仅当 LPTMR 禁用时才能修改 TFC。

下一页继续介绍此表...

LPTMRx_CSR 字段描述 (继续)

字段	描述
	0 TCF 置位后即可复位 CNR。 1 CNR 在溢出时复位。
1 TMS	定时器模式选择 配置 LPTMR 模式。仅当 LPTMR 禁用时才能修改 TMS。 0 时间计数器模式。 1 脉冲计数器模式。
0 TEN	定时器使能 TEN 清零时会复位 LPTMR 内部逻辑，包括 CNR 和 TCF。TEN 置位时，LPTMR 使能。将 1 写入该字段时，不得修改 CSR[5:1]。 0 LPTMR 已禁用，内部逻辑复位。 1 LPTMR 使能。

39.4.2 低功耗定时器预分频寄存器 (LPTMRx_PSR)

地址: 4004_0000h 基准 + 4h 偏移 = 4004_0004h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PRESCALE				PBYP	PCS		
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_PSR 字段描述

字段	描述
31-7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
6-3 PRESCALE	预分频值 配置时间计数器模式下预分频器的大小或脉冲计数器模式下毛刺滤波器的宽度。仅当 LPTMR 禁用时才能修改预分频。 0000 预分频器采用 2 分频预分频器时钟，毛刺滤波器不支持这种配置。 0001 预分频器采用 4 分频预分频器时钟，毛刺滤波器在 2 个上升时钟沿后识别输入引脚的更改。 0010 预分频器采用 8 分频预分频器时钟，毛刺滤波器在 4 个上升时钟沿后识别输入引脚的更改。 0011 预分频器采用 16 分频预分频器时钟，毛刺滤波器在 8 个上升时钟沿后识别输入引脚的更改。 0100 预分频器采用 32 分频预分频器时钟，毛刺滤波器在 16 个上升时钟沿后识别输入引脚的更改。 0101 预分频器采用 64 分频预分频器时钟，毛刺滤波器在 32 个上升时钟沿后识别输入引脚的更改。 0110 预分频器采用 128 分频预分频器时钟，毛刺滤波器在 64 个上升时钟沿后识别输入引脚的更改。 0111 预分频器采用 256 分频预分频器时钟，毛刺滤波器在 128 个上升时钟沿后识别输入引脚的更改。

下一页继续介绍此表...

LPTMRx_PSR 字段描述 (继续)

字段	描述
	1000 预分频器采用 512 分频预分频器时钟，毛刺滤波器在 256 个上升时钟沿后识别输入引脚的更改。 1001 预分频器采用 1024 分频预分频器时钟，毛刺滤波器在 512 个上升时钟沿后识别输入引脚的更改。 1010 预分频器采用 2048 分频预分频器时钟，毛刺滤波器在 1024 个上升时钟沿后识别输入引脚的更改。 1011 预分频器采用 4096 分频预分频器时钟，毛刺滤波器在 2048 个上升时钟沿后识别输入引脚的更改。 1100 预分频器采用 8192 分频预分频器时钟，毛刺滤波器在 4096 个上升时钟沿后识别输入引脚的更改。 1101 预分频器采用 16,384 分频预分频器时钟，毛刺滤波器在 8192 个上升时钟沿后识别输入引脚的更改。 1110 预分频器采用 32,768 分频预分频器时钟，毛刺滤波器在 16,384 个上升时钟沿后识别输入引脚的更改。 1111 预分频器采用 65,536 分频预分频器时钟，毛刺滤波器在 32,768 个上升时钟沿后识别输入引脚的更改。
2 PBYP	预分频器旁通 PBYP 置位时，时间计数器模式下选择的预分频器时钟或脉冲计数器模式下选择的输入源可直接对 CNR 进行计时。PBYP 清零时，预分频器/毛刺滤波器输出对 CNR 进行计时。仅当 LPTMR 禁用时才能修改 PBYP。 0 预分频器/毛刺滤波器使能。 1 预分频器/毛刺滤波器旁路。
PCS	预分频器时钟选择 选择 LPTMR 预分频/毛刺滤波器使用的时钟。仅当 LPTMR 禁用时才能修改 PCS。时钟连接根据具体器件决定。 注：有关这些输入的连接信息，请参见芯片配置详情。 00 已选择预分频器/毛刺滤波器时钟 0 01 已选择预分频器/毛刺滤波器时钟 1 10 已选择预分频器/毛刺滤波器时钟 2 11 已选择预分频器/毛刺滤波器时钟 3

39.4.3 低功耗定时器比较寄存器 (LPTMRx_CMR)

地址: 4004_0000h 基准 + 8h 偏移 = 4004_0008h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COMPARE															
W	0																0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CMR 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
COMPARE	比较值 LPTMR 使能且 CNR 等于 CMR 的值和递增时，TCF 置位且硬件触发置位直至 CNR 下一次递增。若 CMR 为 0，硬件触发将保持置位直至 LPTMR 禁用。若 LPTMR 使能，CMR 只有在 TCF 置位时才能更改。

39.4.4 低功耗定时器计数器寄存器 (LPTMRx_CNR)

注

有关计数器值读取方法的详情，请参见 [LPTMR 计数器](#)。

地址: 4004_0000h 基准 + Ch 偏移 = 4004_000Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNTER															
W	0																0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CNR 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
COUNTER	计数器值 CNR 返回 LPTMR 计数器寄存器上一次写入时候的值。CNR 写入将 LPTMR 值锁定以供后续读取，写入值被忽略。

39.5 功能说明

39.5.1 LPTMR 功耗和复位

LPTMR 在所有功耗模式下均会保持工作，包括低漏电模式。如果在低功耗模式下，LPTMR 无需保持运行，则必须在进入该模式之前将其禁用。

LPTMR 只会在全局上电复位(POR)或低电压检测(LVD)时复位。在配置 LPTMR 寄存器时，必须在最初写入 CSR 时禁用定时器，然后配置 PSR 和 CMR。接着，必须将 CSR[TIE]设为初始化中的最后一步。这样可以确保 LPTMR 配置正确，并且 LPTMR 计数器在热复位之后重置为零。

39.5.2 LPTMR 时钟

LPTMR 预分频器/毛刺滤波器可由四个时钟之一计时。在使能 LPTMR 之前，必须使能时钟源。

注

选择的时钟源需要进行配置，使其在低功耗模式下保持使能，否则，LPTMR 在低功耗模式下不会工作。

在脉冲计数器模式下，当预分频器/毛刺滤波器处于旁通状态时，所选输入源直接对 CNR 计时，无需其他时钟源。这种情况下，为了尽量降低功耗，要把预分频器时钟源配置为未切换的时钟。

注

为 LPTMR 选择的时钟源或脉冲输入源不得超过器件数据手册中定义的频率 f_{LPTMR} 。

39.5.3 LPTMR 预分频器/毛刺滤波器

LPTMR 预分频器和毛刺滤波器共用同一逻辑，在时间计数器模式下作为预分频器工作，在脉冲计数器模式下作为毛刺滤波器工作。

注

当 LPTMR 使能时，不得更改预分频器/毛刺滤波器配置。

39.5.3.1 预分频器使能

在时间计数器模式下，当预分频器使能时，预分频器的输出直接对 CNR 计时。当 LPTMR 使能时，CNR 将每隔 2^2 至 2^{16} 个预分频器时钟周期递增。LPTMR 使能后，由于同步逻辑，CNR 的首次递增需要额外一个或两个预分频器时钟周期。

39.5.3.2 预分频器旁通

在时间计数器模式下，当预分频器旁通时，选定的预分频器在每个时钟周期递增一次 CNR。当 LPTMR 使能时，由于同步逻辑，首次递增需要额外一个或两个预分频器时钟周期。

39.5.3.3 毛刺滤波器

在脉冲计数器模式下，当毛刺滤波器使能时，毛刺滤波器的输出会直接对 CNR 计时。当先使能 LPTMR 时，毛刺滤波器的输出置位，即高电平有效为逻辑 1，低电平有效为逻辑 0。下表列出了在选定输入源条件下毛刺滤波器输出的变化。

条件	结果
选定输入源至少在 2^1 至 2^{15} 个连续预分频器时钟上升沿保持解除置位状态	毛刺滤波器输出也将解除置位。
选定输入源至少在 2^1 至 2^{15} 个连续预分频器时钟上升沿保持置位状态	毛刺滤波器输出也将置位。

注

仅在上升时钟边沿对输入采样。

每当毛刺滤波器输出置位时，CNR 递增。在脉冲计数器模式下，CNR 可以递增的最大速率为每隔 2^2 至 2^{16} 个预分频器时钟边沿。当首次使能时，由于同步逻辑，毛刺滤波器会等待额外一个或两个预分频器时钟边沿。

39.5.3.4 毛刺滤波器旁通

在脉冲计数器模式下，当毛刺滤波器旁通时，选定输入源每置位一次就会递增一次 CNR。LPTMR 首次使能之前，选定输入源将被强行置位。当 LPTMR 首次使能时，如果选定输入源已经置位，这样做可以阻止 CNR 递增。

39.5.4 LPTMR 比较

当 CNR 等于 CMR 的值并递增时，将发生以下事件：

- CSR[TCF]置位。
- 如果 CSR[TIE]同时置位，则会生成 LPTMR 中断。
- 生成 LPTMR 硬件触发器。
- 如果 CSR[TFC]清零，CNR 将复位。

当 LPTMR 使能时，CMR 只有在 CSR[TCF]置位时才能更改。在更新 CMR 时，必须在 LPTMR 计数器递增值超过 LPTMR 比较值之前，对 CMR 执行写操作并使 CSR[TCF]清零。

39.5.5 LPTMR 计数器

CNR 在下列情况中递增 1：

- 时间计数器模式下的预分频器时钟（预分频器已旁路）
- 时间计数器模式下的预分频器输出（预分频器已使能）
- 脉冲计数器模式下的输入源置位（毛刺滤波器已旁路）
- 脉冲计数器模式下的毛刺滤波器输出（毛刺滤波器已使能）

LPTMR 禁用或计数器寄存器溢出时 CNR 复位。若 CSR[TFC]清零，则 CNR 在 CSR[TCF]置位时复位。

在脉冲计数器模式下配置时，若内核在调试模式下停止，CNR 仍可继续递增，在时间计数器模式下配置时，若内核在调试模式下停止，CNR 将停止递增。

CNR 无法初始化，但可随时读取。每次读取 CNR 时，软件必须先将任意值写入 CNR。这将 CNR 中的当前值同步并存储到一个临时寄存器中。在每次读取 CNR 时，将返回临时寄存器中的内容。

读取 CNR 时，总线时钟必须比 LPTMR 计数器递增速率至少快两倍，否则可能返回不正确数据。

39.5.6 LPTMR 硬件触发

LPTMR 硬件触发与 CSR[TCF]同时置位，在没有软件干预的情况下也可用来触发其他外设的硬件。硬件触发始终使能。

当	结果
CSR[TFC]清零时 CMR 置位为 0。	LPTMR 硬件触发将在第一次比较时置位且不会解除置位。
若 CSR[TFC]置位，则 CMR 置位为非零值。	LPTMR 硬件触发器将在每次比较时置位，并在后续的 CNR 递增中解除置位。

39.5.7 LPTMR 中断

每当 CSR[TIE]和 CSR[TCF]置位时，将生成 LPTMR 中断。CSR[TCF]通过禁用 LPTMR 或向其写入逻辑 1 清零。

当 LPTMR 使能时，可以更改 CSR[TIE]并清零 CSR[TCF]。

LPTMR 中断的生成与系统时钟异步，可以用于从任意低功耗模式（包括低漏电模式）唤醒，只要将 LPTMR 作为唤醒源使能即可。

第 40 章 实时时钟 (RTC)

40.1 此模块的芯片实现细节

40.1.1 RTC_CLKOUT 信号

当 RTC 使能且端口控制模块选择 RTC_CLKOUT 功能时, RTC_CLKOUT 引脚将输出来自 RTC 振荡器的 1 Hz 或 32 kHz 输出信号, 如下所示。

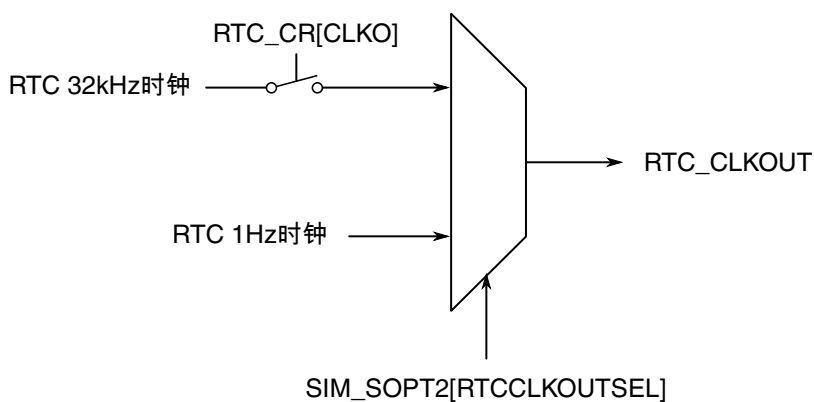


图 40-1. RTC_CLKOUT 生成

40.2 简介

40.2.1 特性

RTC 模块特性包括:

- 独立的电源、POR 和 32 kHz 晶体振荡器

- 带翻转保护功能的 32 位秒计数器和 32 位闹钟
- 带补偿功能的 16 位预分频器，可以校正 0.12 ppm 至 3906 ppm 的误差
- 寄存器写入保护
 - 锁定寄存器功能，需要 VBAT POR 或软件重置以启用写访问
 - 访问控制寄存器要求系统复位，以启用读"与"或者"或"写访问
- 带可选中断的 1 Hz 方波输出

40.2.2 操作模式

RTC 将在所有低功耗模式下保持正常工作，并且可以生成一个中断以退出任何低功耗模式。RTC 有两种工作模式：芯片上电和芯片掉电。

在芯片掉电模式中，RTC 由备用电源(VBAT)供电并与芯片其余部分实施电气隔离，但会继续递增时间计数器（若使能）并保留 RTC 寄存器的状态。不可访问 RTC 寄存器。

在芯片上电模式中，RTC 始终由备用电源(VBAT)供电。所有 RTC 寄存器均可由软件访问，并且所有功能均可正常工作。若使能，可以向芯片其余部分提供 32.768 kHz 的时钟。

40.2.3 RTC 信号说明

表 40-1. RTC 信号说明

信号	说明	I/O
EXTAL32	32.768 kHz 振荡器输入	I
XTAL32	32.768 kHz 振荡器输出	O
RTC_CLKOUT	1 Hz 方波输出或 OSCERCLK	O
RTC_WAKEUP	外部器件唤醒	O

40.2.3.1 RTC 时钟输出

秒计数器时钟在 RTC_CLKOUT 信号上提供。这是 1 Hz 方波输出。详情请参见 [RTC_CLKOUT 选项](#)。

40.2.3.2 RTC 唤醒引脚

RTC 唤醒引脚为开漏、低电平有效输出，允许 RTC 通过外部组件唤醒芯片。唤醒引脚使能置位，且 RTC 中断置位或唤醒引脚通过寄存器位开启时，唤醒引脚置位。唤醒引脚不会从 RTC 秒中断置位。

注

唤醒引脚是可选的，不能在所有器件上实施。

40.3 寄存器定义

所有寄存器必须使用 32 位写入访问，所有寄存器访问产生三个等待状态。

当控制寄存器内的管理员访问位清零时，非管理员模式软件对任意寄存器的读取访问将因总线错误终止。

非管理员模式软件的读取访问正常完成。

写入由写入访问寄存器或锁定寄存器保护的寄存器不会生成总线错误，但写入不会完成。

读取由读取访问寄存器保护的寄存器不会生成总线错误，但寄存器将读取为零。

RTC 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4003_D000	RTC 时间秒寄存器 (RTC_TSR)	32	R/W	0000_0000h	40.3.1/800
4003_D004	RTC 时间预分频器寄存器 (RTC_TPR)	32	R/W	0000_0000h	40.3.2/800
4003_D008	RTC 时间警报寄存器 (RTC_TAR)	32	R/W	0000_0000h	40.3.3/800
4003_D00C	RTC 时间补偿寄存器 (RTC_TCR)	32	R/W	0000_0000h	40.3.4/801
4003_D010	RTC 控制寄存器 (RTC_CR)	32	R/W	0000_0000h	40.3.5/802
4003_D014	RTC 状态寄存器 (RTC_SR)	32	R/W	0000_0001h	40.3.6/804
4003_D018	RTC 锁定寄存器 (RTC_LR)	32	R/W	0000_00FFh	40.3.7/805
4003_D01C	RTC 中断使能寄存器 (RTC_IER)	32	R/W	0000_0007h	40.3.8/806
4003_D800	RTC 写入访问寄存器 (RTC_WAR)	32	R/W	0000_00FFh	40.3.9/807
4003_D804	RTC 读取访问寄存器 (RTC_RAR)	32	R/W	0000_00FFh	40.3.10/ 808

40.3.1 RTC 时间秒寄存器 (RTC_TSR)

地址: 4003_D000h 基准 + 0h 偏移 = 4003_D000h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSR																															
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TSR 字段描述

字段	描述
TSR	<p>时间秒寄存器</p> <p>时间计数器使能后, TSR 为只读, 在 SR[TOF]或 SR[TIF]未置位的前提下, 每秒递增一次。SR[TOF]或 SR[TIF]置位时, 时间计数器读取为零。时间计数器停用后, TSR 可读写。在时间计数器停用时对 TSR 写入会清除 SR[TOF]"与"或者"或"SR[TIF]。支持对 TSR 写入零, 但不推荐这样做, 因为当 SR[TIF]或 SR[TOF]置位时 (指示时间无效), TSR 将读取为零。</p>

40.3.2 RTC 时间预分频器寄存器 (RTC_TPR)

地址: 4003_D000h 基准 + 4h 偏移 = 4003_D004h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TPR															
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TPR 字段描述

字段	描述
31-16 保留	<p>此字段为保留字段。</p> <p>此只读字段为保留字段且值始终为 0。</p>
TPR	<p>时间预分频器寄存器</p> <p>时间计数器使能后, TPR 为只读, 每个 32.768 kHz 时钟周期递增一次。SR[TOF]或 SR[TIF]置位时, 时间计数器读取为零。时间计数器停用后, TPR 可读写。TPR 的位 14 从逻辑一转换为逻辑零后, TSR[TSR]递增。</p>

40.3.3 RTC 时间警报寄存器 (RTC_TAR)

地址: 4003_D000h 基准 + 8h 偏移 = 4003_D008h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TAR																															
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TAR 字段描述

字段	描述
TAR	时间警报寄存器 时间计数器使能后，每当 TAR[TAR]等于 TSR[TSR]且 TSR[TSR]递增时，SR[TAF]置位。写入 TAR 会清除 SR[TAF]。

40.3.4 RTC 时间补偿寄存器 (RTC_TCR)

地址: 4003_D000h 基准 + Ch 偏移 = 4003_D00Ch

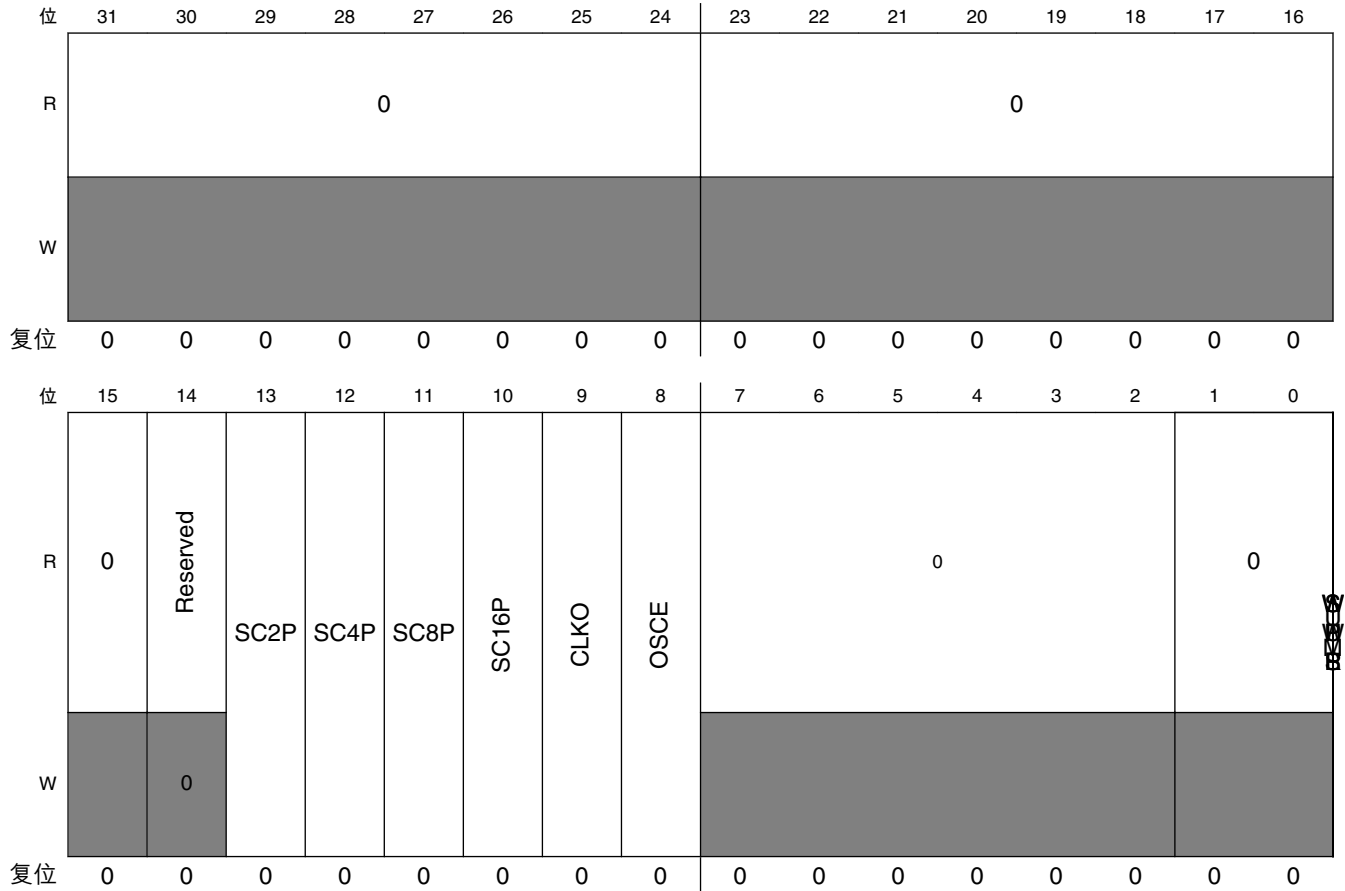
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CIC								TCV								CIR								TCR							
W	0								0								0								0							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TCR 字段描述

字段	描述
31–24 CIC	补偿间隔计数器 补偿间隔计数器的当前值。如果补偿间隔计数器等于零，则为其加载 CIR 的内容。如果 CIC 不等于零，则每秒递减一次。
23–16 TCV	时间补偿值 补偿逻辑用于现在秒间隔的当前值。如果 CIC 等于 0 且带 TCR 字段内容，每秒更新一次。如果 CIC 不等于零，则加载零（该秒增量未使能补偿）。
15–8 CIR	补偿间隔寄存器 在 1 至 256 秒范围内配置补偿间隔，以控制 TCR 每秒内调整 32.768 kHz 周期数的频率。写入的值应比秒数小 1。例如，写入零以配置为一秒补偿间隔。该寄存器双重缓冲，写入在当前补偿间隔结束前不会生效。
TCR	时间补偿寄存器 每秒内配置 32.768 kHz 时钟周期数量。该寄存器双重缓冲，写入在当前补偿间隔结束前不会生效。 80h 时间预分频器寄存器每 32896 个时钟周期溢出。 FFh 时间预分频器寄存器每 32769 个时钟周期溢出。 00h 时间预分频器寄存器每 32768 个时钟周期溢出。 01h 时间预分频器寄存器每 32767 个时钟周期溢出。 7Fh 时间预分频器寄存器每 32641 个时钟周期溢出。

40.3.5 RTC 控制寄存器 (RTC_CR)

地址: 4003_D000h 基准 + 10h 偏移 = 4003_D010h



RTC_CR 字段描述

字段	描述
31-24 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
23-15 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
14 Reserved	此字段为保留字段。 必须始终写入 0。
13 SC2P	振荡器 2pF 负载配置 0 停用负载。 1 使能该负载。
12 SC4P	振荡器 4pF 负载配置 0 停用负载。 1 使能该负载。

下一页继续介绍此表...

RTC_CR 字段描述 (继续)

字段	描述
11 SC8P	振荡器 8pF 负载配置 0 停用负载。 1 使能该负载。
10 SC16P	振荡器 16pF 负载配置 0 停用负载。 1 使能该负载。
9 CLKO	时钟输出 0 将 32 kHz 时钟输出至其他外设。 1 将 32 kHz 时钟输出至其他外设。
8 OSCE	振荡器使能 0 32.768 kHz 振荡器停用。 1 32.768 kHz 振荡器使能。设置该位后，等待振荡器启动时间，然后使能时间计数器，以便 32.768 kHz 时钟时间稳定。
13-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7-5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
4 WPS	唤醒引脚选择 唤醒引脚是可选的，并非所有器件上可用。 0 如果 RTC 中断置位或唤醒引脚开启，则唤醒引脚置位（低电平有效，开漏）。 1 唤醒引脚输出 RTC 32 kHz 时钟，前提是唤醒引脚开启，且将 32 kHz 时钟输出至其他外设。
3 UM	更新模式 允许写入 SR[TCE]，即使状态寄存器锁定。置位后，如果 SR[TIF]或 SR[TOF]置位，或者 SR[TCE]清零，则始终可写入 SR[TCE]。 0 寄存器在锁定状态下无法写入。 1 寄存器在有限条件锁定状态下可写入。
2 SUP	管理员访问 0 不支持非管理员模式写入访问，产生总线错误。 1 支持非管理员模式写入访问。
1 WPE	唤醒引脚使能 唤醒引脚是可选的，并非所有器件上可用。 0 唤醒引脚停用。 1 唤醒引脚使能后，如果 RTC 中断置位或唤醒引脚开启，则唤醒引脚置位。
0 SWR	软件复位 0 无效。 1 复位除 SWR 位以及 RTC_WAR 和 RTC_RAR 寄存器外的所有 RTC 寄存器。SWR 位通过 VBAT POR 清零，由相关软件明确清除。

40.3.6 RTC 状态寄存器 (RTC_SR)

地址: 4003_D000h 基准 + 14h 偏移 = 4003_D014h

位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R	0																
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R	0											TCE	0	TAF	TOF	TIF	
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	1

RTC_SR 字段描述

字段	描述
31-5 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
4 TCE	时间计数器使能 当时间计数器停用时，TSR 寄存器和 TPR 寄存器可写入，但不会递增。当时间计数器使能时，TSR 寄存器和 TPR 寄存器不可写入，但会递增。 0 时间计数器停用。 1 时间计数器使能。
3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 TAF	时间警报标志 TAR[TAR]等于 TSR[TSR]且 TSR[TSR]递增时，时间警报标志置位。该位通过写入 TAR 寄存器清零。 0 时间警报未发生。 1 时间警报已发生。
1 TOF	时间溢出标志 当时间计数器使能且溢出时，时间溢出标志置位。该位置位时，TSR 和 TPR 不会递增并读取为零。时间计数器停用时，该位通过写入 TSR 寄存器清零。 0 时间溢出未发生。 1 时间溢出已发生，且时间计数器读取为零。
0 TIF	时间无效标志 时间无效标志在 VBAT POR 或软件复位上置位。该位置位时，TSR 和 TPR 不会递增并读取为零。时间计数器停用时，该位通过写入 TSR 寄存器清零。 0 时间有效。 1 时间无效，且时间计数器读取为零。

40.3.7 RTC 锁定寄存器 (RTC_LR)

地址: 4003_D000h 基准 + 18h 偏移 = 4003_D018h

位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R	0																
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R	0								1	LRL	SRL	CRL	TCL	1			
W																	
复位	0	0	0	0	0	0	0	0		1	1	1	1	1	1	1	1

RTC_LR 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。
6 LRL	锁定寄存器锁定 清零后，该位只能通过 VBAT POR 或软件复位置位。 0 锁定寄存器被锁定，写入被忽略。 1 锁定寄存器未被锁定，写入正常完成。
5 SRL	状态寄存器锁定 清零后，该位只能通过 VBAT POR 或软件复位置位。 0 状态寄存器被锁定，写入被忽略。 1 状态寄存器未被锁定，写入正常完成。
4 CRL	控制寄存器锁定 清零后，该位只能通过 VBAT POR 置位。 0 控制寄存器被锁定，写入被忽略。 1 控制寄存器未被锁定，写入正常完成。
3 TCL	时间补偿锁定 清零后，该位只能通过 VBAT POR 或软件复位置位。 0 时间补偿寄存器被锁定，写入被忽略。 1 时间补偿寄存器未被锁定，写入正常完成。
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 1。

40.3.8 RTC 中断使能寄存器 (RTC_IER)

地址: 4003_D000h 基准 + 1Ch 偏移 = 4003_D01Ch

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	[Reserved]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								WPON	Reserved	TSIE	Reserved	TAIE	TOIE	TIIE	
W	[Reserved]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

RTC_IER 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 WPON	唤醒引脚开启 唤醒引脚是可选的，并非所有器件上可用。每当唤醒引脚使能且该位置位时，唤醒引脚置位。 0 无效。 1 如果唤醒引脚使能，唤醒引脚将置位。
6-5 Reserved	此字段为保留字段。
4 TSIE	时间秒中断使能 秒中断是带专用中断向量的边沿敏感中断。每秒生成一次，无需软件开销（无需清除对应的状态标志）。秒中断的频率是由 TSIC 配置的。 0 秒中断禁用。 1 秒中断使能。
3 Reserved	此字段为保留字段。
2 TAIE	时间警报中断使能 0 时间警报标志不会生成中断。 1 时间警报标志会生成中断。
1 TOIE	时间溢出中断使能 0 时间溢出标志不会生成中断。 1 时间溢出标志会生成中断。

下一页继续介绍此表...

RTC_IER 字段描述 (继续)

字段	描述
0 TIIE	时间无效中断使能 0 时间无效标志不会生成中断。 1 时间无效标志会生成中断。

40.3.9 RTC 写入访问寄存器 (RTC_WAR)

地址: 4003_D000h 基准 + 800h 偏移 = 4003_D800h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	[阴影区域]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	[阴影区域]								IERW	LRW	SRW	CRW	TCRW	TARW	TPRW	TSRW
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RTC_WAR 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 IERW	中断使能寄存器写入 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对中断使能寄存器的写入被忽略。 1 对中断使能寄存器的写入正常完成。
6 LRW	锁定寄存器写入 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对锁定寄存器的写入被忽略。 1 对锁定寄存器的写入正常完成。
5 SRW	状态寄存器写入 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对状态寄存器的写入被忽略。 1 对状态寄存器的写入正常完成。

下一页继续介绍此表...

RTC_WAR 字段描述 (继续)

字段	描述
4 CRW	控制寄存器写入 清零后，该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对控制寄存器的写入被忽略。 1 对控制寄存器的写入正常完成。
3 TCRW	时间补偿寄存器写入 清零后，该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对时间补偿寄存器的写入被忽略。 1 对时间补偿寄存器的写入正常完成。
2 TARW	时间警报寄存器写入 清零后，该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对时间警报寄存器的写入被忽略。 1 对时间警报寄存器的写入正常完成。
1 TPRW	时间预分频器寄存器写入 清零后，该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对时间预分频器寄存器的写入被忽略。 1 对时间预分频器寄存器的写入正常完成。
0 TSRW	时间秒寄存器写入 清零后，该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对时间秒寄存器的写入被忽略。 1 对时间秒寄存器的写入正常完成。

40.3.10 RTC 读取访问寄存器 (RTC_RAR)

地址: 4003_D000h 基准 + 804h 偏移 = 4003_D804h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								IERR	LRR	SRR	CRR	TCRR	TARR	TPRR	TSRR
W																
复位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RTC_RAR 字段描述

字段	描述
31-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
7 IERR	中断使能寄存器读取 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对中断使能寄存器的读取被忽略。 1 对中断使能寄存器的读取正常完成。
6 LRR	锁定寄存器读取 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对锁定寄存器的读取被忽略。 1 对锁定寄存器的读取正常完成。
5 SRR	状态寄存器读取 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对状态寄存器的读取被忽略。 1 对状态寄存器的读取正常完成。
4 CRR	控制寄存器读取 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对控制寄存器的读取被忽略。 1 对控制寄存器的读取正常完成。
3 TCRR	时间补偿寄存器读取 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对时间补偿寄存器的读取被忽略。 1 对时间补偿寄存器的读取正常完成。
2 TARR	时间警报寄存器读取 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对时间警报寄存器的读取被忽略。 1 对时间警报寄存器的读取正常完成。
1 TPRR	时间预分频器寄存器读取 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对时间预分频器寄存器的读取被忽略。 1 对时间预分频器寄存器的读取正常完成。
0 TSRR	时间秒寄存器读取 清零后, 该位只能通过系统复位置位。它不受 VBAT POR 或软件复位影响。 0 对时间秒寄存器的读取被忽略。 1 对时间秒寄存器的读取正常完成。

40.4 功能说明

40.4.1 功耗、时钟和复位

RTC 是一个始终上电的模块，在所有低功耗模式下保持活动的状态，并由电池电源供电(VBAT)。电池电源确保 RTC 寄存器在芯片掉电期间保持其状态，且 RTC 时间计数器保持运行。

RTC 内的时间计数器由 32.768 kHz 时钟计时，且可将该时钟提供给其他外设。32.768 kHz 时钟只能使用振荡器从外部晶振获取，该振荡器是 RTC 模块的一部分。

RTC 包含自己独有的模拟 POR 模块，每当 RTC 模块上电且将所有 RTC 寄存器初始化为默认状态时，生成 VBAT 上电复位信号。软件复位也可初始化所有 RTC 寄存器。RTC 也会监控芯片电源，并在芯片掉电时与其他部分保持隔离。

注

除访问控制寄存器外，访问 RTC 寄存器的尝试在以下情况下导致总线错误：

- VBAT 掉电，
- RTC 电隔离，或者
- VBAT POR 置位。

要确定 VBAT 域是否激活，软件可使用 ADC，通过内部连接测量 VBAT。请参见芯片特性 ADC 信息以了解 ADC 输入通道分配。

40.4.1.1 振荡器控制

32.768 kHz 晶振在 VBAT POR 下停用，必须通过软件使能。使能晶振后，等待振荡器启动时间，然后设置 SR[TCE]或使用 RTC 外部的振荡器时钟。

晶振包括可调谐电容，可通过软件配置。除非振荡器停用，否则请勿更改电容。

40.4.1.2 软件复位

对 CR[SWR]强制使 RTC 模块剩余部分等同于一次 VBAT POR。CR[SWR]不受软件复位影响，必须通过软件清零。访问控制寄存器不受 VBAT POR 活软件复位影响；它们通过芯片复位来复位。

40.4.1.3 管理员访问

管理员访问控制位清零时，仅管理员模式软件可写入 RTC 寄存器，非管理员模式软件会产生总线错误。管理员和非管理员模式软件始终可读取 RTC 寄存器。

40.4.2 时间计数器

时间计数器包含每秒递增一次的 32 位秒计数器，以及每个 32.768 kHz 时钟周期递增一次的 16 位预分频器寄存器。

在时间计数器（秒或预分频器）递增时进行读取可因读取数据总线同步而返回无效数据。如果软件需要在预分频器或秒计数器递增时进行读取，建议执行两次读取访问，且通过软件验证两次读取返回了相同数据。

时间秒寄存器和时间预分频器只能在 SR[TCE]清零时写入。必须在写入秒寄存器前写入预分频器寄存器，因为秒寄存器在预分频器寄存器位 14 的下降沿递增。

时间预分频器寄存器的递增前提是 SR[TCE]置位，SR[TIF]清零，SR[TOF]清零，且 32.768 kHz 时钟源存在。使能振荡器后，等待振荡器启动时间，然后设置 SR[TCE]，使得振荡器时钟输出的时间稳定。

如果时间秒寄存器溢出，则 SR[TOF]将置位，且时间预分频器寄存器停止递增。通过初始化时间秒寄存器将 SR[TOF]清零。一旦 SR[TOF]置位，时间秒寄存器和时间预分频器寄存器读取为零。

SR[TIF]在 VBAT POR 和软件复位中置位，并通过初始化时间秒寄存器清零。一旦 SR[TIF]置位，时间秒寄存器和时间预分频器寄存器读取为零。

40.4.3 补偿

补偿逻辑提供精确和宽广的补偿范围，并可校正高至 3906 ppm、低至 0.12 ppm 的错误。补偿因子必须在 RTC 外部计算，由软件提供给补偿寄存器。RTC 本身不会计算所需的补偿数量，尽管 1 Hz 会被输出至外部引脚，以支持外部校准逻辑。

晶振补偿可使用固件和晶振特征支持，以决定补偿数量。温度补偿可通过固件支持，后者通过 ADC 周期性地测量外部温度，并基于查找表更新补偿寄存器，该查找表指定晶振频率随温度的变化而变化。

补偿逻辑修改所用的 32.768 kHz 时钟周期数，以便预分频器寄存器溢出并递增时间秒计数器。时间补偿值用于调整时钟周期数，范围为-127 至+128。预分频器寄存器等于 0x3FFF 时对预分频器寄存器添加或减去周期，然后递增。补偿间隔用于调整使用时间补偿值的频率，即从每秒一次至每 256 秒一次。

对时间补偿寄存器的更新不会生效，直至下一次时间秒寄存器递增，前提是上一个补偿间隔已过期。当补偿间隔设置为非每秒一次时，补偿应用于第一秒间隔中，剩余秒间隔未接收补偿。

通过将时间补偿寄存器配置为零停用补偿。

40.4.4 时间警报

时间警报寄存器(TAR)、SR[TAF]和 IER[TAIE]允许 RTC 在预定时间生成中断。32 位 TAR 与每次递增的 32 位秒寄存器进行比较。SR[TAF]将在 TAR 等于 TSR 及 TSR 递增时置位。

SR[TAF]通过写入 TAR 清零。这通常是下一个警报值，尽管写入小于 TSR 的值，例如 0，会阻止 SR[TAF]再次置位。否则，SR[TAF]无法停用，尽管其生成的中断是通过 IER[TAIE]使能或停用。

40.4.5 更新模式

控制寄存器(CR[UM])内的更新模式字段配置对时间计数器使能(SR[TCE])字段的软件写入访问。CR[UM]清零后，SR[TCE]只能在 LR[SRL]置位时写入。CR[UM]置位后，如果 SR[TCE]清零，或者 SR[TIF]或 SR[TOF]置位，也可写入 SR[TCE]。这样一旦时间无效便可初始化时间秒和预分频器寄存器，同时防止时间秒和预分频器寄存器动态更改。LR[SRL]置位后，CR[UM]对 SR[TCE]无效果。

40.4.6 寄存器锁定

锁定寄存器(LR)可用于阻挡对特定寄存器的写入访问，直至下一次 VBAT POR 或软件复位。锁定控制寄存器(CR)将停用软件复位。锁定 LR 将阻挡 LR 的未来更新。

对锁定寄存器的写入访问被忽略，且不会生成总线错误。

40.4.7 访问控制

读取访问和写入访问寄存器在芯片电源域内实施，在芯片复位中复位。它不受 VBAT POR 或软件复位影响。它们用于阻挡对各寄存器的读取或写入访问，直至下一次芯片系统复位。访问受阻时，VBAT 电源内看不到总线访问，且不会生成总线错误。

40.4.8 中断

每当状态标志和相应的中断使能位同时置位时，RTC 中断置位。在 VBAT POR 和软件复位条件下，以及当 VBAT 电源关闭时，RTC 中断始终置位。通过使能芯片特性 RTC 时钟选通控制位，可在芯片级使能 RTC 中断。RTC 中断可以用于从任意低功耗模式唤醒芯片。如果 RTC 唤醒引脚使能且芯片掉电，RTC 中断会导致唤醒引脚置位。

可选 RTC 秒中断是一种边沿敏感型中断，有每秒生成一次的专门中断向量并且无需软件开销（没有需要清零的对应状态标志）。通过时间秒中断使能位可以在 RTC 中使能 RTC 秒中断，通过设置芯片特性 RTC 时钟选通控制位则可在芯片级使能 RTC 秒中断。RTC 秒中断不会引起 RTC 唤醒引脚置位。此中断为可选中断，不能在所有器件上实施。

第 41 章 通用串行总线全速 OTG 控制器 (USBFSOTG)

41.1 此模块的芯片实现细节

41.1.1 通用串行总线 (USB) FS 子系统

USB FS 子系统包括这些组件:

- 两用 USB OTG 功能 (On-The-Go) 控制器, 支持全速 (FS) 从设备或 FS/LS 主机。该模块符合 USB 2.0 规范。
- USB 收发器, 其 D+ 和 D- 线路上包含内部 15 k Ω 下拉电阻, 可用于主机模式功能。
- IRC48 带时钟恢复模块, 无需 48MHz 晶振。这仅可用于 USB 从设备模式。
- 连接可配置, 可将任何 UART 发送与接收引脚连接到全速 USB 物理层。

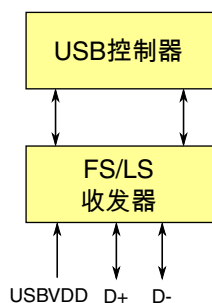


图 41-1. USB FS/LS 子系统概述

注

使用以下代码序列选择 USB 时钟源、USB 时钟分频比, 并使能其时钟门控, 以免引起可能会导致 USB 枚举级故障的潜在时钟毛刺。

1. 通过配置 SIM_SOPT2 选择 USB 时钟源。
2. 通过配置 SIM_CLKDIV2 选择所需的时钟分频比。

3. 通过置位 SIM_SCGC4 使能 USB 时钟门控。

41.1.1.1 FS/LS USB OTG 实例化

器件不含独立的 VBUS 检测信号。要在器件模式下检测有效的 VBUS，需要使用 GPIO。

USB 收发器，其 D+和 D-线路上包含 15 k 下拉电阻，可用于主机模式功能。

41.1.1.1.1 USB 唤醒功能

支持以下 USB 唤醒功能：

- 在所有功耗模式下检测到 VBUS 时唤醒。如果已通过外部调节器连接 VBUS 和 USBVDD 输入，则其可用于检测停止 /VLPS 模式下的连接或断开事件（或利用 LLWU 输入检测 LLS/VLLS3/VLLS2）。如果 USBVDD 是通过电路板而非 VBUS 供电，则此功能应使用 GPIO 引脚。应使用电阻分频器使检测输入电压保持在 GPIO 的有效输入范围内。
- DP/DM 上的任何活动都会将系统从低功耗模式唤醒，但 USB 不通电的 LLS/VLLS 模式除外。

41.1.1.1.2 通过 USB 供电的 UART 的性能

本器件支持一种连接，UART（通过 SIM_MISCCTL 配置）可借此连接到 FS USB 物理层（USB 连接器已断开），从而使不能直接访问 UARTx/LPUARTx 的应用具备简单的调试功能。有关控制位的更多详情，请参考 USB 和 SIM 章节的相关内容。

通过在 USBx_USBCTRL 寄存器中设置 UARTSEL 位，在 SIM_MISCCTL 寄存器中设置 UARTSELONUSB 位，其中一个 UART（UART0、1、2 或 LPUART0）会连接至 FS USB 物理层。这样会将 FS USB DP/DM 信号配置为常规 UART 信号，它不会在差分模式下运行。当 USBx_USBCTRL 寄存器中的 UARTCHLS 置位为 1'b0 时，FS USB DP/DM 信号引脚将用作 UART TX/RX。当 USBx_USBCTRL 寄存器中的 UARTCHLS 置位为 1'b1 时，FS USB DP/DM 信号引脚将用作 UART RX/TX。

注

USBVDD 必须连接电源才能使用通过 USB 的 UART 功能。

41.1.1.2 USB 唤醒

当 USB 检测到 USB 总线在超过 3 ms 无活动时，INT_STAT[SLEEP]位将置位。此位可导致中断，软件会确定相应的操作。

通过 USB 总线上的活动触发异步中断，从低功耗模式下唤醒 (USB 不通电的 LLS/VLLS 模式除外)。置位 USBTRC0[USBRESMEN]位会使能此功能。

此外，也支持以下唤醒功能：

- 在停止/VLPS 模式下，USB 控制器可在检测到 USBVDD 时生成中断。
- 在 LLS/VLLS 模式 (VLLS1/0 除外) 下，USBVDD 是 LLWU 的输入引脚，在其上转换能够生成一个唤醒。
- 在 LLS/VLLS 中，USB0_DP 和 USB0_DM 是 LLWU 的输入引脚，USB 已通电且处于主机模式时，在这些引脚上转换能够生成一个唤醒。

41.1.1.3 USB 电源分配

芯片具有一个独立的 USBVDD 供电引脚，为 USB 收发器供电。USBVDD 可接收 USB 电压等级(3.3V)供电，而 MCU 主电源引脚则可接受器件的整个操作范围供电。

41.1.1.3.1 AA/AAA 电池供电

芯片可由两节 AA/AAA 电池供电。在这种情况下，MCU 由 VDD 供电 (1.8-3.0 V)。检测到 USB 线缆插入之后，片外调节器使能，为 USB 收发器供电。

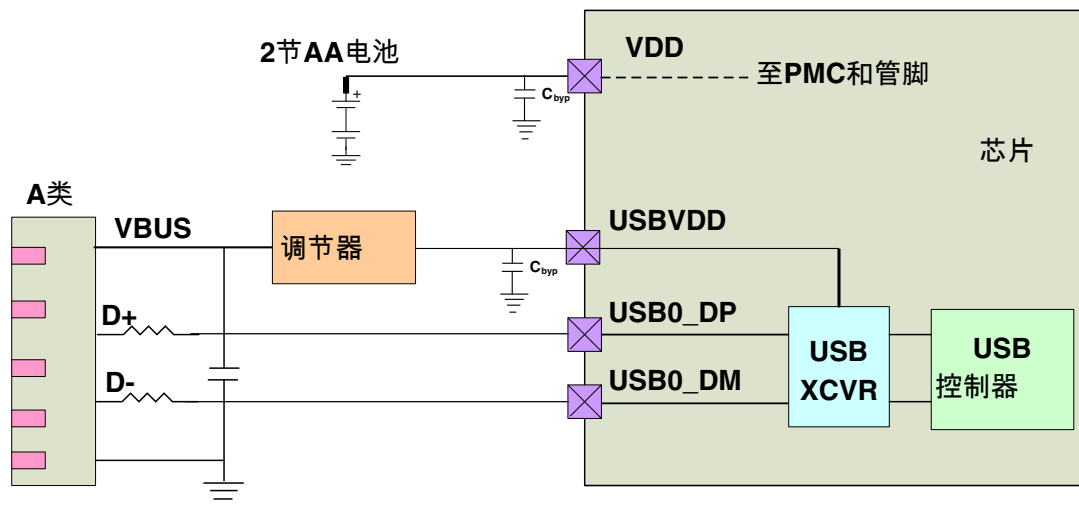


图 41-2. USB 调节器 AA 电池使用案例

41.1.1.3.2 锂离子电池供电

芯片也可由单个锂离子电池供电。在这种情况下，调节器提供 VDD 和 USBVDD 电源输入。连接到 USB 主机上时，调节器的输入源从锂离子电池切换为 USB 总线电源。

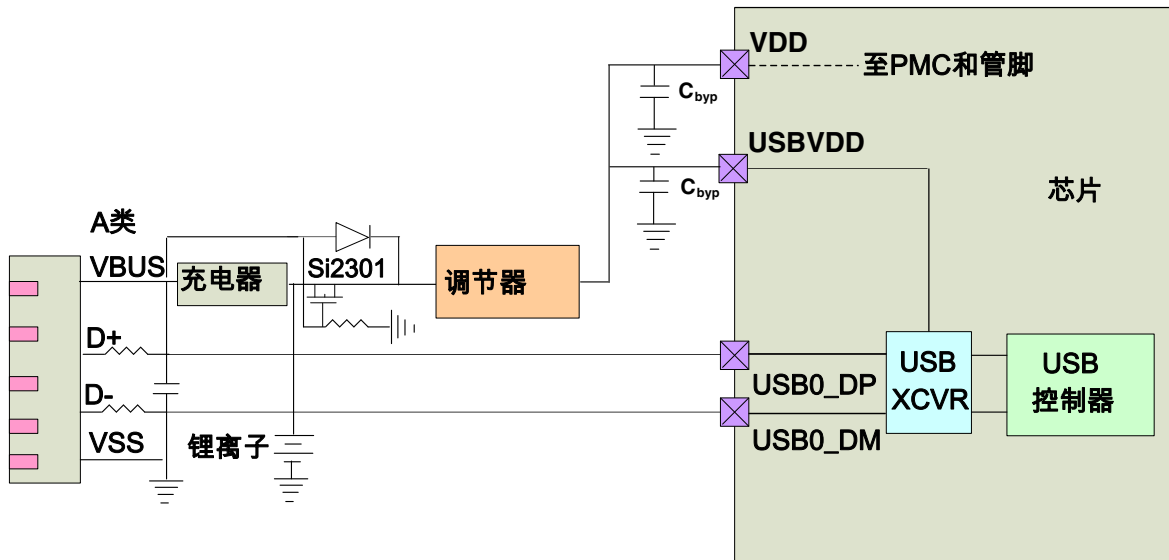


图 41-3. USB 调节器锂离子电池使用案例

41.1.1.3.3 USB 总线电源

芯片也可直接由 USB 总线供电。在这种情况下，调节器提供 VDD 和 USBVDD 电源输入。

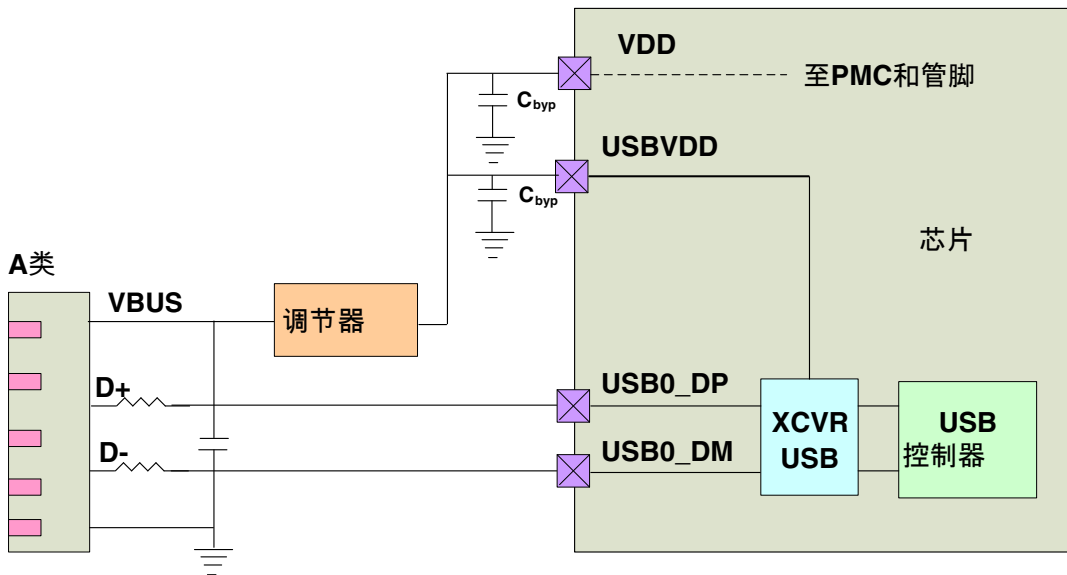


图 41-4. USB 调节器总线供电

41.1.1.4 USB 控制器配置

本节总结如何在芯片上配置模块。

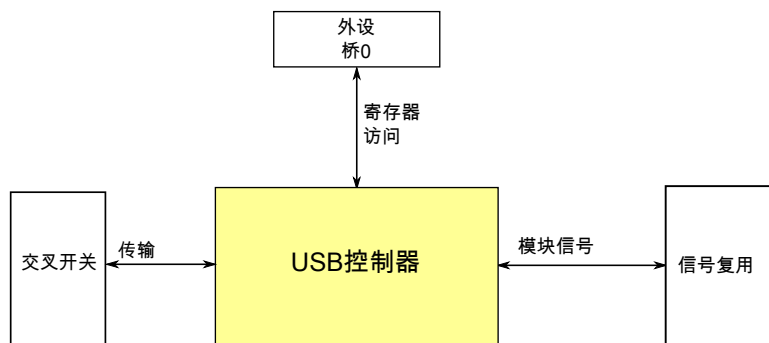


图 41-5. USB 控制器配置

表 41-1. 相关信息的参考链接

主题	相关模块	参考
完整说明	USB 控制器	USB 控制器
系统存储器映射		系统存储器映射
时钟		时钟分布
传输	交叉开关	交叉开关
信号复用	端口控制	信号复用

注

当应用中不使用 USB 时，建议将 USB 电源管脚，USBVDD，通过 10 kΩ 电压接地，不建议保留该引脚浮空。

41.2 简介

本章说明 USB 全速 OTG 控制器。此模块中的 OTG 设置提供有限的主机功能和从设备解决方案，以使用 USB 2.0 全速/低速兼容外设。OTG 逻辑实施 *On-The-Go* 和嵌入式主机 USB 2.0 规范附录(usb.org, 2008)所需的特性。USB 全速控制器接入 USBFS/LS 收发器。

注

本章说明名称相似的下列寄存器：USB_OTGCTL、USB_CTL、USB_CTRL、和 USB_CONTROL。这些都是独立寄存器。

41.2.1 参考资料

本文引用了下列出版资料。有关这些规则的副本或更新，请参见 USB Implementers Forum, Inc. 的网站 <http://www.usb.org>。

- *Universal Serial Bus Specification*, 修订版 2.0, 2000, 包括下列修订
- *Errata for “USB Revision 2.0 April 27, 2000” as of 12/7/2000*
- *Errata for “USB Revision 2.0 April 27, 2000” as of May 28, 2002*
- *Pull-up / Pull-down Resistors* (USB 设计更改说明)
- *Suspend Current Limit Changes* (USB 设计更改说明)
- *Device Capacitance* (USB 设计更改说明)
- *USB 2.0 Connect Timing Update* (USB 设计更改说明, 2013 年 4 月 4 日)
- *USB 2.0 VBUS Max Limit* (USB 设计更改说明)
- *On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification*, 修订版 2.0, 版本 1.1a, 2012 年 7 月 27 日
- *Maximum VBUS Voltage* (USB OTG/EH 设计更改说明)
- *Universal Serial Bus Micro-USB Cables and Connectors Specification*, 修订版 1.01, 2007 年

41.2.2 USB—概述

USB 是一个电缆总线，支持主机计算机和各种可同时访问的外设之间的数据交换。连接的外设通过基于令牌的主机调度协议共享 USB 带宽。在主机和其他外设运行时，可通过该总线连接、配置、使用和断开外设。

USB 软件可以为所有应用程序软件提供统一的系统视图，并且隐藏实施详细信息，从而使应用程序软件更加便携。它用于管理外设的动态连接和断开。

任何 USB 系统上仅有一个主机。至主机计算机系统的 USB 接口被称为主机控制器。

任何系统中均存在多个 USB 器件，例如人机接口器件、扬声器、打印机等。USB 从设备具有广泛，响应和标准功能的 USB 接口。

主机发起至特定外设的事务，而从设备则作出响应以控制各个事务。从设备使用标准 USB 数据格式接收来自主机的数据或将数据发送至主机。USB 2.0 全速/低速外设以 12 Mbit/s 或 1.5 Mbit/s 速率运行。

有关其他信息，请参见 USB 2.0 规范。

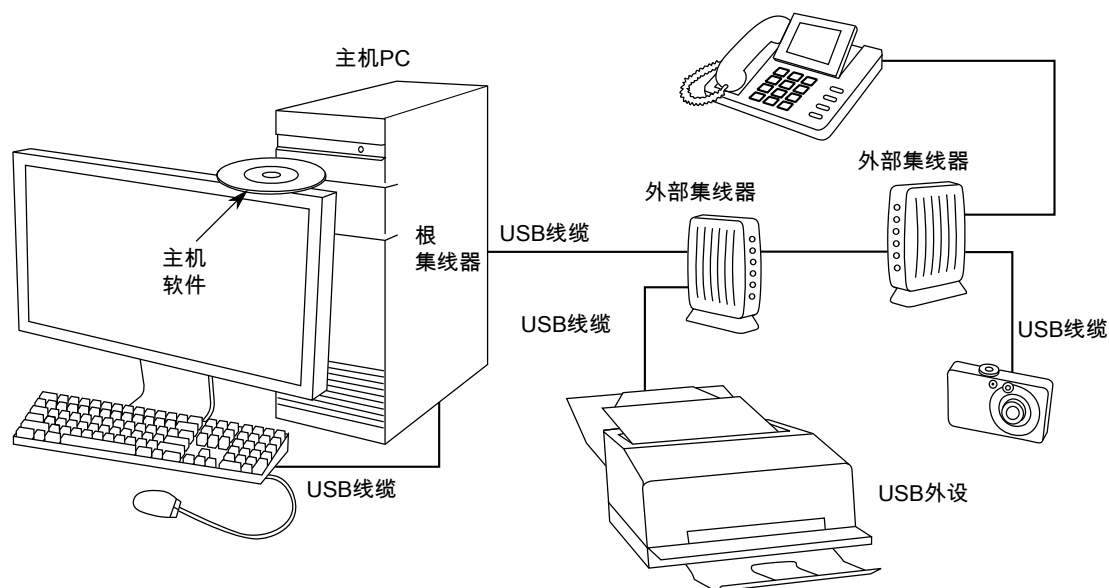


图 41-6. USB 2.0 系统配置示例

41.2.3 USB OTG

USB 是一种用于将外设和便携式消费电子器件（如数字摄像头和平板电脑）连接至主机 PC 的常用标准。USB On-The-Go (OTG)规范补充将 USB 扩展到点对点应用。使用 USB OTG 技术、消费电子、外设和便携式器件可以相互连接，以交换数据。例如，数字摄像头可以直接连接至打印机，或者键盘可以连接至平板电脑以交换数据。

借助 USB On-The-Go 产品，用户可以部署一个完成兼容 USB 的外设器件，这可假定 USB 主机角色。软件根据硬件信号确定器件的角色，然后再根据其连接方式在相应的操作模式（主机或外设）下初始化器件。连接之后，器件可以根据要完成的任务使用 OTG 协议交换以假定主机角色或外设。

有关其他信息，请参见 *On-The-Go* 和嵌入式主机 *USB 2.0* 规范附录。

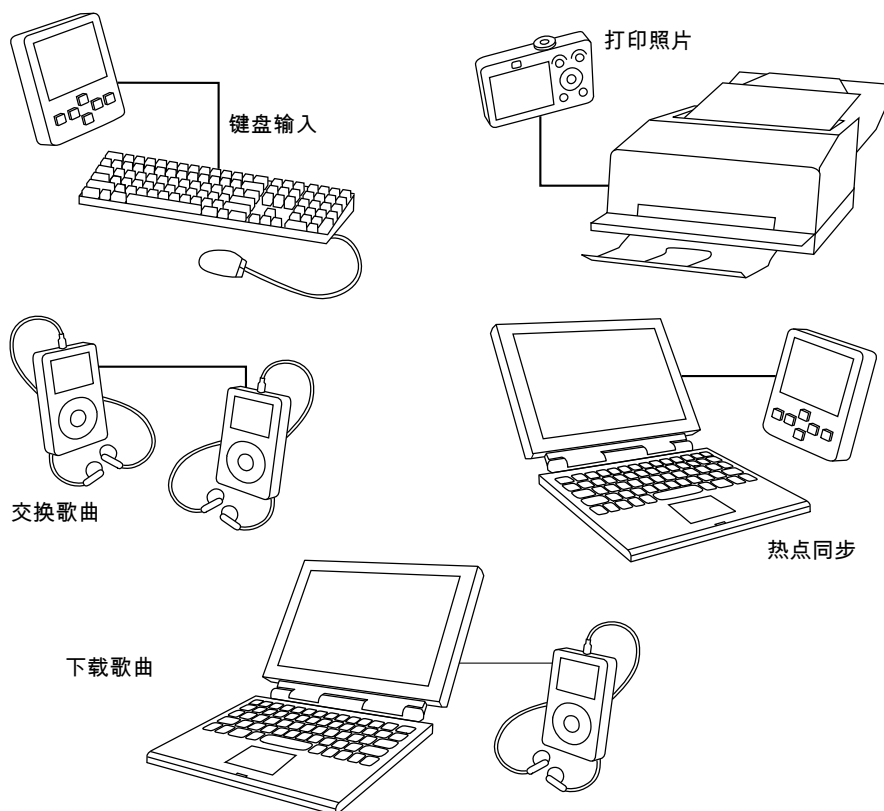


图 41-7. USB 2.0 On-The-Go 配置示例

41.2.4 USBFS 特性

- 符合 USB 1.1 和 2.0 的全速器件和带 On-The-Go 协议逻辑的 FS/LS 主机控制器
- 16 个双向端点
- DMA 或 FIFO 数据流接口
- 低功耗
- 支持带时钟恢复功能的 IRC48M，无需 48 MHz 晶体。仅限用于 USB 从设备设置。

41.3 功能说明

USBOTG 通过状态寄存器、控制寄存器和存储器中的数据结构与处理器内核通信。

41.3.1 数据结构

为了高效地管理 USB 端点通信，USBFS 使用系统存储器中的缓冲区描述符表 (BDT)。请参见图 41-11。

41.3.2 需要外部组件的片上收发器

USB 系统操作需要外部组件，以确保符合驱动器输出阻抗、眼图和 VBUS 线缆的容错要求。DM 和 DP I/O 管脚必须通过串联电阻（每个约 33 Ω ）连到应用印刷电路板 (PCB) 上的 USB 连接器。此外，如果这些 33 Ω 电阻安装位置相较于 USB 板级连接器更靠近处理器，则信号质量能得到优化。USB 收发器包括：

- 适合全速器件的 USB_DP 线路上的内部 1.5 k Ω 上拉电阻（由 USB_CONTROL[DPPULLUPNONOTG] 或 USB_OTGCTL[DPHIGH]控制）
- USB_DP 和 USB_DM 信号上的内部 15 k Ω 下拉电阻主要用于主机工作模式，但对从设备模式也有用，如下所述。

注

对于从设备操作，当 VBUS 检测软件确定 USB 连接未激活时（包括无线缆连接至主机的情况），内部 15 k Ω 下拉电阻应使能，确保 DP 和 DM 口在一个已知的静止状态。在此情况下，内部 15 k Ω 下拉电阻应由 USB_CTRL[PDE]控制。

对于主机操作，当 USB_CTL[HOSTMODEEN]认定为高电平时，依据 USB 2.0 规范的要求，内部 15 k Ω 下拉电阻应自动使能。

下图分别显示仅主机、仅从设备以及双角色连接的概览。图中的引脚连接仅作为示例。VDD 供电电压可以在该 SOC 的器件数据手册中提及的范围内选择，而 USBVDD 供电必须是 USB 标准，3.3 v。更多详情，请参见 *Kinetis* 外围模块快速参考 (KQRUG)。

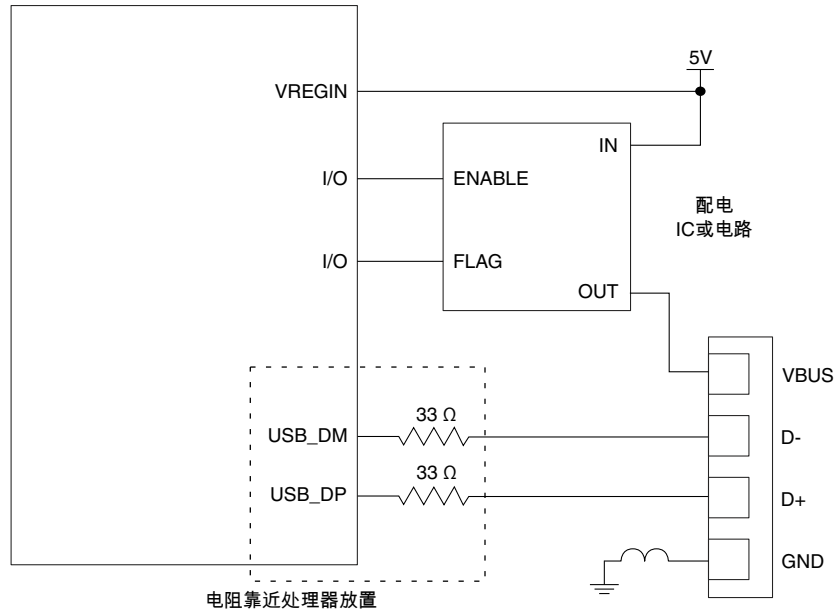


图 41-8. 仅主机图表

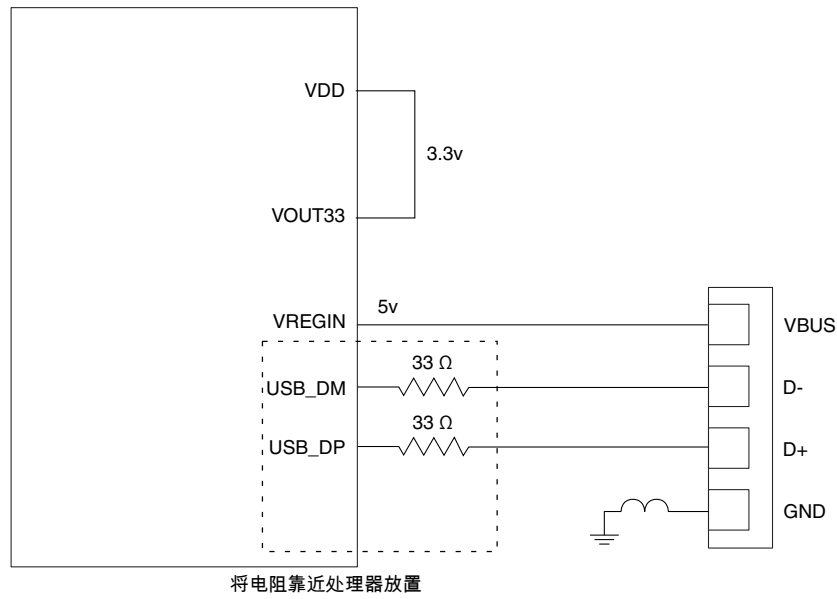


图 41-9. 典型的仅从设备框图（总线供电）

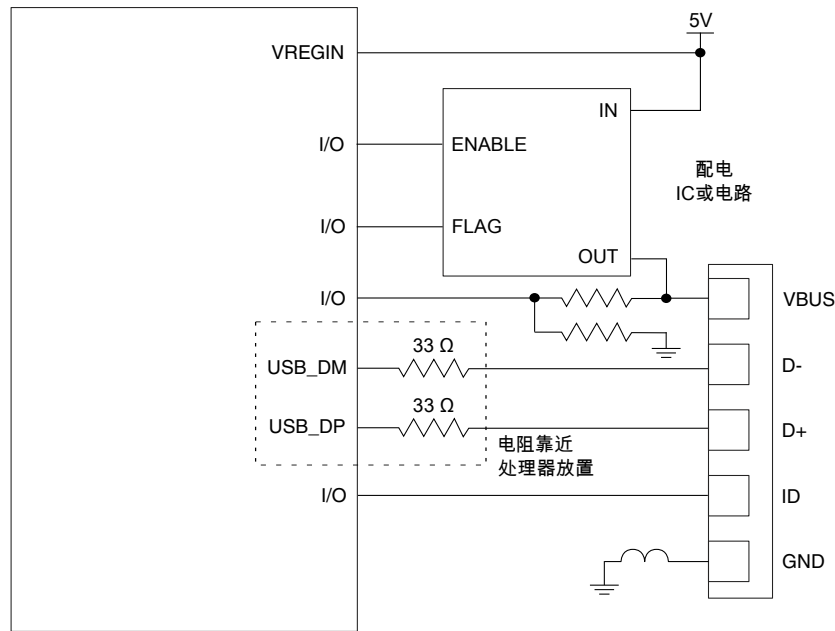


图 41-10. 双角色图

41.4 程序员接口

本节讨论 USB 模块编程模型的主要组件。

41.4.1 缓冲区描述符表

为了有效地管理 USB 端点通信，USBFS 使用系统存储器中的缓冲区描述符表 (BDT)。BDT 位于系统存储器中的 512 字节边界上，由 BDT 页面寄存器指向。每个端点方向需要两个 8 字节缓冲区描述符 (BD) 条目。因此，带 16 个完整双向端点的系统需要系统存储器的 512 字节才能实施 BDT。两个 BD 条目允许每个端点方向具有一个 EVEN BD 和 ODD BD 条目。这允许微处理器处理一个 BD，同时 USBFS 处理另一个 BD。以此方式双重缓冲 BD 使得 USBFS 能够以 USB 提供的最大吞吐量来轻松传输数据。

软件应通过更新 BDT (需要时) 管理 USBFS 的缓冲区。这使得 USBFS 能够有效地管理数据传输和接收，同时微控制器可以执行通信处理和其它相关的应用。由于缓冲区在微控制器和 USBFS 之间共享，因此，使用简单的信号量机制来区分谁有权更新系统存储器中的 BDT 和缓冲区。当 BD 条目为微控制器所有时，信号量 OWN 位将被清零。当 OWN 位为 0 时，允许微控制器读取和写入访问系统存储器中的 BD 条目和缓冲区。当 OWN 位设为 1 时，系统存储器中的 BD 条目和缓冲区为

USBFS 所有。USBFS 现在具有完整的读取和访问权限，并且微控制器不得修改 BD 或其对应的数据缓冲区。BD 还包含间接地址指针，其中，实际缓冲区位于系统存储器中。该间接地址机制如下图所示。

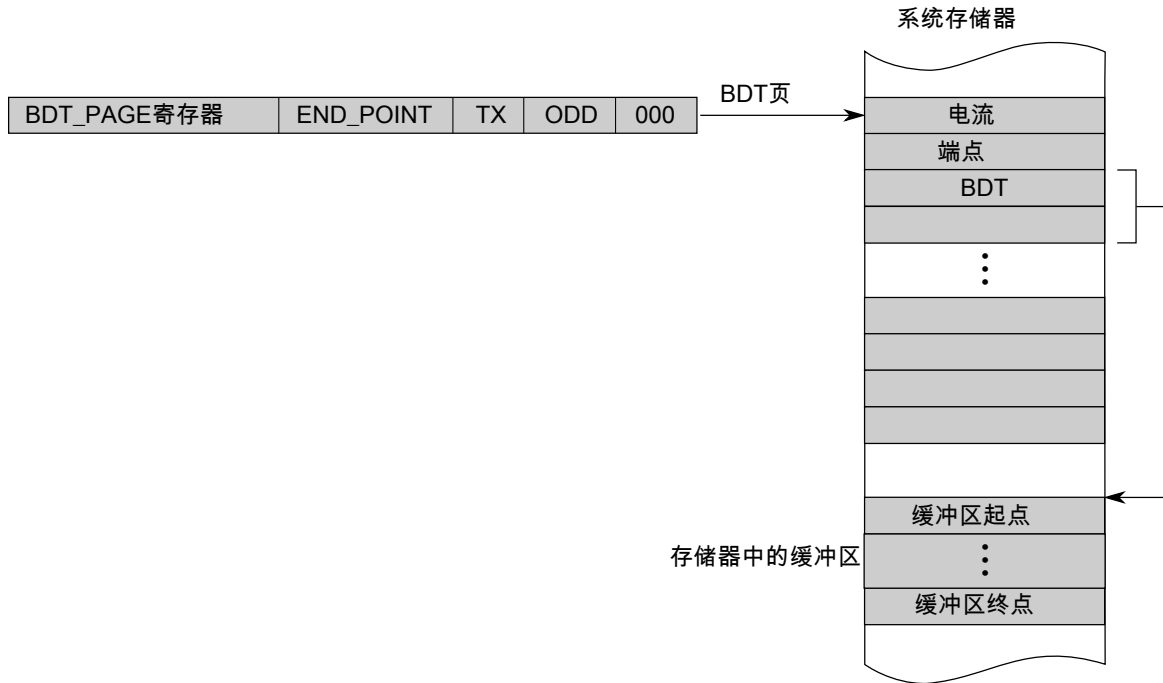


图 41-11. 缓冲区描述符表

41.4.2 RX 与 TX 用作 USB 外设或 USB 主机

USBFS 内核使用软件控制来在两种操作模式之间切换：

- USB 外设
- USB 主机

在任一模式下 (USB 主机或 USB 外设)，相同的数据路径和缓冲区描述符用于数据传输和接收。为此，使用 USBFS 内核中心命名法来描述 USBFS 内核与 USB 之间的数据传输方向：

- "RX" (或“接收”) 描述的是从 USB 向存储器传输数据。
- "TX" (或“发送”) 描述的是从存储器向 USB 传输数据。

下表介绍了数据方向与主机和外设应用中 USB 令牌类型是如何对应的。

表 41-2. USB 主机或 USB 外设的数据方向

	RX	TX
器件	OUT 或 SETUP	在
主机	在	OUT 或 SETUP

41.4.3 寻址 BDT 条目

理解缓冲区描述符表的寻址机制在通过 USBFS 或微控制器访问端点时非常有用。部分关注点如下：

- BDT 占用多达 512 字节的系统内存。
- 支持 16 个双向端点（带完整 512 字节 BDT）。
- 每个 USB 端点方向需要 16 字节。
- 少于 16 个端点的应用实施 BDT 所需的 RAM 更少。
- 至 BDT 起始位置的 BDT 页面寄存器 (BDT_PAGE) 点。
- BDT 必须位于系统存储器内的 512 字节边界上。
- 所有已使能的 TX 和 RX 端点 BD 条目索引至 BDT，以通过 USBFS 或 MCU 内核轻松访问。

接收到已使能端点上的 USB 令牌时，USBFS 使用其集成的 DMA 控制器询问 BDT。USBFS 读取对应的端点 BD 条目，以确定系统存储器中是否拥有自己的 BD 以及对应的缓冲区。

为了计算 BDT 中的入口点，必须将 BDT_PAGE 寄存器与当前端点和 TX 和 ODD 字段级联起来，以形成 32 位地址。该地址机制如下表所示：

表 41-3. BDT 地址计算

31:24	23:16	15:9	8:5	4	3	2:0
BDT_PAGE_03	BDT_PAGE_02	BDT_PAGE_01[7:1]	端点	TX	ODD	000

表 41-4. BDT 地址计算字段

字段	说明
BDT_PAGE	控制寄存器模块中的 BDT_PAGE 寄存器
端点	USB 令牌中的端点字段
TX	发送传输为 1，接收传输为 0
ODD	维持在 USBFS SIE 内。它与当前使用的缓冲区对应。这些缓冲区以乒乓方式使用。

41.4.4 缓冲区描述符 (BD)

缓冲区描述符 (BD) 提供 USBFS 和处理器的端点缓冲区控制信息。缓冲区描述符具有不同的意义，具体取决于其是 USBFS 或处理器读取存储器中的 BD。

USBFS 控制器使用存储在 BD 中的数据来确定：

- 谁拥有系统内存中的缓冲区
- 数据 0 或数据 1 PID
- 是否在数据包完成时释放所有权
- 无地址递增 (FIFO 模式)
- 是否在数据包完成时释放所有权
- 要发送或接收的数据量
- 缓冲区在系统内存中的位置

而处理器使用存储在 BD 中的数据来确定：

- 谁拥有系统内存中的缓冲区
- 数据 0 或数据 1 PID
- 接收到的 TOKEN PID
- 已发送或接收的数据量
- 缓冲区在系统内存中的位置

下图显示的是 BD 的格式。

表 41-5. 缓冲区描述符格式

31:26	25:16	15:8	7	6	5	4	3	2	1	0
RSVD	BC (10 bits)	RSVD	OWN	DATA0/1	KEEP/ TOK_PID[3]	NINC/ TOK_PID2	DTS/ TOK_PID[1]	BDT_STALL/ TOK_PID[0]	0	0
缓冲区地址 (32 位)										

表 41-6. 缓冲区描述符字段

字段	说明
31-26	保留

下一页继续介绍此表...

表 41-6. 缓冲区描述符字段 (继续)

字段	说明
RSVD	
25–16 BC	字节计数 表示 10 位字节计数。完成 RX 传输时，USBFS SIE 将使用接收到的数据字节计数更改此字段。
15–8 RSVD	保留
7 OWN	确定当前是处理器还是 USBFS 拥有该缓冲区。除 KEEP=1 时以外，SIE 在通过清除此位完成口令之后会将所有权还给处理器。 它必须始终是处理器初始化 BD 时更新的最后字节。 0 处理器可访问 BD。USBFS 将忽略 BD 中的所有其他字段。 1 USBFS 可访问 BD。当 USBFS 拥有 BD 时，处理器不应更改 BD 中的任何其他字段。
6 DATA0/1	定义 DATA0 字段 (DATA0/1=0) 或 DATA1 (DATA0/1=1) 字段是发送的还是接收的。USBFS 未对其进行更改。
5 KEEP/ TOK_PID[3]	此位有两种功能： <ul style="list-style-type: none"> KEEP 位 — 由处理器写入时，它用作 KEEP 位。通常情况下，此位为 1，且 ISO 端点传输到 FIFO。不会通知微处理器令牌已处理完毕，只会传输至或来自 FIFO 的数据。如已置位 KEEP，则一般也将置位 NINC 位，以防止地址递增。 当令牌处理完毕之后，允许 USBFS 释放 BD。 1 此位没有被 USBFS 更改。当前令牌 PID 的位 3 将被 USBFS 写回 BD。 TOK_PID[3] — 如果 OWN 位也已置位，则 BD 仍归 USBFS 无限期所有；由 USB 写入时，它被用作 TOK_PID[3] 位。 0 或 1 当前令牌 PID 的位 3 将被 USBFS 写回 BD。 通常情况下，此位为 1，且 ISO 端点传输到 FIFO。不会通知微处理器令牌已处理完毕，只会传输至或来自 FIFO 的数据。如已设置 KEEP，则一般也将设置 NINC 位，以防止地址递增。
4 NINC/ TOK_PID2	无递增 (NINC) 禁用 DMA 引擎地址递增。这将强制 DMA 引擎通过相同地址读取或写入。当需要读取来自单个位置 (如 FIFO) 的数据或将数据写入单个位置时，这对端点非常有用。一般情况下，此位将置位，且 ISO 端点的 KEEP 位与 FIFO 相连。 0 USBFS 将当前令牌 PID 的位 2 写入 BD。 1 此位没有被 USBFS 更改。
3 DTS/ TOK_PID[1]	置位此位将使能 USBFS 执行数据切换同步。 <ul style="list-style-type: none"> 如果 KEEP=0，则当前令牌 PID 的位 1 将被写回 BD。 如果 KEEP=1，则此位没有被 USBFS 更改。 0 数据切换同步已被禁用。 1 使能 USBFS 执行数据切换同步。
2 BDT_STALL TOK_PID[0]	如果令牌被使用此位置 BD 的 SIE 接收，则置位此位将导致 USBFS 发出 STALL 握手。当 BDT_STALL 位置位时，BDT 将由 SIE 占用 (OWN 位保持置位且其余 BDT 未更改)。 <ul style="list-style-type: none"> 如果 KEEP=0，则当前令牌 PID 的位 0 将被写回 BD。 如果 KEEP=1，则此位没有被 USBFS 更改。 0 未发出 STALL。 1 SIE 不占用 BDT (OWN 位保持置位且其余 BDT 未更改)。

下一页继续介绍此表...

表 41-6. 缓冲区描述符字段 (继续)

字段	说明
	置位 BDT_STALL 还会导致相应的 USB_ENDPTn[EPSTALL]位置位。这导致 USBOTG 为相关端点的两个方向发布 STALL 信号交换。要清除终止情况： <ol style="list-style-type: none"> 清除相关的 USB_ENDPTn[EPSTALL]位。 写入 BDT 以清除 OWN 和 BDT_STALL。
TOK_PID[n]	位[5:2]也可表示当前令牌 PID。传输完成之后，当前令牌 PID 将由 USBFS 写回 BD。写回的值为 USB 规格中的令牌 PID 值： <ul style="list-style-type: none"> 0x1h 是 OUT 令牌。 0x9h 是 IN 令牌。 0xDh 是 SETUP 令牌。 <p>在主机模式下，此字段用于报告最后返回的 PID 或传输状态指示。可能的返回值如下：</p> <ul style="list-style-type: none"> 0x3h DATA0 0xBh DATA1 0x2h ACK 0xEh STALL 0xAh NAK 0x0h 总线超时 0xFh 数据错误
1-0 保留	预留，因读取为 0。
ADDR[31:0]	地址 表示系统内存中的 32 位缓冲区地址。这些位未被 USBFS 更改。

41.4.5 USB 事务

当 USBFS 发送或接收数据时，它使用“寻址缓冲区描述符条目”表中所示的地址生成计算 BDT 地址。

如果 OWN =1，则发生以下流程：

1. USBFS 读取 BDT。
2. SIE 通过 DMA 传输数据至 ADDR BD 字段指向的缓冲区或传输来自该缓冲区的数。
3. 当 TOKEN 完整时，USBFS 更新 BDT，如果 KEEP=0，则将 OWN 位更改为 0。
4. 更新 STAT 寄存器并置位 TOK_DNE 中断。
5. 当处理器处理 TOK_DNE 中断时，它通过状态寄存器读取处理端点所需所有信息。
6. 此时，处理器将分配新的 BD，从而可以发送或接收该端点的其他 USB 数据，然后再处理最后的 BD。

下图显示了读取 BDT 且 OWN=1 之后如何处理典型 USB 令牌的时间线。

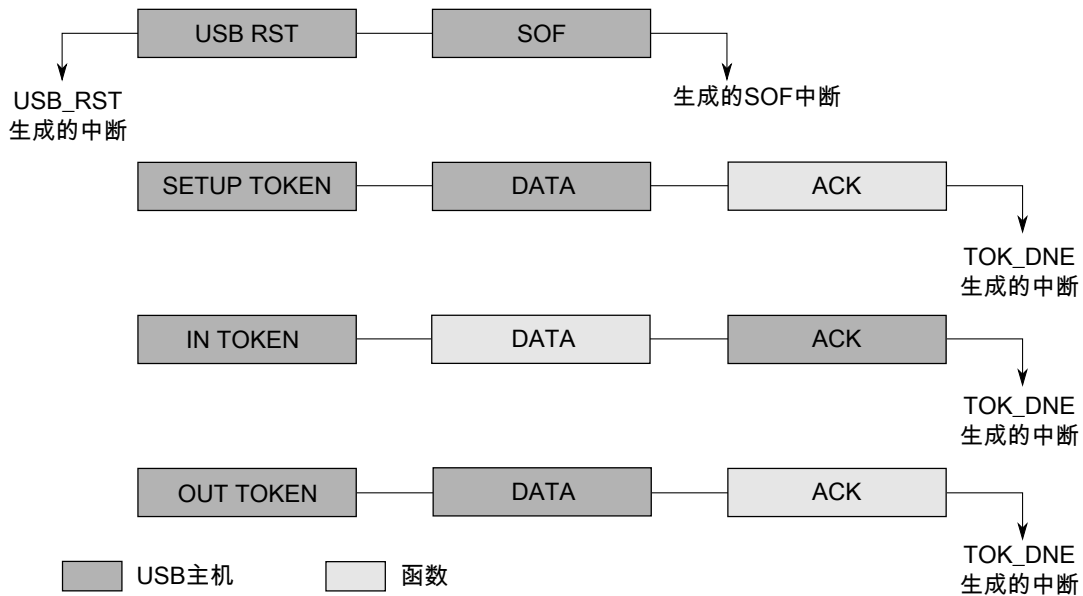


图 41-12. USB 令牌事务

USB 具有两个 DMA 溢出错误源：

存储器延迟

存储器延迟可能过高并导致接收 FIFO 溢出。这主要是硬件性能问题，通常由瞬态存储器访问问题所致。

过大数据包

接收到的数据包可能大于议定的 *MaxPacket* 大小。通常来说，这由软件 bug 所致。对于因过大数据包导致的 DMA 溢出错误，USB 规格不明确。它假定两侧具有正确的软件驱动程序。非应答数据包将导致重发过大数据包数据。因此，为了响应过大数据包，USB 内核继续应答数据包，以实现非同步传输。

表 41-7. USB 响应 DMA 溢出错误

因存储器延迟导致的错误	因数据包过大导致的错误
非应答(NAK)或总线超时 (BTO) — 请参见相应事务类“错误中断状态寄存器(ERRSTAT)”中的位 4。	继续应答 (ACKing) 数据包，以实现非同步传输。
—	写入存储器的数据将根据 <i>MaxPacket</i> 大小裁剪，以免损坏系统存储器。
在主机和器件操作模式下，ERRSTAT 寄存器中的 DMAERR 位将置位。根据 INTENB 和 ERRENB 寄存器的值，内核可能会置位中断，以通知 DMA 处理器错误。	置位 ERRSTAT[DMAERR]可能引起中断和 TOKDNE 中断触发。(注：BDT 的 TOK_PID 字段不是 1111，因为 DMAERR 不是因延迟所致。)
<ul style="list-style-type: none"> 对于主机模式，生成 TOKDNE 中断且 BDT 的 TOK_PID 字段为 1111 用以表示 DMA 延迟错误。主机模式软件可以决定是重试还是转至下一计划项目。 在从设备模式下，BDT 既不会被写回也不是由 TOKDNE 中断所触发，因为假定会进行第二次尝试并且将在未来取得成功。 	写回 BDT 的数据包长度字段为 <i>MaxPacket</i> 值，表示实际写入存储器的已裁剪数据的长度。
此时，软件可以确定未来事务相应的操作源，例如停止端点、取消传输、禁用端点等。	

41.5 存储器映射/寄存器定义

本节提供所有 USB 接口寄存器的存储映射和详细说明。

USB 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4007_2000	外设 ID 寄存器 (USB0_PERID)	8	不适用	04h	41.5.1/834
4007_2004	外设 ID 补码寄存器 (USB0_IDCOMP)	8	不适用	FBh	41.5.2/835
4007_2008	外设修订版寄存器 (USB0_REV)	8	不适用	33h	41.5.3/835
4007_200C	外设其他信息寄存器 (USB0_ADDINFO)	8	不适用	01h	41.5.4/836
4007_2010	OTG 中断状态寄存器 (USB0_OTGISTAT)	8	不适用	00h	41.5.5/836
4007_2014	OTG 中断控制寄存器 (USB0_OTGICR)	8	不适用	00h	41.5.6/837
4007_2018	OTG 状态寄存器 (USB0_OTGSTAT)	8	不适用	00h	41.5.7/838
4007_201C	OTG 控制寄存器 (USB0_OTGCTL)	8	不适用	00h	41.5.8/839
4007_2080	中断状态寄存器 (USB0_ISTAT)	8	不适用	00h	41.5.9/840
4007_2084	中断使能寄存器 (USB0_INTEN)	8	不适用	00h	41.5.10/841
4007_2088	错误中断状态寄存器 (USB0_ERRSTAT)	8	不适用	00h	41.5.11/842
4007_208C	错误中断使能寄存器 (USB0_ERREN)	8	不适用	00h	41.5.12/843
4007_2090	状态寄存器 (USB0_STAT)	8	不适用	00h	41.5.13/844
4007_2094	控制寄存器 (USB0_CTL)	8	不适用	00h	41.5.14/845
4007_2098	地址寄存器 (USB0_ADDR)	8	不适用	00h	41.5.15/846
4007_209C	BDT 页寄存器 1 (USB0_BDTPAGE1)	8	不适用	00h	41.5.16/846
4007_20A0	低帧号寄存器 (USB0_FRMNUML)	8	不适用	00h	41.5.17/847
4007_20A4	高帧号寄存器 (USB0_FRMNUMH)	8	不适用	00h	41.5.18/847
4007_20A8	令牌寄存器 (USB0_TOKEN)	8	不适用	00h	41.5.19/848
4007_20AC	SOF 阈值寄存器 (USB0_SOFTHLTD)	8	不适用	00h	41.5.20/848
4007_20B0	BDT 页寄存器 2 (USB0_BDTPAGE2)	8	不适用	00h	41.5.21/849
4007_20B4	BDT 页寄存器 3 (USB0_BDTPAGE3)	8	不适用	00h	41.5.22/850

下一页继续介绍此表...

USB 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4007_20C0	端点控制寄存器 (USB0_ENDPT0)	8	不适用	00h	41.5.23/ 850
4007_20C4	端点控制寄存器 (USB0_ENDPT1)	8	不适用	00h	41.5.23/ 850
4007_20C8	端点控制寄存器 (USB0_ENDPT2)	8	不适用	00h	41.5.23/ 850
4007_20CC	端点控制寄存器 (USB0_ENDPT3)	8	不适用	00h	41.5.23/ 850
4007_20D0	端点控制寄存器 (USB0_ENDPT4)	8	不适用	00h	41.5.23/ 850
4007_20D4	端点控制寄存器 (USB0_ENDPT5)	8	不适用	00h	41.5.23/ 850
4007_20D8	端点控制寄存器 (USB0_ENDPT6)	8	不适用	00h	41.5.23/ 850
4007_20DC	端点控制寄存器 (USB0_ENDPT7)	8	不适用	00h	41.5.23/ 850
4007_20E0	端点控制寄存器 (USB0_ENDPT8)	8	不适用	00h	41.5.23/ 850
4007_20E4	端点控制寄存器 (USB0_ENDPT9)	8	不适用	00h	41.5.23/ 850
4007_20E8	端点控制寄存器 (USB0_ENDPT10)	8	不适用	00h	41.5.23/ 850
4007_20EC	端点控制寄存器 (USB0_ENDPT11)	8	不适用	00h	41.5.23/ 850
4007_20F0	端点控制寄存器 (USB0_ENDPT12)	8	不适用	00h	41.5.23/ 850
4007_20F4	端点控制寄存器 (USB0_ENDPT13)	8	不适用	00h	41.5.23/ 850
4007_20F8	端点控制寄存器 (USB0_ENDPT14)	8	不适用	00h	41.5.23/ 850
4007_20FC	端点控制寄存器 (USB0_ENDPT15)	8	不适用	00h	41.5.23/ 850
4007_2100	USB 控制寄存器 (USB0_USBCTRL)	8	不适用	C0h	41.5.24/ 851
4007_2104	USB OTG 观察寄存器 (USB0_OBSERVE)	8	不适用	50h	41.5.25/ 852
4007_2108	USB OTG 控制寄存器 (USB0_CONTROL)	8	不适用	00h	41.5.26/ 853
4007_210C	USB 收发器控制寄存器 0 (USB0_USBTRC0)	8	不适用	00h	41.5.27/ 853
4007_2114	帧调整寄存器 (USB0_USBFRMADJUST)	8	不适用	00h	41.5.28/ 855
4007_212C	其他控制寄存器 (USB0_MISCCTRL)	8	不适用	00h	41.5.29/ 855

下一页继续介绍此表...

USB 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4007_2130	用于 IN 方向 7-0 端点的外设模式停止禁用 (USB0_STALL_IL_DIS)	8	不适用	00h	41.5.30/ 856
4007_2134	用于 IN 方向 15-8 端点的外设模式停止禁用 (USB0_STALL_IH_DIS)	8	不适用	00h	41.5.31/ 857
4007_2138	用于 OUT 方向 7-0 端点的外设模式停止禁用 (USB0_STALL_OL_DIS)	8	不适用	00h	41.5.32/ 859
4007_213C	用于 OUT 方向 15-8 端点的外设模式停止禁用 (USB0_STALL_OH_DIS)	8	不适用	00h	41.5.33/ 860
4007_2140	USB 时钟恢复控制 (USB0_CLK_RECOVER_CTRL)	8	不适用	00h	41.5.34/ 861
4007_2144	IRC48M 振荡器使能寄存器 (USB0_CLK_RECOVER_IRC_EN)	8	不适用	01h	41.5.35/ 862
4007_2154	时钟恢复组合中断使能 (USB0_CLK_RECOVER_INT_EN)	8	不适用	10h	41.5.36/ 863
4007_215C	时钟恢复分离中断状态 (USB0_CLK_RECOVER_INT_STATUS)	8	w1c	00h	41.5.37/ 864

41.5.1 外设 ID 寄存器 (USBx_PERID)

读回值 0x04。此值针对 USB 外设定义。

地址: 4007_2000h 基准 + 0h 偏移 = 4007_2000h

位	7	6	5	4	3	2	1	0
读	0		ID					
写	0		ID					
复位	0	0	0	0	0	1	0	0

USBx_PERID 字段描述

字段	描述
7-6 保留	此只读字段为保留字段且值始终为 0。
ID	外设标识 该字段始终读取为 0x4h。

41.5.2 外设 ID 补码寄存器 (USBx_IDCOMP)

读回外设 ID 寄存器的补码。对于 USB 外设，此值为 0xFB。

地址: 4007_2000h 基准 + 4h 偏移 = 4007_2004h

位	7	6	5	4	3	2	1	0
读	1		NID					
写	1		NID					
复位	1	1	1	1	1	0	1	1

USBx_IDCOMP 字段描述

字段	描述
7-6 保留	此只读字段为保留字段且值始终为 1。
NID	PERID[ID]位的一补码。

41.5.3 外设修订版寄存器 (USBx_REV)

包含 USB 模块的版本号。

地址: 4007_2000h 基准 + 8h 偏移 = 4007_2008h

位	7	6	5	4	3	2	1	0
读	REV							
写	REV							
复位	0	0	1	1	0	0	1	1

USBx_REV 字段描述

字段	描述
REV	修订版 指示 USB 内核的修订版本号。

41.5.4 外设其他信息寄存器 (USBx_ADDINFO)

读回主机使能位的值。

地址: 4007_2000h 基准 + Ch 偏移 = 4007_200Ch

位	7	6	5	4	3	2	1	0
读	0				0		IEHOST	
写							IEHOST	
复位	0	0	0	0	0	0	0	1

USBx_ADDINFO 字段描述

字段	描述
7-3 保留	此只读字段为保留字段且值始终为 0。
2-1 保留	此只读字段为保留字段且值始终为 0。
0 IEHOST	如果已使能主机模式，则置位此位。

41.5.5 OTG 中断状态寄存器 (USBx_OTGISTAT)

记录更改为一毫秒定时器和线路状态稳定逻辑的状态。软件可以读取此寄存器，以确定触发中断的事件。仅置位上次软件读取之后发生更改的位。向此位写入 1 将清除关联的中断。

地址: 4007_2000h 基准 + 10h 偏移 = 4007_2010h

位	7	6	5	4	3	2	1	0
读	保留	ONEMSEC	LINE_STATE_CHG	0	保留	保留	0	保留
写	w1c	w1c	w1c		w1c	w1c		w1c
复位	0	0	0	0	0	0	0	0

USBx_OTGISTAT 字段描述

字段	描述
7 保留	软件不得改变该位域的值。
6 ONEMSEC	当 1 毫秒定时器到期时，将置位此位。此位在被软件清除之前保持认定。必须每毫秒为中断提供服务，以免丢失 1 毫秒计数。

下一页继续介绍此表...

USBx_OTGISTAT 字段描述 (继续)

字段	描述
5 LINE_STATE_ CHG	如果 USB 线路状态 (CTL[SE0]和 CTL[JSTATE]位) 稳定, 1 毫秒无变化, 并且线路状态的值不同于上次线路状态稳定时的值, 则此中断置位。它在 SE0 和 J 状态间转换、SE0 和 K 状态间转换以及 J 状态和 K 状态间转换时置位。当 SE0 为真时, J 状态的改变不会引起中断。此中断可用于检测复位、恢复、连接和数据线路脉冲信号。
4 保留	此只读字段为保留字段且值始终为 0。
3 保留	软件不得改变该位域的值。
2 保留	软件不得改变该位域的值。
1 保留	此只读字段为保留字段且值始终为 0。
0 保留	软件不得改变该位域的值。

41.5.6 OTG 中断控制寄存器 (USBx_OTGICR)

使能 OTG 中断状态寄存器中定义的对应该中断状态位。

地址: 4007_2000h 基准 + 14h 偏移 = 4007_2014h

位	7	6	5	4	3	2	1	0
读	保留	ONEMSEC EN	LINESTATE EN	0	保留	保留	0	保留
写	保留	ONEMSECEN	LINESTATEEN		保留	保留		保留
复位	0	0	0	0	0	0	0	0

USBx_OTGICR 字段描述

字段	描述
7 保留	软件不得改变该位域的值。
6 ONEMSECEN	1 毫秒中断使能 0 禁用 1ms 定时器中断。 1 使能 1ms 定时器中断。
5 LINESTATEEN	线路状态变更中断使能 0 禁用 LINE_STAT_CHG 中断。 1 使能 LINE_STAT_CHG 中断。
4 保留	此只读字段为保留字段且值始终为 0。
3 保留	软件不得改变该位域的值。
2 保留	软件不得改变该位域的值。

下一页继续介绍此表...

USBx_OTGICR 字段描述 (继续)

字段	描述
1 保留	此只读字段为保留字段且值始终为 0。
0 保留	软件不得改变该位域的值。

41.5.7 OTG 状态寄存器 (USBx_OTGSTAT)

显示来自一毫秒定时器和线路状态稳定逻辑的实际值。

地址: 4007_2000h 基准 + 18h 偏移 = 4007_2018h

位	7	6	5	4
读	0	ONEMSECEN	LINESTATESTABLE	0
写		ONEMSECEN	LINESTATESTABLE	
复位	0	0	0	0
位	3	2	1	0
读	0	0	0	0
写				
复位	0	0	0	0

USBx_OTGSTAT 字段描述

字段	描述
7 保留	此只读字段为保留字段且值始终为 0。
6 ONEMSECEN	对于 1 ms 计数, 此位保留, 但对软件来说无作用。
5 LINESTATESTABLE	表示控制 OTGSTAT 的 LINE_STATE_CHG 字段的内部信号保持稳定至少 1 ms。当检测连接、断开和恢复信号时, 此位用于提供线路状态的硬件去抖。先读取 LINE_STATE_CHG 字段, 再读取此字段。如果此位读取为 1, 则 LINE_STATE_CHG 的值被视为稳定。 0 LINE_STAT_CHG 位尚未稳定。 1 LINE_STAT_CHG 位已去抖并且稳定。
4 保留	此只读字段为保留字段且值始终为 0。
3 保留	此只读字段为保留字段且值始终为 0。
2 保留	此只读字段为保留字段且值始终为 0。
1 保留	此只读字段为保留字段且值始终为 0。
0 保留	此只读字段为保留字段且值始终为 0。

41.5.8 OTG 控制寄存器 (USBx_OTGCTL)

控制 VBUS 和数据线路终端电阻的操作。

地址: 4007_2000h 基准 + 1Ch 偏移 = 4007_201Ch

位	7	6	5	4	3	2	1	0
读	DPHIGH	0	DPLow	DMLOW	0	OTGEN	0	
写	DPHIGH		DPLow	DMLOW		OTGEN		
复位	0	0	0	0	0	0	0	0

USBx_OTGCTL 字段描述

字段	描述
7 DPHIGH	D+数据线路上拉电阻使能 0 D+上拉电阻未使能 1 D+上拉电阻已使能
6 保留	此只读字段为保留字段且值始终为 0。
5 DPLow	D+数据线路下拉电阻使能 此位应始终与位 4 (DMLOW) 一起使能 0 D+下拉电阻未使能。 1 D+下拉电阻已使能。
4 DMLOW	D-数据线路下拉电阻使能 0 D-下拉电阻未使能。 1 D-下拉电阻已使能。
3 保留	此只读字段为保留字段且值始终为 0。
2 OTGEN	On-The-Go 上拉/下拉电阻使能 0 如果控制寄存器(CTL)中的 USB_EN 为 1 且 HOST_MODE 为 0, 则 D+数据线路上拉电阻使能。如果 HOST_MODE 为 1, 则 D+和 D-数据线路下拉电阻启用。 1 使用的是该寄存器中的上拉和下拉控制。
保留	此只读字段为保留字段且值始终为 0。

41.5.9 中断状态寄存器 (USBx_ISTAT)

包含 USB 模块中每个中断源的字段。每个字段受各自的中断使能位限制。该寄存器的所有字段按照 OR 逻辑与 OTG 中断状态寄存器 (OTGSTAT) 结合，以形成处理器中断控制器的一个中断源。中断位置位后，可通过向对应的中断位写入 1 进行清除。复位之后，该寄存器的值为 0x00。

地址: 4007_2000h 基准 + 80h 偏移 = 4007_2080h

位	7	6	5	4	3	2	1	0
读	STALL	ATTACH	RESUME	SLEEP	TOKDNE	SOFTOK	ERROR	USBRST
写	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
复位	0	0	0	0	0	0	0	0

USBx_ISTAT 字段描述

字段	描述
7 STALL	<p>停止中断</p> <p>在从设备模式下，当 SIE 发送 STALL 握手时将置位此位。</p> <p>在主机模式下，当 USB 模块在 USB 事务的握手阶段检测到 STALL 应答时，将置位此位。此中断可用于确定上个 USB 事务是已成功完成还是已停止。</p>
6 ATTACH	<p>连接中断</p> <p>当 USB 模块检测到连接 USB 器件时，将置位此字段。此字段仅在 CTL[HOSTMODEEN]=1 时有效。此中断表示现有一个外设必须配置；如果 USB 上持续 2.5 μs 无转换且当前状态不是 SE0，此中断将被认定。</p> <p>0 自上次 ATTACH 位清零以来，没有检测到连接。</p> <p>1 现有一个外设必须配置（检测到稳定的非 SE0 状态，持续时间超过 2.5 μs）。</p>
5 RESUME	<p>当在 DP/DM 信号上观察到 K 状态并持续 2.5 μs 时，将置位此位。如未处于挂起模式，必须禁用此中断。</p>
4 SLEEP	<p>当 USB 模块检测到 USB 总线持续闲置时间达 3 ms 时，将置位此位。USB 总线上的活动将会使此睡眠定时器复位。</p>
3 TOKDNE	<p>如果当前处理的令牌已完成，则置位此位。处理器必须立即读取状态 (STAT) 寄存器，以确定此令牌使用的端点和 BD。清除此位（通过写入 1）将导致 STAT 清零或 STAT 保持寄存器加载到 STAT 寄存器。</p>
2 SOFTOK	<p>当 USB 模块接收到起始帧 (SOF) 令牌时，将置位此位。</p> <p>在主机模式下，当达到 SOF 阈值时 (MISCCTRL[SOFBUSSET]=0)，或者当 SOF 计数器达到 0 (MISCCTRL[SOFBUSSET]=1)，将置位此字段，以便软件可以为下一个 SOF 做准备。</p>
1 ERROR	<p>当错误中断状态(ERRSTAT)寄存器内发生任何错误时，将置位此位。处理器随后必须读取 ERRSTAT 寄存器，以确定错误源。</p>
0 USBRST	<p>当 USB 模块收到一个有效的 USB 复位时，将置位此位。这将通知处理器应将 0x00 写入地址寄存器并置位使能端点 0。在检测到 USB 复位之后 2.5 毫秒，将置位 USBRST。直到消失并且重新产生 USB 复位条件，USBRST 才会再次置位。</p>

41.5.10 中断使能寄存器 (USBx_INTEN)

包含 USB 模块中每个中断源的使能字段。置位这些位中的任意一个将使能 ISTAT 寄存器中的相应中断源。复位之后，该寄存器的值为 0x00。

地址: 4007_2000h 基准 + 84h 偏移 = 4007_2084h

位	7	6	5	4	3	2	1	0
读	STALLEN	ATTACHEN	RESUMEEN	SLEEPEN	TOKDNEEN	SOFTOKEN	ERROREN	USBRSTEN
写	STALLEN	ATTACHEN	RESUMEEN	SLEEPEN	TOKDNEEN	SOFTOKEN	ERROREN	USBRSTEN
复位	0	0	0	0	0	0	0	0

USBx_INTEN 字段描述

字段	描述
7 STALLEN	STALL 中断使能 0 禁用 STALL 中断。 1 使能 STALL 中断。
6 ATTACHEN	ATTACH 中断使能 0 禁用 ATTACH 中断。 1 使能 ATTACH 中断。
5 RESUMEEN	RESUME 中断使能 0 禁用 RESUME 中断。 1 使能 RESUME 中断。
4 SLEEPEN	SLEEP 中断使能 0 禁用 SLEEP 中断。 1 使能 SLEEP 中断。
3 TOKDNEEN	TOKDNE 中断使能 0 禁用 TOKDNE 中断。 1 使能 TOKDNE 中断。
2 SOFTOKEN	SOFTOK 中断使能 0 禁用 SOFTOK 中断。 1 使能 SOFTOK 中断。
1 ERROREN	ERROR 中断使能 0 禁用 ERROR 中断。 1 使能 ERROR 中断。
0 USBRSTEN	USBRST 中断使能 0 禁用 USBRST 中断。 1 使能 USBRST 中断。

41.5.11 错误中断状态寄存器 (USBx_ERRSTAT)

包含 USB 模块中每个错误源的使能位。每个位受各自的错误使能位限制。该寄存器的所有位按逻辑 OR 结合，结果位于 ISTAT 寄存器的 ERROR 位中。中断位置位后，可通过向对应的中断位写入 1 进行清除。只要检测到错误情况，就会置位对应的位。因此，该中断通常不与处理的令牌末尾对应。复位之后，该寄存器的值为 0x00。

地址: 4007_2000h 基准 + 88h 偏移 = 4007_2088h

位	7	6	5	4	3	2	1	0
读	BTSERR	OWNERR	DMAERR	BTOERR	DFN8	CRC16	CRC5EOF	PIDERR
写	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
复位	0	0	0	0	0	0	0	0

USBx_ERRSTAT 字段描述

字段	描述
7 BTSERR	当检测到位填充错误时，将置位此位。置位后，对应的数据包将因错误遭拒。
6 OWNERR	当 USB 模块运行于外设模式 (CTL[HOSTMODEEN]=0) 时，该字段有效。如果 USB 模块要求新的 BD 以用于 SETUP、ISO IN 或 ISO OUT 传输但无新的 BD 可用，则该字段置位。
5 DMAERR	如果 USB 模块请求 DMA 访问读取新 BDT，但之前并未提供用于接收或发送数据的总线，则置位此位。如果处理 TX 传输，则这可能导致发送数据下溢情况。如果处理 RX 传输，则这可能会导致接收数据过溢情况。此中断在开发微控制器和 USB 模块的器件仲裁硬件时非常有用，可以最小化总线请求和总线许可延迟。如果至主机或来自主机的数据包大于 BDT 中分配的缓冲器尺寸，也会置位此位。在此情况下，在将数据包置于缓冲存储器时将其截断。
4 BTOERR	当发生总线响应超时错误时，将置位此位。USB 模块包含一个总线响应定时器，用于跟踪计量令牌与 SETUP 的数据段或者是 OUT TOKEN 或者是 IN TOKEN 的数据和握手段之间的时间。如果从 IDLE 转换之前的上一次 EOP 的计数超过 16 位时间，则会发生总线响应超时错误。
3 DFN8	如果接收的数据字段长度不是 8 位，则置位此位。USB 规范 1.0 要求数据字段为整数字节。如果数据字段不是整数字节，则置位此位。
2 CRC16	当数据包因 CRC16 错误遭拒时，将置位此位。
1 CRC5EOF	此错误中断具有两个功能。当 USB 模块在外设模式 (CTL[HOSTMODEEN]=0) 下运行时，此中断将检测到主机生成的令牌数据包中的 CRC5 错误。如果已置位，则令牌数据包将因 CRC5 错误遭拒。 当 USB 模块在主机模式 (CTL[HOSTMODEEN]=1) 下运行时，此中断将检测结束帧 (EOF) 错误。当 USB 模块正在发送或接收数据且 SOF 计数器达到 0 时会发生此情况。。此中断在开发 USB 数据包调度软件时非常有用，可以确保在下一帧开始时不发生 USB 事务。
0 PIDERR	当 PID 检查字段故障时，将置位此位。

41.5.12 错误中断使能寄存器 (USBx_ERREN)

包含 USB 模块中每个错误中断源的使能位。置位这些位中的任意一位将使能 ERRSTAT 中的相应中断源。只要检测到错误情况，就会置位对应的位。因此，该中断通常不与处理的令牌末尾对应。复位之后，该寄存器的值为 0x00。

地址: 4007_2000h 基准 + 8Ch 偏移 = 4007_208Ch

位	7	6	5	4	3	2	1	0
读	BTSERREN	OWNERREN	DMAERREN	BTOERREN	DFN8EN	CRC16EN	CRC5EOFEN	PIDERREN
写	BTSERREN	OWNERREN	DMAERREN	BTOERREN	DFN8EN	CRC16EN	CRC5EOFEN	PIDERREN
复位	0	0	0	0	0	0	0	0

USBx_ERREN 字段描述

字段	描述
7 BTSERREN	BTSERR 中断使能 0 禁用 BTSERR 中断。 1 使能 BTSERR 中断。
6 OWNERREN	OWNERR 中断使能 当 USB 模块运行于外设模式 (CTL[HOSTMODEEN]=0) 时，该字段有效。 0 禁用 OWNERR 中断。 1 使能 OWNERR 中断。
5 DMAERREN	DMAERR 中断使能 0 禁用 DMAERR 中断。 1 使能 DMAERR 中断。
4 BTOERREN	BTOERR 中断使能 0 禁用 BTOERR 中断。 1 使能 BTOERR 中断。
3 DFN8EN	DFN8 中断使能 0 禁用 DFN8 中断。 1 使能 DFN8 中断。
2 CRC16EN	CRC16 中断使能 0 禁用 CRC16 中断。 1 使能 CRC16 中断。
1 CRC5EOFEN	CRC5/EOF 中断使能 0 禁用 CRC5/EOF 中断。 1 使能 CRC5/EOF 中断。
0 PIDERREN	PIDERR 中断使能 0 禁用 PIDERR 中断。 1 使能 PIDERR 中断。

41.5.13 状态寄存器 (USBx_STAT)

报告 USB 模块内的事务状态。当处理器的中断控制器接收到 TOKDNE 中断时，必须读取状态寄存器，以确定上一端点通信的状态。当 TOKDNE 中断有效时，状态寄存器中的数据有效。状态寄存器实际上是一个进入由 USB 模块维持的状态 FIFO 的读取窗口。当 USB 模块使用 BD 时，它将更新状态寄存器。如果在处理 TOKDNE 中断之前执行了其他 USB 事务，则 USB 模块会将下一事务的状态存储到 STAT FIFO 中。因此，STAT 实际上是一个四字节 FIFO，允许在处理器内核处理一个事务的同时，SIE 可以处理下一事务。清除 ISTAT 寄存器中的 TOKDNE 位将导致 SIE 用下一 STAT 值更新 STAT。如果 STAT 保持寄存器中的数据有效，则 SIE 将立即重新认定 TOKDNE 中断。

地址: 4007_2000h 基准 + 90h 偏移 = 4007_2090h

位	7	6	5	4	3	2	1	0
读	ENDP				TX	ODD	0	
写	ENDP				TX	ODD		
复位	0	0	0	0	0	0	0	0

USBx_STAT 字段描述

字段	描述
7-4 ENDP	此四位字段用于编码接收或发送上一令牌的端点地址。这允许处理器内核确定上一 USB 事务更新的 BDT 条目。
3 TX	发送指示器 0 最近事务为接收操作。 1 最近事务为发送操作。
2 ODD	如果更新的上一缓冲区描述符位于 BDT 的奇数库中，则置位此位。
保留	此只读字段为保留字段且值始终为 0。

41.5.14 控制寄存器 (USBx_CTL)

提供 USB 模块的各种控制和配置信息。

地址: 4007_2000h 基准 + 94h 偏移 = 4007_2094h

位	7	6	5	4
读	JSTATE	SE0	TXSUSPENDTOKENBUSY	RESET
写	JSTATE	SE0	TXSUSPENDTOKENBUSY	RESET
复位	0	0	0	0
位	3	2	1	0
读	HOSTMODEEN	RESUME	ODDRST	USBENSOFEN
写	HOSTMODEEN	RESUME	ODDRST	USBENSOFEN
复位	0	0	0	0

USBx_CTL 字段描述

字段	描述
7 JSTATE	实时 USB 差分接收器 JSTATE 信号 此信号的极性受 LSEN 当前状态影响。
6 SE0	实时 USB 单端零信号
5 TXSUSPENDTOKENBUSY	在主机模式下，当 USB 模块忙于执行 USB 令牌时，TOKEN_BUSY 置位。当 TOKEN_BUSY 置位时，软件不得向令牌寄存器写入其它令牌命令。将任何令牌写入令牌寄存器之前，软件应检查此字段，以确保令牌命令不会丢失。 在从设备模式下，如果 SIE 已禁用数据包发送和接收，则 TXD_SUSPEND 置位。清除此位将允许 SIE 继续处理令牌。接收到 SETUP 令牌时，此位将由 SIE 置位，以允许软件在恢复令牌处理之前清除 BDT 中挂起的数据包队列。
4 RESET	置位此位将使 USB 模块生成 USB 复位信号。这允许 USB 模块复位 USB 外设。该控制信号仅在主机模式(CTL[HOSTMODEEN]=1)下有效。软件必须将 RESET 置位为 1，以实现所需的时间量，然后再将其清零以结束复位信号。
3 HOSTMODEEN	置位为 1 时，此位将使能 USB 模块在主机模式下运行。在主机模式下，USB 模块在主机处理器的设定控制下执行 USB 事务。
2 RESUME	置位为 1 后，此位将使能 USB 模块执行恢复信号。这允许 USB 模块执行远程唤醒。软件必须将 RESUME 置位为 1，并保持要求的时间，然后将其清零。如果置位 HOSTMODEEN 位，则当 RESUME 位清零时，USB 模块会将低速数据包端附加到恢复信号。
1 ODDRST	将此位置位为 1 会将所有 BDT ODD 乒乓字段复位为 0，这随后会指定 EVEN BDT 库。
0 USBENSOFEN	USB 使能 置位此位将使能 USB-FS 工作；此为清零将禁用 USB-FS。置位此位将导致 SIE 将所有 ODD 位复位为 BDT。因此，置位此位将会复位 SIE 中的大部分逻辑。 使能主机模式之后，将此位清零将导致 SIE 停止发送 SOF 令牌。 0 禁用 USB 模块。 1 使能 USB 模块。

41.5.15 地址寄存器 (USBx_ADDR)

处于外设模式时 (CTL[HOSTMODEEN]=0)，地址寄存器用于保持 USB 模块解码的唯一 USB 地址。在主机模式 (CTL[HOSTMODEEN]=1) 下运行时，USB 模块通过 TOKEN 数据包发送此地址。这将使能 USB 模块以唯一地址寻址 USB 外设。任一模式下，CTL[USBENSOFEN]必须为 1。在复位输入或者是 USB 收到一个复位信号之后，地址寄存器将复位为 0x00。此操作将初始化地址寄存器，以按照 USB 规范解码地址 0x00。

地址: 4007_2000h 基准 + 98h 偏移 = 4007_2098h

位	7	6	5	4	3	2	1	0
读	LSEN		ADDR					
写	LSEN		ADDR					
复位	0	0	0	0	0	0	0	0

USBx_ADDR 字段描述

字段	描述
7 LSEN	低速使能位 通知 USB 模块必须以低速执行写入令牌寄存器的下一令牌命令。这将使能 USB 模块执行低速数据传输所需的前同步码。
ADDR	USB 地址 定义 USB 模块解码 (在外设模式下) 或发送 (在主机模式下) 的 USB 地址。

41.5.16 BDT 页寄存器 1 (USBx_BDTPAGE1)

提供基地址的地址位 15 至 9，其中，当前缓冲区描述符表 (BDT) 位于系统存储器中。请参见 [缓冲区描述符表](#)。32 位 BDT 基地址始终与 512 字节边界对齐，从而使基地址的位 8 至 0 始终为零。

地址: 4007_2000h 基准 + 9Ch 偏移 = 4007_209Ch

位	7	6	5	4	3	2	1	0
读	BDTBA							0
写	BDTBA							0
复位	0	0	0	0	0	0	0	0

USBx_BDTPAGE1 字段描述

字段	描述
7-1 BDTBA	提供 BDT 基地址的地址位 15 至 9。
0 保留	此只读字段为保留字段且值始终为 0。

41.5.17 低帧号寄存器 (USBx_FRMNUML)

帧号寄存器 (低和高) 包含 11 位帧号。只要收到 SOF TOKEN, 就会用当前帧号更新这些寄存器。

地址: 4007_2000h 基准 + A0h 偏移 = 4007_20A0h

位	7	6	5	4	3	2	1	0
读	FRM[7:0]							
写	FRM[7:0]							
复位	0	0	0	0	0	0	0	0

USBx_FRMNUML 字段描述

字段	描述
FRM[7:0]	这 8 位字段和高帧号寄存器的 3 位字段用于计算当前缓冲区描述符表 (BDT) 位于系统存储器时的地址。

41.5.18 高帧号寄存器 (USBx_FRMNUMH)

帧号寄存器 (低和高) 包含 11 位帧号。只要收到 SOF TOKEN, 就会用当前帧号更新这些寄存器。

地址: 4007_2000h 基准 + A4h 偏移 = 4007_20A4h

位	7	6	5	4	3	2	1	0
读	0				FRM[10:8]			
写	0				FRM[10:8]			
复位	0	0	0	0	0	0	0	0

USBx_FRMNUMH 字段描述

字段	描述
7-3 保留	此只读字段为保留字段且值始终为 0。
FRM[10:8]	这 3 位字段和低帧号寄存器的 8 位字段用于计算当前缓冲区描述符表 (BDT) 位于系统存储器时的地址。

41.5.19 令牌寄存器 (USBx_TOKEN)

在主机模式下 (CTL[HOSTMODEEN]=1)，令牌寄存器用于执行 USB 事务。当软件需要对外设执行 USB 事务时，它会将 TOKEN 类型和端点写入此寄存器。写入此寄存器之后，USB 模块开始对地址寄存器中包含的地址执行指定的 USB 事务。写入令牌寄存器之前，处理器内核务必确认控制寄存器中的 TOKEN_BUSY 位不是 1。这将确保令牌命令在被执行之前不会被覆盖。执行令牌命令时，还会使用地址寄存器和端点控制寄存器 0，因此，必须在令牌寄存器之前将其写入。地址寄存器用于选择令牌命令发送的 USB 外设地址。端点控制寄存器用于决定传输过程中使用的握手和重试策略。

地址: 4007_2000h 基准 + A8h 偏移 = 4007_20A8h

位	7	6	5	4	3	2	1	0
读	TOKENPID				TOKENENDPT			
写	TOKENPID				TOKENENDPT			
复位	0	0	0	0	0	0	0	0

USBx_TOKEN 字段描述

字段	描述
7-4 TOKENPID	包含 USB 模块执行的令牌类型。 0001 OUT 令牌。USB 模块执行 OUT (TX) 事务。 1001 IN 令牌。USB 模块执行 IN (RX) 事务。 1101 SETUP 令牌。USB 模块执行 SETUP (TX) 事务。
TOKENENDPT	保存令牌命令的端点地址。写入的 4 位值必须为有效端点。

41.5.20 SOF 阈值寄存器 (USBx_SOFTHLD)

SOF 阈值寄存器仅用于主机模式(CTL[HOSTMODEEN]=1)。处于主机模式时，14 位 SOF 计数器为 SOF 帧之间的间隔计数。必须每 1 ms 发送一次 SOF，以便将值 12000 载入 SOF 计数器。当 SOF 计数器达到 0 时，将会发送起始帧 (SOF) 令牌。

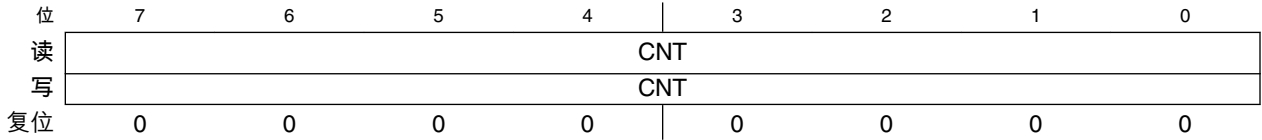
SOF 阈值寄存器用于在 SOF 停止发起令牌数据包事务之前，当 SOFDYNTHLD=0 时，设定 USB 字节时间量，或者当 SOFDYNTHLD=1 时，设定 8 个字节时间量。此寄存器必须置位为一个可确保在 SOF 时间计数为 0 时不会主动发送其他数据包的值。。当 SOF 计数器达到阈值时，在发送完 SOF 之前不会再发送令牌。

该阈值寄存器中设定的值必须保留充足的时间，以确保最差情况事务完成。一般来说，最差情况事务为 IN 令牌，其后是一个来自外设器件的数据包，其后是来自主机的响应。所需的实际时间为总线上最大数据包大小的函数。

SOF 阈值的典型值为：

- 64 字节数据包=74;
- 32 字节数据包=42;
- 16 字节数据包=26;
- 8 字节数据包=18。

地址: 4007_2000h 基准 + ACh 偏移 = 4007_20ACh



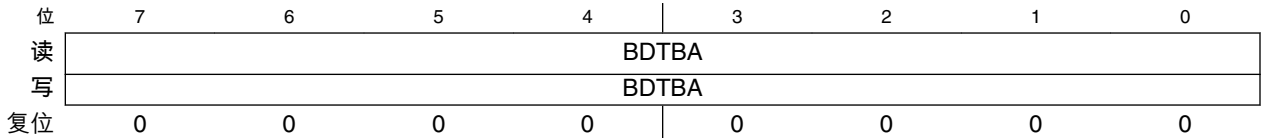
USBx_SOFTHLTD 字段描述

字段	描述
CNT	当 SOFDYNTHLD=0 时, 表示 SOF 计数阈值 (单位为字节时间量), 当 SOFDYNTHLD=1 时, 表示 8 字节时间量。

41.5.21 BDT 页寄存器 2 (USBx_BDTPAGE2)

包含一个 8 位值, 用于计算当前缓冲区描述符表 (BDT) 位于系统存储器中的地址。请参见[缓冲区描述符表](#)。

地址: 4007_2000h 基准 + B0h 偏移 = 4007_20B0h



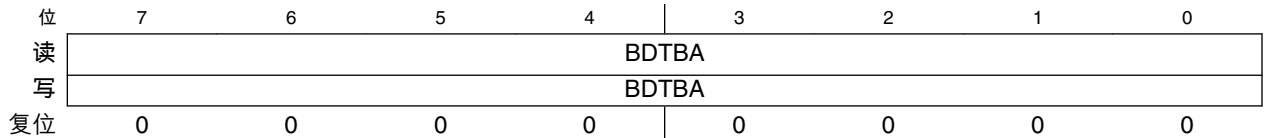
USBx_BDTPAGE2 字段描述

字段	描述
BDTBA	提供 BDT 基地址的地址位 23 至 16, 用于定义缓冲区描述符表在系统存储器中所处的位置。

41.5.22 BDT 页寄存器 3 (USBx_BDTPAGE3)

包含一个 8 位值，用于计算当前缓冲区描述符表 (BDT) 位于系统存储器中的地址。请参见[缓冲区描述符表](#)。

地址: 4007_2000h 基准 + B4h 偏移 = 4007_20B4h



USBx_BDTPAGE3 字段描述

字段	描述
BDTBA	提供 BDT 基地址的地址位 31 至 24，用于定义缓冲区描述符表在系统存储器中所处的位置。

41.5.23 端点控制寄存器 (USBx_ENDPTn)

对于一个解码地址，端点控制寄存器包含 USB 模块中提供的 16 个端点的控制位。下图显示的是这些寄存器的格式。端点 0 (ENDPT0) 与控制通道 0 关联，这是所有 USB 功能必须的。因此，在 USBRST 中断发生之后，处理器内核应设置 ENDPT0 以包含 0x0D。

在主机模式下，ENDPT0 用于确定主机传输时的握手、重试和低速特性。为了实现控制、批量和中断传输，EPHSK 位应为 1。为了实现同步传输，该位应为 0。主机模式下 ENDPT0 的常用值 0x4D 用于控制、批量和中断传输，而 0x4C 用于同步传输。

EPCTLDIS、EPRXEN 和 EPTXEN 这三位定义一个端点是否使能以及该端点的方向。端点使能/方向控制在下表中定义。

表 41-105. 端点使能和方向控制

EPCTLDIS	EPRXEN	EPTXEN	端点使能/方向控制 I
X	0	0	禁用端点
X	0	1	仅使能端点进行 Tx 传输
X	1	0	仅使能端点进行 Rx 传输
1	1	1	使能端点进行 Rx 和 Tx 传输
0	1	1	使能端点进行 RX、TX 和控制 (SETUP) 传输。

地址: 4007_2000h 基准 + C0h 偏移 + (4d × i), 其中 i=0d 到 15d

位	7	6	5	4	3	2	1	0
读	HOSTWOHUB	RETRYDIS	0	EPCTLDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK
写	HOSTWOHUB	RETRYDIS		EPCTLDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK
复位	0	0	0	0	0	0	0	0

USBx_ENDPTn 字段描述

字段	描述
7 HOSTWOHUB	无集线器的主机该字段仅用于主机模式，并且仅存在于端点 0 (ENDPT0) 的控制寄存器中。 0 低速器件通过集线器连接到主机。需要时会产生 PRE_PID。 1 低速器件直接连接。无集线器，或未连接低速器件。
6 RETRYDIS	该位仅用于主机模式，并且仅存在于端点 0 (ENDPT0) 的控制寄存器中。置位后，此位将导致主机不重试 NAK'ed (否定应答) 事务。否定应答事务之后，BDT PID 字段将更新为 NAK PID 并且会置位 TOKEN_DNE 中断。此位清零之后，硬件会重试无应答的事务。当主机尝试轮询中断端点时，必须置位此位。
5 保留	此只读字段为保留字段且值始终为 0。
4 EPCTLDIS	置位后，此位将禁用控制 (SETUP) 传输。清零后，将会使能控制传输。当并且仅当置位 EPRXEN 和 EPTXEN 位时才会应用。请参见 表 41-32
3 EPRXEN	置位后，此位将使能 RX 传输的端点。请参见 表 41-32
2 EPTXEN	置位后，此位将使能 TX 传输的端点。请参见 表 41-32
1 EPSTALL	置位后，此位表示端点已停止。此位优先于该寄存器中的所有其他控制位，但它仅在 EPTXEN=1 或 EPRXEN=1 时有效。至此端点的任何访问均会导致 USB 模块返回 STALL 握手。端点停止之后，需要主机控制器进行干预。
0 EPHSHK	置位后，此位将使能端点在事务处理过程中与此端点进行握手。此位一般为 1，除非端点为同步端点。

41.5.24 USB 控制寄存器 (USBx_USBCTRL)

地址: 4007_2000h 基准 + 100h 偏移 = 4007_2100h

位	7	6	5	4	3	2	1	0
读	SUSP	PDE	UARTCHLS	UARTSEL	0			
写	SUSP	PDE	UARTCHLS	UARTSEL				
复位	1	1	0	0	0	0	0	0

USBx_USBCTRL 字段描述

字段	描述
7 SUSP	将 USB 收发器置于挂起状态。 0 USB 收发器未处于挂起状态。 1 USB 收发器处于挂起状态。
6 PDE	使能 USB 收发器上的弱下拉。

下一页继续介绍此表...

USBx_USBCTRL 字段描述 (继续)

字段	描述
	0 D+和 D-上的弱下拉禁用。 1 D+和 D-上的弱下拉使能。
5 UARTCHLS	UART 信号通道选择 此字段仅在选择了 USB 信号以用作 UART 信号时有效。 0 USB DP/DM 信号用作 UART TX/RX。 1 USB DP/DM 信号用作 UART RX/TX。
4 UARTSEL	选择将用作 UART 信号的 USB 信号。 0 USB 信号未用作 UART 信号。 1 USB 信号已用作 UART 信号。
保留	此只读字段为保留字段且值始终为 0。

41.5.25 USB OTG 观察寄存器 (USBx_OBSERVE)

提供收发器上拉和下拉状态的可视性。当通过串行接口连接到外部 OTG 控制模块时非常有用。

地址: 4007_2000h 基准 + 104h 偏移 = 4007_2104h

位	7	6	5	4	3	2	1	0
读	DPPU	DPPD	0	DMPD	0			
写	DPPU	DPPD		DMPD				
复位	0	1	0	1	0	0	0	0

USBx_OBSERVE 字段描述

字段	描述
7 DPPU	提供 USB 收发器处使能信号输出的 D+上拉的可视性。 0 D+ 上拉禁用。 1 D+ 上拉使能。
6 DPPD	提供 USB 收发器处使能信号输出的 D+下拉的可视性。 0 D+ 下拉禁用。 1 D+ 下拉使能。
5 保留	此只读字段为保留字段且值始终为 0。
4 DMPD	提供 USB 收发器处使能信号输出的 D-下拉的可视性。 0 D- 下拉禁用。 1 D- 下拉使能。
保留	此只读字段为保留字段且值始终为 0。

41.5.26 USB OTG 控制寄存器 (USBx_CONTROL)

地址: 4007_2000h 基准 + 108h 偏移 = 4007_2108h

位	7	6	5	4
读	0			DPPULLUPNONOTG
写	0			DPPULLUPNONOTG
复位	0	0	0	0
位	3	2	1	0
读	0			
写	0			
复位	0	0	0	0

USBx_CONTROL 字段描述

字段	描述
7-5 保留	此只读字段为保留字段且值始终为 0。
4 DPPULLUPNONOTG	提供对 USBOTG 中的 DP 上拉的控制，如果 USB 配置为非 OTG 器件模式。 0 非 OTG 器件模式下 DP 上拉未使能。 1 非 OTG 器件模式下 DP 上拉已使能。
保留	此只读字段为保留字段且值始终为 0。

41.5.27 USB 收发器控制寄存器 0 (USBx_USBTRC0)

包括用于片上 USB 全速收发器基本操作的信号和 USB 数据连接的配置，这些信号和配置未包括在 USB 全速控制器寄存器中。在总线供电型器件应用中，当 USB 槽的 VBUS 引脚连接 VREGIN 时，可以将采用 VFEDG_DET 或 VREDG_DET 位域的 VREGIN 中断检测功能用于检测 VBUS。

地址: 4007_2000h 基准 + 10Ch 偏移 = 4007_210Ch

位	7	6	5	4
读	0		USBRESMEN	VFEDG_DET
写	USBRESET	0		VFEDG_DET
复位	0	0	0	0
位	3	2	1	0
读	VREDG_DET	USB_CLK_RECOVERY_INT	SYNC_DET	USB_RESUME_INT
写	VREDG_DET	USB_CLK_RECOVERY_INT	SYNC_DET	USB_RESUME_INT
复位	0	0	0	0

USBx_USBTRC0 字段描述

字段	描述
7 USBRESET	<p>USB 复位</p> <p>产生 USBOTG 的硬复位信号。置位此位且发生复位之后，此位将会自动清零。</p> <p>注：此位始终读为 0。在该位置位之后，访问其他 USB 寄存器位域之前，等待两个 USB 时钟周期。</p> <p>0 正常 USB 模块操作。 1 将 USB 模块返回至其复位状态。</p>
6 保留	此只读字段为保留字段且值始终为 0。
5 USBRESMEN	<p>异步恢复中断使能</p> <p>置位后，此位允许 USB 模块将在检测到 USB 总线上的恢复信号时向 MCU 发送异步唤醒事件。MCU 随后重新使能 USB 模块的时钟。当 USB 模块时钟停止或 USB 收发器处于挂起模式时，它用于低功耗挂起模式。异步唤醒仅在从设备模式下工作。</p> <p>0 USB 从挂起模式异步唤醒禁用。 1 USB 从挂起模式异步唤醒使能。异步恢复中断与同步恢复中断的不同之处在于，前者使用 D+ 和 D- 引脚的未滤波状态异步检测 K 状态。仅当收发器挂起时才使能此中断。</p>
4 VFEDG_DET	<p>VREGIN 下降沿中断检测</p> <p>用 USBx_MISCCTRL[VFEDG_EN]使能此位域。</p> <p>0 未检测到 VREGIN 下降沿中断。 1 已检测到 VREGIN 下降沿中断。</p>
3 VREDG_DET	<p>VREGIN 上升沿中断检测</p> <p>用 USBx_MISCCTRL[VREDG_EN]使能此位域。</p> <p>0 未检测到 VREGIN 上升沿中断。 1 已检测到 VREGIN 上升沿中断。</p>
2 USB_CLK_RECOVERY_INT	<p>组合 USB 时钟恢复中断状态</p> <p>当检测到任意 USB 时钟恢复中断条件且这些中断未被屏蔽时，此字段将被设置为高值 1'b1。</p> <p>为方便用户使用，唯一未屏蔽的 USB 时钟恢复中断条件产生自频率调整设置值的溢出，表示计算的频率调整超出 IRC48M 输出时钟的调整范围。</p> <p>此位置位后，要清除此位，应将 0xFF 写入寄存器 USB_CLK_RECOVER_INT_STATUS。</p>
1 SYNC_DET	<p>同步 USB 中断检测</p> <p>0 未检测到同步中断。 1 已检测到同步中断。</p>
0 USB_RESUME_INT	<p>USB 异步中断</p> <p>0 未生成任何中断。 1 因 USB 异步中断生成中断。</p>

41.5.28 帧调整寄存器 (USBx_USBFRMADJUST)

地址: 4007_2000h 基准 + 114h 偏移 = 4007_2114h

位	7	6	5	4	3	2	1	0
读	ADJ							
写	ADJ							
复位	0	0	0	0	0	0	0	0

USBx_USBFRMADJUST 字段描述

字段	描述
ADJ	<p>帧调整</p> <p>在主机模式下, 帧调整为 2 的补码, 可在 12-MHz 时钟周期中调整每个 USB 帧的周期。SOF 通常产生 12,000 个 12-MHz 时钟周期。帧调整寄存器可在 -128 至 +127 范围内进行调节, 以补偿 USB 48 MHz 时钟的不精确性。对 ADJ 位的更改将在下一帧的起点开始生效。</p>

41.5.29 其他控制寄存器 (USBx_MISCCTRL)

地址: 4007_2000h 基准 + 12Ch 偏移 = 4007_212Ch

位	7	6	5	4
读	STL_ADJ_EN	0		VFEDG_EN
写	STL_ADJ_EN			VFEDG_EN
复位	0	0	0	0
位	3	2	1	0
读	VREDG_EN	OWNERRISODIS	SOFBUSSET	SOFDYNTHLD
写	VREDG_EN	OWNERRISODIS	SOFBUSSET	SOFDYNTHLD
复位	0	0	0	0

USBx_MISCCTRL 字段描述

字段	描述
7 STL_ADJ_EN	<p>USB 外设模式停止调整使能</p> <p>此字段仅在外设模式下(CTL[HOSTMODEEN]=0)有效。默认情况下(STL_ADJ_EN = 0), 当一个端点停止(ENDPn[END_STALL]=1)时, 该端点的 IN 和 OUT 两个方向将同时停止。如果 STL_ADJ_EN = 1, 则当一个端点停止(ENDPn[END_STALL]=1)时, 可以用 USBx_STALL_xx_DIS 寄存器来控制受影响的端点方向。</p> <p>0 如果 USB_ENDPn[END_STALL] = 1, 关联端点的 IN 和 OUT 两个方向将同时停止</p> <p>1 如果 USB_ENDPn[END_STALL] = 1, 则 USB_STALL_xx_DIS 寄存器控制将停止关联端点的哪个方向。</p>
6-5 保留	此只读字段为保留字段且值始终为 0。
4 VFEDG_EN	VREGIN 下降沿中断使能

下一页继续介绍此表...

USBx_MISCCTRL 字段描述 (继续)

字段	描述
	0 VREGIN 下降沿中断被禁用。 1 VREGIN 下降沿中断被使能。
3 VREDG_EN	VREGIN 上升沿中断使能 0 VREGIN 上升沿中断被禁用。 1 VREGIN 上升沿中断被使能。
2 OWNERRISODIS	用于 ISO IN / ISO OUT 的 OWN 错误检测禁用 此字段仅对外设模式 (即 CTL[HOSTMODEEN]=0) 有效。 0 用于 ISO IN / ISO OUT 的 OWN 错误检测未被禁用。 1 用于 ISO IN / ISO OUT 的 OWN 错误检测被禁用。
1 SOFBUSSET	SOF_TOK 中断生成模式选择 此字段仅对主机模式 (即 CTL[HOSTMODEEN]=1) 有效。 0 根据 SOF 阈值置位 SOF_TOK 中断。 1 当 SOF 计数器达到 0 时, SOF_TOK 中断置位。
0 SOFDYNTHLD	动态 SOF 阈值比较模式 此字段仅对主机模式 (即 CTL[HOSTMODEEN]=1) 有效。 0 当达到字节与 SOF 阈值之积时, SOF_TOK 中断置位。 1 当达到或超过 8 字节与 SOF 阈值之积时, SOF_TOK 中断置位。

41.5.30 用于 IN 方向 7-0 端点的外设模式停止禁用 (USBx_STALL_IL_DIS)

此寄存器仅在外设模式(CTL[HOSTMODEEN]=0)下以及当停止调整使能位置位(MISCCTRL[STL_ADJ_EN]=1)时有效。

当一个端点停止(ENDPTn[END_STALL]=1)时, 此寄存器中的字段将使能或禁用 7-0 端点 IN 方向的停止功能。

地址: 4007_2000h 基准 + 130h 偏移 = 4007_2130h

位	7	6	5	4
读	STALL_I_DIS7	STALL_I_DIS6	STALL_I_DIS5	STALL_I_DIS4
写	STALL_I_DIS7	STALL_I_DIS6	STALL_I_DIS5	STALL_I_DIS4
复位	0	0	0	0
位	3	2	1	0
读	STALL_I_DIS3	STALL_I_DIS2	STALL_I_DIS1	STALL_I_DIS0
写	STALL_I_DIS3	STALL_I_DIS2	STALL_I_DIS1	STALL_I_DIS0
复位	0	0	0	0

USBx_STALL_IL_DIS 字段描述

字段	描述
7 STALL_IL_DIS7	禁用端点 7 IN 方向。 0 端点 7 IN 方向停止被使能。 1 端点 7 IN 方向停止被禁用。
6 STALL_IL_DIS6	禁用端点 6 IN 方向。 0 端点 6 IN 方向停止被使能。 1 端点 6 IN 方向停止被禁用。
5 STALL_IL_DIS5	禁用端点 5 IN 方向。 0 端点 5 IN 方向停止被使能。 1 端点 5 IN 方向停止被禁用。
4 STALL_IL_DIS4	禁用端点 4 IN 方向。 0 端点 4 IN 方向停止被使能。 1 端点 4 IN 方向停止被禁用。
3 STALL_IL_DIS3	禁用端点 3 IN 方向。 0 端点 3 IN 方向停止被使能。 1 端点 3 IN 方向停止被禁用。
2 STALL_IL_DIS2	禁用端点 2 IN 方向。 0 端点 2 IN 方向停止被使能。 1 端点 2 IN 方向停止被禁用。
1 STALL_IL_DIS1	禁用端点 1 IN 方向。 0 端点 1 IN 方向停止被使能。 1 端点 1 IN 方向停止被禁用。
0 STALL_IL_DIS0	禁用端点 0 IN 方向。 0 端点 0 IN 方向停止被使能。 1 端点 0 IN 方向停止被禁用。

41.5.31 用于 IN 方向 15-8 端点的外设模式停止禁用 (USBx_STALL_IH_DIS)

此寄存器仅在外设模式(CTL[HOSTMODEEN]=0)下以及当停止调整使能位置位(MISCCTRL[STL_ADJ_EN]=1)时有效。

当一个端点停止(ENDPTn[END_STALL]=1)时，此寄存器中的字段将使能或禁用 15-8 端点 IN 方向的停止功能。

地址: 4007_2000h 基准 + 134h 偏移 = 4007_2134h

位	7	6	5	4
读	STALL_I_DIS15	STALL_I_DIS14	STALL_I_DIS13	STALL_I_DIS12
写	STALL_I_DIS15	STALL_I_DIS14	STALL_I_DIS13	STALL_I_DIS12
复位	0	0	0	0
位	3	2	1	0
读	STALL_I_DIS11	STALL_I_DIS10	STALL_I_DIS9	STALL_I_DIS8
写	STALL_I_DIS11	STALL_I_DIS10	STALL_I_DIS9	STALL_I_DIS8
复位	0	0	0	0

USBx_STALL_IH_DIS 字段描述

字段	描述
7 STALL_I_DIS15	禁用端点 15 IN 方向。 0 端点 15 IN 方向停止被使能。 1 端点 15 IN 方向停止被禁用。
6 STALL_I_DIS14	禁用端点 14 IN 方向。 0 端点 14 IN 方向停止被使能。 1 端点 14 IN 方向停止被禁用。
5 STALL_I_DIS13	禁用端点 13 IN 方向。 0 端点 13 IN 方向停止被使能。 1 端点 13 IN 方向停止被禁用。
4 STALL_I_DIS12	禁用端点 12 IN 方向。 0 端点 12 IN 方向停止被使能。 1 端点 12 IN 方向停止被禁用。
3 STALL_I_DIS11	禁用端点 11 IN 方向。 0 端点 11 IN 方向停止被使能。 1 端点 11 IN 方向停止被禁用。
2 STALL_I_DIS10	禁用端点 10 IN 方向。 0 端点 10 IN 方向停止被使能。 1 端点 10 IN 方向停止被禁用。
1 STALL_I_DIS9	禁用端点 9 IN 方向。 0 端点 9 IN 方向停止被使能。 1 端点 9 IN 方向停止被禁用。
0 STALL_I_DIS8	禁用端点 8 IN 方向。 0 端点 8 IN 方向停止被使能。 1 端点 8 IN 方向停止被禁用。

41.5.32 用于 OUT 方向 7-0 端点的外设模式停止禁用 (USBx_STALL_OL_DIS)

此寄存器仅在外设模式(CTL[HOSTMODEEN]=0)下以及当停止调整使能位置位(MISCCTRL[STL_ADJ_EN]=1)时有效。

当一个端点停止(ENDPTn[END_STALL]=1)时, 此寄存器中的字段将使能或禁用 7-0 端点 OUT 方向的停止功能。

地址: 4007_2000h 基准 + 138h 偏移 = 4007_2138h

位	7	6	5	4
读	STALL_O_DIS7	STALL_O_DIS6	STALL_O_DIS5	STALL_O_DIS4
写	STALL_O_DIS7	STALL_O_DIS6	STALL_O_DIS5	STALL_O_DIS4
复位	0	0	0	0
位	3	2	1	0
读	STALL_O_DIS3	STALL_O_DIS2	STALL_O_DIS1	STALL_O_DIS0
写	STALL_O_DIS3	STALL_O_DIS2	STALL_O_DIS1	STALL_O_DIS0
复位	0	0	0	0

USBx_STALL_OL_DIS 字段描述

字段	描述
7 STALL_O_DIS7	禁用端点 7 OUT 方向。 0 端点 7 OUT 方向停止被使能。 1 端点 7 OUT 方向停止被禁用。
6 STALL_O_DIS6	禁用端点 6 OUT 方向。 0 端点 6 OUT 方向停止被使能。 1 端点 6 OUT 方向停止被禁用。
5 STALL_O_DIS5	禁用端点 5 OUT 方向。 0 端点 5 OUT 方向停止被使能。 1 端点 5 OUT 方向停止被禁用。
4 STALL_O_DIS4	禁用端点 4 OUT 方向。 0 端点 4 OUT 方向停止被使能。 1 端点 4 OUT 方向停止被禁用。
3 STALL_O_DIS3	禁用端点 3 OUT 方向。 0 端点 3 OUT 方向停止被使能。 1 端点 3 OUT 方向停止被禁用。
2 STALL_O_DIS2	禁用端点 2 OUT 方向。 0 端点 2 OUT 方向停止被使能。 1 端点 2 OUT 方向停止被禁用。

下一页继续介绍此表...

USBx_STALL_OL_DIS 字段描述 (继续)

字段	描述
1 STALL_O_DIS1	禁用端点 1 OUT 方向。 0 端点 1 OUT 方向停止被使能。 1 端点 1 OUT 方向停止被禁用。
0 STALL_O_DIS0	禁用端点 0 OUT 方向。 0 端点 0 OUT 方向停止被使能。 1 端点 0 OUT 方向停止被禁用。

41.5.33 用于 OUT 方向 15-8 端点的外设模式停止禁用 (USBx_STALL_OH_DIS)

此寄存器仅在外设模式(CTL[HOSTMODEEN]=0)下以及当停止调整使能位置位(MISCCTRL[STL_ADJ_EN]=1)时有效。

当一个端点停止(ENDPTn[END_STALL]=1)时，此寄存器中的字段将使能或禁用 15-8 端点 OUT 方向的停止功能。

地址: 4007_2000h 基准 + 13Ch 偏移 = 4007_213Ch

位	7	6	5	4
读	STALL_O_DIS15	STALL_O_DIS14	STALL_O_DIS13	STALL_O_DIS12
写	STALL_O_DIS15	STALL_O_DIS14	STALL_O_DIS13	STALL_O_DIS12
复位	0	0	0	0
位	3	2	1	0
读	STALL_O_DIS11	STALL_O_DIS10	STALL_O_DIS9	STALL_O_DIS8
写	STALL_O_DIS11	STALL_O_DIS10	STALL_O_DIS9	STALL_O_DIS8
复位	0	0	0	0

USBx_STALL_OH_DIS 字段描述

字段	描述
7 STALL_O_DIS15	禁用端点 15 OUT 方向。 0 端点 15 OUT 方向停止被使能。 1 端点 15 OUT 方向停止被禁用。
6 STALL_O_DIS14	禁用端点 14 OUT 方向。 0 端点 14 OUT 方向停止被使能。 1 端点 14 OUT 方向停止被禁用。
5 STALL_O_DIS13	禁用端点 13 OUT 方向。 0 端点 13 OUT 方向停止被使能。 1 端点 13 OUT 方向停止被禁用。

下一页继续介绍此表...

USBx_STALL_OH_DIS 字段描述 (继续)

字段	描述
4 STALL_O_DIS12	禁用端点 12 OUT 方向。 0 端点 12 OUT 方向停止被使能。 1 端点 12 OUT 方向停止被禁用。
3 STALL_O_DIS11	禁用端点 11 OUT 方向。 0 端点 11 OUT 方向停止被使能。 1 端点 11 OUT 方向停止被禁用。
2 STALL_O_DIS10	禁用端点 10 OUT 方向。 0 端点 10 OUT 方向停止被使能。 1 端点 10 OUT 方向停止被禁用。
1 STALL_O_DIS9	禁用端点 9 OUT 方向。 0 端点 9 OUT 方向停止被使能。 1 端点 9 OUT 方向停止被禁用。
0 STALL_O_DIS8	禁用端点 8 OUT 方向。 0 端点 8 OUT 方向停止被使能。 1 端点 8 OUT 方向停止被禁用。

41.5.34 USB 时钟恢复控制 (USBx_CLK_RECOVER_CTRL)

此寄存器中的信号控制无晶振 USB 时钟模式，在这种模式下，调谐内部 IRC48M 振荡器来和从 USB 输入数据流提取的时钟匹配。

对于此模式，必须在寄存器 USB_CLK_RECOVER_IRC_EN 中使能 IRC48M 内部振荡器模块。

地址: 4007_2000h 基准 + 140h 偏移 = 4007_2140h

位	7	6	5	4	3	2	1	0
读	CLOCK_RECOVER_EN	RESET_RESUME_ROUGH_EN	RESTART_IFRTRIM_EN	保留	保留	保留	保留	保留
写	CLOCK_RECOVER_EN	RESET_RESUME_ROUGH_EN	RESTART_IFRTRIM_EN	保留	保留	保留	保留	保留
复位	0	0	0	0	0	0	0	0

USBx_CLK_RECOVER_CTRL 字段描述

字段	描述
7 CLOCK_RECOVER_EN	无晶振 USB 使能 对于全速 USB 控制器和收发器，若用户希望使用无晶振 USB 模式，必须使能此位。 注：对于 USB 主机模式或 OTG，此位不应置位。

下一页继续介绍此表...

USBx_CLK_RECOVER_CTRL 字段描述 (继续)

字段	描述
	0 禁用时钟恢复模块 (默认) 1 使能时钟恢复模块
6 RESET_ RESUME_ ROUGH_EN	复位/恢复到粗调阶段使能 时钟恢复模块跟踪 IRC48M 以获得精确的 48MHz 时钟。用户使能 clock_recover_en 位之后，有两个阶段：粗调阶段和跟踪阶段。在这两个阶段，通过调整精调值来精调 IRC48M 的步进是不同的。粗调阶段的步进大于跟踪阶段的步进。只要 USB 总线复位或发生总线恢复事件，就会切换回粗调阶段。 0 完成第一次粗调到跟踪转换之后，始终工作在跟踪阶段 (默认)。 1 只要总线复位或发生总线恢复事件，就会回到粗调阶段。
5 RESTART_ IFRTRIM_EN	从 IFR 调整值重启 IRC48M 有一个默认精调值，其默认值已在工厂进行调整 (IFR 调整值)。时钟恢复模块跟踪 48MHz 时钟的精度，并相应地更新精调值 0 精密调整始终基于先前更新的精调值工作 (默认)。 1 只要检测到 bus_reset/bus_resume 或模块使能解除认定，精调就会从 IFR 调整值重新开始。
4-3 保留	
2 保留	此位仅供恩智浦使用。客户不应改变此位的默认状态。
1 保留	此位仅供恩智浦使用。客户不应改变此位的默认状态。
0 保留	不应更改默认值

41.5.35 IRC48M 振荡器使能寄存器 (USBx_CLK_RECOVER_IRC_EN)

控制片上 IRC48M 模块的基本操作；对于 USB 无晶振操作和其它功能，该模块用于产生标称 48MHz 时钟。

有关 IRC48M 操作的更多信息，请参见“时钟分布”一章。

地址: 4007_2000h 基准 + 144h 偏移 = 4007_2144h

位	7	6	5	4	3	2	1	0
读	保留						IRC_EN	REG_EN
写	保留						IRC_EN	REG_EN
复位	0	0	0	0	0	0	0	1

USBx_CLK_RECOVER_IRC_EN 字段描述

字段	描述
7-2 保留	
1 IRC_EN	IRC48M 使能

下一页继续介绍此表...

USBx_CLK_RECOVER_IRC_EN 字段描述 (继续)

字段	描述
	此位用于使能片上 IRC48M 模块以产生无晶振 USB 所用的时钟。它可以用于 FS USB 器件工作模式。使用无晶振 USB 时钟配置之前，此位必须置位。 0 禁用 IRC48M 模块 (默认) 1 使能 IRC48M 模块
0 REG_EN	IRC48M 稳压器使能 此位用于使能 IRC48M 模块的本地模拟稳压器。若用户希望使用无晶振 USB 时钟配置，此位必须置位。 0 禁用 IRC48M 本地稳压器 1 使能 IRC48M 本地稳压器 (默认)

41.5.36 时钟恢复组合中断使能 (USBx_CLK_RECOVER_INT_EN)

使能或屏蔽个别中断标志；如果 USB 时钟恢复算法操作检测到指示的条件，这些标志通过逻辑 OR 运算在 USB_USBTRC0 寄存器的 USB_CLK_RECOVERY_INT 位产生组合中断指示。

地址: 4007_2000h 基准 + 154h 偏移 = 4007_2154h

位	7	6	5	4
读		保留		OVF_ERROR_EN
写		保留		OVF_ERROR_EN
复位	0	0	0	1
位	3	2	1	0
读			保留	
写			保留	
复位	0	0	0	0

USBx_CLK_RECOVER_INT_EN 字段描述

字段	描述
7-5 保留	务必写入 0。
4 OVF_ERROR_EN	决定 OVF_ERROR 条件信号是否用于 USB_CLK_RECOVERY_INT 的产生。 0 屏蔽该中断。 1 使能该中断 (默认)。
保留	务必写入 0。

41.5.37 时钟恢复分离中断状态 (USBx_CLK_RECOVER_INT_STATUS)

对于各位的任意组合，写 1 操作将清除这些位。

地址: 4007_2000h 基准 + 15Ch 偏移 = 4007_215Ch

位	7	6	5	4	3	2	1	0
读	保留			OVF_ERROR	保留			
写	w1c			w1c	w1c			
复位	0	0	0	0	0	0	0	0

USBx_CLK_RECOVER_INT_STATUS 字段描述

字段	描述
7-5 保留	务必写入 0。
4 OVF_ERROR	表示 USB 时钟恢复算法已检测到 IRC48M 输出时钟所需的频率调整超出了 IRC48M 模块的可用 TRIM_FINE 调整范围。 0 不报告中断。 1 已产生未屏蔽的中断。
保留	务必写入 0。

41.6 OTG 和主机模式操作

借助主机模式逻辑，可采用此 SOC 的便携式、移动和其他器件用来充当 USB 嵌入式主机 (EH) 控制器。OTG 逻辑添加了一个接口，用于允许在软件中执行 OTG 主机交换和会话请求协议 (HNP 和 SRP)。

主机模式专用于手持式便携设备，允许轻松连接至简单的 HID 类设备，如打印机和键盘。它不用于执行 PC 主板上的完整 OHCI 或 UHCI 主机控制器功能。主机模式允许批量、同步、中断和控制传输。将以几乎完整的 USB 接口带宽执行批量数据传输。提供 ISO 传输支持，但实际支持的 ISO 流数受 SIE 中断过程中执行令牌的处理器的中断延迟影响。必须写入自定义驱动程序，以支持主机模式操作。

置位 CTL 寄存器中的 HOST_MODE_EN 位将使能主机模式。USB-FS 内核仅可用作外设器件或用于主机模式。它无法在两种模式下同时运行。当使能 HOST_MODE 时，仅使用端点 0。所有其他端点应由软件禁用。

41.7 主机模式操作示例

以下小节介绍了使用 USB-FS 内核执行 USB 主机功能所需的步骤。有关这些程序的更多信息，请参见《通用串行总线规范，第 2.0 修订版》第 9 章“USB 器件框架”。

要启用主机模式并发现已连接器件：

1. 启用主机模式 (CTL[HOST_MODE_EN]=1)。下拉电阻启用且上拉电阻禁用。开始生成起始帧 (SOF)。SOF 计数器加载 12,000。通过向 USB 启用位写入 0 (CTL[USB_EN]=0) 禁用 SOF 数据包生成，以消除 USB 上的噪音。
2. 启用 ATTACH 中断 (INTEN[ATTACHEN]=1)。
3. 等待 ATTACH 中断 (ISTAT[ATTACH])。由 USB 外设上拉电阻通知，将 DPLUS 或 DMINUS 的状态从 0 改为 1 (SE0 至 J 或 K 状态)。
4. 检查控制寄存器中 JSTATE 和 SE0 位的状态。如果连接器件为低速 (JSTATE 位为 0)，则置位地址寄存器中的低速位 (ADDR[LS_EN]=1) 和主机，无需置位端点 0 寄存器控制中的集线器位 (ENDPT0[HOSTWOHUB]=1)。
5. 启用 RESET (CTL[RESET]=1) 10 ms。
6. 启用 SOF 数据包，以防止已连接器件挂起 (CTL[USB_EN]=1)。
7. 通过向连接器件的默认控制通道发送适当的命令来枚举连接的器件。

要完成至已连接器件的控制交易：

1. 请完成所有步骤来发现已连接器件。
2. 设置端点控制寄存器，以实现双向控制传输 $ENDPT0[4:0] = 0x0d$ 。
3. 将器件框架建立命令置于存储器缓冲区中。
4. 初始化当前 (偶数或奇数) TX EP0 BDT，以传输器件框架命令的 8 字节命令数据 (例如，GET DEVICE DESCRIPTOR)。
 - 将 BDT 命令字设置为 0x00080080 - 字节计数设置为 8，OWN 位置 1。
 - 将 BDT 缓冲区地址字段设置为 8 字节命令缓冲区的起始地址。
5. 设置地址寄存器 (ADDR[6:0]) 中的 USB 外设地址。USB 总线复位之后，器件 USB 地址为 0。它通过置位地址器件框架命令设为其他值 (通常为 1)。

6. 将外设默认控制通道端点 0 的 SETUP 令牌写入 TOKEN 寄存器 (TOKEN=0xD0)。这将初始化数据包后总线上的设置令牌。数据包完成之后, 将在 BDT PID 字段中返回器件握手。写入 BDT 之后, 将置位令牌完成的 (ISTAT[TOKDNE]) 中断。这将完成设置事务的设置阶段。
7. 要初始化设置事务的数据阶段 (也就是, 获取 GET DEVICE DESCRIPTOR 命令的数据), 请设置存储器中的缓冲区, 以便传输数据。
8. 初始化当前 (偶数或奇数) TX EP0 BDT 以传输数据。
 - 将 BDT 命令字设置为 0x004000C0 -BC 设置为 64 (在此示例中数据缓冲区的字节计数), OWN 位置 1, 数据切换设置为 Data1。
 - 将 BDT 缓冲区地址字段设置为数据缓冲区的起始地址。
9. 将外设默认控制通道端点 0 的 IN 或 OUT 令牌写入令牌寄存器, IN 令牌用于 GET DEVICE DESCRIPTOR 命令 (TOKEN=0x90)。这将在数据包从器件传输到主机之后初始化总线上的 IN 令牌。数据包完成之后, 将写入 BDT 并置位令牌完成的 (ISTAT[DNE]) 中断。为了控制单个数据包数据阶段的传输, 这将完成设置事务的数据阶段。
10. 要初始化设置事务的状态阶段, 请设置存储器中的缓冲区, 以接收或发送零长度状态阶段数据数据包。
11. 初始化当前 (偶数或奇数) TX EP0 BDT 以传输状态数据。
 - 将 BDT 命令字设置为 0x00000080 -BC 设置为 0 (在此示例中数据缓冲区的字节计数), OWN 位置 1, 数据切换设置为 Data1。
 - 将 BDT 缓冲区地址字段设置为数据缓冲区的起始地址。
12. 将外设默认控制通道端点 0 的 IN 或 OUT 令牌写入令牌寄存器, OUT 令牌用于 GET DEVICE DESCRIPTOR 命令 (TOKEN=0x10)。这将在零长度数据包从主机传输到器件之后初始化总线上的 OUT 令牌。数据包完成之后, 将通过器件握手写入 BDT 并置位令牌完成的 (ISTAT[TOKDNE]) 中断。这将完成设置事务的数据阶段。

要向外设发送全速批量数据传输:

1. 请完成所有步骤来发现已连接器件并配置已连接器件。将外设的地址写入 ADDR 寄存器。通常情况下, 在主机模式下, USB 总线上仅有一个其他器件, 因此, 预期地址为 0x01 并且应保持恒定。
2. 向 0x1D 寄存器写入 ENDPT0, 以在握手使能的情况下使能发送和接收传输。
3. 设置偶数 TX EP0 BDT 以传输最多 64 字节。

4. 设置地址寄存器 (ADDR[6:0]) 中的 USB 外设地址。
5. 将所需端点的 OUT 令牌写入 TOKEN 寄存器。写入此寄存器将触发 USB-FS 发送状态机开始发送 TOKEN 和数据。
6. 将奇数 TX EP0 BDT 设置为传输不超过 64 字节。
7. 如步骤 4 所述将 OUT 令牌写入 TOKEN 寄存器。两个令牌可以按时间排序, 从而允许双重缓冲数据包, 以实现最大吞吐量。
8. 等待 TOKDNE 中断。这表示其中一个 BDT 已被释放回处理器且传输已完成。如果外设置位 NAK, USB-FS 不受处理器干预, 继续无限期重试传输, 除非 ENDPT0[RETRYDIS] 为 1。如果已设置重试禁止字段, 则握手 (ACK、NAK、STALL 或 ERROR (0xf)) 会返回 BDT PID 字段。如果发生停止中断, 则挂起的数据包必须出列并且应清零外设中的错误情况。如果发生复位中断 (SEO 超过 2.5 us), 则外设将断开。
9. 发生 TOK_DNE 中断之后, 可以检查 BDT 并通过返回步骤 2 排列下一数据包。

41.8 On-The-Go 操作

USB-OTG 内核提供用于使能 On-The-Go (OTG) 操作的传感器和控件。OTG 驱动器软件使用这些传感器来实施主机交换协议 (HNP) 和会话请求协议 (SRP), 并访问 OTG 协议控制信号。以下状态机显示了与来自 USB 电缆任意一端的 HNP 和 SRP 协议相关的 OTG 操作。

41.8.1 OTG 双重角色 A 器件操作

A 器件因其连接的电缆类型而被视为 A 器件。如果将 USB 类标准 A 或 Micro-A 插头插入器件, 则该器件被认为是 A 器件。

双重角色 A 器件按照以下流程图和状态描述表所示运行。

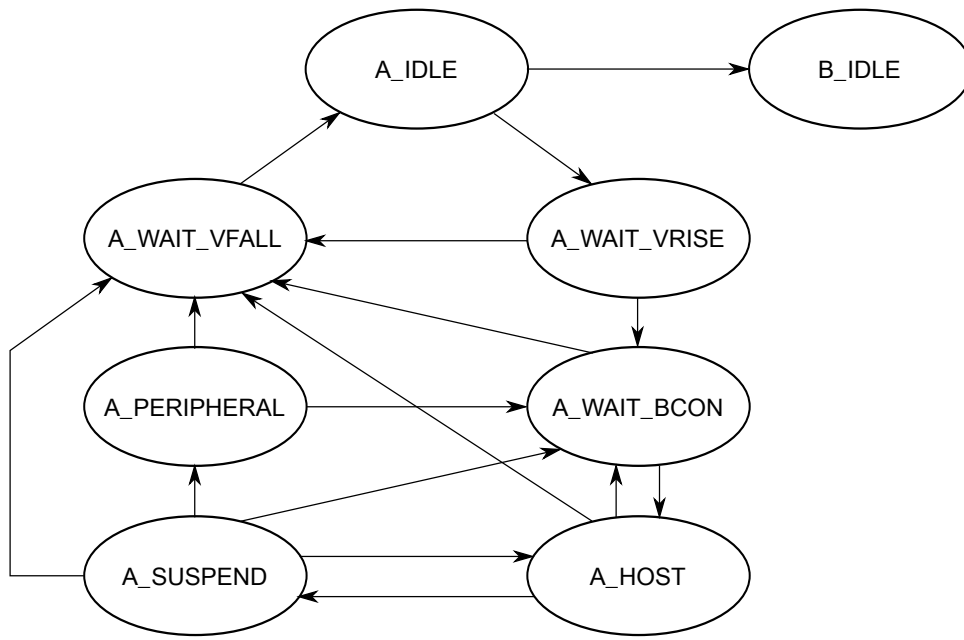


图 41-119. 双重角色 A 器件流程图

表 41-150. 双重角色 A 器件的流程状态描述

状态	操作	响应
A_IDLE	如果 ID 中断,	转至 B_IDLE。
	如果已拔下电缆或已连接 B 型电缆, 器件现在用作 B 型器件, 如果 A 应用想要使用总线或者如果 B 器件正在按照 5-10 毫秒脉冲内的 A_SESS_VLD 中断或连接或端口状态更改中断检查数据线路执行 SRP,	转至 A_WAIT_VRISE。 打开 DRV_VBUS。
A_WAIT_VRISE	如果 ID 中断或如果 A_VBUS_VLD 在 100 毫秒后故障,	转至 A_WAIT_VFALL。
	如果电缆已更换或 A 器件无法支持来自 B 器件的当前请求, 如果 A_VBUS_VLD 中断,	关闭 DRV_VBUS。 转至 A_WAIT_BCON。
A_WAIT_BCON	如果在 200 ms 之后 (无连接或 ID 中断) (需要时, 这将持续等待。),	转至 A_WAIT_FALL。 关闭 DRV_VBUS。
	如果 A_VBUS_VLD 中断和 B 器件连接,	转至 A_HOST。 打开主机模式。
A_HOST	枚举器件确定 OTG 支持。	
	如果 A_VBUS_VLD/中断或 A 器件已完成并且认为它不想在短时间内执行某一任务或 B 器件断开,	转至 A_WAIT_VFALL。 关闭主机模式。 关闭 DRV_VBUS。
	如果 A 器件已完成会话或者如果 A 器件想要允许 B 器件占用总线,	转至 A_SUSPEND。
	如果 ID 中断或 B 器件断开,	转至 A_WAIT_BCON。
A_SUSPEND	如果 ID 中断, 或者如果 B 器件发生 150 ms 断开超时 (此超时值可能更长) 或如果 A_VBUS_VLD\中断,	转至 A_WAIT_VFALL。 关闭 DRV_VBUS。
	如果 HNP 已使能, 且 B 器件在 150 ms 内断开, 则 B 器件将变成主机,	转至 A_PERIPHERAL。 关闭主机模式。

下一页继续介绍此表...

表 41-150. 双重角色 A 器件的流程状态描述 (继续)

状态	操作	响应
	如果 A 器件想要开始另一会话,	转至 A_HOST。
A_PERIPHERAL	如果 ID 中断或如果 A_VBUS_VLD 中断,	转至 A_WAIT_VFALL。 关闭 DRV_VBUS。
	如果总线闲置 3–200 ms,	转至 A_WAIT_BCON。 打开主机模式。
A_WAIT_VFALL	如果 ID 中断或 (A_SESS_VLD/ & b_conn/),	转至 A_IDLE。

41.8.2 OTG 双重角色 B 器件操作

如果一个器件以 USB Standard-B、Mini-B 或 Micro-B 型插头插入本地 USB 接口, 则认为此器件为 B 型器件。Mini-B 型插头和接口仅支持专用外设, 不支持双重角色/OTG 器件。

双重角色 B 器件按照以下流程图和状态描述表所示运行。

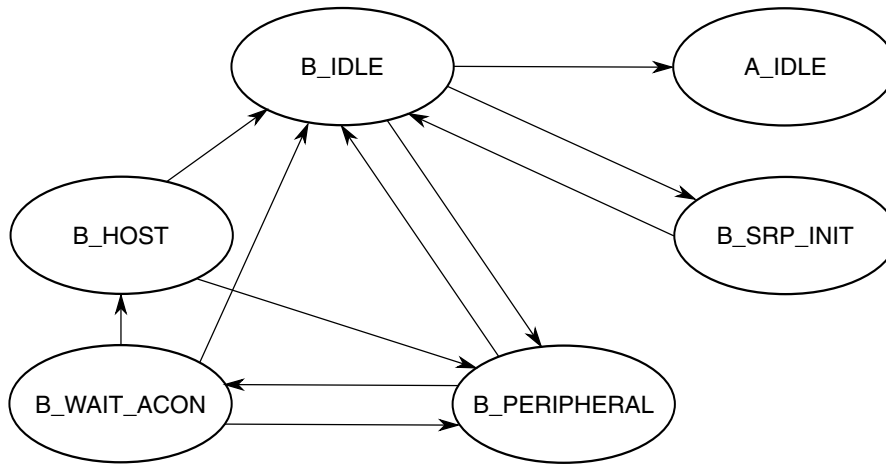


图 41-120. 双重角色 B 器件流程图

表 41-151. 双重角色 B 器件的流程状态描述

状态	操作	响应
B_IDLE	如果 ID\ 中断, 已插入 A 型电缆, 器件现在用作 A 型器件,	转至 A_IDLE。
	如果 B_SESS_VLD 中断, A 器件已打开 VBUS 并开始会话,	转至 B_PERIPHERAL。 打开 DP_HIGH。
	如果 B 应用需要总线, 总线闲置 2 ms 且 B_SESS_END 位已置位, 则 B 器件可执行 SRP,	转至 B_SRPR_INIT。 脉冲 CHRG_VBUS 脉冲 DP_HIGH 5-10 ms。

下一页继续介绍此表...

表 41-151. 双重角色 B 器件的流程状态描述 (继续)

状态	操作	响应
B_SRP_INIT	如果 ID\ 中断或 SRP 完成 (必须在不到 100 ms 内完成 SRP。),	转至 B_IDLE。
B_PERIPHERAL	如果 HNP 已使能, 总线挂起并且 B 器件想要使用总线, 则 B 器件变成主机,	转至 B_WAIT_ACON。 关闭 DP_HIGH。
B_WAIT_ACON	如果 A 连接, 则会接收到连接中断,	转至 B_HOST。 打开主机模式。
	如果 ID\ 中断或 B_SESS_VLD\ 中断, 如果电缆更换或如果 VBUS 离开, 则主机不支持 us, 转至 B_IDLE,	转至 B_IDLE。
	如果 3.125 ms 到期或如果发生恢复,	转至 B_PERIPHERAL。
B_HOST	如果 ID\ 中断或 B_SESS_VLD\ 中断, 如果电缆更换或如果 VBUS 离开, 则主机不支持,	转至 B_IDLE。
	如果 B 应用完成或 A 断开,	转至 B_PERIPHERAL。

41.9 从设备模式 IRC48M 操作

下面是 IRC48M 初始化代码步骤:

1. 使能 IRC48M 时钟: `USB_CLK_RECOVER_IRC_EN[IRC_EN] = 1b` in the SCG module
2. 使能 USB 时钟恢复调谐:
`USB_CLK_RECOVER_CTRL[CLOCK_RECOVER_EN] = 1b`
3. 通过配置多路复用器和分频器, 选择 USB 的时钟源。通过设置 `SIM_SOPT2[MCGPLLFL] = 11b` 以供 USB 使用, IRC48M 实现多路复用。
4. 所选多路复用器输出时钟可由 USB 时钟分频器分频, 所以要在设置这些字段时, 确保不会使能时钟分频。分频公式如下:

分频器输出时钟 = 分频器输入时钟 × [(USBFRAC+1) / (USBDIV+1)].
因此, 设置 `SIM_CLKDIV2[USBDIV] = 000b` 且 `SIM_CLKDIV2[USBFRAC] = 0b`。

5. USB 时钟源必须选择分频时钟输出。

有关芯片特性的详细信息, 请参见“时钟分布”一章中的 USB FS OTG 控制器时钟信息。

第 42 章

CAN (FlexCAN)

42.1 此模块的芯片实现细节

42.1.1 FlexCAN 模块的数量

对于 KS22，该器件包含 2 个相同的 FlexCAN 模块。对于 KS20，该器件只有一个 FlexCAN 模块。

42.1.2 MDIS 位复位值

CAN_MCR[MDIS]位在复位后置位。因此，FlexCAN 模块将在复位之后被禁用。

42.1.3 消息缓冲区数

外部 FlexCAN 模块包含 16 个消息缓冲区。每个消息缓冲区为 16 个字节。

42.1.4 FlexCAN 时钟

42.1.4.1 时钟选项

CAN_CTRL1[CLKSRC]寄存器位选择从内部总线时钟或外部输入时钟 (OSCERCLK)来为 FlexCAN 计时。在这种情况下，不得使用带有 PLL FM 抖动的时钟源。

42.1.4.2 时钟门控

每个 CAN 模块的时钟可通过 SCGCn[CANx] 位打开或关闭门控。复位后这些位默认为 0，从而将禁用相应模块的时钟。在 FlexCAN 模块初始化之前，应置位对应的 FlexCAN 门控位。只有使能门控位后，FlexCAN 的寄存器才能正常被访问。

42.1.5 FlexCAN 中断

FlexCAN 有多个中断请求源。然而，其中部分源都经 OR 运算相结合，以生成单个中断请求。有关各中断源与中断请求的映射关系，请参阅以下信息：

请求	来源
消息缓冲器	消息缓冲器 0-15
总线关闭	总线关闭
错误	<ul style="list-style-type: none"> • 位 1 错误 • 位 0 错误 • 应答错误 • 循环冗余校验(CRC)错误 • 形式错误 • 填充错误 • 发送错误警告 • 接收错误警告
发送警告	发送警告
接收警告	接收警告
唤醒	唤醒

42.1.6 低功耗模式下的 FlexCAN 操作

FlexCAN 模块可在 VLPR 和 VLPW 模式下操作。采用 2 MHz 总线时钟时，支持的最快 FlexCAN 传输速率为 250 kbps。必须根据新频率调节模块的位时序参数，在 VLPR 和 VLPW 模式下，FlexCAN 仍然保留全部功能。

可配置 FlexCAN 模块，以便在 STOP 和 VLPS 模式下生成唤醒中断。当 FlexCAN 配置用于生成唤醒时，CAN 总线上的隐性至显性转换会生成一个中断。

42.1.7 FlexCAN 等待模式

FlexCAN 模块的等待模式等同于芯片的等待和 VLPW 模式。

42.1.8 FlexCAN 毛刺滤波器

此芯片支持通过 CAN 唤醒中断，将 FlexCAN 模块从停止和等待模式中唤醒。CAN 总线的隐性至显性转换(CAN_RX)可将芯片从停止或等待模式中唤醒。一个可选的毛刺滤波器连接至 CAN_RX 上的中断生成逻辑路径。

毛刺滤波器提供以下功能：

- 滤除 CAN 总线中无用的噪音
- 通过已过滤的或未过滤的 CAN 总线选择唤醒源
- 将唤醒源布设到 FlexCAN 模块的同步（等待）或异步（停止）唤醒路径中

毛刺滤波器的参考时钟是一个源自 MCGIRCLK 的 4 MHz 时钟。如果用户想要通过毛刺滤波器实现低功耗唤醒，则必须让 MCGIRCLK 保持开启状态。毛刺滤波器在计数 11 个 4 MHz 时钟周期之后，才会将其认定为有效的隐性至显性转换。

42.2 简介

FlexCAN 模块是一个通信控制器，它根据 ISO 11898-1 标准和 CAN 2.0 B 协议规范实施 CAN 协议。一般结构框图如下图所示，该图介绍了 FlexCAN 模块中设置的主要子块，包括一个相关存储器，用于存储消息缓冲区、接收全局掩码寄存器、接收单独掩码寄存器、接收 FIFO 滤波器和接收 FIFO ID 滤波器。子模块的功能如以下几节所述。

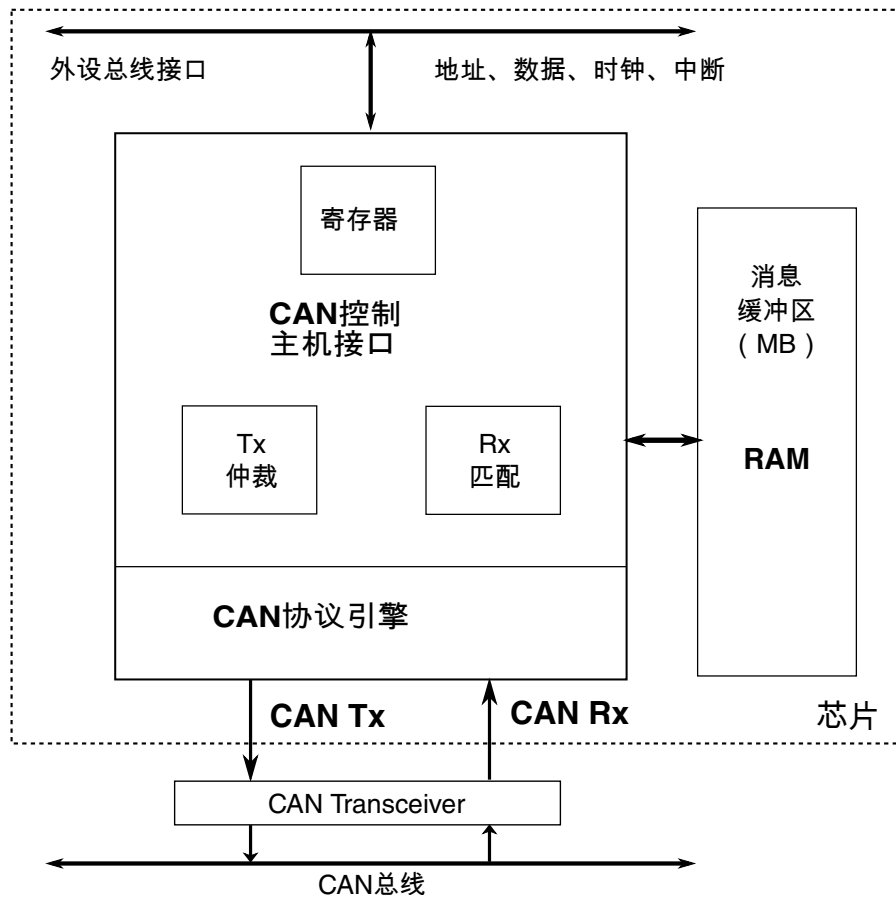


图 42-1. FlexCAN 结构框图

42.2.1 概述

CAN 协议主要设计用作车载串行数据总线，满足该领域的特定要求：

- 实时处理
- 在车辆 EMI 环境下可靠运行
- 高性价比
- 所需带宽

FlexCAN 全面支持 CAN 2.0 B 版本规范，支持传输标准和扩展消息帧。消息缓冲器存储于 FlexCAN 模块专用的嵌入式 RAM 中。请参阅芯片配置详情，了解芯片中实际配置的消息缓冲器数。

协议引擎(PE)子模块负责管理 CAN 总线上的串行通信：

- 请求 RAM 访问，以便接收和发送消息帧
- 验证接收到的消息
- 执行错误处理

控制器主机接口(CHI)子模块处理接收和发送所用的消息缓冲器选择，以及。

总线接口单元(BIU)子模块控制往来于内部接口总线的访问，以便建立到 CPU 和其他数据块的连接。时钟、地址和数据总线、中断输出、DMA 和测试信号均通过 BIU 访问。

42.2.2 FlexCAN 模块特性

FlexCAN 模块包括这些显著特性:

- 全面实现 CAN 协议规范，2.0 B 版
 - 标准数据帧
 - 扩展数据帧
 - 0 至 8 字节数据长度
 - 可编程比特率最高达 1 Mb/s
 - 内容相关寻址
- 符合 ISO 11898-1 标准
- 0 至 8 字节数据长度的灵活邮箱
- 每个邮箱可配置为接收或发送，全部支持标准和扩展消息
- 每个邮箱的单独 Rx 掩码寄存器
- 功能完备的 Rx FIFO, 提供高达 6 帧的存储容量和带 DMA 支持的自动内部指针处理功能
- 传输终止功能
- 灵活的消息缓冲器(MB), 总共 16 个数据长度为 8 字节的消息缓冲器, 可配置为 Rx 或 Tx
- 可编程时钟源 (外设时钟或振荡器时钟)
- 接收或传输结构未使用的 RAM 可用作通用 RAM 空间
- 仅监听模式功能
- 可编程回环模式支持自检操作
- 可编程传输优先级机制: 最低 ID、最低缓冲区数或最高优先级
- 基于 16 位自由运行定时器的时间戳
- 全局网络时间, 通过特定消息同步

- 可屏蔽中断
- 不受传输媒介影响（假定为外部收发器）
- 适用于高优先级报文的仲裁方案可缩短延迟时间
- 低功耗模式，CAN 总线上的信号可触发唤醒中断
- 远程请求帧可自动或由软件处理
- 仅可在冻结模式下写入 CAN 位时间设置和配置位
- Tx 邮箱状态（最低优先级缓冲器或空缓冲器）
- 用于已接收帧的标识符验收滤波器命中指示器（IDHIT）
- SYNCH 位可用于状态 1 寄存器中的错误，用于通知此模块与 CAN 总线同步
- 已发送消息的 CRC 状态
- Rx FIFO 全局掩码寄存器
- 匹配过程中邮箱和 Rx FIFO 之间的可选优先级
- 强大的 Rx FIFO ID 过滤器，可以根据 128 扩展、256 标准或 512 部分（8 位）ID 匹配输入 ID，带高达 32 个单独屏蔽功能
- 100% 向后兼容以前版本的 FlexCAN

42.2.3 工作模式

FlexCAN 模块有三种功能模式：

- 正常模式（普通模式或者管理者模式）：

正常模式下，模块接收"与"或者"或"发送消息帧，正常处理错误，并且所有 CAN 协议功能均使能。普通模式或者管理者模式的区别在于对某些受限控制寄存器的访问。
- 冻结模式：

冻结模式在 MCR 中的 FRZ 位置位时使能。如果使能，在 MCR[HALT]置位或者在芯片进入调试模式并且 MCR[FRZ_ACK]被 FlexCAN 置位时，模块将进入冻结模式。在这种模式下，不会进行帧发送和接收操作，还会丢失与 CAN 总线的同步。更多详情，请参阅[冻结模式](#)。
- 回环模式：

当控制 1 寄存器中的 LPB 字段被置位时，模块会进入此模式。在此模式下，FlexCAN 会执行内部回环自测。发送器的比特流输出在内部回馈接收器输入端。Rx CAN 输入引脚被忽略，且 Tx CAN 输出进入隐性状态（逻辑‘1’）。FlexCAN 同正常模式一样发送消息，并将其自己发送的消息视为从远程节点接收到的消息。在该状态下，FlexCAN 会忽略 CAN 帧确认字段中 ACK 槽期间传送的位，从而确保正确接收到自己的消息。发送中断和接收中断均会生成。

- 只听模式：

当控制 1 寄存器中的 LOM 字段置位时，模块会进入此模式。在此模式下，传输被禁用，所有错误计数器都被冻结，并且模块在 CAN 错误忽略模式下工作。只会接收另一 CAN 站点确认的消息。如果 FlexCAN 检测到尚未被确认的消息，则会标志 BIT0 错误（不改变 REC），就如它要尝试确认消息一样。

对于低功耗工作，FlexCAN 模块有：

- 模块禁用模式：

当 MCR 中的 MDIS 位由 CPU 置位且 LPM_ACK 由 FlexCAN 置位时，模块将进入此低功耗模式。禁用时，模块会请求禁用 CAN 协议引擎和控制器主机接口子模块的时钟。取反 MCR 寄存器中的 MDIS 位则可退出此模式。更多详情，请参阅[模块禁用模式](#)。

- 休眠模式：

当 MCR 中的 DOZE 位置位且在芯片级请求休眠模式，并且 MCR 寄存器中的 LPM_ACK 位由 FlexCAN 置位时，模块将进入此低功耗模式。处于休眠模式时，模块会请求禁用 CAN 协议引擎和 CAN 控制器主机接口子模块的时钟。当 MCR 中的 DOZE 位清 0 时，当芯片退出休眠模式时，或者当在 CAN 总线上检测到活动并且自我唤醒机制使能时，模块将退出此模式。更多详情，请参阅[休眠模式](#)。

- 停止模式：

当在芯片级请求停止模式，并且 MCR 寄存器中的 LPM_ACK 位由 FlexCAN 置位时，模块将进入此低功耗模式。当在停止模式下时，模块将自己置于无效状态，然后通知 CPU 可以关闭全局时钟。当停止请求消除时，或者当在 CAN 总线上检测到活动并且自我唤醒机制使能时，模块将退出此模式。更多详情，请参阅[Stop 模式](#)。

42.3 FlexCAN 信号说明

FlexCAN 模块具有两个 I/O 信号连接至外部芯片引脚。这些信号在下表中综述，并且将在后续章节详述。

表 42-1. FlexCAN 信号说明

信号	说明	I/O
CAN Rx	CAN 接收引脚	输入
CAN Tx	CAN 发送引脚	输出

42.3.1 CAN Rx

该引脚是 CAN 总线收发器的接收引脚。显性状态表示为逻辑电平 0。接收状态表示为逻辑电平 1。

42.3.2 CAN Tx

该引脚是 CAN 总线收发器的发送引脚。显性状态表示为逻辑电平 0。接收状态表示为逻辑电平 1。

42.4 存储器映射/寄存器定义

本节描述 FlexCAN 模块中的各寄存器和数据结构。模块的基地址取决于芯片的特定存储器映射。

42.4.1 FlexCAN 存储器映射

FlexCAN 模块存储器映射如下表所示。

FlexCAN 占用的地址空间有 128 个字节用于从模块基地址开始的寄存器，其后是从地址偏移 0x0080 开始的嵌入式 RAM。

各个寄存器由其全名及对应的助记符加以识别。The access type can be Supervisor (S) or Unrestricted (U). 通过设置 MCR 寄存器中的 SUPV 字段，可以将多数寄存器配置为管理员访问或无限制访问。这些寄存器可在 [表 42-2](#)。

表 42-2. 寄存器访问和复位信息

寄存器	访问类型	受硬复位影响	受软复位影响
模块配置寄存器(CAN_MCR)	S	有	是
控制 1 寄存器(CAN_CTRL1)	S/U	是	否
自由运行定时器寄存器(CAN_TIMER)	S/U	是	是
Rx 邮箱全局掩码寄存器(CAN_RXMGMASK)	S/U	否	否
Rx 缓冲器 14 掩码寄存器(CAN_RX14MASK)	S/U	否	否
Rx 缓冲器 15 掩码寄存器(CAN_RX15MASK)	S/U	否	否
错误计数器寄存器(CAN_ECR)	S/U	是	是
错误和状态 1 寄存器(CAN_ESR1)	S/U	是	是
中断掩码 1 寄存器(CAN_IMASK1)	S/U	是	是
中断标志 1 寄存器(CAN_IFLAG1)	S/U	是	是
控制 2 寄存器(CAN_CTRL2)	S/U	是	否
错误和状态 2 寄存器(CAN_ESR2)	S/U	是	是
CRC 寄存器(CAN_CRCR)	S/U	是	是
Rx FIFO 全局掩码寄存器(CAN_RXFGMASK)	S/U	否	否
Rx FIFO 信息寄存器(CAN_RXFIR)	S/U	否	否
CAN 位计时寄存器(CAN_CBT)	S/U	是	否
消息缓冲器	S/U	否	否
Rx 单独掩码寄存器	S/U	否	否

FlexCAN 模块可以用邮箱和 Rx FIFO 结构存储 CAN 传送和接收消息。

下表显示了 FlexCAN 存储器映射情况。

偏移量 0x80 至 0x17F 范围内的地址分配 16 个 128 位的消息缓冲器 (MB)。

关于信息缓存的存储器映射情况，可参见 [FlexCAN 消息缓冲器寄存器映射](#)。

CAN 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4002_4000	模块配置寄存器 (CAN0_MCR)	32	不适用	参见章节	42.4.2/883
4002_4004	控制 1 寄存器 (CAN0_CTRL1)	32	不适用	0000_0000h	42.4.3/887
4002_4008	自由运行定时器 (CAN0_TIMER)	32	不适用	0000_0000h	42.4.4/891
4002_4010	Rx 邮箱全局掩码寄存器 (CAN0_RXMGMASK)	32	不适用	未定义	42.4.5/892
4002_4014	Rx 14 掩码寄存器 (CAN0_RX14MASK)	32	不适用	未定义	42.4.6/893
4002_4018	Rx 15 掩码寄存器 (CAN0_RX15MASK)	32	不适用	未定义	42.4.7/894
4002_401C	错误计数器 (CAN0_ECR)	32	不适用	0000_0000h	42.4.8/894
4002_4020	错误和状态 1 寄存器 (CAN0_ESR1)	32	不适用	参见章节	42.4.9/896

下一页继续介绍此表...

CAN 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4002_4028	中断掩码 1 寄存器 (CAN0_IMASK1)	32	不适用	0000_0000h	42.4.10/902
4002_4030	中断标志 1 寄存器 (CAN0_IFLAG1)	32	不适用	0000_0000h	42.4.11/902
4002_4034	控制 2 寄存器 (CAN0_CTRL2)	32	不适用		42.4.12/904
4002_4038	错误和状态 2 寄存器 (CAN0_ESR2)	32	不适用	0000_0000h	42.4.13/908
4002_4044	CRC 寄存器 (CAN0_CRCCR)	32	不适用	0000_0000h	42.4.14/909
4002_4048	Rx FIFO 全局掩码寄存器 (CAN0_RXFGMASK)	32	不适用	未定义	42.4.15/909
4002_404C	Rx FIFO 信息寄存器 (CAN0_RXFIR)	32	不适用	未定义	42.4.16/910
4002_4050	CAN 位时序寄存器 (CAN0_CBT)	32	不适用	参见章节	42.4.17/911
4002_4880	Rx 单独掩码寄存器 (CAN0_RXIMR0)	32	不适用	未定义	42.4.18/912
4002_4884	Rx 单独掩码寄存器 (CAN0_RXIMR1)	32	不适用	未定义	42.4.18/912
4002_4888	Rx 单独掩码寄存器 (CAN0_RXIMR2)	32	不适用	未定义	42.4.18/912
4002_488C	Rx 单独掩码寄存器 (CAN0_RXIMR3)	32	不适用	未定义	42.4.18/912
4002_4890	Rx 单独掩码寄存器 (CAN0_RXIMR4)	32	不适用	未定义	42.4.18/912
4002_4894	Rx 单独掩码寄存器 (CAN0_RXIMR5)	32	不适用	未定义	42.4.18/912
4002_4898	Rx 单独掩码寄存器 (CAN0_RXIMR6)	32	不适用	未定义	42.4.18/912
4002_489C	Rx 单独掩码寄存器 (CAN0_RXIMR7)	32	不适用	未定义	42.4.18/912
4002_48A0	Rx 单独掩码寄存器 (CAN0_RXIMR8)	32	不适用	未定义	42.4.18/912
4002_48A4	Rx 单独掩码寄存器 (CAN0_RXIMR9)	32	不适用	未定义	42.4.18/912
4002_48A8	Rx 单独掩码寄存器 (CAN0_RXIMR10)	32	不适用	未定义	42.4.18/912
4002_48AC	Rx 单独掩码寄存器 (CAN0_RXIMR11)	32	不适用	未定义	42.4.18/912
4002_48B0	Rx 单独掩码寄存器 (CAN0_RXIMR12)	32	不适用	未定义	42.4.18/912
4002_48B4	Rx 单独掩码寄存器 (CAN0_RXIMR13)	32	不适用	未定义	42.4.18/912

下一页继续介绍此表...

CAN 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4002_48B8	Rx 单独掩码寄存器 (CAN0_RXIMR14)	32	不适用	未定义	42.4.18/912
4002_48BC	Rx 单独掩码寄存器 (CAN0_RXIMR15)	32	不适用	未定义	42.4.18/912
4002_5000	模块配置寄存器 (CAN1_MCR)	32	不适用	参见章节	42.4.2/883
4002_5004	控制 1 寄存器 (CAN1_CTRL1)	32	不适用	0000_0000h	42.4.3/887
4002_5008	自由运行定时器 (CAN1_TIMER)	32	不适用	0000_0000h	42.4.4/891
4002_5010	Rx 邮箱全局掩码寄存器 (CAN1_RXMGMASK)	32	不适用	未定义	42.4.5/892
4002_5014	Rx 14 掩码寄存器 (CAN1_RX14MASK)	32	不适用	未定义	42.4.6/893
4002_5018	Rx 15 掩码寄存器 (CAN1_RX15MASK)	32	不适用	未定义	42.4.7/894
4002_501C	错误计数器 (CAN1_ECR)	32	不适用	0000_0000h	42.4.8/894
4002_5020	错误和状态 1 寄存器 (CAN1_ESR1)	32	不适用	参见章节	42.4.9/896
4002_5028	中断掩码 1 寄存器 (CAN1_IMASK1)	32	不适用	0000_0000h	42.4.10/902
4002_5030	中断标志 1 寄存器 (CAN1_IFLAG1)	32	不适用	0000_0000h	42.4.11/902
4002_5034	控制 2 寄存器 (CAN1_CTRL2)	32	不适用		42.4.12/904
4002_5038	错误和状态 2 寄存器 (CAN1_ESR2)	32	不适用	0000_0000h	42.4.13/908
4002_5044	CRC 寄存器 (CAN1_CRCCR)	32	不适用	0000_0000h	42.4.14/909
4002_5048	Rx FIFO 全局掩码寄存器 (CAN1_RXFGMASK)	32	不适用	未定义	42.4.15/909
4002_504C	Rx FIFO 信息寄存器 (CAN1_RXFIR)	32	不适用	未定义	42.4.16/910
4002_5050	CAN 位时序寄存器 (CAN1_CBT)	32	不适用	参见章节	42.4.17/911
4002_5880	Rx 单独掩码寄存器 (CAN1_RXIMR0)	32	不适用	未定义	42.4.18/912
4002_5884	Rx 单独掩码寄存器 (CAN1_RXIMR1)	32	不适用	未定义	42.4.18/912
4002_5888	Rx 单独掩码寄存器 (CAN1_RXIMR2)	32	不适用	未定义	42.4.18/912
4002_588C	Rx 单独掩码寄存器 (CAN1_RXIMR3)	32	不适用	未定义	42.4.18/912
4002_5890	Rx 单独掩码寄存器 (CAN1_RXIMR4)	32	不适用	未定义	42.4.18/912
4002_5894	Rx 单独掩码寄存器 (CAN1_RXIMR5)	32	不适用	未定义	42.4.18/912
4002_5898	Rx 单独掩码寄存器 (CAN1_RXIMR6)	32	不适用	未定义	42.4.18/912
4002_589C	Rx 单独掩码寄存器 (CAN1_RXIMR7)	32	不适用	未定义	42.4.18/912

下一页继续介绍此表...

CAN 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4002_58A0	Rx 单独掩码寄存器 (CAN1_RXIMR8)	32	不适用	未定义	42.4.18/912
4002_58A4	Rx 单独掩码寄存器 (CAN1_RXIMR9)	32	不适用	未定义	42.4.18/912
4002_58A8	Rx 单独掩码寄存器 (CAN1_RXIMR10)	32	不适用	未定义	42.4.18/912
4002_58AC	Rx 单独掩码寄存器 (CAN1_RXIMR11)	32	不适用	未定义	42.4.18/912
4002_58B0	Rx 单独掩码寄存器 (CAN1_RXIMR12)	32	不适用	未定义	42.4.18/912
4002_58B4	Rx 单独掩码寄存器 (CAN1_RXIMR13)	32	不适用	未定义	42.4.18/912
4002_58B8	Rx 单独掩码寄存器 (CAN1_RXIMR14)	32	不适用	未定义	42.4.18/912
4002_58BC	Rx 单独掩码寄存器 (CAN1_RXIMR15)	32	不适用	未定义	42.4.18/912

42.4.2 模块配置寄存器 (CANx_MCR)

此寄存器定义全局系统配置，比如模块工作模式和最大消息缓冲器配置。

地址: 基址 基准 + 0h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R	MDIS	FRZ	RFEN	HALT	NOTRDY	WAKMSK	SOFTTRST	FRZACK	SUPV	SLFWAK	WRNEN	LPMACK	WAKSRC	DOZE	SRXDIS	IRMQ	
W	MDIS	FRZ	RFEN	HALT	NOTRDY	WAKMSK	SOFTTRST	FRZACK	SUPV	SLFWAK	WRNEN	LPMACK	WAKSRC	DOZE	SRXDIS	IRMQ	
复位	1	1	0	1	1	0	0	0	1	0	0	1	0	0	0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	DMA	0	LPRIOEN	AEN	0	0	IDAM	0	MAXMB								
W	DMA		LPRIOEN	AEN			IDAM		MAXMB								
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

CANx_MCR 字段描述

字段	描述
31 MDIS	<p>模块禁用</p> <p>此位控制是否使能 FlexCAN。禁用时，FlexCAN 会控制 CAN 协议引擎和控制器主机接口子模块的时钟。此位不受软复位的影响。</p> <p>0 使能 FlexCAN 模块。 1 禁用 FlexCAN 模块。</p>
30 FRZ	<p>冻结使能</p> <p>FRZ 位指定 FlexCAN 在 CAN_MCR 寄存器中的 HALT 位置位或者在芯片级请求调试模式时的行为。当 FRZ 置位时，FlexCAN 可进入冻结模式。此位字段清 0 会导致 FlexCAN 退出冻结模式。</p> <p>0 退出冻结模式。 1 进入冻结模式。</p>
29 RFEN	<p>Rx FIFO 使能</p> <p>此位控制是否使能 Rx FIFO 特性。当 RFEN 置位时，MB 0 至 5 可用于正常接收和传输，因为 FIFO 引擎和额外的 MB (最多 32 个，取决于 CAN_CTRL2[RFFN]设置) 都会使用对应的存储器区域(0x80-0xDC)，这些 MB 又被用作 Rx FIFO ID 滤波器表元素。RFEN 也会影响每个 CAN 位的最小外设时钟数量，详见表格“外设时钟频率与 CAN 比特率的最小比值”(见仲裁和匹配时序)。此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>0 Rx FIFO 未使能。 1 Rx FIFO 已使能。</p>
28 HALT	<p>冻结 FlexCAN</p> <p>此位置位将使 FlexCAN 模块进入冻结模式。CPU 应在初始化消息缓冲器和定制寄存器 CAN_CTRL1 及 CAN_CTRL2 之后清零此位。在此位清零之前，FlexCAN 不会执行接收或传输操作。当 FlexCAN 处于低功耗模式时，不能进入冻结模式。</p> <p>0 无冻结模式请求。 1 如果 FRZ 位置位则进入冻结模式。</p>
27 NOTRDY	<p>FlexCAN 未就绪</p> <p>此只读位指示 FlexCAN 处于禁用模式、休眠模式、停止模式或冻结模式。一旦 FlexCAN 退出这些模式，此位将清 0。此位不受软复位的影响。</p> <p>0 FlexCAN 模块处于正常模式、仅监听模式或回环模式。 1 FlexCAN 处于禁用模式、休眠模式、停止模式或冻结模式。</p>
26 WAKMSK	<p>唤醒中断掩码</p> <p>此位在自我唤醒机制下使能唤醒中断生成功能。</p> <p>0 唤醒中断已禁用。 1 唤醒中断已使能。</p>
25 SOFTRST	<p>软复位</p> <p>当此位置位时，FlexCAN 会复位其内部状态机和部分存储器映射寄存器。</p> <p>SOFTRST 位可在写入 MCR 寄存器时直接由 CPU 置位，但也可在芯片级请求全局软复位时置位。由于软复位是同步的并且必须跨时钟域遵循请求/确认程序，所以，可能需要一些时间才能生效。SOFTRST 位在复位挂起时将保持置位，并在复位完成时自动清 0。因此，软件可以轮询此位，以了解软复位的完成时间。</p>

下一页继续介绍此表...

CANx_MCR 字段描述 (继续)

字段	描述
	<p>当时钟在低功耗模式下关闭时，不能运用软复位。必须先使模块退出低功耗模式，然后才能运用软复位。此位不受软复位的影响。</p> <p>0 无复位请求。 1 复位受软复位影响的寄存器。</p>
24 FRZACK	<p>冻结模式确认</p> <p>此只读位指示 FlexCAN 处于冻结模式并且其预分频器已停止。在当前传输或接收进程完成之后，不能允许冻结模式请求。因此，软件可以轮询 FRZACK 位，以了解 FlexCAN 已实际于何时进入冻结模式。如果冻结模式请求清 0，则此位在 FlexCAN 预分频器再次运行之后清 0。如果在 FlexCAN 处于低功耗模式时请求冻结模式，FRZACK 位只会在退出低功耗模式时置位。见“冻结模式”一节。此位不受软复位的影响。</p> <p>注：FRZACK 将在 CPU 发出冻结模式请求 178 CAN 位之内置位，并在冻结模式请求消除后 2 CAN 位内清 0 (见“协议时序”一节)。</p> <p>0 FlexCAN 不处于冻结模式，预分频器正在运行。 1 FlexCAN 处于冻结模式，预分频器已停止。</p>
23 SUPV	<p>管理员模式</p> <p>此位将 FlexCAN 配置为管理员模式或用户模式。在模块存储器映射表的访问类型栏中，受此位影响的寄存器标记为 S/U。此位的复位值为 1，所有寄存器都可以被访问。此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>0 FlexCAN 处于用户模式。受影响的寄存器允许管理员和无限制两种访问。 1 FlexCAN 处于管理员模式。受影响的寄存器只允许管理员访问。无限制访问行为就如对不存在的寄存器位置的访问一样。</p>
22 SLFWAK	<p>自我唤醒</p> <p>此位在 FlexCAN 处于禁用模式之外的低功耗模式时，使能自我唤醒特性。当此特性使能时，FlexCAN 模块将监控总线中的唤醒事件，即从隐性至显性的转换事件。</p> <p>如果在休眠模式中检测到唤醒事件，FlexCAN 会请求恢复其时钟，并且如果允许这样做，会向 CPU 生成一个唤醒中断。</p> <p>如果在停止模式中检测到唤醒事件，FlexCAN (若允许这样做) 会向 CPU 生成一个唤醒中断，这样，CPU 就能全面退出停止模式，并且 FlexCAN 可以请求恢复时钟。</p> <p>当 FlexCAN 处于禁用模式以外的低功耗模式时，不能写入此位，因为硬件阻止这样做。</p> <p>0 FlexCAN 自我唤醒特性已禁用。 1 FlexCAN 自我唤醒特性已使能。</p>
21 WRNEN	<p>警告中断使能</p> <p>置位时，此位将使能在错误和状态寄存器 1(ESR1)中生成 TWRNINT 和 RWRNINT 标志。如果 WRNEN 清 0，TWRNINT 和 RWRNINT 标志将始终为零，不受错误计数器值的影响，并且不会生成警告中断。此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>0 TWRNINT 和 RWRNINT 位为零，独立于错误计数器的值。 1 TWRNINT 和 RWRNINT 位在相应的错误计数器从小于 96 转换为大于或等于 96 时置位。</p>
20 LPMACK	<p>低功耗模式确认</p> <p>此只读位指示 FlexCAN 处于低功耗模式 (禁用模式、休眠模式、停止模式)。在全部当前传输或接收进程完成之前，不能进入低功耗模式，因而此，CPU 可以轮询 LPMACK 位，以了解 FlexCAN 实际上是何时进入低功耗模式的。此位不受软复位的影响。</p>

下一页继续介绍此表...

CANx_MCR 字段描述 (继续)

字段	描述
	<p>注: LPMACK 将在 CPU 发出低功耗模式请求 180 CAN 位时间内置位, 并在低功耗模式请求消除后 2 CAN 时间内清零 (见“协议时序”一节)。</p> <p>0 FlexCAN 不处于低功耗模式。 1 FlexCAN 处于低功耗模式。</p>
19 WAKSRC	<p>唤醒源</p> <p>此位定义是否采用集成式低通滤波器来防止 Rx CAN 输入受杂散唤醒影响。此位仅可在冻结模式下写入, 因为在其他模式下它被硬件禁用。</p> <p>0 FlexCAN 运用未经滤波的 Rx 输入来检测 CAN 总线上的隐性至显性边沿。 1 FlexCAN 运用经滤波的 Rx 输入来检测 CAN 总线上的隐性至显性边沿。</p>
18 DOZE	<p>休眠模式使能</p> <p>此位定义当在芯片级请求休眠模式时, 是否允许 FlexCAN 进入低功耗模式。当 FlexCAN 因在 CAN 总线上检测活动而从休眠模式唤醒时 (自我唤醒被使能), 此位将自动复位。</p> <p>0 FlexCAN 不能在休眠模式被请求时进入低功耗模式。 1 FlexCAN 可以在休眠模式被请求时进入低功耗模式。</p>
17 SRXDIS	<p>自我接收禁用</p> <p>此位定义是否允许 FlexCAN 接收自己发送的帧。如果此位置位, 则模块发送的帧不会存入任何 MB, 无论为该 MB 配置的 ID 是否与发送帧匹配, 并且因帧接收的影响, 不会生成中断标志或中断信号。此位仅可在冻结模式下写入, 因为在其他模式下它被硬件禁用。</p> <p>0 自我接收已使能。 1 自我接收已禁用。</p>
16 IRMQ	<p>单独 Rx 掩码和队列使能</p> <p>此位指示 Rx 匹配过程将基于单独掩码和队列, 还是基于掩码方案 (即 CAN_RXMGMASK、CAN_RX14MASK、CAN_RX15MASK 和 CAN_RXFGMASK)。此位仅可在冻结模式下写入, 因为在其他模式下它被硬件禁用。</p> <p>0 单独 Rx 掩码和队列特性已禁用。为了实现与传统应用的向后兼容性, 读取 C/S 字即会锁定 MB, 即使该 MB 为 EMPTY, 也是如此。 1 单独 Rx 掩码和队列特性已使能。</p>
15 DMA	<p>DMA 使能</p> <p>DMA 使能位控制是否使能 DMA 特性。DMA 特性只能用于 Rx FIFO, 结果, CAN_MCR[RFEN]位必须置位。当 DMA 和 RFEN 置位时, CAN_IFLAG1[BUF5]会生成 DMA 请求, 并且不会生成 RX FIFO 中断。此位仅可在冻结模式下写入, 因为在其他模式下它被硬件禁用。</p> <p>0 RX FIFO 的 DMA 特性已禁用。 1 RX FIFO 的 DMA 特性已使能。</p>
14 保留	<p>此只读字段为保留字段且值始终为 0。</p>
13 LPRIOEN	<p>本地优先级使能</p> <p>此位用于实现与传统应用的向后兼容性。它控制是否使能本地优先级特性, 用于扩展仲裁过程中使用的 ID。在这种扩展 ID 概念下, 仲裁过程基于完整的 32 字, 但实际发送的 ID 仍然有 11 位用于标准帧, 29 位用于扩展帧。此位仅可在冻结模式下写入, 因为在其他模式下它被硬件禁用。</p>

下一页继续介绍此表...

CANx_MCR 字段描述 (继续)

字段	描述
	<p>0 本地优先级已禁用。</p> <p>1 本地优先级已使能。</p>
12 AEN	<p>中止使能</p> <p>置位时，此位使能 Tx 中止机制。此机制可以确保挂起传输中止程序的安全性，这样，在 CAN 总线中发送任何帧时都会发出通知。此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>注：当 CAN_MCR[AEN]置位时，只需要使用中止机制（见 传输中止机制）来更新针对传输配置的邮箱。</p> <p>警告：当 CAN_MCR[AEN]置位时，将中止代码写入 Rx 邮箱可能导致不可预测的结果。</p> <p>0 中止已禁用。</p> <p>1 中止已使能。</p>
11 保留	此只读字段为保留字段且值始终为 0。
10 保留	此只读字段为保留字段且值始终为 0。
9–8 IDAM	<p>ID 接受模式</p> <p>此 2 位字段标识 Rx FIFO ID 滤波器表元素的格式。注意，该表中的所有元素都由此字段同时配置（它们具有相同的格式）。见“Rx FIFO 结构”一节。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>00 格式 A：每个 ID 滤波器表元素一个完整 ID（标准和扩展）。</p> <p>01 格式 B：每个 ID 滤波器元素两个完整标准 ID 或两个部分 14 位（标准和扩展）ID。</p> <p>10 格式 C：每个 ID 滤波器表元素四个部分 8 位标准 ID。</p> <p>11 格式 D：拒绝全部帧。</p>
7 保留	此只读字段为保留字段且值始终为 0。
MAXMB	<p>消息邮箱个数</p> <p>这个 7 位的字段定义将参与匹配和仲裁过程的最后消息缓冲器数量。复位值(0x0F)等于一个 16 MB 配置。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>最后 MB 的数量 = MAXMB</p> <p>注：MAXMB 的配置值必须小于或等于可用消息缓冲器数量所述。</p> <p>另外，MAXMB 值的定义必须考虑 Rx FIFO 占用的 MB 区域以及由 CAN_CTRL2 寄存器中 RFFN 位定义的其 ID 滤波器表空间。MAXMB 也会影响每个 CAN 位的最小外设时钟数量，详见表格“外设时钟频率与 CAN 比特率的最小比值”（见 仲裁和匹配时序）。</p>

42.4.3 控制 1 寄存器 (CANx_CTRL1)

此寄存器针对与 CAN 总线相关的特定 FlexCAN 控制特性而定义，比如比特率、Rx 位中的可编程采样点、回环模式、仅监听模式、总线关闭恢复行为和中断使能（总线关闭、错误、警告）。同时也决定着时钟预分频器的分频系数。

CAN 位时序变量 (PRES DIV、PROPSEG、PSEG1、PSEG2 和 RJW) 也可在 CAN_CBT 寄存器中配置，这样可以扩展所有这些变量的范围。如果 CAN_CBT[BTF]置位，CAN_CTRL1 的 PRES DIV、PROPSEG、PSEG1、PSEG2 和 RJW 将变为只读。

此寄存器的内容不受软复位的影响。

注

CAN_CTRL1 和 CAN_CBT 中的 CAN 位变量存储在相同的寄存器中。

地址: 基址 基准 + 4h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PRESDIV								RJW	PSEG1			PSEG2			
W	PRESDIV								RJW	PSEG1			PSEG2			
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BOFFMSK	ERRMSK	CLKSRC	LPB	TWRNMSK	RWRNMSK	保留	保留	SMP	BOFFREC	TSYN	LBUF	LOM	PROPSEG		
W	BOFFMSK	ERRMSK	CLKSRC	LPB	TWRNMSK	RWRNMSK	保留	保留	SMP	BOFFREC	TSYN	LBUF	LOM	PROPSEG		
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_CTRL1 字段描述

字段	描述
31-24 PRESDIV	<p>预分频器分频系数</p> <p>此 8 位字段定义 PE 时钟频率与串行时钟(Sclock)频率之比。Sclock 周期定义 CAN 协议的时间量子。对于复位值，Sclock 频率等于 PE 时钟频率。此字段的最大值为 0xFF，结果得到的最小 Sclock 频率等于 PE 时钟频率/256。见“协议时序”一节。此字段仅在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>Sclock 频率 = PE 时钟频率 时钟频率 / (PRESDIV + 1)</p>
23-22 RJW	<p>再同步跳转宽度</p> <p>此 2 位字段定义在一位时间中一次再同步可以改变的最大时间量子数。一个时间量子等于 Sclock 周期。有效可编程值为 0-3。此字段仅在冻结模式下写入，因为在其他模式下它被硬件禁用。</p>

下一页继续介绍此表...

CANx_CTRL1 字段描述 (继续)

字段	描述
	再同步跳转宽度 = RJW + 1。
21-19 PSEG1	<p>相位段 1</p> <p>此 3 位字段以位时间为单位定义相位段 1 的长度。有效可编程值为 0-7。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>相位缓冲器段 2 = (PSEG1 + 1) × 时间量子。</p>
18-16 PSEG2	<p>相位段 2</p> <p>此 3 位字段以位时间为单位定义相位段 2 的长度。有效可编程值为 1-7。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>相位缓冲器段 2 = (PSEG2 + 1) × 时间量子。</p>
15 BOFFMSK	<p>总线关闭中断掩码</p> <p>此位为 CAN_ESR1 寄存器中的总线关闭中断 BOFFINT 提供掩码。</p> <p>0 总线关闭中断已禁用。 1 总线关闭中断已使能。</p>
14 ERRMSK	<p>错误中断掩码</p> <p>此位为 CAN_ESR1 寄存器中的错误中断 ERRINT 提供掩码。</p> <p>0 错误中断已禁用。 1 错误中断已使能。</p>
13 CLKSRC	<p>CAN 引擎时钟源</p> <p>此位选择 CAN 协议引擎(PE)的时钟源，要么为外设时钟，要么为振荡器时钟。所选时钟为馈入预分频器的时钟，用于生成串行时钟(Sclock)。为了确保工作的可靠性，此位仅可在禁用模式下写入，因为在其他模式下它被硬件禁用。见协议时序。</p> <p>0 CAN 引擎时钟源是振荡器时钟。在此条件下，振荡器时钟频率必须低于总线时钟。 1 CAN 引擎时钟源是外设时钟。</p>
12 LPB	<p>回环模式</p> <p>此位将 FlexCAN 配置为在回环模式下工作。在此模式下，FlexCAN 执行可用于自测操作的内部回环。发送器的位流输出在内部反馈到接收器输入端。Rx CAN 输入引脚被忽略，且 Tx CAN 输出进入隐性状态（逻辑 1）。FlexCAN 如同发送时一样正常工作，并将其自己发送的报文视为从远程节点接收到的报文。</p> <p>在该模式下，FlexCAN 会忽略 CAN 帧确认字段中 ACK 槽期间传送的位，生成一个内部确认位，以确保正确接收到自己的消息。发送中断和接收中断均会生成。此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>注：在此模式下，CAN_MCR[SRXDIS]无法置位，因为这会妨碍已发送消息的自动接收。</p> <p>0 回环已禁用。 1 回环已使能。</p>
11 TWRNMSK	<p>Tx 警告中断掩码</p> <p>此位为与错误和状态寄存器 1 (ESR1)中的 TWRNINT 标志相关联的 Tx 警告中断提供掩码。此位在 CAN_MCR[WRNEN]位清 0 时读为零。此位只有在 CAN_MCR[WRNEN]位置位时才能写入。</p> <p>0 Tx 警告中断已禁用。 1 Tx 警告中断已使能。</p>

下一页继续介绍此表...

CANx_CTRL1 字段描述 (继续)

字段	描述
10 RWRNMSK	<p>Rx 警告中断掩码</p> <p>此位为与错误和状态寄存器 1 (ESR1)中的 RWRNINT 标志相关联的 Rx 警告中断提供掩码。此位在 CAN_MCR[WRNEN]位清 0 时读为零。此位只有在 CAN_MCR[WRNEN]位置位时才能写入。</p> <p>0 Rx 警告中断已禁用。 1 Rx 警告中断已使能。</p>
9 保留	此字段为保留字段。
8 保留	此字段为保留字段。
7 SMP	<p>CAN 位采样</p> <p>此位定义 CAN 位在 Rx 输入端的采样模式。此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>注：为了确保正常工作，要置位 SMP，就要确保 CAN_CTRL1[PSEG1] (或 CAN_CBT[EPSEG1]) 中的最小值为 2 个时间量子。</p> <p>0 只用一个样本来确定此位的值。 1 用三个样本来确定接收位的值：常规样本 (采样点) 和 2 个先前样本；运用了多数决定规则。</p>
6 BOFFREC	<p>总线关闭恢复</p> <p>此位定义 FlexCAN 从总线关闭状态恢复的方式。如果此位清 0，则会根据 CAN 规范 2.0B 的规定，自动从总线关闭状态恢复。如果此位置位，则会禁用从总线关闭状态自动恢复的功能，并且在此位被用户清 0 之前，模块将保持总线关闭状态。如果清 0 发生于在 CAN 总线上检测到 128 个 11 隐性位序列之前，则会发生总线关闭恢复操作，就如 BOFFREC 位从未置位一样。如果清 0 发生于 128 个 11 隐性位序列发生之后，FlexCAN 会先等待 11 个隐性位，然后加入总线，从而与总线重新同步。清 0 后，可以在总线关闭过程中重新置位 BOFFREC 位，但只会在模块下次进入总线关闭状态时生效。如果 BOFFREC 在模块进入总线关闭状态时清 0，则在总线关闭过程中置位它，在当前总线关闭恢复过程中将是无效的。</p> <p>0 自动从总线关闭状态恢复已使能。 1 自动从总线关闭状态恢复已禁用。</p>
5 TSYN	<p>定时器同步</p> <p>此位使能一种机制，该机制可在消息缓冲器 0 中每次收到消息时复位自由运行的定时器。此特性为用一条特殊的“同步”消息 (即全局网络时间) 同步多个 FlexCAN 站点提供了一种手段。如果 CAN_MCR 中的 RFEN 位置位 (Rx FIFO 使能)，则第一个可用的邮箱 (据 CAN_CTRL2[RFFN]设置) 而非 MB0 将用于定时器同步。此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>0 定时器同步特性已禁用 1 定时器同步特性已使能</p>
4 LBUF	<p>先发送最低缓冲器</p> <p>此位定义消息缓冲器传输的排序机制。置位时，CAN_MCR[LPRIOEN]位不会影响优先级仲裁。此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>0 先发送优先级最高的缓冲器。 1 先发送最低编号缓冲器</p>
3 LOM	<p>仅监听模式</p> <p>此位将 FlexCAN 配置为在仅监听模式下工作。在此模式下，传输被禁用，CAN_ECR 寄存器中描述的所有错误计数器都被冻结，并且模块在 CAN 错误无效模式下工作。只会接收另一 CAN 站点确认的消息。如果</p>

下一页继续介绍此表...

CANx_CTRL1 字段描述 (继续)

字段	描述
	<p>FlexCAN 检测到尚未被确认的消息，则会标志 BIT0 错误，而不改变 CAN_ECR 寄存器中的接收错误计数器 (RXERRCNT)，就如它要尝试确认消息一样。</p> <p>仅监听模式由 CAN_ESR1[FLTCONF] 字段指示无效错误的状态确认。允许仅监听模式请求与确认之间存在一定的延迟。</p> <p>此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>0 仅监听模式已被停用。 1 FlexCAN 模块工作于仅监听模式。</p>
PROPSEG	<p>传播段</p> <p>此 3 位字段以位时间为单位定义传播段的长度。有效可编程值为 0–7。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>传播段时间 = (PROPSEG + 1) × 时间量子。 时间量子 = 一个 Sclock 周期。</p>

42.4.4 自由运行定时器 (CANx_TIMER)

此寄存器代表一个可由 CPU 进行读写的 16 位自由运行计数器。此定时器在复位后从 0x0 开始，线性计数至 0xFFFF，并如此循环下去。

定时器将由 CAN 位时钟递增，它可定义 CAN 总线上的波特率。在消息传送/接收过程中，接收或传送每个位时，它将会递增 1。当总线上无消息时，它将使用先前设定的波特率计数。在禁用、休眠、停止和冻结模式下，定时器不会递增。

定时器值将在任意帧的标识符字段的第二位处于 CAN 总线上时捕获。在成功接收或发送一条消息之后，该捕获值将写入消息缓冲器的时间戳条目。

如果 CAN_CTRL1[TSYN] 位置位，则定时器会根据 CAN_CTRL2[RFFN] 的设置，在第一个可用邮箱中每次收到消息时复位。

CPU 可以随时写入此寄存器。然而，如果写操作与第一个邮箱的接收操作导致的定时器复位同时发生，则会丢弃写入值。

读取此寄存器会影响邮箱解锁程序，详见“邮箱锁定机制”一节。

地址: 基址 基准 + 8h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TIMER															
W	0																TIMER															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_TIMER 字段描述

字段	描述
31-16 保留	此只读字段为保留字段且值始终为 0。
TIMER	定时器值 含有自由运行计数器值。

42.4.5 Rx 邮箱全局掩码寄存器 (CANx_RXMGMASK)

此寄存器位于 RAM 中。

RXMGMASK 用于支持传统应用。

- 当 CAN_MCR[IRMQ]位清 0 时，RXMGMASK 将始终有效 (MG 字段中的位将屏蔽邮箱滤波器位)。
- 当 CAN_MCR[IRMQ]位置位时，RXMGMASK 不起作用 (MG 字段中的位不会屏蔽邮箱滤波器位)。

RXMGMASK 用于屏蔽所有 Rx MB 的滤波器字段，不包括第 14-15 号 MB，因为它们有自己的掩码寄存器。

此寄存器仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。

地址: 基址 基准 + 10h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																	MG															
W																	MG															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	

* 注:

- x = 复位时未定义。

CANx_RXMGMASK 字段描述

字段	描述						
MG	Rx 邮箱全局掩码位						
这些位屏蔽邮箱滤波器位。注意，与邮箱 ID 字的对齐并不完美，因为两个最高有效位会影响 RTR 和 IDE 字段，而这两个字段位于邮箱的控制和状态字中。下表详细列出了哪些 MG 位负责屏蔽各个邮箱滤波器字段。							
	SMB[RTR]¹	CAN_CTRL2[RRS]	CAN_CTRL2[EACEN]	邮箱滤波器字段			
				MB[RTR]	MB[IDE]	MB[ID]	保留
0	-	-	0	注 ²	注 ³	MG[28:0]	MG[31:29]
0	-	-	1	MG[31]	MG[30]	MG[28:0]	MG[29]

CANx_RXMGMASK 字段描述 (继续)

字段	描述						
	SMB[RTR] ¹	CAN_CTRL2[RRS]	CAN_CTRL2[EACEN]	邮箱滤波器字段			
MB[RTR]				MB[IDE]	MB[ID]	保留	
1	0	-	-	-	-	MG[31:0]	
1	1	0	-	-	MG[28:0]	MG[31:29]	
1	1	1	MG[31]	MG[30]	MG[28:0]	MG[29]	
0 滤波器中的对应位为“无关”。 1 检查滤波器中的对应位。							

1. 输入帧的 RTR 位。将存入称为 Rx 串行消息缓冲器(Rx SMB)的辅助 MB。
2. 如果 CTRL2[EACEN]位清 0，则永远不会将邮箱的 RTR 位与输入帧的 RTR 位进行比较。
3. 如果 CAN_CTRL2[EACEN]位清 0，则会始终将邮箱的 IDE 位与输入帧的 IDE 位进行比较。

42.4.6 Rx 14 掩码寄存器 (CANx_RX14MASK)

此寄存器位于 RAM 中。

RX14MASK 用于支持传统应用。当 CAN_MCR[IRMQ]位置位时，RX14MASK 不起作用。

RX14MASK 用于屏蔽消息缓冲器 14 的滤波器字段。

此寄存器仅可在模块处于冻结模式时进行配置，因为在其他模式下它被硬件禁用。

地址: 基址 基准 + 14h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX14M																															
W	RX14M																															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	

* 注:

- x = 复位时未定义。

CANx_RX14MASK 字段描述

字段	描述
RX14M	<p>Rx 缓冲器 14 掩码位</p> <p>每个掩码位以 RXMGMASK 屏蔽其他邮箱的滤波器相同的方式屏蔽对应的邮箱 14 滤波器字段。参见 CAN_RXMGMASK 寄存器说明。</p> <p>0 滤波器中的对应位为“无关”。</p> <p>1 检查滤波器中的对应位。</p>

42.4.7 Rx 15 掩码寄存器 (CANx_RX15MASK)

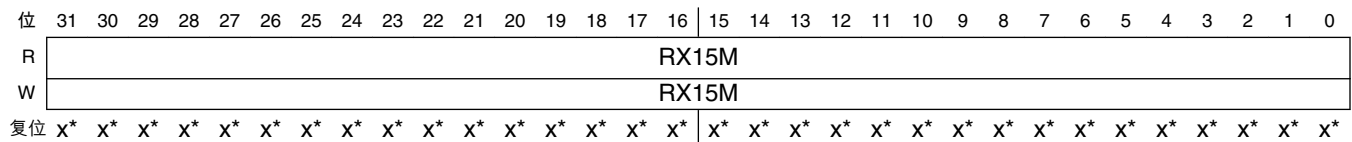
此寄存器位于 RAM 中。

RX15MASK 用于支持传统应用。当 CAN_MCR[IRMQ]位置位时, RX15MASK 不起作用。

RX15MASK 用于屏蔽消息缓冲器 15 的滤波器字段。

此寄存器仅可在模块处于冻结模式时进行配置, 因为在其他模式下它被硬件禁用。

地址: 基址 基准 + 18h 偏移



*注:

- x = 复位时未定义。

CANx_RX15MASK 字段描述

字段	描述
RX15M	<p>Rx 缓冲器 15 掩码位</p> <p>每个掩码位以 RXMGMASK 屏蔽其他邮箱的滤波器相同的方式屏蔽对应的邮箱 15 滤波器字段。参见 CAN_RXMGMASK 寄存器说明。</p> <p>0 滤波器中的对应位为“无关”。</p> <p>1 检查滤波器中的对应位。</p>

42.4.8 错误计数器 (CANx_ECR)

此寄存器有 2 个 8 位字段, 反映了 FlexCAN 错误计数器的值:

- 发送错误计数器 (TXERRCNT 字段)
- 接收错误计数器 (RXERRCNT 字段)

故障限制状态 (错误和状态寄存器 1 - CAN_ESR1 中的 FLTCONF 字段) 基于 TXERRCNT 和 RXERRCNT 计数器更新。TXERRCNT 和 RXERRCNT 计数器只能在冻结模式下写入。这些计数器增减的规则在 CAN 协议中有详细描述, 并且在 FlexCAN 模块中得到了全面实现。

以下是 FlexCAN 总线状态转换的基本规则:

- 如果 TXERRCNT 或 RXERRCNT 的值递增至大于或等于 128，错误和状态寄存器中的 FLTCONF 字段将更新以反映“错误无效”状态。
- 如果 FlexCAN 状态为“错误无效”，TXERRCNT 或 RXERRCNT 递减至小于或等于 127 的值，并且 RXERRCNT 或 TXERRCNT 已经满足此条件，则错误和状态寄存器中的 FLTCONF 字段会更新以反映“错误有效”状态。
- 如果 TXERRCNT 的值递增至大于 255，错误和状态寄存器中的 FLTCONF 字段将更新以反映“总线关闭”状态，并且可能生成一个中断。然后将 TXERRCNT 的值复位为零。
- 如果 FlexCAN 处于“总线关闭”状态，则 TXERRCNT 将与另一个内部计数器级联，以便对总线上 11 连续隐性位的第 128 个实例计数。因此，TXERRCNT 将复位至零并按以下方式计数：内部计数器计数 11 个这种位，然后在递增 TXERRCNT 的同时循环。当 TXERRCNT 达到 128 时，错误和状态寄存器中的 FLTCONF 字段将更新为“错误有效”，两个错误计数器同时复位至零。当显性位的任何实例出现在少于 11 个连续隐性位的流之后时，内部计数器会自行复位至零，而不影响 TXERRCNT 的值。
- 如果在系统启动过程中，只有一个节点在工作，则其 TXERRCNT 会在尝试发送的每条消息中递增，这是确认错误导致的结果（由错误和状态寄存器中的 ACKERR 位指示）。在转换至“错误无效”状态之后，确认错误将不再导致 TXERRCNT 递增。因此，器件永远都不会进入“总线关闭”状态。
- 如果 RXERRCNT 递增至大于 127 的值，则不会再递增，即使在作为接收器的过程中检测到更多错误。在下次成功接收消息时，计数器将复位至 119 和 127 之间的值，以恢复至“错误有效”状态。

地址: 基址 基准 + 1Ch 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0								RXERRCNT				TXERRCNT											
W																	RXERRCNT				TXERRCNT											
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_ECR 字段描述

字段	描述
31-24 保留	此只读字段为保留字段且值始终为 0。
23-16 保留	此只读字段为保留字段且值始终为 0。
15-8 RXERRCNT	接收错误计数器 接收错误计数器，用于在已接收消息中检测到的所有错误。RXERRCNT 计数器为只读，冻结模式下除外，在该模式下，可由 CPU 进行写入操作。
TXERRCNT	发送错误计数器 发送错误计数器，用于在已发送消息中检测到的所有错误。TXERRCNT 计数器为只读，冻结模式下除外，在该模式下，可由 CPU 进行写入操作。

42.4.9 错误和状态 1 寄存器 (CANx_ESR1)

此寄存器报告在接收和发送 CAN 帧时检测到的各种错误条件以及器件的一些一般状态，同时也是部分 CPU 中断的源。

对于在任何格式的 CAN 帧中检测到的错误，报告的错误条件为 BIT1ERR、BIT0ERR、ACKKERR、CRCERR、FRMERR 和 STFERR。

在单个 CAN 帧中检测到的错误可能由一个或多个错误标志报告。另外，在后续帧中发生更多错误事件而 CPU 并未尝试读取此寄存器时，错误报告会进行累加。

TXWRN、RXWRN、IDLE、TX、FLTCONF、RX、SYNCH 为状态位。

BOFFINT、BOFFDONEINT、ERRINT、WAKINT、TWRNINT 和 RWRNINT 为中断位。建议 CPU 在处理这些位生成的中断请求时使用以下程序：

- 读取此寄存器，以捕获所有错误条件和状态位。此操作会清零自上次读取访问以来置位的相应位。
- 写入 1，清零触发了中断请求的中断位。
- 写入 1，清零 ERR_OVR 位（若已置位）。

从已清零的所有错误标志开始，第一个错误事件将置位 ERRINT（条件是相应的掩码位已置位）。如果后续帧中的其他错误事件在 CPU 处理中断请求之前发生，则 ERR_OVR 位会置位，以指示来自不同帧的错误已经累加。

SYNCH	IDLE	TX	RX	FlexCAN 状态
0	0	0	0	未与 CAN 总线同步
1	1	x	x	闲置
1	0	1	0	正在发送
1	0	0	1	正在接收

地址: 基址 基准 + 20h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	保留	保留	0	保留	保留	保留	0			ERROVR	保留	BOFFDONEINT	SYNCH	TWRNINT	RWRNINT	
W											w1c		w1c	SYNCH	w1c	w1c
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BIT1ERR	BIT0ERR	ACKERR	CRCERR	FRMERR	STFERR	TXWRN	RXWRN	IDLE	TX	FLTCONF	RX	BOFFINT	ERRINT	WAKINT	
W	BIT1ERR	BIT0ERR	ACKERR	CRCERR	FRMERR	STFERR	TXWRN	RXWRN	IDLE	TX	FLTCONF	RX	w1c	w1c	w1c	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_ESR1 字段描述

字段	描述
31 保留	此字段为保留字段。
30 保留	此字段为保留字段。
29 保留	此只读字段为保留字段且值始终为 0。
28 保留	此字段为保留字段。
27 保留	此字段为保留字段。
26 保留	此字段为保留字段。
25-22 保留	此只读字段为保留字段且值始终为 0。
21 ERROVR	错误溢出位 此位指示在任何错误标志已置位时发生了一个错误条件。通过向其写入 1 可将其清零。

下一页继续介绍此表...

CANx_ESR1 字段描述 (继续)

字段	描述
	0 溢出未发生。 1 溢出已发生。
20 保留	此字段为保留字段。
19 BOFFDONEINT	总线关闭完成中断 此位在 Tx 错误计数器(TXERRCNT)在 CAN 总线上完成计数 11 连续隐性位的 128 个实例且准备退出总线关闭状态时置位。如果控制 2 寄存器(BOFFDONEMSK)中对应的掩码位置位, 则会对 CPU 生成一个中断。通过向其写入 1 可将此位清零。写入 0 则不起作用。 0 未发生此情况。 1 FlexCAN 模块已完成总线关闭过程。
18 SYNCH	CAN 同步状态 该只读标志指示 FlexCAN 是否与 CAN 总线同步, 并且能参与通信过程。由 FlexCAN 置位和清零。请参阅整体 CAN_ESR1 寄存器描述中的表格。 0 FlexCAN 未与 CAN 总线同步。 1 FlexCAN 已与 CAN 总线同步。
17 TWRNINT	Tx 警告中断标志 如果 CAN_MCR 中的 WRNEN 位置位, 则 TWRNINT 位在 WRNEN 标志从 0 转换为 1 时置位, 即是在 Tx 错误计数器达到 96 时。如果控制 1 寄存器(CAN_CTRL1[TWRNMSK])中对应的掩码位置位, 则会对 CPU 生成一个中断。此位通过写入 1 清零。当 WRNEN 清 0 时, 此标志将被屏蔽。在禁用此位之前, CPU 必须清零此标志。否则, 它会在 WRNEN 再次置位时置位。写入 0 则不起作用。此标志不会在总线关闭状态中生成。此位在冻结模式下不会更新。 0 未发生此情况。 1 Tx 错误计数器从小于 96 转换为大于或等于 96。
16 RWRNINT	Rx 警告中断标志 如果 CAN_MCR 中的 WRNEN 位置位, 则 RWRNINT 位在 RXWRN 标志从 0 转换为 1 时置位, 即是在 Rx 错误计数器达到 96 时。如果控制 1 寄存器(CAN_CTRL1[RWRNMSK])中对应的掩码位置位, 则会对 CPU 生成一个中断。此位通过写入 1 清零。当 WRNEN 清 0 时, 此标志将被屏蔽。在禁用此位之前, CPU 必须清零此标志。否则, 它会在 WRNEN 再次置位时置位。写入 0 则不起作用。此位在冻结模式下不会更新。 0 未发生此情况。 1 Rx 错误计数器从小于 96 转换为大于或等于 96。
15 BIT1ERR	Bit1 错误 此位指示在消息的仲裁或数据阶段中的已发送位与已接收位之间出现了不一致。 注: 在仲裁字段或 ACK 槽的情况下, 或者在一个节点发送无效错误标志却检测到显性位的情况下, 发送器不会置位此位。 0 未发生此情况。 1 至少有一位以隐性位发送, 却以显性位接收。
14 BIT0ERR	Bit0 错误 此位指示在消息的仲裁或数据阶段中的已发送位与已接收位之间出现了不一致。

下一页继续介绍此表...

CANx_ESR1 字段描述 (继续)

字段	描述
	0 未发生此情况。 1 至少有一位以显性位发送，却以隐性位接收。
13 ACKERR	确认错误 此位指示发送器节点检测到了一个确认错误，即是说，在 ACK 槽中检测到了一个显性位。 0 未发生此情况。 1 自上次读取此寄存器以来发生了一个 ACK 错误。
12 CRCERR	循环冗余校验错误 此位指示接收器节点在，即是说，计算得到的 CRC 与接收到的 CRC 不同。 0 未发生此情况。 1 自上次读取此寄存器以来发生了一个 CRC 错误。
11 FRMERR	形式错误 此位指示接收器节点在形式错误，即是说，一个形式固定的位字段含有至少一个非法位。 0 未发生此情况。 1 自上次读取此寄存器以来发生了一个形式错误。
10 STFERR	填充错误 此位指示接收器节点在填充错误。 0 未发生此情况。 1 自上次读取此寄存器以来发生了一个填充错误。
9 TXWRN	Tx 错误警告 此位指示消息发送过程中发生重复性错误，并且只会受 CAN_ECR 寄存器中 TXERRCNT 的值得影响。此位在冻结模式下不会更新。 0 未发生此情况。 1 TXERRCNT 大于或等于 96。
8 RXWRN	Rx 错误警告 此位指示消息接收过程中发生重复性错误，并且只会受 CAN_ECR 寄存器中 RXERRCNT 的值得影响。此位在冻结模式下不会更新。 0 未发生此情况。 1 RXERRCNT 大于或等于 96。
7 IDLE	此位指示 CAN 总线处于 IDLE 状态的时间。请参阅整体 CAN_ESR1 寄存器描述中的表格。 0 未发生此情况。 1 CAN 总线现在为 IDLE 状态。
6 TX	FlexCAN 正在发送 此位指示 FlexCAN 是否正在发送消息。请参阅整体 CAN_ESR1 寄存器描述中的表格。 0 FlexCAN 未在发送消息。 1 FlexCAN 正在发送消息。
5-4 FLTCONF	故障限制状态 此 2 位字段指示 FlexCAN 模块的限制状态。

下一页继续介绍此表...

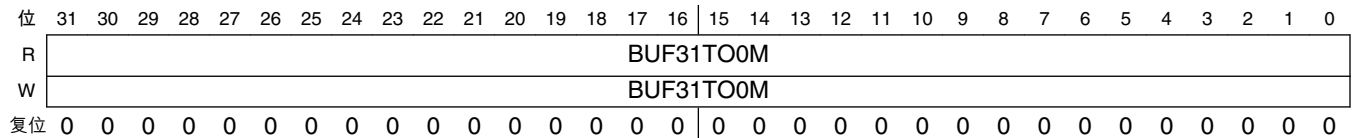
CANx_ESR1 字段描述 (继续)

字段	描述
	<p>如果控制寄存器 1 中的 LOM 位置位,则在取决于 CAN 位时序的一定延迟之后,FLTCONF 字段将指示“错误无效”。相同的延迟会影响 FLTCONF 反映 CPU 对 CAN_ECR 寄存器进行更新的方式。可能需要最多一个 CAN 位时间才能使它们再次一致起来。</p> <p>此位字段会受软复位的影响,但是,如果 LOM 位置位,则其复位值只会持续一个 CAN 位。在该时间之后,FLTCONF 会报告“错误无效”。</p> <p>00 错误有效 01 错误无效 1x 总线关闭</p>
3 RX	<p>FlexCAN 正在接收</p> <p>此位指示 FlexCAN 是否正在接收消息。请参阅整体 CAN_ESR1 寄存器描述中的表格。</p> <p>0 FlexCAN 未在接收消息。 1 FlexCAN 正在接收消息。</p>
2 BOFFINT	<p>总线关闭中断</p> <p>此位在 FlexCAN 进入“总线关闭”状态时置位。如果控制寄存器 1 (CAN_CTRL1[BOFFMSK])中对应的掩码位置位,则会对 CPU 生成一个中断。通过向其写入 1 可将此位清零。写入 0 则不起作用。</p> <p>0 未发生此情况。 1 FlexCAN 模块已进入总线关闭状态。</p>
1 ERRINT	<p>错误中断</p> <p>此位指示错误位中至少有一个 (BIT1ERR、BIT0ERR、ACKERR、CRCERR、FRMERR 或 STFERR) 置位。如果对应的掩码位 CAN_CTRL1[ERRMSK]置位 则会对 CPU 生成一个中断。通过向其写入 1 可将此位清零。写入 0 则不起作用。</p> <p>0 未发生此情况。 1 指示错误和状态寄存器中的任何错误位置位。</p>
0 WAKINT	<p>唤醒中断</p> <p>在自我唤醒机制下,此字段在 FlexCAN 处于低功耗模式时适用:</p> <ul style="list-style-type: none"> • 休眠模式 • 停止模式 <p>当在 CAN 总线上检测到隐性至显性转换并且 CAN_MCR[WAKMSK]位置位时,会对 CPU 生成一个中断。通过向其写入 1 可将此位清零。</p> <p>当 CAN_MCR[SLFWAK]清 0 时,此标志将被屏蔽。在禁用此位之前,CPU 必须清零此标志。否则,它会在 SLFWAK 再次置位时置位。写入 0 则不起作用。</p> <p>0 未发生此情况。 1 指示在 CAN 总线上收到了隐性至显性转换。</p>

42.4.10 中断掩码 1 寄存器 (CANx_IMASK1)

此寄存器允许，使能或禁用 MB15 至 MB0 范围内 16 个消息缓冲器中断中的任意一个。其中每个缓冲器有一个中断掩码位，使 CPU 可以确定在成功发送或接收之后，即在对应的 CAN_IFLAG1 位置位时，由哪个缓冲器生成中断。

地址: 基址 基准 + 28h 偏移



CANx_IMASK1 字段描述

字段	描述
BUF31TO0M	缓冲器 MB _i 掩码 每位使能或禁用 MB31 至 MB0 范围内对应的 FlexCAN 消息缓冲器中断。 注: 如果对应的 IFLAG1 位置位,置位或清零 CAN_IMASK1 寄存器中的一位可以置位或清 0 一个中断请求。 0 对应缓冲器中断已禁用。 1 对应缓冲器中断已使能。

42.4.11 中断标志 1 寄存器 (CANx_IFLAG1)

此寄存器定义，以及 MB15 至 MB0 范围内的 16 个消息缓冲器中断的标志。其中每个缓冲器都有一个中断标志位。每次发送或接收成功都会置位对应的 CAN_IFLAG1 位。如果对应的 CAN_IMASK1 位置位，则会生成一个中断。中断标志必须通过向其写入 1 来清零。写入 0 则不起作用。如下所述，当使能用于 Rx FIFO 的 DMA 时，存在例外。

当 Rx FIFO 使能时，BUF7I 至 BUF5I 标志也用于表示 FIFO 中断。当 CAN_MCR[RFEN]位置位且 CAN_MCR[DMA]位清零时，8 个最低有效中断标志的功能会发生变化：BUF7I、BUF6I 和 BUF5I 指示 FIFO，BUF0I 用于清空 FIFO，BUF4I 至 BUF1I 位保留。

在使能 CAN_MCR[RFEN]之前，CPU 必须处理在 Rx FIFO 区域中置位的 IFLAG 位；见"Rx FIFO"一节。否则，这些 IFLAG 位将错误地展示现在属于 FIFO 的相关 MB，就如它们有内容要处理一样。当 CAN_MCR[RFEN]位清 0 时，FIFO 标志必须清零。当选择 CAN_CTRL2[RFFN]值以将 Rx FIFO 滤波器扩展至 MB7 以外时，必须同样小心。例如，当 RFFN 为 0x8 时，MB0-23 范围被 Rx FIFO 滤波器占据，必须清零相关的 IFLAG 位。

当 CAN_MCR[RFEN]和 CAN_MCR[DMA]位都置位时 (用于 Rx FIFO 的 DMA 特性被使能), 8 个最低有效中断标志(BUF7I - BUF0I)的功能将发生变化以支持 DMA 操作。不使用 BUF7I 和 BUF6I 以及 BUF4I 至 BUF1I。BUF5I 指示 FIFO 的工作条件, BUF0I 用于清空 FIFO。另外, BUF5I 不会生成 CPU 中断, 而会生成 DMA 请求。当 CAN_MCR[DMA]位使能时, 不会考虑 Rx FIFO 区域中的 IMASK1 位。另外, 当 DMA 使能时, CPU 不得清零标志 BUF5I。在使能 CAN_MCR[DMA]位之前, CPU 必须处理在 Rx FIFO 区域中置位的 IFLAG。当 CAN_MCR[DMA]位清 0 时, FIFO 必须为空。

在更新 CAN_MCR[MAXMB]字段之前, CPU 必须处理其 MB 值大于要更新的 CAN_MCR[MAXMB]的 CAN_IFLAG1 位; 否则, 它们将保持置位并与可用 MB 的数量不一致。

地址: 基址 基准 + 30h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF31TO8I															
W	w1c															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF31TO8I								BUF7I	BUF6I	BUF5I	BUF4TO1I				BUF0I
W	w1c								w1c	w1c	w1c	w1c				w1c
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_IFLAG1 字段描述

字段	描述
31-8 BUF31TO8I	<p>缓冲器 MB_i 中断</p> <p>每位标志 MB15 至 MB8 范围内对应的 FlexCAN 消息缓冲器中断。</p> <p>0 对应的缓冲器无成功完成的发送或接收实例。</p> <p>1 对应的缓冲器已成功完成发送或接收。</p>
7 BUF7I	<p>缓冲器 MB7 中断或“Rx FIFO 溢出”</p> <p>当 CAN_MCR 寄存器中的 RFEN 位清零时 (Rx FIFO 被禁用), 此位将标记 MB7 的中断。</p> <p>注: 每当 CAN_MCR[RFEN]位被 CPU 写操作更改时, FlexCAN 都会清零此标志。</p> <p>当 CAN_MCR[RFEN]置位时, BUF7I 标志表示“Rx FIFO 溢出”。在这种情况下, 此标志指示有一条消息因 Rx FIFO 已满而丢失。注意, 当 Rx FIFO 已满且消息被邮箱捕捉到时, 此标志不会置位。</p>

下一页继续介绍此表...

CANx_IFLAG1 字段描述 (继续)

字段	描述
	<p>0 当 MCR[RFEN]=0 时, 无 MB7 完成发送/接收的实例; 当 MCR[RFEN]=1 时, 无 Rx FIFO 溢出实例</p> <p>1 当 MCR[RFEN]=0 时, MB7 已完成发送/接收; 当 MCR[RFEN]=1 时, Rx FIFO 溢出</p>
6 BUF6I	<p>缓冲器 MB6 中断或“Rx FIFO 警告”</p> <p>当 CAN_MCR 寄存器中的 RFEN 位清零时 (Rx FIFO 被禁用), 此位将标记 MB6 的中断。</p> <p>注: 每当 CAN_MCR[RFEN]位被 CPU 写操作更改时, FlexCAN 都会清零此标志。</p> <p>当 CAN_MCR[RFEN]置位时, BUF6I 标志表示“Rx FIFO 警告”。这种情况下, 此标志指示 Rx FIFO 中未读取的消息数量因收到新消息而从 4 增至 5 的时候, 这表示 Rx FIFO 几乎已满。注意, 如果此标志在未读取消息数量大于 4 时清零, 则在 Rx FIFO 中未读取消息的数量减少至等于或小于 4 之前, 该标志不会再次置位。</p> <p>0 当 MCR[RFEN]=0 时, 无 MB6 完成发送/接收的实例; 当 MCR[RFEN]=1 时, 无 Rx FIFO 几乎已满实例</p> <p>1 当 MCR[RFEN]=0 时, MB6 已完成发送/接收; 当 MCR[RFEN]=1 时, Rx FIFO 几乎已满</p>
5 BUF5I	<p>缓冲器 MB5 中断或“Rx FIFO 中的可用帧”</p> <p>当 MCR 中的 RFEN 位清零时 (Rx FIFO 被禁用), 此位将标记 MB5 的中断。</p> <p>注: 每当 MCR[RFEN]位被 CPU 写操作更改时, FlexCAN 都会清零此标志。</p> <p>当 MCR[RFEN]置位时 (Rx FIFO 已使能), BUF5I 标志表示“Rx FIFO 中的可用帧”并指示至少可从 Rx FIFO 读取一个帧。当 MCR[DMA]位使能时, 此标志会生成一个 DMA 请求, CPU 必须通过 BUF5I 中写入 1 来清零此位。</p> <p>0 当 MCR[RFEN]=0 时, 无 MB5 完成发送/接收的实例; 当 MCR[RFEN]=1 时, FIFO 中无可用帧实例</p> <p>1 当 MCR[RFEN]=0 时, MB5 已完成发送/接收; 当 MCR[RFEN]=1 时, Rx FIFO 中有可用帧。当 MCR[RFEN]和 MCR[DMA]被使能时, 它会生成一个 DMA 请求。</p>
4-1 BUF4TO1I	<p>缓冲器 MB_i 中断或“保留”</p> <p>当 CAN_MCR 寄存器中的 RFEN 位清零时 (Rx FIFO 被禁用), 这些位将标记 MB4 至 MB1 的中断。</p> <p>注: 每当 CAN_MCR[RFEN]位被 CPU 写操作更改时, FlexCAN 都会清零这些标志。</p> <p>当 CAN_MCR[RFEN]置位时, BUF4TO1I 标志将保留。</p> <p>0 当 MCR[RFEN]=0 时, 对应的缓冲器无成功完成的发送或接收实例。</p> <p>1 当 MCR[RFEN]=0 时, 对应的缓冲器已经成功完成发送或接收。</p>
0 BUF0I	<p>缓冲器 MB0 中断或清零 FIFO 位</p> <p>当 MCR 中的 RFEN 位清零时 (Rx FIFO 被禁用), 此位将标记 MB0 的中断。如果 Rx FIFO 被使能, 此位用于触发 FIFO 清零操作。该操作会清空 FIFO 的内容。在执行该操作之前, CPU 必须处理所有 FIFO 相关 IFLAG。当 MCR[DMA]位使能时, 该操作也会清零 BUF5I 标志, 并且结果会中止 DMA 请求。FIFO 清零操作发生在 CPU 在 BUF0I 中写入 1 的时候。只有在冻结模式下才能这样做, 在其他条件下被硬件禁用。</p> <p>0 当 MCR[RFEN]=0 时, 对应的缓冲器无成功完成的发送或接收实例。</p> <p>1 当 MCR[RFEN]=0 时, 对应的缓冲器已经成功完成发送或接收。</p>

42.4.12 控制 2 寄存器 (CANx_CTRL2)

此寄存器为控制 1 寄存器的补充, 为冻结模式下的存储器写访问提供控制位, 用于扩展 FIFO 滤波器量, 以及调整内部 FlexCAN 进程 (如匹配和仲裁) 的工作。

此寄存器的内容不受软复位的影响。

地址: 基址 基准 + 34h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	BOFFDONEMSK	0	0	RFFN				TASD				MRP	RRS	EACEN	
W		BOFFDONEMSK			RFFN				TASD				MRP	RRS	EACEN	
复位	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	保留	0	0	0	0	0										
W	保留															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_CTRL2 字段描述

字段	描述
31 保留	此只读字段为保留字段且值始终为 0。
30 BOFFDONEMSK	总线关闭完成中断掩码 此位为 CAN_ESR1 寄存器中的总线关闭完成中断提供掩码。 0 总线关闭完成中断已禁用。 1 总线关闭完成中断已使能。
29 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

CANx_CTRL2 字段描述 (继续)

字段	描述																																																																																																						
28 保留	此只读字段为保留字段且值始终为 0。																																																																																																						
27-24 RFFN	<p>Rx FIFO 滤波器数</p> <p>此 4 位字段定义 Rx FIFO 滤波器的数量,如下表所示。最大可选滤波器数由芯片决定。此字段仅可在冻结模式下写入,因为在其他模式下它被硬件禁用。此字段的配置值不得使 Rx FIFO 和 ID 滤波器占据的消息缓冲器的数量超过现有邮箱的数量(由 CAN_MCR[MAXMB]定义)。</p> <p>注: 由 8 个滤波器组成的每一组都占据着相当于两个消息缓冲器的存储器空间,即是说,实现的滤波器越多,可用的邮箱就越少。</p> <p>考虑到 Rx FIFO 占据着最初为 MB0-5 保留的存储器空间,所以,RFFN 的配置值应对应于不大于可用存储器字数的滤波器数;可用存储器字数计算公式如下:</p> $(SETUP_MB - 6) \times 4$ <p>其中, SETUP_MB 为参数 NUMBER_OF_MB 和 CAN_MCR[MAXMB]之间的最小值。</p> <p>剩余的可用邮箱数为:</p> $(SETUP_MB - 8) - (RFFN \times 2)$ <p>如果通过 RFFN 配置的 Rx FIFO 滤波器数超过 SETUP_MB 的值(可用存储器空间),则多余滤波器将不起作用。</p> <p>注:</p> <ul style="list-style-type: none"> 最后剩余可用邮箱数由(NUMBER_OF_MB-1)与 CAN_MCR[MAXMB]字段之间的最小值定义。 如果 Rx 单独掩码寄存器未使能,则所有 Rx FIFO 滤波器都会受 Rx FIFO 全局掩码影响。 <table border="1"> <thead> <tr> <th>RFFN[3:0]</th> <th>Rx FIFO 滤波器元素数</th> <th>Rx FIFO 和 ID 滤波器表占用的消息缓冲器</th> <th>剩余的可用邮箱</th> <th>受 Rx 单独掩码影响的 Rx FIFO ID 滤波器表元素</th> <th>受 Rx FIFO 全局掩码影响的 Rx FIFO ID 滤波器表元素</th> </tr> </thead> <tbody> <tr><td>0x0</td><td>8</td><td>MB 0-7</td><td>MB 8-63</td><td>元素 0-7</td><td>无</td></tr> <tr><td>0x1</td><td>16</td><td>MB 0-9</td><td>MB 10-63</td><td>元素 0-9</td><td>元素 10-15</td></tr> <tr><td>0x2</td><td>24</td><td>MB 0-11</td><td>MB 12-63</td><td>元素 0-11</td><td>元素 12-23</td></tr> <tr><td>0x3</td><td>32</td><td>MB 0-13</td><td>MB 14-63</td><td>元素 0-13</td><td>元素 14-31</td></tr> <tr><td>0x4</td><td>40</td><td>MB 0-15</td><td>MB 16-63</td><td>元素 0-15</td><td>元素 16-39</td></tr> <tr><td>0x5</td><td>48</td><td>MB 0-17</td><td>MB 18-63</td><td>元素 0-17</td><td>元素 18-47</td></tr> <tr><td>0x6</td><td>56</td><td>MB 0-19</td><td>MB 20-63</td><td>元素 0-19</td><td>元素 20-55</td></tr> <tr><td>0x7</td><td>64</td><td>MB 0-21</td><td>MB 22-63</td><td>元素 0-21</td><td>元素 22-63</td></tr> <tr><td>0x8</td><td>72</td><td>MB 0-23</td><td>MB 24-63</td><td>元素 0-23</td><td>元素 24-71</td></tr> <tr><td>0x9</td><td>80</td><td>MB 0-25</td><td>MB 26-63</td><td>元素 0-25</td><td>元素 26-79</td></tr> <tr><td>0xA</td><td>88</td><td>MB 0-27</td><td>MB 28-63</td><td>元素 0-27</td><td>元素 28-87</td></tr> <tr><td>0xB</td><td>96</td><td>MB 0-29</td><td>MB 30-63</td><td>元素 0-29</td><td>元素 30-95</td></tr> <tr><td>0xC</td><td>104</td><td>MB 0-31</td><td>MB 32-63</td><td>元素 0-31</td><td>元素 32-103</td></tr> <tr><td>0xD</td><td>112</td><td>MB 0-33</td><td>MB 34-63</td><td>元素 0-31</td><td>元素 32-111</td></tr> <tr><td>0xE</td><td>120</td><td>MB 0-35</td><td>MB 36-63</td><td>元素 0-31</td><td>元素 32-119</td></tr> <tr><td>0xF</td><td>128</td><td>MB 0-37</td><td>MB 38-63</td><td>元素 0-31</td><td>元素 32-127</td></tr> </tbody> </table>	RFFN[3:0]	Rx FIFO 滤波器元素数	Rx FIFO 和 ID 滤波器表占用的消息缓冲器	剩余的可用邮箱	受 Rx 单独掩码影响的 Rx FIFO ID 滤波器表元素	受 Rx FIFO 全局掩码影响的 Rx FIFO ID 滤波器表元素	0x0	8	MB 0-7	MB 8-63	元素 0-7	无	0x1	16	MB 0-9	MB 10-63	元素 0-9	元素 10-15	0x2	24	MB 0-11	MB 12-63	元素 0-11	元素 12-23	0x3	32	MB 0-13	MB 14-63	元素 0-13	元素 14-31	0x4	40	MB 0-15	MB 16-63	元素 0-15	元素 16-39	0x5	48	MB 0-17	MB 18-63	元素 0-17	元素 18-47	0x6	56	MB 0-19	MB 20-63	元素 0-19	元素 20-55	0x7	64	MB 0-21	MB 22-63	元素 0-21	元素 22-63	0x8	72	MB 0-23	MB 24-63	元素 0-23	元素 24-71	0x9	80	MB 0-25	MB 26-63	元素 0-25	元素 26-79	0xA	88	MB 0-27	MB 28-63	元素 0-27	元素 28-87	0xB	96	MB 0-29	MB 30-63	元素 0-29	元素 30-95	0xC	104	MB 0-31	MB 32-63	元素 0-31	元素 32-103	0xD	112	MB 0-33	MB 34-63	元素 0-31	元素 32-111	0xE	120	MB 0-35	MB 36-63	元素 0-31	元素 32-119	0xF	128	MB 0-37	MB 38-63	元素 0-31	元素 32-127
RFFN[3:0]	Rx FIFO 滤波器元素数	Rx FIFO 和 ID 滤波器表占用的消息缓冲器	剩余的可用邮箱	受 Rx 单独掩码影响的 Rx FIFO ID 滤波器表元素	受 Rx FIFO 全局掩码影响的 Rx FIFO ID 滤波器表元素																																																																																																		
0x0	8	MB 0-7	MB 8-63	元素 0-7	无																																																																																																		
0x1	16	MB 0-9	MB 10-63	元素 0-9	元素 10-15																																																																																																		
0x2	24	MB 0-11	MB 12-63	元素 0-11	元素 12-23																																																																																																		
0x3	32	MB 0-13	MB 14-63	元素 0-13	元素 14-31																																																																																																		
0x4	40	MB 0-15	MB 16-63	元素 0-15	元素 16-39																																																																																																		
0x5	48	MB 0-17	MB 18-63	元素 0-17	元素 18-47																																																																																																		
0x6	56	MB 0-19	MB 20-63	元素 0-19	元素 20-55																																																																																																		
0x7	64	MB 0-21	MB 22-63	元素 0-21	元素 22-63																																																																																																		
0x8	72	MB 0-23	MB 24-63	元素 0-23	元素 24-71																																																																																																		
0x9	80	MB 0-25	MB 26-63	元素 0-25	元素 26-79																																																																																																		
0xA	88	MB 0-27	MB 28-63	元素 0-27	元素 28-87																																																																																																		
0xB	96	MB 0-29	MB 30-63	元素 0-29	元素 30-95																																																																																																		
0xC	104	MB 0-31	MB 32-63	元素 0-31	元素 32-103																																																																																																		
0xD	112	MB 0-33	MB 34-63	元素 0-31	元素 32-111																																																																																																		
0xE	120	MB 0-35	MB 36-63	元素 0-31	元素 32-119																																																																																																		
0xF	128	MB 0-37	MB 38-63	元素 0-31	元素 32-127																																																																																																		
23-19 TASD	Tx 仲裁开始延迟																																																																																																						

下一页继续介绍此表...

CANx_CTRL2 字段描述 (继续)

字段	描述
	此 5 位字段指示 Tx 仲裁进程起点从 CAN 总线上 CRC 字段的第一位开始可以延迟多少 CAN 位。更多详情，请参见 Tx 仲裁开始延迟 。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。
18 MRP	<p>邮箱接收优先级</p> <p>如果此位置位，则匹配进程将从邮箱开始，如果无匹配，则匹配将在 Rx FIFO 上继续执行。此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>0 匹配从 Rx FIFO 开始并在邮箱上继续执行。 1 匹配从邮箱开始并在 Rx FIFO 上继续执行。</p>
17 RRS	<p>远程请求存储</p> <p>如果此位置位，则远程请求帧将被提交给一个匹配进程，并以与数据帧相同的方式存储在对应的消息缓冲器中。不会生成自动远程响应帧。</p> <p>如果此位置清 0，则远程请求帧将被提交给一个匹配进程，并且如果发现 ID 相同的消息缓冲器 (CODE=0b1010)，则会生成一个自动远程响应帧。</p> <p>此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>0 远程响应帧已生成。 1 远程请求帧已存储。</p>
16 EACEN	<p>用于 Rx 邮箱的完整帧仲裁字段比较使能</p> <p>此位控制匹配进程对 Rx 邮箱滤波器中的 IDE 和 RTR 位与它们在输入帧中的对应位的比较。此位不会影响 Rx FIFO 的匹配。此位仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>0 始终比较 Rx 邮箱滤波器的 IDE 位；尽管有掩码位，但从不比较 RTR。 1 使能 Rx 邮箱滤波器的 IDE 和 RTR 位与它们在输入帧中的对应位的比较。掩码位确实适用。</p>
15 保留	在写入此字段时，会始终写入复位值。
14 保留	此只读字段为保留字段且值始终为 0。
13 保留	此只读字段为保留字段且值始终为 0。
12 保留	此只读字段为保留字段且值始终为 0。
11 保留	此只读字段为保留字段且值始终为 0。
保留	此只读字段为保留字段且值始终为 0。

42.4.13 错误和状态 2 寄存器 (CANx_ESR2)

此寄存器报告一些通用的状态信息。

地址: 基址 基准 + 38h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								LPTM							
W									LPTM							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	VPS	IMB	0												
W		VPS	IMB													
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_ESR2 字段描述

字段	描述
31-23 保留	此只读字段为保留字段且值始终为 0。
22-16 LPTM	<p>最低优先级 Tx 邮箱</p> <p>如果 CAN_ESR2[VPS]置位, 则此字段指示最小编号的无效邮箱 (见 CAN_ESR2[IMB]位说明)。 如果不存在无效邮箱, 则指示的邮箱取决于 CAN_CTRL1[LBUF]位的值。 如果 CAN_CTRL1[LBUF]位清 0, 则指示的邮箱是拥有最大仲裁值的邮箱 (见“最高优先级邮箱优先”一节)。 如果 CAN_CTRL1[LBUF]位置位, 则指示的邮箱为最大编号的有效 Tx 邮箱。 如果 Tx 邮箱正在被发送, 则在 LPTM 计算中不会考虑它。 如果 CAN_ESR2[IMB]未置位且一个帧发送成功, 则 LPTM 将以其邮箱编号更新。</p>
15 保留	此只读字段为保留字段且值始终为 0。
14 VPS	<p>有效优先级状态</p> <p>此位指示 CAN_ESR2[IMB]和 CAN_ESR2[LPTM]的内容当前是否有效。 它会在 Tx 仲裁过程每次完成时置位, CPU 写入已经在 Tx 仲裁过程中被扫描的邮箱的控制和状态字时除外, 即是说, 它位于 Tx 仲裁指针之后。 如果没有无效邮箱且只有一个 Tx 邮箱正在被发送, 则 VPS 不会置位。 在每个 Tx 仲裁过程开始时, 或者在对任何邮箱的控制和状态字进行写操作时, 此位将清 0。</p> <p>注: CPU 对被中止机制禁用的 MB 的控制状态(C/S)进行的任何写操作都不会影响 CAN_ESR2[VPS]。 当 CAN_MCR[AEN]置位时, 对正在被发送 (挂起中止) 的 MB 的 C/S 进行的中止代码写操作, 或者在 CAN_IFLAG 置位条件下对 Tx MB 尝试进行的任何写操作都会被阻止。</p> <p>0 IMB 和 LPTM 的内容无效。 1 IMB 和 LPTM 的内容有效。</p>
13 IMB	<p>无效邮箱</p> <p>如果 ESR2[VPS]置位, 此位指示是否存在任何无效邮箱 (CODE 字段为 0b1000 或 0b0000)。 此位在下列情况下置位:</p> <ul style="list-style-type: none"> 仲裁过程中, 如果发现一个 CAN_ESR2[LPTM]且其无效。 如果 CAN_ESR2[IMB]未置位且一个帧发送成功。 <p>此位在所有仲裁的开始清零 (见“仲裁过程”一节)。</p>

下一页继续介绍此表...

CANx_ESR2 字段描述 (继续)

字段	描述
	注: CAN_ESR2[LPTM]机制拥有以下行为: 如果 MB 发送成功且 CAN_ESR2[IMB]=0 (无无效邮箱), 则 CAN_ESR2[VPS]和 CAN_ESR2[IMB]将置位且会把与刚发送的 MB 相关的索引载入 CAN_ESR2[LPTM]。 0 如果 ESR2[VPS]置位, 则 ESR2[LPTM]不是无效邮箱。 1 如果 ESR2[VPS]置位, 则至少有一个无效邮箱。LPTM 的内容是第一个邮箱的编号。
保留	此只读字段为保留字段且值始终为 0。

42.4.14 CRC 寄存器 (CANx_CRCR)

此寄存器为非 FD 消息提供关于已发送消息。此寄存器在 Tx 中断标志置位的同时更新。

地址: 基址 基准 + 44h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								MBCRC							
W	MBCRC															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TXCRC														
W	TXCRC															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_CRCR 字段描述

字段	描述
31-23 保留	此只读字段为保留字段且值始终为 0。
22-16 MBCRC	CRC 邮箱 此字段指示与 CAN_CRCR[TXCRC]字段中的值对应的邮箱的编号。
15 保留	此只读字段为保留字段且值始终为 0。
TXCRC	已发送 CRC 值 此字段指示针对非 FD 帧的最后发送的消息中报告。

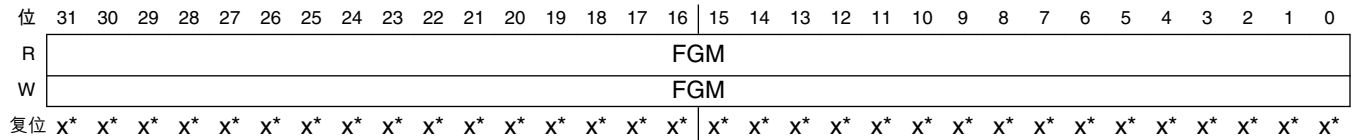
42.4.15 Rx FIFO 全局掩码寄存器 (CANx_RXFGMASK)

此寄存器位于 RAM 中。

如果 Rx FIFO 使能，则 RXFGMASK 用于屏蔽特定的 RXIMR Rx FIFO ID 滤波器表元素，根据 CAN_CTRL2[RFFN]字段的设置，这些元素无对应的 RXIMR。

此寄存器仅在冻结模式下写入，因为在其他模式下它被硬件禁用。

地址: 基址 基准 + 48h 偏移



* 注:

- x = 复位时未定义。

CANx_RXFGMASK 字段描述

字段	描述					
FGM	Rx FIFO 全局掩码位					
	这些位屏蔽 ID 滤波器表元素位，并且完美对齐。					
	下表列出了 FGM 位与各个 IDAF 字段的对应关系。					
	Rx FIFO ID 滤波器表元素格式 (CAN_MCR[IDAM])	标识符接受滤波器字段				
		RTR	IDE	RXIDA	RXIDB ¹	RXIDC ²
A	FGM[31]	FGM[30]	FGM[29:1]	-	-	FGM[0]
B	FGM[31], FGM[15]	FGM[30], FGM[14]	-	FGM[29:16], FGM[13:0]	-	-
C	-	-	-	-	FGM[31:24], FGM[23:16], FGM[15:8], FGM[7:0]	-
0 滤波器中的对应位为“无关”。 1 检查滤波器中的对应位。						

- 如果 CAN_MCR[IDAM]字段等效于格式 B，则只会将输入帧标识符的 14 个最高有效位与 Rx FIFO 滤波器进行比较。
- 如果 CAN_MCR[IDAM]字段等效于格式 C，则只会将输入帧标识符的 8 个最高有效位与 Rx FIFO 滤波器进行比较。

42.4.16 Rx FIFO 信息寄存器 (CANx_RXFIR)

RXFIR 提供关于 Rx FIFO 的信息。

此寄存器是 CPU 通过其访问 RAM 中的 RXFIR FIFO 输出的端口。RXFIR FIFO 在每次将新消息移入 Rx FIFO 时由 FlexCAN 写入，在每次用下一条消息更新 Rx FIFO 的输出时，更新其输出。有关读取此寄存器的说明，请参阅“Rx FIFO”一节。

地址: 基址 基准 + 4Ch 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																IDHIT															
W																	IDHIT															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	

* 注:

- x = 复位时未定义。

CANx_RXFIR 字段描述

字段	描述
31–9 保留	此只读字段为保留字段且值始终为 0。
IDHIT	标识符接受滤波器命中指示器 此字段指示 Rx FIFO 输出中已接收消息命中的标识符接受滤波器。如果有多个滤波器匹配输入消息 ID ,则会指示匹配进程找到的第一个匹配 IDAF (最小编号)。此字段仅在 CAN_IFLAG1[BUF5]置位时有效。

42.4.17 CAN 位时序寄存器 (CANx_CBT)

此寄存器是存储 CAN_CTRL1 寄存器中描述的 CAN 位时序变量的一种备用方式。EPRES DIV、EPROPSEG、EPSEG1、EPSEG2 和 ERJW 分别为 PRES DIV、PROPSEG、PSEG1、PSEG2 和 RJW 位字段的扩展版本。

BTF 位选择使用此寄存器中定义的时序变量。

此寄存器的内容不受软复位的影响。

注

CAN_CTRL1 和 CAN_CBT 中的 CAN 位变量存储在相同的寄存器中。

地址: 基址 基准 + 50h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BTF	EPRES DIV											保留	ERJW		
W	BTF	EPRES DIV												ERJW		
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EPROPSEG							EPSEG1					EPSEG2			
W	EPROPSEG							EPSEG1					EPSEG2			
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_CBT 字段描述

字段	描述
31 BTF	<p>位时序格式使能</p> <p>使能运用扩展 CAN 位时序字段 EPRES DIV、EPROPSEG、EPSEG1、EPSEG2 和 ERJW ,替代 CAN_CTRL1 寄存器中定义的 CAN 位时序变量。此字段只能在冻结模式下写入。</p> <p>0 扩展位时间定义已禁用。 1 扩展位时间定义已使能。</p>
30–21 EPRES DIV	<p>扩展预分频器分频系数</p> <p>此 10 位字段定义在 CAN_CBT[BTF]位置位时，PE 时钟频率与串行时钟(Sclock)频率之比，否则，不起作用。此字段可扩展 CAN_CTRL1[PRES DIV]值的范围。</p> <p>Sclock 周期定义 CAN 协议的时间量子。对于复位值，Sclock 频率等于 PE 时钟频率 (见协议时序)。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>Sclock 频率 = PE 时钟频率 / (EPRES DIV + 1)</p>
20 保留	此字段为保留字段。
19–16 ERJW	<p>扩展再同步跳转宽度</p> <p>此 4 位字段定义在 CAN_CBT[BTF]位置位时，一位时间中一次再同步可以改变的最大时间量子数，否则，不起作用。此字段可扩展 CAN_CTRL1[RJW]值的范围。</p> <p>一个时间量子等于 Sclock 周期。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>再同步跳转宽度 = ERJW + 1。</p>
15–10 EPROPSEG	<p>扩展传播段</p> <p>此 6 位字段定义在当 CAN_CBT[BTF]位置位时，位时间中传播段的长度，否则，不起作用。此字段可扩展 CAN_CTRL1[PROPSEG]值的范围。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>传播段时间 = (EPROPSEG + 1) × 时间量子。 时间量子 = 一个 Sclock 周期。</p>
9–5 EPSEG1	<p>扩展相位段 1</p> <p>此 5 位字段定义在当 CAN_CBT[BTF]位置位时，位时间中相位段 1 的长度，否则，不起作用。此字段可扩展 CAN_CTRL1[PSEG1]值的范围。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>相位缓冲器段 1 = (EPSEG1 + 1) × 时间量子。 时间量子 = 一个 Sclock 周期。</p>
EPSEG2	<p>扩展相位段 2</p> <p>此 5 位字段定义在当 CAN_CBT[BTF]位置位时，位时间中相位段 2 的长度，否则，不起作用。此字段可扩展 CAN_CTRL1[PSEG2]值的范围。此字段仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。</p> <p>相位缓冲器段 1 = (EPSEG2 + 1) × 时间量子。 时间量子 = 一个 Sclock 周期。</p>

42.4.18 Rx 单独掩码寄存器 (CANx_RXIMRn)

RX 单独掩码寄存器用于将 ID 滤波接受掩码存储在 Rx MB 和 Rx FIFO 中。

当 Rx FIFO 被禁用时 (CAN_MCR[RFEN]位清 0)，为每个可用的 Rx 邮箱提供一个单独的掩码，并且一一对应。当 Rx FIFO 被使能时 (CAN_MCR[RFEN]位置位)，基于一一对应的关系，为每个 Rx FIFO ID 滤波器表元素提供一个单独的掩码，具体取决于 CAN_CTRL2[RFFN]的设置 (见 Rx FIFO)。

CAN_RXIMR0 存储与 MB0 或 ID 滤波器表元素 0 相关联的单独掩码，CAN_RXIMR1 存储与 MB1 或 ID 滤波器表元素 1 相关联的单独掩码，如此等等，不一而足。

CAN_RXIMR 寄存器只能在模块处于冻结模式时由 CPU 访问；否则，会被硬件阻止。这些寄存器不受复位影响。它们位于 RAM 中，必须在任何接收操作之前显性初始化。

地址: 基址 + 880h 偏移 + (4d × i), 其中 i=0d 到 15d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																	MI															
W																	MI															
复位	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	

*注:

- x = 复位时未定义。

CANx_RXIMRn 字段描述

字段	描述
MI	<p>单独掩码位</p> <p>每个单独掩码位以不同方式屏蔽邮箱滤波器和 Rx FIFO ID 滤波器表元素中的对应位。</p> <p>对于邮箱滤波器，请参阅 RXMGMASK 寄存器说明。</p> <p>对于 Rx FIFO ID 滤波器表元素，请参阅 RXFGMASK 寄存器说明。</p> <p>0 滤波器中的对应位为“无关”。</p> <p>1 检查滤波器中的对应位。</p>

42.4.53 消息缓冲器结构

FlexCAN 模块使用的消息缓冲器结构如下图所示。其中同时展示了 CAN 规格 (第 2.0 版 B 部分) 里用到的扩展版 (29 位标识符) 和标准版 (11 位标识符)。每个 MB 均由 16 个字节构成。

从 0x80 到 0x17F 的存储器区域由邮箱使用。

表 42-105. 消息缓冲器结构

	31	30	29	28	27	24	23	22	21	20	19	18	17	16	15	8	7	0
0x0	EDL	BRS	ESI		代码		SRR	IDE	RTR		DLC				时间戳			

下一页继续介绍此表...

表 42-105. 消息缓冲器结构 (继续)

0x4	PRI0	ID (标准/扩展)		ID (扩展)	
0x8	数据字节 0		数据字节 1		数据字节 2
0xC	数据字节 4		数据字节 5		数据字节 6
		= 未生效或保留			

CODE——消息缓冲器代码

作为消息缓冲器匹配和仲裁过程的一部分，此 4 位字段可通过 CPU 或 FlexCAN 模块本身访问 (读或写)。编码如表 42-106 和表 42-107 所示。其他信息请参阅功能说明。

表 42-106. Rx 缓冲器消息缓冲器

CODE 说明	在接收新帧之前的 Rx 代码	SRV ¹	在成功接收之后的 Rx 代码 ²	RRS ³	备注
0b0000: INACTIVE ——MB 无效。	INACTIVE	-	-	-	MB 不参与匹配过程。
0b0100: EMPTY ——MB 有效且为空。	EMPTY	-	FULL	-	当一个帧被成功接收 (在移入过程之后) 时, CODE 字段将自动更新为 FULL。
0b0010: FULL—— MB 已满。	FULL	是	FULL	-	先读取 C/S 字, 然后解锁 MB (SRV), 这一行为不会使代码返回 EMPTY, 代码将保持 FULL 状态。如果在对 MB 进行操作之后将一个新帧移至 MB, 代码仍将保持 FULL。有关 FULL 代码的详情, 请参阅匹配过程。
		否	OVERRUN	-	如果 MB 为 FULL, 并且在 CPU 对其进行操作之前将一个新帧移至此 MB, 则 CODE 字段将自动更新为 OVERRUN。有关溢出行为的详情, 请参阅匹配过程。
0b0110: OVERRUN——MB	OVERRUN	是	FULL	-	如果 CODE 字段指示 OVERRUN 且 CPU 已经对 MB 进

下一页继续介绍此表...

表 42-106. Rx 缓冲器消息缓冲器 (继续)

CODE 说明	在接收新帧之前的 Rx 代码	SRV ¹	在成功接收之后的 Rx 代码 ²	RRS ³	备注
正在被重写至已满的缓冲器。					行过操作, 则当一个新帧移至 MB 时, 代码将返回 FULL 状态。
		否	OVERRUN	-	如果 CODE 字段已经指示 OVERRUN 并且必须移动另一个新帧, 则将再次重写 MB, 代码将保持 OVERRUN。有关溢出行为的详情, 请参阅 匹配过程 。
0b1010: RANSWER ⁴ ——有一个帧被配置为识别远程请求帧, 并随后发送响应帧。	RANSWER	-	TANSWER(0b1110)	0	远程回答被配置为识别收到的远程请求帧。随后, 一个 MB 将被设置为发送响应帧。代码自动变更为 TANSWER (0b1110)。详情请参见 匹配过程 。如果 CAN_CTRL2[RRS] 清零, 则在每次收到 ID 相同的远程请求帧时发送一个响应帧。
		-	-	1	在匹配和仲裁过程中, 此代码将被忽略。详情请参见 匹配过程 。
CODE[0]=1: BUSY——FlexCAN 正在更新 MB 的内容。CPU 不得访问 MB。	BUSY ⁵	-	FULL	-	指示此 MB 正在更新。将自动清零, 并且不与干扰下一个 CODE。
		-	OVERRUN	-	

1. SRV: 经过操作的 MB。通过读取 TIMER 或其他 MB, 读取并解锁 MB。
2. 在帧移至 MB (移入过程) 之后, 则认为该帧被成功接收。详情请参见[移入](#)。
3. 远程请求存储位, 详见“控制 2 寄存器(CAN_CTRL2)”。
4. 不认为代码 0b1010 是 Tx, 不应终止含有此代码的 MB。
5. 注意, 对于 Tx MB, 应在读取时忽略 BUSY 位, 当 AEN 位在 MCR 寄存器中置位时除外。如果此位置位, 对应的 MB 不会参与匹配过程。

表 42-107. Tx 缓冲器消息缓冲器

CODE 说明	在 tx 帧之前的 Tx 代码	MB RTR	成功传输之后的 Tx 代码	备注
0b1000: INACTIVE——MB 无效。	INACTIVE	-	-	MB 不参与仲裁过程。
0b1001: ABORT——MB 终止	ABORT	-	-	MB 不参与仲裁过程。

下一页继续介绍此表...

表 42-107. Tx 缓冲器消息缓冲器 (继续)

CODE 说明	在 tx 帧之前的 Tx 代码	MB RTR	成功传输之后的 Tx 代码	备注
0b1100: DATA——MB 是一个 Tx 数据帧 (MB RTR 必须为 0)	DATA	0	INACTIVE	无条件发送一次数据帧。传输后, MB 自动返回 INACTIVE 状态。
0b1100: REMOTE——MB 是一个 Tx 远程请求帧 (MB RTR 必须为 1)	REMOTE	1	EMPTY	无条件发送一次远程请求帧。传输后, MB 自动变成 ID 相同的空 Rx MB。
0b1110: ANSWER——MB 是一个 Tx 响应帧, 来自输入远程请求帧	ANSWER	-	RANSWER	这是一个中间代码, 由 CHI 在与远程请求帧匹配之后自动写入 MB。远程响应帧会无条件发送一次, 然后, 代码就会返回 RANSWER (0b1010) 状态。CPU 也可写入此代码, 其作用相同。远程响应帧既可以是一个数据帧, 也可以是另一个远程请求帧, 具体取决于 RTR 位的值。更多详情, 请参见 匹配过程 和 仲裁过程 。

SRR ——替代远程请求

固定隐性位, 只能用于扩展格式。对于发送 (Tx 缓冲区), 用户必须将其置 1; 对于 Rx 接收缓冲区, 将与接收到的值一起存储在 CAN 总线上。既可隐性接收, 也可显性接收。如果 FlexCAN 显性接收此位, 则将其解读为仲裁丢失。

1 = 对于以扩展格式帧发送的情况, 隐性值是必须的

0 = 对于以扩展格式帧发送的情况, 显性值无效

IDE ——ID 扩展位

此字段用于标识帧格式是标准格式还是扩展格式。

1 = 帧格式为扩展格式

0 = 帧格式为标准格式

RTR ——远程发送请求

此位会影响远程帧的行为, 并且是接收滤波器的一部分。有关详情, 请参阅[表 42-106](#)、[表 42-107](#) 以及“控制 2 寄存器(CAN_CTRL2)”中关于 RRS 位的说明。

如果 FlexCAN 以'1'发送此位 (隐性), 以'0'接收此位 (显性), 则将其解读为仲裁丢失。如果此位发送为'0' (显性), 然后接收为'1' (隐性), 则 FlexCAN 模块会将其视为位错误。如果接收的值与发送的值相匹配, 则认为位发送成功。

1 = 指示如果 MB 为 Tx，当前 MB 可能有一个远程请求帧需要发送。如果 MB 为 Rx，则可能存储输入远程请求帧。

0 = 指示当前 MB 有一个数据帧需要发送。在 Rx MB 下，可在匹配过程中予以考虑。

DLC——数据的字节长度

此 4 位字段为 Rx 或 Tx 数据的字节长度，位于偏移量为 0x8 至 0xF 的 MB 空间中（见表 42-105）。在接收过程中，此字段由 FlexCAN 模块进行写操作，由已接收帧的 DLC（数据长度代码）字段复制而来。在发送过程中，此字段由 CPU 进行写操作，并且对应于待发送帧的 DLC 字段值。RTR = 1 时，待发送帧为一个远程帧，不包括数据字段，与 DLC 字段无关（见表 42-108）。

TIME STAMP——自由运行计数器时间戳

此 16 位字段为自由运行定时器的副本，在标识符字段出现在 CAN 总线上时为 Tx 和 Rx 帧捕捉而得。

PRI0——本地优先级

此 3 位字段只用于 LPRIO_EN 位在 CAN_MCR 中置位的情况，并且仅对 Tx 邮箱有意义。这些位不会发送。它们要附加到常规 ID 上，以定义发送优先级。请参见仲裁过程。

ID——帧标识符

采用标准帧格式，接收和发送都只使用 11 个最高有效位（28 至 18）来实现帧标识符目的。18 个最低有效位会被忽略。在扩展帧格式中，接收和发送都会使用全部位来标识帧。

DATA BYTE 0 - 7——数据字段

最多可将 8 个字节用于一个数据帧。

对于 Rx 帧，数据的存储方式与从 CAN 总线接收的数据相同。DATA BYTE (n) 仅在 n 小于 DLC 时有效，如下表所示。

表 42-108. DATA BYTE 有效性

DLC	有效 DATA BYTE
0	无
1	DATA BYTE 0
2	DATA BYTE 0 - 1
3	DATA BYTE 0 - 2
4	DATA BYTE 0 - 3

下一页继续介绍此表...

表 42-108. DATA BYTE 有效性 (继续)

DLC	有效 DATA BYTE
5	DATA BYTE 0 - 4
6	DATA BYTE 0 - 5
7	DATA BYTE 0 - 6
8 或以上	DATA BYTE 0 - 7

42.4.54 Rx FIFO 结构

当 CAN_MCR[RFEN]位置位时，从 0x80 至 0xDC 的存储器区域 (通常由 0-5 号 MB 占据) 将由接收 FIFO 引擎所用。

0x80-0x8C 区域含有必须由 CPU 作为消息缓冲器读取的 FIFO 的输出。该输出含有已接收但尚未读取的最老旧的消息。区域 0x90-0xDC 保留供 FIFO 引擎内部使用。

另一个存储器区域 (始于 0xE0 且可能延伸到 0x17C (通常被 6-15 号 MB 占用), 具体取决于 CAN_CTRL2[RFFN]字段的设置) 含有 ID 滤波器表 (可配置 8 至 40 个表元素), 该表指定用于将帧接受进 FIFO 的滤波标准。

复位后, ID 滤波器表灵活存储器区域默认为 0xE0 且仅延伸至 0xFC, 在 RFFN = 0 时对应于 6-7 号 MB, 以便向后兼容以前版本的 FlexCAN。

下面展示了 Rx FIFO 数据结构。

表 42-109. Rx FIFO 结构

	31	28	24	23	22	21	20	19	18	17	16	15	8	7	0	
0x80	IDHIT				SRR	IDE	RTR	DLC				时间戳				
0x84	ID 标准								ID 扩展							
0x88	数据字节 0				数据字节 1				数据字节 2				数据字节 3			
0x8C	数据字节 4				数据字节 5				数据字节 6				数据字节 7			
0x90 到 0xDC	保留															
0xE0	ID 滤波器表元素 0															
0xE4	ID 滤波器表元素 1															
0xE8 到 0x2D4	ID 滤波器表元素 2 至 125															
0x2D8	ID 滤波器表元素 126															
0x2DC	ID 滤波器表元素 127															
	= 未生效或保留															

ID 滤波器表中的每个元素都占用一整个 32 位字，可由 1、2 或 4 个标识符接受滤波器(IDAF)构成，具体取决于 CAN_MCR[IDAM]字段的设置。下面的图展示的是 IDAF 索引情况。

下表展示了 ID 表元素的三种不同格式。注意，表中的所有元素都必须具有相同的格式。更多详情，请参阅 [Rx FIFO](#)。

表 42-110. ID 表结构

格式	31	30	29	24	23	16	15	14	13	8	7	1	0	
A	RTR	IDE	RXIDA (标准 = 29-19, 扩展 = 29-1)											
B	RTR	IDE	RXIDB_0 (标准 = 29-19, 扩展 = 29-16)				RTR	IDE	RXIDB_1 (标准 = 13-3, 扩展 = 13-0)					
C	RXIDC_0 (std/ext = 31-24)			RXIDC_1 (std/ext = 23-16)			RXIDC_2 (std/ext = 15-8)			RXIDC_3 (std/ext = 7-0)				
	= 未生效或保留													

RTR——远程帧

此位指定在远程帧匹配目标 ID 时，是否将其接受进 FIFO 中。

1 = 可以接受远程帧并拒绝数据帧

0 = 拒绝远程帧且可接受数据帧

IDE——扩展帧

指定在它们匹配目标 ID 时，是将扩展还是标准帧接受进 FIFO 中。

1 = 可以接受扩展帧并拒绝标准帧

0 = 拒绝扩展帧且可接受标准帧

RXIDA——Rx 帧标识符 (格式 A)

指定将一个 ID 作为 FIFO 的接受标准。在标准帧格式中，只有 11 个最高有效位 (29 至 19) 会被用于帧标识目的。在扩展帧格式下，会使用所有位。

RXIDB_0, RXIDB_1——Rx 帧标识符 (格式 B)

指定将一个 ID 作为 FIFO 的接受标准。在标准帧格式中，只有 11 个最高有效位 (完整标准 ID) (29 至 19 和 13 至 3) 会被用于帧标识目的。在扩展帧格式中，会将字段的全部 14 位与已接收 ID 的 14 个最高有效位进行比较。

RXIDC_0, RXIDC_1, RXIDC_2, RXIDC_3——Rx 帧标识符 (格式 C)

指定将一个 ID 作为 FIFO 的接受标准。在标准和扩展帧格式中，会将字段的全部 8 位与已接收 ID 的 8 个最高有效位进行比较。

IDHIT ——标识符接受滤波器命中指示器

此 9 位字段指示 Rx FIFO 输出中已接收消息命中的标识符接受滤波器。更多详情，请参阅 [Rx FIFO](#)。

42.5 功能说明

FlexCAN 模块是一个 CAN 协议引擎，有一个非常灵活的邮箱系统来发送和接收 CAN 帧。邮箱系统由一组消息缓冲器(MB)构成，这些缓冲器存储配置和控制数据、时间戳、消息 ID 和数据（见[消息缓冲器结构](#)）。对应于前 38 个 MB 的存储器可以配置为支持 FIFO 接收方案，加上一种强大的 ID 过滤机制，可以根据一张 ID 表检查输入帧（最多 128 个扩展 ID 或 256 个标准 ID 或 512 个 8 位 ID 片），同时还为最多 32 个 ID 滤波器表元素提供单独的掩码寄存器。

支持通过 FIFO 和邮箱同时接收。对于邮箱接收，可以借助一种匹配算法将接收到的帧仅存入其 ID 字段中配置的 ID 相同的 MB 中。借助一种掩码方案可以将 MB 上配置的 ID 与接收到的 CAN 帧上的 ID 范围进行匹配。传输时，仲裁算法基于消息 ID（可选择通过 3 个局部优先级位增强）或 MB 排序决定要发送的 MB 的优先顺序。

在继续说明功能之前，必须解释一个重要概念。如果一个消息缓冲器可以参与匹配和仲裁两个过程，则称此消息缓冲器在给定时间“有效”。代码为 0b0000 的 Rx MB 被视为无效（请参阅[表 42-106](#)）。类似地，代码为 0b1000 或 0b1001 的 Tx MB 也被视为无效（请参阅[表 42-107](#)）。

42.5.1 传输流程

要传输 CAN 帧，CPU 必须通过执行以下程序来配置用于传输的消息缓冲区：

1. 检查相应的中断位是否已置位和清零。
2. 如果 MB 处于活动状态（传输挂起），则将 ABORT 代码（0b1001）写入控制和状态字的 CODE 字段，以请求中止传输。通过轮询 CAN_IFLAG 寄存器或相应 IMASK 位的中断请求（如使能）等待置位对应的 IFLAG 位。然后回读 CODE 字段，以检查传输已中止还是已传输完毕（参见[传输中止机制](#)）。如果需要向后兼容（CAN_MCR[AEN]位清零），则仅将 INACTIVE 代码（0b1000）写入 CODE 字段，以禁用 MB，但是，这种操作可能导致将目前挂起的 CAN 帧传输出去（参见[邮箱禁用](#)）。

3. 写入 ID 字。
4. 写入数据字节。
5. 写入控制和状态字的 DLC、Control 和 CODE 字段，以激活 MB。

激活 MB 时，它将参与仲裁过程，并最终根据其优先级传输。

成功传输结束后，自由运行定时器的值将写入时间戳字段，控制和状态字的 CODE 字段将会更新，CRC 寄存器将会更新，中断标志位寄存器中将置位状态标志位，并生成中断（如果对应的中断屏蔽寄存器位允许）。传输后的新 CODE 字段取决于用于激活 MB 的代码（参见消息缓冲器结构中的表 42-106 和表 42-107）。

中止功能使能时（CAN_MCR[AEN]被置位），当一个被配置为 Tx 的邮箱的中断标志置位时，邮箱将会 Block 住。因此，在 CPU 清掉中断标志位之前，CPU 无法对其进行更新。这意味着 CPU 必须清除对应的 IFLAG 位，然后再开始准备将此 MB 用于新的传输或接收。

42.5.2 仲裁过程

仲裁过程对邮箱扫描，搜索其中保存着要在下一次机会发送的消息的 Tx 邮箱。该邮箱被称为仲裁优胜者。

扫描从编号最低的邮箱开始，然后处理编号较高的。

仲裁过程在下列事件中触发：

- 从 CAN 帧的 CRC 字段。起点取决于 CAN_CTRL2[TASD]字段值。
- 在 CAN 帧的错误定界符字段中。
- 在 CAN 帧的过载定界符字段中。
- 当优胜者被禁用且 CAN 总线仍然未达到间隙字段的第一位时。
- 当 CPU 向优胜者 MB 的 C/S 字进行写入操作且 CAN 总线仍然未达到间隙字段的第一位时。
- 当 CHI 处于空闲状态且 CPU 向任意 MB 的 C/S 字进行写入操作时。
- 当 FlexCAN 退出总线关闭状态时。
- 当退出冻结模式或低功耗模式时。

如果仲裁过程不能在 CAN 总线达到间隙字段第一位之前完成检查所有邮箱，则临时仲裁优胜者将被禁用，并且 FlexCAN 不会在下一次机会中竞争 CAN 总线。

仲裁过程在扫描结束时，根据 CAN_CTRL1[LBUF]和 CAN_MCR[LPRIOEN]位的设置，从有效 Tx 邮箱中选择优胜者。

42.5.2.1 编号最低的邮箱优先

如果 CAN_CTRL1[LBUF]位置位，则找到的第一个（编号最低）有效 Tx 邮箱为仲裁优胜者。CAN_MCR[LPRIOEN]位在 CAN_CTRL1[LBUF]置位时不起作用。

42.5.2.2 最高优先级邮箱优先

如果 CAN_CTRL1[LBUF]位清零，则仲裁过程搜索优先级最高的有效 Tx 邮箱，即就是说，当有多个外部节点同时竞争 CAN 总线时，此邮箱的帧赢得 CAN 总线上仲裁的概率更大。

此仲裁考虑的位序列称为邮箱的仲裁值。优先级最高的 Tx 邮箱为所有 Tx 邮箱中仲裁值最低者。

如果两个或以上邮箱的仲裁值相等，则编号最低的邮箱为仲裁优胜者。

仲裁值的构成取决于 CAN_MCR[LPRIOEN]位的设置。

42.5.2.2.1 本地优先级被禁用

如果 CAN_MCR[LPRIOEN]位清 0，则仲裁值的构建位序列与以下情况完全相同：它们在 CAN 帧中发送时（见下表）会使本地优先级被禁用。

表 42-111. 本地优先级被禁用时的仲裁值构成

格式	邮箱仲裁值 (32 位)				
标准(IDE = 0)	标准 ID (11 位)	RTR (1 位)	IDE (1 位)	- (18 位)	- (1 位)
扩展(IDE = 1)	扩展 ID[28:18] (11 位)	SRR (1 位)	IDE (1 位)	扩展 ID[17:0] (18 位)	RTR (1 位)

42.5.2.2.2 本地优先级使能

如果需要本地优先级，则 CAN_MCR[LPRIOEN]必须置位。在这种情况下，邮箱 PRIO 字段包括在仲裁值最左侧（见下表）。

表 42-112. 本地优先级被使能时的仲裁值构成

格式	邮箱仲裁值 (35 位)					
标准(IDE = 0)	PRIO (3 位)	标准 ID (11 位)	RTR (1 位)	IDE (1 位)	- (18 位)	- (1 位)

下一页继续介绍此表...

表 42-112. 本地优先级被使能时的仲裁值构成 (继续)

格式	邮箱仲裁值 (35 位)					
扩展(IDE = 1)	PRIO (3 位)	扩展 ID[28:18] (11 位)	SRR (1 位)	IDE (1 位)	扩展 ID[17:0](18 位)	RTR (1 位)

由于 PRIO 字段为仲裁值中最有效的部分，因此，PRIO 值较低的邮箱的优先级高于 PRIO 值较高的邮箱，无论其仲裁值的其余部分如何。

注意，PRIO 字段不是 CAN 总线上的帧的一部分。其上的是只影响内部仲裁过程。

42.5.2.3 仲裁过程 (续)

在找到仲裁优胜者之后，其内容将被复制到一个称为 Tx 串行消息缓冲器 (Tx SMB) 的隐藏辅助 MB 中，该缓冲器的结构与正常 MB 相同，只是用户不可访问。此操作称为移出，并且在其完成后，对相应 MB 的 C/S 字进行的写入访问将被阻止 (如果 CAN_MCR 寄存器中的 AEN 位置位)。在下列事件中将恢复写入访问：

- 在 CPU 发送 MB 且清零对应的 IFLAG 位之后
- FlexCAN 进入冻结模式或总线关闭模式
- FlexCAN 在总线仲裁中失利或者传输过程中出现错误

在 CAN 总线上的第一个机会窗口，Tx SMB 上的消息按照 CAN 协议规则发送。

在下列情况下，可以触发仲裁过程：

- 在从 CAN CRC 字段到帧结束的 Rx 和 Tx 帧中。CAN_CTRL2[TASD]位的值可以更改以优化仲裁起点。
- 在从 TX_ERR_CNT=124 到 128 的 CAN 总线关闭状态中。CAN_CTRL2[TASD]位的值可以更改以优化仲裁起点。
- CPU 在总线空闲状态下进行的 C/S 写入操作中。第一个 C/S 写入操作启动仲裁过程，同一仲裁过程中的第二个 C/S 写入操作重启此过程。如果执行了其他 C/S 写入操作，则 Tx 仲裁过程将挂起。如果在仲裁过程结束后无仲裁优胜者，则 TX 仲裁机会开始新的仲裁过程。如果有一个挂起的仲裁过程且总线空闲状态开始，则会触发一个仲裁过程。在这种情况下，总线空闲状态下的第一和第二个 C/S 写入操作不会重启仲裁过程。可能没有足够的时间来在等待总线空闲状态下完成仲裁，下一状态为空闲。这种情况下，扫描不会中断，并且会在总线空闲状态下完成。在此仲裁过程中，C/S 写入操作不会导致仲裁重启。
- 在有效仲裁窗口中禁用仲裁优胜者。
- 在退出冻结模式时 (等待总线空闲状态的第一位)。如果在等待总线空闲状态中有再同步，则仲裁过程重启。

在下列情况下，仲裁过程将停止：

- 所有邮箱扫描完毕
- 在使能最低缓冲器特性的情况下找到了一个 Tx 有效邮箱
- 在任意仲裁过程中禁用或中止仲裁优胜者
- 没有足够的时间来完成 Tx 仲裁过程（例如，当接近帧结束时执行禁用操作）。这种情况下，仲裁过程将挂起。
- 总线中的错误或过载标志
- 空闲状态下的低功耗或冻结模式请求

在下述情况下，称仲裁挂起：

- 不能及时完成仲裁过程
- 仲裁过程中执行 C/S 写入操作（如果写操作是在其编号低于 Tx 仲裁指针的 MB 中执行的）
- 任何 C/S 写入操作（如果无正在进行的 Tx 仲裁过程）
- Rx 匹配刚刚将 Rx 代码更新为 Tx 代码
- 进入总线关闭状态

在仲裁过程中进行 C/S 写入操作具有下列作用：

- 如果 C/S 写入操作是在仲裁优胜者执行的，则会立即重启新的仲裁过程。
- 如果 C/S 写入操作是在其编号高于 Tx 仲裁指针的 MB 中执行的，则正在进行的仲裁过程会如正常情况下一样扫描此 MB。

42.5.3 接收过程

为将 CAN 帧收入邮箱，CPU 必须通过执行以下步骤使邮箱做好接收准备：

1. 如果邮箱有效 (Tx 或 Rx)，则禁用邮箱（见[邮箱禁用](#)），最好采用安全禁用方式（见[传输中止机制](#)）。
2. 写入 ID 字
3. 将 EMPTY 代码(0b0100)写入控制和状态字的 CODE 字段以激活邮箱。

在 MB 激活之后，就可以接收与编程滤波器相匹配的帧了。在成功接收结束时，邮箱将由移进过程更新（见[移入](#)）如下：

1. 存储已接收数据字段（经典 CAN 消息格式最多 8 字节）。
2. 存储已接收标识符字段。
3. 自由运行定时器在帧标识符字段第二位的值被写入邮箱时间戳字段。
4. 存储已接收 SRR、IDE、RTR 和 DLC 字段。

5. 更新控制和状态字中的 CODE 字段 (见第表 42-106 节中的表 42-107 和消息缓冲器结构)。
6. 中断标志寄存器中的一个状态标志置位, 如果相应的中断掩码寄存器位允许则生成一个中断。

CPU 处理 (读取) 邮箱中已接收帧的建议方式是使用以下程序:

1. 读取该邮箱的控制和状态字。
2. 检查 BUSY 位是否已解除置位, 用于表示邮箱已锁定。在置位时, 重复步骤 1)。请参见邮箱锁定机制。
3. 读取邮箱的内容。邮箱锁定后, 其内容就不能被 FlexCAN 移入过程修改。请参见移入。
4. 应答 IFLAG 寄存器的相应标志。
5. 读取自由运行定时器。虽然是可选的, 但仍然建议尽快对邮箱解锁, 并使其准备进行接收操作。

CPU 应通过 IFLAG 寄存器之一中的特定邮箱的状态标志位, 而非该邮箱的 CODE 字段来轮询帧接收。轮询 CODE 字段不起作用, 因为帧一旦被接收并且 CPU 开始处理邮箱 (通过读取 C/S 字, 然后对邮箱解锁), CODE 字段就不会返回 EMPTY 状态。该字段将保持 FULL 状态, 具体说明见表 42-106。如果 CPU 试图避开这种行为, 即在事先未进行安全禁用的情况下, 在读取邮箱之后, 通过写入 C/S 字的方式强制写入 EMPTY 代码, 则可能丢失与该邮箱的滤波器相匹配的新接收到的帧。

警告

总之: 不要通过直接读取邮箱的 C/S 字进行轮询。要读取 IFLAG 寄存器。

请注意, 已接收帧的标识符字段始终存储在匹配的邮箱中, 因此, 如果匹配是掩码造成的, 则邮箱 ID 字段中的内容就可能发生变化。当 CAN_MCR[SRXDIS]位置位时, FlexCAN 不会把其本身发送的帧存入任何 MB 中, 即使含有一个匹配的 Rx Mailbox, 也是如此; 并且不会生成中断标志或中断信号。否则, 当 CAN_MCR[SRXDIS]位解除置位时, 如果存在匹配的 Rx 邮箱, FlexCAN 就可能接收它自己发送的帧。

为了在 Rx FIFO 过程中接收 CAN 帧, CPU 必须在冻结模式下使能并配置 Rx FIFO (见 Rx FIFO)。在接收到 Rx FIFO 中断中可用的帧时 (见对 CAN_IFLAG1 寄存器中 BUF5I 位 "Rx FIFO 中可用帧" 位的说明), CPU 应使用下列程序处理接收的帧:

1. 读取控制和状态字 (可选: 仅在为 IDE 和 RTR 位使用掩码时才需要)

2. 读取 ID 字段 (可选: 仅在使用掩码时需要)
3. 读取数据字段
4. 读取 CAN_RXFIR 寄存器 (可选)
5. 向 CAN_IFLAG1[BUF5I]位写入 1, 由此清零 Rx FIFO 中断中的可用帧 (强制: 释放 MB, 并允许 CPU 读取下一条 Rx FIFO)

当 CAN_MCR[DMA]置位时, 在 FIFO 中接收到帧时, CAN_IFLAG1[BUF5I]会生成一个 DMA 请求, 且不会生成 CPU 中断 (见 [DMA 工作模式下的 Rx FIFO](#))。不使用 Rx FIFO 区的 CAN_IMASK1 位。

DMA 控制器必须使用以下程序处理已接收帧:

1. 读取控制和状态字 (读取地址 0x80, 可选)
2. 读取 ID 字段 (读取地址 0x84, 可选)
3. 读取所有数据字节 (在地址 0x88 处开始读取, 可选)
4. 读取最后数据字节 (必须读取地址 0x8C)

42.5.4 匹配过程

匹配过程扫描 MB 存储器, 寻找与从 CAN 总线接收到的 Rx MB 同 ID 的 Rx MB。如果 FIFO 使能, 扫描优先级可以在邮箱和 FIFO 滤波器之间进行选择。匹配从编号最低的消息缓冲器开始, 然后处理编号较高的。如果在第一个结构中未找到匹配项, 则接着扫描下一个结构。在 FIFO 已满的情况下, 匹配算法将始终在 FIFO 区域之外寻找匹配的 MB。

被接收后, 帧将存储在一个称为 Rx 串行消息缓冲器(Rx SMB)的隐藏辅助 MB 中。

匹配过程的起点取决于下列条件:

- 如果接收帧为一个远程帧, 则起点为该帧的 CRC 字段
- 如果接收帧为一个 DLC 字段等于零的数据帧, 则起点为该帧的 CRC 字段
- 如果接收帧为一个 DLC 字段不为零的数据帧, 则起点为该帧的 DATA 字段

如果在 FIFO 表或其中一个邮箱中找到了一个匹配 ID, 则 Rx SMB 的内容将被移入过程传输至 FIFO 或匹配的邮箱中。如果检测到任意 CAN 协议错误, 则不会在接收结束时将匹配结果传输至 FIFO 或匹配的邮箱。

匹配过程扫描 Rx FIFO (若使能) 和有效 Rx 邮箱中的所有匹配元素 (CODE 为 EMPTY、FULL、OVERRUN 或 RANSWER), 以便与正在接收 CAN 总线上的该帧的 Rx SMB 的匹配元素进行成功的比较。Rx SMB 具有与邮箱相同的结构。与已经成功比较的匹配元素相关联的接收结构 (Rx FIFO 或邮箱) 为匹配的结构。在扫描结束时, 从这些匹配的结构中选出匹配优胜者, 具体取决于前面描述的条件。参见下表。

表 42-113. 匹配架构

结构	SMB[RTR]	CTRL2[RRS]	CTRL2[EAC EN]	MB[IDE]	MB[RTR]	MB[ID ¹]	MB[CODE]
邮箱	0	-	0	cmp ²	no_cmp ³	cmp_msk ⁴	EMPTY 或 FULL 或 OVERRUN
邮箱	0	-	1	cmp_msk	cmp_msk	cmp_msk	EMPTY 或 FULL 或 OVERRUN
邮箱	1	0	-	cmp	no_cmp	cmp	RANSWER
邮箱	1	1	0	cmp	no_cmp	cmp_msk	EMPTY 或 FULL 或 OVERRUN
邮箱	1	1	1	cmp_msk	cmp_msk	cmp_msk	EMPTY 或 FULL 或 OVERRUN
FIFO ⁵	-	-	-	cmp_msk	cmp_msk	cmp_msk	-

1. 对于邮箱结构, 如果 SMB[IDE]置位, 则 ID 为 29 位 (标准 ID + 扩展 ID)。如果 SMB[IDE]清零, 则 ID 只有 11 位 (标准 ID)。对于 FIFO 结构, ID 取决于 IDAM。
2. cmp: 将 Rx SMB 的内容与 MB 的内容进行比较, 无论是否使用掩码。
3. no_cmp: Rx SMB 的内容不与 MB 的内容进行比较。
4. cmp_msk: 将 Rx SMB 的内容与 MB 的内容进行比较, 要考虑掩码的使用。
5. 当 IDAM 为 C 类时, 不会考虑 SMB[IDE]和 SMB[RTR]。

符合下列任意条件时, 接收结构可自由接收:

- 邮箱的 CODE 字段为 EMPTY
- 邮箱的 CODE 字段为 FULL 或 OVERRUN, 并且已经过处理 (C/S 字已由 CPU 读取并解锁, 如[邮箱锁定机制](#)所述)
- 邮箱的 CODE 字段为 FULL 或 OVERRUN, 并且形成了禁用 (见[邮箱禁用](#))
- Rx FIFO 不满

邮箱和 Rx FIFO 的扫描顺序为从编号最低的匹配元素到编号较高的匹配元素。

邮箱匹配优胜者的搜索受 CAN_MCR[IRMQ]位的影响。如果清零, 则匹配优胜者为匹配的第一个邮箱, 无论是否为自由接收邮箱。如果置位, 则根据下列优先级选择匹配优胜者:

1. 第一个自由接收匹配邮箱;
2. 最后一个非自由接收匹配邮箱。

可以通过 CAN_CTRL2[MRP]位在邮箱与 Rx FIFO 之间选择扫描优先级。

如果所选优先级为 Rx FIFO 优先:

- 如果 Rx FIFO 为匹配结构且为自由接收, 则 Rx FIFO 为匹配优胜者, 与邮箱扫描无关
- 否则 (Rx FIFO 不是匹配结构或不是自由接收), 则如上所述, 在邮箱中搜索匹配优胜者

如果所选优先级为邮箱优先:

- 如果找到自由接收匹配邮箱, 则该邮箱为匹配优胜者, 与 Rx FIFO 扫描无关
- 如果未找到匹配邮箱, 则在 Rx FIFO 扫描中搜索匹配优胜者
- 如果以上两个条件都不满足并且找到一个非自由接收匹配邮箱, 则匹配优胜者的决定由 CAN_MCR[IRMQ]调节:
 - 如果 CAN_MCR[IRMQ]位清零, 则匹配优胜者为第一个匹配邮箱
 - 如果 CAN_MCR[IRMQ]位置位, 则匹配优胜者为 Rx FIFO (若其为自由接收匹配结构), 否则, 匹配优胜者为最后一个非自由接收匹配邮箱

有关各种匹配可能性的总结, 请参阅下表。

表 42-114. 匹配可能性与结果的接收结构

RFEN	IRMQ	MRP	在 MB 中匹配	在 FIFO 中匹配	接收结构	说明
无 FIFO, 仅 MB, 匹配始终为 MB 优先						
0	0	X ¹	无 ²	- ³	无	无匹配导致帧丢失
0	0	X	自由 ⁴	-	第一 MB	
0	1	X	无	-	无	无匹配导致帧丢失
0	1	X	免费	-	第一 MB	
0	1	X	非自由	-	最后 MB	溢出
FIFO 使能, FIFO 中无匹配, 就如 FIFO 不存在一样						
1	0	X	无	无 ⁵	无	无匹配导致帧丢失
1	0	X	免费	无	第一 MB	
1	1	X	无	无	无	无匹配导致帧丢失
1	1	X	免费	无	第一 MB	
1	1	X	非自由	无	最后 MB	溢出
FIFO 使能, 队列禁用						
1	0	0	X	非满 ⁶	FIFO	
1	0	0	无	满 ⁷	无	FIFO 满造成的帧丢失 (FIFO 溢出)
1	0	0	免费	满	第一 MB	

下一页继续介绍此表...

表 42-114. 匹配可能性与结果的接收结构 (继续)

RFEN	IRMQ	MRP	在 MB 中匹配	在 FIFO 中匹配	接收结构	说明
1	0	0	非自由	满	第一 MB	
1	0	1	无	非满	FIFO	
1	0	1	无	满	无	FIFO 满造成的 帧丢失 (FIFO 溢 出)
1	0	1	免费	X	第一 MB	
1	0	1	非自由	X	第一 Mb	溢出
FIFO 使能, 队列使能						
1	1	0	X	非满	FIFO	
1	1	0	无	满	无	FIFO 满造成的 帧丢失 (FIFO 溢 出)
1	1	0	免费	满	第一 MB	
1	1	0	非自由	满	最后 MB	溢出
1	1	1	无	非满	FIFO	
1	1	1	免费	X	第一 MB	
1	1	1	非自由	非满	FIFO	
1	1	1	非自由	满	最后 MB	溢出

1. 此为无关条件。
2. 在 MB 中匹配“无”表示，该帧尚未匹配任何 MB (自由接收或非自由接收)。
3. 此为禁止条件。
4. 在 MB 中匹配“自由”表示该帧匹配了至少一个 MB 自由接收，无论其是否匹配了 MB 非自由接收。
5. 在 FIFO 中匹配“无”表示，该帧尚未匹配 FIFO 中的任何滤波器。就如 FIFO 不存在一样(CAN_CTRL2[RFEN]=0)。
6. 在 FIFO 中匹配“非满”表示，该帧匹配了一个 FIFO 滤波器，但其有空槽来接收它。
7. 在 FIFO 中匹配“满”表示，该帧匹配了一个 FIFO 滤波器，但无法存储，因为无空槽来接收它。

如果在匹配过程中发生非安全邮箱禁用 (见 [邮箱禁用](#)) 并且被禁用的邮箱为临时匹配优胜者，则该临时匹配优胜者无效。匹配元素扫描不会停止，也不会重启，而是正常继续执行。其结果，当前匹配过程将继续工作，就如在禁用之前比较的匹配元素不存在一样，因此，可能导致消息丢失。

例如，假设 FIFO 禁用，IRMQ 使能且有两个 MB 的 ID 相同，并且 FlexCAN 开始接收该 ID 的消息。我们假定，这些 MB 是阵列中的第二和第五元素。当第一条消息到达时，匹配算法会在 2 号 MB 中查找第一个匹配项。MB 的代码为 EMPTY，因此将消息存储于此。当第二条消息到达时，匹配算法会再次查找 2 号 MB，但此 MB 不是“自由接收”邮箱，因此算法会继续查找，结果找到 5 号 MB 并把消息存储于此。如果还有一条 ID 相同的消息到达，则匹配算法会发现，不存在“自由接收”匹配 MB，因而决定覆盖最后一个匹配 MB，即 5 号 MB。这样做的时候，会置位 MB 的 CODE 字段，以指示 OVERRUN 状态。

可以利用在一个以上的 MB 中匹配相同 ID 的能力，以实现接收队列（在全功能 FIFO 以外），以便为 CPU 处理 MB 提供更多时间。通过将一个以上的 MB 配置为相同的 ID，可将接收的消息排队存入 MB 中。CPU 可以检查 MB 的时间戳字段，以确定消息的到达顺序。

通过运用 ID 接受掩码可以与多个 ID 匹配。FlexCAN 支持对各 MB 进行单独掩码。请参阅关于 Rx 独立掩码寄存器(CAN_RXIMRx)。在匹配算法执行过程中，如果掩码位置位，则会比较对应的 ID 位。如果掩码位清零，则对应的 ID 位无关位。请注意，独立掩码寄存器是在 RAM 中实现的，因此，它们不会在复位后初始化。另外，它们只能在模块处于冻结模式时配置；否则，会被硬件阻塞。

FlexCAN 也支持一种备选掩码方案，只有四个掩码寄存器 (CAN_RXFGMASK、CAN_RXMGMASK、CAN_RX14MASK 和 CAN_RX15MASK)，以便向后兼容传统应用程序。当 CAN_MCR 寄存器中的 IRMQ 位清零时，这种备选掩码方案会启用。

42.5.5 移动过程

有两类移动过程：移入和移出。

42.5.5.1 移入

移入是将 Rx SMB 接收到的消息复制到与其匹配的 Rx 邮箱或 FIFO 的过程。如果移动的目标为 Rx FIFO，则同时会把消息属性复制到 CAN_RXFIRFIFO。每个 Rx SMB 有自己的移入过程，但如上所述，给定时间只能执行一个移位过程。仅在 Rx SMB 保存的消息有对应的匹配优胜者（见[匹配过程](#)）并且符合所有下列条件时，移入才会开始：

- CAN 总线已达到或通过：
 - 与 Rx SMB 中消息携带帧相邻的间隙字段的第二位
 - 与 Rx SMB 中消息携带帧相邻的过载帧的第一位
- 无正在进行的匹配过程
- 目标邮箱未被 CPU 锁定
- 无来自另一 Rx SMB 的正在进行的移入过程。如果有超过一个移入过程要同时启动，则都予以执行，最新的过程替代最旧的过程。

术语挂起移入在全部文档中均有使用，表示尚不满足所有前述条件的待移操作。

如果符合任一下列条件，移入将被取消且 Rx SMB 能接收另一条消息：

- 目标邮箱在 CAN 达到与消息携带帧相邻的间隙字段的第一位之后被禁用并且其匹配过程已完成
- 前面存在以同一邮箱为目标的挂起移入

- Rx SMB 正在接收由 FlexCAN 本身发送的帧并且自动接收已禁用 (CAN_MCR[SRXDIS]位认定)
- 检测到任何 CAN 协议错误

注意，如果模块进入冻结或低功耗模式，则不会取消挂起移入只会保持观望，等待退出冻结和低功耗模式并解锁。如果 MB 在冻结模式下解锁，则移入会立即进行。

移入是 FlexCAN 执行下列步骤的过程：

1. 如果消息以 Rx FIFO 为目标，将 IDHIT 推进 RXFIR FIFO。
2. 从 Rx SMB 读取 DATA0-3 和 DATA4-7 字。
3. 将 DATA0-3 和 DATA4-7 字写入 Rx 邮箱
4. 从 Rx SMB 读取控制/状态和 ID 字。
5. 将控制/状态和 ID 字写入 Rx 邮箱，并更新 CODE 字段。

移入过程不是原子过程，因为在目标邮箱禁用之后会被立即取消（见[邮箱禁用](#)），并且在这种情况下，邮箱可能仅更新一部分，因而会出现不一致的问题。移入目标为 Rx FIFO 消息缓冲器时为例外情况，此时，该过程不能取消。

在执行移入以提醒 CPU，消息缓冲器的内容暂时不一致时，目标消息缓冲器的 BUSY 位（CODE 字段的最低有效位）认定。

42.5.5.2 移出

移出是在有可供传输的消息时，将来自 Tx 邮箱的内容复制到 Tx SMB 的过程（见“仲裁过程”一节）。移出发生在下列条件下：

- 间隙字段的第一位
- TX 错误计数器处于 124 至 128 这一范围时，在总线关闭状态中
- 总线空闲状态中
- 等待总线空闲状态中

移出过程不是原子过程。只有 CPU 有同时访问退出总线空闲状态的存储器的优先权。在总线空闲状态下，移出过程具有最低的存储器同时访问优先权。

42.5.6 数据一致性

为了维持数据一致性并确保 FlexCAN 能正常运行，CPU 必须遵循[传输流程](#)和[接收过程](#)中描述的规则。

42.5.6.1 传输中止机制

中止机制是一种请求中止挂起传输的安全方式。反馈机制用于通知 CPU 传输是否中止或者帧是否无法中止而被传输。

中止传输必须满足两个主要条件：

- CAN_MCR[AEN]位必须已被认定
- 第一个 CPU 动作必须为将中止代码(0b1001)写入控制和状态字的 CODE 字段。

必须在更新之前先中止配置用于传输的活动 MB。如果中止代码已被写入当前正在传输的邮箱或已载入 Tx SMB 传输的邮箱，则写入操作将受阻，但传输将不会中断。但是，中止请求将被捕捉并保持挂起，直到满足以下条件之一：

- 模块丢失总线仲裁
- 传输过程中出错
- 模块处于冻结模式
- 模块进入总线关断状态
- 存在过载帧

如果未达到上述任何条件，MB 将正常传输，在 IFLAG 寄存器中置位中断标志位，并生成 CPU 中断（如果已使能）。当置位中断标志位时，将自动清除中止请求。另一方面，如果达到上述条件之一，将不会传输帧；因此，中止代码将写入 CODE 字段，在 IFLAG 中置位中断标志位，并（可选）生成 CPU 中断。

如果在开始内部传输之前 CPU 写入中止代码，则写入操作将不会受阻；因此，MB 将会更新并且中断标志位将置位。这样 CPU 只需要读取中止代码即可确保活动 MB 已安全禁用。虽然 AEN 位已被认定并且 CPU 已写入中止代码，在这种情况下，MB 将禁用并且不会中止，因为传输尚未开始。仅当中止请求被捕捉并保持挂起时才会中止一个邮箱，除非满足前面的一个条件。

中止过程可概述如下：

- CPU 检查对应的 IFLAG 并清除它（如果已认定）。
- CPU 将 0b1001 写入 C/S 字的 CODE 字段。
- CPU 等待对应的 IFLAG 指示已传输或中止帧。
- CPU 读取 CODE 字段以检查已传输(CODE=0b1000)还是已中止(CODE=0b1001)帧。
- 需要清除对应的 IFLAG，以便重新配置 MB。

42.5.6.2 邮箱禁用

禁用机制用于保护邮箱，防止 FlexCAN 内部过程更新邮箱，从而允许 CPU 能在更新邮箱之后继续依赖邮箱数据，甚至是在正常模式下。

发送邮箱禁用必须在 MCR[AEN]位解除认定时执行。

如果一个邮箱被禁用，则在重新激活之前，它不会参与仲裁过程，也不会参与匹配过程。有关如何禁用和重新激活邮箱的详细说明，请参阅[传输流程](#)和[接收过程](#)。

要禁用一个邮箱，CPU 必须将其 CODE 字段更新为 INACTIVE(0b0000 或 0b1000)。

由于用户无法将 CODE 字段更新与 FlexCAN 内部过程同步，因此，禁用操作可能会导致下列后果：

- 可能在无任何通知的情况下丢失总线上匹配被禁用 Rx 邮箱过滤标准的帧，即使存在过滤条件相同的其他邮箱
- 可能在不置位各自 IFLAG 的情况下发送含有被禁用 Tx 邮箱中的消息的帧

为了进行安全禁用并避免给 Tx 邮箱造成上述后果，CPU 必须使用传输中止机制(见[传输中止机制](#))。

禁用会自动解锁邮箱（见[邮箱锁定机制](#)）。

注

作为 Rx FIFO 一部分的消息缓冲器不能禁用。FlexCAN 不对 FIFO 区域进行写保护。当 RFEN 认定时，CPU 必须维持 FIFO 区域中数据的一致性。

42.5.6.3 邮箱锁定机制

除邮箱禁用以外，FlexCAN 还有一种针对接收过程的数据一致性机制。当 CPU 读取具有代码 FULL 或 OVERRUN 的 Rx MB 的控制和状态字时，FlexCAN 假定，CPU 要在原子操作中读取整个 MB，因此会为该 MB 置位一个内部锁定标志。在 CPU 读取自由运行定时器（全局解锁操作）时，或者在 CPU 读取另一个 MB（无论其代码为何）的控制和状态字时，该锁定将被释放。CPU 对 C/S 字进行写入操作也会解锁 MB，但不建议将此程序用于正常解锁目，因为它会取消挂起的移动，并且可能丢失已接收的消息。MB 锁定可以防止在 CPU 读取 MB 时将新帧写入该 MB。

注

锁定机制仅适用于不是 FIFO 一部分并且代码不为 INACTIVE (0b0000)或 EMPTY 的 Rx MB¹ (0b0100)。另外，Tx MB 不能锁定。

例如，假设 FIFO 被禁用，阵列中的第二和第五 MB 配置为同一 ID，并且 FlexCAN 已经接收消息并将消息存进了两个 MB 中。现在假设 CPU 决定读取 5 号 MB，并同时读取将到达的 ID 相同的另一条消息。当 CPU 读取 5 号 MB 的控制和状态字时，

1. 在以前的 FlexCAN 版本中，读取 C/S 字就会锁定 MB，即使该 MB 为 EMPTY。当 IRMQ 位取反时，这种行为将得以维持。

此 MB 将被锁定。新消息到达，并且匹配算法发现，不存在“自由接收”MB，因此，它决定覆盖 5 号 MB。然而，该 MB 已锁定，因此新消息不能写进那里。新消息将停留在 Rx SMB 中，等待 MB 解锁，只有在该 MB 解锁时，它才会写进 MB。

如果 MB 未及时解锁，并且另一条同 ID 的新消息到达，则该新消息会覆盖 Rx SMB 中的消息，而且 MB 的 CODE 字段中或错误和状态寄存器中都无消息已丢失的指示信息。

当消息从 Rx SMB 移入 MB 时，CODE 字段中的 BUSY 位认定。如果 CPU 读取控制和状态字并发现 BUSY 位置位，则在 BUSY 位取反之前，它不会访问该 MB。

注

如果 BUSY 位认定或者 MB 为空，则读取控制和状态字不会锁定 MB。

禁用优先于锁定。如果 CPU 禁用锁定的 Rx MB，则其锁定状态取反，并且在当前匹配过程中，MB 将被标记为无效。Rx SMB 中的任何挂起消息不会再传输至 MB。当 CPU 读取自由运行定时器寄存器（见“自由运行定时器寄存器(CAN_TIMER)”一节）或者另一 MB 的 C/S 字时，MB 会锁定。

锁定和解锁机制在正常和冻结模式下具有相同的功能。

在正常或冻结模式下，一次解锁会导致挂起消息移入。然而，如果解锁在低功耗模式（见[工作模式](#)）期间发生，移入操作将延迟；仅当模块恢复为正常或冻结模式时，移入操作才会发生。

42.5.7 Rx FIFO

Rx FIFO 为只接收且通过认定 CAN_MCR[RFEN]位使能。此位的复位值为零，以便维持与无 FIFO 特性的前期版本模块的软件向后兼容性。

FIFO 的深度为 6 条消息。FIFO 结构占有的存储器区域（消息缓冲器和 FIFO 引擎）见[Rx FIFO 结构](#)。通过在 FIFO 输出端重复读取消息缓冲器结构，CPU 可以按接收顺序，按序读取接收的消息。

CAN_IFLAG1[BUF5I] (Rx FIFO 中的可用帧) 在至少有一个帧可从 FIFO 读取时认定。如果它由对应的掩码位使能，则会生成一个中断。在接收到中断时，CPU 可以读取消息（以消息缓冲器访问 FIFO 的输出）和 CAN_RXFIR 寄存器，然后清除中断。如果 FIFO 中有更多的消息，则清除中断的行为会用下一条消息更新 FIFO 的输出，并用该消息的属性更新 CAN_RXFIR，向 CPU 重发中断。否则，该标志将保持取反。FIFO 的输出仅在 CAN_IFLAG1[BUF5I]认定时有效。

CAN_IFLAG1[BUF6I] (Rx FIFO 警告) 在 Rx FIFO 中未读取的消息数量因收到新消息而从 4 增至 5 时认定, 这表示 Rx FIFO 几乎已满。该标志将在 CPU 将其清零之前保持认定。

CAN_IFLAG1[BUF7I] (Rx FIFO 溢出) 在一条输入消息因 Rx FIFO 已满而丢失时认定。注意, 当 Rx FIFO 已满且消息被邮箱捕捉到时, 此标志不会认定。该标志将在 CPU 将其清零之前保持认定。

清零这三个标志之一不会影响另外两个的状态。

如果 IFLAG 位认定且对应的掩码位也认定, 则会生成一个中断。

提供了一种强大的过滤方案, 只接受面向目标应用的帧, 从而降低中断处理工作负载。指定过滤标准的方法为: 根据 CAN_CTRL2[RFFN]的设置, 配置一张表, 其中最多可以包括可以配置为下列格式之一的 128 个 32 位寄存器 (另见 [Rx FIFO 结构](#)):

- 格式 A: 128 个 IDAF (扩展或标准 ID, 包括 IDE 和 RTR)
- 格式 B: 256 个 IDAF (标准 ID 或扩展型 14 位 IDE 片, 包括 IDE 和 RTR)
- 格式 C: 512 个 IDAF (标准或扩展型 8 位 ID 片)

注

选定格式适用于滤波器表中的所有条目。不能混淆表中的格式。

FIFO 中的每个可用帧都有一个对应的 IDHIT (标识符接受滤波器命中指示器), 可从 C/S 字 IDHIT 字段读取, 如 Rx FIFO 结构描述所示。CPU 取得此信息的另一种方式是访问 CAN_RXFIR 寄存器。CAN_RXFIR[IDHIT]字段指向 FIFO 输出端的消息, 并且在 CAN_IFLAG1[BUF5I]标志认定时有效。CAN_RXFIR 寄存器只得在清零标志之前读取, 这样可以确保该信息指向的是 FIFO 中的正确帧。

根据 CAN_CTRL2[RFFN]的设置, 最多有 16 个滤波器表元素会单独受到单独掩码寄存器(CAN_RXIMRx)的影响, 这样就可以定义非常强大的过滤标准。如果 CAN_MCR[IRMQ]位取反, 则 FIFO 滤波器表会受到 CAN_RXFGMASK 的影响。

42.5.7.1 DMA 工作模式下的 Rx FIFO

只接收 FIFO 可以支持 DMA, 此特性通过同时认定 CAN_MCR[RFEN]和 CAN_MCR[DMA]位使能。CAN_MCR[DMA]位的复位值为零, 以便维持与无 DMA 特性的前期版本模块的向后兼容性。

DMA 控制器可以读取接收的消息, 方法是在 0x80-0x8C 地址范围, 在 FIFO 输出端口读取消息缓冲器结构。

当 CAN_MCR[DMA]认定时，CPU 不得访问 FIFO 输出端口地址范围。在使能 CAN_MCR[DMA]之前，CPU 必须处理在 Rx FIFO 区域中认定的 IFLAG。否则，这些 IFLAG 可能会显示，该 FIFO 有数据需要处理，并错误地生成 DMA 请求。在禁用 CAN_MCR[DMA]之前，CPU 必须执行 FIFO 清零操作。

CAN_IFLAG1[BUF5I] (Rx FIFO 中的可用帧) 在至少有一个帧可从 FIFO 读取时认定，结果同时生成 DMA 请求。在接收到请求时，DMA 控制器可以读取消息 (以消息缓冲器访问 FIFO 的输出)。DMA 读取过程必须通过读取地址 0x8C 结束，结果会清零 CAN_IFLAG1[BUF5I]并以下一条消息更新 FIFO 输出 (如果 FIFO 不为空)，并以新消息的属性更新 CAN_RXFIR 寄存器。如果 FIFO 存储着更多的消息，CAN_IFLAG1[BUF5I]会重新认定，并再发一个 DMA 请求。否则，该标志将保持取反。

注

在 DMA 完成 FIFO 读取操作之后，CAN_RXFIR 寄存器内容不能读取。在地址 0x080 处的 C/S 字中也有 IDHIT 信息 (见 [Rx FIFO 结构](#))。

当 DMA 特性被使能时，不会使用 CAN_IFLAG1[BUF6I]和 CAN_IFLAG1[BUF7I]。

当 FlexCAN 采用 DMA 功能时，CPU 不会收到任何 Rx FIFO 中断，并且不得清零相关的 IFLAG。另外，不得将相关的 IMASK 用于屏蔽 DMA 请求的生成。

42.5.7.2 FIFO 清零操作

当 CAN_MCR[RFEN]认定时，FIFO 清零操作用于清空 FIFO 的内容。CAN_MCR[RFEN]认定时，FIFO 清零发生在 CPU 将 1 写入 CAN_IFLAG1[BUF0I]时。此操作仅可在冻结模式下执行，因为在其他模式下它被硬件禁用。此操作不会清零 FIFO IFLAG，结果，在执行 FIFO 清零任务之前，CPU 必须处理所有 FIFO IFLAG。

当 Rx FIFO 采用 DMA 特性时，FIFO 清零操作会清除 CAN_IFLAG1[BUF5I]，并取消 DMA 请求。

警告

FIFO 清零操作不会清零 IFLAG, CAN_MCR[DMA]认定时除外，此时，只会清零 CAN_IFLAG1[BUF5I]。

42.5.8 CAN 协议相关特性

本节介绍 CAN 协议相关特性。

42.5.8.1 远程帧

远程帧是一种特殊的帧。用户可以将一个邮箱配置为远程请求帧，其方法是将 RTR 位置为'1'，并将邮箱配置为发送。在远程请求帧成功发送之后，邮箱将成为一个接收消息缓冲器，其 ID 保持不变。

当远程请求帧被 FlexCAN 收到时，可能有三种处理方式，具体取决于远程请求存储 (CTRL2[RRS]) 和 Rx FIFO 使能 (MCR[RFEN]) 位：

- 如果 RRS 取反，会用 CODE 字段 0b1010，将帧的 ID 与发送消息缓冲器的 ID 进行比较。如果有匹配的 ID，则会发送此邮箱帧。注意，如果匹配的邮箱的 RTR 位置位，则 FlexCAN 会发送一个远程帧作为响应。接收到的远程请求帧不会存储在接收缓冲器中。只是用于触发帧发送，以此作为响应。在远程帧匹配过程中不会使用掩码寄存器，已接收的输入帧的所有 ID 位 (RTR 除外) 都应匹配。在收到远程请求帧且匹配一个邮箱的情况下，该消息缓冲器会立即进入内部仲裁过程，但会被视为一个正常的 Tx 邮箱，不享有更高的优先级。此帧的数据长度独立于启动传输的远程帧中的 DLC 字段。
- 如果 RRS 认定，则会用 CODE 字段 0b0100、0b0010 或 0b0110，将帧 ID 与接收邮箱的 ID 进行比较。如果有匹配的 ID，则此邮箱会以与数据帧相同的方式存储远程帧。不会生成自动远程响应帧。匹配过程中要用到掩码寄存器。
- 如果 RFEN 认定，则 FlexCAN 不会为匹配 FIFO 滤波标准的远程请求帧生成自动响应。如果远程帧与其中一个目标 ID 相匹配，则此远程帧会存储在 FIFO 中，并提供给 CPU。注意，对于滤波格式 A 和 B，可以选择是否接受远程帧。对于格式 C，始终都会接受远程帧 (如果与 ID 相匹配)。远程请求帧被视为正常帧，并会在发生正常接收事件且 FIFO 已满的情况下，生成 FIFO 溢出。

42.5.8.2 过载帧

FlexCAN 确实会因在 CAN 总线上检测到下列条件而发送过载帧：

- 在间隙的第一/第二位中检测到显性位
- 在帧结束字段 (Rx 帧) 的第 7 位 (最后位) 中检测到显性位
- 在错误帧定界符或过载帧定界符的第 8 位 (最后位) 检测到显性位

42.5.8.3 时间戳

在 CAN 总线上的标识符字段开始处对自由运行定时器的值进行采样，并且该值将存储于 TIME STAMP 字段的“移入”结束处，以提供与时间相关的网络行为。

自由运行定时器将由 FlexCAN 位时钟计时，它可定义 CAN 总线上的波特率。在消息传送/接收过程中，接收或传送每个位时，它将会递增 1。当总线上无消息时，它将使用先前设定的波特率计数。

在禁用、等待、停止和冻结模式下，自由运行定时器不会递增。它可在收到特定帧时复位，从而使能网络时间同步。请参见控制 1 寄存器(CAN_CTRL1)中的 TSYN 描述。

42.5.8.4 协议时序

下图所示为馈入 CAN 协议引擎(PE)子模块的时钟生成电路的结构。CAN_CTRL1 寄存器中的时钟源位 CLKSRC 定义内部时钟是连接到晶振（振荡器时钟）还是外设时钟。为了确保运行的可靠性，应在模块处于禁用模式（模块配置寄存器中的 MDIS 位置位）时选择时钟源。

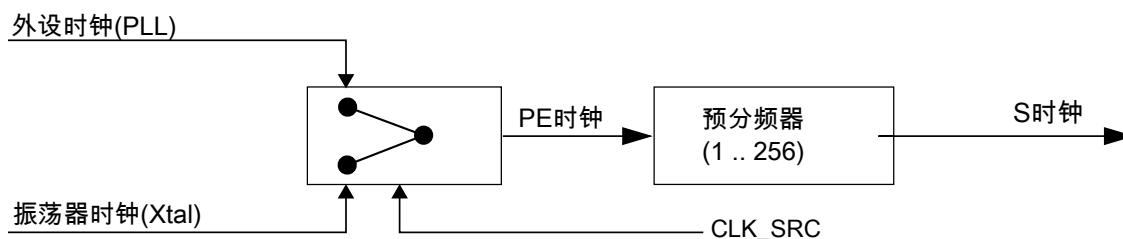


图 42-101. CAN 引擎计时方案

每当 CAN 总线时序中要求严格的容差（最高 0.1%）时，都应选择此振荡器时钟。晶振时钟比外设时钟有着更出色的抖动性能。

FlexCAN 模块支持以多种方式来设定 CAN 协议要求的位时序参数。控制 1 寄存器 (CAN_CTRL1) 有多个字段，可用于控制位时序参数：PRES DIV、PROPSEG、PSEG1、PSEG2 和 RJW。

CAN 位时序寄存器(CAN_CBT)可扩展 CAN_CTRL1 中 CAN 位时序变量的范围。

PRES DIV 字段（及其扩展范围 EPRES DIV 数据阶段位的 FDPRES DIV）定义生成串行时钟 (S_{clock}) 的预分频器值（见下面的等式），而该串行时钟的周期则定义用于比较 CAN 波形的“时间量子”。时钟量子(T_q)是 CAN 引擎处理的原子时间单位。

$$f_{Tq} = \left(\frac{f_{CANCLK}}{\text{预分频值}} \right)$$

比特率（定义 CAN 消息的接收或发送速率）由以下公式计算得到：

$$\text{比特率} = \frac{f_{Tq}}{\text{时间量子数量}}$$

位时间细分为三段¹（见图 42-102 和表 42-115）：

- SYNC_SEG: 该区段有一个固定长度的时间量子。在该部分内预期发生信号边沿
- 时间段 1: 此段包括 CAN 标准的传播段和相位段 1。可通过置位 CAN_CTRL1 寄存器的 PROPSEG 和 PSEG1 字段来配置，这样，它们的和（加 2）就在 4 至 16 时间量子之内。当 CAN_CBT[BTF] 位置位时，FlexCAN 会使用来自 CAN_CBT 寄存器的 EPROPSEG 和 EPSEG1 字段，这样，它们的和（加 2）就在 2 至 96 时间量子之内。
- 时间段 2: 该区段代表 CAN 标准的相位段 2。可通过置位 CAN_CTRL1 寄存器（加 1）的 PSEG2 字段，将其配置为 2 至 8 个时间量子长。当 CAN_CBT[BTF] 位置位时，FlexCAN 会使用 CAN_CBT 寄存器的 EPSEG2 字段，这样，其值（加 1）就在 2 至 32 时间量子之内。时间段 2 不能小于信息处理时间（IPT），在 FlexCAN 中，其值为 2 时间量子。

注

由上述时间段定义的位时间不得小于 5 时间量子。要计算位时间，请使用信息处理时间(IPT) 2，这是在 FlexCAN 模块中实现的值。

1. 有关基本概念的进一步说明，请参阅 ISO 11898-1。另请参阅 CAN 2.0A/B 协议规格，了解有关位时序的信息。

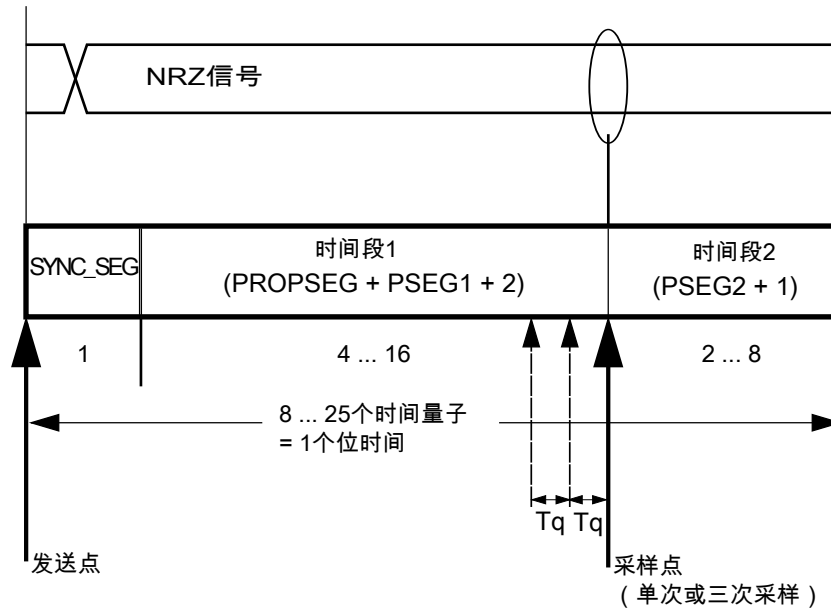


图 42-102. 位时间内的区段（比如在经典 CAN 格式下使用 CAN_CTRL1 位时序变量）

表 42-115. 时间段句法

句法	说明
SYNC_SEG	系统期望在该周期内在总线上发生转换。
TSEG1	对应于 PROPSEG 和 PSEG1 之和。
TSEG2	对应于 PSEG2 的值。
发送点	处于发送模式的节点于此时将新值传输到 CAN 总线。
采样点	节点在此点对总线采样。如果选择每位三样本选项，那么该点标志第三个样本的位置。

下表给出了一些例子, 说明了经典 CAN 格式消息的 CAN 合规区段设置(Bosch CAN 2.0B)。

表 42-116. 符合 Bosch CAN 2.0B 标准的位时间段设置

时间段 1	时间段 2	再同步跳转宽度
5 .. 10	2	1 .. 2
4 .. 11	3	1 .. 3
5 .. 12	4	1 .. 4
6 .. 13	5	1 .. 4
7 .. 14	6	1 .. 4
8 .. 15	7	1 .. 4
9 .. 16	8	1 .. 4

注

用户必须确保位时间设置符合 CAN 协议标准(ISO 11898-1)的要求。

每当将 CAN 位用作时间持续长度的一种称量指标时 (即估算 CAN 位事件在消息中的出现), 一个 CAN 位中的外设时钟数量(NumClkBit)可按以下等式计算:

$$NCCP = \frac{f_{SYS} \times [1 + (PSEG1 + 1) + (PSEG2 + 1) + (PROPSEG + 1)] \times (PRES DIV + 1)}{f_{CANCLK}}$$

其中:

- NumClkBit 为一个 CAN 位中的外设时钟数量;
- f_{CANCLK} 为协议引擎(PE)时钟 (见图"CAN 引擎计时方案"), 单位为 Hz;
- f_{SYS} 为系统(CHI)时钟的工作频率, 单位为 Hz;
- PSEG1 为 CAN_CTRL1[PSEG1]字段中的值;
- PSEG2 为 CAN_CTRL1[PSEG2]字段中的值;
- PROPSEG 为 CAN_CTRL1[PROPSEG]字段中的值;
- PRES DIV 为 CAN_CTRL1[PRES DIV]字段中的值。

上式也可用于 CAN 位时序寄存器(CAN_CBT)。

例如, 180 CAN 位 = (180 x NumClkBit)外设时钟周期。

42.5.8.5 仲裁和匹配时序

在正常接收和传输中, 匹配、仲裁、移入和移出过程都是在 CAN 帧的某些时间窗口中执行的, 如下图所示。

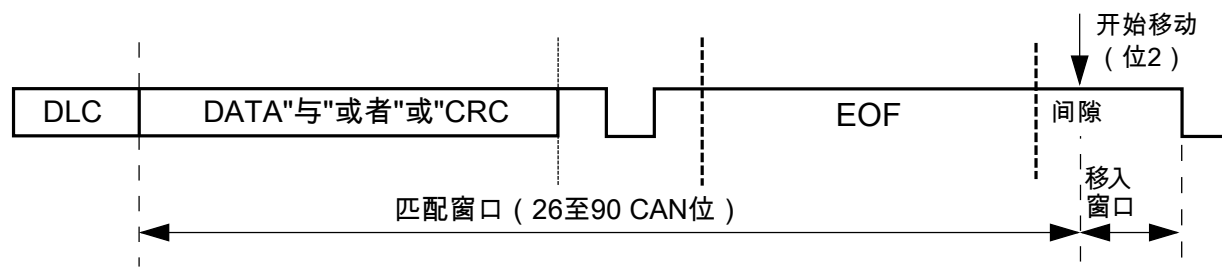


图 42-103. 匹配移入时间窗口

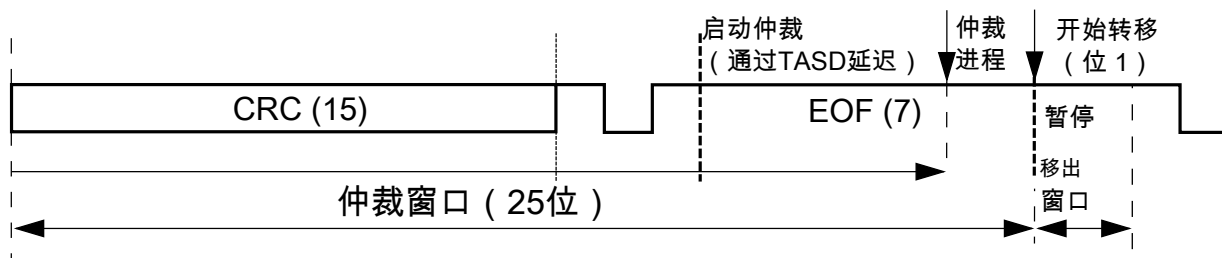


图 42-104. 仲裁和移出时间窗口

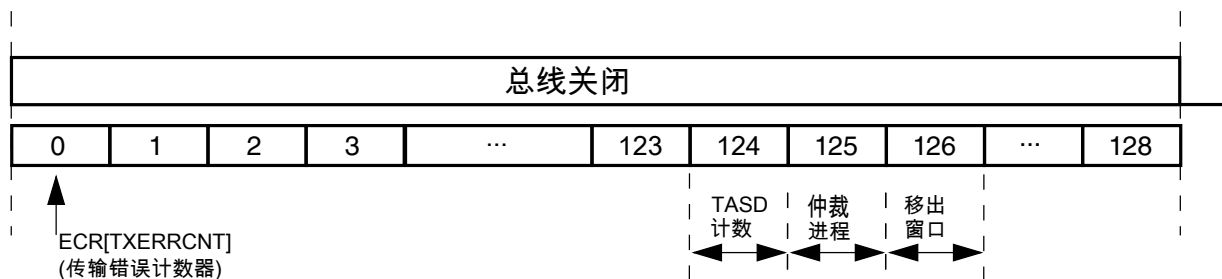


图 42-105. 总线关闭结束时的仲裁和移出时间窗口

注

在前面的图中，匹配和仲裁时序不包括因 CPU 或其他内部 FlexCAN 子块同时访问存储器而导致的延迟。

42.5.8.6 Tx 仲裁开始延迟

控制 2 寄存器(CAN_CTRL2[TASD])中的 Tx 仲裁开始延迟(TASD)位域为变量,表示 FlexCAN 从当前帧的第一个 CRC 位域延迟 Tx 仲裁流程开始点所使用的 CAN 位数。此变量仅可在冻结模式下写入，因为在其他模式下它被硬件禁用。

传输性能受 CPU 在内部仲裁流程结束后重新配置传输用消息缓冲区(MB)能力的影响，其中，FlexCAN 将找到用于传输的最佳 MB（参见[仲裁过程](#)）。如果仲裁在第一个间歇位域之前过早结束，则 CPU 可能会重新配置部分 Tx MB，并且最佳 MB 可能不再是要传输的最佳候选 MB。

TASD 对于优化传输性能非常有用，它可如下图所示根据以下因素定义仲裁开始点：

- 外设至振荡器的时钟比率
- 决定 CAN 比特率的 CAN 位时间变量
- 匹配和仲裁流程使用的消息缓冲区(MB)数。

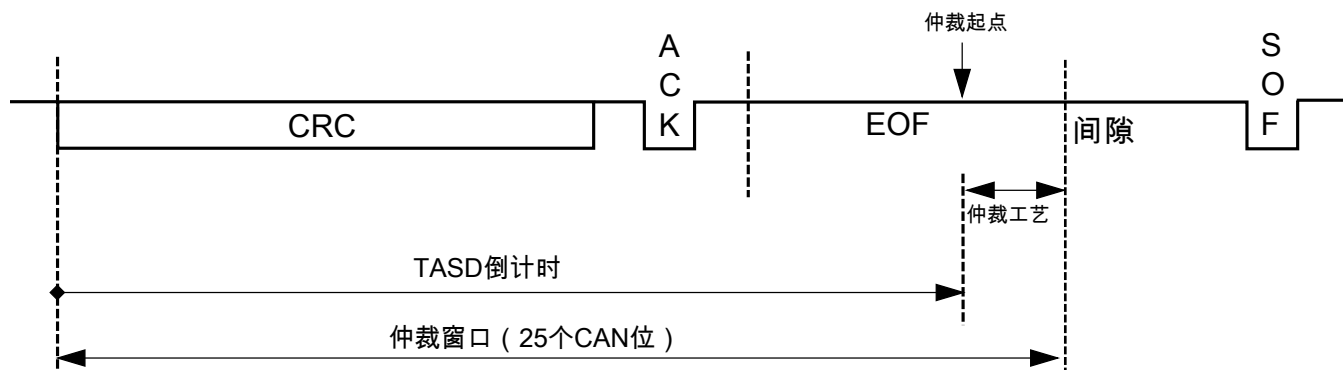


图 42-106. 最优 Tx 仲裁开始点

仲裁流程的持续时间（按 CAN 位计算）与可用的 MB 数和 CAN 比特率成正比，并与外设时钟频率成反比。

最优仲裁时间指的是在当前帧的第一个间隙位域之前扫描最后一个 MB 时。例如，如果存在多个 MB 且外设/振荡器时钟比较高且 CAN 波特率较低，则可将仲裁靠近帧结束位置，从而为其开始点增加延迟，反之亦然。

如果 TASD 置位为 0 且未延迟仲裁开始，则保留的仲裁时间较长。另一方面，如果 TASD 接近 24，则 CPU 可稍后配置 Tx MB，且保留的仲裁时间较短。如果保留的仲裁时间过短，则 FlexCAN 可能无法及时找到要传输的最佳 MB，并提供针对 CAN 总线上外部节点仲裁的最佳时机。

最优 TASD 值计算方法如下：

$$NCCP = \frac{f_{\text{SYS}} \times [1 + (\text{PSEG1} + 1) + (\text{PSEG2} + 1) + (\text{PROPSEG} + 1)] \times (\text{PRES DIV} + 1)}{f_{\text{CANCLK}}}$$

其中：

- MAXMB 为 CAN_CTRL1[MAXMB] 字段中的值
- f_{CANCLK} 为振荡器时钟频率(Hz)
- f_{SYS} 为外设时钟频率(Hz)
- RFEN 为 CAN_CTRL1[RFEN] 位中的值
- RFFN 为 CAN_CTRL2[RFFN] 字段中的值
- PSEG1 为 CAN_CTRL1[PSEG1] 字段中的值
- PSEG2 为 CAN_CTRL1[PSEG2] 字段中的值
- PROPSEG 为 CAN_CTRL1[PROPSEG] 字段中的值
- PRES DIV 为 CAN_CTRL1[PRES DIV] 字段中的值

另请参见[协议时序](#)”，了解更多详情。

42.5.9 时钟域和限制

FlexCAN 模块有两个相互异步的时钟域:

- 总线域馈入控制主机接口 (CHI) 子模块, 源于外设时钟。
- 振荡器域馈入 CAN 协议引擎 (PE) 子模块, 直接源于一个晶振时钟, 这样, 即可在 CAN 总线上实现超低抖动性能。

当 CAN_CTRL1[CLKSRC]位置位时, 会发生同步操作, 因为两个域都连接至外设时钟 (在外设时钟与振荡器时钟之间形成 1:1 的比率)。

当两个域连接至不同频率"与"或者"或"相位的时钟时, 对两个时钟域之间的频率关系存在限制。在异步操作情况下, 总线域时钟必须始终大于振荡器域时钟频率。

注

不允许在外设时钟与振荡器时钟之间按 1:1 的比率执行异步操作。

在进行匹配和仲裁时, FlexCAN 需要在一个 CAN 帧的时间槽之内扫描整个消息缓冲器存储器, 该 CAN 帧由若干个 CAN 位构成。为了有足够的时间来完成该操作, 必须遵循以下要求:

- 外设时钟频率不得小于振荡器时钟频率。
- 对于 16 个邮箱, 每个 CAN 位的外设时钟的最低数量为 16

每个 CAN 位的外设时钟的最低数量决定着预期 CAN 比特率的最低外设时钟频率。CAN 比特率取决于一个 CAN 位中的时间量子数, 该数量可通过调节控制 1 寄存器 (CAN_CTRL1) 或 CAN 位时间寄存器 (CAN_CBT) 中含有的一个或多个位时序值来定义。时间量子 (Tq) 的定义见[协议时序](#)。每个 CAN 位的最低时间量子数必须为 8; 因此, 振荡器时钟频率应至少为 CAN 比特率的 8 倍。

42.5.10 工作模式详情

FlexCAN 模块有功能模式和低功耗模式。有关全部工作模式的简介, 请参阅[工作模式](#)。下面的各小节将详细描述冻结模式和低功耗模式的功能。

警告

FlexCAN 不支持 CAN 总线上的“永久显性”故障。如果低功耗请求或冻结模式请求在“永久显性”期间提出, 则对应的确认将永远无法置位。

42.5.10.1 冻结模式

此模式由 CPU 通过认定 CAN_MCR 寄存器中的 HALT 位来请求，或者在芯片进入调试模式时请求。在两种情况下，同时还需要 CAN_MCR 寄存器中的 FRZ 位认定且模块不处于低功耗模式。

通过 FlexCAN 认定同一寄存器中的 FRZ_ACK 位来取得确认。仅当符合请求和应答条件时，CPU 才将 FlexCAN 视作处于冻结模式。

当请求冻结模式时，FlexCAN 会执行下列操作：

- 等待进入间隙、无效错误、总线关闭或空闲状态
- 等待所有内部活动（如仲裁、匹配、移入和移出）完成。挂起的移入操作不会防止进入冻结模式。
- 忽略 Rx 输入引脚并将 Tx 引脚驱动为隐性状态
- 停止预分频器，从而停止所有 CAN 协议活动
- 允许对错误计数器寄存器进行写入访问（其在其他模式下均为只读）
- 置位 NOT_RDY 中的 FRZ_ACK 和 CAN_MCR 位

在请求冻结模式之后，用户必须在执行任何其他操作之前，等待 FRZ_ACK 位在 CAN_MCR 中认定，否则，FlexCAN 可能出现不可预测的情况。在冻结模式下，所有存储器映射寄存器均可访问，可以读取但不可写入的 CAN_CTRL1[CLKSRC]位除外。

冻结模式的退出以下列方式之一实现：

- CPU 取反 CAN_MCR 寄存器中的 FRZ 位
- 芯片退出调试模式且/或 HALT 位取反

FRZ_ACK 位在协议引擎识别出冻结请求取反之后取反。当退出冻结模式之后，FlexCAN 通过等待 11 个连续隐性位，尝试与 CAN 总线再同步。

42.5.10.2 模块禁用模式

此低功耗模式通常用于临时禁用完整的 FlexCAN 数据块，而不消耗电能。此模式由 CPU 通过认定 CAN_MCR[MDIS]位来请求，确认则由 FlexCAN 通过认定 CAN_MCR[LPMACK]位来取得。仅当符合请求和确认条件时，CPU 才将 FlexCAN 视作处于禁用模式。

如果模块在冻结模式中被禁用，则会请求禁用 PE 和 CHI 子模块的时钟，置位 LPMACK 位并取反 FRZACK 位。

如果在发送或接收过程中禁用模块，则 FlexCAN 将进行以下操作：

- 等待进入或总线关闭状态，否则将会等待第三个间隙位并将其置于隐性状态
- 等待所有内部活动（如仲裁、匹配、移入和移出）完成。挂起的移入活动不在考虑范围之内。
- 忽略其 Rx 输入引脚并将其 Tx 引脚驱动为隐性状态
- 关闭 PE 和 CHI 子模块的时钟
- 置位中的 NOTRDY 和 LPMACK 位 CAN_MCR

总线接口单元继续工作，使 CPU 可以访问存储器映射寄存器，Rx 邮箱全局掩码寄存器、Rx 缓冲器 14 掩码寄存器、Rx 缓冲器 15 掩码寄存器、Rx FIFO 全局掩码寄存器除外。当模块处于禁用模式时，不能访问 Rx FIFO 信息寄存器、消息缓冲器、Rx 单独掩码寄存器和 RAM 中的保留字。CPU 取反 MDIS 位即可退出此模式，结果会导致 FlexCAN 请求在 CAN 协议引擎识别出 CPU 请求的禁用模式取反之后，恢复时钟并取反 LPMACK 位。

42.5.10.3 休眠模式

这是一种系统低功耗模式，其中，CPU 总线保持运行，并将全局休眠模式请求发送至所有外设，要求它们进入低功耗模式。当全局请求休眠模式时，需要先前已经认定 CAN_MCR 寄存器中的 DOZE 位，这样才能触发休眠模式。通过 FlexCAN 认定同一寄存器中的 LPMACK 位来取得确认。仅当符合请求和确认条件时，CPU 才将 FlexCAN 视作处于休眠模式。

如果在冻结模式中触发休眠模式，则 FlexCAN 会请求关闭 PE 和 CHI 子模块的时钟，置位 LPMACK 位并取反 FRZACK 位。如果在发送或接收过程中触发休眠模式，则 FlexCAN 将进行以下操作：

- 等待进入闲置或总线关闭状态，否则将会等待第三个间歇位并将其置于隐性状态
- 等待所有内部活动（如仲裁、匹配、移入和移出）完成。挂起的移入活动不在考虑范围之内。
- 忽略其 Rx 输入引脚并将其 Tx 引脚驱动为隐性状态
- 关闭 PE 和 CHI 子模块的时钟
- 置位中的 NOTRDY 和 LPMACK 位 CAN_MCR

总线接口单元继续工作，使 CPU 可以访问存储器映射寄存器，Rx 邮箱全局掩码寄存器、Rx 缓冲器 14 掩码寄存器、Rx 缓冲器 15 掩码寄存器、Rx FIFO 全局掩码寄存器除外。当模块处于休眠模式时，不能访问 Rx FIFO 信息寄存器、消息缓冲器、Rx 单独掩码寄存器和 RAM 中的保留字。

休眠模式的退出以下列方式之一实现：

- CPU 移除休眠模式请求
- CPU 取反 CAN_MCR 寄存器的 DOZE 位
- 自我唤醒机制

在自我唤醒机制下，如果 CAN_MCR 寄存器中的 SLFWAK 位在 FlexCAN 进入休眠模式时置位，则当在 CAN 总线上检测到隐性至显性转换时，FlexCAN 会取反 DOZE 位，请求恢复其时钟并在 CAN 协议引擎识别出休眠模式请求取反之后取反 LPMACK。同时还会置位 ESR 寄存器中的 WAKINT 位，并且如果由 CAN_MCR 中的 WAKMSK 位使能，则会向 CPU 生成一个唤醒中断。随后 FlexCAN 等待 11 个连续的隐性位，以便与 CAN 总线同步。因此，它不会接收到将其唤醒的帧。下表详细介绍了 SLFWAK 和 WAKMSK 对从休眠模式唤醒的影响。

表 42-117. 从休眠模式唤醒

SLFWAK	WAKINT	WAKMSK	FlexCAN 时钟已使能	生成的唤醒中断
0	-	-	否	否
0	-	-	否	否
1	0	0	否	否
1	0	1	否	否
1	1	0	是	否
1	1	1	是	是

对 CAN 总线活动的敏感度是可以修改的，方法是：在休眠模式时，将低通滤波器功能应用到 Rx CAN 输入线路。请参见模块配置寄存器 (CAN_MCR) 说明中的 WAKSRC 位。该特性可用于防止 FlexCAN 因 CAN 总线线路上的短时毛刺而唤醒。这类毛刺可能由嘈杂环境中的电磁干扰而引起。

42.5.10.4 Stop 模式

它是一种系统低功耗模式，在此模式下，所有芯片时钟停止，以实现最大程度节能。Stop 模式由 CPU 全局请求，并通过 FlexCAN 认定停止应答信号来获得应答。仅当符合请求和应答条件时，CPU 才将 FlexCAN 视作处于 Stop 模式。

如果 FlexCAN 在冻结模式下接收到全局 Stop 模式请求，则其将置位 LPMACK 位，否定 FRZACK 位并向 CPU 发送停止应答信号，以便全部关闭时钟。

如果在发送或接收过程中请求 Stop 模式，则 FlexCAN 将进行以下操作：

- 等待进入闲置或总线关闭状态，否则将会等待第三个间歇位并将其置于隐性状态
- 等待所有内部活动（如仲裁、匹配、移入和移出）完成。挂起的移入活动不在考虑范围之内。
- 忽略其 Rx 输入引脚并将其 Tx 引脚驱动为隐性状态
- 置位 CAN_MCR 中的 NOTRDY 和 LPMACK 位
- 向 CPU 发送停止应答信号，以便其全部关闭时钟

当 CPU 恢复时钟并删除 Stop 模式请求时，将会退出 Stop 模式。这是自动唤醒机制所致。

在自动唤醒机制下，如果 CAN_MCR 寄存器中的 SLFWAK 位在 FlexCAN 进入 Stop 模式时置位，则在检测到 CAN 总线上存在由隐性到显性的转换时，FlexCAN 将置位 CAN_ESR 寄存器中的 WAKINT 位，并且若通过 CAN_MCR 中的 WAKMSK 位使能，将会生成 CPU 唤醒中断。接收到中断时，CPU 应恢复时钟并删除 Stop 模式请求。随后 FlexCAN 等待 11 个连续的隐性位，以便与 CAN 总线同步。因此，它不会接收到将其唤醒的帧。下表详细介绍了从 Stop 模式唤醒时 SLFWAK 和 WAKMSK 的影响。请注意，仅当认定两个位时才会从 Stop 模式唤醒。

CAN 协议引擎识别出否定 Stop 模式请求后，FlexCAN 将否定 LPMACK 位。随后 FlexCAN 等待 11 个连续的隐性位，以便与 CAN 总线同步。因此，它不会接收到将其唤醒的帧。

表 42-118. 从 Stop 模式唤醒

SLFWAK	WAKINT	WAKMSK	芯片 时钟已使能	生成的唤醒中断
0	-	-	否	否
0	-	-	否	否
1	0	0	否	否
1	0	1	否	否
1	1	0	否	否
1	1	1	是	是

对 CAN 总线活动的敏感度是可以修改的，方法是：在 Stop 模式时，将低通滤波器功能应用到 Rx CAN 输入线路。请参见模块配置寄存器 (CAN_MCR) 说明中的 WAKSRC 位。该特性可用于防止 FlexCAN 因 CAN 总线线路上的短时毛刺而唤醒。这类毛刺可能由嘈杂环境中的电磁干扰而引起。

42.5.11 中断

此模式有多种中断源：因消息缓冲器导致的中断以及因来自 MB 且经 OR 运算的中断的中断、总线关闭、总线关闭完成、错误、唤醒、Tx 警告和 Rx 警告。

如果其对应的 IMASK 位置位，这些消息缓冲器中的每一个都可能是中断源。对于特定缓冲器来说，Tx 和 Rx 中断并无区别，其假设是缓冲器是针对发送或接收操作而初始化的。这些缓冲器中的每一个都在 CAN_IFLAG 寄存器中有一个分配的标志位。此位在对应的缓冲器成功完成传输时置位，并在 CPU 将其写为 1 时清零（除非同时生成另一个中断）。

注

必须保证 CPU 只清除造成当前中断的位。因此，位操作指令(BSET)不得用于清零中断标志。这些指令可能会意外清零在进入当前中断服务程序后置位的中断标志。

如果 Rx FIFO 被使能(CAN_MCR[RFEN] = 1)且 DMA 被禁用(CAN_MCR[DMA] = 0)，则对应于 MB 0 至 7 的中断拥有不同的意义。CAN_IFLAG1 寄存器的位 7 变成 "FIFO 溢出"标志；位 6 变成 "FIFO 警告"标志，位 5 变成 "FIFO 中的可用帧"标志，并且位 4-0 未使用。有关更多信息，请参阅关于中断标志 1 寄存器(CAN_IFLAG1)的说明。

如果 Rx FIFO 和 DMA 均使能 (CAN_MCR[RFEN]且 CAN_MCR[DMA] = 1)，FlexCAN 不会生成任何 FIFO 中断。CAN_IFLAG1 寄存器的位 5 仍然指示 "FIFO 中的可用帧"并生成 DMA 请求。位 7、6、4-0 未使用。

对于组合中断（其中，多个 MB 中断源通过 OR 运算组合起来），当任意关联 MB（或 FIFO）生成中断时，则产生中断请求。在这种情况下，CPU 必须读取 CAN_IFLAG 寄存器，以确定是哪个 MB 或 FIFO 源导致了中断。

总线关闭、总线关闭完成、错误、唤醒、Tx 警告和 Rx 警告的中断源会像 MB 中断源一样，生成中断，并且可以从 CAN_ESR1 寄存器读取。总线关闭、错误、Tx 警告和 Rx 警告中断掩码位位于 CAN_CTRL1 寄存器中；唤醒中断掩码位位于 CAN_MCR 中。

42.5.12 总线接口

CPU 对 FlexCAN 寄存器的访问操作须遵循以下规则：

- 对管理员寄存器的非法的读取和写入访问（在“模块存储器映射”表中，管理员模式的寄存器用 S/U 或仅 S 标识）会导致访问错误。

- 对已实现保留地址空间的读取和写入访问会导致访问错误。
- 对只读寄存器的写操作会导致访问错误。如果其中至少一位不为只读，则不会发出访问错误。对位置或其部分位的写入许可可能根据工作模式或临时状态而改变。有关详情，请参见寄存器和位的说明。
- 对未实现地址空间的读取和写入访问会导致访问错误。
- 在低功耗模式下对 RAM 中位置的读取和写入访问会导致访问错误。
- 如果 CAN MCR 寄存器中 MAXMB 的配置值小于可用 MB 数量，则可将未使用的存储器空间作为通用 RAM 空间。注意，RAM 中的保留字不能使用。例如，假设 FlexCAN 的 RAM 最多可以支持 16 个 MB，CAN_CTRL2[RFFN]为 0x0 且 CAN_MCR[MAXMB]配置为零。这种情况下，最大 MB 数成为一。RAM 开始于 0x0080，从 0x0080 至 0x008F 的空间由一个 MB 使用。从 0x0090 至 0x017F 的存储器空间可供使用。0x0180 和 0x087F 之间为保留空间。从 0x0880 至 0x0883 的空间由一个单独掩码位使用，掩码寄存器中的可用存储器范围为 0x0884 至 0x08BF。从 0x08C0 至 0x09DF 是供内部使用的保留字，不能用作通用 RAM。一般规则是，用于一般目的的自由存储器空间仅取决于 MAXMB。

42.6 初始化/应用信息

本节介绍关于 FlexCAN 模块初始化的说明。

42.6.1 FlexCAN 初始化序列

FlexCAN 模块可以三种方式复位：

- 芯片级和硬复位，这会异步复位所有存储器映射寄存器。
- MCR 中的 SOFTRST 位，负责同步复位部分存储器映射寄存器。请参阅表 42-2，了解哪些寄存器会受软复位的影响。
- 芯片级软复位，其作用与 MCR 中的 SOFTRST 位相同

软复位是同步的，需要跨时钟域遵循内部请求/确认程序。因此，其作用要完全体现出来可能需要一定的时间。CAN_MCR 当软复位挂起时，[SOFTRST]位保持认定，因此，软件可以轮询此位，以了解复位的完成时间。另外，当时钟在低功耗模式下关闭时，不能运用软复位。在运用软复位之前，需要退出低功耗模式并恢复时钟。

时钟源应在模块处于禁用模式时切换（见 CAN_CTRL1[CLKSRC]位）。在选定时钟源且模块使能之后（CAN_MCR[MDIS]位取反）FlexCAN 会自动进入冻结模式。在冻结模式下，FlexCAN 与 CAN 总线不同步，CAN_MCR 寄存器中的 HALT 和 FRZ 位置位，内部状态机被禁用且 CAN_MCR 寄存器中的 FRZACK 和 NOTRDY 位置位。Tx 引脚处于隐性状态，FlexCAN 不会发起 CAN 帧的任何发送或接收请求。注意，消息缓冲器和 Rx 单独掩码寄存器不受复位影响，因此，它们不会自动初始化。

对于任何配置更改/初始化，需要将 FlexCAN 置于冻结模式（见[冻结模式](#)）。以下是适用于 FlexCAN 模块的通用初始化序列：

- 初始化模块配置寄存器(CAN_MCR)
 - 置位 IRMQ 位以使能基于 MB 的独立过滤特性和接收队列特性
 - 置位 WRNEN 位以使能警告中断
 - 若需要，置位 SRXDIS 位以禁用帧自接收
 - 置位 RFEN 位以使能 Rx FIFO
 - 如果 Rx FIFO 使能且需要 DMA，则置位 DMA 位
 - 置位 AEN 位以使能中断传输机制
 - 置位 LPRIOEN 位以使能本地优先级特性
- 初始化控制 1 寄存器(CAN_CTRL1)以及可选的 CAN 位时序寄存器(CAN_CBT)。
 - 确定位时序参数：PROPSEG、PSEG1、PSEG2、RJW
 - 选择性确定位时序参数：EPROPSEG、EPSEG1、EPSEG2、ERJW
 - 对 PRES DIV 字段和可选的 EPRES DIV 字段编程以确定比特率
 - 确定内部仲裁模式（LBUF 位）
- 初始化 Message Buffer
 - 必须初始化所有 Message Buffer 的控制和状态字
 - 如果 Rx FIFO 已使能，则必须初始化 ID 滤波器表
 - 各消息缓冲器中其他条目须按要求初始化
- 初始化 Rx 单独掩码寄存器(CAN_RXIMRn)

- 在 CAN_IMASK 寄存器（用于所有 MB 中断）、CAN_MCR 寄存器（用于唤醒中断）和 CAN_CTRL1/CAN_CTRL2 寄存器（用于总线关闭和错误中断）中置位需要的中断掩码位
- 取反 CAN_MCR 中的 HALT 位

在上面列出的最后一步之后，FlexCAN 将尝试与 CAN 总线同步。

第 43 章 串行外设接口 (SPI)

43.1 此模块的芯片实现细节

43.1.1 SPI 模块配置

此器件包含两个 SPI 模块。

43.1.2 SPI 时钟

SPI 模块由内部总线时钟计时 (DSPI 将它称为系统时钟)。该模块有一个内部分频器，最少为二分频。因此，SPI 可在 1/2 总线时钟的最大频率下运行。

43.1.3 CTAR 的数目

SPI CTAR 寄存器定义不同的传输属性配置。SPI 模块支持多达 8 个 CTAR 寄存器。在所有 SPI 实例中，此器件支持两个 CTAR。

在主机模式中，CTAR 寄存器定义传输属性的组合，比如帧大小、时钟相位、时钟极性、数据位顺序、波特率以及各种延时。在从机模式中，仅使用 CTAR0，且由其位域的子集设置从机传输属性。

43.1.4 TX FIFO 大小

表 43-1. SPI transmit FIFO 大小

SPI 模块	Transmit FIFO 大小
SPI0	4
SPI1	1

43.1.5 RX FIFO 大小

在接收过程中，SPI 最多支持 16 位帧大小。

表 43-2. SPI receive FIFO 大小

SPI 模块	Receive FIFO 大小
SPI0	4
SPI1	1

43.1.6 PCS 信号数量

下表显示了每个 SPI 模块可用的外设芯片选择信号的数量。

表 43-3. SPI PCS 信号

SPI 模块	PCS 信号
SPI0	对于超过 64 个引脚的封装: SPI_PCS[5:0] 对于具有 64 个引脚的封装: SPI_PCS[4:0]
SPI1	对于超过 64 个引脚的封装: SPI_PCS[3:0] 对于具有 64 个引脚的封装: SPI_PCS[1:0]

43.1.7 低功耗模式下的 SPI 操作

在 VLPR 和 VLPW 模式下，SPI 可工作；但是，如果系统频率降低，则 SPI 操作的最大频率也会降低。在 VLPR 和 VLPW 模式下，最大 SPI_CLK 频率为 2MHz。

在停止和 VLPS 模式下，SPI 模块时钟会禁用。此模块不工作，但它已开启，所以会保持状态。

有一种方式通过 SPI 可从停止模式中唤醒，具体参见以下章节。

43.1.7.1 利用 GPIO 中断从停止模式中唤醒

下面介绍使用 GPIO 在从机模式下接收到 SPI 数据时创建唤醒的步骤：

1. 让 GPIO 中断向量指向所需的中断处理程序。

2. 使能 GPIO 输入，以在上升沿或下降沿生成一个中断（具体取决于芯片选择信号的极性）。
3. 进入 Stop 或 VLPS 模式，然后等待 GPIO 中断。

注

使用此方法时，可能无法正确接收 SPI 主机数据的第一个字。这取决于 SPI 所用的传输速率片选有效和数据有效之间的延时以及系统的中断延迟。

43.1.8 SPI 等待模式

SPI 模块的等待模式等同于芯片的等待和 VLPW 模式。

43.1.9 SPI 中断

SPI 有多个中断请求源。然而，这些源都经 OR 运算相结合，以生成针对每个 SPI 模块的中断控制器的单个中断请求。当出现 SPI 中断时，读取 SPI_SR 以确定确切的中断源。

43.1.10 SPI 时钟

此表显示了 SPI 模块时钟和相应的芯片时钟。

表 43-4. SPI 时钟连接

模块时钟	芯片时钟
系统时钟	总线时钟

43.2 简介

此串行通信接口 (SPI) 模块可提供同步串行总线，以实现芯片与外设器件之间的通信。

43.2.1 结构框图

此模块的结构框图如下：

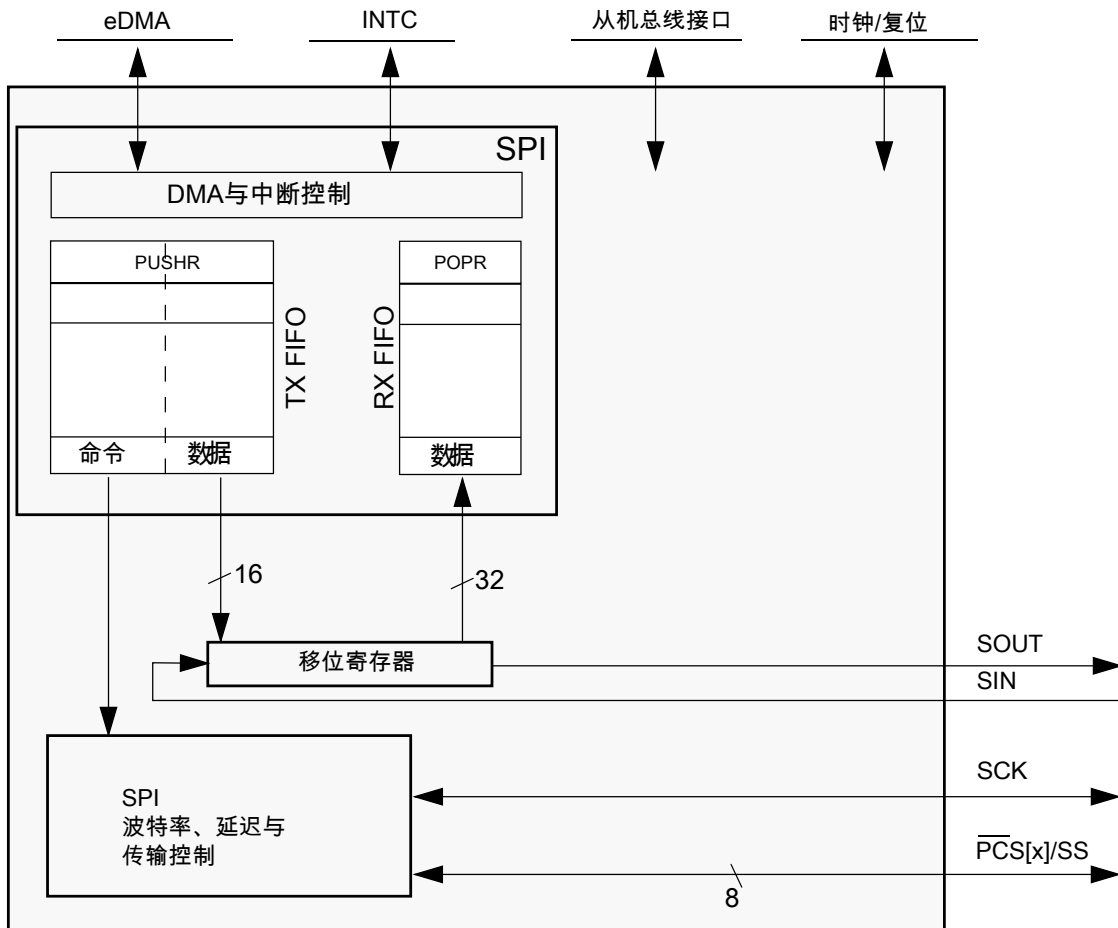


图 43-1. SPI 结构框图

43.2.2 特性

此模块支持下列特性：

- 全双工、3 线同步传输
- 主机模式
- 从机模式
- 从机模式下的数据流操作（持续从机片选有效）
- 使用发送 first in first out (TX FIFO)（深度为 4 条目）的缓冲发射操作
- 使用接收 FIFO (RX FIFO)（深度为 4 条目）的缓冲接收操作
- TX 和 RX FIFO 可以单独禁用，以实现 SPI 队列的低延迟更新
- TX 和 RX FIFO 可见性，以便于调试
- 可基于每帧设定传输属性：

- 2 传输属性寄存器
- 带可编程极性和相位的串行时钟 (SCK)
- 各种可编程延迟
- 可编程串行帧大小: 4 至 16 位
 - 用连续片选有效格式可支持长于 16 位的 SPI 帧。
 - 连续保持片选有效的能力
- 6 外设芯片选择 (PCS), 可用外部多路解调器扩展至 64
- 通过外部多路解调器支持对最多 32 个外设芯片选择 (PCS) 去毛刺
- DMA 支持将数据写入 TX FIFO 和将数据从 RX FIFO 中取出:
 - TX FIFO 未满足 (TFFF)
 - RX FIFO 不为空 (RFDF)
- 中断条件:
 - 队列结束 (EOQF)
 - TX FIFO 未满足 (TFFF)
 - 当前帧传输完成 (TCF)
 - 试图在 FIFO 空时发送 (TFUF)
 - RX FIFO 不为空 (RFDF)
 - 在接收 FIFO 已满时收到帧 (RFOF)
- 全局中断请求
- 可调整的 SPI 传输格式以支持与较慢的外围设备通信
- 省电架构特性:
 - 支持停止模式
 - 支持休眠模式

43.2.3 接口配置

43.2.3.1 SPI 配置

串行外设接口(SPI)配置允许模块发送和接收串行数据。此配置允许模块作为一个带内部 FIFO (支持外部队列操作) 的基本 SPI 数据块工作。发送的数据和接收的数据驻留在独立的 FIFO 中。主机 CPU 或 DMA 控制器读取从接收 FIFO 接收的数据, 并将发送数据写入发送 FIFO。

对于队列操作, SPI 队列可以驻留在模块外部的系统 RAM 中。队列和模块 FIFO 间的数据传输由一个 DMA 控制器或主机 CPU 完成。下图所示为一个系统示例, DMA、SPI 和外部队列在系统 RAM 中。

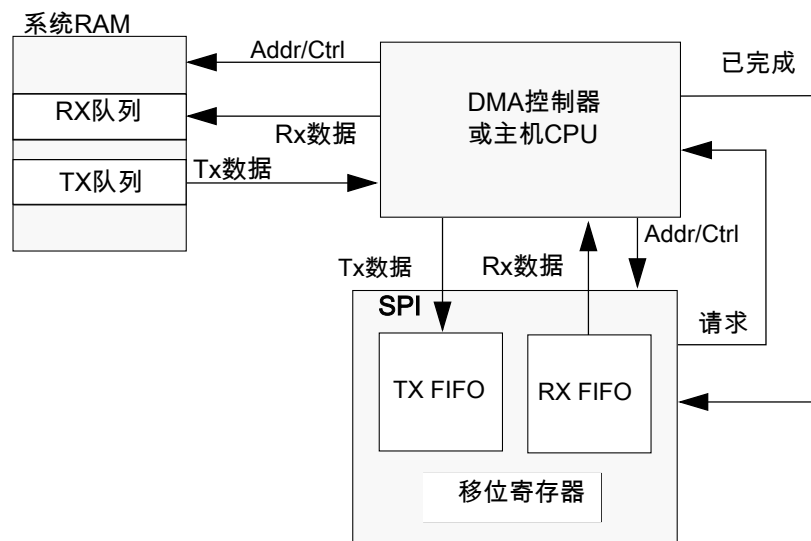


图 43-2. 带队列和 DMA 的 SPI

43.2.4 工作模式

此模块支持下列工作模式, 而这些模式可以分为两类:

- 特定模块模式:
 - 主机模式
 - 从机模式
 - 模块禁用模式
- 芯片特性模式:

- 外部停止模式
- 调试模式

此模块在主机写入模块寄存器时进入特定模块模式。芯片特性模式由模块外部的信号控制。芯片特性模式指芯片可与特定数据块模式并行进入的模式。

43.2.4.1 主机模式

主机模式允许模块启动和控制串行通信。在此模式下，这些信号由模块控制并配置为输出：

- SCK
- SOUT
- PCS[x]

43.2.4.2 从机模式

从机模式允许模块与 SPI 总线主机通信。在此模式下，模块会对外部控制的串行传输作出响应。SCK 信号和 PCS[0]/ \overline{SS} 信号配置为输入，并由一个 SPI 总线主机驱动。

43.2.4.3 模块禁用模式

模块禁用模式可以用于芯片电源管理。用于模块中非存储器映射逻辑的时钟可以在模块禁用模式下停用。

43.2.4.4 外部停止模式

外部停止模式用于芯片电源管理。该模块支持外设总线停止模式机制。当要求进入外部停止模式时，模块会确认请求并完成正在进行的传输。当模块到达帧边界时，会通知模块的协议时钟可能被关闭。

43.2.4.5 调试模式

调试模式用于系统开发和调试。MCR[FRZ]位控制调试模式下的模块行为：

- 如果该位置位，当芯片处于调试模式下时，模块会停止所有串行传输。
- 如果该位被清除，则芯片调试模式不会对模块产生影响。

43.3 模块信号说明

此表介绍芯片上可对外连接（按字母顺序）的模块信号。

表 43-5. 模块信号说明

信号	主机模式	从机模式	I/O
PCS0/ \overline{SS}	外设芯片选择 0 (O)	从机选择(I)	I/O
PCS[1:3]	外设芯片选择 1-3	(不使用)	O
PCS4	外设芯片选择 4	(不使用)	O
PCS5/ \overline{PCSS}	外设芯片选择 5 /外设芯片选择选通	(不使用)	O
SCK	串行时钟(O)	串行时钟	I/O
SIN	串行数据输入	串行数据输入	I
SOUT	串行数据输出	串行数据输出	O

43.3.1 PCS0/ \overline{SS} —外设片选/从机选择

主机模式：外设片选 0 (O)—选择一个 SPI 从机，以接收从模块传来的数据。

从机模式：从机选择(I)—选择模块，以接收从 SPI 主机传来的数据。

注

请勿把 SPI 从机选择引脚接地。否则，SPI 不能正常工作。

43.3.2 PCS1–PCS3—外设片选 1–3

主机模式：外设片选 1-3 (O)—选择一个 SPI 从机，以接收模块传输的数据。

从机模式：未使用

43.3.3 PCS4—外设片选 4

主机模式：外设片选 4 (O)—选择一个 SPI 从机，以接收模块传输的数据。

从机模式：未使用

43.3.4 PCS5/PCSS—外设片选 5 /外设片选选通

主机模式:

- 外设片选 5 (O)—仅用于外设片选选通被禁用时(MCR[PCSSE])。选择一个 SPI 从机，以接收模块传输的数据。
- 外设片选选通 (O)—仅用于外设片选选通使能时(MCR[PCSSE])。选通一个模块外外设片选多路解调器，该解调器把模块除 PCS5 以外的 PCS 信号解码，由此避免多路解调器输出上出现毛刺。

从机模式: 未使用

43.3.5 SCK—串行时钟

主机模式: 串行时钟 (O)—将来自模块的时钟信号提供给 SPI 从机。

从机模式: 串行时钟 (I)—从 SPI 主机将时钟信号提供给模块。

43.3.6 SIN—串行输入

主机模式: 串行输入 (I)—接收串行数据。

从机模式: 串行输入 (I)—接收串行数据。

43.3.7 SOUT—串行输出

主机模式: 串行输出 (O)—传输串行数据。

从机模式: 串行输出 (O)—传输串行数据。

43.4 存储器映射/寄存器定义

如果寄存器访问保留或未定义的存储器地址，结果会导致传输错误。对 POPR 和 RXFRn 的写访问也会导致传输错误。

SPI 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4002_C000	模块配置寄存器 (SPI0_MCR)	32	R/W	0000_4001h	43.4.1/963
4002_C008	传输计数寄存器 (SPI0_TCR)	32	R/W	0000_0000h	43.4.2/966

下一页继续介绍此表...

SPI 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4002_C00C	时钟和传输属性寄存器 (主机模式下) (SPI0_CTAR0)	32	R/W	7800_0000h	43.4.3/966
4002_C00C	时钟和传输属性寄存器(从机模式下) (SPI0_CTAR0_SLAVE)	32	R/W	7800_0000h	43.4.4/971
4002_C010	时钟和传输属性寄存器 (主机模式下) (SPI0_CTAR1)	32	R/W	7800_0000h	43.4.3/966
4002_C02C	状态寄存器 (SPI0_SR)	32	R/W	0200_0000h	43.4.5/972
4002_C030	DMA/中断请求选择和使能寄存器 (SPI0_RSER)	32	R/W	0000_0000h	43.4.6/974
4002_C034	主机模式下的 PUSH TX FIFO 寄存器 (SPI0_PUSHR)	32	R/W	0000_0000h	43.4.7/976
4002_C034	从机模式下的 PUSH TX FIFO 寄存器 (SPI0_PUSHR_SLAVE)	32	R/W	0000_0000h	43.4.8/978
4002_C038	POP RX FIFO 寄存器 (SPI0_POPR)	32	R	0000_0000h	43.4.9/978
4002_C03C	发送 FIFO 寄存器 (SPI0_TXFR0)	32	R	0000_0000h	43.4.10/ 979
4002_C040	发送 FIFO 寄存器 (SPI0_TXFR1)	32	R	0000_0000h	43.4.10/ 979
4002_C044	发送 FIFO 寄存器 (SPI0_TXFR2)	32	R	0000_0000h	43.4.10/ 979
4002_C048	发送 FIFO 寄存器 (SPI0_TXFR3)	32	R	0000_0000h	43.4.10/ 979
4002_C07C	接收 FIFO 寄存器 (SPI0_RXFR0)	32	R	0000_0000h	43.4.11/ 979
4002_C080	接收 FIFO 寄存器 (SPI0_RXFR1)	32	R	0000_0000h	43.4.11/ 979
4002_C084	接收 FIFO 寄存器 (SPI0_RXFR2)	32	R	0000_0000h	43.4.11/ 979
4002_C088	接收 FIFO 寄存器 (SPI0_RXFR3)	32	R	0000_0000h	43.4.11/ 979
4002_D000	模块配置寄存器 (SPI1_MCR)	32	R/W	0000_4001h	43.4.1/963
4002_D008	传输计数寄存器 (SPI1_TCR)	32	R/W	0000_0000h	43.4.2/966
4002_D00C	时钟和传输属性寄存器 (主机模式下) (SPI1_CTAR0)	32	R/W	7800_0000h	43.4.3/966
4002_D00C	时钟和传输属性寄存器(从机模式下) (SPI1_CTAR0_SLAVE)	32	R/W	7800_0000h	43.4.4/971
4002_D010	时钟和传输属性寄存器 (主机模式下) (SPI1_CTAR1)	32	R/W	7800_0000h	43.4.3/966
4002_D02C	状态寄存器 (SPI1_SR)	32	R/W	0200_0000h	43.4.5/972
4002_D030	DMA/中断请求选择和使能寄存器 (SPI1_RSER)	32	R/W	0000_0000h	43.4.6/974
4002_D034	主机模式下的 PUSH TX FIFO 寄存器 (SPI1_PUSHR)	32	R/W	0000_0000h	43.4.7/976
4002_D034	从机模式下的 PUSH TX FIFO 寄存器 (SPI1_PUSHR_SLAVE)	32	R/W	0000_0000h	43.4.8/978
4002_D038	POP RX FIFO 寄存器 (SPI1_POPR)	32	R	0000_0000h	43.4.9/978
4002_D03C	发送 FIFO 寄存器 (SPI1_TXFR0)	32	R	0000_0000h	43.4.10/ 979
4002_D040	发送 FIFO 寄存器 (SPI1_TXFR1)	32	R	0000_0000h	43.4.10/ 979
4002_D044	发送 FIFO 寄存器 (SPI1_TXFR2)	32	R	0000_0000h	43.4.10/ 979

下一页继续介绍此表...

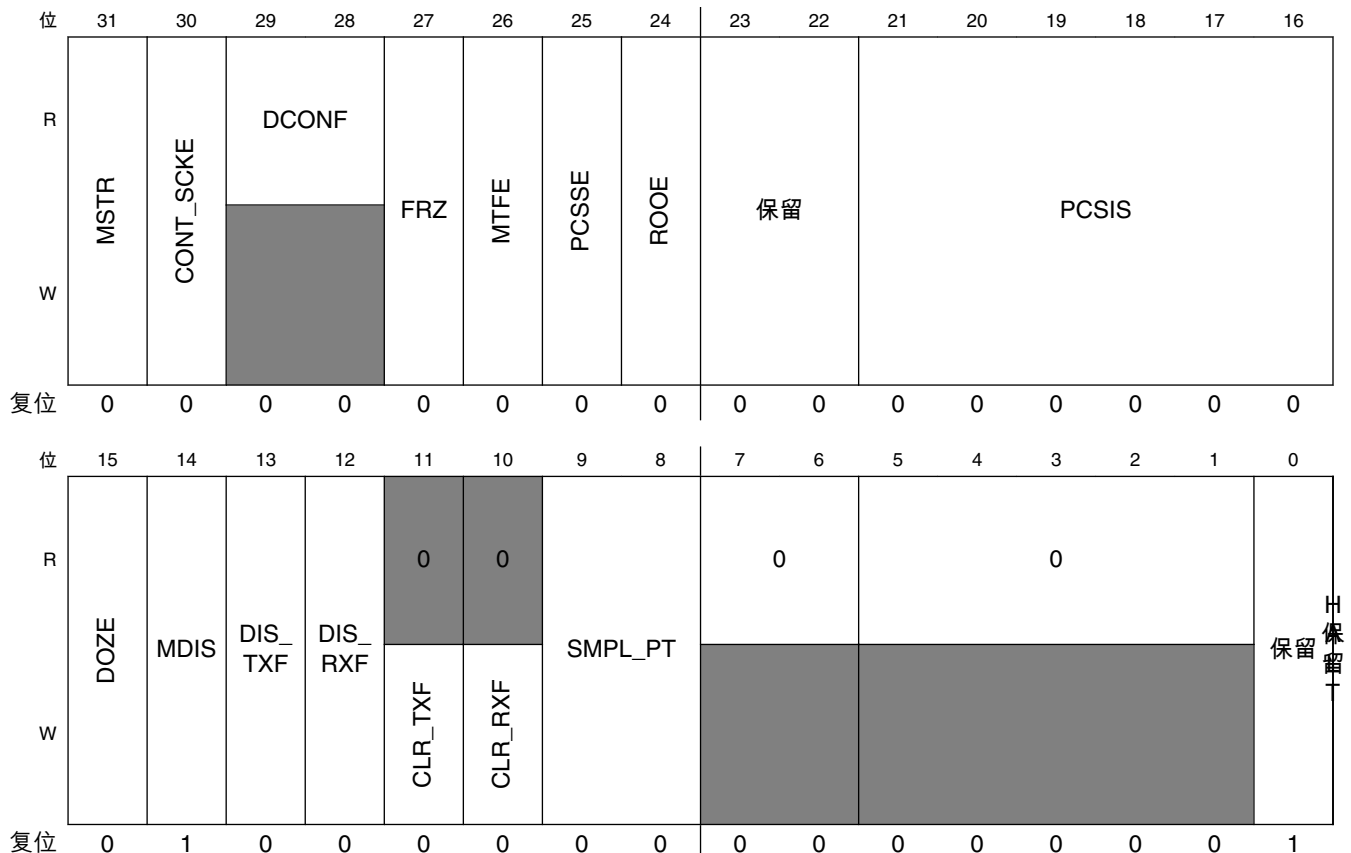
SPI 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4002_D048	发送 FIFO 寄存器 (SPI1_TXFR3)	32	R	0000_0000h	43.4.10/ 979
4002_D07C	接收 FIFO 寄存器 (SPI1_RXFR0)	32	R	0000_0000h	43.4.11/ 979
4002_D080	接收 FIFO 寄存器 (SPI1_RXFR1)	32	R	0000_0000h	43.4.11/ 979
4002_D084	接收 FIFO 寄存器 (SPI1_RXFR2)	32	R	0000_0000h	43.4.11/ 979
4002_D088	接收 FIFO 寄存器 (SPI1_RXFR3)	32	R	0000_0000h	43.4.11/ 979

43.4.1 模块配置寄存器 (SPIx_MCR)

包含配置与模块操作相关的各种属性的位。HALT 和 MDIS 位可以随时更改，但在下一帧才起作用。模块处于运行状态时，只有 MCR 中的 HALT 和 MDIS 位可以更改。

地址: 基址 基准 + 0h 偏移



SPIx_MCR 字段描述

字段	描述
31 MSTR	<p>主机/从机模式选择</p> <p>使能主机工作模式 (若支持) 或从机工作模式 (若支持)。</p> <p>0 使能从机模式 1 使能主机模式</p>
30 CONT_SCKE	<p>连续 SCK 使能</p> <p>使串行通信时钟 (SCK) 可连续运行。</p> <p>0 连续 SCK 禁用。 1 连续 SCK 使能。</p>
29–28 DCONF	<p>SPI 配置。</p> <p>在模块的不同配置中选择。</p> <p>00 SPI 01 保留 10 保留 11 保留</p>
27 FRZ	<p>冻结</p> <p>使传输在设备进入调试模式时可停在下一帧边界上。</p> <p>0 不暂停调试模式下的串行传输。 1 暂停调试模式下的串行传输。</p>
26 MTFE	<p>修正传输格式使能</p> <p>使修正传输格式可用。</p> <p>0 修正 SPI 传输模式禁用。 1 修正 SPI 传输模式使能。</p>
25 PCSSE	<p>外设芯片选择选通使能</p> <p>使 PCS5/ \overline{PCSS} 可用作 PCS 选通输出信号。</p> <p>0 PCS5/ \overline{PCSS} 用作外设芯片选择[5]信号。 1 PCS5/ \overline{PCSS} 用作有效低电平 PCS 选通信号。</p>
24 ROOE	<p>接收 FIFO 溢出覆盖使能</p> <p>在 RX FIFO 溢出情况下, 配置模块以忽略传入串行数据或覆盖现有数据。如果 RX FIFO 已满且收到新数据, 则产生溢出的传输数据被忽略或移入移位寄存器。</p> <p>0 传入数据被忽略。 1 传入数据移入移位寄存器。</p>
23–22 保留	始终向该字段写入复位值。
21–16 PCSIS	<p>外设芯片选择 x 无效状态</p> <p>确定 PCSx 的无效状态。有关芯片中所用 PCS 信号的数量, 请参见芯片特定 SPI 信息。</p> <p>注: 该位仅在模块使能时才起作用。确保在使能 SPI 接口前正确配置该位。</p>

下一页继续介绍此表...

SPIx_MCR 字段描述 (继续)

字段	描述
	0 PCSx 无效状态为低电平。 1 PCSx 无效状态为高电平。
15 DOZE	休眠使能 为休眠模式节电机制提供支持。 0 休眠模式对模块没有影响 1 休眠模式禁用模块
14 MDIS	模块禁用 允许有效地停止用于模块中非存储器映射逻辑的时钟，将其置于软件控制的节电状态。MDIS 位的复位值参数化，默认复位值为 1。模块在从机模式下使用时，建议此位留为 0，因为从机无法控制主机事务。 0 使能模块时钟。 1 允许外部逻辑禁用模块时钟。
13 DIS_TXF	禁用发送 FIFO TX FIFO 禁用时，模块的发送部分用作简化的双缓冲 SPI。只有 MDIS 位清零才可写入此位。 0 TX FIFO 已使能。 1 TX FIFO 已禁用。
12 DIS_RXF	禁用接收 FIFO RX FIFO 禁用时，模块的接收部分用作简化的双缓冲 SPI。只有 MDIS 位清零才可写入此位。 0 RX FIFO 已使能。 1 RX FIFO 已禁用。
11 CLR_TXF	清零 TX FIFO 刷新 TX FIFO。向 CLR_TXF 写入 1 清零 TX FIFO 计数器。CLR_TXF 位始终读为 0。 0 不清零 TX FIFO 计数器。 1 清零 TX FIFO 计数器。
10 CLR_RXF	CLR_RXF 刷新 RX FIFO。向 CLR_RXF 写入 1 清零 RX 计数器。CLR_RXF 位始终读为 0。 0 不清零 RX FIFO 计数器。 1 清零 RX FIFO 计数器。
9-8 SMPL_PT	采样点 在修正传输格式中，控制模块主机何时采样 SIN。该字段仅在 CTARn[CPHA]中的 CPHA 位是 0 时有效。 00 SCK 边沿和 SIN 采样之间有 0 个协议时钟周期 01 SCK 边沿和 SIN 采样之间有 1 个协议时钟周期 10 SCK 边沿和 SIN 采样之间有 2 个协议时钟周期 11 保留
9-8 保留	此只读字段为保留字段且值始终为 0。
7-3 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

SPIx_MCR 字段描述 (继续)

字段	描述
2 保留	
1 保留	
0 HALT	Halt HALT 位开始和停止帧传输。请参见 模块传输的开始和停止 0 开始传输。 1 停止传输。

43.4.2 传输计数寄存器 (SPIx_TCR)

TCR 内置指示创建的 SPI 传输数量的计数器。传输计数器旨在帮助队列管理。请勿在模块处于运行状态时写入 TCR。

地址: 基址 基准 + 8h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPI_TCNT																0															
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_TCR 字段描述

字段	描述
31-16 SPI_TCNT	SPI 传输计数器 计数模块创建的 SPI 传输的数量。每次 SPI 帧最后位发送完成后，SPI_TCNT 字段递增。向 SPI_TCNT 写入的值将计数器预设为此值。当 CTCNT 字段在执行 SPI 命令中设定时，每帧开始时 SPI_TCNT 复位到 0。传输计数器环回；递增计数器超出 65535，复位计数器到 0。
保留	此只读字段为保留字段且值始终为 0。

43.4.3 时钟和传输属性寄存器 (主机模式下) (SPIx_CTARn)

CTAR 寄存器用于定义不同传输属性。请勿在模块处于运行状态时写入 CTAR 寄存器。

在主机模式下，CTAR 寄存器定义传输属性的组合，比如帧大小、时钟相位和极性、数据位顺序、波特率以及各种延时。在从机模式下，CTAR0 中的一组位域用于设置从机传输属性。

模块配置为 SPI 主机时，TX FIFO 条目的命令部分的 CTAS 字段选择使用哪个 CTAR 寄存器。模块配置为 SPI 总线从机时，它使用 CTAR0 寄存器。

地址: 基址 + Ch 偏移 + (4d × i), 其中 i=0d 到 1d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R	DBR		FMSZ				CPOL	CPHA	LSBFE	PCSSCK		PASC		PDT		PBR	
W																	
复位	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	CSSCK				ASC				DT				BR				
W																	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

SPIx_CTARn 字段描述

字段	描述																																								
31 DBR	<p>双波特率</p> <p>串行通信时钟 (SCK) 的有效波特率翻倍。该字段仅在主机模式下使用。它有效地减半了串行通信时钟 (SCK) 的波特率分频比 (以支持更快的频率) 和奇数分频率。当 DBR 位置位时, 串行通信时钟 (SCK) 的占空比取决于波特率预分频器中的值和下表中所列的时钟相位。有关波特率计算方法的详细信息, 请参见 BR 字段描述。</p> <p style="text-align: center;">表 43-48. SPI SCK 占空比</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>DBR</th> <th>CPHA</th> <th>PBR</th> <th>SCK 占空比</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>任意值</td> <td>任意值</td> <td>50/50</td> </tr> <tr> <td>1</td> <td>0</td> <td>00</td> <td>50/50</td> </tr> <tr> <td>1</td> <td>0</td> <td>01</td> <td>33/66</td> </tr> <tr> <td>1</td> <td>0</td> <td>10</td> <td>40/60</td> </tr> <tr> <td>1</td> <td>0</td> <td>11</td> <td>43/57</td> </tr> <tr> <td>1</td> <td>1</td> <td>00</td> <td>50/50</td> </tr> <tr> <td>1</td> <td>1</td> <td>01</td> <td>66/33</td> </tr> <tr> <td>1</td> <td>1</td> <td>10</td> <td>60/40</td> </tr> <tr> <td>1</td> <td>1</td> <td>11</td> <td>57/43</td> </tr> </tbody> </table> <p>0 通常波特率计算采用 50/50 占空比。 1 波特率翻倍, 并且占空比可根据分频器配置。</p>	DBR	CPHA	PBR	SCK 占空比	0	任意值	任意值	50/50	1	0	00	50/50	1	0	01	33/66	1	0	10	40/60	1	0	11	43/57	1	1	00	50/50	1	1	01	66/33	1	1	10	60/40	1	1	11	57/43
DBR	CPHA	PBR	SCK 占空比																																						
0	任意值	任意值	50/50																																						
1	0	00	50/50																																						
1	0	01	33/66																																						
1	0	10	40/60																																						
1	0	11	43/57																																						
1	1	00	50/50																																						
1	1	01	66/33																																						
1	1	10	60/40																																						
1	1	11	57/43																																						
30-27 FMSZ	<p>帧长度</p> <p>每帧传输的位数等于 FMSZ 值加上 1。无论何种发送模式, 最小有效帧大小值为 4。</p>																																								
26 CPOL	时钟极性																																								

下一页继续介绍此表...

SPIx_CTARn 字段描述 (继续)

字段	描述
	<p>选择串行通信时钟 (SCK) 的无效状态。该位在主机和从机模式下使用。串行设备间要成功通信, 设备必须具有相同的时钟极性。当选择连续选择格式时, 在不停止模块的情况下在时钟极性之间切换可能导致传输错误, 因为外设设备将时钟极性切换视为一个有效时钟边沿。</p> <p>注: 如果是连续 SCK 模式, 模块进入低功耗模式 (已禁用) 时, 不能保证 SCK 的无效状态。</p> <p>0 SCK 的无效状态为低电平。 1 SCK 的无效状态为高电平。</p>
25 CPHA	<p>时钟相位</p> <p>选择使数据更改的 SCK 边沿和使数据被捕捉的边沿。该位在主机和从机模式下使用。串行设备间要成功通信, 设备必须具有相同的时钟相位设置。在连续 SCK 模式下, 该位值被忽略且传输如同在 CPHA 位设置为 1 时一样完成。</p> <p>0 数据在 SCK 的前沿被捕捉并在下一边沿更改。 1 数据在 SCK 的前沿更改并在下一边沿被捕捉。</p>
24 LSBFE	<p>LSB 优先</p> <p>指定帧的 LSB 或 MSB 优先传输。</p> <p>0 数据传输 MSB 优先。 1 数据传输 LSB 优先。</p>
23–22 PCSSCK	<p>PCS 至 SCK 延迟预分频器</p> <p>为 PCS 置位与 SCK 首个边沿之间的延迟选择预分频器值。有关 PCS 至 SCK 延迟计算方法的信息, 请参见 CSSCK 字段描述。有关更多详细信息, 请参见 PCS 至 SCK 延迟 (t_{csc})。</p> <p>00 PCS 至 SCK 预分频器值为 1。 01 PCS 至 SCK 预分频器值为 3。 10 PCS 至 SCK 预分频器值为 5。 11 PCS 至 SCK 预分频器值为 7。</p>
21–20 PASC	<p>SCK 后延迟预分频器</p> <p>为 SCK 最后边沿和 PCS 无效之间的延迟选择预分频器值。有关 SCK 后延迟计算方法的信息, 请参见 ASC 字段描述。有关更多详细信息, 请参见 SCK 后延迟 (t_{asc})。</p> <p>00 传输后延迟预分频器值为 1。 01 传输后延迟预分频器值为 3。 10 传输后延迟预分频器值为 5。 11 传输后延迟预分频器值为 7。</p>
19–18 PDT	<p>传输后延迟预分频器</p> <p>为某帧结束时的 PCS 信号无效与下一帧开始时的 PCS 有效之间的延迟选择预分频器值。PDT 字段仅在主机模式下使用。有传输后延迟计算方法的详细信息, 请参见 DT 字段描述。有关更多详细信息, 请参见 传输后延迟 (t_{dt})。</p> <p>00 传输后延迟预分频器值为 1。 01 传输后延迟预分频器值为 3。 10 传输后延迟预分频器值为 5。 11 传输后延迟预分频器值为 7。</p>
17–16 PBR	<p>波特率预分频器</p>

下一页继续介绍此表...

SPIx_CTARn 字段描述 (继续)

字段	描述																																		
	<p>选择波特率的预分频值。该字段仅在主机模式下使用。波特率是 SCK 的频率。在产生波特率之前,协议时钟由预分频器值分频。有关波特率计算方法的详细信息,请参见 BR 字段描述。</p> <p>00 波特率预分频器值为 2。 01 波特率预分频器值为 3。 10 波特率预分频器值为 5。 11 波特率预分频器值为 7。</p>																																		
15-12 CSSCK	<p>PCS 至 SCK 延迟分频器</p> <p>选择 PCS 至 SCK 延迟的分频器值。该字段仅在主机模式下使用。PCS 至 SCK 延迟是 PCS 有效到 SCK 首个边沿之间的延迟。该延迟是协议时钟周期的倍数,按照以下等式计算:</p> $t_{CSC} = (1/f_P) \times PCSSCK \times CSSCK.$ <p>下表所列为分频器值。</p> <p style="text-align: center;">表 43-47. 延迟分频器编码</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>字段值</th> <th>延迟分频器值</th> </tr> </thead> <tbody> <tr><td>0000</td><td>2</td></tr> <tr><td>0001</td><td>4</td></tr> <tr><td>0010</td><td>8</td></tr> <tr><td>0011</td><td>16</td></tr> <tr><td>0100</td><td>32</td></tr> <tr><td>0101</td><td>64</td></tr> <tr><td>0110</td><td>128</td></tr> <tr><td>0111</td><td>256</td></tr> <tr><td>1000</td><td>512</td></tr> <tr><td>1001</td><td>1024</td></tr> <tr><td>1010</td><td>2048</td></tr> <tr><td>1011</td><td>4096</td></tr> <tr><td>1100</td><td>8192</td></tr> <tr><td>1101</td><td>16384</td></tr> <tr><td>1110</td><td>32768</td></tr> <tr><td>1111</td><td>65536</td></tr> </tbody> </table> <p>有关更多详细信息,请参见 PCS 至 SCK 延迟 (t_{CSC})。</p>	字段值	延迟分频器值	0000	2	0001	4	0010	8	0011	16	0100	32	0101	64	0110	128	0111	256	1000	512	1001	1024	1010	2048	1011	4096	1100	8192	1101	16384	1110	32768	1111	65536
字段值	延迟分频器值																																		
0000	2																																		
0001	4																																		
0010	8																																		
0011	16																																		
0100	32																																		
0101	64																																		
0110	128																																		
0111	256																																		
1000	512																																		
1001	1024																																		
1010	2048																																		
1011	4096																																		
1100	8192																																		
1101	16384																																		
1110	32768																																		
1111	65536																																		
11-8 ASC	<p>SCK 后延迟分频器</p> <p>选择 SCK 后延迟的分频器值。该字段仅在主机模式下使用。SCK 后延迟是 SCK 最后边沿与 PCS 无效之间的延迟。该延迟是协议时钟周期的倍数,按照以下等式计算:</p> $t_{ASC} = (1/f_P) \times PASC \times ASC$ <p>有关分频器值,请参见 CTARn[CSSCK]位字段描述中的延迟编码表。有关更多详细信息,请参见 SCK 后延迟 (t_{ASC})。</p>																																		
7-4 DT	传输后延迟分频器																																		

下一页继续介绍此表...

SPIx_CTARn 字段描述 (继续)

字段	描述																																		
	<p>选择传输后延迟分频器。该字段仅在主机模式下使用。传输后延迟是某帧结束时的 PCS 信号无效与下一帧开始时的 PCS 有效之间的时间。</p> <p>在连续串行通信时钟操作中，DT 值固定为一个 SCK 时钟周期。传输后延迟是协议时钟周期的倍数，按照以下等式计算：</p> $t_{DT} = (1/f_P) \times PDT \times DT$ <p>有关分频器值，请参见 CTARn[CSSCK]位字段描述中的延迟编码表。</p>																																		
BR	<p>波特率分频器</p> <p>选择波特率的分频器值。该字段仅在主机模式下使用。预分频的协议时钟由波特率分频器分频，以生成 SCK 频率。波特率按照以下等式计算：</p> $SCK \text{ 波特率} = (f_P / PBR) \times [(1+DBR)/BR]$ <p>下表所列为波特率分频器值。</p> <p style="text-align: center;">表 43-46. 波特率分频器</p> <table border="1" data-bbox="349 733 1474 1437"> <thead> <tr> <th>CTARn[BR]</th> <th>波特率分频器值</th> </tr> </thead> <tbody> <tr><td>0000</td><td>2</td></tr> <tr><td>0001</td><td>4</td></tr> <tr><td>0010</td><td>6</td></tr> <tr><td>0011</td><td>8</td></tr> <tr><td>0100</td><td>16</td></tr> <tr><td>0101</td><td>32</td></tr> <tr><td>0110</td><td>64</td></tr> <tr><td>0111</td><td>128</td></tr> <tr><td>1000</td><td>256</td></tr> <tr><td>1001</td><td>512</td></tr> <tr><td>1010</td><td>1024</td></tr> <tr><td>1011</td><td>2048</td></tr> <tr><td>1100</td><td>4096</td></tr> <tr><td>1101</td><td>8192</td></tr> <tr><td>1110</td><td>16384</td></tr> <tr><td>1111</td><td>32768</td></tr> </tbody> </table>	CTARn[BR]	波特率分频器值	0000	2	0001	4	0010	6	0011	8	0100	16	0101	32	0110	64	0111	128	1000	256	1001	512	1010	1024	1011	2048	1100	4096	1101	8192	1110	16384	1111	32768
CTARn[BR]	波特率分频器值																																		
0000	2																																		
0001	4																																		
0010	6																																		
0011	8																																		
0100	16																																		
0101	32																																		
0110	64																																		
0111	128																																		
1000	256																																		
1001	512																																		
1010	1024																																		
1011	2048																																		
1100	4096																																		
1101	8192																																		
1110	16384																																		
1111	32768																																		

43.4.4 时钟和传输属性寄存器 (从机模式下) (SPIx_CTARn_SLAVE)

模块配置为 SPI 总线从机时, 使用 CTAR0 寄存器。

地址: 基址 + Ch 偏移 + (0d × i), 其中 i=0d 到 0d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R								0								
W	保留	FMSZ				CPOL	CPHA			保留	保留					
复位	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	保留															
W	保留															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_CTARn_SLAVE 字段描述

字段	描述
31 保留	始终向该字段写入复位值。
30–27 FMSZ	帧长度 每帧传输的位数等于 FMSZ 字段值加上 1。注意帧大小的最小有效值为 4。
26 CPOL	时钟极性 选择串行通信时钟 (SCK) 的无效状态。 注: 如果是连续 SCK 模式, 模块进入低功耗模式 (已禁用) 时, 不能保证 SCK 的无效状态。 0 SCK 的无效状态为低电平。 1 SCK 的无效状态为高电平。
25 CPHA	时钟相位 选择使数据更改的边沿和使数据被捕获的边沿。该位用于主机和从机模式。串行设备间要成功通信, 设备必须具有相同的时钟相位设置。在连续 SCK 模式下, 该位值被忽略且传输如同在 CPHA 位设置为 1 时一样完成。 0 数据在 SCK 的前沿被捕捉并在下一边沿更改。 1 数据在 SCK 的前沿更改并在下一边沿被捕捉。
24–23 保留	此只读字段为保留字段且值始终为 0。
22 保留	此只读字段为保留字段。
保留	此只读字段为保留字段。

43.4.5 状态寄存器 (SPIx_SR)

SR 包含状态和标志位。该位反映模块的状态，指示可能产生中断或 DMA 请求的事件的发生。软件可以向 SR 中的标志位写入 1 将其清零。向标志位写入 0 无效。由于使用节电机制，该寄存器在模块禁用模式下可能不可写。

地址: 基址 基准 + 2Ch 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCF	TXRXS	0	EOQF	TFUF	0	TEFF	0	0	0	0	0	RFOF	0	RDFD	0
W	w1c			w1c	w1c		w1c						w1c		w1c	
复位	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXCTR				TXNXPTR				RXCTR				POPNXPTR			
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_SR 字段描述

字段	描述
31 TCF	传输完成标志 指示帧中的所有位已经移出。TCF 保持置位，直到向其写入 1 将其清零。 0 传输未完成。 1 传输完成。
30 TXRXS	TX 和 RX 状态 反映模块的运行状态。 0 发送和接收操作禁用 (模块处于停止状态)。 1 发送和接收操作使能 (模块处于运行状态)。

下一页继续介绍此表...

SPIx_SR 字段描述 (继续)

字段	描述
29 保留	此只读字段为保留字段且值始终为 0。
28 EOQF	<p>队列结束标志</p> <p>指示当模块在主机模式下时队列中的最后条目已经发送。TX FIFO 条目在命令半字中置位 EOQ 位且到达传输结束时，EOQF 位置位。EOQF 位保持置位，直到向其写入 1 将其清零。EOQF 位置位时，TXRXS 位自动清零。</p> <p>0 EOQ 在执行命令中不置位。 1 EOQ 在执行 SPI 命令中置位。</p>
27 TFUF	<p>发送 FIFO 下溢标志</p> <p>指示 TX FIFO 中的下溢情况。只会针对工作于从机模式下的 SPI 模块和 SPI 配置检测发送下溢情况。在 SPI 从机模式下运行的模块的 TX FIFO 为空且外部 SPI 主机发起传输时，TFUF 置位。TFUF 位保持置位，直到向其写入 1 将其清零。</p> <p>0 无 TX FIFO 下溢。 1 已经发生 TX FIFO 下溢。</p>
26 保留	此只读字段为保留字段且值始终为 0。
25 TFFF	<p>发送 FIFO 填充标志</p> <p>为模块请求更多条目加至 TX FIFO 提供一种方法。TX FIFO 不为空时，TFFF 位置位。TX FIFO 为空时，向 TFFF 位写入 1 或者 DMA 控制器应答可将其清零。 注：当模块禁用时 (MCR[MDIS]=1)，该位的复位值为 0。</p> <p>0 TX FIFO 已满。 1 TX FIFO 未滿。</p>
24 保留	此只读字段为保留字段且值始终为 0。
23 保留	此只读字段为保留字段且值始终为 0。
22 保留	此只读字段为保留字段且值始终为 0。
21 保留	此只读字段为保留字段且值始终为 0。
20 保留	此只读字段为保留字段且值始终为 0。
19 RFOF	<p>接收 FIFO 溢出标志</p> <p>指示 RX FIFO 中的溢出情况。RX FIFO 和移位寄存器为满且传输开始时，该位置位。该位保持置位，直到向其写入“1”将其清零。</p> <p>0 没有 Rx FIFO 溢出。 1 已经发生 Rx FIFO。</p>
18 保留	此只读字段为保留字段且值始终为 0。
17 RDFD	接收 FIFO 耗尽标志

下一页继续介绍此表...

SPIx_SR 字段描述 (继续)

字段	描述
	为模块请求从 RX FIFO 移出条目提供的一种方法。RX FIFO 不为空时，该位置位。RX FIFO 为空时，向 RFDF 位写入 1 或者 DMA 控制器应答可将其清零。 0 RX FIFO 为空。 1 RX FIFO 不为空。
16 保留	此只读字段为保留字段且值始终为 0。
15-12 TXCTR	TX FIFO 计数器 指示 TX FIFO 中条目的数量。每当 PUSHX 写入时，TXCTR 都会递增。每当 SPI 命令执行和 SPI 数据传输至移位寄存器时，TXCTR 都会递减。
11-8 TXNXTPTR	发送下一指针 指示下一传输中发送的 TX FIFO 条目。每当 SPI 数据从 TX FIFO 传输至移位寄存器时，TXNXTPTR 字段都会更新。
7-4 RXCTR	RX FIFO 计数器 指示 RX FIFO 中条目的数量。每当读取 POPR 时，RXCTR 都会递减。每当将数据从移位寄存器传输至 RX FIFO 时，RXCTR 都会递增。
POPXTPTR	读取下一指针 包含指向读取 POPR 时需要返回的 RX FIFO 条目的指针。读取 POPR 时，更新 POPXTPTR。

43.4.6 DMA/中断请求选择和使能寄存器 (SPIx_RSER)

RSER 控制 DMA 和中断请求。请勿在模块处于运行状态时写入 RSER。

地址: 基址 基准 + 30h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCF_RE	保留	保留	EOQF_RE	TFUF_RE	保留	TFFF_RE	TFFF_DIRS	保留	保留	保留	保留	RFOF_RE	保留	RFDF_RE	RFDF_DIRS
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	保留	保留	0													
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_RSER 字段描述

字段	描述
31 TCF_RE	发送完成请求使能 使能 SR 中的 TCF 标志以生成中断请求。 0 TCF 中断请求禁用。 1 TCF 中断请求使能。
30 保留	始终向该字段写入复位值。
29 保留	始终向该字段写入复位值。
28 EOQF_RE	已完成请求使能 使能 SR 中的 EOQF 标志以生成中断请求。 0 EOQF 中断请求禁用。 1 EOQF 中断请求使能。
27 TFUF_RE	发送 FIFO 下溢请求使能 使能 SR 中的 TFUF 标志以生成中断请求。 0 TFUF 中断请求禁用。 1 TFUF 中断请求使能。
26 保留	始终向该字段写入复位值。
25 TFFF_RE	发送 FIFO 填充请求使能 使能 SR 中的 TFFF 标志以生成请求。TFFF_DIRS 位选择生成中断请求或 DMA 请求。 0 TFFF 中断或 DMA 请求禁用。 1 TFFF 中断或 DMA 请求使能。
24 TFFF_DIRS	发送 FIFO 填充 DMA 或中断请求选择 选择生成 DMA 请求或中断请求。当 SR[TFFF]和 RSER[TFFF_RE]置位时,该字段选择生成中断或 DMA 请求。 0 TFFF 标志生成中断请求。 1 TFFF 标志生成 DMA 请求。
23 保留	始终向该字段写入复位值。
22 保留	始终向该字段写入复位值。
21 保留	始终向该字段写入复位值。
20 保留	始终向该字段写入复位值。
19 RFOF_RE	接收 FIFO 溢出请求使能 使能 SR 中的 RFOF 标志以生成中断请求。

下一页继续介绍此表...

SPIx_RSER 字段描述 (继续)

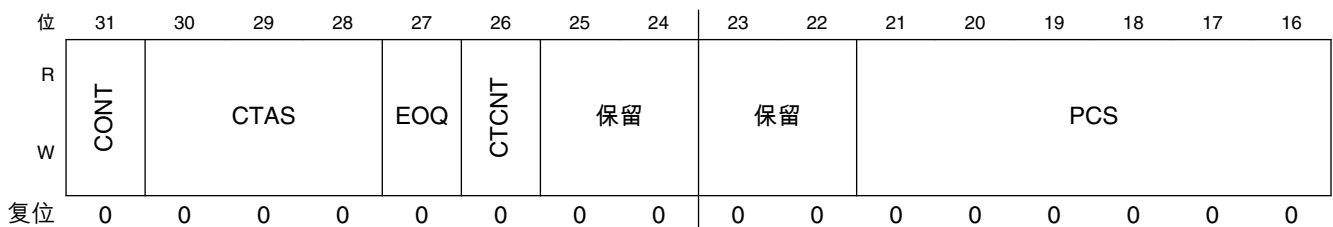
字段	描述
	0 RFOF 中断请求禁用。 1 RFOF 中断请求使能。
18 保留	始终向该字段写入复位值。
17 RFDF_RE	接收 FIFO 耗尽请求使能 使能 SR 中的 RFDF 标志以生成请求。RFDF_DIRS 位选择生成中断请求或 DMA 请求。 0 RFDF 中断或 DMA 请求禁用。 1 RFDF 中断或 DMA 请求使能。
16 RFDF_DIRS	接收 FIFO 耗尽 DMA 或中断请求选择 选择生成 DMA 请求或中断请求。当 SR[TFFF]和 RSER[TFFF_RE]置位时,该字段选择生成中断或 DMA 请求。 0 中断请求。 1 DMA 请求。
15 保留	始终向该字段写入复位值。
14 保留	始终向该字段写入复位值。
保留	此只读字段为保留字段且值始终为 0。

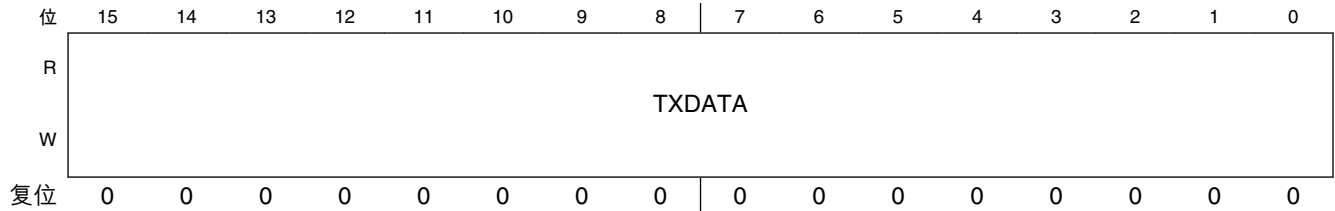
43.4.7 主机模式下的 PUSH TX FIFO 寄存器 (SPIx_PUSHR)

指定传输至 TX FIFO 的数据。8 位或 16 位写入访问传输全部 32 位数据至 TX FIFO。在主机模式下，寄存器传输 16 位数据和 16 位命令信息。对于 PUSHR 寄存器的读取访问返回最上方的 TX FIFO 条目。

模块禁用时，写入寄存器不更新 FIFO。因此，模块禁用时执行的任何读取操作，将返回模块使能时执行的最后一次 PUSHR 写入的值。

地址: 基址 基准 + 34h 偏移





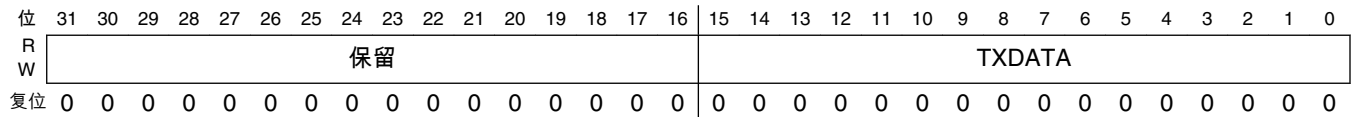
SPIx_PUSHR 字段描述

字段	描述
31 CONT	<p>连续外设芯片选择使能</p> <p>选择连续选择模式。该位在 SPI 主机模式下使用。该位使所选 PCS 信号在传输之间保持有效。</p> <p>0 在传输之间将 RCSn 信号置位无效状态。 1 在传输之间保持 PCSn 信号有效。</p>
30-28 CTAS	<p>时钟和传输属性选择</p> <p>选择主机模式下使用的 CTAR，以指定相关 SPI 帧的传输属性。在 SPI 从机模式下，使用 CTAR0。有关确定设备拥有的 CTAR 数量的详细信息，请参见芯片特定章节。不得在该字段中为不存在的寄存器编入值。</p> <p>000 CTAR0 001 CTAR1 010 保留 011 保留 100 保留 101 保留 110 保留 111 保留</p>
27 EOQ	<p>队列结束</p> <p>主机软件使用该位向模块发送信号，指示当前 SPI 传输是队列中最后一个。传输结束时，SR 中的 EOQF 位置位。</p> <p>0 SPI 数据不是要传输的最后数据。 1 SPI 数据是要传输的最后数据。</p>
26 CTCNT	<p>清零传输计数器</p> <p>清零 TCR 寄存器中的 TCNT 字段。模块开始发送当前 SPI 帧之前清零 TCNT 字段。</p> <p>0 不清零 TCR[TCNT]字段。 1 清零 TCR[TCNT]字段。</p>
25-24 保留	始终向该字段写入复位值。
23-22 保留	始终向该字段写入复位值。
21-16 PCS	<p>选择传输所需要使用的 PCS 信号，有关芯片中所用 PCS 信号的数量，请参见芯片特定 SPI 信息。</p> <p>0 PCS[x]信号无效。 1 PCS[x]信号有效。</p>
TXDATA	<p>发送数据</p> <p>存储和当前命令相关的 SPI 传输数据。</p>

43.4.8 从机模式下的 PUSH TX FIFO 寄存器 (SPIx_PUSHR_SLAVE)

指定从机模式下要传输至 TX FIFO 的数据。对 PUSHR 的 8 位或 16 位写入访问传输 16 位 TXDATA 字段至 TX FIFO。

地址: 基址 基准 + 34h 偏移



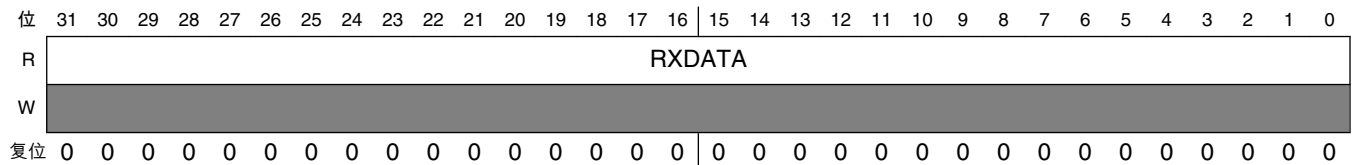
SPIx_PUSHR_SLAVE 字段描述

字段	描述
31-16 保留	
TXDATA	发送数据 存储和当前命令相关的 SPI 传输数据。

43.4.9 POP RX FIFO 寄存器 (SPIx_POPR)

POPR 用于读取 RX FIFO。对 POPR 的 8 位或 16 位读取访问在 RX FIFO 上于 32 位读取访问具有相同效果。对该寄存器的写入将生成传输错误。

地址: 基址 基准 + 38h 偏移



SPIx_POPR 字段描述

字段	描述
RXDATA	接收数据 包含来自读取下一数据指针指向的 RX FIFO 的 SPI 数据。

43.4.10 发送 FIFO 寄存器 (SPIx_TXFRn)

通过 TXFRn 寄存器可以深入了解 TX FIFO 调试用途。每个寄存器是 TX FIFO 中的一个条目。寄存器位只读，无法修改。读取 TXFRx 寄存器不改变 TX FIFO 的状态。

地址: 基址 + 3Ch 偏移 + (4d × i), 其中 i=0d 到 3d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
R	TXCMD_TXDATA																TXDATA																				
W	[Grayed out]																																				
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_TXFRn 字段描述

字段	描述
31–16 TXCMD_ TXDATA	发送命令或发送数据 在主机模式下，TXCMD 字段包含设置 SPI 数据传输属性的命令。在从机模式下，该字段保留。
TXDATA	发送数据 包含要移出的 SPI 数据。

43.4.11 接收 FIFO 寄存器 (SPIx_RXFRn)

通过 RXFRn 寄存器可以深入了解 RX FIFO 调试用途。每个寄存器是 RX FIFO 中的一个条目。RXFR 寄存器为只读。读取 RXFRx 寄存器不改变 RX FIFO 的状态。

地址: 基址 + 7Ch 偏移 + (4d × i), 其中 i=0d 到 3d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
R	RXDATA																																				
W	[Grayed out]																																				
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_RXFRn 字段描述

字段	描述
RXDATA	接收数据 包含接收的 SPI 数据。

43.5 功能说明

此模块支持芯片与外围设备之间的全双工同步串行通信。SPI 配置利用一个移位寄存器和部分可编程传输属性以串行方式传输数据。

该模块有以下配置

- 模块可充当一个基本 SPI 或一个队列 SPI。

模块配置寄存器(MCR)中的 DCONF 字段决定模块配置。当 SPIx_MCR 中的 DCONF 为 0b00 时，将选择 SPI 配置。

CTARn 寄存器保存着时钟和传输属性。SPI 配置允许通过设置 SPI 命令中的一个字段，逐帧选择要使用的 CTAR。

有关 CTAR 寄存器字段的信息，请参阅[时钟和传输属性寄存器（主机模式下）\(SPI_CTARn\)](#)。

下图所示为典型的主机至从机连接。在执行数据传输操作时，将按序把数据移动预定的位数。由于各模块是相连的，所以在主机和从机之间要交换数据。本来在主机移位寄存器中的数据现在则在从机移位寄存器中，反之亦然。在传输结束时，移位寄存器(SR)中的传输控制标志(TCF)位置位，以指示帧传输已完成。

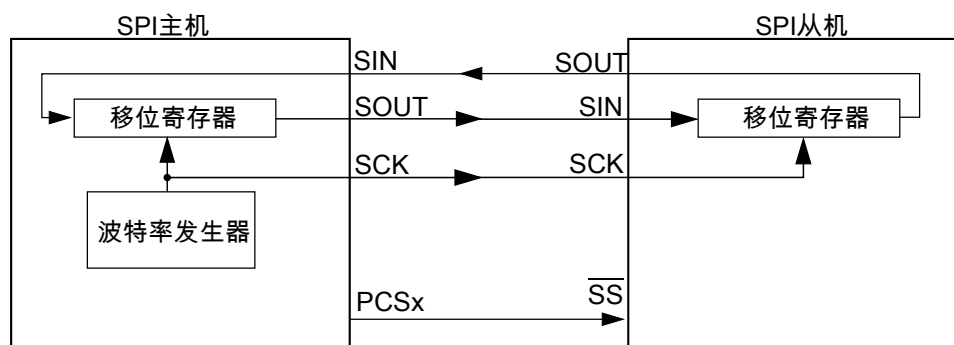


图 43-69. 串行协议概述

一般来说，可以将一个以上的从机器件连接至主机。6 个片选信号 (PCS) 可用于选择要与其通信的从机。请参阅芯片特性部分，了解本芯片中所用 PCS 信号的数量信息。

SPI 配置共享传输协议和时序属性，相关信息详见[传输格式](#)，其描述独立于配置说明。传输速率和延迟设置见[模块波特率和时钟延迟生成](#)。

43.5.1 模块传输的开始和停止

模块有两个工作状态：停止和运行。两个状态均与其配置无关。模块的默认状态为停止。处于停止状态时，在主机模式下，不会发起串行传输；在从机模式下，不会对传输作出响应。停止状态也是对模块各种配置寄存器进行写入操作的安全状态，不会导致未知结果。在运行状态下，会发生串行传输。

SR 中的 TXRXS 位指示模块的状态。如果模块处于运行状态，此位置位。

当所有下列条件为真时，模块开始或转换至运行状态：

- SR[EOQF]位已清零
- 芯片未处于调试模式或者 MCR[FRZ]位已清零
- MCR[HALT]位已清零

当存在下列任一条件时，模块将在当前帧之后停止或者从运行转换为停止状态：

- SR[EOQF]位已置位
- 芯片处于调试模式且 MCR[FRZ]位已置位
- MCR[HALT]位已置位

如果有传输正在进行，则会在下一个帧边界从运动转换为停止状态；如果无正在进行的传输，则立即转换。

43.5.2 串行外设接口(SPI)配置

SPI 配置利用一个移位寄存器和部分可编程传输属性以串行方式传输数据。MCR 中的 DCONF 字段为 0b00 时，模块处于 SPI 配置。SPI 帧的长度可以为 32 位。主机 CPU 或 DMA 控制器将 SPI 数据从模块外部的 RAM 传输至一个 TX FIFO 缓冲器。接收的数据存储在 RX FIFO 缓冲器的条目中。主机 CPU 或 DMA 控制器将收到的数据从 RX FIFO 传输至模块外部的存储器。FIFO 缓冲器的工作方式见下面各节：

- [发送先进先出\(TX FIFO\)缓冲机制](#)
- [接收先进先出\(RX FIFO\)缓冲机制](#)

中断和 DMA 请求条件见 [中断/DMA 请求](#)。

SPI 配置支持两种模式——主机模式和从机模式。在主机模式下，模块根据正在执行的 SPI 命令的字段设置，发起并控制传输。在从机模式下，模块只对其外部总线主机发起的传输作出响应，并且 SPI 命令字段空间保留。

43.5.2.1 主机模式

在 SPI 主机模式下, 模块通过控制 SCK 和 PCS 信号发起串行传输。正在执行的 SPI 命令决定用哪些 CTAR 来设定传输属性, 以及使能哪些 PCS 信号。命令字段也包括有助于队列管理和传输协议的各种位。有关 SPI 命令字段的详情, 请参阅[主机模式下的 PUSH TX FIFO 寄存器 \(SPI_PUSHR\)](#)。正在执行的 TX FIFO 条目中的数据将载入移位寄存器并在串行输出(SOOUT)引脚上移出。在 SPI 主机模式下, 要发送的每个 SPI 帧都有一个关联命令, 允许逐帧进行传输属性控制。

43.5.2.2 从机模式

在 SPI 从机模式下, 模块会对 SPI 总线主机发起的传输作出响应。模块不会发起传输。必须设定某些传输属性 (如时钟极性、时钟相位和帧大小), 才能确保与 SPI 主机成功通信。SPI 从机模式传输属性在 CTAR0 中设定。移出数据时遵循 MSB 优先原则。此模式下不支持移出 LSB。

43.5.2.3 FIFO 禁用操作

FIFO 禁用机制允许在不使用 TX FIFO 或 RX FIFO 的情况下进行 SPI 传输。当 FIFO 禁用时, 模块作为一个简化的双缓冲器 SPI。FIFO 的发送和接收端分别禁用。置位 MCR[DIS_TXF]位会禁用 TX FIFO, 置位 MCR[DIS_RXF]位则会禁用 RX FIFO。

FIFO 禁用机制对用户和主机软件是透明的。发送数据和命令写入 PUSHR, 接收的数据则从 POPR 读取。

当 TX FIFO 被禁用时:

- SR[TFFF]、SR[TFUF]和 SR[TXCTR]的行为就如存在一个单一条目的 FIFO 一样
- TXFR、SR[TXNXTPTR]的内容未定义

类似地, 当 RX FIFO 被禁用时, SR 中的 RFDF、RFOF 和 RXCTR 字段的行为就如存在一个单一条目的 FIFO 一样, 但 RXFR 寄存器和 POPNXTPTR 的内容未定义。

43.5.2.4 发送先进先出(TX FIFO)缓冲机制

TX FIFO 用作 SPI 数据传输的缓冲区。TX FIFO 保存着 4 字, 每个字均由 SPI 数据构成。TX FIFO 中的条目数量因器件而异。通过写入模块 PUSH FIFO 寄存器 (PUSHR) 的数据字段, 向 TX FIFO 添加 SPI 数据。只能通过移出条目或刷新 TX FIFO, 从 TX FIFO 中删除 TX FIFO 条目。

模块状态寄存器 (SR) 中的 TX FIFO 计数器字段 (TXCTR) 指示 TX FIFO 中有效条目的数量。每当将 8 位或 16 位数据写入 PUSH_R[TXDATA] 或将 SPI 数据从 TX FIFO 传输至移位寄存器时, TXCTR 更新。

TXNXTPTR 字段指示将在下一次传输过程中发送的 TX FIFO 条目。每当将 SPI 数据从 TX FIFO 传输至移位寄存器时, TXNXTPTR 字段都会递增。该字段的最大值等于最大 TXFR 编号, 到达最大值时将重新开始。

43.5.2.4.1 填充 TX FIFO

主机软件或 DMA 可以通过写入 PUSH_R, 向 TX FIFO 添加条目。当 TX FIFO 未滿时, SR 中的 TX FIFO 填充标志(TFFF)置位。在 TX FIFO 已滿且 DMA 控制器指示对 PUSH_R 的写入操作完成时, TFFF 位清零。将'1'写入 TFFF 位也会清零该位。TFFF 可以生成 DMA 请求或中断请求。详见[发送 FIFO 填充中断或 DMA 请求](#)。

模块会忽略在 TX FIFO 已滿的情况下, 写入的数据。并且 TX FIFO 的状态不变, 也不会出错。

43.5.2.4.2 耗尽 TX FIFO

通过移位寄存器移出 SPI 数据, 可以删除 TX FIFO 条目(耗尽)。将条目从 TX FIFO 传输至移位寄存器, 并且在 TX FIFO 中存在有效条目的情况下移出。每当将一个条目从 TX FIFO 传输至移位寄存器时, TX FIFO 计数器都会减 1。在传输结束时, SR 中的 TCF 位置位, 以指示传输完成。通过将'1'写入 MCR 中的 CLR_TXF 位, 清除 TX FIFO。

如果外部总线主机发起与模块从机的传输操作, 同时该从机的 TX FIFO 为空, 则从机 SR 中的发送 FIFO 下溢标志(TFUF)置位。详见[发送 FIFO 下溢中断请求](#)。

43.5.2.5 接收先进先出(RX FIFO)缓冲机制

RX FIFO 充当在 SIN 引脚上接收的数据的缓冲器。RX FIFO 保存着 4 个接收的 SPI 数据帧。RX FIFO 中的条目数量因器件而异。当移位寄存器中接收的数据被传输进 RX FIFO 时, 在传输完成时, SPI 数据将添加至 RX FIFO。通过读取模块 POP RX FIFO 寄存器(POPR), 可从 RX FIFO 中删除(取出) SPI 数据。只能通过读取 POPR 或清除 RX FIFO, 从 RX FIFO 中删除 RX FIFO 条目。

模块状态寄存器 (SR) 中的 RX FIFO 计数器字段 (RXCTR) 指示 RX FIFO 中有效条目的数量。每当 POPR 被读取时或者 SPI 数据被从移位寄存器复制至 RX FIFO 时, RXCTR 更新。

SR 中的 POPNXTPTR 字段指向读取 POPR 时返回的 RX FIFO 条目。POPNXTPTR 含有若干个 32 位寄存器中来自 RXFR0 的正偏移量。例如，POPNXTPTR 等于 2 表示，RXFR2 含有将在读取 POPR 时返回的已接收 SPI 数据。每当 POPR 被读取时，POPNXTPTR 字段都会递增。该字段的最大值等于实现的最大 RXFR 编号，到达最大值时将重新开始。

43.5.2.5.1 填充 RX FIFO

RX FIFO 中填充的是来自移位寄存器的已接收 SPI 数据。当 RX FIFO 未滿时，来自移位寄存器的 SPI 帧会被传输至 RX FIFO。每当一个 SPI 帧被传输至 RX FIFO 时，RX FIFO 计数器都会加 1。

如果 RX FIFO 和移位寄存器已滿并且已经发起传输，则 SR 中的 RFOF 位置位，指示存在溢出条件。根据 MCR 中 ROOE 位的状态，生成溢出的传输数据被忽略或被转移到移位寄存器。如果 ROOE 位置位，输入数据将被转移到移位寄存器中。如果 ROOE 位清零，输入数据将被忽略。

43.5.2.5.2 耗尽 RX FIFO

主机 CPU 或 DMA 可以通过读取模块 POP RX FIFO 寄存器 (POPR) 从 RX FIFO 删除 (弹出) 条目。一个 POPR 读数使 RX FIFO 计数器减一。从空 RX FIFO 弹出数据的尝试被忽略，且 RX FIFO 计数器保持不变。从空 RX FIFO 读取的数据不确定。

当 RX FIFO 不为空时，SR 中的 RX FIFO 耗尽标志 (RFDF) 置位。当 RX_FIFO 为空且 DMA 控制器指示 POPR 的一次读取成功后，或者通过向 RFDF 位写入 1，RFDF 位清零。

43.5.3 模块波特率和时钟延迟生成

串行传输的 SCK 频率和延迟值的方法是通过一个预分频器和可选的倍频器将系统时钟频率分频得到。下图在概念上展示了 SCK 信号的生成方式。



图 43-70. 通信时钟预分频器和分频器

43.5.3.1 波特率生成器

波特率是 SCK 的频率。协议时钟由预分频器 (PBR) 和分频器 (BR) 分频以生成 SCK，可能将分频器分频减少一半。CTAR 中的 DBR、PBR 和 BR 字段通过 BR 字段描述中的公式选择 SCK 频率。下表所示位计算波特率的方法示例。

表 43-102. 波特率计算示例

f_p	PBR	预分频器	BR	分频器	DBR	波特率
100 MHz	0b00	2	0b0000	2	0	25 Mb/s
20 MHz	0b00	2	0b0000	2	1	10 Mb/s

注

前面表格中提到的时钟频率只是一个例子。要了解用于驱动器件中此模块的频率，请参见“时钟”一章。

43.5.3.2 PCS 至 SCK 延迟(t_{csc})

PCS 至 SCK 延迟为从 PCS 信号有效到第一个 SCK 边沿的时长。参见图 43-72，了解 PCS 至 SCK 延迟的说明。CTAR_x 寄存器中的 PCSSCK 和 CSSCK 字段通过 CSSCK 字段描述中的公式选择 PCS 至 SCK 延迟。下表所示为计算 PCS 至 SCK 延迟的方法示例。

表 43-103. PCS 至 SCK 延迟计算示例

f_{sys}	PCSSCK	预分频器	CSSCK	分频器	PCS 至 SCK 延迟
100 MHz	0b01	3	0b0100	32	0.96 μ s

注

前面表格中提到的时钟频率只是一个例子。要了解用于驱动器件中此模块的频率，请参见“时钟”一章。

43.5.3.3 SCK 后延迟 (t_{Asc})

SCK 后延迟是 SCK 最后边沿和 PCS 失效之间的时长。请参见图 43-72 和图 43-73 了解 SCK 后延迟的说明。CTAR_x 寄存器中的 PASC 和 ASC 字段通过 ASC 字段描述中的公式选择 SCK 后延迟。下表所示为计算 SCK 后延迟的方法示例。

表 43-104. SCK 后延迟计算示例

f_p	PASC	预分频器	ASC	分频器	SCK 后延迟
100 MHz	0b01	3	0b0100	32	0.96 μ s

注

前面表格中提到的时钟频率只是一个例子。要了解用于驱动器件中此模块的频率，请参见“时钟”一章。

43.5.3.4 传输后延迟 (t_{DT})

传输后延迟是某帧 PCS 信号无效与下一帧 PCS 信号有效之间的最短时间。有关传输后延迟的说明，请参见图 43-72。CTAR_x 寄存器中的 PDT 和 DT 字段通过 DT 字段描述中的公式选择传输后延迟。下表所示为计算传输后延迟的方法示例。

表 43-105. 传输后延迟计算示例

f_p	PDT	预分频器	DT	分频器	传输后延迟
100 MHz	0b01	3	0b1110	32768	0.98 ms

注

前面表格中提到的时钟频率只是一个例子。要了解用于驱动器件中此模块的频率，请参见“时钟”章节。

当在非连续时钟模式下时， t_{DT} 延迟根据 CTAR[DT] 字段描述中指定的等式配置。当在连续时钟模式下时，延迟固定在 1 个 SCK 周期。

43.5.3.5 外设芯片选择选通使能(PCSS)

$\overline{\text{PCSS}}$ 信号提供一个延迟，以允许 PCS 信号在转换之后得以建立，从而避免出现毛刺。当模块处于主机模式且 MCR 中的 PCSSE 位置位时， $\overline{\text{PCSS}}$ 将提供一个信号，供外部多路解调器将 PCS5 以外的外设芯片选择解码为无毛刺的 PCS 信号。下图所示为 $\overline{\text{PCSS}}$ 信号相对于 PCS 信号的时序信息。

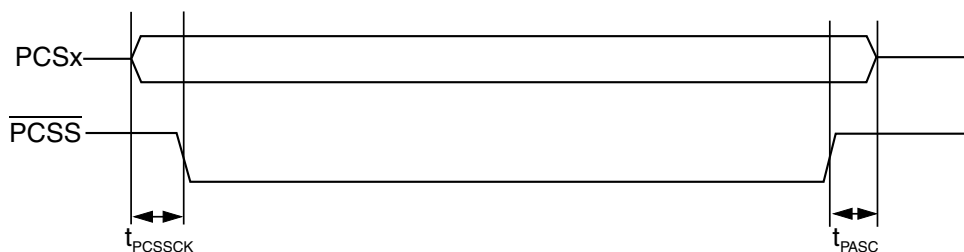


图 43-71. 外设芯片选择选通时序

PCS 信号认定与 $\overline{\text{PCSS}}$ 认定之间的延迟由 CTAR 中的 PCSSCK 字段根据以下公式选择：

$$t_{\text{PCSSCK}} = \frac{1}{f_{\text{P}}} \times \text{PCSSCK}$$

传输结束时， $\overline{\text{PCSS}}$ 取反与 PCS 取反之间的延迟由 CTAR 中的 PASC 字段根据以下公式选择：

$$t_{\text{PASC}} = \frac{1}{f_{\text{P}}} \times \text{PASC}$$

下表所示为计算 t_{pcssck} 延迟的方法示例。

表 43-106. 外设芯片选择选通认定计算示例

f_{P}	PCSSCK	预分频器	传输前延迟
100 MHz	0b11	7	70.0 ns

下表所示为计算 t_{pasc} 延迟的方法示例。

表 43-107. 外设芯片选择选通取反计算示例

f_{P}	PASC	预分频器	传输后延迟
100 MHz	0b11	7	70.0 ns

在连续串行通信 SCK 模式使能时，不支持 $\overline{\text{PCSS}}$ 信号。

注

前面的表格中提到的时钟频率只是一个例子。要了解用于驱动器件中此模块的频率，请参见“时钟”一章。

43.5.4 传输格式

SPI 串行通信由串行通信时钟 (SCK) 信号和 PCS 信号控制。由主机器件提供的 SCK 信号同步 SIN 和 SOUT 引脚上数据的移位和采样。PCS 信号充当从机器件的使能信号。

在主机模式下，时钟和转换属性寄存器 (CTARn) 中的 CPOL 和 CPHA 位选择串行时钟 SCK 的极性和相位。

- CPOL——选择 SCK 的空闲状态极性
- CPHA——选择 SOUT 上的数据在第一个 SCK 边沿之前或其上是否有效

尽管总线从机不控制 SCK 信号，但在从机模式下，CPOL 和 CPHA 的值必须与主机器件设置相同，才能确保传输的正确进行。在 SPI 从机模式下，只会使用 CTAR0。

此模块支持四种不同的传输格式：

- 经典 SPI，CPHA=0
- 经典 SPI，CPHA=1
- 改进型传输格式，CPHA = 0
- 改进型传输格式，CPHA = 1

支持改进型传输格式，以便在要求较长建立时间的外设之间实现高速通信。模块可以在晚于半个周期时对输入数据采样，以便为外设提供更多的建立时间。MCR 中的 MTFE 位在经典 SPI 格式与改进型传输格式之间进行选择。

在接口配置中，模块提供了一个选项——使 PCS 信号在帧之间保持有效状态。详见[连续选择格式](#)。

43.5.4.1 典型 SPI 传输格式 (CPHA = 0)

下图中显示的传输格式用于与外设 SPI 从机设备通信，其中首个数据位在首个时钟边沿可用。在这种格式中，主机和从机设备在奇数编号的 SCK 边沿上采样它们的 SIN 引脚，在偶数编号的 SCK 边沿上更改它们 SOUT 引脚上的数据。

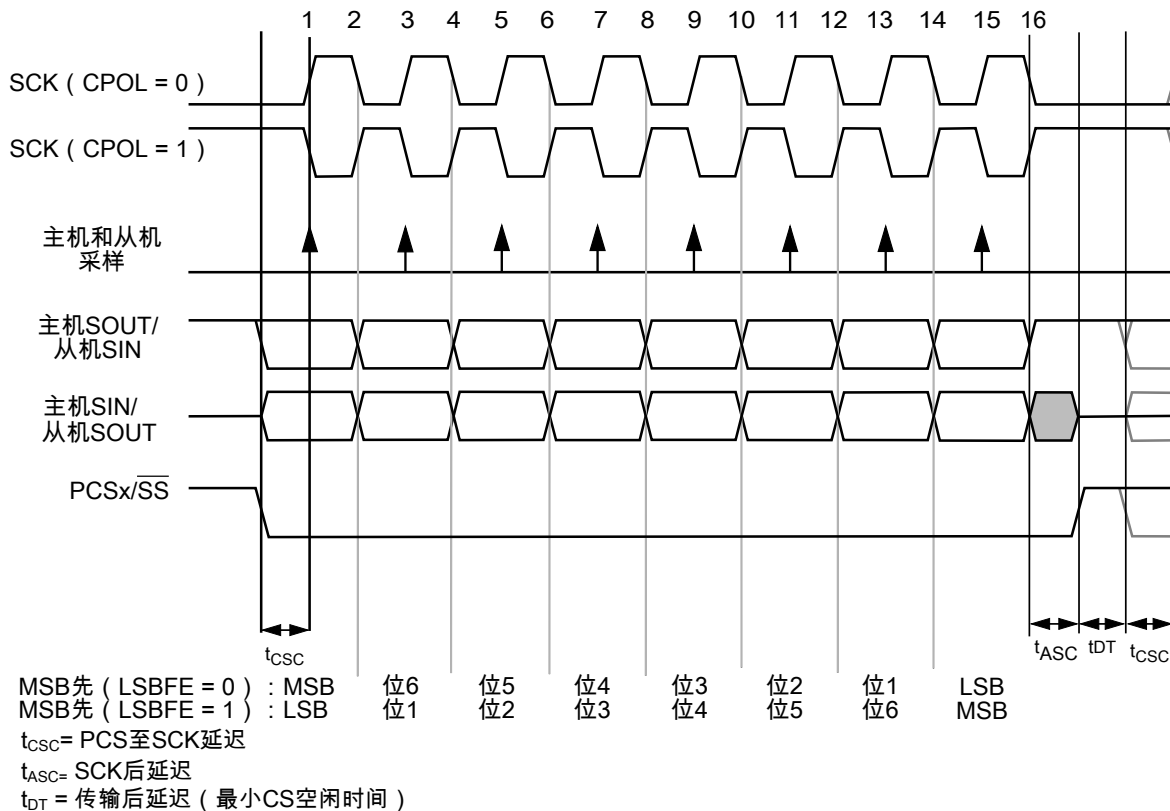


图 43-72. 模块传输时序图 (MTFE=0, CPHA=0, FMSZ=8)

主机将第一个数据位置于 SOUT 引脚上, 并使能适当的外设芯片选择信号到从机设备, 从而发起传输。从机将首个数据位置于其 SOUT 引脚上以响应。 t_{CSC} 延迟结束后, 主机输出首个 SCK 边沿。主机和从机设备使用此边沿采样它们串行数据输入信号上的首个输入数据位。在第二个 SCK 边沿处, 主机和从机设备将第二个数据位置于它们的串行数据输出信号上。剩余帧中, 主机和从机在奇数编号的时钟边沿采样它们的 SIN 引脚, 在偶数编号的时钟边沿更改它们 SOUT 引脚上的数据。最后时钟边沿发生后, 在主机取反 PCS 信号前插入 t_{ASC} 延迟。在新帧传输可以通过主机发起前, 插入 t_{DT} 延迟。

43.5.4.2 典型 SPI 传输格式 (CPHA = 1)

下图中显示的传输格式用于与外设 SPI 从机设备通信, 这需要首个数据位前的首个 SCK 边沿在从机 SOUT 引脚上变为可用。在这种格式中, 主机和从机设备在奇数编号的 SCK 边沿上更改它们 SOUT 引脚上的数据, 在偶数编号的 SCK 边沿上采样它们 SIN 引脚上的数据。

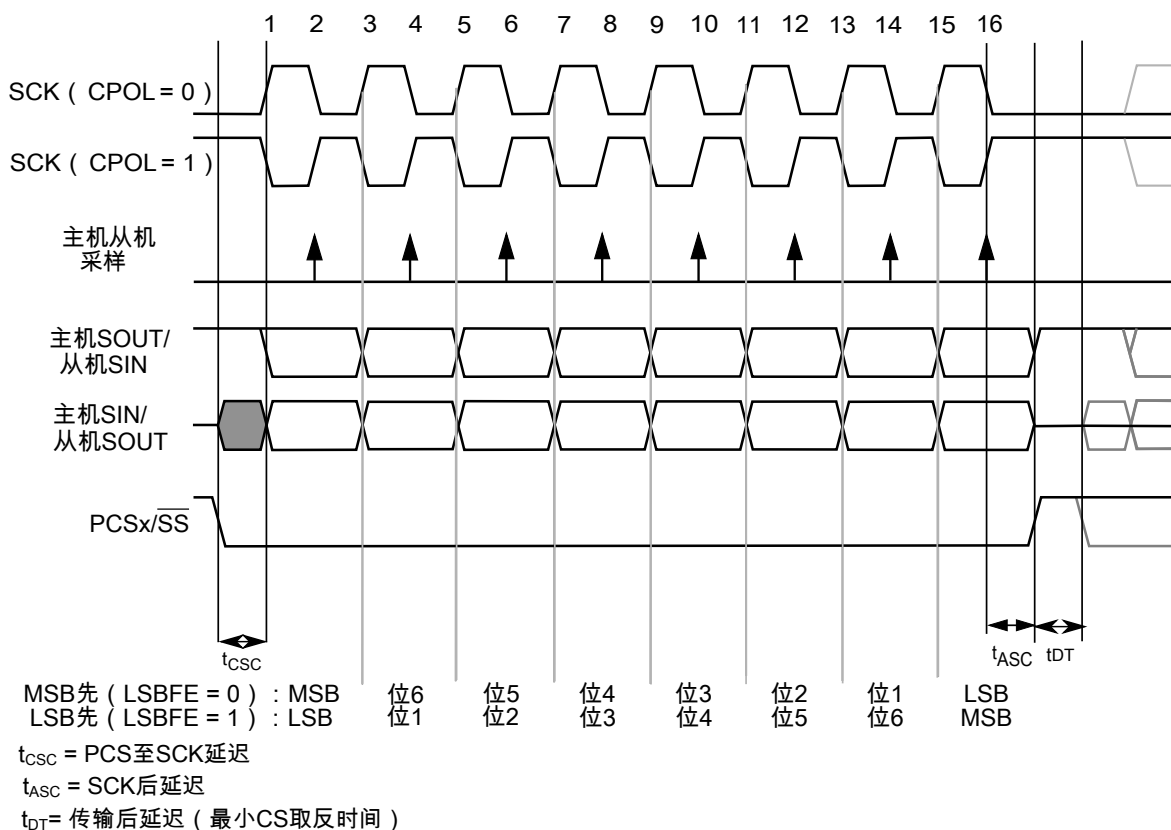


图 43-73. 模块传输时序图 (MTFE=0, CPHA=1, FMSZ=8)

主机通过使能 PCS 信号到从机发起传输。 t_{CSC} 延迟结束后，主机生成首个 SCK 边沿，同时将有效值置于主机 SOUT 引脚上。从机将其首个数据位置于其从机 SOUT 引脚上，响应首个 SCK 边沿。

主机和从机在 SCK 的第二个边沿采样它们的 SIN 引脚。剩余帧中，主机和从机在奇数编号的时钟边沿更改它们 SOUT 引脚上的数据，在偶数编号的时钟边沿采样它们的 SIN 引脚。最后时钟边沿发生后，在主机取反 PCS 信号前插入 t_{ASC} 延迟。在新帧传输可以通过主机发起前，插入 t_{DT} 延迟。

43.5.4.3 改进型 SPI 传输格式(MTFE = 1, CPHA = 0)

在改进型传输格式中，主机和从机在 SCK 周期中的采样时间都晚于经典 SPI 模式，使逻辑容许器件焊盘与电路板走线中存在更多延迟。当 SCK 周期随着波特率的增加而缩短时，这些延迟在 SCK 周期中所占份量不断加大。

在 PCS 信号有效时，主机和从机把数据置于 SOUT 引脚上。在 PCS 至 SCK 的延迟到期时，将生成第一个 SCK 边沿。从机在每个奇数编号的 SCK 边沿对主机 SOUT 信号采样。在从机模式下，当 MTFE 位置位时，DSPI 也会在每个奇数编号的时钟边沿把数据置于从机 SOUT 上。常规外部从机 (配置为 CPHA=0 格式) 在每个偶数编号的 SCK 时钟边沿驱动其 SOUT 输出。

如果协议时钟频率与 SCK 频率之比大于 3，DSPI 主机将在奇数编号 SCK 边沿一个协议时间之后把第二数据位置于 SOUT 线路上。如果该比值低于 4，则主机将在奇数编号 SCK 边沿更改 SOUT。主机对 SIN 进行采样的点由 DSPI_MCR[SMPL_PT] 字段选择。主机采样点可以延迟一个或两个协议时钟周期。如果协议与 SCK 频率之比小于 4，SMPL_PT 字段应置为 0。然而，如果该比值小于 4，根据设计，实际采样点会自动延迟一个协议时钟周期。

以下时序图展示了 DSPI 在 MTFE=1 时的工作情况。所示时序延迟为：

- T_{csc} ——PCS 至 SCK 认定延迟
- T_{acs} ——SCK PCS 取反后延迟
- T_{su_ms} ——主机 SIN 建立时间
- T_{hd_ms} ——主机 SIN 保持时间
- T_{vd_st} ——从机数据输出有效时间，从机数据输出 SCK 驱动边沿与数据变成有效之间的时间。
- T_{su_st} ——从机数据输入上的数据建立时间
- T_{hd_st} ——从机数据输入上的数据保持时间
- T_{sys} ——协议时钟周期。

下图所示为 CPHA = 0 且 Fsys/Fsck = 4 条件下的改进型传输格式。只展示了 CPOL = 0 时的条件。实线三角形表示数据采样时钟边沿。图中显示了从机的两种可能行为。

- 信号（标记为“外部从机的 SOUT”）呈现出典型的 SPI 从机串行输出。
- 信号（标记为“DSPI 从机的 SOUT”）表示 DSPI 在 MTFE 位置位时，在从机模式下的情况。

在其他 MTFE = 1 条件下的示意图中，DSPI SIN 输入由一个常规外部 SPI 从机驱动，其配置依据为 DSPI 主机 CPHA 编程情况。

注

在下面的图中， f_{sys} 表示产生波特频率 f_{sck} 的协议时钟频率。

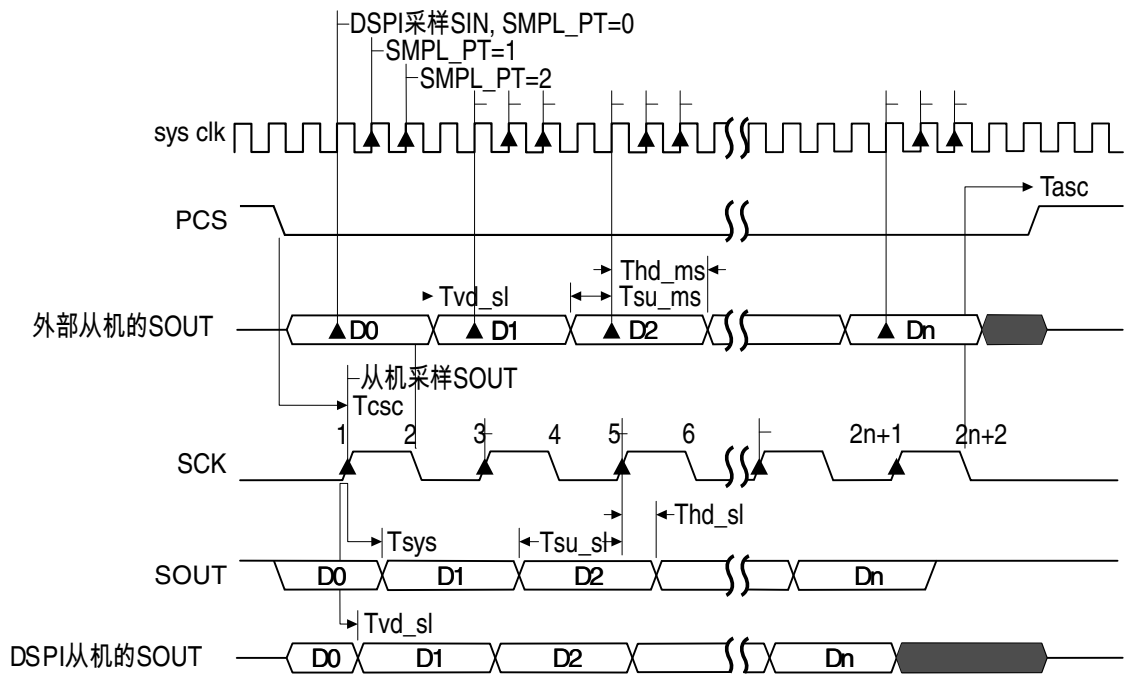


图 43-74. DSPI 改进型传输格式(MTFE=1, CPHA=0, $f_{sck} = f_{sys}/4$)

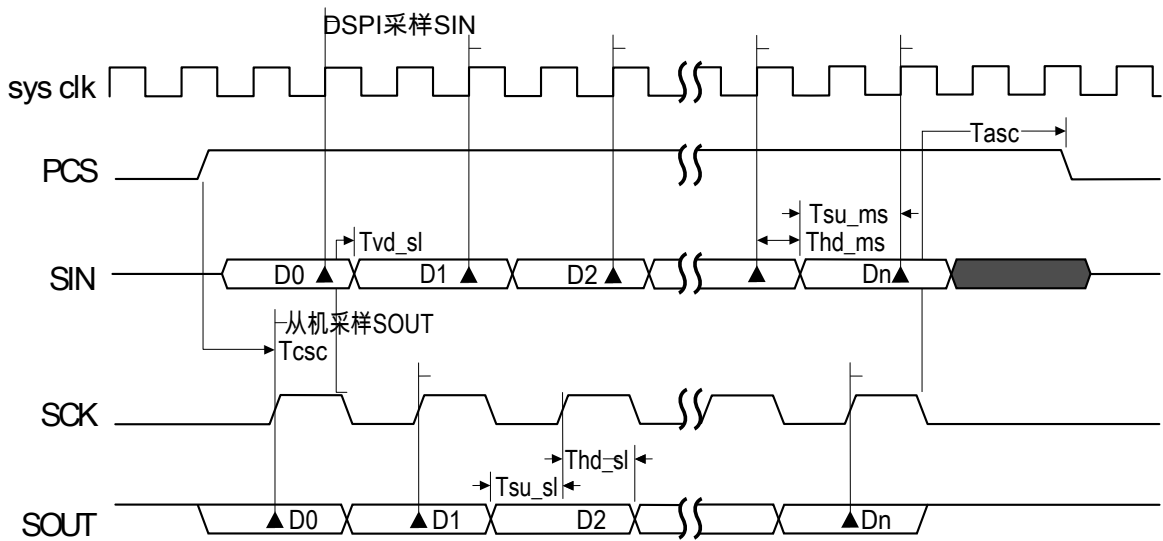
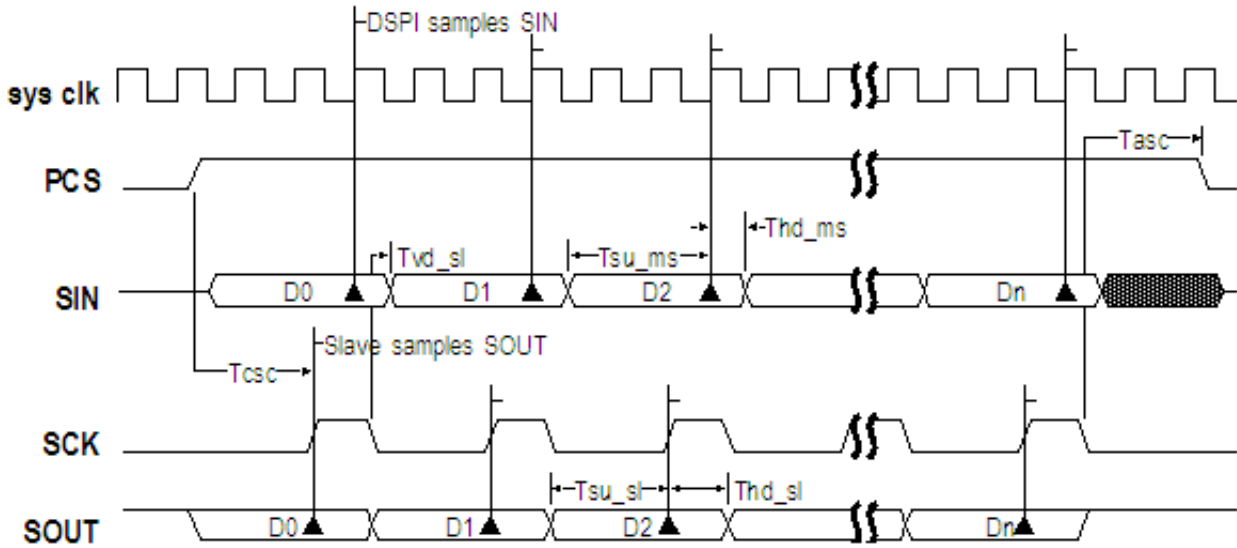


图 43-75. 改进型传输格式(MTFE=1, CPHA=0, $f_{sck} = f_{sys}/2$)

图 43-76. 改进型传输格式(MTFE=1, CPHA=0, $f_{sck} = f_{sys}/3$)

43.5.4.4 改进型 SPI 传输格式(MTFE = 1, CPHA = 1)

下面的图所示为 CPHA = 1 条件下的改进型传输格式。只展示了 CPOL = 0 时的条件。在传输开始时，DSPI 使能至从机器件的 PCS 信号。在 PCS 至 SCK 延迟结束时，主机和从机会在 SCK 的第一个边沿，把数据置于其 SOUT 引脚上。从机在偶数编号的 SCK 边沿对主机 SOUT 信号采样。主机在从第三个 SCK 边沿开始的奇数编号 SCK 边沿对从机 SOUT 信号采样。从机对 SCK 最后边沿上的最后一位采样。主机在 SCK 最后边沿半个 SCK 周期之后，对最后从机 SOUT 位采样。在对最后位采样时，在主机 SCK 引脚上将看不到时钟边沿。SCK 至 PCS 延迟和 SCK 后延迟必须大于或等于 SCK 周期的一半。

注

在主机模式下连续 SCK 被使能(MCR [CONT_SCKE] =1)且 MTFE=1 时，配置 CTAR[LSBFE]=0，以便在接收长度不相等的帧时能正确工作。如果 PUSHR[CONT]也针对背靠背帧传输设置，还要将第一帧的大小配置为小于或等于下一帧的大小。在这种情况下，要确保对所有已接收帧来说，读取位等于各自的帧大小，并且将 POP 操作过程中的任何额外位都屏蔽起来。

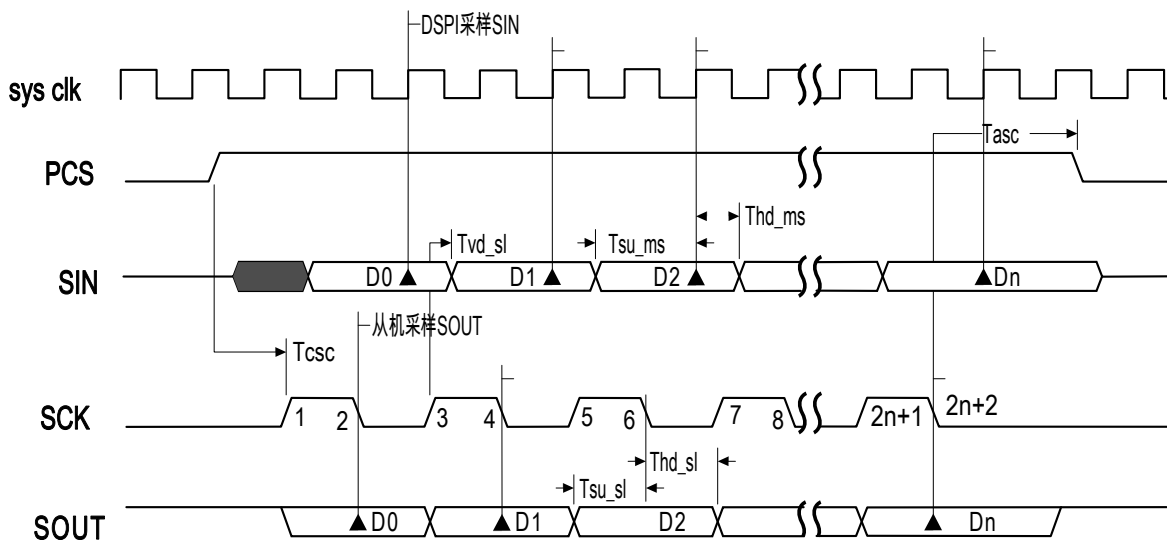


图 43-77. 改进型传输格式(MTFE=1, CPHA=1, $f_{sck} = f_{sys}/2$)

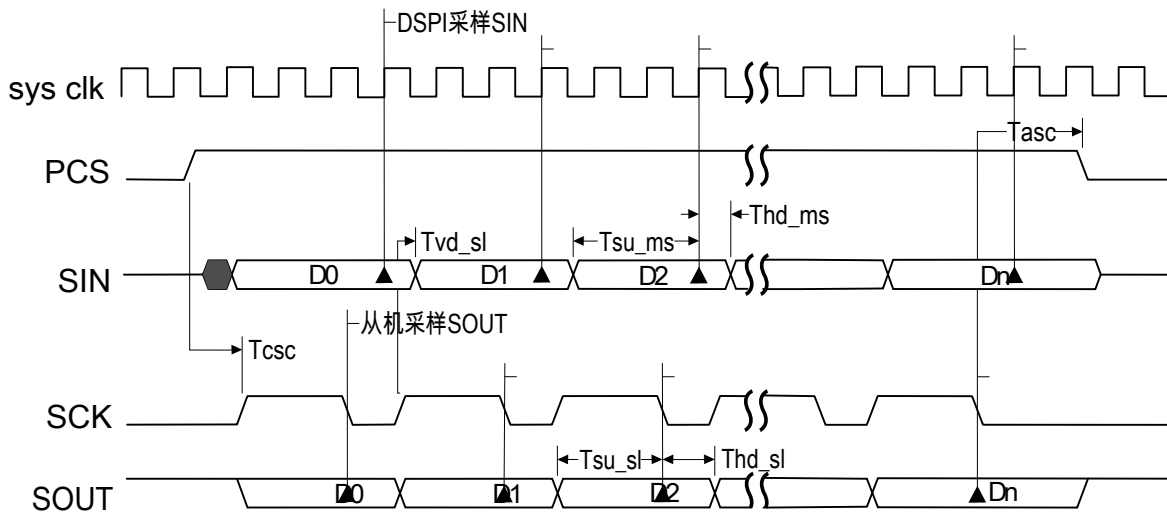
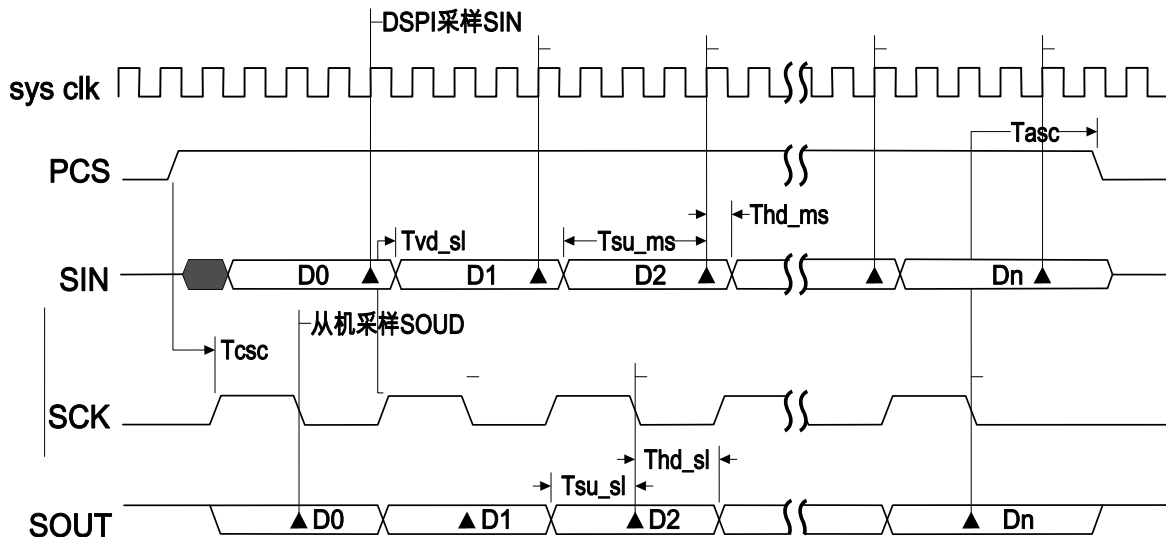


图 43-78. 改进型传输格式(MTFE=1, CPHA=1, $f_{sck} = f_{sys}/3$)

图 43-79. 改进型传输格式(MTFE=1, CPHA=1, $f_{sck} = f_{sys}/4$)

43.5.4.5 连续选择格式

对于一些外设来说，每个传输之间，必须取消片选。其他的外设则需要保持片选有效。连续选择格式提供处理以下案例的灵活性。通过在 SPI 命令中置位 CONT 位，可为 SPI 配置使能连续选择格式。

当 CONT 位 = 0 时，在帧之间模块驱动片选信号至无效。芯片选择信号的空闲状态通过 MCR 中的 PCSISn 位选择。以下是 CPHA = 1 和 CONT = 0 的两个四位传输的时序图。

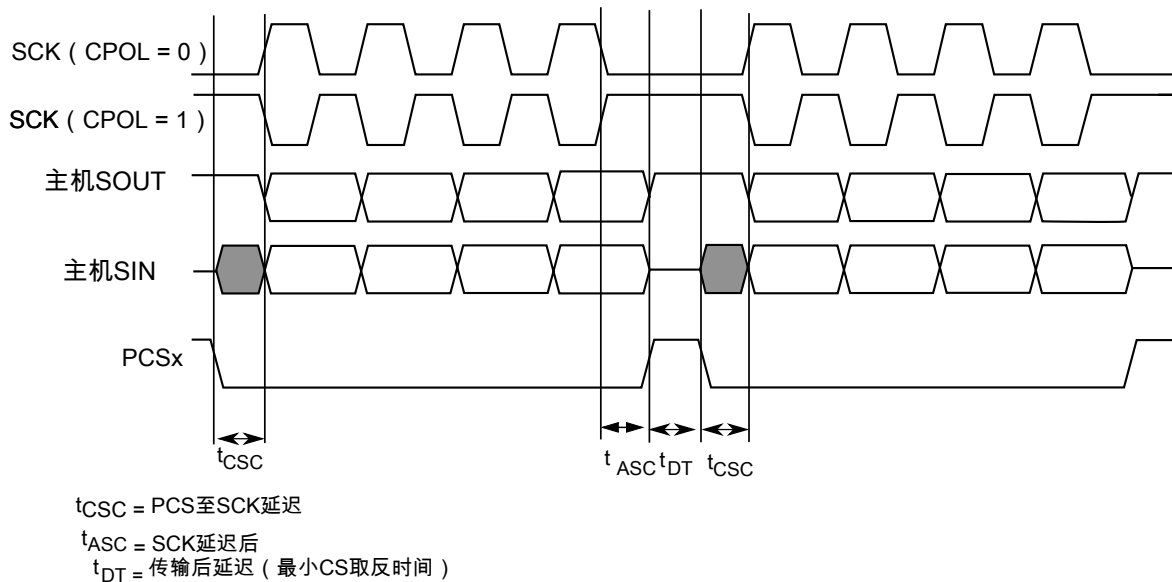


图 43-80. 非连续格式示例 (CPHA=1, CONT=0)

当 CONT 位 = 1 时，PCS 信号对于两个传输的持续时间保持有效。传输之间的延迟 (t_{DT}) 未插入传输之间。下图所示为 CPHA = 1 和 CONT = 1 的两个四位传输的时序图。

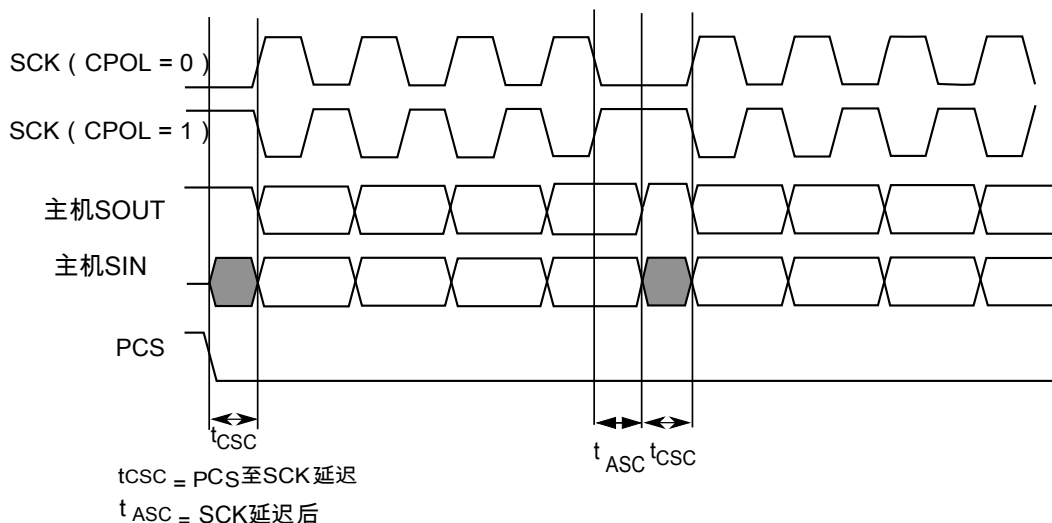


图 43-81. 连续传输示例 (CPHA=1, CONT=1)

使用连续选择模块时遵循以下规则：

- 所有发送命令必须具有相同的 PCSn 位编程。
- 通过发送命令选择的 CTAR 必须使用相同的传输属性编程。这些 CTAR 中只有 FMSZ 字段可以进行不同编程。
- 以此种模式发送多个帧时，在主机模式下用户软件必须确保最后帧解除置位 PUSHR[CONT]位，在从机模式下用户软件必须在 TX_FIFO 中提供足够的发出帧，并且主机模式在最后帧传输结束时使 PCSn 失效。
- 在主机模式下，置位 MCR[HALT]之前必须解除置位 PUSHR[CONT]。这将确保 PCSn 信号无效。连续传输中置位 MCR[HALT]将导致 PCSn 信号保持有效，且从机设备因此无法从运行转换至停止状态。

注

用户必须在 TX FIFO 变空之前，将所需要的数据在一次片选有效时填入 TX FIFO。

以从机模式运行时，请确保 TX FIFO 中最后一个条目完全发送时（即相应 TCF 标志有效且 TXFIFO 为空），取消选择从机便于进一步串行通信；否则，会发生下溢错误。

43.5.5 连续串行通信时钟

模块能够产生连续的时钟，以支持需要连续 SCK 信号的从机外设。

连续 SCK 通过置位 MCR 中的 CONT_SCKE 位使能。只有 MCR[HALT]位为低时，使能该位生成连续 SCK。连续 SCK 在所有配置中有效。

连续 SCK 仅在 CPHA=1 的情况下支持。如果 CONT_SCKE 位置位，清零 CPHA 被忽略。修正传输格式时连续 SCK 受支持。

连续 SCK 模式的时钟和传输属性按照以下规则设置：

- 模块在 SPI 配置中时，最初使用 CTAR0。每个 SPI 帧传输开始时，使用 CTAS 为帧指定的 CTAR。
- 在所有配置中，一直使用当前选定的 CTAR，直到不同的 CTAR 帧开始，或者连续 SCK 模式终止。

建议使用连续 SCK 时保持波特率相同。使用连续 SCK 时在帧之间切换时钟极性可能导致传输错误。如果模块置于外部停止模式或模块禁用模式，无法保障连续 SCK 操作。

使能连续 SCK 禁用 PCS 至 SCK 延迟，且传输后延迟 (t_{DT}) 固定为一个 SCK 周期。下图为连续选择禁用的连续 SCK 格式的时序图。

注

在连续 SCK 模式下，应始终使用 SPI 传输 CTAR0，并且在开始传输前必须使用 MCR[CLR_TXF]字段清零 TX FIFO。

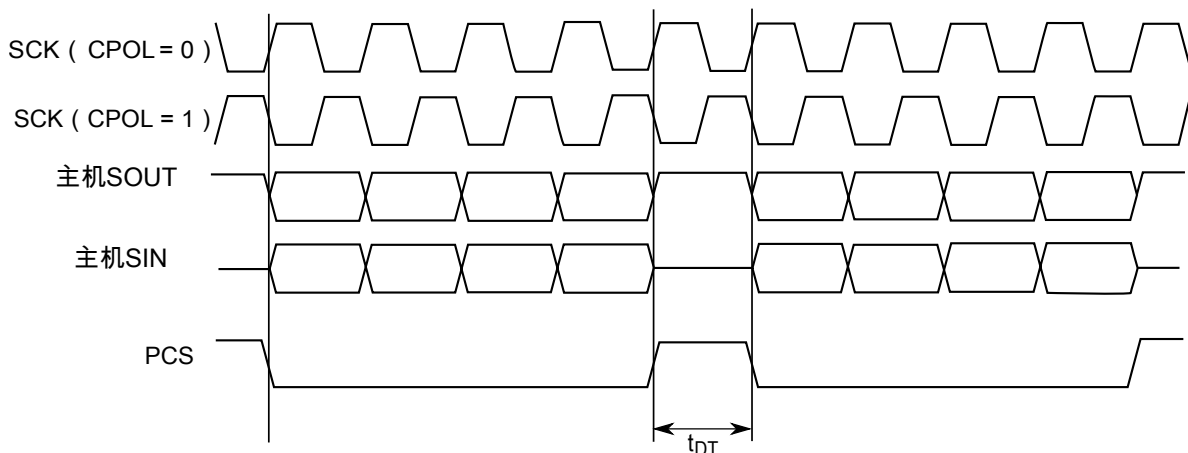


图 43-82. 连续 SCK 时序图 (CONT=0)

如果 TX FIFO 条目中的 CONT 位置位，PCS 在传输之间保持有效。在特定情况下，SCK 可以继续保持 PCS 认定，但是没有数据移出 SOUT，也就是 SOUT 拉至高电平。这可能导致从机接收不正确数据。这些情况包括：

- 连续 SCK 的 CONT 位置位，但 TX FIFO 中没有数据。
- 连续 SCK 的 CONT 位置位，且进入 Stopped 状态（参见[模块传输的开始和停止](#)）。
- 连续 SCK 的 CONT 位置位，且进入 Stop 模式或模块禁用模式。

下图为连续选择使能的连续 SCK 格式的时序图。

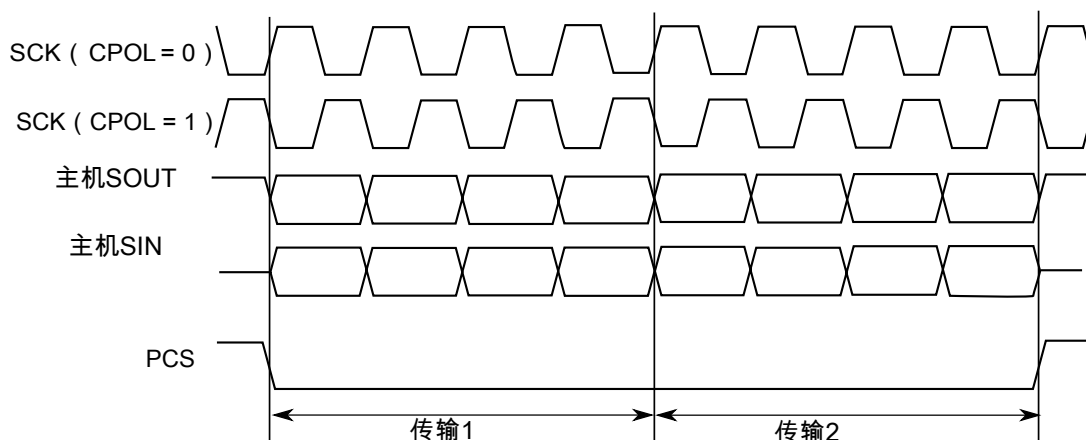


图 43-83. 连续 SCK 时序图 (CONT=1)

43.5.6 从机模式操作限制

从机模式逻辑移位寄存器搭载了缓冲机制。当模块被永久选择并且以恒定速率移入数据时，这样做可以支持数据流操作。

如果 \overline{SS} 信号有效以及每当发送数据准备就绪且 \overline{SS} 信号无效时，发送数据在每帧的第二个 SCK 时钟边沿传输至移位寄存器。

已接收数据在每帧的最后 SCK 边沿转换至接收缓冲器，该边沿由编进 CTAR0/1 寄存器的帧大小定义。然后将来自缓冲器的数据传输至 RXFIFO 寄存器。

如果 \overline{SS} 在该最后 SCK 边沿之前失效，则来自移位寄存器的数据将丢失。

43.5.7 中断/DMA 请求

此模块有多个只能生成中断请求的条件，以及两个可以生成中断或 DMA 请求的条件。下表列出了这些条件。

表 43-108. 中断和 DMA 请求条件

条件	标志	中断	DMA
队列结束(EOQ)	EOQF	是	-
TX FIFO 填充	TFFF	是	是
传输完成	TCF	是	-
TX FIFO 下溢	TFUF	是	-
RX FIFO 耗尽	RFDF	是	是
RX FIFO 溢出	RFOF	是	-

每个条件都在模块状态寄存器(SR)中有一个标志位，在 DMA/中断请求选择与使能寄存器(RSER)中有一个请求使能位。某些标志（如上表所示）会生成中断请求或 DMA 请求，具体取决于 RSER 寄存器的配置。

模块同时提供了一个全局中断请求，该全局中断在任一中断请求有效时置位有效。

43.5.7.1 队列结束中断请求

队列结束(EOQ)中断请求指示已经到达发送队列的末尾。模块将在 EOQ 中断请求使能(RSER[EOQF_RE])且正在执行的 SPI 命令中的 EOQ 位为 1 时生成中断请求。

当 EOQ 位置位且 SPI 帧的最后一位已经发送完毕时，模块会生成中断请求。

43.5.7.2 发送 FIFO 填充中断或 DMA 请求

发送 FIFO 填充请求指示 TX FIFO 未滿。发送 FIFO 填充请求在 TX FIFO 中的条目数少于最大可能条目数且 RSER 中的 TFFF_RE 位置位时生成。RSER 中的 TFFF_DIRS 位选择是生成 DMA 请求，还是生成中断请求。

注

TFFF 标志在 DMA 用于填充 TX FIFO 时自动清零。

要在未使用 DMA 时清零 TFFF，请针对执行的每个 PUSH 操作实施以下步骤，用 CPU 填充 TX FIFO：

1. 等到 TFFF = 1。
2. 用 CPU 向 PUSH 写入数据。
3. 通过向 TFFF 的位置写入 1，清零 TFFF。如果 TX FIFO 未滿，此标志不会清零。

43.5.7.3 传输完成中断请求

传输完成请求指示串行帧传输结束。当 RSER 中的 TCF_RE 位置位时，传输完成请求在每帧传输结束时生成。

43.5.7.4 发送 FIFO 下溢中断请求

发送 FIFO 下溢请求指示 TX FIFO 中出现下溢条件。只会针对工作于从机模式下的模块和 SPI 配置检测发送下溢条件。TFUF 位在模块的 TX FIFO 为空时置位，同时还会从一个外部 SPI 主机发起传输。如果 TFUF 位在 RSER 中的 TFUF_RE 位置位时置位，则会生成一个中断请求。

43.5.7.5 接收 FIFO 耗尽中断或 DMA 请求

接收 FIFO 耗尽请求指示 RX FIFO 非空。接收 FIFO 耗尽请求在 RX FIFO 中的条目数不为零且 RSER 中的 RFDF_RE 位置位时生成。RSER 中的 RFDF_DIRS 位选择是生成 DMA 请求，还是生成中断请求。

43.5.7.6 接收 FIFO 溢出中断请求

接收 FIFO 溢出请求指示 RX FIFO 中出现溢出条件。接收 FIFO 溢出请求在 RX FIFO 和移位寄存器已满且已发起传输时生成。要生成中断请求，RSER 中的 RFOF_RE 位必须置位。

根据 MCR 中 ROOE 位的状态，生成溢出的传输数据被忽略或被转移到移位寄存器。如果 ROOE 位置位，输入数据将被转移到移位寄存器中。如果 ROOE 位清零，输入数据将被忽略。

43.5.8 节电特性

此模块支持以下节电策略：

- 外部停止模式
- 模块禁用模式——对非存储器映射逻辑进行时钟门控

43.5.8.1 停止模式 (外部停止模式)

此模块支持停止模式协议。当要求进入外部停止模式时，模块会确认请求。如果正在进行串行传输，则此模块会一直等到达到帧边界为止，然后再准备好关闭其时钟。当时钟关闭时，将不可访问此模块的存储器映射逻辑。这样也会使模块进入 STOPPED 状态。SR[TXRXS]位清零，以标示 STOPPED 状态。处于外部停止模式时，不能改变中断和 DMA 请求信号的状态。

43.5.8.2 模块禁用模式

模块禁用模式是一种因模块而异的模式，模块可以进入该模式以省电。主机 CPU 可通过置位 MCR 中的 MDIS 位来启动模块禁用模式。模块禁用模式也可通过硬件来启动。

当 MDIS 位置位时，模块会使下一个帧边界处的时钟使能信号无效。时钟使能信号一旦变为无效，则视为已经进入模块禁用模式。这样也会使模块进入 STOPPED 状态。SR[TXRXS]位清零，以标示 STOPPED 状态。如果得以实施，时钟使能信号可以停止用于非存储器映射逻辑的时钟。当时钟使能变为无效时，模块将进入睡眠模式，但存储器映射寄存器仍然可以访问。当模块处于模块禁用模式时，某些读或写操作会有不同的作用。读取 RX FIFO Pop 寄存器不会改变 RX FIFO 的状态。类似地，写入 PUSHR 寄存器也不会改变 TX FIFO 的状态。使任一 FIFO 清零均不会对模块禁用模式产生影响。MCR 的 DIS_TXF 和 DIS_RXF 字段的变更不会影响模块禁用模式。在模块禁用模式下，模块中的所有状态位和寄存器标志都会在读取时返回正确的值，但写入它们并不起作用。在模块禁用模式下写入 TCR 不起作用。在模块禁用模式下不能清除中断和 DMA 请求信号。

43.6 初始化/应用信息

本节介绍了如何初始化模块。

43.6.1 如何管理队列

队列不是模块的一部分，但其中包括的一些特性有利于进行队伍管理。队列主要在 SPI 配置中支持。

1. 当模块执行队列中的最后一个命令字，此命令字中的 EOQ 位将置位，标示此为队列中的最后一个条目。
2. 在传输结束时，带有 EOQ 标志的命令被执行，SR 中的 EOQ 标志(EOQF)将置位。

3. EOQF 标志置位会禁用数据的串行传输和接收，使模块进入 STOPPED 状态。TXRXS 位清零，以标示 STOPPED 状态。
4. DMA 可以继续填充 TX FIFO，直至填满或步骤 5 发生为止。
5. 通过禁用为分配给 TX FIFO 和 RX FIFO 的 DMA 通道的 DMA 使能请求，禁用 DMA 传输。这是通过清除 DMA 控制器中的对应 DMA 使能请求位来实现的。
6. 通过读取 SR 中的 RXCNT，或者在对 POPR 的每次读取操作之后检查 SR 中的 RFDF，确保 RX FIFO 中的所有已接收数据已经传输到存储器接收队列中。
7. 修改 TX 和 RX 通道中用于新队列的 DMA 描述符
8. 通过向 MCR 中的 CLR_TXF 位写入 1 清除 TX FIFO。通过向 MCR 中的 CLR_RXF 位写入‘1’清除 RX FIFO。
9. 通过置位新队列中第一个条目的命令字中的 CTCNT 位，或者通过直接对 TCR 中的 SPI_TCNT 字段进行 CPU 写入，清除传输计数。
10. 通过置位对应的 DMA 置位使能请求位，使能针对分配给模块 TX FIFO 和 RX FIFO 的 DMA 通道的 DMA 使能请求，使能 DMA 通道。
11. 通过清零 EOQF 位，使能对数据的串行传输和串行接收。

43.6.2 切换主机和从机模式

在模块中改变模式时，请执行以下步骤，以确保操作正确无误。

1. 通过置位 MCR[HALT]，停止模块。
2. 通过向 MCR 中的 CLR_TXF 和 CLR_RXF 位写入 1，清除传输和接收 FIFO。
3. 在 MCR[MSTR]中设置合适的模式，并通过清除 MCR[HALT]使能模块。

43.6.3 在主机/从机模式下初始化模块

一旦在 MCR[MSTR]中配置了合适的模式，即可通过清除 MCR[HALT]使能模块。应确保在使能模块主机之前使能模块从机。这样可以确保在主机发起通信之前，从机已经做好通信的准备。

43.6.4 波特率设置

下表列出了基于 CTAR 中的波特率预分频器 PBR 和波特率分频器 BR 组合而产生的波特率。计算所得值假定协议频率为 100 MHz，并且双波特率 DBR 位已清零。

注

在本章中，上面提到的时钟频率只是一个例子。要了解用于驱动器件中此模块的频率，请参见“时钟”一章。

表 43-109. 波特率值(bps)

		波特率分频器预分频器值			
		2	3	5	7
波特率分频器值	2	25.0M	16.7M	10.0M	7.14M
	4	12.5M	8.33M	5.00M	3.57M
	6	8.33M	5.56M	3.33M	2.38M
	8	6.25M	4.17M	2.50M	1.79M
	16	3.12M	2.08M	1.25M	893k
	32	1.56M	1.04M	625k	446k
	64	781k	521k	312k	223k
	128	391k	260k	156k	112k
	256	195k	130k	78.1k	55.8k
	512	97.7k	65.1k	39.1k	27.9k
	1024	48.8k	32.6k	19.5k	14.0k
	2048	24.4k	16.3k	9.77k	6.98k
	4096	12.2k	8.14k	4.88k	3.49k
	8192	6.10k	4.07k	2.44k	1.74k
	16384	3.05k	2.04k	1.22k	872
32768	1.53k	1.02k	610	436	

43.6.5 延迟设置

下表列出了传输后延迟(t_{DT})和 CS 至 SCK 延迟(T_{CSC})的值，这些值可以基于预分频器值和 CTARs 中设定的分频器值生成。计算得到的值假定协议频率为 100 MHz。

注

在本章中，上面提到的时钟频率只是一个例子。要了解用于驱动器件中此模块的频率，请参见“时钟”一章。

表 43-110. 延迟值

		延迟预分频器值			
		1	3	5	7
延迟分频器值	2	20.0 ns	60.0 ns	100.0 ns	140.0 ns
	4	40.0 ns	120.0 ns	200.0 ns	280.0 ns
	8	80.0 ns	240.0 ns	400.0 ns	560.0 ns
	16	160.0 ns	480.0 ns	800.0 ns	1.1 μ s
	32	320.0 ns	960.0 ns	1.6 μ s	2.2 μ s
	64	640.0 ns	1.9 μ s	3.2 μ s	4.5 μ s
	128	1.3 μ s	3.8 μ s	6.4 μ s	9.0 μ s
	256	2.6 μ s	7.7 μ s	12.8 μ s	17.9 μ s
	512	5.1 μ s	15.4 μ s	25.6 μ s	35.8 μ s
	1024	10.2 μ s	30.7 μ s	51.2 μ s	71.7 μ s
	2048	20.5 μ s	61.4 μ s	102.4 μ s	143.4 μ s
	4096	41.0 μ s	122.9 μ s	204.8 μ s	286.7 μ s
	8192	81.9 μ s	245.8 μ s	409.6 μ s	573.4 μ s
	16384	163.8 μ s	491.5 μ s	819.2 μ s	1.1 ms
	32768	327.7 μ s	983.0 μ s	1.6 ms	2.3 ms
65536	655.4 μ s	2.0 ms	3.3 ms	4.6 ms	

43.6.6 FIFO 指针地址的计算

通过 FIFO 寄存器可以完全了解 FIFO 内容，有效条目可以通过针对各 FIFO 的存储器映射指针和计数器予以识别。指向各 FIFO 中先进条目的指针为存储器映射指针。对于 TX FIFO，先进指针为下次发送指针(TXNXTPTR)。对于 RX FIFO，先进指针为下次弹出指针(POPNXTPTR)。下图展示了先进和后进 FIFO 条目及 FIFO 计数器的概念。虽然图中选择的是 TX FIFO，但所示概念具有通用性。有关 FIFO 操作的详情，请参见[发送先进先出\(TX FIFO\)缓冲机制](#)和[接收先进先出\(RX FIFO\)缓冲机制](#)。

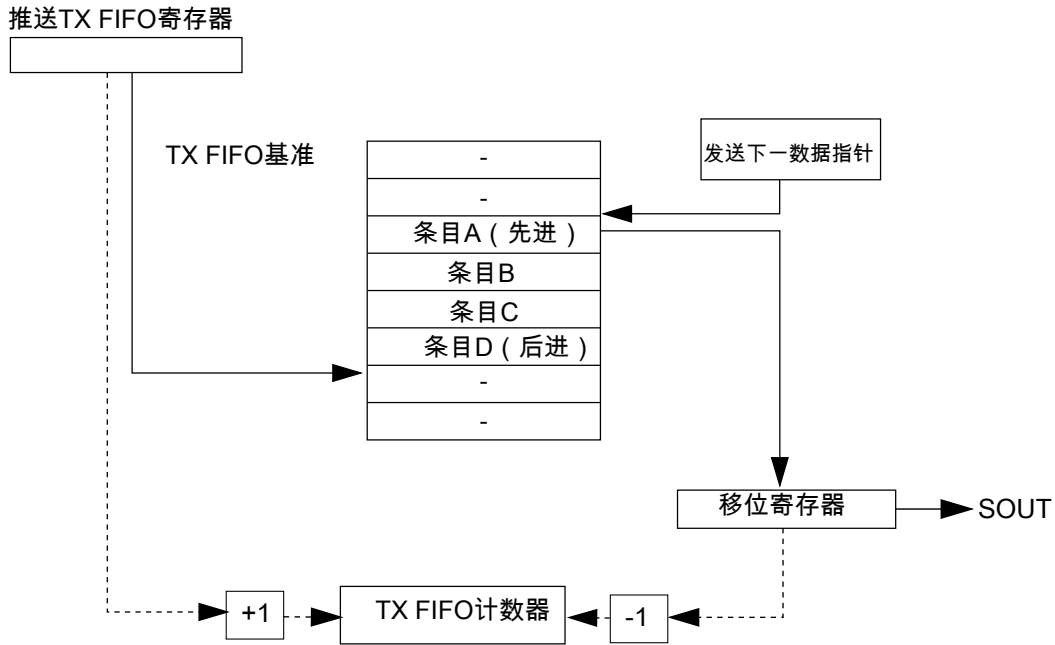


图 43-84. TX FIFO 指针和计数器

43.6.6.1 TX FIFO 中先进条目和后进条目地址的计算

TX FIFO 中先进条目的存储器地址通过下式计算:

$$\text{First-in EntryAddress} = \text{TXFIFOBase} + (4 \times \text{TXNXPTR})$$

TX FIFO 中后进条目的存储器地址通过下式计算:

$$\text{Last-inEntryaddress} = \text{TXFIFOBase} + 4 \times (\text{TXCTR} + \text{TXNXPTR} - 1) \bmod (\text{TXFIFOdepth})$$

TX FIFO 基地址 - TX FIFO 的基地址

TXCTR - TX FIFO 计数器

TXNXPTR - 下次发送指针

TX FIFO 深度 - 发送 FIFO 深度, 跟具体实现有关

43.6.6.2 RX FIFO 中先进条目和后进条目地址的计算

RX FIFO 中先进条目的存储器地址通过下式计算:

$$\text{First-in EntryAddress} = \text{RX FIFOBase} + (4 \times \text{POPXPTR})$$

RX FIFO 中后进条目的存储器地址通过下式计算:

$$\text{Last-inEntryaddress} = \text{RX FIFO Base} + 4 \times (\text{RXCTR} + \text{POPXPTR} - 1) \bmod (\text{RXFIFOdepth})$$

RX FIFO 基地址 - RX FIFO 的基地址

RXCTR - RX FIFO 计数器

POPNXTPTR - 下次弹出指针

RX FIFO 深度 - 接收 FIFO 深度，跟具体实现有关

第 44 章 低功耗 I2C (LPI2C)

44.1 此模块的芯片实现细节

44.1.1 LPI2C 实例化信息

该器件有两个 LPI2C 模块。LPI2C 模块提供低功耗 I2C 模块，该模块可在低功耗停止模式下运行（如需要）。

两个 LPI2C 模块中的每一个都有一个 4 字（32 位）FIFO，用于发送和接收消息。

注

FIFO 仅适用于主机而非从机。

注

对于 LPI2Cx_SCFGR1[TXCFG]：发送标志配置。发送数据标志始终都会在从机-发送传输结束、在检测到 NACK 之前置位。这可能导致一个额外字被写入发送数据 FIFO。

注

对于 LPI2Cx_SCFGR1[RXSTALL]：在 VLPS 模式下，必须使能 RXSTALL，LPI2C 从机才能工作。

44.2 简介

44.2.1 概述

LPI2C 为低功耗集成电路互联 (I2C) 模块，支持作为主机"与"或者"或"从机高效连接至 I2C 总线。只要存在可用的合适时钟，LPI2C 即可在停止模式下持续工作，设计用于使用 DMA 来访问 FIFO 寄存器，大幅降低 CPU 开销。LPI2C 逻辑支持 standard-mode、fast-mode、fast-mode 和 ultra-fast mode。LPI2C 模块还符合系统管理总线 (SMBus) 规范版本 2。

44.2.2 特性

LPI2C 支持 I2C 规范的下列特性：

- 支持 Standard、Fast、Fast+和 Ultra Fast mode
- 在从机模式下支持 HS 模式。
- 在主机模式下支持 HS 模式，但须在 SCL 引脚外部加上上拉电流源。
- 多主机支持，包括同步和仲裁。
- 时钟拉伸。
- 通用调用、7 位和 10 位寻址。
- 软件复位、START 字节和器件 ID 需要软件支持。

LPI2C 主机支持下列特性：

- 命令/发送 4 字 FIFO。
- 接收 4 字 FIFO。
- 命令 FIFO 会在发起传输之前等待 I2C 总线空闲
- 命令 FIFO 可以发起（重复性）START 和 STOP 信号以及一次或多次主机-接收器传输。
- STOP 信号可由命令 FIFO 生成，也可在发送 FIFO 为空时自动生成。
- 主机请求输入可以用于控制 I2C 总线传输的起始时间。
- 灵活接收数据匹配可以在数据匹配时生成中断，并且/或者丢弃无用数据。
- 标志和可选中断用于指示重复性 START 信号、STOP 信号、仲裁丢失、意外 NACK 和命令字错误。
- 支持可配置总线空闲超时和引脚停滞低电平超时。

LPI2C 从机支持下列特性：

- 独立的 I2C 从机寄存器，以最小化主机/从机切换所需的软件开销。
- 支持 7 位或 10 位寻址、地址范围、SMBus 提醒和通用调用地址。
- 发送数据寄存器，支持中断或 DMA 请求。
- 接收数据寄存器，支持中断或 DMA 请求。
- 软件可控 ACK 或 NACK，ACK/NACK 位可选时钟拉伸。

- 可配置时钟拉伸，用于避免发送 FIFO 下溢和接收 FIFO 溢出。
- 数据包结束、STOP 信号或位错误检测标志和可选中断。

44.2.3 框图

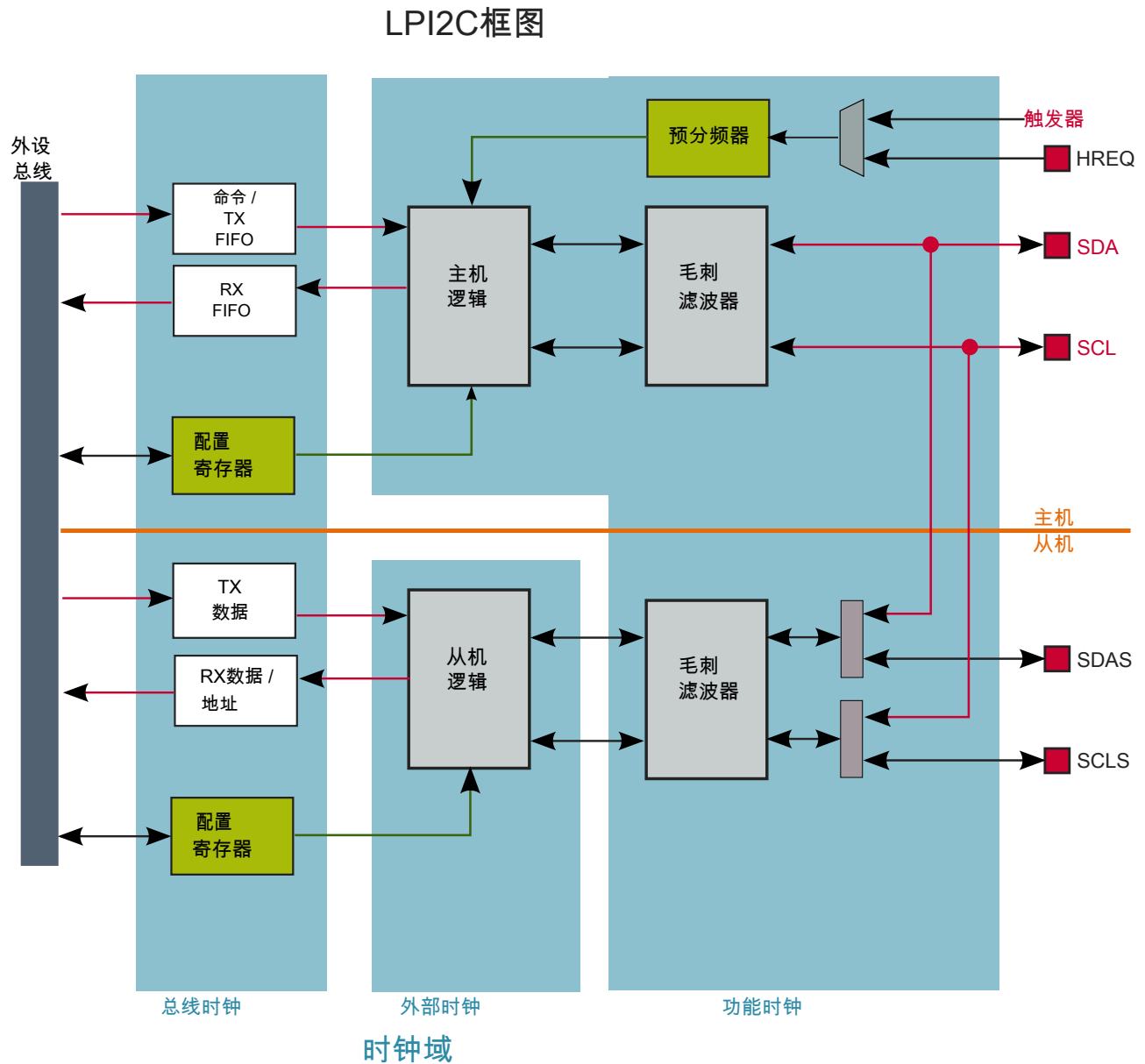


图 44-1. LPI2C 框图

44.2.4 操作模式

LPI2C 模块支持下表描述的芯片模式。

表 44-1. LPI2C 模块支持的芯片模式

芯片模式	LPI2C 操作
运行	正常操作
停止/等待	在休眠使能位(MCR[DOZEN])置位且 LPI2C 采用在停止/等待模式下保持运行的外部或内部时钟源的情况下，可以继续工作。
低漏电停止	休眠使能(MCR[DOZEN])位被忽略，LPI2C 将等待当前传输完成任何挂起操作后，才确认进入低漏电模式。
调试	在调试使能位(MCR[DBG E])置位的条件下，可以继续工作。

44.2.5 信号说明

信号	说明	I/O
SCL	LPI2C 时钟线路。在 4 线模式下，此为 SCL 输入引脚。	I/O
SDA	LPI2C 数据线路。在 4 线模式下，此为 SDA 输入引脚。	I/O
HREQ	主机请求，可在置位且 I2C 总线闲置时启动 LPI2C 主机传输。	I
SCLS	辅助 I2C 时钟线路。在 4 线模式下，此为 SCL 输出引脚。如果将 LPI2C 主机/从机配置为使用单独引脚，则此为 LPI2C 从机 SCL 引脚。	I/O
SDAS	辅助 I2C 数据线路。在 4 线模式下，此为 SDA 输出引脚。如果将 LPI2C 主机/从机配置为使用单独引脚，则此为 LPI2C 从机 SDA 引脚。	I/O

44.3 存储器映射和寄存器

LPI2C 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_6000	版本 ID 寄存器 (LPI2C0_VERID)	32	R	参见章节	44.3.1/1013
4006_6004	参数寄存器 (LPI2C0_PARAM)	32	R	参见章节	44.3.2/1013
4006_6010	主机控制寄存器 (LPI2C0_MCR)	32	R/W	0000_0000h	44.3.3/1014
4006_6014	主机状态寄存器 (LPI2C0_MSR)	32	R/W	0000_0001h	44.3.4/1015
4006_6018	主机中断使能寄存器 (LPI2C0_MIER)	32	R/W	0000_0000h	44.3.5/1017

下一页继续介绍此表...

LPI2C 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_601C	主机 DMA 使能寄存器 (LPI2C0_MDER)	32	R/W	0000_0000h	44.3.6/1019
4006_6020	主机配置寄存器 0 (LPI2C0_MCFGR0)	32	R/W	0000_0000h	44.3.7/1020
4006_6024	主机配置寄存器 1 (LPI2C0_MCFGR1)	32	R/W	0000_0000h	44.3.8/1021
4006_6028	主机配置寄存器 2 (LPI2C0_MCFGR2)	32	R/W	0000_0000h	44.3.9/1023
4006_602C	主机配置寄存器 3 (LPI2C0_MCFGR3)	32	R/W	0000_0000h	44.3.10/ 1023
4006_6040	主机数据匹配寄存器 (LPI2C0_MDMR)	32	R/W	0000_0000h	44.3.11/ 1024
4006_6048	主时钟配置寄存器 0 (LPI2C0_MCCR0)	32	R/W	0000_0000h	44.3.12/ 1024
4006_6050	主时钟配置寄存器 1 (LPI2C0_MCCR1)	32	R/W	0000_0000h	44.3.13/ 1025
4006_6058	主机 FIFO 控制寄存器 (LPI2C0_MFCR)	32	R/W	0000_0000h	44.3.14/ 1026
4006_605C	主机 FIFO 状态寄存器 (LPI2C0_MFSR)	32	R	0000_0000h	44.3.15/ 1027
4006_6060	主机数据发送寄存器 (LPI2C0_MTDR)	32	W	0000_0000h	44.3.16/ 1027
4006_6070	主机数据接收寄存器 (LPI2C0_MRDR)	32	R	0000_4000h	44.3.17/ 1028
4006_6110	从机控制寄存器 (LPI2C0_SCR)	32	R/W	0000_0000h	44.3.18/ 1029
4006_6114	从机状态寄存器 (LPI2C0_SSR)	32	R/W	0000_0000h	44.3.19/ 1031
4006_6118	从机中断使能寄存器 (LPI2C0_SIER)	32	R/W	0000_0000h	44.3.20/ 1033
4006_611C	从机 DMA 使能寄存器 (LPI2C0_SDER)	32	R/W	0000_0000h	44.3.21/ 1035
4006_6124	从机配置寄存器 1 (LPI2C0_SCFGR1)	32	R/W	0000_0000h	44.3.22/ 1036
4006_6128	从机配置寄存器 2 (LPI2C0_SCFGR2)	32	R/W	0000_0000h	44.3.23/ 1038
4006_6140	从机地址匹配寄存器 (LPI2C0_SAMR)	32	R/W	0000_0000h	44.3.24/ 1039
4006_6150	从机地址状态寄存器 (LPI2C0_SASR)	32	R	0000_4000h	44.3.25/ 1040
4006_6154	从机 ACK 发送寄存器 (LPI2C0_STAR)	32	R/W	0000_0000h	44.3.26/ 1040
4006_6160	从机数据发送寄存器 (LPI2C0_STDR)	32	W	0000_0000h	44.3.27/ 1041
4006_6170	从机数据接收寄存器 (LPI2C0_SRDR)	32	R	0000_4000h	44.3.28/ 1042
4006_7000	版本 ID 寄存器 (LPI2C1_VERID)	32	R	参见章节	44.3.1/1013
4006_7004	参数寄存器 (LPI2C1_PARAM)	32	R	参见章节	44.3.2/1013

下一页继续介绍此表...

LPI2C 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_7010	主机控制寄存器 (LPI2C1_MCR)	32	R/W	0000_0000h	44.3.3/1014
4006_7014	主机状态寄存器 (LPI2C1_MSR)	32	R/W	0000_0001h	44.3.4/1015
4006_7018	主机中断使能寄存器 (LPI2C1_MIER)	32	R/W	0000_0000h	44.3.5/1017
4006_701C	主机 DMA 使能寄存器 (LPI2C1_MDER)	32	R/W	0000_0000h	44.3.6/1019
4006_7020	主机配置寄存器 0 (LPI2C1_MCFGR0)	32	R/W	0000_0000h	44.3.7/1020
4006_7024	主机配置寄存器 1 (LPI2C1_MCFGR1)	32	R/W	0000_0000h	44.3.8/1021
4006_7028	主机配置寄存器 2 (LPI2C1_MCFGR2)	32	R/W	0000_0000h	44.3.9/1023
4006_702C	主机配置寄存器 3 (LPI2C1_MCFGR3)	32	R/W	0000_0000h	44.3.10/ 1023
4006_7040	主机数据匹配寄存器 (LPI2C1_MDMR)	32	R/W	0000_0000h	44.3.11/ 1024
4006_7048	主时钟配置寄存器 0 (LPI2C1_MCCR0)	32	R/W	0000_0000h	44.3.12/ 1024
4006_7050	主时钟配置寄存器 1 (LPI2C1_MCCR1)	32	R/W	0000_0000h	44.3.13/ 1025
4006_7058	主机 FIFO 控制寄存器 (LPI2C1_MFCR)	32	R/W	0000_0000h	44.3.14/ 1026
4006_705C	主机 FIFO 状态寄存器 (LPI2C1_MFSR)	32	R	0000_0000h	44.3.15/ 1027
4006_7060	主机数据发送寄存器 (LPI2C1_MTDR)	32	W	0000_0000h	44.3.16/ 1027
4006_7070	主机数据接收寄存器 (LPI2C1_MRDR)	32	R	0000_4000h	44.3.17/ 1028
4006_7110	从机控制寄存器 (LPI2C1_SCR)	32	R/W	0000_0000h	44.3.18/ 1029
4006_7114	从机状态寄存器 (LPI2C1_SSR)	32	R/W	0000_0000h	44.3.19/ 1031
4006_7118	从机中断使能寄存器 (LPI2C1_SIER)	32	R/W	0000_0000h	44.3.20/ 1033
4006_711C	从机 DMA 使能寄存器 (LPI2C1_SDER)	32	R/W	0000_0000h	44.3.21/ 1035
4006_7124	从机配置寄存器 1 (LPI2C1_SCFGR1)	32	R/W	0000_0000h	44.3.22/ 1036
4006_7128	从机配置寄存器 2 (LPI2C1_SCFGR2)	32	R/W	0000_0000h	44.3.23/ 1038
4006_7140	从机地址匹配寄存器 (LPI2C1_SAMR)	32	R/W	0000_0000h	44.3.24/ 1039
4006_7150	从机地址状态寄存器 (LPI2C1_SASR)	32	R	0000_4000h	44.3.25/ 1040
4006_7154	从机 ACK 发送寄存器 (LPI2C1_STAR)	32	R/W	0000_0000h	44.3.26/ 1040
4006_7160	从机数据发送寄存器 (LPI2C1_STDR)	32	W	0000_0000h	44.3.27/ 1041

下一页继续介绍此表...

LPI2C 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_7170	从机数据接收寄存器 (LPI2C1_SRDR)	32	R	0000_4000h	44.3.28/ 1042

44.3.1 版本 ID 寄存器 (LPI2Cx_VERID)

。

地址: 基址 基准 + 0h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								FEATURE															
W	[Shaded]																															
复位	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

LPI2Cx_VERID 字段描述

字段	描述
31–24 MAJOR	主要版本号 本只读字段返回规格的主要版本号。
23–16 MINOR	次要版本号 本只读字段返回规格的次要版本号。
FEATURE	功能规格编号 本只读字段恢复功能集编号。 0x0002 仅主机配有标准功能集。 0x0003 主机和从机配有标准功能集。

44.3.2 参数寄存器 (LPI2Cx_PARAM)

。

地址: 基址 基准 + 4h 偏移

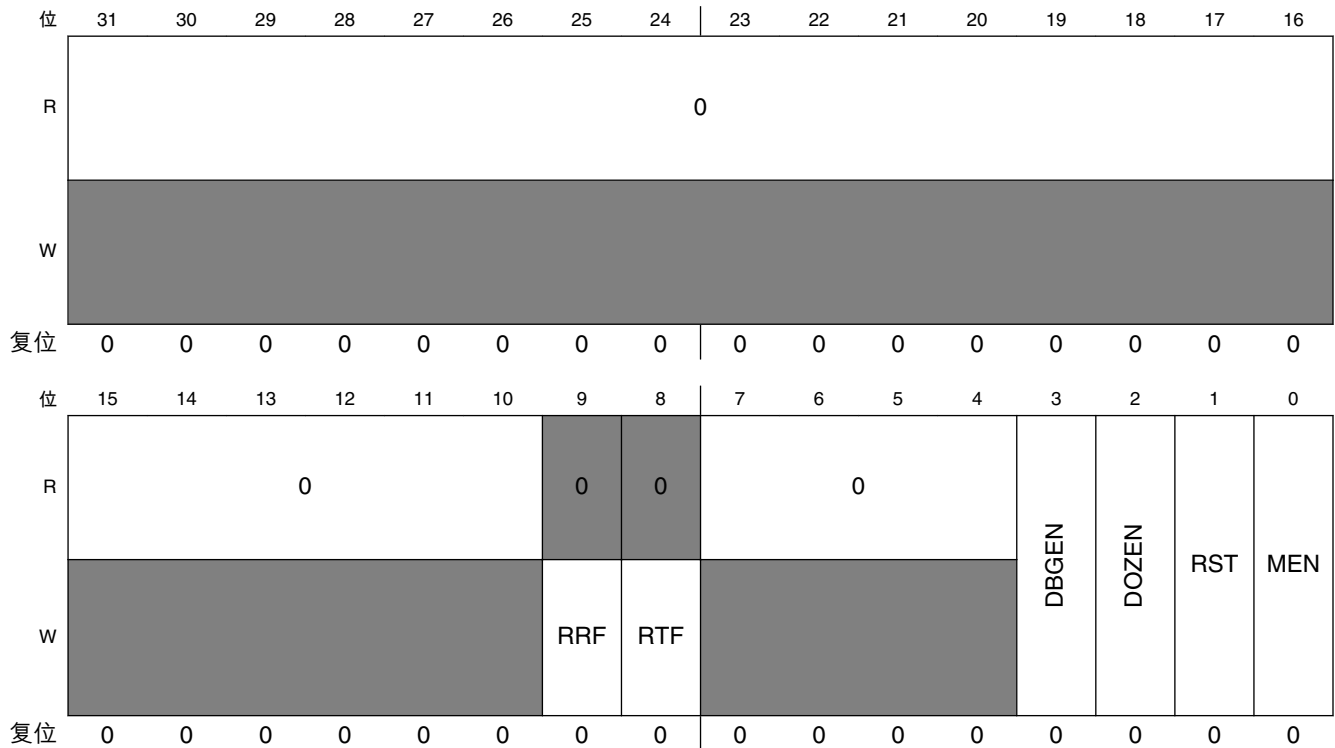
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0		MRXFIFO		0		MTXFIFO																	
W	[Shaded]																															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

LPI2Cx_PARAM 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15-12 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
11-8 MRXFIFO	主机接收 FIFO 大小 接收 FIFO 的主机中字数为 2^{MRXFIFO} 。
7-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
MTXFIFO	主机发送 FIFO 大小 发送 FIFO 的主机中字数为 2^{MTXFIFO} 。

44.3.3 主机控制寄存器 (LPI2Cx_MCR)

地址: 基址 基准 + 10h 偏移



LPI2Cx_MCR 字段描述

字段	描述
31-10 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

LPI2Cx_MCR 字段描述 (继续)

字段	描述
9 RRF	复位接收 FIFO 0 无效。 1 接收 FIFO 复位。
8 RTF	复位发送 FIFO 0 无效。 1 发送 FIFO 复位。
7-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 DBGEN	调试模式使能 0 主机在调试模式中禁用。 1 主机在调试模式中使能。
2 DOZEN	休眠模式使能 为主机使能或禁用休眠模式。 0 主机在休眠模式中使能。 1 主机在休眠模式中禁用。
1 RST	软件复位 复位所有内部主机逻辑和寄存器，主机控制寄存器除外。在由软件清零之前保持置位。 0 主机逻辑未复位。 1 主机逻辑复位。
0 MEN	主机使能 0 主机逻辑禁用。 1 主机逻辑使能。

44.3.4 主机状态寄存器 (LPI2Cx_MSR)

地址: 基址 基准 + 14h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BBF	MBF	0							
W	0															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	DMF	PLTF	FEF	ALF	NDF	SDF	EPF	0						RDF	TDF
W		w1c	w1c	w1c	w1c	w1c	w1c	w1c	0							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

LPI2Cx_MSR 字段描述

字段	描述
31–26 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
25 BBF	总线忙碌标志 0 I2C 总线空闲。 1 I2C 总线忙。
24 MBF	主机忙碌标志 0 I2C 主机空闲。 1 I2C 主机忙碌。
23–15 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
14 DMF	数据匹配标志 表示接收的数据正如 MATCFG 配置一样可以匹配 MATCH0 和/或 MATCH1 字段。CMD 字段丢弃的接收数据不会导致此标志置位。 0 未接收到匹配数据。 1 接收到匹配数据。
13 PLTF	引脚低位超时标志 即使 LPI2C 主机空闲，在 SCL 和/或 SDA 输入电平低，超过 PINLOW 周期时仍会置位。软件负责解决低引脚输入条件。只要低引脚输入超时继续，此标志就无法清零，且在 LPI2C 可以启动 START 信号前必须清零。 0 低引脚输入超时未发生或禁用。 1 低引脚输入超时已发生。
12 FEF	FIFO 错误标志 检测到未生成（重复）START 信号便发送或接收数据的尝试。AUTOSTOP 位置位时发送 FIFO 下溢可能会发生这种情况。此标志置位后，LPI2C 主机将发送 STOP 信号（若忙），且在此标志清零后才会启动新 START 信号。 0 无错误。 1 主机未发送 START 信号的情况下，发送或者接受数据。
11 ALF	仲裁丢失标志 若 LPI2C 主机发送逻辑 1 并在 I2C 总线上检测到逻辑 0，或在发送数据时检测到 START 或 STOP 信号，则此标志置位。此标志置位后，LPI2C 主机将会释放总线（空闲），且在标志清零后才会启动新的 START 信号。 0 主机未丢失仲裁。 1 主机丢失仲裁。
10 NDF	NACK 检测标志 LPI2C 在主机发送数据地址的过程中检测到 NACK 时，此标志将置位。如果在给定的地址上，未检测到预期的 NACK（由命令字配置），若未生成 NACK 则标志置位。置位后，主机将发送 STOP 信号，且在此标志清零后才会启动新 START 信号。 0 未检测到非预期 NACK。 1 检测到非预期 NACK。

下一页继续介绍此表...

LPI2Cx_MSR 字段描述 (继续)

字段	描述
9 SDF	STOP 检测标志 LPI2C 主机生成 STOP 信号时，此标志将置位。 0 主机未生成 STOP 信号。 1 主机生成 STOP 信号。
8 EPF	数据包结束标志 LPI2C 主机生成重复 START 或单个 STOP 信号时，此标志将置位。主机首次生成 START 信号时此标志不置位。 0 主机未生成单个 STOP 或重复 START 信号。 1 主机生成单个 STOP 或重复 START 信号。
7-2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 RDF	接收数据标志 接收 FIFO 中的字数大于 RXWATER 时接收数据标志复位。 0 接收数据未就绪。 1 接收数据就绪。
0 TDF	发送数据标志 发送 FIFO 中的字数等于或小于于 TXWATER 时发送数据标志复位。 0 发送数据未请求。 1 发送数据已请求。

44.3.5 主机中断使能寄存器 (LPI2Cx_MIER)

地址: 基址 基准 + 18h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	[阴影]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0							
W	[阴影]	DMIE	PLTE	FEIE	ALIE	NDIE	SDIE	EPIE	[阴影]	[阴影]	[阴影]	[阴影]	[阴影]	[阴影]	RDIE	TDIE
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI2Cx_MIER 字段描述

字段	描述
31-15 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
14 DMIE	数据匹配中断使能 0 中断禁用。 1 中断使能。
13 PLTIE	低引脚超时中断使能 0 中断禁用。 1 中断使能。
12 FEIE	FIFO 错误中断使能 0 中断禁用。 1 中断使能。
11 ALIE	仲裁丢失中断使能 0 中断禁用。 1 中断使能。
10 NDIE	NACK 检测中断使能 0 中断禁用。 1 中断使能。
9 SDIE	STOP 检测中断使能 0 中断禁用。 1 中断使能。
8 EPIE	数据包结束中断使能 0 中断禁用。 1 中断使能。
7-2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 RDIE	接收数据中断使能 0 中断禁用。 1 中断使能。
0 TDIE	发送数据中断使能 0 中断禁用。 1 中断使能。

44.3.6 主机 DMA 使能寄存器 (LPI2Cx_MDER)

地址: 基址 基准 + 1Ch 偏移

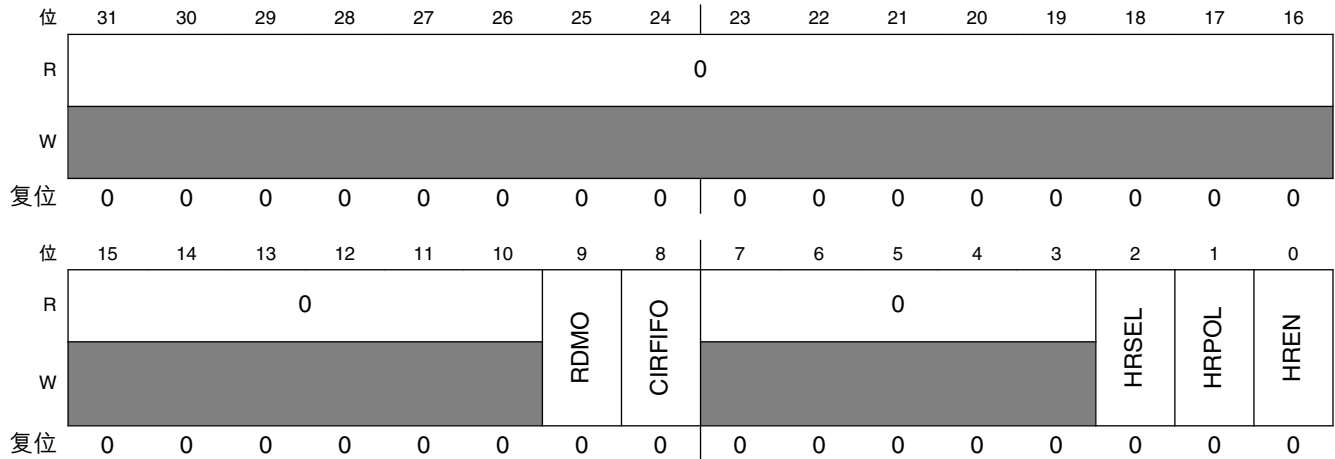
位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R	0																
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R	0														RDDE	TDDE	
W																	
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0

LPI2Cx_MDER 字段描述

字段	描述
31–2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 RDDE	接收数据 DMA 使能 0 DMA 请求禁用。 1 DMA 请求使能。
0 TDDE	发送数据 DMA 使能 0 DMA 请求禁用。 1 DMA 请求使能

44.3.7 主机配置寄存器 0 (LPI2Cx_MCFGR0)

地址: 基址 基准 + 20h 偏移



LPI2Cx_MCFGR0 字段描述

字段	描述
31-10 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
9 RDMO	<p>仅接收数据匹配</p> <p>使能后，将丢弃所有不能引起 DMF 置位的接收数据。DMF 置位后，RDMO 配置被忽略。禁用 RDMO 时，在 DMF 清零前清零 RDMO，以确保无接收数据丢失。</p> <p>0 接收的数据按正常情况存储在接收 FIFO 中。 1 接收的数据被丢弃，除非 RMF 置位。</p>
8 CIRFIFO	<p>循环 FIFO 使能</p> <p>使能后，发送 FIFO 读取指针保存至临时寄存器。发送 FIFO 将被清空，但是一旦 LPI2C 主机空闲且发送 FIFO 为空，读取指针值将从临时寄存器恢复。这将导致发送 FIFO 的内容重复循环。若 AUTOSTOP 置位，将在发送 FIFO 为空且读取指针恢复时发送 STOP 信号。</p> <p>0 循环 FIFO 禁用。 1 循环 FIFO 使能。</p>
7-3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 HRSEL	<p>主机请求选择</p> <p>选择主机请求输入源。</p> <p>0 主机请求输入为引脚 LPI2C_HREQ。 1 主机请求输入为输入触发。</p>
1 HRPOL	<p>主机请求极性</p> <p>配置主机请求输入引脚极性。</p>

下一页继续介绍此表...

LPI2Cx_MCFGR0 字段描述 (继续)

字段	描述
	0 低电平有效。 1 高电平有效。
0 HREN	主机请求使能 使能后, 若主机请求输入置位且总线空闲, 则 LPI2C 主机将仅启动 START 信号。重复 START 信号不受主机请求影响。 0 主机请求输入禁用。 1 主机请求输入使能。

44.3.8 主机配置寄存器 1 (LPI2Cx_MCFGR1)

MCFGR1 仅在 I2C 主机禁用时写入。

地址: 基址 基准 + 24h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					PINCFG			0					MATCFG		
W	■								■							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					TIMECFG	IGNACK	AUTOSTOP	0					PRESCALE		
W	■								■							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI2Cx_MCFGR1 字段描述

字段	描述
31–27 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
26–24 PINCFG	引脚配置 配置引脚模式。 000 LPI2C 配置用于 2 引脚开漏模式。 001 LPI2C 配置用于 2 引脚仅输出模式 (超快模式)。 010 LPI2C 配置用于 2 引脚推挽模式。 011 LPI2C 配置用于 4 引脚推挽模式。 100 LPI2C 配置用于 2 引脚开漏模式, 配备单独的 LPI2C 从机。 101 LPI2C 配置用于 2 引脚仅输出模式 (超快模式), 配备单独的 LPI2C 从机。

下一页继续介绍此表...

LPI2Cx_MCFGR1 字段描述 (继续)

字段	描述
	110 LPI2C 配置用于 2 引脚推挽模式，配备单独的 LPI2C 从机。 111 LPI2C 配置用于 4 引脚推挽模式 (反相输出)。
23-19 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
18-16 MATCFG	匹配配置 配置会引起 DMF 置位的条件。 000 匹配已禁用。 001 保留。 010 匹配使能 (第一个数据等于 MATCH0 或 MATCH1)。 011 匹配使能 (任意数据等于 MATCH0 或 MATCH1)。 100 匹配使能 (第一个数据等于 MATCH0 且第二个数据等于 MATCH1)。 101 匹配使能 (任意数据等于 MATCH0 且下一个数据等于 MATCH1)。 110 匹配使能 (第一个数据和 MATCH1 等于 MATCH0 和 MATCH1)。 111 匹配使能 (任意数据和 MATCH1 等于 MATCH0 和 MATCH1)。
15-11 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
10 TIMECFG	超时配置 0 若 SCL 处于低电平状态的时间长于超时配置，则引脚低电平超时标志置位。 1 若 SCL 或 SDA 处于低电平状态的时间长于超时配置，则引脚低电平超时标志置位。
9 IGNACK	置位后，接收的 NACK 字段被忽略，并假定为 ACK。此位需要在超快模式下置位。 0 LPI2C 主机将正常接收 ACK 和 NACK。 1 LPI2C 主机将对 NACK 采用与 ACK 相同的处理方式。
8 AUTOSTOP	自动 STOP 生成 使能后，LPI2C 主机忙碌且发送 FIFO 为空时生成 STOP 信号。还可以使用发送 FIFO 命令来生成 STOP 信号。 0 无效。 1 发送 FIFO 为空且 LPI2C 主机忙碌时自动生成 STOP 信号。
7-3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
PRESCALE	预分频器 配置用于所有 LPI2C 主机逻辑的时钟预分频器，数字毛刺滤波器除外。 000 1 分频。 001 2 分频。 010 4 分频。 011 8 分频。 100 16 分频。 101 32 分频。 110 64 分频。 111 128 分频。

44.3.9 主机配置寄存器 2 (LPI2Cx_MCFGR2)

MCFGR2 仅在 I2C 主机禁用时写入。

地址: 基址 基准 + 28h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FILTSDA				0				FILTSCS				0				BUSIDLE											
W	0				0				0				0				0				0											
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI2Cx_MCFGR2 字段描述

字段	描述
31–28 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
27–24 FILTSDA	SDA 信号毛刺滤波器 针对 SDA 输入配置 I2C 主机数字毛刺滤波器，配置为 0 即可禁用毛刺滤波器。与 FILTSDA 周期相等或较短的毛刺将被过滤和忽略。毛刺滤波器的延迟等于 FILTSDA 周期，且必须配置为小于最小 SCL 低电平或高电平周期。 毛刺滤波器计数不受预分频配置影响，且在高速模式中自动旁路。
23–20 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
19–16 FILTSCS	SCL 信号毛刺滤波器 针对 SCL 输入配置 I2C 主机数字毛刺滤波器，配置为 0 即可禁用毛刺滤波器。与 FILTSCS 周期相等或较短的毛刺将被过滤和忽略。毛刺滤波器的延迟等于 FILTSCS 周期，且必须配置为小于最小 SCL 低电平或高电平周期。 毛刺滤波器计数不受预分频配置影响，且在高速模式中自动旁路。
15–12 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
BUSIDLE	总线空闲超时 在时钟周期内配置总线空闲超时周期。若 SCL 和 SDA 高电平的时间长于 BUSIDLE 周期，则 I2C 总线假定为空闲且主机可生成 START 信号。置位为 0 时，该功能禁用。

44.3.10 主机配置寄存器 3 (LPI2Cx_MCFGR3)

MCFGR3 仅在 I2C 主机禁用时写入。

地址: 基址 基准 + 2Ch 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												PINLOW												0							
W	0												0												0							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

LPI2Cx_MCFGR3 字段描述

字段	描述
31-20 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
19-8 PINLOW	引脚低电平超时 以时钟周期为单位的引脚低电平超时。若 SCL 和/或 SDA 低电平持续时间长于(PINLOW * 256)周期，则 PLTF 置位。置位为 0 时，该功能禁用。
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

44.3.11 主机数据匹配寄存器 (LPI2Cx_MDMR)

地址: 基址 基准 + 40h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								MATCH1								0								MATCH0							
W	0								0								0								0							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI2Cx_MDMR 字段描述

字段	描述
31-24 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
23-16 MATCH1	匹配 1 值 接收数据匹配使能后与接收到的数据进行比较。
15-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
MATCH0	匹配 0 值 接收数据匹配使能后与接收到的数据进行比较。

44.3.12 主时钟配置寄存器 0 (LPI2Cx_MCCR0)

该 MCCR0 寄存器在 I2C 主机使能的情况下无法更改。该寄存器用于 Standard、Fast、Fast+ 和 Ultra Fast 模式。

地址: 基址 基准 + 48h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
R	0	DATAVD								0	SETHOLD								0	CLKHI								0	CLKLO							
W	0	0								0	0								0	0								0	0							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

LPI2Cx_MCCR0 字段描述

字段	描述
31–30 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
29–24 DATAVD	数据有效延迟 用作 SDA 数据保持时间的最小周期数 (减 1)。必须配置为小于最小 SCL 低电平周期。
23–22 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
21–16 SETHOLD	设置保持延迟 主机用来设置并保持单个 (重复) START 及单个 STOP 信号的最小周期 (减 1)。设置时间随检测到外部 SCL 引脚上升沿所需的时间有所延长。忽略外部加载导致的额外板级延迟, 相当于 $(2 + \text{FILTSCCL}) / 2^{\text{PRESCALE}}$ 周期。
15–14 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
13–8 CLKHI	时钟高电平周期 SCL 时钟高电平受主机驱动的最小周期数 (减 1)。高电平时间随检测到外部 SCL 引脚上升沿所需的时间有所延长。忽略外部加载导致的额外板级延迟, 相当于 $(2 + \text{FILTSCCL}) / 2^{\text{PRESCALE}}$ 周期。
7–6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLKLO	时钟低电平周期 SCL 时钟低电平受主机驱动的最小周期数 (减 1)。此数值还是 STOP 和 START 信号之间的最短总线空闲时间。

44.3.13 主时钟配置寄存器 1 (LPI2Cx_MCCR1)

I2C 主机使能并用于高速模式传输时, 无法更改 MCCR1。高度模式的单独时钟配置允许在快速模式下发生仲裁 (由 MCCR0 定时配置), 或在切换至高速模式前发生仲裁 (由 MCCR1 定时配置)。

地址: 基址 基准 + 50h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0								0								0							
W																																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI2Cx_MCCR1 字段描述

字段	描述
31–30 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
29–24 DATAVD	数据有效延迟 用作 SDA 数据保持时间的最小周期数 (减 1)。必须配置为小于最小 SCL 低电平周期。

下一页继续介绍此表...

LPI2Cx_MCCR1 字段描述 (继续)

字段	描述
23-22 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
21-16 SETHOLD	设置保持延迟 主机用来设置并保持单个 (重复) START 及单个 STOP 信号的最小周期 (减 1)。设置时间随检测到外部 SCL 引脚上升沿所需的时间有所延长。忽略外部加载导致的额外板级延迟, 相当于 $(2 + \text{FILTSCCL}) / 2^{\text{PRESCALE}}$ 周期。
15-14 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
13-8 CLKHI	时钟高电平周期 SCL 时钟高电平受主机驱动的最小周期数 (减 1)。高电平时间随检测到外部 SCL 引脚上升沿所需的时间有所延长。忽略外部加载导致的额外板级延迟, 相当于 $(2 + \text{FILTSCCL}) / 2^{\text{PRESCALE}}$ 周期。
7-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLKLO	时钟低电平周期 SCL 时钟低电平受主机驱动的最小周期数 (减 1)。此数值还是 STOP 和 START 信号之间的最短总线空闲时间。

44.3.14 主机 FIFO 控制寄存器 (LPI2Cx_MFCR)

地址: 基址 基准 + 58h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RXWATER								0								TXWATER							
W	0								0								0								0							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI2Cx_MFCR 字段描述

字段	描述
31-24 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
23-16 RXWATER	接收 FIFO 水印 接收 FIFO 中的字数大于 RXWATER 时接收数据标志复位。写入的值等于或大于 FIFO 大小将会被截断。
15-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
TXWATER	发送 FIFO 水印 发送 FIFO 中的字数等于或小于 TXWATER 时发送数据标志复位。写入的值等于或大于 FIFO 大小将会被截断。

44.3.15 主机 FIFO 状态寄存器 (LPI2Cx_MFSR)

地址: 基址 基准 + 5Ch 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RXCOUNT								0				TXCOUNT											
W	0																															
复位	0																															

LPI2Cx_MFSR 字段描述

字段	描述
31–24 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
23–16 RXCOUNT	接收 FIFO 计数 返回接收 FIFO 中的字数。
15–8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
TXCOUNT	发送 FIFO 计数 返回发送 FIFO 中的字数。

44.3.16 主机数据发送寄存器 (LPI2Cx_MTD R)

CMD 字段的 8 位写入将数据存储存储在命令 FIFO，但不会递增 FIFO 写入指针。若 CMD 字段自上次 FIFO 写入后未单独写入，则 DATA 字段 8 位写入将零扩展 CMD 字段，同时递增 FIFO 写入指针。16 位或 32 位将写入 CMD 和 DATA 字段，同时递增 FIFO。

地址: 基址 基准 + 60h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0								0							
W	Reserved																CMD								DATA							
复位	0																															

LPI2Cx_MTD R 字段描述

字段	描述
31–11 Reserved	此字段为保留字段。
10–8 CMD	命令数据 000 发送 DATA[7:0]。 001 接收(DATA[7:0] + 1)字节。

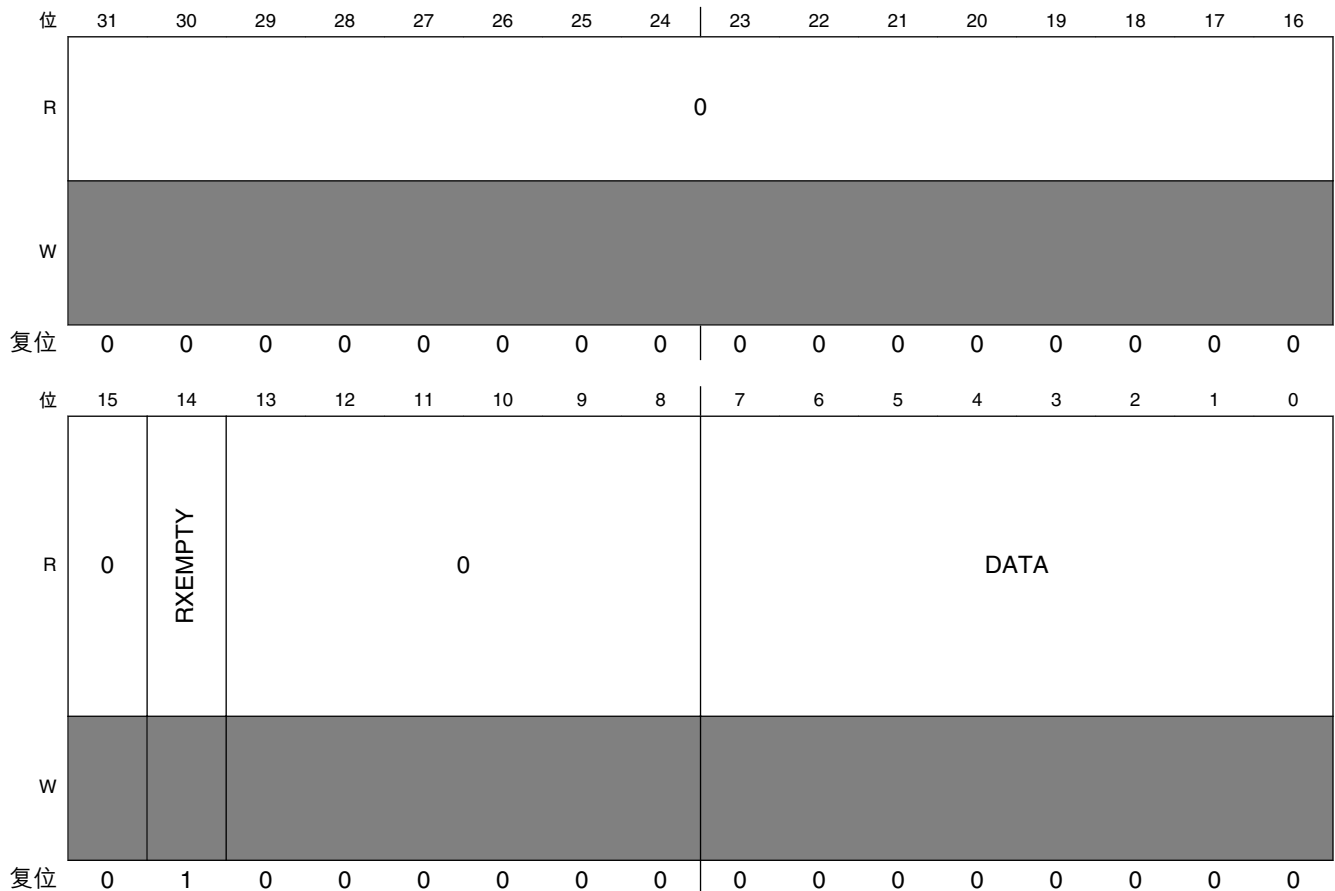
下一页继续介绍此表...

LPI2Cx_MTDR 字段描述 (继续)

字段	描述
	010 生成 STOP 信号。 011 接收并丢弃(DATA[7:0] + 1)字节。 100 生成 (重复的) START 信号并发送 DATA[7:0]中的地址。 101 生成 (重复的) START 信号并发送 DATA[7:0]中的地址。此传输预计返回 NACK。 110 利用高速模式生成 (重复的) START 信号并发送 DATA[7:0]中的地址。 111 利用高速模式生成 (重复的) START 信号并发送 DATA[7:0]中的地址。此传输预计返回 NACK。
DATA	发送数据 在 DATA 字段写入 8 位将零扩展 CMD 字段。

44.3.17 主机数据接收寄存器 (LPI2Cx_MRDR)

地址: 基址 基准 + 70h 偏移



LPI2Cx_MRDR 字段描述

字段	描述
31-15 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

LPI2Cx_MRDR 字段描述 (继续)

字段	描述
14 RXEMPTY	RX 为空 0 接收 FIFO 非空。 1 接收 FIFO 为空。
13–8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
DATA	接收数据 读取此寄存器, 返回 I2C 主机接收的未丢弃的数据。CMD 字段可以丢弃接收数据, 或配置主机丢弃不匹配数据。

44.3.18 从机控制寄存器 (LPI2Cx_SCR)

地址: 基址 基准 + 110h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R	0																
W	[保留]																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0						0	0	0	FILTDC		FILTEN		0		RST	SEN
W	[保留]						RRF	RTF	[保留]		[保留]		[保留]		RST	SEN	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

LPI2Cx_SCR 字段描述

字段	描述
31–10 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
9 RRF	复位接收 FIFO 0 无效。 1 数据接收寄存器现在为空。

下一页继续介绍此表...

LPI2Cx_SCR 字段描述 (继续)

字段	描述
8 RTF	复位发送 FIFO 0 无效。 1 数据发送寄存器现在为空。
7-6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 FILTDZ	滤波器休眠使能 0 滤波器在休眠模式中保持使能。 1 滤波器在休眠模式中禁用。
4 FILTEN	滤波器使能 0 在从机模式下禁用数字滤波器和输出延迟计数器。 1 在从机模式下使能数字滤波器和输出延迟计数器。
3-2 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
1 RST	软件复位 0 从机逻辑未复位。 1 从机逻辑复位。
0 SEN	从机使能 0 从机模式禁用。 1 从机模式使能。

44.3.19 从机状态寄存器 (LPI2Cx_SSR)

地址: 基址 基准 + 114h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					BBF	SBF		0							
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SARF	GCF	AM1F	AM0F	FEF	BEF	SDF	RSF	0				TAF	AVF	RDF	TDF
W					w1c	w1c	w1c	w1c								
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI2Cx_SSR 字段描述

字段	描述
31–26 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
25 BBF	总线忙碌标志 0 I2C 总线空闲。 1 I2C 总线忙。
24 SBF	从机忙碌标志 0 I2C 从机空闲。 1 I2C 从机忙碌。
23–16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15 SARF	SMBus 提醒响应标志 此标志通过读取地址状态寄存器清零。此标志无法生成异步唤醒。 0 SMBus 提醒响应禁用或未检测到。 1 SMBus 提醒响应使能并检测到。

下一页继续介绍此表...

LPI2Cx_SSR 字段描述 (继续)

字段	描述
14 GCF	<p>通用调用标志</p> <p>此标志通过读取地址状态寄存器清零。此标志无法生成异步唤醒。</p> <p>0 从机未检测到通用调用地址或通用调用地址禁用。 1 从机检测到通用调用地址。</p>
13 AM1F	<p>地址匹配 1 标志</p> <p>表明接收的地址与在 ADDRCFG 中配置的 ADDR1 字段或 ADDR0 至 ADDR1 范围匹配。此标志通过读取地址状态寄存器清零。此标志无法生成异步唤醒。</p> <p>0 未接收到 ADDR1 或 ADDR0/ADDR1 范围匹配地址。 1 接收到 ADDR1 或 ADDR0/ADDR1 范围匹配地址。</p>
12 AMOF	<p>地址匹配 0 标志</p> <p>表明接收的地址与在 ADDRCFG 中配置的 ADDR0 字段匹配。此标志通过读取地址状态寄存器清零。此标志无法生成异步唤醒。</p> <p>0 未接收到 ADDR0 匹配地址。 1 接收到 ADDR0 匹配地址。</p>
11 FEF	<p>FIFO 错误标志</p> <p>FIFO 错误只能在时钟拉伸禁用时置位。</p> <p>0 未检测到 FIFO 下溢或溢出。 1 检测到 FIFO 下溢或溢出。</p>
10 BEF	<p>位错误标志</p> <p>若 LPI2C 从机发送逻辑 1 并在 I2C 总线上检测到逻辑 0，则此标志置位。从机将忽略剩余发送部分，直至下一个 (重复) START 信号。</p> <p>0 从机未检测到位错误。 1 从机检测到位错误。</p>
9 SDF	<p>STOP 检测标志</p> <p>假定 LPI2C 从机与上一个地址字节匹配，当 LPI2C 从机检测到 STOP 信号时此标志置位。</p> <p>0 从机未检测到 STOP 信号。 1 从机检测到 STOP 信号。</p>
8 RSF	<p>重复 START 标志</p> <p>假定 LPI2C 从机与上一个地址字节匹配，当 LPI2C 从机检测到重复 START 信号时此标志置位。从机首次检测到 START 信号时此标志不置位。</p> <p>0 从机未检测到 START 信号。 1 从机检测到 START 信号。</p>
7-4 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
3 TAF	<p>发送 ACK 标志</p> <p>此标志通过写入发送 ACK 寄存器清零。</p>

下一页继续介绍此表...

LPI2Cx_SSR 字段描述 (继续)

字段	描述
	0 无需发送 ACK/NACK。 1 需要发送 ACK/NACK。
2 AVF	地址有效标志 此标志通过读取地址状态寄存器清零。RXCFG 置位时，此标志通过读取数据接收寄存器清零。 0 地址状态寄存器无效。 1 地址状态寄存器有效。
1 RDF	接收数据标志 此标志通过读取数据接收寄存器清零。RXCFG 置位时，读取数据接收寄存器时此标志不清零，且 AVF 置位。 0 接收数据未就绪。 1 接收数据就绪。
0 TDF	发送数据标志 此标志通过写入数据发送寄存器清零。TXCFG 清零时，若检测到 NACK 或重复 START 或 STOP 信号，则此标志同时清零。 0 发送数据未请求。 1 发送数据已请求。

44.3.20 从机中断使能寄存器 (LPI2Cx_SIER)

地址: 基址 基准 + 118h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SARIE	GCIE	AM1F	AM0IE	FEIE	BEIE	SDIE	RSIE	0				TAIE	AVIE	RDIE	TDIE
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI2Cx_SIER 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

LPI2Cx_SIER 字段描述 (继续)

字段	描述
15 SARIE	SMBus 提醒响应中断使能 0 中断禁用。 1 中断使能。
14 GCIE	通用调用中断使能 0 中断禁用。 1 中断使能。
13 AM1F	地址匹配 1 中断使能 0 中断禁用。 1 中断使能。
12 AMOIE	地址匹配 0 中断使能 0 中断使能。 1 中断禁用。
11 FEIE	FIFO 错误中断使能 0 中断禁用。 1 中断使能。
10 BEIE	位错误中断使能 0 中断禁用。 1 中断使能。
9 SDIE	STOP 检测中断使能 0 中断禁用。 1 中断使能。
8 RSIE	重复 START 中断使能 0 中断禁用。 1 中断使能。
7-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
3 TAIE	发送 ACK 中断使能 0 中断禁用。 1 中断使能。
2 AVIE	地址有效中断使能 0 中断禁用。 1 中断使能。
1 RDIE	接收数据中断使能 0 中断禁用。 1 中断使能。
0 TDIE	发送数据中断使能

下一页继续介绍此表...

LPI2Cx_SIER 字段描述 (继续)

字段	描述
0	中断禁用。
1	中断使能

44.3.21 从机 DMA 使能寄存器 (LPI2Cx_SDER)

地址: 基址 基准 + 11Ch 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	[Reserved]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													AVDE	RDDE	TDDE
W	[Reserved]													AVDE	RDDE	TDDE
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

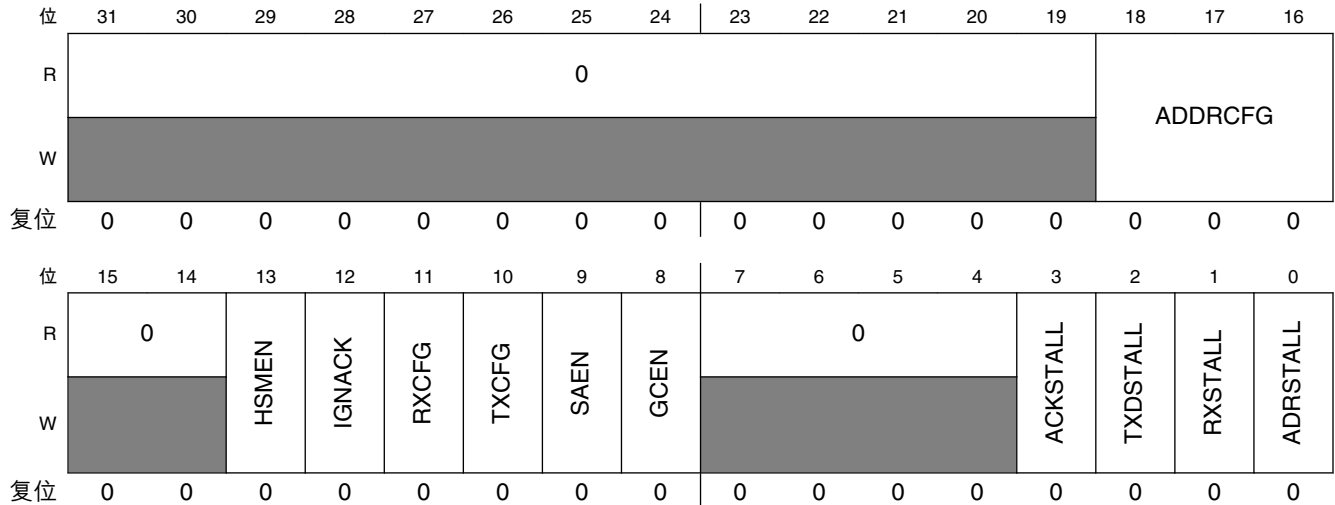
LPI2Cx_SDER 字段描述

字段	描述
31-3 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
2 AVDE	地址有效 DMA 使能 地址有效 DMA 请求与接收数据 DMA 请求共享。若两种请求均使能，则 RXCFG 置位允许 DMA 读取数据接收寄存器的地址。 0 DMA 请求禁用。 1 DMA 请求使能。
1 RDDE	接收数据 DMA 使能 0 DMA 请求禁用。 1 DMA 请求使能。
0 TDDE	发送数据 DMA 使能 0 DMA 请求禁用。 1 DMA 请求使能。

44.3.22 从机配置寄存器 1 (LPI2Cx_SCFGR1)

SCFGR1 仅在 I2C 从机禁用时写入。

地址: 基址 基准 + 124h 偏移



LPI2Cx_SCFGR1 字段描述

字段	描述
31-19 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
18-16 ADDRCFG	地址配置 配置会引起地址匹配的条件。 000 地址匹配 0 (7 位)。 001 地址匹配 0 (10 位)。 010 地址匹配 0 (7 位) 或地址匹配 1 (7 位)。 011 地址匹配 0 (10 位) 或地址匹配 1 (10 位)。 100 地址匹配 0 (7 位) 或地址匹配 1 (10 位)。 101 地址匹配 0 (10 位) 或地址匹配 1 (7 位)。 110 从地址匹配 0 (7 位) 至地址匹配 1 (7 位)。 111 从地址匹配 0 (10 位) 至地址匹配 1 (10 位)。
15-14 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
13 HSMEN	高速模式使能 使能从机地址 0000_1XX 的高速模式主机代码检测，但不会导致此代码的地址匹配。置位且检测到任意高速模式主机代码时，FILTEN 和 ACKSTALL 位被忽略，直至检测到下一个 STOP 信号。 0 禁用高速模式主机代码检测。 1 使能高速模式主机代码检测。
12 IGNACK	忽略 NACK 置位时，LPI2C 从机将在检测到 NACK 后继续传输。此位需要在超快模式下置位。

下一页继续介绍此表...

LPI2Cx_SCFG1 字段描述 (继续)

字段	描述
	<p>0 从机将在检测到 NACK 后结束传输。</p> <p>1 从机在检测到 NACK 后不会结束传输。</p>
11 RXCFG	<p>接收数据配置</p> <p>0 读取数据接收寄存器将返回接收数据并清零接收数据标志。</p> <p>1 地址有效标志置位时读取数据接收寄存器将返回地址状态寄存器并清零地址有效标志。地址有效标志清零时读取数据接收寄存器将返回接收数据并清零接收数据标志。</p>
10 TXCFG	<p>发送标志配置</p> <p>发送数据标志始终都会在从机-发送传输结束、在检测到 NACK 之前置位。这可能导致一个额外字被写入发送数据 FIFO。</p> <p>TXCFG=0 时,数据发送寄存器在检测到从机-发送传输时自动为空。这会导致发送数据标志在检测到从机-发送传输时置位,并在从机-发送传输结束时取消。</p> <p>TXCFG=1 时,发送数据标志在数据发送寄存器为空时置位,并在数据发送寄存器为满时取消。允许数据发送寄存器在检测到从机-发送传输之前完成填充,但可能导致数据发送寄存器在从机-发送传输的上一个字节上检测到 NACK 之前就写入。</p> <p>0 数据发送寄存器为空时,发送数据标志在从机-发送传输期间仅置位。</p> <p>1 数据发送寄存器为空时,发送数据标志置位。</p>
9 SAEN	<p>SMBus 提醒使能</p> <p>0 禁用 SMBus 提醒的匹配。</p> <p>1 使能 SMBus 提醒的匹配。</p>
8 GCEN	<p>通用调用使能</p> <p>0 通用调用地址禁用。</p> <p>1 通用调用地址使能。</p>
7-4 保留	<p>此字段为保留字段。</p> <p>此只读字段为保留字段且值始终为 0。</p>
3 ACKSTALL	<p>ACK SCL 终止</p> <p>在从机-发送地址字节、从机-接收地址和数据字节器件使能 SCL 时钟拉伸,允许软件在 ACK 或 NACK 发送之前写入发送 ACK 寄存器。时钟拉伸发生在发送第 9 位时,因此不兼容高速模式。</p> <p>ACKSTALL 使能时,不需要置位 RXSTALL 或 ADRSTALL</p> <p>0 时钟拉伸禁用。</p> <p>1 时钟拉伸使能。</p>
2 TXDSTALL	<p>TX 数据 SCL 终止</p> <p>从机-发送传输期间发送数据标志置位时,使能 SCL 时钟拉伸。时钟拉伸发生在第 9 位之后,因此兼容高速模式。</p> <p>0 时钟拉伸禁用。</p> <p>1 时钟拉伸使能。</p>
1 RXSTALL	<p>RX SCL 终止</p> <p>从机-接收传输期间接收数据标志置位时,使能 SCL 时钟拉伸。时钟拉伸发生在第 9 位之后,因此兼容高速模式。</p>

下一页继续介绍此表...

LPI2Cx_SCFGR1 字段描述 (继续)

字段	描述
	0 时钟拉伸禁用。 1 时钟拉伸使能。
0 ADRSTALL	地址 SCL 终止 地址有效标志置位时使能 SCL 时钟拉伸。时钟拉伸仅发生在第 9 位之后，因此兼容高速模式。 0 时钟拉伸禁用。 1 时钟拉伸使能。

44.3.23 从机配置寄存器 2 (LPI2Cx_SCFGR2)

SCFGR2 仅在 I2C 从机禁用时写入。

地址: 基址 基准 + 128h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FILTSDA				0				FILTSCS				0				DATAVD				0				CLKHOLD			
W	0				0				0				0				0				0				0							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

LPI2Cx_SCFGR2 字段描述

字段	描述
31-28 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
27-24 FILTSDA	SDA 毛刺滤波器 为 SDA 输入配置 I2C 从机数字毛刺滤波器，配置 0 将禁用毛刺滤波器。与 FILTSDA 周期相等或较短的毛刺将被过滤和忽略。毛刺滤波器的延迟等于 FILTSDA+3 周期，且必须配置为小于最小 SCL 低电平或高电平周期。 毛刺滤波器循环计数不受预分频配置影响，且在高速模式中禁用。
23-20 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
19-16 FILTSCS	SCL 信号毛刺滤波器 针对 SCL 输入配置 I2C 主机数字毛刺滤波器，配置为 0 即可禁用毛刺滤波器。与 FILTSCS 周期相等或较短的毛刺将被过滤和忽略。毛刺滤波器的延迟等于 FILTSCS+3 周期，且必须配置为小于最小 SCL 低电平或高电平周期。 毛刺滤波器循环计数不受预分频配置影响，且在高速模式中禁用。
15-14 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
13-8 DATAVD	数据有效延迟 为 I2C 从机配置的 SDA 数据有效延迟时间等于 FILTSCS+DATAVD+3 周期。此数据有效延迟必须配置为小于最小 SCL 低电平周期。 I2C 从机数据有效延迟时间不受预分频配置影响，且在高速模式中禁用。

下一页继续介绍此表...

LPI2Cx_SCFGR2 字段描述 (继续)

字段	描述
7-4 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
CLKHOLD	时钟保持时间 时钟拉伸使能时,为 I2C 从机配置最短时钟保持时间。最短保持时间等于 CLKHOLD+3 周期。I2C 从机时钟保持时间不受预分频配置影响,且在高速模式中禁用。

44.3.24 从机地址匹配寄存器 (LPI2Cx_SAMR)

地址: 基址 基准 + 140h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					ADDR1										0
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					ADDR0										0
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI2Cx_SAMR 字段描述

字段	描述
31-27 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
26-17 ADDR1	地址 1 值 与接收的地址相比较以检测从机地址。在 10 位模式下,第一个地址字节与{ 11110, ADDR1[10:9] }相比较,第二个地址字节与 ADDR1[8:1]相比较。在 7 位模式下,地址与 ADDR1[7:1]相比较。
16-11 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
10-1 ADDR0	地址 0 值 与接收的地址相比较以检测从机地址。在 10 位模式下,第一个地址字节与{ 11110, ADDR0[10:9] }相比较,第二个地址字节与 ADDR0[8:1]相比较。在 7 位模式下,地址与 ADDR0[7:1]相比较。
0 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

44.3.25 从机地址状态寄存器 (LPI2Cx_SASR)

地址: 基址 基准 + 150h 偏移

位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R	0																
W	[Shaded]																
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R	0	ANV	0			RADDR											
W	[Shaded]																
复位	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0	0

LPI2Cx_SASR 字段描述

字段	描述
31-15 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
14 ANV	地址无效 0 RADDR 有效。 1 RADDR 无效。
13-11 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
RADDR	接收地址 RADDR 在 AMF 置位以及读取此寄存器将 AMF 清零时更新。在 7 位模式下，地址字节存储在 RADDR[7:0] 中。在 10 位模式下，第一个地址字节为 { 11110, RADDR[10:9], RADDR[0] }，第二个地址字节 RADDR[8:1]。因此，R/W 位始终存储在 RADDR[0] 中。

44.3.26 从机 ACK 发送寄存器 (LPI2Cx_STAR)

地址: 基址 基准 + 154h 偏移

位	31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
R	0																
W	[Shaded]																
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
R	0																
W	[Shaded]																
复位	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
																	TXNACK

LPI2Cx_STAR 字段描述

字段	描述
31-1 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
0 TXNACK	发送 NACK NACKSTALL 置位后，该位必须在每个地址匹配，每次接收到字节时写入一次。也可在 LPI2C 从机禁用或空闲时写入，以配置默认 ACK/NACK。 0 针对接收的字发送 ACK。 1 针对接收的字发送 NACK。

44.3.27 从机数据发送寄存器 (LPI2Cx_STDR)

地址: 基址 基准 + 160h 偏移

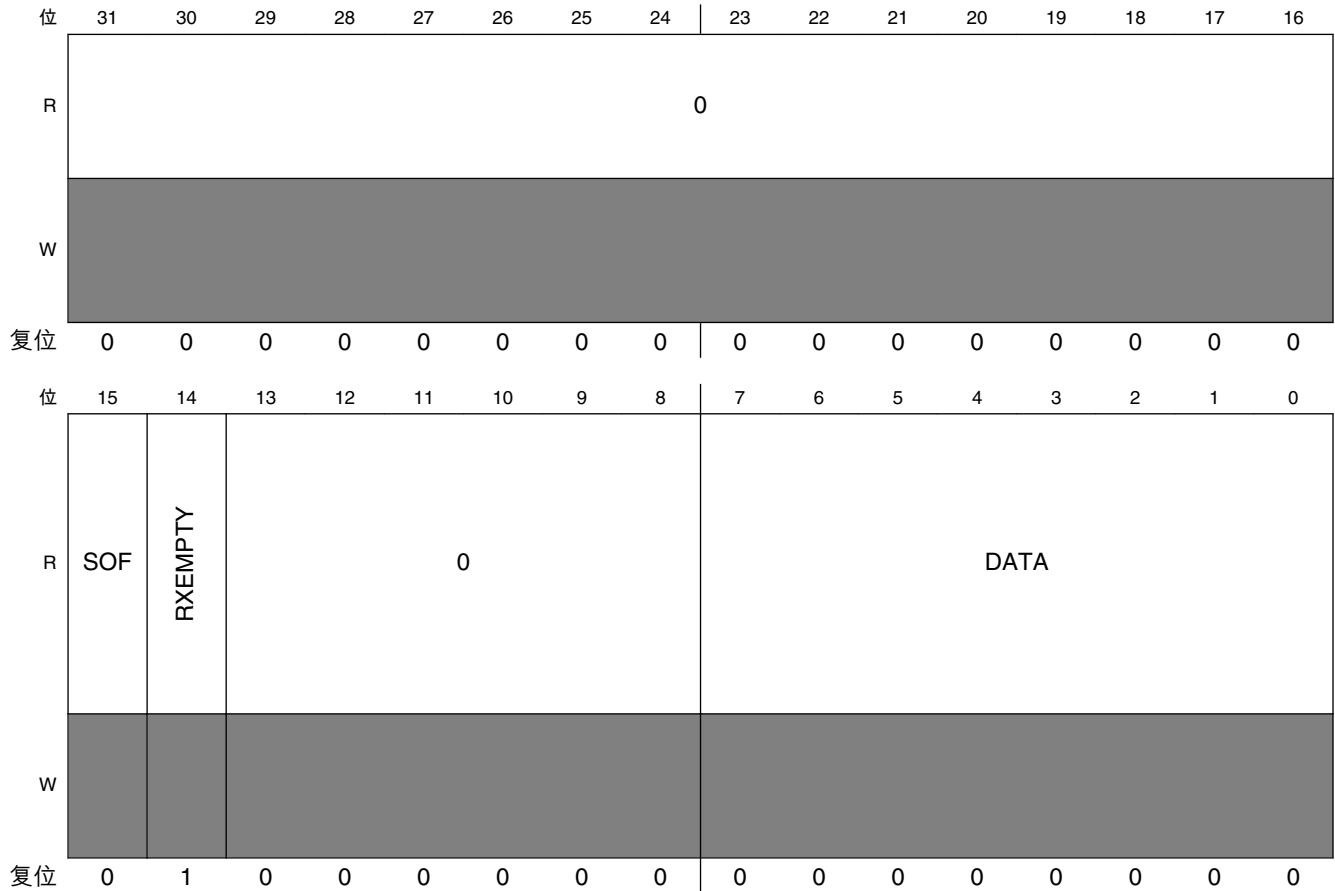
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	Reserved																DATA															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPI2Cx_STDR 字段描述

字段	描述
31-8 Reserved	此字段为保留字段。
DATA	发送数据 写入此寄存器将 I2C 从机-发送数据存储于发送寄存器中。

44.3.28 从机数据接收寄存器 (LPI2Cx_SRDR)

地址: 基址 基准 + 170h 偏移



LPI2Cx_SRDR 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15 SOF	帧开始 0 表明这不是单个 (重复) START 或 STOP 信号下第一个数据字。 1 表明这是单个 (重复) START 或 STOP 信号下第一个数据字。
14 RXEMPTY	RX 为空 0 数据接收寄存器不为空。 1 数据接收寄存器为空。
13-8 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
DATA	接收数据 读取此寄存器，返回 I2C 从机接收的数据。

44.4 功能说明

44.4.1 时钟和复位

44.4.1.1 功能时钟

LPI2C 功能时钟是异步于总线时钟的，并且可以在低功耗模式下保持使能，以支持 LPI2C 主机进行 I2C 总线传输。LPI2C 从机也用此功能时钟来支持数字滤波器和数据保持时间配置。LPI2C 主机通过一个预分频器对功能时钟分频，得到的频率必须比 I2C 总线带宽至少快 8 倍。

44.4.1.2 外部时钟

LPI2C 从机逻辑直接由外部引脚 LPI2C_SCL 和 LPI2C_SDA（如果主机和从机通过独立的引脚实现，则为 LPI2C_SCLS 和 LPI2C_SDAS）直接计时。这样，LPI2C 从机就可以保持正常工作，即使 LPI2C 功能时钟被禁用也是如此。注意，如果 LPI2C 功能时钟被禁用，则必须禁用 LPI2C 从机数字滤波器，这可能会影响 I2C 规范中部分时序参数的合规要求，比如数据保持时间等参数。

44.4.1.3 总线时钟

总线时钟仅用于通过总线访问控制和配置寄存器。总线时钟频率必须足以支持 LPI2C 主机和从机寄存器的数据带宽要求。

44.4.1.4 芯片复位

当芯片复位时，LPI2C 主机和从机的逻辑和寄存器将复位至其默认状态。

44.4.1.5 软件复位

LPI2C 主机在其控制寄存器中设有一个软件复位位。MCR[RST]将把所有主机逻辑和寄存器复位至其默认状态，MCR 本身除外。

LPI2C 从机在其控制寄存器中设有一个软件复位位。SCR[RST]将把所有从机逻辑和寄存器复位至其默认状态，SCR 本身除外。

44.4.1.6 FIFO 复位

LPI2C 主机设有只写控制位，用于复位发送 FIFO (MCR[RTF])和接收 FIFO (MCR[RRF])。FIFO 复位后为空。

LPI2C 从机设有只写控制位，用于复位发送数据寄存器(SCR[RTF])和接收数据寄存器(SCR[RRF])。数据寄存器复位后为空。

44.4.2 主机模式

LPI2C 主机逻辑独立于从机逻辑，用于在 I2C 总线上执行所有主机模式传输。

44.4.2.1 发送和命令 FIFO

发送 FIFO 存储命令数据，用于启动各种 I2C 操作。通过发送 FIFO 中的命令可以启动下列操作：

- 带地址字节和预期 ACK 或 NACK 的 START 或重复 START 信号
- 发送数据（默认为写入发送 FIFO 的零扩展字节）。
- 接收 1-256 数据字节（也可配置为丢弃接收数据并不存储在接收 FIFO 中）。
- STOP 信号（也可配置为在发送 FIFO 为空时发送 STOP 信号）。

在 START 和 STOP 信号之间可插入多个发送和接收命令，发送和接收命令不得交错，以符合 I2C 规格。接收数据命令和接收数据并丢弃的命令可以交错，以确保仅接收到的数据存储在接受 FIFO 中（或与数据匹配逻辑比较）。

LPI2C 主机在（重复）START 信号下支持 10 位地址，第二个地址字节后接发送字节，主机发送数据后接多个数据字节。

期望 NACK（如 hs 模式主机代码）的 START 或重复 START 信号必须后接 STOP 或（重复）START 信号。

44.4.2.2 主机工作

LPI2C 使能后会监控 I2C 总线以检测 I2C 总线什么时候空闲 (MSR[BBF])。I2C 总线在 SCL 或 SDA 为低电平时不空闲，在检测到 STOP 信号或总线空闲超时时变为空闲（如 MCFGR2[BUSIDLE]配置）。I2C 总线空闲时，发送 FIFO 不为空，且主机请求置位或禁用，随后 LPI2C 主机将在 I2C 总线上启动传输。该过程步骤如下：

- 等待总线空闲时间等于(MCCR0[CLKLO] + 1)乘以预分频器个周期。

- 使用 MCCR0 中的时序配置来发送 START 信号和地址字节，若配置了高速模式传输，则使用 MCCR1 的时序配置。
- 根据发送 FIFO 配置来执行主机-发送或主机-接收传输。
- 根据发送 FIFO"与"或者"或"MCFGR1[AUTOSTOP]配置来发送重复 START 或 STOP 信号。使用时序配置寄存器可以更改重复 START 信号。

LPI2C 主机禁用（因 MCR[MEN]清零或因模式进入而自动禁用），LPI2C 将持续清空发送 FIFO 直至 STOP 信号发送完成。但是，它将不会为发送或接收 FIFO 停止 I2C 总线，且一旦发送 FIFO 为空，将自动生成 STOP 信号。

LPI2C 主机在特定条件下可以停止 I2C 总线，这将会导致 SCL 在字节第一位持续被拉至低电平直至以下条件解除：

- LPI2C 主机使能且忙碌，发送 FIFO 为空，且 MCFGR1[AUTOSTOP]清零。
- LPI2C 主机使能并接收数据，接收数据（因命令或接受数据匹配）不丢弃，且接收 FIFO 为满。

44.4.2.3 接收 FIFO 和数据匹配

接收 FIFO 用于在主机-接收机传输之间存储接收数据。接收数据也可配置为丢弃接收数据，而不是存储在接收 FIFO 中，具体取决于发送 FIFO 中的命令字配置。

接收数据支持接收数据匹配功能，可以将接收到的数据与两个字节之一或带屏蔽位的数据字节进行匹配。数据匹配功能还可配置为仅比较最后（重复）START 信号后第一个或第二个接收到的数据字。已根据命令字丢弃的接收数据无法引起数据匹配位，且会延误第一个接收数据的匹配，直至接收到丢弃的数据。接收机匹配功能也可使用 MCFGR0[RDMO]控制位来配置为丢弃所有接收数据，直至检测到数据匹配。MCFGR0[RDMO]控制位清零后便可进行数据匹配，在 MSR[DMF]清零前需先清除 MCFGR0[RDMO]以接收所有的后续数据。

44.4.2.4 时序参数

下列时序参数可由 LPI2C 主机配置。高速模式（MCCR1）和其他模式（MCCR0）参数被分开配置。这样一来，便可使用常用时序参数发送高速模式主机代码，然后切换至高速模式时序（后接重复 START）直至下一个 STOP 信号。

LPI2C 功能时钟周期中的 LPI2C 主机时序参数配置如下。必须配置为符合所需模式的 I2C 时序规格。

- 总线空闲时间始终为(MCCR0[CLKLO] + 1) 乘以预分频器。随检测外部 SDA 上升沿所需的时间而有所扩展。
- START 或重复 START 保持时间等于(MCCR0/1[SETHOLD] + 1)乘以预分频器。

- START、重复 START 或 STOP 设置时间等于 $(MCCR0/1[SETHOLD] + 1)$ 乘以预分频器。随检测外部 SCL 上升沿所需的时间而有所扩展。
- SCL 低电平时间（时钟拉伸前）等于 $(MCCR0/1[CLKLO] + 1)$ 乘以预分频器。
- SCL 高电平时间等于 $(MCCR0/1[CLKHI] + 1)$ 乘以预分频器。随检测外部 SCL 上升沿所需的时间而有所扩展。
- SDA 输出延迟等于 $(MCCR0/1[DATAVD] + 1)$ 乘以预分频器。

检测外部上升沿所需的时间取决于一些参数（包括总线加载和外部上拉电阻大小）。最短延迟等于 2 加上引脚输入数字滤波器设置（为 SCL 和 SDA 单独配置），再除以预分频器（引脚输入数字滤波器不受预分频器设置影响）。

必须加强下列时序限制以避免 I2C 总线上或 LPI2C 主机检测到的意外 START 或 STOP 信号。可概括为 SDA，在发送（重复）START 或 STOP 信号以外且 SCL 为高电平的情况下无法更改。

表 44-90. 时序参数

时序参数	最小值	最大值	备注
CLKLO	0x03	-	CLKLO 必须大于 SCL 滤波器的延迟。
CLKHI	0x01	-	
SETHOLD	0x02	-	
DATAVD	0x01	$CLKLO - [(FILTSDA+2) / (2 \wedge PRESCALER)]$	DATAVD 必须小于 CLKLO 减去 SDA 滤波器延迟。
FILTSC	0x00	$[CLKLO \times (2 \wedge PRESCALER)] - 3$	
FILTSDA	FILTSC	$[CLKLO \times (2 \wedge PRESCALER)] - 3$	不适用于 SCL 和 SDA 之间板级偏移的补偿。
BUSIDLE	$(CLKLO+SETHOLD+2) \times 2$	-	必须大于 CLKHI+1。

时序参数配置必须符合 I2C 规格要求，具体取决于支持的模式和 LPI2C 功能时钟频率。部分配置示例如下。

表 44-91. LPI2C 示例时序配置

I2C 模式	时钟频率	波特率	预分频器	FILTSC/ FILTSDA	SETHOLD	CLKLO	CLKHI	DATAVD
Fast	8 MHz	400 kbps	0x0	0x0/0x0	0x04	0x0B	0x05	0x02
Fast+	8 MHz	1 Mbps	0x0	0x0/0x0	0x02	0x03	0x01	0x01
Fast	48 MHz	400 kbps	0x2	0x1/0x1	0x07	0x11	0x0B	0x03
Fast+	48 MHz	1 Mbps	0x2	0x1/0x1	0x03	0x06	0x04	0x04
Fast+	48 MHz	1 Mbps	0x0	0x1/0x1	0x1D	0x18	0x13	0x0F
HS-mode	48 MHz	3.2 Mbps	0x0	0x0/0x0	0x07	0x08	0x03	0x01
Fast	60 MHz	400 kbps	0x1	0x2/0x2	0x11	0x28	0x21	0x08
Fast+	60 MHz	1 Mbps	0x1	0x2/0x2	0x07	0x0F	0x0B	0x01

下一页继续介绍此表...

表 44-91. LPI2C 示例时序配置 (继续)

I2C 模式	时钟频率	波特率	预分频器	FILTSCL/ FILTSDA	SETHOLD	CLKLO	CLKHI	DATAVD
HS-mode	60 MHz	3.33 Mbps	0x1	0x0/0x0	0x04	0x03	0x04	0x01
Ultrafast	60 MHz	5 Mbps	0x0	0x0/0x0	0x02	0x05	0x03	0x01

根据位计算周期数的公式如下:

$$\text{波特率分频器} = ((\text{CLKLO} + \text{CLKHI} + 2) * 2^{\text{PRESCALER}}) + \text{ROUNDDOWN}((2 + \text{FILTSCL}) / 2^{\text{PRESCALER}})$$

假定 SCL 在 LPI2C 功能时钟信号在一个周期内上拉, 具体取决于上拉电阻和 SCL 引脚上的负载。

44.4.2.5 错误条件

LPI2C 主机激活时将监控错误, 在下列条件下生成错误标志, 并阻止发送新的 START 信号, 直至软件清除错误。

- LPI2C 主机检测到并非由自身发出的 START 或 STOP 信号 (置位 MSR[ALF])。
- SDA 上的数据与发送的数据不一致 (置位 MSR[ALF])。
- 假定 MCFGR1[IGNACK]清零, 传输数据时检测到 NACK (置位 MSR[NDF])。
- 假定 MCFGR1[IGNACK]清零, 检测到 NACK 且预期 ACK 的地址字节 (置位 MSR[NDF])。
- 假定 MCFGR1[IGNACK]清零, 检测到 ACK 且预期 NACK 的地址字节 (置位 MSR[NDF])。
- 发送 FIFO 要求发送或接收无 START 信号的数据 (置位 MSR[FEF])。
- SCL (MCFGR1[TIMECFG]置位时为 SDA) 信号处于低电平状态长达 (MCFGR2[TIMELOW] * 256) 预分频器周期, 且没有引脚转换 (置位 MSR[PLTF])。

软件必须响应 MSR[PTLF]标志以终止现有命令, 采用干净 (清零 MCR[MEN]) 或突然 (置位 MCR[SWRST]) 的方式。

SCL 和 SDA 在(BUSIDLE+1)预分频周期中保持高电平时, MCFGR2[BUSIDLE]字段可用来强制 I2C 总线为空闲。LPI2C 主机首先使能时, I2C 总线通常被认为空闲。但在 BUSIDLE 配置大于 0 时, 在 I2C 总线被首先认为空闲前, SCL"与"或者"或"SDA 必须高于(BUSIDLE+1) 预分频器周期。

44.4.2.6 引脚配置

LPI2C 主机默认 LPI2C_SDA 和 LPI2C_SCL 引脚采用开漏配置。有效开漏支持取决于具体的器件，且需要引脚复用以支持有效开漏。高速模式支持也取决于具体的器件，且需要 LPI2C_SCL 引脚来支持 I2C 规格要求的当前源上拉。

LPI2C 主机还支持仅使用 LPI2C_SDA 和 LPI2C_SCL 引脚的 I2C 超快模式所需的推挽功能输出。超快模式支持还需要置位 IGNACK 位。

推挽 2 线配置还可用于 LPI2C 主机，以支持部分高速模式（假定 LPI2C 是唯一主机，且总线上的所有 I2C 引脚采用相同电压）。将第 9 个时钟脉冲以外的每个时钟的 LPI2C_SCL 引脚配置为推挽，允许高速模式兼容的从机以执行时钟拉伸。在这种模式下，LPI2C_SDA 引脚三态为主机-接收数据位和主机-发送 ACK/NACK 位。

推挽 4 线配置将 SCL 输入和输出以及 SDA 输入和输出分为不同的引脚，使用 SCL/SDA 作为输入引脚，SCLS/SDAS 作为输出引脚，并带有可配置的极性。这在连接 I2C 总线至外部电平位移器时能够简化外部连接。使用此配置时，LPI2C 主机逻辑和 LPI2C 从机逻辑无法连接至单独的 I2C 总线。

44.4.3 从机模式

LPI2C 从机逻辑独立于主机逻辑，用于在 I2C 总线上执行所有从机模式传输。

44.4.3.1 地址匹配

LPI2C 从机可以配置为匹配两个地址之一，为每个地址使用 7 位或 10 位寻址模式，也可配置为以 7 位或 10 位寻址模式匹配地址范围。另外，也可以配置为匹配通用调用地址或 SMBus 报警地址并生成相应的标志。LPI2C 从机也可配置为检测高速模式主机代码，在检测到下一 STOP 信号之前，禁用数字滤波器并输出有效的延迟时间。

一旦匹配到有效的地址，LPI2C 从机会自动执行从机-发送或从机-接收传输，直到检测到 NACK（除非 IGNACK 置位）、位误差（LPI2C 从机驱动 SDA，但采样值不同）或（重复性）START 或 STOP 信号为止。

44.4.3.2 发送与接收

发送与接收数据寄存器为双缓冲器配置，它们分别只会从从机-发送和从机-接收传输过程中更新。接收到的从机地址可以配置为从接收数据寄存器（例如，在使用 DMA 传输数据时）或从地址状态寄存器读取。发送数据寄存器可以配置为仅在检测到从机-发送传输时请求数据，或者配置为每当发送数据寄存器为空时请求新数据。

发送数据寄存器只能在发送数据标志置位时写入。接收数据寄存器只能在已接收数据标志置位时（或者地址有效标志置位且 RXCFG=1）读取。地址状态寄存器只能在地址有效标志置位时读取。

44.4.3.3 时钟拉伸

LPI2C 从机支持多种可配置选项，用于指定时钟拉伸的时机。可以配置以下条件以执行时钟拉伸。

- 在地址字节第 9 个时钟脉冲中，地址有效标志置位。
- 在从机-发送传输第 9 个时钟脉冲中，发送数据标志置位。
- 在从机-接收传输第 9 个时钟脉冲中，接收数据标志置位。
- 在地址字节或从机-接收传输第 8 个时钟脉冲中，发送 ACK 标志置位。在高速模式下，该功能禁用。
- 也可以将时钟拉伸延长 CLKHOLD 个周期，增加建立时间，以对 SDA 引脚进行外部采样。在高速模式下，该功能禁用。

除非由 CLKHOLD 配置延长，否则，在时钟拉伸功能使能时，时钟拉伸将在 SDA 更新之后延长一个外设总线时钟周期。

44.4.3.4 时序参数

LPI2C 从机可以配置下列时序参数，当 SCR[FILTEN] 清零时，当 SCR[FILTDZ] 在休眠模式下置位时，以及当 LPI2C 从机检测到高速模式时，这些参数将被禁用。禁用时，LPI2C 从机直接从 I2C 总线时钟采样，可能不能达到 I2C 规格中列出的所有时序要求（比如，标准/快速模式下的 SDA 最小保持时间）。

- 从 SCL 取反到 SDA 更新的 SDA 数据有效时间。
- 时钟拉伸使能时的 SCL 保持时间，用于在对 SDA 进行外部采样时增加建立时间。
- SCL 毛刺滤波器时间。
- SDA 毛刺滤波器时间。

LPI2C 从机对时序要求提出了下列限制。

- FILTSDA 必须配置为大于或等于 FILTSCL (除非对 SDA 与 SCL 之间的电路板电平偏移进行补偿)。
- DATAVD 必须配置为小于最小 SCL 低电平周期。

44.4.3.5 错误条件

LPI2C 从机可以检测下列错误条件。

- 当 LPI2C 从机驱动 SDA 时，采样的值和发送的值不一致，会导致位错误标志置位。
- FIFO 错误标志将因发送数据下溢或接收数据溢出而置位。可以使能时钟拉伸以消除发生下溢或溢出的可能性。
- 当 RXCFG 置位时，FIFO 错误标志也会因地址溢出而置位，否则地址溢出将无标志。可以使能时钟拉伸以消除发生溢出的可能性。

LPI2C 从机未实现因 SCL"与"或者"或"SDA 陷入低电平状态而产生的超时。如果需要进行这种检测，则应使用 LPI2C 主机逻辑；在检测到这种条件时，软件可以使 LPI2C 从机复位。

44.4.4 中断和 DMA 请求

根据具体器件，LPI2C 主机和从机中断可以组合起来。

根据具体器件，LPI2C 主机和从机发送 DMA 请求可以组合起来。

根据具体器件，LPI2C 主机和从机接收 DMA 请求可以组合起来。

44.4.4.1 主机模式

下表列出了可以生成 LPI2C 主机中断和 LPI2C 主机发送/接收 DMA 请求的主机模式资源。

表 44-92. 主机中断和 DMA 请求

标志	说明	中断	DMA 请求	低功耗唤醒
TDF	数据可以写入发送 FIFO，具体由 TXWATER 配置。	是	TX	是
RDF	数据可以从接收 FIFO 读取，具体由 RXWATER 配置。	是	RX	是

下一页继续介绍此表...

表 44-92. 主机中断和 DMA 请求 (继续)

标志	说明	中断	DMA 请求	低功耗唤醒
EPF	主机已经发送重复性 START 或 STOP 信号。	是	N	是
SDF	主机已经发送 STOP 信号。	是	N	Y
NDF	主机在地址字节中检测到 NACK, 预期为 ACK 主机在地址字节中检测到 ACK, 预期为 NACK 或者主机在主机-发送器数据字节中检测到 NACK。	是	N	是
ALF	主机因在错误时间检测到 START/STOP 信号而丢失了仲裁, 或者主机正在发送数据但收到的数据与发送的数据不同。	是	N	是
FEF	主机在命令 FIFO 中的预期信号为 START, FIFO 中的下一个条目不是 START 信号。	是	N	是
PLTF	引脚低电平超时已使能, SCL (或 SDA, 如果已配置) 处于低电平状态的时间长于超时配置。	是	N	是
DMF	接收的数据与配置的匹配数据相匹配, 并且命令 FIFO 条目未丢弃接收数据。	是	N	是
MBF	LPI2C 主机正忙于发送/接收数据。	否	否	否
BBF	LPI2C 主机已使能, 在 I2C 总线上检测到活动, 但未检测到 STOP 信号, 且未发生总线闲置超时 (若使能)。	N	否	否

44.4.4.2 从机模式

下表列出了可以生成 LPI2C 从机中断和 LPI2C 从机发送/接收 DMA 请求的从机模式资源。

表 44-93. 从机中断和 DMA 请求

标志	说明	中断	DMA 请求	低功耗唤醒
TDF	数据可以写入发送数据寄存器。	是	TX	是

下一页继续介绍此表...

表 44-93. 从机中断和 DMA 请求 (继续)

标志	说明	中断	DMA 请求	低功耗唤醒
RDF	数据可以从接收数据寄存器读取。	是	RX	是
AVF	地址可以从地址状态寄存器读取。	是	RX	是
TAF	ACK/NACK 可以写入发送 ACK 寄存器。	是	N	是
RSF	在一个重复 START 信号后, 从机检测到地址匹配。	是	N	是
SDF	在一个 STOP 信号后, 从机检测到地址匹配。	是	N	是
BEF	从机正在发送数据, 但接收的数据与发送的数据不同。	是	N	是
FEF	发送数据下溢, 接收数据溢出或地址状态溢出(当 RXCFG=1 时)。此标志只能在时钟拉伸禁用时置位。	是	N	是
AM0F	从机在 ADDR0 字段中检测到地址匹配。	是	N	否
AM1F	从机在 ADDR1 字段或地址范围中检测到地址匹配。	是	N	否
GCF	从机在通用调用地址中检测到地址匹配。	是	N	否
SARF	从机在 SMBus 报警地址中检测到地址匹配。	是	N	否
SBF	LPI2C 从机正忙于接收地址字节或发送/接收数据。	否	否	否
BBF	LPI2C 从机已使能, 在 I2C 总线上检测到 START 信号, 但未检测到 STOP 信号。	否	否	N

44.4.5 外设触发信号

LPI2C 外设触发信号与其他外设的连接为器件专用。

44.4.5.1 主机输出触发信号

LPI2C 主机生成可以连接至器件上其他外设的输出触发信号。主机在发出一个重复 START 或者 STOP 信号时，发出触发信号，并保持 LPI2C 功能时钟经过预分频器分频后的一个时钟周期。

44.4.5.2 从机输出触发信号

LPI2C 从机生成可以连接至器件上其他外设的输出触发信号。从机在接收到一个重复 START 或者 STOP 信号（丛集地址匹配发生后）输出触发信号。并且保持有效至下一个从机 SCL 信号被取反。

44.4.5.3 输入触发信号

LPI2C 输入触发信号可在 LPI2C_HREQ 引脚中进行选择，以控制 LPI2C 主机总线传输的开始。输入触发信号有效时间必须超过一个 LPI2C 功能时钟周期才能检测到。

第 45 章

通用异步接收器/发送器 (UART)

45.1 此模块的芯片实现细节

45.1.1 UART 配置信息

本章节介绍此器件上每个模块的配置方式。

1. 所有 UART 的标准功能:
 - RS-485 支持
 - 硬件流量控制(RTS/CTS)
 - 9 位 UART, 用以支持带校验位的地址标记
 - 数据的 MSB/LSB 配置
2. UART0 和 UART1 使用内核时钟信号驱动, 而剩余的 UART 使用总线时钟信号驱动。最大波特率为相关源时钟频率的 1/16。
3. 所有 UART 均支持 IrDA
4. UART0 包含标准功能及 ISO7816
5. UART0 可容纳 8 条目数据的发送 FIFO 缓存和接收 FIFO 缓存
6. 所有其他 UART 包含只容纳一个条目数据的发送和接收 FIFO

注

USB DP/DM 引脚可以根据 SIM_MISCCTRL[UARTSELONUSB], 配置为 UART TX/RX。更多详情参见 USBFSOTG 一章中的“[USB 上的 UART 性能](#)”一节。

注

UART1 和 UART2 不支持 ISO7816, 因此相关的寄存器不适用于此器件。

45.1.2 UART 唤醒

可配置 UART，在其收到的第一个有效边沿上生成中断/唤醒。

45.1.3 UART 中断

UART 有多个中断请求源。然而，其中部分源都经 OR 运算相结合，以生成单个中断请求。有关各中断源与中断请求的映射关系，请参阅以下信息：

状态中断组合了下列中断源：

源	UART 0	UART 1	UART 2
发送数据为空	x	x	x
发送完成	x	x	x
空闲线路	x	x	x
接收数据已满	x	x	x
LIN 断点检测	x	x	x
RxD 引脚有效边沿	x	x	x
初始字符检测	x	—	—

错误中断组合了下列中断源：

源	UART 0	UART 1	UART 2
接收器溢出	x	x	x
噪声标志位	x	x	x
帧错误	x	x	x
奇偶校验错误	x	x	x
发送器缓冲器溢出	x	x	x
接收缓冲器溢出	x	x	x
接收器缓冲器下溢	x	x	x
发送阈值(ISO7816)	x	—	—
接收器阈值(ISO7816)	x	—	—
等待定时器(ISO7816)	x	—	—
字符等待定时器(ISO7816)	x	—	—
数据块等待定时器(ISO7816)	x	—	—
保护时间违规(ISO7816)	x	—	—
ATR 持续时间定时器(ISO7816)	x	—	—

45.2 简介

UART 允许与外围设备和 CPU 进行异步串行通信。

45.2.1 特性

UART 包括以下特性：

- 全双工操作
- 标准传号/空号不归零 (NRZ) 格式
- 带可编程脉冲宽度的可选择 IrDA 1.4 归零反相(RZI)格式
- 带/32 小数分频、基于模块时钟频率的 13 位波特率选择
- 可编程 8 位或 9 位数据格式
- 数据帧中的 1 或 2 个可编程停止位
- 单独使能的发送器和接收器
- 发送器输出极性可编程
- 接收输入极性可编程
- 高达 16 位中断字符传输。
- 11 位中断字符检测选项
- 用于发送与接收的独立 FIFO 结构
- 两种接收器唤醒方法：
 - 空闲线路唤醒
 - 地址标志唤醒
- 接收器中的地址匹配功能可减少地址标志唤醒 ISR 开销
- 可以将 MSB 或 LSB 配置为线上传输的首位
- 硬件流控制支持请求发送(RTS)和清除发送(CTS)信号
- 支持与 SIM 卡和智能卡连接的 ISO 7816 协议
 - 支持 T=0 和 T=1 协议

- 通过可编程重试阈值自动重新发送 NACK'd 数据包
- 支持 11 和 12 ETU 传输
- 检测初始数据包并自动传输参数编程
- 中断驱动操作，带 7 个 ISO-7816 特定中断：
 - 等待时间违规
 - 字符等待时间违规
 - 块等待时间违规
 - 检测到初始帧
 - 超过发送错误阈值
 - 超过接收错误阈值
 - 保护时间违规
- 带标志位的中断驱动操作，不限于 ISO-7816 支持
 - 水印中或下面的发送器数据缓冲区
 - 发送完成
 - 水印中或上面的接收器数据缓冲区
 - 空闲接收器输入
 - 接收器数据缓冲区溢出
 - 接收器数据缓冲区下溢
 - 发送数据缓冲区溢出
 - 噪声错误
 - 帧错误
 - 奇偶校验错误
 - 接收引脚上的有效边沿
 - LIN 断点检测
- 接收帧错误检测
- 硬件奇偶生成和校验

- 1/16 位时间噪声检测
- 支持 DMA

45.2.2 工作模式

UART 在所有正常模式下的运行方式相同。

它具有以下低功耗模式：

- 等待模式
- 停止模式

45.2.2.1 运行模式

这是正常工作模式。

45.2.2.2 等待模式

UART 在等待模式下的工作情况取决于 C1[UARTSWAI]字段的状况。

- 如果 C1[UARTSWAI]清零且 CPU 处于等待模式，则 UART 正常运行。
- 如果 C1[UARTSWAI]置位且 CPU 处于等待模式，则 UART 时钟发生器停止工作，UART 进入省电状态。

C1[UARTSWAI]不会为 ISO-7816 智能卡接口发起任何掉电或上电操作。

置位 C1[UARTSWAI]不会影响 C2[RE]或 C2[TE]的状态。

如果 C1[UARTSWAI]置位，则在进入等待模式时正在进行的任何发送或接收都将停止。当内部或外部中断使 CPU 退出等待模式时，发送或接收重新开始。通过复位使 CPU 退出等待模式，将会停止任何正在进行的发送或接收并且将会复位 UART。

45.2.2.3 停止模式

为降低功耗，UART 在停止模式期间处于不活动状态。STOP 指令不会影响 UART 寄存器状态，但 UART 模块时钟将会被禁用。当外部中断使 CPU 退出停止模式后，UART 操作将重新开始运行。通过复位使 CPU 退出停止模式，将会停止任何正在进行的发送或接收并且将会复位 UART。进入或退出停止模式不会为 ISO-7816 智能卡接口发起任何掉电或上电操作。

45.3 UART 信号说明

UART 信号如下表所示。

表 45-1. UART 信号说明

信号	说明	I/O
CTS	清除发送	I
RTS	请求发送	O
RXD	接收数据	I
TXD	发送数据	O

45.3.1 详细信号说明

UART 的详细信号说明如下表所示。

表 45-2. UART—详细信号说明

信号	I/O	说明
CTS	I	清除发送。表示流控制使能时 UART 是否可以开始发送数据。
		状态含义 断言—数据传输可以开始。 否定—数据传输不可以开始。
		定时 断言—当发送设备的 RTS 有效时。 否定—当发送设备的 RTS 无效时。
RTS	O	请求发送。由接收器驱动时，表示 UART 是否已准备好接收数据。由发送器驱动时，可以在传输过程中使能外部收发器。
		状态含义 断言—由接收器驱动时，准备好接收数据。由发送器驱动时，使能外部发送器。 否定—由接收器驱动时，未准备好接收数据。由发送器驱动时，禁用外部发送器。
		定时 断言—可随时发生；可异步断言其他输入信号。 否定—可随时发生；可异步取消断言其他输入信号。
RXD	I	接收数据。接收器的串行数据输入。

下一页继续介绍此表...

表 45-2. UART—详细信号说明 (继续)

信号	I/O	说明	
		状态含义	RXD 是被解析为 1 还是 0 取决于位编码方法以及其他配置设置。
		定时	以模块时钟除以波特率所得频率进行采样。
TXD	O	发送数据。发送器的串行数据输出。	
		状态含义	TXD 是被解析为 1 还是 0 取决于位编码方法以及其他配置设置。
		定时	根据位编码方法以及其他配置设置，在开始时或位时间内驱动。否则，发送与接收时序无关。

45.4 存储器映射和寄存器

本节提供所有存储器和寄存器的详细说明。

访问存储器映射中的保留地址将导致传输错误。此访问不会导致任何实施地址内容修改。

仅支持字节访问。

UART 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_A000	UART 波特率寄存器: 高位 (UART0_BDH)	8	不适用	00h	45.4.1/1066
4006_A001	UART 波特率寄存器: 低位 (UART0_BDL)	8	不适用	04h	45.4.2/1067
4006_A002	UART 控制寄存器 1 (UART0_C1)	8	不适用	00h	45.4.3/1067
4006_A003	UART 控制寄存器 2 (UART0_C2)	8	不适用	00h	45.4.4/1069
4006_A004	UART 状态寄存器 1 (UART0_S1)	8	不适用	C0h	45.4.5/1070
4006_A005	UART 状态寄存器 2 (UART0_S2)	8	不适用	00h	45.4.6/1073
4006_A006	UART 控制寄存器 3 (UART0_C3)	8	不适用	00h	45.4.7/1075
4006_A007	UART 数据寄存器 (UART0_D)	8	不适用	00h	45.4.8/1076
4006_A008	UART 匹配地址寄存器 1 (UART0_MA1)	8	不适用	00h	45.4.9/1077
4006_A009	UART 匹配地址寄存器 2 (UART0_MA2)	8	不适用	00h	45.4.10/1077
4006_A00A	UART 控制寄存器 4 (UART0_C4)	8	不适用	00h	45.4.11/1078
4006_A00B	UART 控制寄存器 5 (UART0_C5)	8	不适用	00h	45.4.12/1078
4006_A00C	UART 扩展数据寄存器 (UART0_ED)	8	不适用	00h	45.4.13/1079
4006_A00D	UART 调制解调器寄存器 (UART0_MODEM)	8	不适用	00h	45.4.14/1080
4006_A00E	UART 红外寄存器 (UART0_IR)	8	不适用	00h	45.4.15/1081

下一页继续介绍此表...

UART 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_A010	UART FIFO 参数 (UART0_PFIFO)	8	不适用	参见章节	45.4.16/1082
4006_A011	UART FIFO 控制寄存器 (UART0_CFIFO)	8	不适用	00h	45.4.17/1083
4006_A012	UART FIFO 状态寄存器 (UART0_SFIFO)	8	不适用	C0h	45.4.18/1084
4006_A013	UART FIFO 发送水印 (UART0_TWFIFO)	8	不适用	00h	45.4.19/1085
4006_A014	UART FIFO 发送计数 (UART0_TCFIFO)	8	不适用	00h	45.4.20/1086
4006_A015	UART FIFO 接收水印 (UART0_RWFIFO)	8	不适用	01h	45.4.21/1086
4006_A016	UART FIFO 接收计数 (UART0_RCFIFO)	8	不适用	00h	45.4.22/1087
4006_A018	UART 7816 控制寄存器 (UART0_C7816)	8	不适用	00h	45.4.23/1087
4006_A019	UART 7816 中断使能寄存器 (UART0_IE7816)	8	不适用	00h	45.4.24/1089
4006_A01A	UART 7816 中断状态寄存器 (UART0_IS7816)	8	不适用	00h	45.4.25/1090
4006_A01B	UART 7816 等待参数寄存器 (UART0_WP7816)	8	不适用	00h	45.4.26/1091
4006_A01C	UART 7816 等待 N 寄存器 (UART0_WN7816)	8	不适用	00h	45.4.27/1092
4006_A01D	UART 7816 等待 FD 寄存器 (UART0_WF7816)	8	不适用	01h	45.4.28/1092
4006_A01E	UART 7816 错误阈值寄存器 (UART0_ET7816)	8	不适用	00h	45.4.29/1093
4006_A01F	UART 7816 发送长度寄存器 (UART0_TL7816)	8	不适用	00h	45.4.30/1093
4006_A03A	UART 7816 ATR 持续定时器寄存器 A (UART0_AP7816A_T0)	8	不适用	00h	45.4.31/1094
4006_A03B	UART 7816 ATR 持续定时器寄存器 B (UART0_AP7816B_T0)	8	不适用	00h	45.4.32/1094
4006_A03C	UART 7816 等待参数寄存器 A (UART0_WP7816A_T0)	8	不适用	00h	45.4.33/1095
4006_A03C	UART 7816 等待参数寄存器 A (UART0_WP7816A_T1)	8	不适用	00h	45.4.34/1095
4006_A03D	UART 7816 等待参数寄存器 B (UART0_WP7816B_T0)	8	不适用	14h	45.4.35/1096
4006_A03D	UART 7816 等待参数寄存器 B (UART0_WP7816B_T1)	8	不适用	14h	45.4.36/1096
4006_A03E	UART 7816 等待和保护参数寄存器 (UART0_WGP7816_T1)	8	不适用	06h	45.4.37/1097

下一页继续介绍此表...

UART 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_A03F	UART 7816 等待参数寄存器 C (UART0_WP7816C_T1)	8	不适用	0Bh	45.4.38/1097
4006_B000	UART 波特率寄存器 : 高位 (UART1_BDH)	8	不适用	00h	45.4.1/1066
4006_B001	UART 波特率寄存器 : 低位 (UART1_BDL)	8	不适用	04h	45.4.2/1067
4006_B002	UART 控制寄存器 1 (UART1_C1)	8	不适用	00h	45.4.3/1067
4006_B003	UART 控制寄存器 2 (UART1_C2)	8	不适用	00h	45.4.4/1069
4006_B004	UART 状态寄存器 1 (UART1_S1)	8	不适用	C0h	45.4.5/1070
4006_B005	UART 状态寄存器 2 (UART1_S2)	8	不适用	00h	45.4.6/1073
4006_B006	UART 控制寄存器 3 (UART1_C3)	8	不适用	00h	45.4.7/1075
4006_B007	UART 数据寄存器 (UART1_D)	8	不适用	00h	45.4.8/1076
4006_B008	UART 匹配地址寄存器 1 (UART1_MA1)	8	不适用	00h	45.4.9/1077
4006_B009	UART 匹配地址寄存器 2 (UART1_MA2)	8	不适用	00h	45.4.10/1077
4006_B00A	UART 控制寄存器 4 (UART1_C4)	8	不适用	00h	45.4.11/1078
4006_B00B	UART 控制寄存器 5 (UART1_C5)	8	不适用	00h	45.4.12/1078
4006_B00C	UART 扩展数据寄存器 (UART1_ED)	8	不适用	00h	45.4.13/1079
4006_B00D	UART 调制解调器寄存器 (UART1_MODEM)	8	不适用	00h	45.4.14/1080
4006_B00E	UART 红外寄存器 (UART1_IR)	8	不适用	00h	45.4.15/1081
4006_B010	UART FIFO 参数 (UART1_PFIFO)	8	不适用	参见章节	45.4.16/1082
4006_B011	UART FIFO 控制寄存器 (UART1_CFIFO)	8	不适用	00h	45.4.17/1083
4006_B012	UART FIFO 状态寄存器 (UART1_SFIFO)	8	不适用	C0h	45.4.18/1084
4006_B013	UART FIFO 发送水印 (UART1_TWFIFO)	8	不适用	00h	45.4.19/1085
4006_B014	UART FIFO 发送计数 (UART1_TCFIFO)	8	不适用	00h	45.4.20/1086
4006_B015	UART FIFO 接收水印 (UART1_RWFIFO)	8	不适用	01h	45.4.21/1086
4006_B016	UART FIFO 接收计数 (UART1_RCFIFO)	8	不适用	00h	45.4.22/1087
4006_B018	UART 7816 控制寄存器 (UART1_C7816)	8	不适用	00h	45.4.23/1087
4006_B019	UART 7816 中断使能寄存器 (UART1_IE7816)	8	不适用	00h	45.4.24/1089
4006_B01A	UART 7816 中断状态寄存器 (UART1_IS7816)	8	不适用	00h	45.4.25/1090

下一页继续介绍此表...

UART 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_B01B	UART 7816 等待参数寄存器 (UART1_WP7816)	8	不适用	00h	45.4.26/1091
4006_B01C	UART 7816 等待 N 寄存器 (UART1_WN7816)	8	不适用	00h	45.4.27/1092
4006_B01D	UART 7816 等待 FD 寄存器 (UART1_WF7816)	8	不适用	01h	45.4.28/1092
4006_B01E	UART 7816 错误阈值寄存器 (UART1_ET7816)	8	不适用	00h	45.4.29/1093
4006_B01F	UART 7816 发送长度寄存器 (UART1_TL7816)	8	不适用	00h	45.4.30/1093
4006_B03A	UART 7816 ATR 持续定时器寄存器 A (UART1_AP7816A_T0)	8	不适用	00h	45.4.31/1094
4006_B03B	UART 7816 ATR 持续定时器寄存器 B (UART1_AP7816B_T0)	8	不适用	00h	45.4.32/1094
4006_B03C	UART 7816 等待参数寄存器 A (UART1_WP7816A_T0)	8	不适用	00h	45.4.33/1095
4006_B03C	UART 7816 等待参数寄存器 A (UART1_WP7816A_T1)	8	不适用	00h	45.4.34/1095
4006_B03D	UART 7816 等待参数寄存器 B (UART1_WP7816B_T0)	8	不适用	14h	45.4.35/1096
4006_B03D	UART 7816 等待参数寄存器 B (UART1_WP7816B_T1)	8	不适用	14h	45.4.36/1096
4006_B03E	UART 7816 等待和保护参数寄存器 (UART1_WGP7816_T1)	8	不适用	06h	45.4.37/1097
4006_B03F	UART 7816 等待参数寄存器 C (UART1_WP7816C_T1)	8	不适用	0Bh	45.4.38/1097
4006_C000	UART 波特率寄存器: 高位 (UART2_BDH)	8	不适用	00h	45.4.1/1066
4006_C001	UART 波特率寄存器: 低位 (UART2_BDL)	8	不适用	04h	45.4.2/1067
4006_C002	UART 控制寄存器 1 (UART2_C1)	8	不适用	00h	45.4.3/1067
4006_C003	UART 控制寄存器 2 (UART2_C2)	8	不适用	00h	45.4.4/1069
4006_C004	UART 状态寄存器 1 (UART2_S1)	8	不适用	C0h	45.4.5/1070
4006_C005	UART 状态寄存器 2 (UART2_S2)	8	不适用	00h	45.4.6/1073
4006_C006	UART 控制寄存器 3 (UART2_C3)	8	不适用	00h	45.4.7/1075
4006_C007	UART 数据寄存器 (UART2_D)	8	不适用	00h	45.4.8/1076
4006_C008	UART 匹配地址寄存器 1 (UART2_MA1)	8	不适用	00h	45.4.9/1077
4006_C009	UART 匹配地址寄存器 2 (UART2_MA2)	8	不适用	00h	45.4.10/1077
4006_C00A	UART 控制寄存器 4 (UART2_C4)	8	不适用	00h	45.4.11/1078
4006_C00B	UART 控制寄存器 5 (UART2_C5)	8	不适用	00h	45.4.12/1078
4006_C00C	UART 扩展数据寄存器 (UART2_ED)	8	不适用	00h	45.4.13/1079

下一页继续介绍此表...

UART 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_C00D	UART 调制解调器寄存器 (UART2_MODEM)	8	不适用	00h	45.4.14/1080
4006_C00E	UART 红外寄存器 (UART2_IR)	8	不适用	00h	45.4.15/1081
4006_C010	UART FIFO 参数 (UART2_PFIFO)	8	不适用	参见章节	45.4.16/1082
4006_C011	UART FIFO 控制寄存器 (UART2_CFIFO)	8	不适用	00h	45.4.17/1083
4006_C012	UART FIFO 状态寄存器 (UART2_SFIFO)	8	不适用	C0h	45.4.18/1084
4006_C013	UART FIFO 发送水印 (UART2_TWFIFO)	8	不适用	00h	45.4.19/1085
4006_C014	UART FIFO 发送计数 (UART2_TCFIFO)	8	不适用	00h	45.4.20/1086
4006_C015	UART FIFO 接收水印 (UART2_RWFIFO)	8	不适用	01h	45.4.21/1086
4006_C016	UART FIFO 接收计数 (UART2_RCFIFO)	8	不适用	00h	45.4.22/1087
4006_C018	UART 7816 控制寄存器 (UART2_C7816)	8	不适用	00h	45.4.23/1087
4006_C019	UART 7816 中断使能寄存器 (UART2_IE7816)	8	不适用	00h	45.4.24/1089
4006_C01A	UART 7816 中断状态寄存器 (UART2_IS7816)	8	不适用	00h	45.4.25/1090
4006_C01B	UART 7816 等待参数寄存器 (UART2_WP7816)	8	不适用	00h	45.4.26/1091
4006_C01C	UART 7816 等待 N 寄存器 (UART2_WN7816)	8	不适用	00h	45.4.27/1092
4006_C01D	UART 7816 等待 FD 寄存器 (UART2_WF7816)	8	不适用	01h	45.4.28/1092
4006_C01E	UART 7816 错误阈值寄存器 (UART2_ET7816)	8	不适用	00h	45.4.29/1093
4006_C01F	UART 7816 发送长度寄存器 (UART2_TL7816)	8	不适用	00h	45.4.30/1093
4006_C03A	UART 7816 ATR 持续定时器寄存器 A (UART2_AP7816A_T0)	8	不适用	00h	45.4.31/1094
4006_C03B	UART 7816 ATR 持续定时器寄存器 B (UART2_AP7816B_T0)	8	不适用	00h	45.4.32/1094
4006_C03C	UART 7816 等待参数寄存器 A (UART2_WP7816A_T0)	8	不适用	00h	45.4.33/1095
4006_C03C	UART 7816 等待参数寄存器 A (UART2_WP7816A_T1)	8	不适用	00h	45.4.34/1095
4006_C03D	UART 7816 等待参数寄存器 B (UART2_WP7816B_T0)	8	不适用	14h	45.4.35/1096

下一页继续介绍此表...

UART 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4006_C03D	UART 7816 等待参数寄存器 B (UART2_WP7816B_T1)	8	不适用	14h	45.4.36/1096
4006_C03E	UART 7816 等待和保护参数寄存器 (UART2_WGP7816_T1)	8	不适用	06h	45.4.37/1097
4006_C03F	UART 7816 等待参数寄存器 C (UART2_WP7816C_T1)	8	不适用	0Bh	45.4.38/1097

45.4.1 UART 波特率寄存器: 高位 (UARTx_BDH)

该寄存器与 BDL 寄存器一起控制 UART 波特率生成的预分频因子。要更新 13 位波特率设置(SBR[12:0])，首先应写入 BDH 以缓存高电平新值的一半，然后写入 BDL。BDH 中的工作值要等到写入 BDL 之后才改变。

BDL 复位为非零值，但在复位后，波特率发生器保持禁用直到首次使能接收器或发送器时为止，也就是，到 C2[RE]或 C2[TE]置位为止。

地址: 基址 基准 + 0h 偏移

位	7	6	5	4	3	2	1	0
读	LBKDIE	RXEDGIE	SBNS			SBR		
写	LBKDIE	RXEDGIE	SBNS			SBR		
复位	0	0	0	0	0	0	0	0

UARTx_BDH 字段描述

字段	描述
7 LBKDIE	LIN 断点检测中断 或 DMA 请求使能 使能 LIN 断点检测标志 LBKDIF，以生成中断请求基于 LBKDDMAS 的状态。或 DMA 传输请求， 0 LBKDIF 中断和 DMA 传输 请求已禁用。 1 LBKDIF 中断或 DMA 传输 请求已使能。
6 RXEDGIE	RX 输入有效边沿中断使能 使能接收输入有效边沿 RXEDGIF，以生成中断请求。 0 来自 RXEDGIF 的硬件中断禁用 (使用轮询)。 1 1RXEDGIF 中断请求已使能。
5 SBNS	停止位数选择 SBNS 选择了数据帧中显示的停止位数目。此字段对可用的 8、9、10 位数据格式有效。此字段在 C7816[ISO7816E]使能时无效。 0 数据帧包含一个停止位。 1 数据帧包含两个停止位。
SBR	UART 波特率位

下一页继续介绍此表...

UARTx_BDH 字段描述 (继续)

字段	描述
	UART 的波特率由 13 SBR 字段决定。详情参见 波特率生成 。 注： <ul style="list-style-type: none"> • 波特率发生器保持禁用，直到 C2[TE]或 C2[RE]在复位之后被首次置位。波特率发生器在 SBR = 0 时被禁用。 • 如果不写入 BDL，只写入 BDH 没有任何作用，因为写入 BDH 只是将数据暂时放置在某个位置，直到写入 SDL。

45.4.2 UART 波特率寄存器: 低位 (UARTx_BDL)

该寄存器与 BDH 寄存器一起控制 UART 波特率生成的预分频因子。要更新 13 位波特率设置 SBR[12:0]，首先应写入 BDH 以缓存高电平新值的一半，然后写入 BDL。BDH 中的工作值要等到写入 BDL 之后才改变。BDL 复位为非零值，但在复位后，波特率发生器保持禁用直到首次使能接收器或发送器时为止，也就是，到 C2[RE]或 C2[TE]置位为止。

地址: 基址 基准 + 1h 偏移

位	7	6	5	4	3	2	1	0
读	SBR							
写	SBR							
复位	0	0	0	0	0	1	0	0

UARTx_BDL 字段描述

字段	描述
SBR	UART 波特率位 UART 的波特率由 13 SBR 字段决定。详情参见 波特率生成 。 注： <ul style="list-style-type: none"> • 波特率发生器保持禁用，直到 C2[TE]或 C2[RE]在复位之后被首次置位。波特率发生器在 SBR = 0 时被禁用。 • 如果不写入 BDL，只写入 BDH 没有任何作用，因为写入 BDH 只是将数据暂时放置在某个位置，直到写入 SDL。 • 当 1/32 窄脉宽选择用于红外(IrDA)，则波特率字段必须为偶数，最低有效位为 0。更多详情参见 MODEM 寄存器。

45.4.3 UART 控制寄存器 1 (UARTx_C1)

该读/写寄存器控制 UART 系统的各种可选特性。

地址: 基址 基准 + 2h 偏移

位	7	6	5	4	3	2	1	0
读	LOOPS	UARTSWAI	RSRC	M	WAKE	ILT	PE	PT
写	LOOPS	UARTSWAI	RSRC	M	WAKE	ILT	PE	PT
复位	0	0	0	0	0	0	0	0

UARTx_C1 字段描述

字段	描述
7 LOOPS	<p>环路模式选择</p> <p>当 LOOPS 置位时, RxD 引脚从 UART 上断开, 发送器输出从内部连接到接收器输入。要使用循环功能, 则必须使能发送器和接收器。</p> <p>0 正常工作。 1 环路模式, 发送器输出从内部连接到接收器输入的。接收器输入由 RSRC 确定。</p>
6 UARTSWAI	<p>UART 在等待模式下停止</p> <p>0 UART 时钟继续在等待模式下运行。 1 CPU 处于等待模式的同时 UART 时钟冻结。</p>
5 RSRC	<p>接收器源选择</p> <p>除非 LOOPS 字段置位, 否则, 此字段无意义或作用。当 LOOPS 置位时, RSRC 字段决定接收器移位寄存器输入源。</p> <p>0 选择内部回环模式。接收器输入从内部连接到发送器输出。 1 单线 UART 模式, 接收器输入连接到发送引脚输入信号。</p>
4 M	<p>9 位或 8 位模式选择</p> <p>当 C7816[ISO_7816E]置位/使能时, 必须置位此字段。</p> <p>0 正常-起始+8 数据位 (MSB/LSB 优先, 由 MSBF 决定) +停止。 1 使用-起始+9 数据位 (MSB/LSB 优先, 由 MSBF 决定) +停止。</p>
3 WAKE	<p>接收器唤醒方法选择</p> <p>决定哪个条件唤醒 LUART:</p> <ul style="list-style-type: none"> • 地址标志位于接收的数据字符最高有效位位置, 或者 • 接收引脚输入信号上的空闲条件。 <p>0 空闲线路唤醒。 1 地址标志唤醒。</p>
2 ILT	<p>空闲线路类型选择</p> <p>决定接收器何时开始将逻辑 1 计为空闲字符位。计数在有效起始位后或停止位后开始。如果计数在起始位后开始, 则在停止位之前的逻辑 1 字符串可能导致空闲字符错误识别。在停止位后开始计数可避免错误闲置字符识别, 但需要合适的同步传输。</p> <p>注:</p> <ul style="list-style-type: none"> • 在 UART 设为 ILT = 1 的情况下, 逻辑 1'b0 将在接收到停止位之后自动移位, 从而使空闲计数再复位。 • 当 UART 设置用于 IDLE 线路唤醒 (RWU = 1 且 WAKE = 0) 时, ILT 对于接收器何时开始将逻辑 1 当做空闲字符位计数没有影响。在空闲线路唤醒中, 只要接收器检测到 10、11 或 12 1s 取决于 M、PE 和 C4[M10]字段, 即会识别空闲字符。 <p>0 空闲字符位计数开始于起始位之后。 1 空闲字符位计数开始于停止位之后。</p>
1 PE	<p>奇偶校验使能</p> <p>使能奇偶校验功能。奇偶校验使能后, 奇偶校验功能在停止位之前的位插入奇偶校验位。当 C7816[ISO_7816E]置位/使能时, 必须置位此字段。</p> <p>0 奇偶校验功能禁用。 1 奇偶校验功能使能。</p>

下一页继续介绍此表...

UARTx_C1 字段描述 (继续)

字段	描述
0 PT	<p>奇偶校验类型</p> <p>确定 UART 是否生成校验位，并检查是偶数校验还是奇数校验。对于偶数校验，1 中的偶数清除奇偶校验位，1 中的奇数置位奇偶校验位。对于奇数校验，1 中的奇数清除奇偶校验位，1 中的偶数置位奇偶校验位。当 C7816[ISO_7816E]置位/使能时，必须清除此字段。</p> <p>0 偶数校验。 1 奇数校验。</p>

45.4.4 UART 控制寄存器 2 (UARTx_C2)

可随时对该寄存器进行读写操作。

地址: 基址 基准 + 3h 偏移

位	7	6	5	4	3	2	1	0
读	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
写	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
复位	0	0	0	0	0	0	0	0

UARTx_C2 字段描述

字段	描述
7 TIE	<p>发送器中断或 DMA 传输使能。</p> <p>使能 S1[TDRE]，基于 C5[TDMAS]的状态生成中断请求或 DMA 传输请求。</p> <p>注: 如果 C2[TIE]和 C5[TDMAS] 均置位，则必须清除 TCIE，且不能写入 D[D]，除非是服务一个 DMA 请求。</p> <p>0 TDRE 中断和 DMA 传输请求已禁用。 1 TDRE 中断或 DMA 传输请求已使能。</p>
6 TCIE	<p>发送完成中断使能</p> <p>使能发送完成标志 S1[TC]，以生成中断请求。</p> <p>0 TC 中断请求已禁用。 1 TC 中断请求已使能。</p>
5 RIE	<p>接收器已满中断或 DMA 传输使能</p> <p>使能 S1[RDRF]，基于 C5[RDMAS]的状态生成中断请求或 DMA 传输请求。</p> <p>0 RDRF 中断和 DMA 传输请求已禁用。 1 RDRF 中断或 DMA 传输请求已使能。</p>
4 ILIE	<p>空闲线路中断使能</p> <p>使能空闲线路标志 S1[IDLE]，以生成中断请求</p>

下一页继续介绍此表...

UARTx_C2 字段描述 (继续)

字段	描述
	0 IDLE 中断请求已禁用。 1 IDLE 中断请求已使能。
3 TE	发送器使能 使能 UART 发送器。通过先清除、然后置位 TE，也可用 TE 使空闲前同步码进入队列。当 C7816[ISO_7816E]置位/使能且 C7816[TTYTYPE] = 1 时，在传输请求的数据块之后会自动清除此字段。当 TL7816[TLEN] = 0 且传输 4 个额外字符之后，会检测到此条件。 0 发送器关闭。 1 发送器开启。
2 RE	接收器使能 使能 UART 接收器。 0 接收器关闭。 1 接收器开启。
1 RWU	接收器唤醒控制 此字段可以设置为将 UART 接收器置于待机状态。发生 RWU 事件时，即是说，C1[WAKE]清零时为 IDLE 事件，C1[WAKE]置位为地址匹配事件，RWU 将自动清零。当 C7816[ISO_7816E]置位时，必须清除此字段。 注：如果通道当前未处于空闲状态，RWU 只能在 C1[WAKE] = 0 (空闲时唤醒) 条件下置位。这种情况可由 S2[RAF]决定。如果标志设为唤醒 IDLE 事件且通道已经处于空闲状态，UART 有可能会丢弃数据。这是因为在允许 IDLE 重新断言之前，在检测到 IDLE 之后，必须收到这些数据，或者检测到 LIN 断点。 0 正常工作。 1 RWU 使能唤醒功能并抑制更多的接收器中断请求。通常，硬件通过自动清除 RWU 来唤醒接收器。
0 SBK	发送断点 切换 SBK 会从下列位置发送一个分隔字符：请参考 发送分隔字符 了解不同配置的逻辑 0 数目。切换表示在分隔字符完成发送之前，清除 SBK 字段。只要 SBK 已置位，发送器就继续发送完整的分隔字符 (10、11 或 12 位，或 13 或 14 位)。请确保至少在断言此位 1 个时钟之前，断言 C2[TE]。 <ul style="list-style-type: none"> • 如果 S2[BRK13]被清除，则为 10、11 或 12 逻辑 0。 • 如果 S2[BRK13]置位，则为 13 或 14 逻辑 0。 当 C7816[ISO_7816E]置位时，必须清除此字段。 0 发送器正常工作。 1 将分隔字符加入发送队列。

45.4.5 UART 状态寄存器 1 (UARTx_S1)

S1 寄存器提供到 MCU 的输入，以产生 UART 中断或 DMA 请求。同样，这个寄存器可由 MCU 轮询，以检查这些字段的状况。要清除标志位，应在读或写入 D 寄存器之后读取状态寄存器，具体由中断标志位的类型决定。其他指令可在两个步骤之间执行，只要不影响 I/O 的处理即可，但操作顺序对于标志位清除来说非常重要。当标志位的配置是为了触发 DMA 请求时，断言 DMA 控制器的相关 DMA 完成信号会清除标志位。

注

- 如果在清除标志位之前，没有解决这种会断言标志位、中断或 DMA 请求的情况，则标志位和中断/DMA 请求会重新断言。例如，如果 DMA 或中断服务程序未能成功在传输缓冲区中写入足够的数，以让它超过水印级，则标志位会重新断言并生成另一个中断或 DMA 请求。
- 通过读取一个空数据寄存器来清除 S1 寄存器的其中一个标志位会导致 FIFO 指针出现对准偏差。接收 FIFO 清空会重新初始化指针。一种比较好的做法是：始终在 FIFO 中留一个字节，在清除标志位的过程中，此位会在最后被读取。

地址: 基址 基准 + 4h 偏移

位	7	6	5	4	3	2	1	0
读	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
写	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
复位	1	1	0	0	0	0	0	0

UARTx_S1 字段描述

字段	描述
7 TDRE	<p>发送数据寄存器空标志</p> <p>当发送缓冲器中 (D 和 C3[T8]) 的数据字数等于或少于 TWFIFO[TXWATER]指示的数量时，TDRE 置位。正在发送的字符不包含在计数之内。要将 TDRE 清零，应在 TDRE 置位的情况下读取 S1，然后写入 UART 数据寄存器(D)。为了实现更为有效的中断服务，除了将最后数值写入缓冲区，所有数据都必须写入 D/C3[T8]。然后可在写入最后数据值之前读取 S1，从而清除 TRDE 标志位。这更为有效，因为 TDRE 会重新断言，直到超过水印。因此，在写入足够的数之前，尝试在每次写入时清除 TDRE 并不高效。</p> <p>0 发送缓冲区中的数据量要大于 TWFIFO[TXWATER]指示的数值。 1 自清除标志位之后，在某些时候，发送缓冲区中的数据数量少于或等于 TWFIFO[TXWATER]指示的数值。</p>
6 TC	<p>发送完成标志</p> <p>当发送缓冲器为空且无数据、前同步码或分隔字符正在发送时，TC 置位。TC 置位后，发送数据输出信号变为空闲 (逻辑 1)。在 TC 置位时读取 S1，即会清除 TC，然后执行以下任一操作：当 C7816[ISO_7816E]置位/使能时，此字段在接收到任何 NACK 信号、在任何对应的保护时间过期前置位。</p> <ul style="list-style-type: none"> • 写入 D，以传输新数据。 • 通过清除和随后置位 C2[TE]来排列前同步码。 • 通过在 C2 中向 SBK 写入 1 来排列分隔字符。 <p>0 发送器有效 (发送数据、前同步码或断点)。 1 发送器空闲 (发送活动完成)。</p>
5 RDRF	<p>接收数据寄存器满标志</p> <p>当接收数据缓冲器中的数据字数等于或大于 RWFIFO[RXWATER]中指示的数量时，RDRF 置位。正在接收的数据字不包含在计数之内。要清除 RDRF，请在 RDRF 置位时读取 S1，然后读取 D。如需进行更为有效的中断和 DMA 操作，请使用 D/C3[T8]/ED 读取除缓冲区中最终值外的所有数据。然后读取 S1 和最终数据值，从而清除 RDRF 标志位。即使 RDRF 置位，也会继续接收数据，直到发生溢出条件时为止。当 S2[LBKDE]置位时，RDRF 不能置位。此外，在 S2[LBKDE]置位时，接收到的数据字会存储在接收缓冲区中，但是会覆盖彼此。</p>

下一页继续介绍此表...

UARTx_S1 字段描述 (继续)

字段	描述
	<p>0 接收缓冲器中的数据字数少于 RXWATER 指示的数量。</p> <p>1 自上次清除此标志位以来, 某些时刻, 接收缓冲区中的数据字数目等于或大于 RXWATER 指定的数目。</p>
4 IDLE	<p>空闲线路标志</p> <p>在清除 IDLE 标志位之后, 必须接收一个帧 (尽管无需存储在数据缓冲区内, 例如若 C2[RWU]置位), 或者在空闲条件可置位 IDLE 标志位之前, LIN 分隔字符必须置位 S2[LBKDIF]标志位。要清除 IDLE, 应在 IDLE 置位的情况下读取 UART 状态, 然后读取 D。</p> <p>当接收器输入中出现以下任意一种情况时, IDLE 置位:</p> <ul style="list-style-type: none"> • 10 个连续逻辑 1, 如果 C1[M] = 0 • 11 个连续逻辑 1, 如果 C1[M] = 1 且 C4[M10] = 0 • 12 个连续逻辑 1, 如果 C1[M] = 1、 C4[M10] = 1、 C1[PE] = 1 <p>在 7816E 置位/使能时, 不支持空闲检测, 因而此标志位会被忽略。</p> <p>注: 当 RWU 置位且 WAKE 清除时, 如果 RWUID 置位, 则空闲线路条件会置位 IDLE 标志位, 否则 IDLE 标志位不会置位。</p> <p>0 接收器输入现在处于活动状态或自上次清除 IDLE 标志后从未变成活动状态</p> <p>1 接收器输入已变成空闲状态, 或者自标志位上次被断言之后尚未被清除。</p>
3 OR	<p>接收器溢出标志</p> <p>当软件未能防止接收数据寄存器因数据溢出时, OR 将置位。在完全接收到导致缓冲器溢出的数据字的停止位之后, OR 位将立即置位, 同时阻止所有其他错误标志 (FE、NF 和 PF) 置位。移位寄存器中的数据将丢失, 但已在 UART 数据寄存器中的数据不会受影响。如果 OR 标志置位, 即使存在足够的空间, 也不会有数据存入数据缓冲器。此外, 当 OR 标志位置位时, RDRF 和 IDLE 标志位不能断言, 也就是说, 不能从无效状态转换到有效状态。要清除 OR, 请在 OR 置位时读取 S1, 然后读取 D。有关 OR 位操作的更多详情参见功能描述。如果 LBKDE 使能且检测到 LIN 断点, 要是 S2[LBKDIF]未在接收到下一个数据字符之前清除, OR 字段将断言。在 7816 模式下, 可以通过设置 C7816[ONACK]配置要返回的 NACK。</p> <p>0 自上次清除此标志位以后, 未发生溢出。</p> <p>1 自上次发生溢出后, 已发生溢出或者溢出标志位未清除。</p>
2 NF	<p>噪声标志</p> <p>当 UART 检测到接收器输入中的噪声时, 置位 NF。在溢出 或使能 LIN 分隔检测功能(S2[LBKDE] = 1) 时, NF 不会置位。如果 NF 置位, 它表示自上次清除数据字之后, 只接收到带噪声的数据字。不能保证从接收缓冲区读取的第一个数据字含有噪声或者缓冲区中只有一个数据字在接收是含有噪声, 除非接收缓冲区的深度为 1。要清除 NF, 先读取 S1, 然后读取 D。</p> <p>0 自上次清除此标志位之后, 尚未检测到噪声。如果接收缓冲区的深度大于 1, 则接收缓冲区内可能存在接收时带有噪声的数据。</p> <p>1 自上次清除此标志位之后, 已至少接收到一个带有噪音的数据字。</p>
1 FE	<p>帧错误标志</p> <p>当接受逻辑 0 作为停止位时, FE 置位。BDH[SBNS]置位时, 当接收逻辑 0 作为任一停止位时, FE 会置位。在溢出 或使能 LIN 分隔检测功能(S2[LBKDE] = 1) 时, FE 不会置位。FE 抑制更多的数据接收, 直到将其清除。要清除 FE, 请在 FE 置位的情况下读取 S1, 然后读取 D。接收缓冲区中的最后一个数据代表在帧错误使能时接收的数据。当 7816E 置位/使能时, 不支持帧错误。但是, 如果此标志位置位, 则在 7816 模式下仍不会接收数据。</p> <p>0 未检测到帧传输错误。</p> <p>1 帧传输错误</p>
0 PF	<p>奇偶校验错误标志</p>

下一页继续介绍此表...

UARTx_S1 字段描述 (继续)

字段	描述
	<p>当 PE 置位且接收到的数据的校验位和其校验位不匹配时，则 PF 置位。在溢出情况下，PF 不会置位。如果 PF 置位，它只表示自上次清除数据字之后，接收数据字时存在校验错误。不能保证从接收缓冲区读取的第一个数据字出现校验错误或者缓冲区中只有一个数据字在接收时存在校验错误，除非接收缓冲区的深度为 1。要清除 PF，先读取 S1，然后读取 D。S2[LBKDE]禁用，在接收缓冲区结构中，如果数据字在接收时出现校验错误，则该数据字会被标记。如果在读取 D 寄存器之前读取 ED 寄存器，则可获得此信息。</p> <p>0 自上次清除此标志位之后，尚未检测到校验错误。如果接收缓冲区的深度大于 1，则接收缓冲区内可能存在接收时出现校验错误的的数据。</p> <p>1 自上次清除此标志位之后，已至少接收到一个存在校验错误的的数据字。</p>

45.4.6 UART 状态寄存器 2 (UARTx_S2)

S2 寄存器提供到 MCU 的输入，以产生 UART 中断或 DMA 请求。同样，这个寄存器可由 MCU 轮询，以检查这些位的状态。这个寄存器可随时读写，但 MSBF 和 RXINV 位除外，它们只能由用户在发送和接收数据包中更改。

地址: 基址 基准 + 5h 偏移

位	7	6	5	4	3	2	1	0
读	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF
写	w1c	w1c	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF
复位	0	0	0	0	0	0	0	0

UARTx_S2 字段描述

字段	描述
7 LBKDIF	<p>LIN 断点检测中断标志</p> <p>LBKDIF 在 LBKDE 置位时置位，且可在接收器输入中检测到 LIN 分隔字符。如果 C1[M] = 0，则 LIN 分隔字符为 11 个连续逻辑 0，如果 C1[M] = 1，则为 12 个连续逻辑 0。LBKDIF 在接收到最后一个 LIN 分隔字符后置位。将 LBKDIF 清零的方法是将 1 写入该位。</p> <p>0 未检测到 LIN 分隔字符。</p> <p>1 已检测到 LIN 分隔字符。</p>
6 RXEDGIF	<p>RxD 引脚有效边沿中断标志</p> <p>RxD 引脚出现有效边沿时，RXEDGIF 将置位。如果 RXINV = 0，有效边沿降低，如果 RXINV=1 则上升。将 RXEDGIF 清零的方法是将 1 写入该位。更多详细信息参见...。 RXEDGIF 说明</p> <p>注：只有在双线模式且在来自 RxD 引脚的接收数据上才能检测到有效边沿。</p> <p>0 接收引脚上未出现有效边沿。</p> <p>1 接收引脚上已出现有效边沿。</p>
5 MSBF	<p>最高有效位优先</p> <p>置位此字段会反转在线上发送和接收的位的顺序。此字段不会影响位的极性、奇偶校验位的位置或起始位或停止位的位置。当 C7816[INIT]和 C7816[ISO7816E] 使能时，此字段自动置位，且在 T = 0 协议模式下检测到初始字符。</p>

下一页继续介绍此表...

UARTx_S2 字段描述 (继续)

字段	描述
	<p>0 LSB (bit0) 是在起始位之后发送的第一位。另外，在起始位之后接收的第一位标为 bit0。</p> <p>1 MSB (bit8、bit7 或 bit6) 是在起始位之后发送的第一位，具体取决于 C1[M]和 C1[PE]的设置。另外，在起始位之后接收的第一位标为 bit8、bit7 或 bit6，具体取决于 C1[M]和 C1[PE]的设置。</p>
4 RXINV	<p>接收数据反转</p> <p>置位该字段会反转接收数据输入的极性。在 NRZ 格式中，一个传号代表 1，一个空号代表 0 时为正常极性，反之为反相极性。在 IrDA 格式下，0 在位时间（在 1 和正常校验位下保持空闲、低位）中间由短高脉冲表示。0 在位时间（在 1 和反相校验位下保持空闲、高位）中间由短低脉冲表示。当 C7816[INIT]和 C7816[ISO7816E] 使能时，此字段自动置位，且在 T = 0 协议模式下检测到初始字符。</p> <p>注：设置 RXINV 可能造成以下情况的 RXD 输入反相：数据位、起始位和停止位、断路和闲置。当 C7816[ISO7816E]置位/使能时，只有数据位和校验位会反相。</p> <p>0 接收数据不反转。</p> <p>1 接收数据反转。</p>
3 RWUID	<p>接收唤醒空闲检测</p> <p>当 RWU 置位且 WAKE 被清除时，此字段控制唤醒接收器的空闲字符是否置位 S1[IDLE]位。当 C7816[ISO7816E]置位/使能时，必须清除此字段。</p> <p>0 检测到空闲字符时，S1[IDLE]不会置位。</p> <p>1 检测到空闲字符时，S1[IDLE]置位。</p>
2 BRK13	<p>发送分隔字符长度</p> <p>确定发送的分隔字符长度为 10、11 或 12 位，还是 13 或 14 位。查看不同配置的分隔字符长度。检查帧错误不受此字段影响。 发送分隔字符</p> <p>0 分隔字符的长度为 10、11 或 12 位。</p> <p>1 分隔字符的长度为 13 位或 14 位。</p>
1 LBKDE	<p>LIN 断点检测使能</p> <p>使能 IIN 分隔检测功能。LBKDE 置位时，禁止 S1[RDRF]、S1[NF]、S1[FE]和 S1[PF]标志置位。当 LBKDE 置位时，请查看。 溢出操作当 C7816[ISO7816E]置位时，必须清除 LBKDE。</p> <p>0 分隔字符检测禁用。</p> <p>1 当 C1[M] = 0 时，在 11 位时间检测到分隔字符，当 C1[M] = 1 时，则在 12 位时间检测到分隔字符。</p>
0 RAF	<p>接收器有效标志</p> <p>当 UART 接收器在起始位搜索的 RT1 时间周期期间检测到逻辑 0 时，RAF 置位。当接收器检测到空闲字符时（ C7816[ISO7816E]清除/禁用），清除 RAF。当 C7816[ISO7816E]使能时，如果 C7816[TTYTYPE] = 0 过期或者 C7816[TTYTYPE] = 1 过期，则 RAF 被清除。</p> <p>注：当 C7816[ISO7816E]置位且 C7816[TTYTYPE] = 0 时，可以将保护时间配置为 12。但是，如果需要发送 NACK，则数据传输实际会占用 13 ETU（第 13 个 ETU 槽是个无效缓冲区）。因此，在这种情况下，RAF 会在 ETU 实际变得无效之前先解除一个 ETU 的断言。</p> <p>0 UART 接收器空闲/无效，等待起始位。</p> <p>1 UART 接收器有效（RxD 输入非空闲）。</p>

45.4.7 UART 控制寄存器 3 (UARTx_C3)

写入 R8 不会产生任何影响。只能在发送和接收数据包期间更改 TXDIR 和 TXINV。

地址: 基址 基准 + 6h 偏移

位	7	6	5	4	3	2	1	0
读	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
写	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
复位	0	0	0	0	0	0	0	0

UARTx_C3 字段描述

字段	描述
7 R8	接收位 8 当 UART 配置为 9 位数据格式时, 也就是 C1[M] = 1 或 C4[M10] = 1 时, R8 是接收到的第 9 位数据位。R8 的值与 UARTx_D 寄存器中的当前数值相对应。要读取第 9 位, 请先读取 UARTx_C3[R8] 的值, 然后读取 UARTx_D 寄存器。
6 T8	发送位 8 当 UART 配置用于 9 位数据格式时, 也就是说, C1[M] = 1 或 C4[M10] = 1 时, T8 是已发送的第九个数据位。 注: 如果 T8 的值与上次发送的值相同, 则不必再重写 T8。在重写 T8 之前, 会一直发送相同的值。 为了正确发送第 9 位, 向 UARTx_C3[T8] 写入所需的值, 然后在 UARTx_D 寄存器中写入剩余的值。
5 TXDIR	单线模式下的发送器引脚数据方向 确定在单线操作模式中, TXD 引脚用作输入还是输出。此字段只与单线模式相关。当 C7816[ISO7816E] 置位/使能且 C7816[TTYPE] = 1 时, 此字段会在传输请求的数据块之后自动被清除。当 TL7816[TLEN] = 0 且传输 4 个额外字符之后, 会检测到此条件。此外, 如果 C7816[ISO7816E] 置位/使能, C7816[TTYPE] = 0 且正在传输一个 NACK, 则硬件会根据需要自动覆盖此字段。在这种情况下, TXDIR 不会反映与 NACK 相关的临时状态。 0 TXD 引脚在单线模式下为输入。 1 TXD 引脚在单线模式下为输出。
4 TXINV	发送数据反转。 置位该字段会反转发送数据输出的极性。在 NRZ 格式中, 一个传号代表 1, 一个空号代表 0 时为正常极性, 反之为反相极性。在 IrDA 格式中, 位时间中间的短高脉冲代表 0, 剩余的空闲低脉冲代表 1 时为正常极性, 位时间中间的短低脉冲代表 0, 剩余的空闲高脉冲代表 1 时为反相极性。当 C7816[INIT] 和 C7816[ISO7816E] 使能时, 此字段自动置位, 且在 T = 0 协议模式下检测到初始字符。 注: 置位 TXINV 会使所有发送值反转, 包括空闲、分隔、起始和停止位。在循环模式下, 如果 TXINV 置位, 则接收器在 RXINV 禁用时获得发送反转位。当 C7816[ISO7816E] 置位/使能时, 只有发送数据位和校验位会反相。 0 发送数据不反转。 1 发送数据反转。
3 ORIE	溢出错误中断使能 使能溢出错误标志 S1[OR] 以生成中断请求。

下一页继续介绍此表...

UARTx_C3 字段描述 (继续)

字段	描述
	0 OR 中断禁用。 1 OR 中断请求使能。
2 NEIE	噪声错误中断使能 使能噪声标志 S1[NF]以生成中断请求。 0 NF 中断请求禁用。 1 NF 中断请求使能。
1 FEIE	帧传输错误中断使能 使能帧传输错误标志 S1[FE]以生成中断请求。 0 FE 中断请求禁用。 1 FE 中断请求使能。
0 PEIE	奇偶校验错误中断使能 使能奇偶校验错误标志 S1[PF]以生成中断请求。 0 PF 中断请求禁用。 1 PF 中断请求使能。

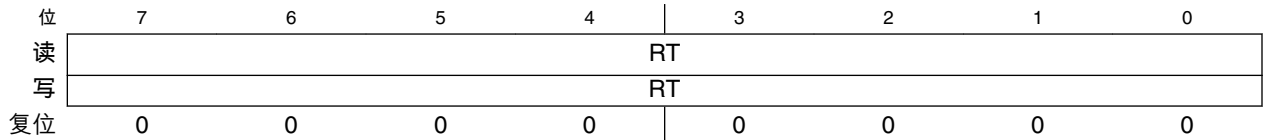
45.4.8 UART 数据寄存器 (UARTx_D)

此寄存器实际上是两个独立的寄存器。读操作返回只读接收数据寄存器的内容，写操作将内容写入只写发送数据寄存器。

注

- 在 8 位或 9 位格式中，只有 UART 数据寄存器(D)需要访问才能清除 S1[RDRF]位（假设接收器缓冲级别低于 RWFIFO[RXWATER]）如果需要捕捉第 9 位数据位，则在读取 D 寄存器之前，需要先读取 C3 寄存器。同样，如果需要捕捉数据字的额外标志位数据，则在读取 D 寄存器之前，需要先读取 ED 寄存器。
- 在常规 8 位模式（M 位被清除）下，如果校验位使能，则用户会获得 7 个数据位和一个校验位。该校验位会加载至 D 寄存器。因此，对于数据位，禁止校验位写入用户从此寄存器读取的值。
- 当以 9 位数据格式发送并使用 8 位写指令时，先执行写操作以在 UART 控制器寄存器 3 (C3[T8])中发送位 8，然后写入 D。向 C3[T8]写入会将数据保存在临时寄存器中。如果先写入 D 寄存器，然后将数据总线的新数据存入 D，则上一次写入 C3[T8]时写入的临时数值会被保存到 C3[T8]寄存器中。

地址: 基址 基准 + 7h 偏移



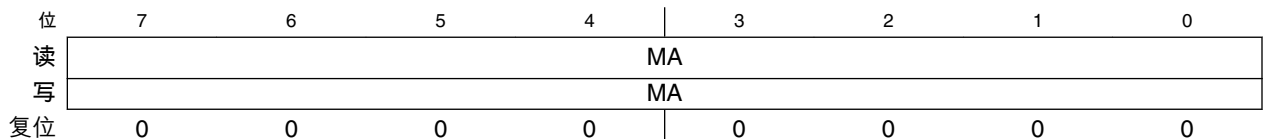
UARTx_D 字段描述

字段	描述
RT	读操作返回只读接收数据寄存器的内容，写操作将内容写入只写发送数据寄存器。

45.4.9 UART 匹配地址寄存器 1 (UARTx_MA1)

在最高有效位置位且相关联的 C4[MAEN]字段置位时，将 MA1 和 MA2 寄存器与输入数据地址进行比较。如果匹配，将把以下数据传输至数据寄存器。如果匹配失败，以下数据将被丢弃。可随时对这些寄存器进行读写操作。

地址: 基址 基准 + 8h 偏移



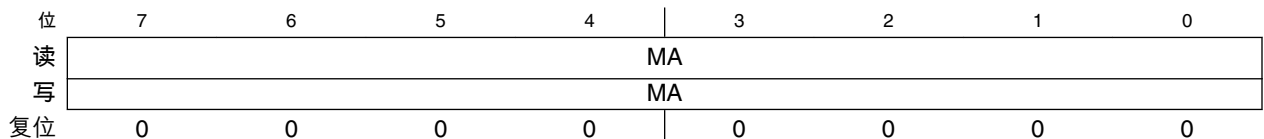
UARTx_MA1 字段描述

字段	描述
MA	匹配地址

45.4.10 UART 匹配地址寄存器 2 (UARTx_MA2)

可随时对这些寄存器进行读写操作。在最高有效位置位且相关联的 C4[MAEN]字段置位时，将 MA1 和 MA2 寄存器与输入数据地址进行比较。如果匹配，将把以下数据传输至数据寄存器。如果匹配失败，以下数据将被丢弃。

地址: 基址 基准 + 9h 偏移



UARTx_MA2 字段描述

字段	描述
MA	匹配地址

45.4.11 UART 控制寄存器 4 (UARTx_C4)

地址: 基址 基准 + Ah 偏移

位	7	6	5	4	3	2	1	0
读	MAEN1	MAEN2	M10			BRFA		
写	MAEN1	MAEN2	M10			BRFA		
复位	0	0	0	0	0	0	0	0

UARTx_C4 字段描述

字段	描述
7 MAEN1	<p>使能匹配地址模式 1</p> <p>更多详情, 请参阅匹配地址操作。</p> <p>0 如果 MAEN2 被清除, 则所有接收到的数据都会被传输至数据缓冲区。 1 在最高有效位清除时接收的所有数据均被丢弃。将在最高有效位置位时接收的所有数据与 MA1 寄存器的内容进行比较。如果不匹配, 则丢弃数据。如果匹配, 则将数据传输至数据缓冲区。当 C7816[ISO7816E]置位/使能时, 必须清除此字段。</p>
6 MAEN2	<p>使能匹配地址模式 2</p> <p>更多详情, 请参阅匹配地址操作。</p> <p>0 如果 MAEN1 被清除, 则所有接收到的数据都会被传输至数据缓冲区。 1 在最高有效位被清除的状态下接收到的所有数据均被丢弃。将在最高有效位置位时接收的所有数据与 MA2 寄存器的内容进行比较。如果不匹配, 则丢弃数据。如果匹配, 则将数据传输至数据缓冲区。当 C7816[ISO7816E]置位/使能时, 必须清除此字段。</p>
5 M10	<p>10 位模式选择</p> <p>导致第 10 位、非存储器映射位成为串行传输的一部分。这个第 10 位被生成和解析为校验位。M10 字段不会影响 LIN 发送或检测分隔行为。如果 M10 置位, 则 C1[M]和 C1[PE]也必须置位。当 C7816[ISO7816E]置位/使能时, 必须清除此字段。</p> <p>更多详情, 请参阅数据格式 (非 ISO-7816)。</p> <p>0 在串行传输中, 校验位为第 9 位。 1 在串行传输中, 校验位为第 10 位。</p>
BRFA	<p>波特率微调</p> <p>使用此位字段为平均波特率增加更多时间分辨率, 增量为 1/32。更多详情, 请参阅波特率生成。</p>

45.4.12 UART 控制寄存器 5 (UARTx_C5)

地址: 基址 基准 + Bh 偏移

位	7	6	5	4	3	2	1	0
读	TDMA5	0	RDMA5	0	LBKDDMA5		0	
写	TDMA5		RDMA5		LBKDDMA5			
复位	0	0	0	0	0	0	0	0

UARTx_C5 字段描述

字段	描述
7 TDMAS	<p>发送器 DMA 选择</p> <p>配置发送数据寄存器空标志 S1[TDRE]以生成中断或 DMA 请求 (若 C2[TIE]置位)。</p> <p>注:</p> <ul style="list-style-type: none"> • 如果 C2[TIE]清零, 那么当 TDRE 标志置位时, 无论 TDMAS 的状态如何, TDRE DMA 和 TDRE 中断请求信号的电平都不会变为有效。 • 如果 C2[TIE]和 TDMAS 均被置位, 则必须清除 C2[TCIE], 且不能写入 D[D], 除非是正在服务一个 DMA 请求。 <p>0 如果 C2[TIE]置位且 S1[TDRE]标志置位, 那么 TDRE 中断请求信号的电平会变为有效值以请求中断服务。</p> <p>1 如果 C2[TIE]置位且 S1[TDRE]标志置位, 那么 TDRE DMA 请求信号的电平会变为有效值以请求 DMA 传输。</p>
6 保留	此只读字段为保留字段且值始终为 0。
5 RDMAS	<p>接收器满 DMA 选择</p> <p>配置接收数据寄存器满标志 S1[RDRF]以生成中断或 DMA 请求 (若 C2[RIE]置位)。</p> <p>注: 如果已清除 C2[RIE], 且 S1[RDRF]置位, 则无论 RDMAS 状态如何, 都不会断言 RDRF DMA 和 RDRF 中断请求信号。</p> <p>0 如果 C2[RIE]和 S1[RDRF]置位, 那么 RDRF 中断请求信号的电平会变为有效值以请求中断服务。</p> <p>1 如果 C2[RIE]和 S1[RDRF]置位, 那么 RDRF DMA 请求信号的电平会变为有效值以请求 DMA 传输。</p>
4 保留	此只读字段为保留字段且值始终为 0。
3 LBKDDMAS	<p>LIN 断点检测 DMA 选择位</p> <p>配置 LIN 分隔检测标志 S2[LBKDIF]以生成中断或 DMA 请求 (若 BDH[LBKDIE] 置位)。</p> <p>注: 如果已清除 BDH[LBKDIE], 且 S2[LBKDIF]置位, 则无论 LBKDDMAS 状态如何, 都不会断言 LBKDIF DMA 和 LBKDIF 中断信号。</p> <p>0 如果 BDH[LBKDIE]和 S2[LBKDIF]置位, 那么 LBKDIF 中断信号的电平会变为有效值以请求中断服务。</p> <p>1 如果 BDH[LBKDIE]和 S2[LBKDIF]置位, 那么 LBKDIF DMA 请求信号的电平会变为有效值以请求 DMA 传输。</p>
保留	此只读字段为保留字段且值始终为 0。

45.4.13 UART 扩展数据寄存器 (UARTx_ED)

此寄存器中包含与接收的数据字一同存储的额外信息标志位。此寄存器可随时被读取, 但只有接收的 FIFO 中包含数据字时才会包含有效数据。

注

- 此寄存器中包含的数据代表有关接收数据字条件的额外信息。此数据的重要性因应用而异, 在某些情况下可能

完全是选择性的。这些字段会自动更新，以在读取 D 时反映下一个数据字的状况。

- 如果自上一次接收缓冲区为空开始，S1[NF]和 S1[PF]一直未置位，则 NOISY 和 PARITYE 字段将为 0。

地址: 基址 基准 + Ch 偏移

位	7	6	5	4	3	2	1	0
读	NOISY	PARITYE	0					
写	NOISY	PARITYE						
复位	0	0	0	0	0	0	0	0

UARTx_ED 字段描述

字段	描述
7 NOISY	D 和 C3[R8] 中所含当前已接收数据字在接收时存在噪声。 0 接收到的数据字无噪声。 1 接收到的数据存在噪声。
6 PARITYE	D 和 C3[R8] 中所含当前已接收数据字在接收时存在奇偶检验错误。 0 接收到的数据字无奇偶校验错误。 1 接收到的数据字存在奇偶校验错误。
保留	此只读字段为保留字段且值始终为 0。

45.4.14 UART 调制解调器寄存器 (UARTx_MODEM)

MODEM 寄存器控制用于设置调制解调器配置的选项。

注

当 C7816[ISO7816EN]使能时，需清除 RXRTSE、TXRTSPOL、TXRTSE 和 TXCTSE。这会导致 RTS 在 ISO-7816 等待期间解除断言。ISO-7816 协议不使用 RTS 和 CTS 信号。

地址: 基址 基准 + Dh 偏移

位	7	6	5	4	3	2	1	0
读	0				RXRTSE	TXRTSPOL	TXRTSE	TXCTSE
写					RXRTSE	TXRTSPOL	TXRTSE	TXCTSE
复位	0	0	0	0	0	0	0	0

UARTx_MODEM 字段描述

字段	描述
7-4 保留	此只读字段为保留字段且值始终为 0。
3 RXRTSE	接收器请求-发送使能

下一页继续介绍此表...

UARTx_MODEM 字段描述 (继续)

字段	描述
	<p>允许 RTS 输出控制发送器件的 CTS 输入，以防止接收器溢出。</p> <p>注：请勿同时置位 RXRTSE 和 TXRTSE。</p> <p>0 接收器对 RTS 不起作用。</p> <p>1 如果接收器数据寄存器 (FIFO) 中的字符数等于或大于 RWFIFO[RXWATER]中的字符数，则 RTS 被解除断言。如果接收器数据寄存器 (FIFO) 中的字符数少于 RWFIFO[RXWATER]中的字符数，则 RTS 被断言。请参见 硬件流量控制</p>
2 TXRTSPOL	<p>发送器请求-发送极性</p> <p>控制发送器 RTS 的极性。TXRTSPOL 不影响接收器 RTS 的极性。除非 TXRTSE 置位，否则，RTS 将在有效低电平状态下保持取反。</p> <p>0 发送器 RTS 为有效低电平。</p> <p>1 发送器 RTS 为有效高电平。</p>
1 TXRTSE	<p>发送器请求-发送使能</p> <p>在发送前后控制 RTS。</p> <p>0 发送器对 RTS 不起作用。</p> <p>1 当将字符存入空发送器数据缓冲器时，RTS 将在起始位发送前一位时间断言。RTS 将在发送器数据缓冲器和移位寄存器中的所有字符（包括最后停止位）完全发送后一位时间解除断言。(FIFO)</p> <p>注：请确保在断言此位之前，断言 C2[TE]。</p>
0 TXCTSE	<p>发送器清除-发送使能</p> <p>TXCTSE 控制发送器的工作模式。TXCTSE 的设置可以独立于 TXRTSE 和 RXRTSE 的状态。</p> <p>0 CTS 对发送器不起作用。</p> <p>1 使能清除-发送工作模式。发送器每次准备好发送字符时都会检查 CTS 的状态。如果 CTS 断言，则发送字符。如果 CTS 解除断言，信号 TXD 将保持标记状态，发送将延迟到 CTS 断言为止。在发送字符过程中，CTS 中的变化不会影响字符的发送。</p>

45.4.15 UART 红外寄存器 (UARTx_IR)

IR 寄存器控制用于设置红外配置的选项。

地址: 基址 基准 + Eh 偏移

位	7	6	5	4	3	2	1	0
读	0					IREN	TNP	
写	0					IREN	TNP	
复位	0	0	0	0	0	0	0	0

UARTx_IR 字段描述

字段	描述
7-3 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

UARTx_IR 字段描述 (继续)

字段	描述
2 IREN	红外使能 使能/禁用红外调制/解调。 0 IR 已禁用。 1 IR 已使能。
TNP	发送器窄脉冲 使能 UART 是否发送 1/16、3/16、1/32 或 1/4 的窄脉冲。 00 31.6 01 1/16. 10 132 11 1/4.

45.4.16 UART FIFO 参数 (UARTx_PFIFO)

此寄存器可以使程序员开启和关闭 FIFO 功能。同时还可提供已经实现的 FIFO 的大小信息。可随时对此寄存器进行读取操作。只有在 C2[RE]和 C2[TE]清零/未置位以及数据缓冲器/FIFO 为空时，才能对此寄存器进行写入操作。

地址: 基址 基准 + 10h 偏移

位	7	6	5	4	3	2	1	0
读	TXFE	TXFIFOSIZE			RXFE	RXFIFOSIZE		
写	TXFE	TXFIFOSIZE			RXFE	RXFIFOSIZE		
复位	0	*	*	*	0	*	*	*

* 注:

- TXFIFOSIZE 字段: 重置值取决于特定的 UART 实例是否支持 FIFO 并在该 FIFO 的大小之内。关于每个 UART 实例支持的 FIFO 大小, 请参见芯片配置。
- RXFIFOSIZE 字段: 重置值取决于特定的 UART 实例是否支持 FIFO 并在该 FIFO 的大小之内。关于每个 UART 实例支持的 FIFO 大小, 请参见芯片配置。

UARTx_PFIFO 字段描述

字段	描述
7 TXFE	发送 FIFO 使能 当此字段置位时,将使能发送缓冲器的内置 FIFO 结构。FIFO 结构的大小由 TXFIFOSIZE 指示。如果此字段未置位,发送缓冲器以 FIFO 模式工作,深度为一个数据字,并且不受 TXFIFOSIZE 中值的影响。必须在更改此字段之前清除 C2[TE]和 C2[RE]。此外,在更改此字段之后,必须立即发布 TXFLUSH 和 RXFLUSH 命令。 0 发送 FIFO 未使能。缓冲器为深度 1。(支持较早型号) 1 发送 FIFO 已使能。缓冲器的深度由 TXFIFOSIZE 指示。
6-4 TXFIFOSIZE	发送 FIFO。缓冲器深度

下一页继续介绍此表...

UARTx_PFIFO 字段描述 (继续)

字段	描述
	<p>可以在发送缓冲器中存储的最大发送数据字数量。此字段是只读字段。</p> <p>000 发送 FIFO/缓冲器深度 = 1 个数据字。 001 发送 FIFO/缓冲器深度 = 4 个数据字。 010 发送 FIFO/缓冲器深度 = 8 个数据字。 011 发送 FIFO/缓冲器深度 = 16 个数据字。 100 发送 FIFO/缓冲器深度 = 32 个数据字。 101 发送 FIFO/缓冲器深度 = 64 个数据字。 110 发送 FIFO/缓冲器深度 = 128 个数据字。 111 保留。</p>
3 RXFE	<p>接收 FIFO 使能</p> <p>当此字段置位时，将使能接收缓冲器的内置 FIFO 结构。FIFO 结构的大小由 RXFIFOSIZE 字段指示。如果此字段未置位，接收缓冲器以 FIFO 模式工作，深度为一个数据字，并且不受 RXFIFOSIZE 中值的影响。必须在更改此字段之前清除 C2[TE]和 C2[RE]。此外，在更改此字段之后，必须立即发布 TXFLUSH 和 RXFLUSH 命令。</p> <p>0 接收 FIFO 未使能。缓冲器为深度 1。(支持较早型号) 1 接收 FIFO 已使能。缓冲器的深度由 RXFIFOSIZE 指示。</p>
RXFIFOSIZE	<p>接收 FIFO。缓冲器深度</p> <p>在溢出发生之前，可在接收缓冲器中存入的最大接收数据字数量。此字段是只读字段。</p> <p>000 接收 FIFO/缓冲器深度 = 1 个数据字。 001 接收 FIFO/缓冲器深度 = 4 个数据字。 010 接收 FIFO/缓冲器深度 = 8 个数据字。 011 接收 FIFO/缓冲器深度 = 16 个数据字。 100 接收 FIFO/缓冲器深度 = 32 个数据字。 101 接收 FIFO/缓冲器深度 = 64 个数据字。 110 接收 FIFO/缓冲器深度 = 128 个数据字。 111 保留。</p>

45.4.17 UART FIFO 控制寄存器 (UARTx_CFIFO)

借助此寄存器，可设置 FIFO 操作的各种控制字段。可随时对此寄存器进行读写操作。注意，写入 TXFLUSH 和 RXFLUSH 可能导致数据丢失，需要小心操作，以防止出现意外/不可预测的行为。因此，在清空响应的 FIFO 之前，建议先清除 TE 和 RE。

地址: 基址 基准 + 11h 偏移

位	7	6	5	4	3	2	1	0
读	0	0	0			RXOFE	TXOFE	RXUFE
写	TXFLUSH	RXFLUSH				RXOFE	TXOFE	RXUFE
复位	0	0	0	0	0	0	0	0

UARTx_CFIFO 字段描述

字段	描述
7 TXFLUSH	<p>发送 FIFO/缓冲器清空</p> <p>写入此字段会导致存储于发送 FIFO/缓冲器中的所有数据被清空。这不会影响发送移位寄存器中的数据。</p> <p>0 不发生清空操作。 1 清空发送 FIFO/缓冲器中的所有数据。</p>
6 RXFLUSH	<p>接收 FIFO/缓冲器清空</p> <p>写入此字段会导致存储于接收 FIFO/缓冲器中的所有数据被清空。这不会影响接收移位寄存器中的数据。</p> <p>0 不发生清空操作。 1 清空接收 FIFO/缓冲器中的所有数据。</p>
5-3 保留	此只读字段为保留字段且值始终为 0。
2 RXOFE	<p>接收 FIFO 溢出中断使能</p> <p>此字段置位时，RXOF 标志会向主机生成一个中断。</p> <p>0 RXOF 标志不会向主机生成一个中断。 1 RXOF 标志会向主机生成一个中断。</p>
1 TXOFE	<p>发送 FIFO 溢出中断使能</p> <p>此字段置位时，TXOF 标志会向主机生成一个中断。</p> <p>0 TXOF 标志不会向主机生成一个中断。 1 TXOF 标志会向主机生成一个中断。</p>
0 RXUFE	<p>接收 FIFO 下溢中断使能</p> <p>此字段置位时，RXUF 标志会向主机生成一个中断。</p> <p>0 RXUF 标志不会向主机生成一个中断。 1 RXUF 标志会向主机生成一个中断。</p>

45.4.18 UART FIFO 状态寄存器 (UARTx_SFIFO)

此寄存器提供关于发送和接收缓冲区/FIFO 的状态信息，包括中断信息。可随时对此寄存器进行读写操作。

地址: 基址 基准 + 12h 偏移

位	7	6	5	4	3	2	1	0
读	TXEMPT	RXEMPT	0			RXOF	TXOF	RXUF
写	TXEMPT	RXEMPT				w1c	w1c	w1c
复位	1	1	0	0	0	0	0	0

UARTx_SFIFO 字段描述

字段	描述
7 TXEMPT	<p>发送缓冲器/FIFO 空</p> <p>当发送 FIFO/缓冲器中无数据时断言。此字段不会考虑发送移位寄存器中的数据。</p> <p>0 发送缓冲器不为空。 1 发送缓冲器为空。</p>
6 RXEMPT	<p>接收缓冲器/FIFO 空</p> <p>当接收 FIFO/缓冲器中无数据时断言。此字段不会考虑接收移位寄存器中的数据。</p> <p>0 接收缓冲器不为空。 1 接收缓冲器为空。</p>
5-3 保留	此只读字段为保留字段且值始终为 0。
2 RXOF	<p>接收缓冲区溢出标志</p> <p>指示写入接收缓冲器的数据超过了其存储能力。无论 CFIFO[RXOFE]的值为何，此字段都会断言。然而，只有在 CFIFO[RXOFE]置位时才会向主机发送一个中断。通过写入 1 可将此标志清零。</p> <p>0 自标志上次清零以后，未发生接收缓冲器溢出。 1 自标志上次清零以后，发生了至少一次接收缓冲器溢出。</p>
1 TXOF	<p>发送器缓冲器溢出标志</p> <p>指示写入发送缓冲器的数据超过了其存储能力。无论 CFIFO[TXOFE]的值为何，此字段都会断言。然而，只有在 CFIFO[TXOFE]置位时才会向主机发送一个中断。通过写入 1 可将此标志清零。</p> <p>0 自标志上次清零以后，未发生发送缓冲器溢出。 1 自标志上次清零以后，发生了至少一次发送缓冲器溢出。</p>
0 RXUF	<p>接收器缓冲器下溢标志</p> <p>指示从接收缓冲器读取的数据多于存在的数据。无论 CFIFO[RXUFE]的值为何，此字段都会断言。然而，只有在 CFIFO[RXUFE]置位时才会向主机发送一个中断。通过写入 1 可将此标志清零。</p> <p>0 自标志上次清零以后，未发生接收缓冲器下溢。 1 自标志上次清零以后，发生了至少一次接收缓冲器下溢。</p>

45.4.19 UART FIFO 发送水印 (UARTx_TWFIFO)

通过此寄存器可以设定一个可配置阈值，用于通知需要额外发送数据的情况。此寄存器可以随时读取，但只能在 C2[TE]未置位时写入。更改水印的值不会清除 S1[TDRE]标志位。

地址: 基址 基准 + 13h 偏移

位	7	6	5	4	3	2	1	0
读	TXWATER							
写	TXWATER							
复位	0	0	0	0	0	0	0	0

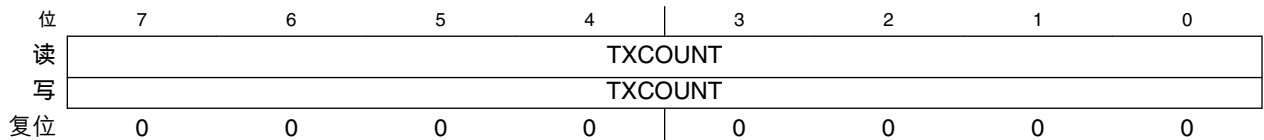
UARTx_TWFIFO 字段描述

字段	描述
TXWATER	<p>发送水印</p> <p>当发送 FIFO/缓冲器中的数据字数等于或少于此寄存器字段中的值时，将如同 C5[TDMA5]和 C2[TIE]确定的一样生成中断（通过 S1[TDRE]）或 DMA 请求（通过 C5[TDMA5]）。为确保正常工作，TXWATER 中的值必须小于由 PFIFO[TXFIFOSIZE]和 PFIFO[TXFE]指示的发送缓冲器/FIFO 大小。</p>

45.4.20 UART FIFO 发送计数 (UARTx_TCFIFO)

这是一个只读寄存器，显示发送缓冲区/FIFO 中目前存在多少个数据字。可随时对此寄存器进行读取操作。

地址: 基址 基准 + 14h 偏移



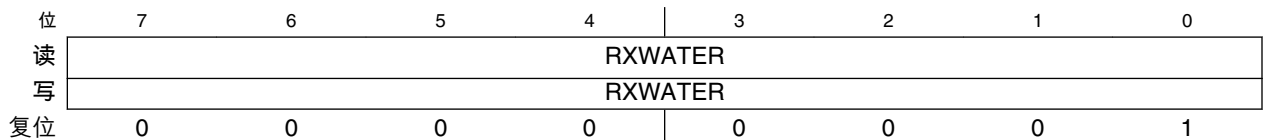
UARTx_TCFIFO 字段描述

字段	描述
TXCOUNT	<p>发送计数器</p> <p>此寄存器中的值指示发送 FIFO/缓冲器中的数据字数量。如果一个数据字处于发送状态，即在发送移位寄存器中，则不会纳入计数。此值可与 PFIFO[TXFIFOSIZE]配合使用，以计算发送 FIFO/缓冲器中剩余的空间有多大。</p>

45.4.21 UART FIFO 接收水印 (UARTx_RWFIFO)

通过此寄存器可以设定一个可配置阈值，用于通知需要移除接收器 FIFO/缓冲区中数据的情况。此寄存器可以随时读取，但只能在 C2[RE]未断言时写入。更改此寄存器中的值不会清除 S1[RDRF]。

地址: 基址 基准 + 15h 偏移



UARTx_RWFIFO 字段描述

字段	描述
RXWATER	接收水印

UARTx_RWFIFO 字段描述 (继续)

字段	描述
	当接收 FIFO/缓冲器中的数据字数等于或多于此寄存器字段中的值时,将如同 C5[RDMA5]和 C2[RIE]确定的一样生成中断 (通过 S1[RDRF]) 或 DMA 请求 (通过 C5[RDMA5])。为确保正常工作, RXWATER 中的值必须小于由 PFIFO[RXFIFOSIZE]和 PFIFO[RXFE]指示的接收 FIFO/缓冲器大小,且不得大于 0。

45.4.22 UART FIFO 接收计数 (UARTx_RCFIFO)

这是一个只读寄存器,显示接收 FIFO/缓冲区中目前存在多少个数据字。可随时对此寄存器进行读取操作。

地址: 基址 基准 + 16h 偏移

位	7	6	5	4	3	2	1	0
读	RXCOUNT							
写	RXCOUNT							
复位	0	0	0	0	0	0	0	0

UARTx_RCFIFO 字段描述

字段	描述
RXCOUNT	接收计数器 此寄存器中的值指示接收 FIFO/缓冲器中的数据字数。如果一个数据字处于接收状态,即在接收移位寄存器中,则不会纳入计数。此值可与 PFIFO[RXFIFOSIZE]配合使用,以计算接收 FIFO/缓冲器中剩余的空间有多大。

45.4.23 UART 7816 控制寄存器 (UARTx_C7816)

C7816 寄存器是适用于 ISO-7816 特定功能的主控制寄存器。此寄存器专门用于 7816 功能,此寄存器中的数值对 UART 操作没有影响,如果 ISO_7816E 没有置位/使能,则可忽略。此寄存器可以随时读取,但只能在 ISO_7816E 未置位时更改数值。

地址: 基址 基准 + 18h 偏移

位	7	6	5	4	3	2	1	0
读	0			ONACK	ANACK	INIT	TTYPE	ISO_7816E
写	0			ONACK	ANACK	INIT	TTYPE	ISO_7816E
复位	0	0	0	0	0	0	0	0

UARTx_C7816 字段描述

字段	描述
7-5 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

UARTx_C7816 字段描述 (继续)

字段	描述
4 ONACK	<p>在溢出时生成 NACK</p> <p>当此字段置位时，如果如同 S1[OR]所述出现接收缓冲区溢出，则接收器会自动生成一个 NACK 响应。在许多系统中，这会导致发送器重新发送溢出的数据包，直到达到该发送器的重发阈值。只有当 TTYPE=0 时才会生成 NACK。此字段的操作与 ANACK 无关。请参见 溢出 NACK 考虑因素</p> <p>0 当数据接收导致出现溢出事件时，收到的数据不会生成一个 NACK。 1 如果接收缓冲区溢出，则在收到字符时即自动发送一个 NACK。</p>
3 ANACK	<p>在出错时生成 NACK</p> <p>此字段置位时，如果校验位出错或 INIT 置位且检测到无效的初始字符，则接收器会自动生成一个 NACK 响应。只有在 TTYPE = 0 时才会生成 NACK。如果 ANACK 置位，则 UART 试图无限重新发送数据。为了停止这种重新发送尝试，清除 C2[TE]或 ISO_7816E，且不要置位，直到 S1[TC]重新置位 C2[TE]。</p> <p>0 没有自动生成 NACK。 1 如果检测到校验位错误或检测到无效的初始字符，则会自动生成一个 NACK。</p>
2 INIT	<p>检测初始字符</p> <p>此字段置位时，会针对所有接收到的字符执行有效初始字符搜索。如果搜索到无效的初始字符，且 ANACK 已置位，则会发送一个 NACK。所有接收到的数据均被丢弃，错误标志位被阻拦(S1[NF], S1[OR], S1[FE], S1[PF], IS7816[WT], IS7816[CWT], IS7816[BWT], IS7816[ADT], IS7816[GTV])，直到检测到有效的初始字符。检测到有效的初始字符时，配置数值 S2[MSBF]、C3[TXINV]和 S2[RXINV]会自动更新，以显示接收到的初始字符。实际的 INIT 数据值不会存储在接收缓冲区中。此外，检测到有效的初始字符时，IS7816[INITD]置位，且会按照 IE7816[INITDE]的设置发出中断。检测到有效初始字符时，INIT 会自动被清除。这种初始字符检测功能只在 T=0 的协议模式下受到支持。</p> <p>0 正常操作模式。接收器不会试图确认初始字符。 1 接收器会搜索初始字符。</p>
1 TTYPE	<p>传输类型</p> <p>显示使用的传输协议。</p> <p>更多详情参见 ISO-7816 / 智能表支持。</p> <p>0 T = 0，根据 ISO-7816 规格。 1 T = 1，根据 ISO-7816 规格。</p>
0 ISO_7816E	<p>ISO-7816 功能性使能</p> <p>表示 UART 正按照 ISO-7816 协议的要求执行操作。</p> <p>注：此字段只有在无发送或接收时才能更改。如果在数据传输期间更改此字段，则正在发送或接收的数据可能会出现传输错误。</p> <p>0 ISO-7816 功能关闭/未使能。 1 ISO-7816 功能开启/使能。</p>

45.4.24 UART 7816 中断使能寄存器 (UARTx_IE7816)

IE7816 寄存器管控由哪个标志位导致发出中断。此寄存器专门针对 7816 功能，驱动中断的相应标志位在 7816E 未置位/使能时不会断言。但是，如果这些标志位在 7816 置位期间被断言，且之后未被清除，则它们会保持置位。可随时对此寄存器进行读写操作。

地址: 基址 基准 + 19h 偏移

位	7	6	5	4	3	2	1	0
读	WTE	CWTE	BWTE	INITDE	ADTE	GTVE	TXTE	RXTE
写	WTE	CWTE	BWTE	INITDE	ADTE	GTVE	TXTE	RXTE
复位	0	0	0	0	0	0	0	0

UARTx_IE7816 字段描述

字段	描述
7 WTE	等待定时器中断使能 0 断言 IS7816[WT]不会生成一个中断。 1 断言 IS7816[WT]会导致生成一个中断。
6 CWTE	字符等待定时器中断使能 0 断言 IS7816[CWT]不会生成一个中断。 1 断言 IS7816[CWT]会导致生成一个中断。
5 BWTE	数据块等待定时器中断使能 0 断言 IS7816[BWT]不会生成一个中断。 1 断言 IS7816[BWT]会导致生成一个中断。
4 INITDE	初始字符检测中断使能 0 断言 IS7816[INITD]不会生成一个中断。 1 断言 IS7816[INITD]会导致生成一个中断。
3 ADTE	ART 持续定时器中断使能 0 断言 IS7816[ADT]不会生成一个中断。 1 断言 IS7816[ADT]会导致生成一个中断。
2 GTVE	保护定时器违规中断使能 0 断言 IS7816[GTV]不会生成一个中断。 1 断言 IS7816[GTV]会导致生成一个中断。
1 TXTE	发送阈值超过中断使能 0 断言 IS7816[TXT]不会生成一个中断。 1 断言 IS7816[TXT]会导致生成一个中断。
0 RXTE	接收阈值超过中断使能 0 断言 IS7816[RXT]不会生成一个中断。 1 断言 IS7816[RXT]会导致生成一个中断。

45.4.25 UART 7816 中断状态寄存器 (UARTx_IS7816)

IS7816 寄存器提供了一种读取和清除中断标志位的机制。在字段位置写入 1 即可清除所有标志位/中断。写入 0 没有影响。所有位都是“粘滞”位，表示它们只显示自此位上次被清除后出现的标志位状况，不会显示位当前存在的状况。无论 IE7816 中的相应字段是置位或被清除，状态标志位都会置位。IE7816 只管控是否向主机处理器发出中断。此寄存器专门用于 7816 功能，此寄存器中的数值对 UART 操作没有影响，如果 7816E 没有置位/使能，则可忽略。可随时对此寄存器进行读写操作。

地址: 基址 基准 + 1Ah 偏移

位	7	6	5	4	3	2	1	0
读	WT	CWT	BWT	INITD	ADT	GTV	TXT	RXT
写	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
复位	0	0	0	0	0	0	0	0

UARTx_IS7816 字段描述

字段	描述
7 WT	<p>等待定时器中断</p> <p>表示等待时间、发送字符的前沿和下一个响应字符前沿之间的时间间隔已经超过设定值。只有当 C7816[TTYE] = 0 时，此标志才会进行断言。写入 1 即可清除此中断。</p> <p>0 未违反等待时间(WT)。 1 已违反等待时间(WT)。</p>
6 CWT	<p>字符等待定时器中断</p> <p>表示字符等待时间、数据块中两个连续字符的前沿之间的时间间隔已经超过设定值。只有当 C7816[TTYE] = 1 时，此标志才会进行断言。写入 1 即可清除此中断。</p> <p>0 未违反字符等待时间(CWT)。 1 已违反字符等待时间(CWT)。</p>
5 BWT	<p>数据块等待定时器中断</p> <p>表示数据块等待时间、收到数据块的首个字符前沿和之前发送数据块的最后一个字符前沿之间的时间间隔已经超过设定值。只有当 C7816[TTYE] = 1 时，此标志才会进行断言。写入 1 即可清除此中断。</p> <p>0 未违反数据块等待时间(BWT)。 1 已违反数据块等待时间(BWT)。</p>
4 INITD	<p>初始字符检测中断</p> <p>表示已收到有效的初始字符。通过写入 1 清除此中断。</p> <p>0 未收到有效的初始字符。 1 已收到有效的初始字符。</p>
3 ADT	<p>ART 持续时间中断</p> <p>表示 ATR 持续时间、接收的 TS 字符前沿和下一个响应字符前沿之间的时间间隔已经超过设定值。只有当 C7816[TTYE] = 0 时，此标志才会进行断言。写入 1 即可清除此中断。</p>

下一页继续介绍此表...

UARTx_IS7816 字段描述 (继续)

字段	描述
	0 未违反 ATR 持续时间 (ADT)。 1 已违反 ATR 持续时间 (ADT)。
2 GTV	保护定时器违规中断 表示已违反一个或多个字符保护时间、数据块保护时间或保护时间。通过写入 1 清除此中断。 0 未违反一项保护时间 (GT、CGT 或 BGT)。 1 已违反一项保护时间 (GT、CGT 或 BGT)。
1 TXT	发送阈值超过中断 表示发送 NACK 阈值如 ET7816[TXTHRESHOLD]所示超出限值。无论此标志位是否置位, UART 都持续无限重新发送。此标志位只在 C7816[TTYTYPE] = 0 时进行断言。如果 7816E 被清除/禁用, ANACK 被清除/禁用, C2[TE]被清除/禁用, C7816[TTYTYPE] = 1, 或者无需接收 NACK 即可传输数据包, 则内部 NACK 检测计数器被清零且对下一个接收的 NACK 重新从 0 开始计数。通过写入 1 清除此中断。 0 重试次数和相应的 NACKS 不超过 ET7816[TXTHRESHOLD]中的值。 1 重试次数和相应的 NACKS 超过 ET7816[TXTHRESHOLD]中的值。
0 RXT	接收阈值超过中断 表示在对收到数据的校验位错误进行响应时生成的连续 NACKS 数量要大于 ET7816[RXTHRESHOLD]。此标志位要求 ANACK 置位。此外, 此标志位只会在 C7816[TTYTYPE] = 0 时断言。清除此字段也会复位跟踪连续 NACKS 的计数器。无论此标志位是否置位, UART 会继续试图接收数据。如果 7816E 被清除/禁用, RE 被清除/禁用, C7816[TTYTYPE] = 1, 或者无需发出 NACK 即可接收数据包, 则内部 NACK 检测计数器被清零且对下一个发送的 NACK 重新从 0 开始计数。通过写入 1 清除此中断。 0 因为校验位错误和缓冲区溢出而生成的连续 NACKS 的数目少于或等于 ET7816[RXTHRESHOLD]中的数值。 1 因为校验位错误和缓冲区溢出而生成的连续 NACKS 的数目大于 ET7816[RXTHRESHOLD]中的数值。

45.4.26 UART 7816 等待参数寄存器 (UARTx_WP7816)

WP7816 寄存器包含生成数据块等待定时器时使用的 WTX 变量。可随时对此寄存器进行读取操作。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作。

地址: 基址 基准 + 1Bh 偏移

位	7	6	5	4	3	2	1	0
读	WTX							
写	WTX							
复位	0	0	0	0	0	0	0	0

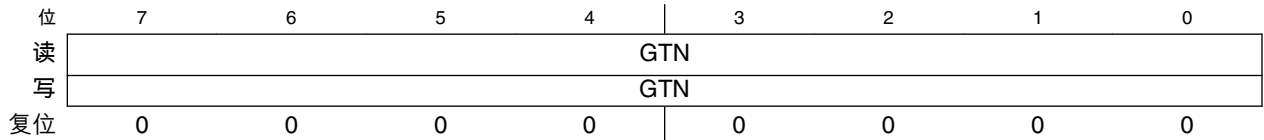
UARTx_WP7816 字段描述

字段	描述
WTX	等待时间乘数(C7816[TTYTYPE] = 1) 用于计算 BWT 计数器所用的数值。它表示 0 和 255 之间的数值。只有在 C7816[TTYTYPE] = 1 时才使用此值。参见 等待时间和保护时间参数 。

45.4.27 UART 7816 等待 N 寄存器 (UARTx_WN7816)

WN7816 寄存器包含一个用于计算保护时间计数器的参数。可随时对此寄存器进行读取操作。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作。

地址: 基址 基准 + 1Ch 偏移



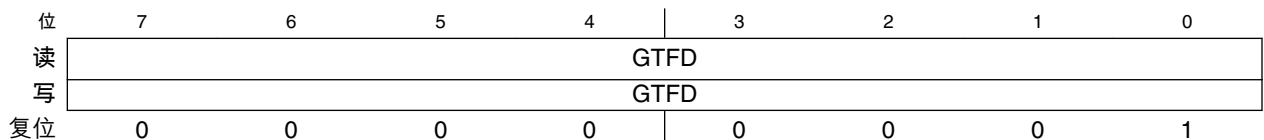
UARTx_WN7816 字段描述

字段	描述
GTN	保护带 N 定义在计算 GT、CGT 和 BGT 计数器期间使用的参数。此数值代表 0 和 255 之间的一个整数。参见 等待时间和保护时间参数 。

45.4.28 UART 7816 等待 FD 寄存器 (UARTx_WF7816)

WF7816 包含生成各种计数器（包括 GT、CGT、BGT、WT 和 BWT）时所用的参数。可随时对此寄存器进行读取操作。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作。

地址: 基址 基准 + 1Dh 偏移



UARTx_WF7816 字段描述

字段	描述
GTFD	FD 乘数 用作计算 BWT 的另一个乘数。它表示 1 和 255 之间的数值。0 值无效。此值不用于波特率生成。请参见 等待时间和保护时间参数 和 波特率生成 。

45.4.29 UART 7816 错误阈值寄存器 (UARTx_ET7816)

ET7816 寄存器包含用于确定在通知主机处理器之前必须接收或发送的 NACK 数目的字段。可随时对此寄存器进行读取操作。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作。

地址: 基址 基准 + 1Eh 偏移

位	7	6	5	4	3	2	1	0
读	TXTHRESHOLD				RXTHRESHOLD			
写	TXTHRESHOLD				RXTHRESHOLD			
复位	0	0	0	0	0	0	0	0

UARTx_ET7816 字段描述

字段	描述
7-4 TXTHRESHOLD	<p>发送 NACK 阈值</p> <p>写入此字段的数值表示在通知主机处理器之前尝试发送字符的最大失败次数 (NACK)。此字段只在 C7816[TTYTYPE] = 0 且 C7816[ANACK] = 1 时有意义。从此字段读取的数值表示的是自上次成功发送以来所接收的连续 NACK 的数目。计数器在 4'hF 时达到饱和且不会回绕。无论接收了多少个 NACK，UART 都持续不断重新发送。只有当 C7816[TTYTYPE] = 0 时此标志位才会进行断言。如需了解其他信息，请参考 IS7816[TXT]字段描述。</p> <p>0 在接收到首个 NACK 时，TXT 进行断言。 1 在接收到第二个 NACK 时，TXT 进行断言。</p>
RXTHRESHOLD	<p>接收 NACK 阈值</p> <p>写入此字段的数值表示在通知主机处理器之前，因为校验位错误或接收器缓冲区溢出导致生成的连续 NACK 的最大数目。当计数器超过此字段中的数值时，IS7816[RXT]即被断言。此字段只在 C7816[TTYTYPE] = 0 时有意义。从此字段读取的数值表示的是自上次成功接收以来所发送的连续 NACK 的数目。计数器在 4'hF 时达到饱和且不会回绕。无论发送了多少 NACK，UART 都持续不断地接收有效数据包。如需了解其他信息，请参考 IS7816[RXT]字段描述。</p>

45.4.30 UART 7816 发送长度寄存器 (UARTx_TL7816)

TL7816 寄存器用于指明发送的数据块中包含的字符数。只有当 C7816[TTYTYPE] = 1 时，才使用此寄存器。可随时读取此寄存器。此寄存器只能在 C2[TE]未使能时写入。

地址: 基址 基准 + 1Fh 偏移

位	7	6	5	4	3	2	1	0
读	TLEN							
写	TLEN							
复位	0	0	0	0	0	0	0	0

UARTx_TL7816 字段描述

字段	描述
TLEN	<p>传输长度</p> <p>此值加上四表示发送的数据块中包含的字符数。对于数据块信息字段部分的每个字符，此寄存器自动减去 1。此外，对于结语字段中的 CRC 的首个字符，此寄存器自动减去 1。因此，如果正在传输 LRC，则必须采用数据包中的字节数来设置此寄存器，而如果正在发送 CRC，则采用字节数+1。对于假定为前言字段一部分的字符，则无需减少此寄存器，也就是说，数据块中传输的前三个字符，或结语字段中的 LRC 或最后的 CRC 字符，也就是传输的最后一个字符。只有当 C2[TE]被清除后，才能设定或调节此字段。</p>

45.4.31 UART 7816 ATR 持续定时器寄存器 A (UARTx_AP7816A_T0)

AP7816A_T0 寄存器包含在生成 ATR 持续定时器时使用的变量。可随时对此寄存器进行读取操作。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作，写入 0 以清除 ADT 计数器时除外。

注

ADT 计数器在检测到完整 TS 字符时开始计数。此时必须注意，自 TS 字符的起始位起已经过 10 个 ETU。在对此寄存器进行编程时，用户必须将其考虑在内。

地址: 基址 基准 + 3Ah 偏移

位	7	6	5	4	3	2	1	0
读	ADTI_H							
写	ADTI_H							
复位	0	0	0	0	0	0	0	0

UARTx_AP7816A_T0 字段描述

字段	描述
ADTI_H	<p>ATR 持续时间整数高(C7816[TTYPER] = 0)</p> <p>用于计算 ADT 计数器所用的数值。此寄存器字段提供 16 位 ATR 持续时间整数字段 ADTI (由 {AP7816A_T0[ADTI_H]、AP7816B_T0[ADTI_L]}构成) 的最高有效字节。设定 ADTI = 0 这样一个数值会禁用 ADT 计数器。只有在 C7816[TTYPER] = 0 时才使用此值。参见 ATR 持续时间计数器。</p>

45.4.32 UART 7816 ATR 持续定时器寄存器 B (UARTx_AP7816B_T0)

AP7816B_T0 寄存器包含在生成 ATR 持续定时器时使用的变量。可随时对此寄存器进行读取操作。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作，写入 0 以清除 ADT 计数器时除外。

注

ADT 计数器在检测到完整 TS 字符时开始计数。此时必须注意，自 TS 字符的起始位起已经过 10 个 ETU。在对此寄存器进行编程时，用户必须将其考虑在内。

地址: 基址 基准 + 3Bh 偏移

位	7	6	5	4	3	2	1	0
读	ADTI_L							
写	ADTI_L							
复位	0	0	0	0	0	0	0	0

UARTx_AP7816B_T0 字段描述

字段	描述
ADTI_L	ATR 持续时间整数低(C7816[TTYPER] = 0) 用于计算 ADT 计数器所用的数值。此寄存器字段提供 16 位 ATR 持续时间整数字段 ADTI (由 {AP7816A_T0[ADTI_H]、AP7816B_T0[ADTI_L]} 构成) 的最低有效字节。设定 ADTI = 0 这样一个数值会禁用 ADT 计数器。只有在 C7816[TTYPER] = 0 时才使用此值。参见 ATR 持续时间计数器 。

45.4.33 UART 7816 等待参数寄存器 A (UARTx_WP7816A_T0)

WP7816A_T0 寄存器包含用于生成各种等待时间计数器的常数。为了节省寄存器空间，当 C7816[TTYPER] = 0 和 C7816[TTYPER] = 1 时，采用不同的使用方法。可随时读取此寄存器。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作。

地址: 基址 基准 + 3Ch 偏移

位	7	6	5	4	3	2	1	0
读	WI_H							
写	WI_H							
复位	0	0	0	0	0	0	0	0

UARTx_WP7816A_T0 字段描述

字段	描述
WI_H	等待时间整数高 (C7816[TTYPER] = 0) 用于计算 WT 计数器所用的数值。此寄存器字段提供 16 位 ATR 等待时间整数字段 WI (由 {WP7816A_T0[WI_H]、WP7816B_T0[WI_L]} 构成) 的最高有效字节。WI = 0 这一数值无效，且不能被设定。只有在 C7816[TTYPER] = 0 时才使用此值。参见 等待时间和保护时间参数 。

45.4.34 UART 7816 等待参数寄存器 A (UARTx_WP7816A_T1)

WP7816A_T1 寄存器包含用于生成各种等待时间计数器的常数。为了节省寄存器空间，当 C7816[TTYPER] = 0 和 C7816[TTYPER] = 1 时，采用不同的使用方法。可随时读取此寄存器。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作。

地址: 基址 基准 + 3Ch 偏移

位	7	6	5	4	3	2	1	0
读	BWI_H							
写	BWI_H							
复位	0	0	0	0	0	0	0	0

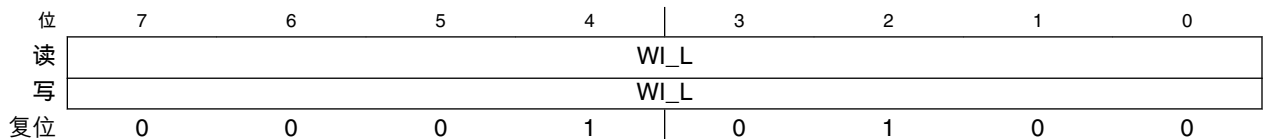
UARTx_WP7816A_T1 字段描述

字段	描述
BWI_H	数据块等待时间整数高 (C7816[TTYPE] = 1) 用于计算 BWT 计数器所用的数值。此寄存器字段提供 16 位数据块 ATR 等待时间整数字段 BWI (由 {WP7816A_T1[BWI_H], WP7816B_T1[BWI_L]} 构成) 的最高有效字节。BWI = 0 这一数值无效, 且不应被设定。只有在 C7816[TTYPE] = 1 时才使用此值。参见 等待时间和保护时间参数 。

45.4.35 UART 7816 等待参数寄存器 B (UARTx_WP7816B_T0)

WP7816B_T0 寄存器包含用于生成各种等待时间计数器的常数。为了节省寄存器空间, 当 C7816[TTYPE] = 0 和 C7816[TTYPE] = 1 时, 采用不同的使用方法。可随时读取此寄存器。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作。

地址: 基址 基准 + 3Dh 偏移



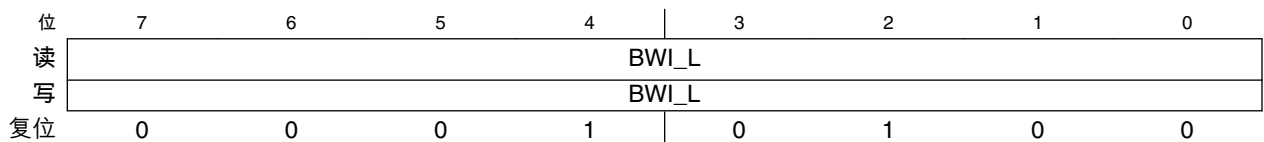
UARTx_WP7816B_T0 字段描述

字段	描述
WI_L	等待时间整数低 (C7816[TTYPE] = 0) 用于计算 WT 计数器所用的数值。此寄存器字段提供 16 位 ATR 等待时间整数字段 WI (由 {WP7816A_T0[WI_H], WP7816B_T0[WI_L]} 构成) 的最低有效字节。WI = 0 这一数值无效, 且不能被设定。只有在 C7816[TTYPE] = 0 时才使用此值。参见 等待时间和保护时间参数 。

45.4.36 UART 7816 等待参数寄存器 B (UARTx_WP7816B_T1)

WP7816B_T1 寄存器包含用于生成各种等待时间计数器的常数。为了节省寄存器空间, 当 C7816[TTYPE] = 0 和 C7816[TTYPE] = 1 时, 采用不同的使用方法。可随时读取此寄存器。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作。

地址: 基址 基准 + 3Dh 偏移



UARTx_WP7816B_T1 字段描述

字段	描述
BWI_L	数据块等待时间整数低 (C7816[TTYPE] = 1)

UARTx_WP7816B_T1 字段描述 (继续)

字段	描述
	用于计算 BWT 计数器所用的数值。此寄存器字段提供 16 位数据块 ATR 等待时间整数字段 BWI (由 {WP7816A_T1[BWI_H], WP7816B_T1[BWI_L]} 构成) 的最低有效字节。BWI = 0 这一数值无效, 且不应被设定。只有在 C7816[TTYE] = 1 时才使用此值。参见 等待时间和保护时间参数 。

45.4.37 UART 7816 等待和保护参数寄存器 (UARTx_WGP7816_T1)

WGP7816_T1 寄存器包含用于生成各种等待和保护定时器计数器的常数。可随时对此寄存器进行读取操作。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作。

地址: 基址 基准 + 3Eh 偏移

位	7	6	5	4	3	2	1	0
读	CWI1				BGI			
写	CWI1				BGI			
复位	0	0	0	0	0	1	1	0

UARTx_WGP7816_T1 字段描述

字段	描述
7-4 CWI1	字符等待时间整数 1 (C7816[TTYE] = 1) 用于计算 CWT 计数器所用的数值。它表示 0 和 15 之间的数值。只有在 C7816[TTYE] = 1 时才使用此值。参见 等待时间和保护时间参数 。
BGI	数据块保护时间整数 (C7816[TTYE] = 1) 用于计算 BGT 计数器所用的数值。它表示 0 和 15 之间的数值。只有在 C7816[TTYE] = 1 时才使用此值。参见 等待时间和保护时间参数 。

45.4.38 UART 7816 等待参数寄存器 C (UARTx_WP7816C_T1)

WP7816C_T1 寄存器包含用于生成各种等待定时器计数器的常数。C_T 可随时对此寄存器进行读取操作。只应在 C7816[ISO_7816E]未置位时对此寄存器执行写操作。

地址: 基址 基准 + 3Fh 偏移

位	7	6	5	4	3	2	1	0
读	0				CWI2			
写	0				CWI2			
复位	0	0	0	0	1	0	1	1

UARTx_WP7816C_T1 字段描述

字段	描述
7-5 保留	此只读字段为保留字段且值始终为 0。
CWI2	字符等待时间整数 2 (C7816[TTYTYPE] = 1) 用于计算 CWT 计数器所用的数值。它表示 0 和 31 之间的数值。只有在 C7816[TTYTYPE] = 1 时才使用此值。参见 等待时间 和 保护时间参数 。

45.5 功能说明

本节提供 UART 块的完整功能说明。

UART 允许 CPU 和远程设备（包括其他 CPU）之间进行全双工、异步、NRZ 串行通信。尽管 UART 发送器和接收器使用相同的波特率发生器，但它们是独立运行的。CPU 监测 UART 的状态，写入要发送的数据，并处理接收到的数据。

45.5.1 发送器

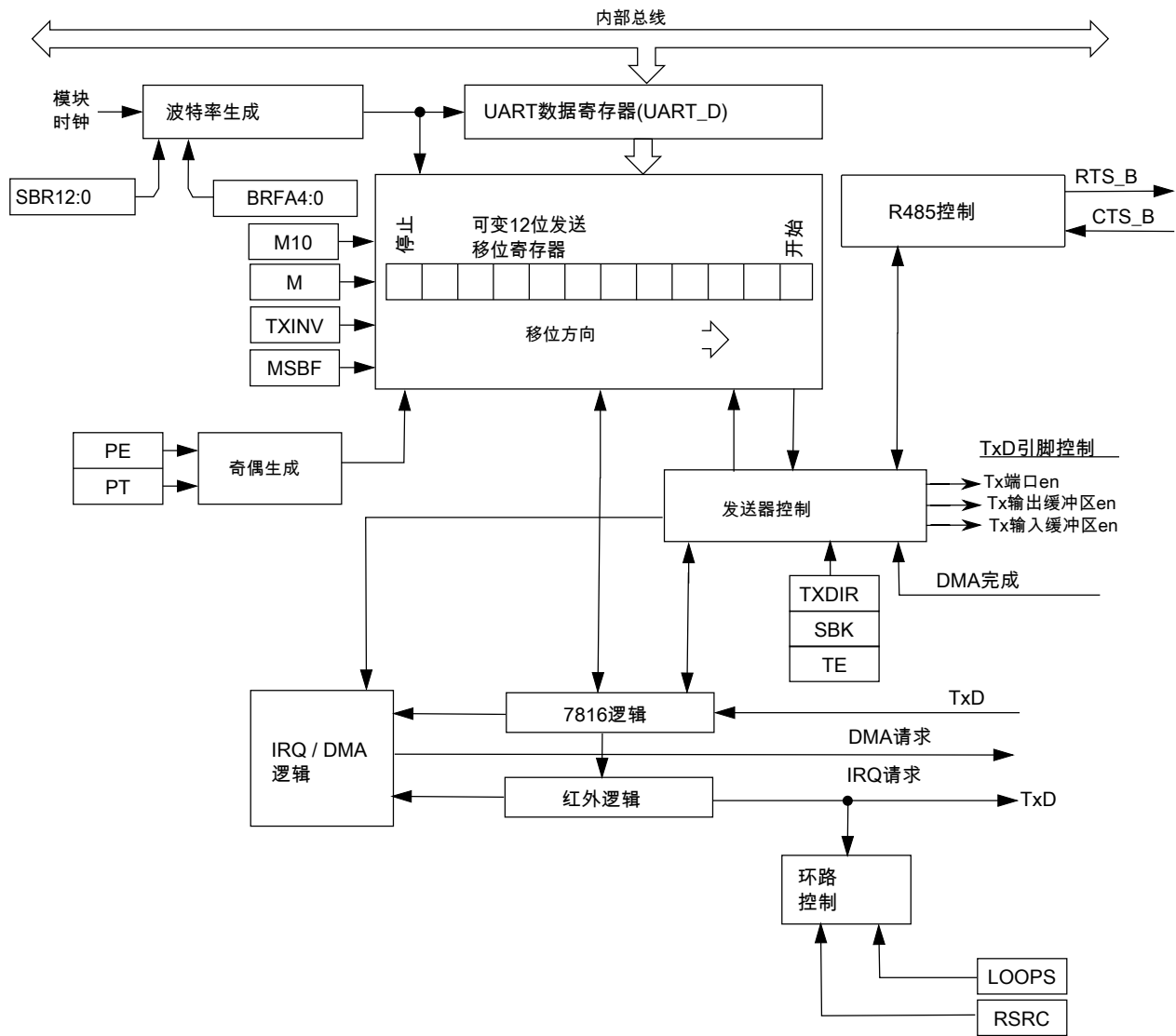


图 45-153. 发送器框图

45.5.1.1 发送器字符长度

UART 发送器可适应 8 位、9 位或 10 位的数据字符。C1[M]和 C1[PE]位以及 C4[M10]位的状态决定数据字符的长度。发送 9 位数据时，位 C3[T8]为第 9 位（位 8）。

45.5.1.2 传输位顺序

置位 S2[MSBF]后，UART 会将数据字 MSB 作为起始位后第一位发送。同样地，数据字 LSB 将在奇偶校验位或停止（如未使能奇偶校验）之前立即发送。所有必需的位排序由模块自动处理。因此，写入 D 以进行传输的数据格式完全与 S2[MSBF]设置无关。

45.5.1.3 字符传输

为了传输数据，MCU 利用 UART 数据寄存器 C3[T8]和 D 将数据位写入 UART 传输缓冲区。然后，传输缓冲区中的数据按照需要传输至发送器移位寄存器。传输移位寄存器以任何所需的开始和停止位开始之后，通过传输数据输出信号移除一个帧。UART 数据寄存器 C3[T8]和 D 提供访问传输缓冲区结构的权限。

此外，UART 还置位标志位，传输数据寄存器清空标志位 S1[TDRE]，生成一个中断或 DMA 请求(C5[TDMAS])（当传输缓冲区为空，且传输缓冲区中的数据字数等于或少于 TWFIFO[TXWATER]）。传输驱动器例程可能在空间允许时，利用 C3[T8]/D 向传输缓冲区写入额外的数据字来响应此标志位。

具体的编程序列参见[应用信息](#)。

置位 C2[TE]会自动加载具有以下报头的传输移位寄存器：

表 45-159. 传输报头长度

BDH[SBNS]	C1[M]	C4[M10]	C1[PE]	已发送位
0	0	—	—	10
1	0	—	—	11
0	1	0	—	11
1	1	0	—	12
0	1	1	1	12
1	1	1	1	13

移出前同步码后，控制逻辑将数据从 D 寄存器传送到发送移位寄存器。发送器在数据字前后自动传输正确的起始位和停止位。在数据字之后传输的停止位位数可利用 BDH[SBNS]字段进行编程。

C7816[ISO_7816E] = 1 时，置位 C2[TE]不会导致生成报头。发送器在相应的保护时间过期时即开始传送。C7816[TTYPE] = 0 时，使用 GT 中的值。C7816[TTYPE] = 1 时，使用 BGT 中的值，因为 C2[TE]在数据块完成传输前都会保持认定。在 C7816[TTYPE] = 1 且数据块传输完成时会自动清除 C2[TE]。C7816[TTYPE] = 0 时，发送器听取 NACK 指示。如果未收到 NACK，则假定收到的字符正确。如果收到 NACK，发送器重新发送数据，则假定该字符的重发次数，也就是说，接收到的 NACK 数目少于或等于 ET7816[TXTHRESHOLD]中的值。

硬件支持奇数或偶数校验。奇偶校验使能后，正好在停止位之前的位将即奇偶校验位。

当发送移位寄存器不发送帧时，传输数据输出信号进入空闲状态，为逻辑 1。任何时间，如果软件清除了 C2[TE]，则发送器使能信号将变成低电平，发送信号将空闲。

如果软件在传输进行时清除了 C2[TE]，传输移位寄存器中的字符继续移出，S1[TC]在数据写序列期间被清除。要清除 S1[TC]，则必须在向 D 寄存器写入之后读取 S1 寄存器。

如果 S1[TC]在字符传输期间被清除，且 C2[TE]被清除，则传输使能信号在当前帧完成时被解除认定。在此之后，传输数据输出信号进入闲置状态，即便 UART 传输数据缓冲区中存在未决数据时也是如此。为了确保写入 FIFO 的所有数据都在清除 C2[TE]之前通过此链接传输，请等待 S1[TC]置位。或者，也可通过将 TWFIFO[TXWATER]置位为 0x0，等待 S1[TDRE]置位来获得相同结果。

45.5.1.4 发送分隔字符

置位 C2[SBK]会将分隔字符载入发送移位寄存器。分隔字符包含所有的逻辑 0，并且无启动、停止或奇偶校验位。分隔字符长度取决于 C1[M]、C1[PE]、S2[BRK13]、BDH[SBNS]和 C4[M10]。参见下表。

表 45-160. 发送分隔字符长度

S2[BRK13]	BDH[SBNS]	C1[M]	C4[M10]	C1[PE]	已发送位
0	0	0	—	—	10
0	1	0	—	—	11
0	0	1	0	—	11
0	1	1	0	—	12
0	0	1	1	1	12
0	1	1	1	1	13
1	0	0	—	—	13
1	0	1	—	—	14
1	1	0	—	—	15
1	1	1	—	—	16

只要置位 C2[SBK]，发送器逻辑就连续向发送移位寄存器载入分隔字符。软件清除 C2[SBK]位后，移位寄存器就完成了最后一个分隔字符的发送，然后至少发送一个逻辑 1。分隔字符末端的自动逻辑 1 确保能够识别下一字符的起始位。当 C7816[ISO_7816E]置位/使能时，不支持分隔位。

注

分隔字符排队时，将在数据值从移位寄存器中移出完成之后发送。这意味着，如果数据排在要发送的数据缓冲区中，则分隔字符将抢占该排队的的数据。分隔字符完成之后才会发送排队的的数据。

45.5.1.5 空闲字符

空闲字符包含所有的逻辑 1，并且无启动、停止或奇偶校验位。空闲字符长度取决于 C1[M]、C1[PE]、BDH[SBNS]和 C4[M10]。报头是一个同步空闲字符，在 C2[TE]置位之后开始发起第一次发送。当 C7816[ISO_7816E]置位/使能时，不发送或检测空闲字符。未传输数据时，数据 I/O 行处于一种非活跃状态。

如果 C2[TE]在发送期间被清除，则正在进行的发送完成后，传输数据输出信号将变为空闲。在当前发送的数据字后，在发送期间清除并随后置位 C2[TE]位会将要发送的空闲字符列入队列。

注

空闲字符排队时，将在数据值从移位寄存器中移出完成之后发送空闲字符。这意味着，如果数据排在要发送的数据缓冲区队列中，则空闲字符将抢占该排队的的数据。空闲字符完成之后才会发送排队的的数据。

如果 C2[TE]被清除，并且发送已完成，则 UART 不再控制 TXD 引脚。

45.5.1.6 硬件流量控制

传送器将通过 CTS 的值来控制是否传输。如果已使能清除-发送操作，则字符会在 CTS 被断言时传输。如果 CTS 在传输中被解除断言，但字符仍留在接收器数据缓冲区中，则将发送移位寄存器中的字符并让 TXD 保留标记状态，直至 CTS 被重新断言。

如果清除-发送操作被禁用，则传送器会忽略 CTS 的状态。此外，如果传送器在发送分隔字符时发送连续低电平，则不管是否已使能清除-发送操作，传送器都会忽略 CTS 状态。

即使同一 UART 接收器的 RTS 信号被禁用，传送器的 CTS 信号也能使能。

45.5.1.7 收发器驱动使能

发送器可将 RTS 用作驱动外部收发器的使能信号。详情参见[利用 RTS 使能收发器驱动器](#)。如果请求发送操作已使能，则将字符存入空发送器数据缓冲器时，RTS 将在起始位发送前断言。RTS 在发送器数据缓冲器具有任何字符的整段时间内保持断言。RTS 将在发送器数据缓冲器和移位寄存器中的所有字符（包括最后停止位）完成发送后一位时间解除断言。发送分隔字符也会断言 RTS，并且与将字符存入发送器数据缓冲器内一样拥有相同的断言和解除断言时序。

仅当发送器使能时才会断言发送器的 RTS 信号。但是，发送器的 RTS 信号不受其 CTS 信号影响。在传输完成之前，RTS 将保持断言状态，即使在数据传输过程中禁用发送器也是如此。

下图显示发送器的功能时序信息。除了实际字符本身，TXD 还会显示起始位。如果需要，也可以使用虚线指示停止位。

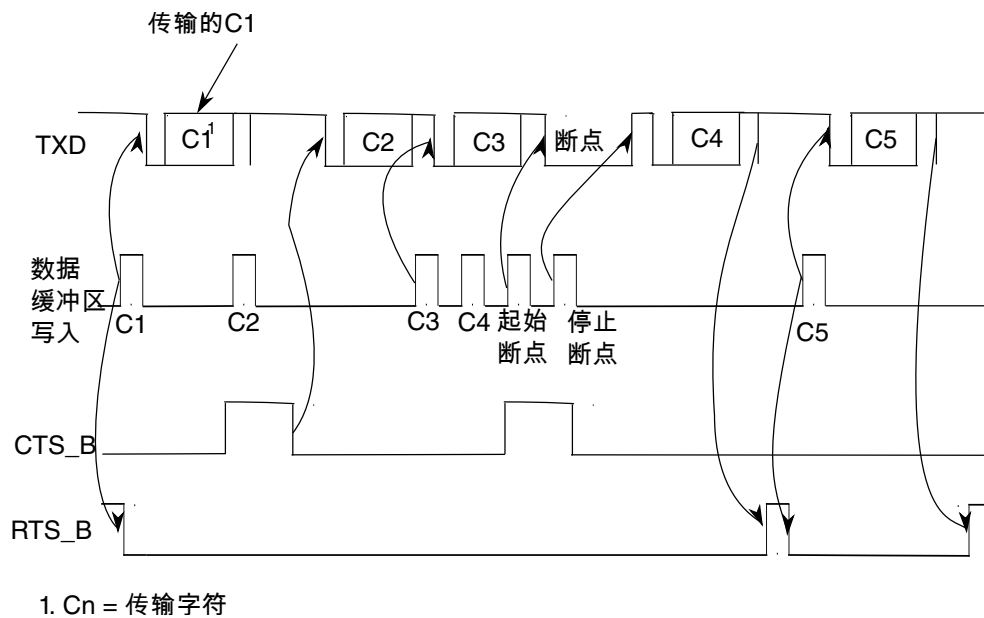


图 45-154. 发送器 RTS 和 CTS 时序图

45.5.2 接收器

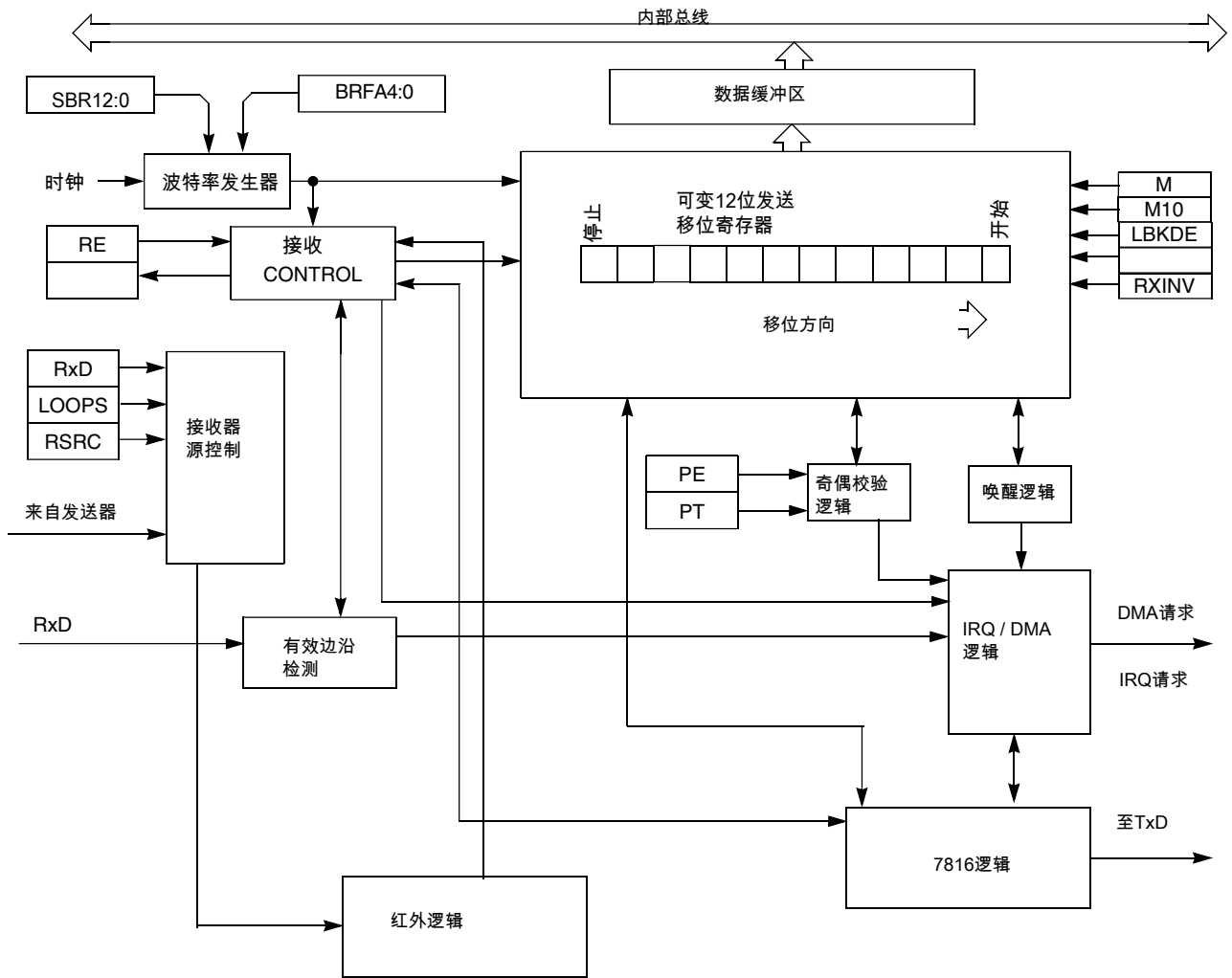


图 45-155. UART 接收器结构框图

45.5.2.1 接收器字符长度

UART 接收器可适应 8 位、9 位或 10 位的数据字符。C1[M]、C1[PE]、BDH[SBNS] 以及 C4[M10] 的状态决定数据字符的长度。接收 9 或 10 位数据时, C3[R8] 为第 9 位 (位 8)。

45.5.2.2 接收器位排序

当 S2[MSBF]置位之后，接收器以这样的方式操作：当开始位成为数据字的 MSB 之后，接收首个位。同样地，将在奇偶校验位或停止位（如未使能奇偶校验）之前接收的位视为数据字的 LSB。所有必需的位排序由模块自动处理。因此，所读取的来自接收数据缓冲区的数据格式完全与 S2[MSBF]无关。

45.5.2.3 字符接收

在 UART 接收过程中，接收移位寄存器通过未同步的接收器输入信号移入帧。在完整帧移至接收移位寄存器后，帧数据部分传输到 UART 接收缓冲器。此外，接收过程中计算得出的噪声和奇偶校验错误标志位也会在 UART 接收缓冲器中捕捉到。可通过 D 和 C3[T8]寄存器访问接收数据缓冲器。可在 ED 寄存器中读取与接收数据字相关的其他接收信息标志位。如果接收缓冲器中得出的数据字数量等于或大于 RWFIFO[RXWATER]所示的数目，则会置位 S1[RDRF]。如果还置位 C2[RIE]，则 RDRF 将生成 RDRF 中断请求。或者，通过编程 C5[RDMAS]也可生成 DMA 请求。

当 C7816[ISO_7816E]置位/使能且 C7816[TTYPE] = 0 时，字符接收操作稍有不同。接收奇偶校验位时，将会检查奇偶校验位的有效性。如果 C7816[ANACK]置位且奇偶校验检查失败，或者 INIT 和接收到的字符不是有效的初始字符，则接收器将会发送 NACK。如果连续接收错误数目超过 ET7816[RXTHRESHOLD]设置的阈值，则当 IE7816[RXTE]置位时，将会置位 IS7816[RXT]并且会生成中断。如果检测到因奇偶校验或无效初始字符导致的错误，则数据将无法从接收移位寄存器传输到接收缓冲器。相反，数据将被下一个传入数据覆盖。

当 C7816[ISO_7816E]置位/使能时，C7816[ONACK]置位/使能，且接收到的字符将导致接收缓冲器溢出，接收器发出 NACK。此外，S1[OR]将置位并且会发出中断（如需要），移位寄存器中的数据将被丢弃。

45.5.2.4 数据采样

接收器以 RT 时钟速率对未同步的接收器输入信号采样。RT 时钟是频率为 16 倍波特率的内部信号。要针对波特率不匹配情况进行调整，需重新同步 RT 时钟（参见下图）：

- 在每个起始位后。
- 在接收器检测到从逻辑 1 到逻辑 0 的数据位更改后（大多数数据位样本 RT8、RT9 和 RT10 返回有效逻辑 1，大多数下一个 RT8、RT9 和 RT10 样本返回有效逻辑 0）。

为找到起始位，数据恢复逻辑执行对逻辑 0 (其前面是三个逻辑 1) 的异步搜索。当可能的起始位出现下降沿时，RT 时钟开始计数到 16。

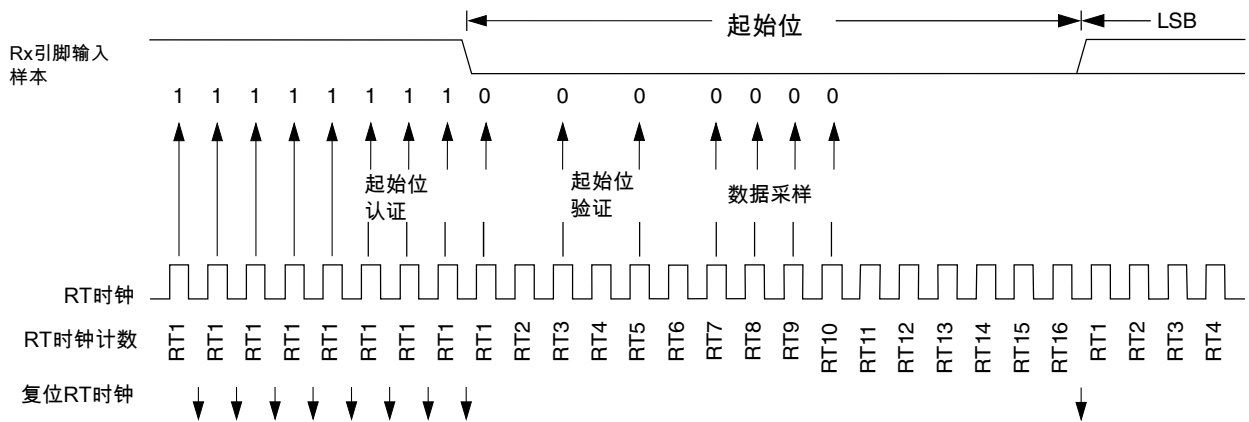


图 45-156. 接收器数据采样

要验证起始位和检测噪声，数据恢复逻辑对 RT3、RT5 和 R7 (C7816[ISO_7816E] 清零/禁用时) 和 RT8、RT9 和 RT10 (C7816[ISO_7816E] 置位/使能时) 进行采样。下表总结了起始位验证取样的结果。

表 45-161. 起始位验证

RT3、RT5 和 RT7 样本 RT8、RT9、RT10 样本 (为 7816E 时)	起始位验证	噪声标志位
000	有	0
001	有	1
010	有	1
011	无	0
100	有	1
101	无	0
110	无	0
111	无	0

如果起始位验证未成功，则 RT 时钟将复位并将开始新的起始位搜索。

要确定数据位的值和检测噪声，恢复逻辑对 RT8、RT9 和 RT10 进行采样。下表总结了数据位取样的结果。

表 45-162. 数据位恢复

RT8、RT9 和 RT10 样本	数据位确定	噪声标志位
000	0	0
001	0	1
010	0	1

下一页继续介绍此表...

表 45-162. 数据位恢复 (继续)

RT8、RT9 和 RT10 样本	数据位确定	噪声标志位
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

注

RT8、RT9 和 RT10 样本不影响起始位验证。如果在起始位成功验证后，任何或所有 RT8、RT9 和 RT10 起始位样本为逻辑 1，噪声标志位(S1[NF])已置位，接收器就假定该位为起始位 (逻辑 0)。除了当 C7816[ISO_7816E]置位/使能时之外，其中 RT8、RT9 和 RT10 的值专门确定是否存在起始位。

为验证停止位和检测噪声，恢复逻辑对 RT8、RT9 和 RT10 进行采样。下表总结了停止位取样的结果。如果 C7816[ISO_7816E]置位/使能且 C7816[TTYTYPE] = 0，则不会发生停止位验证。相反，从 RT8 开始，接收器将按照设置传输 NACK，直到以下时间周期的 RT9。当 C7816[ISO_7816E]置位/使能时，不支持帧错误检测。

表 45-163. 停止位恢复

RT8、RT9 和 RT10 样本	帧错误标志位	噪声标志位
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0

在下图中，验证样本 RT3 和 RT5 确定检测出的第一个低电平为噪声而不是起始位的开头。在本示例中，C7816[ISO_7816E] = 0。RT 时钟复位并重新开始起始位搜索。噪声标志未置位，因为噪声产生于找到起始位之前。

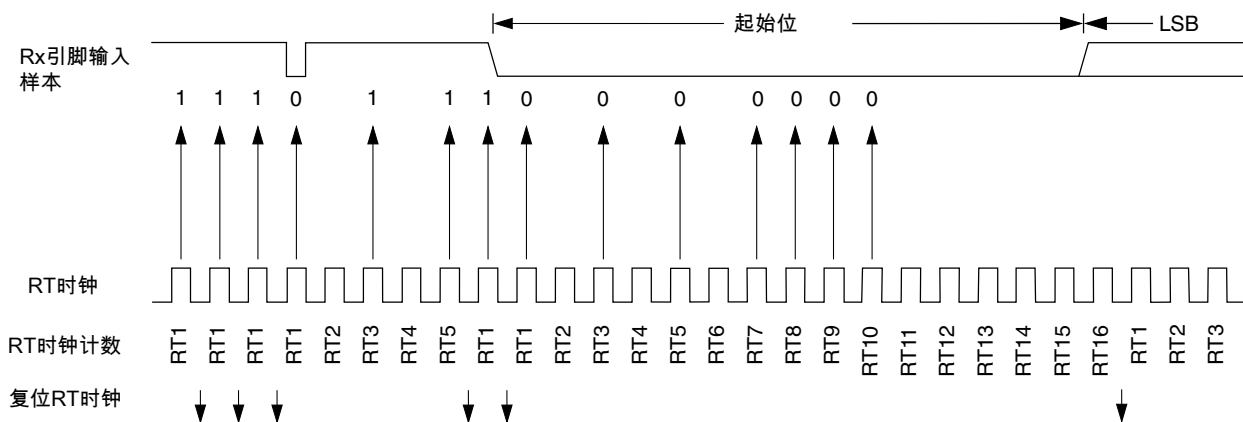


图 45-157. 起始位搜索示例 1 (C7816[ISO_7816E] = 0)

在下图中，验证样本 RT3 为高电平。在本示例中，C7816[ISO_7816E] = 0。RT3 样本置位噪声标志位。尽管感知位时间有偏差，但数据样本 RT8、RT9 和 RT10 在位时间内并且数据恢复是成功的。

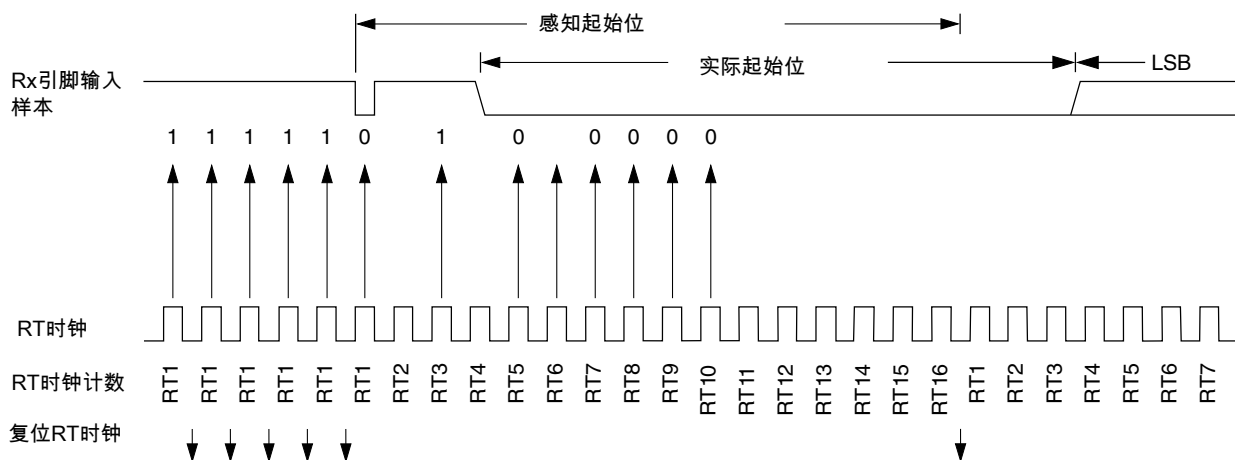


图 45-158. 起始位搜索示例 2 (C7816[ISO_7816E] = 0)

在下图中，大量突发噪声被认为是起始位的开始，尽管测试样本 RT5 为高电平。在本示例中，C7816[ISO_7816E] = 0。RT5 样本置位噪声标志位。尽管这是感知位时间偏差的最坏情况，但数据样本 RT8、RT9 和 RT10 在位时间内并且数据恢复是成功的。

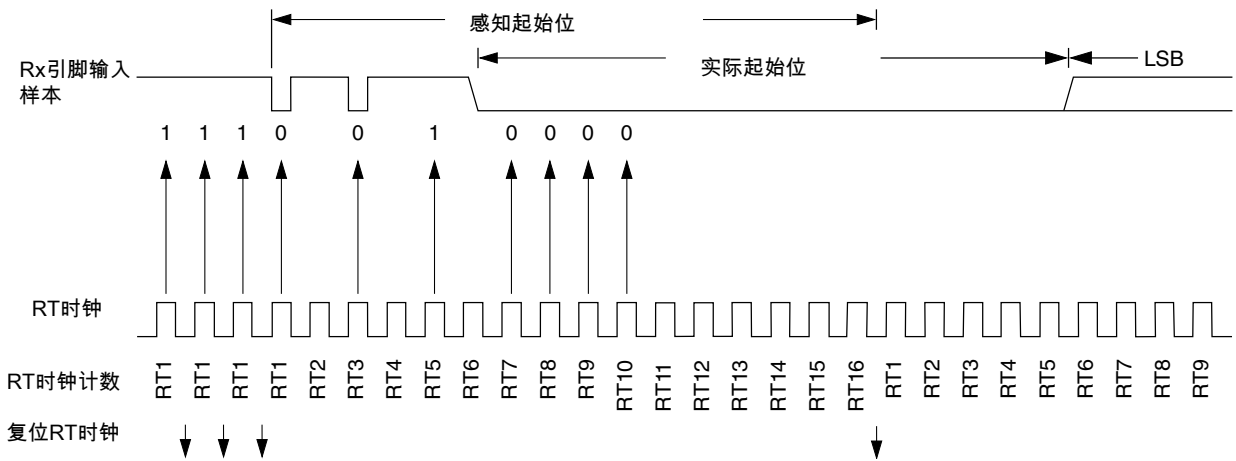


图 45-159. 起始位搜索示例 3 (C7816[ISO_7816E] = 0)

下图显示起始位时间早期噪声的影响。在本示例中，C7816[ISO_7816E] = 0。尽管此噪声不影响与起始位时间的正确同步，但其置位了噪声标志。

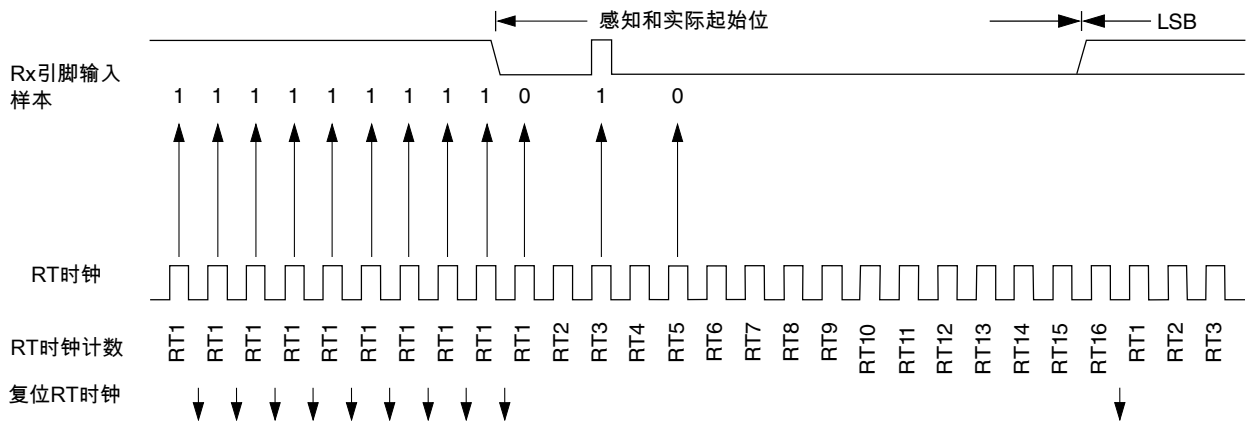


图 45-160. 起始位搜索示例 4 (C7816[ISO_7816E] = 0)

下图显示起始位开头附近使 RT 时钟复位的突发噪声。在本示例中，C7816[ISO_7816E] = 0。复位后此样本为低电平，但其前面不是三个被认证为下降沿的高电平样本。根据起始位搜索和数据的时序，此帧可能会完全丢失或设置成帧错误标志位。

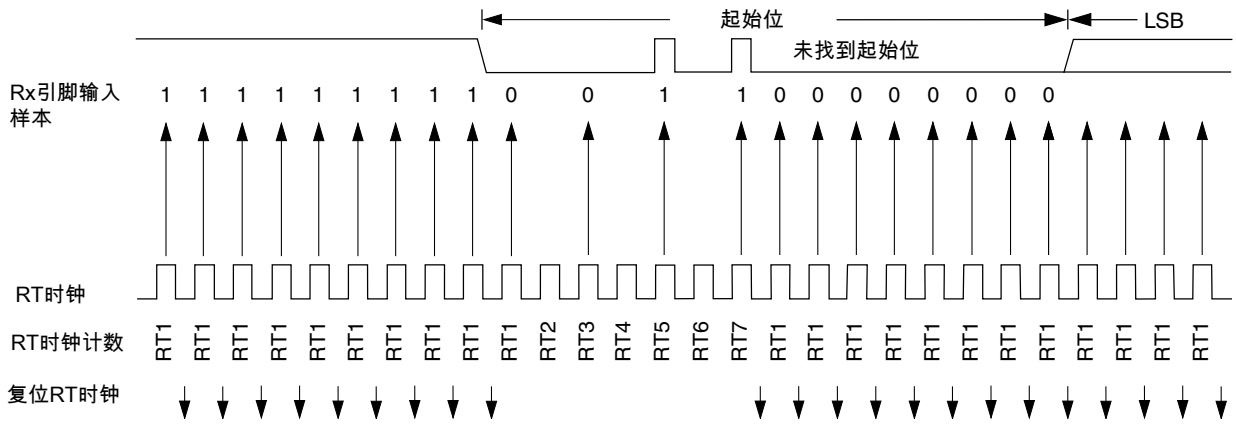


图 45-161. 起始位搜索示例 5 (C7816[ISO_7816E] = 0)

在下图中，突发噪声使大多数数据样本 RT8、RT9 和 RT10 变为高电平。在本示例中，C7816[ISO_7816E] = 0。这将置位噪声标志但不会使 RT 时钟复位。仅在起始位中，RT8、RT9 和 RT10 数据样本被忽略。在本示例中，如果 C7816[ISO_7816E] = 1，则完全检测不到起始位，因为三个样本 (RT8、RT9、RT10) 至少有两个为高电平。

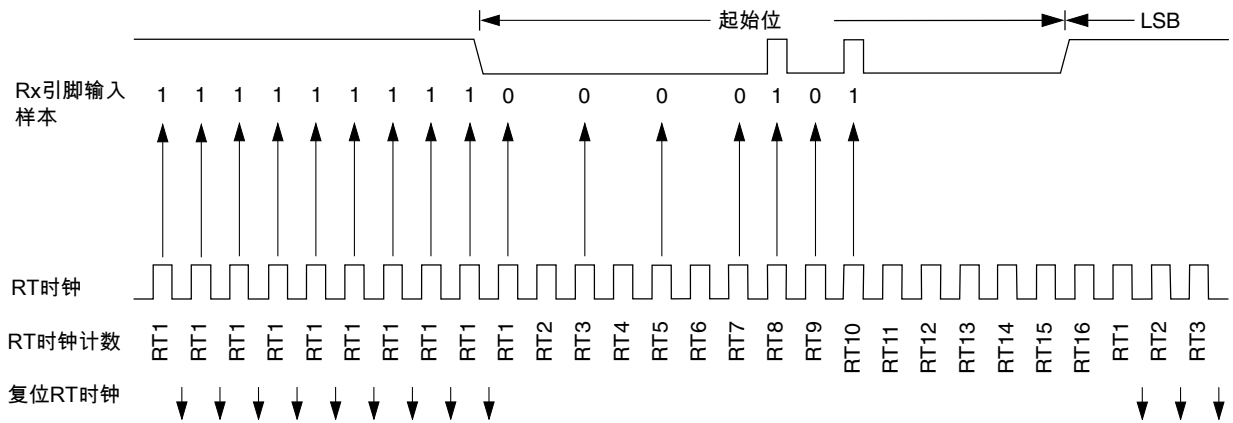


图 45-162. 起始位搜索示例 6

45.5.2.5 帧错误

如果数据恢复逻辑未检测到逻辑 1 且停止位在传入的帧中，则当 S2[LBKDE]禁用时，其将置位帧错误标志位 S1[FE]。当 S2[LBKDE]禁用时，分隔字符也会置位 S1[FE]，因为分隔字符无停止位。当接收到的数据置于接收数据缓冲器中时将置位 S1[FE]。当 C7816[ISO7816E]置位/使能时，不支持帧错误。但是，如果置位 S1[FE]，当 C7816[ISO7816E]置位时不会接收到数据。

45.5.2.6 接收分隔字符

当开始位后接八个、九个或十个逻辑 0 数据位且停止位为一个逻辑 0 时，UART 识别一个分隔字符。接收分隔字符对 UART 寄存器的影响：

- 置位帧错误标志 S1[FE]。
- 向数据缓冲区写入一个全是 0 的数据字，这可能导致 S1[RDRF]置位，具体取决于缓冲区的水印和数值数。
- 可能置位溢出标志 S1[OR]、噪声标志 S1[NF]、奇偶校验错误标记 S1[PE]或接收器激活标记 S2[RAF]。

通过置位 S2[LBKDE]在 LIN 系统中使用内部振荡器时，可调节分隔字符的检测阈值。UART 分隔字符检测阈值取决于 C1[M]、C1[PE]、S2[LBKDE] 和 C4[M10]。参见下表。

表 45-164. 接收分隔字符检测阈值

LBKDE	SBNS	M	M10	PE	阈值 (位)
0	0	0	—	—	10
0	1	0	—	—	11
0	0	1	0	—	11
0	1	1	0	—	12
0	0	1	1	1	12
0	1	1	1	1	13
1	—	0	—	—	11
1	—	1	—	—	12

S2[LBKDE]置位之后，会对 UART 寄存器带来以下影响：

- 阻止 S1[RDRF]、S1[FE]、S1[NF]以及 S1[PF]被置位。但是，如果它们已经置位，则会保持置位。
- 如果接收到 LIN 分隔字符，则置位 LIN 分隔检测中断标志 S2[LBKDIF]。

当 C7816[ISO_7816E]置位/使能时，不检测或支持支持分隔字符。

45.5.2.7 硬件流量控制

为支持硬件流量控制，可以将接收器编程为自动解除断言和断言 RTS。

- 在传输完成之前，RTS 将保持断言状态，即使在数据传输过程中禁用发送器也是如此。更多详情，请参见[利用 RTS 使能收发器驱动器](#)。

- 如果接收器请求发送功能已使能，则接收器将自动解除断言 RTS（如果接收器数据寄存器中的字符数等于或大于接收器数据缓冲区水印 RWFIFO[RXWATER]）。
- 如果接收器数据寄存器中的字符数小于该水印，则接收器将断言 RTS。如果已断言 RDRF，则不会有任何影响。
- 即使已解除断言 RTS，接收器也会继续接收字符，直到接收器数据缓冲区已满或溢出为止。
- 如果接收器请求发送功能已禁用，则接收器 RTS 将保持解除断言状态。

下图显示接收器硬件流量控制功能时序。除了实际字符本身，RXD 还会显示起始位。如果需要，也可以使用虚线指示停止位。水印置位为 2。

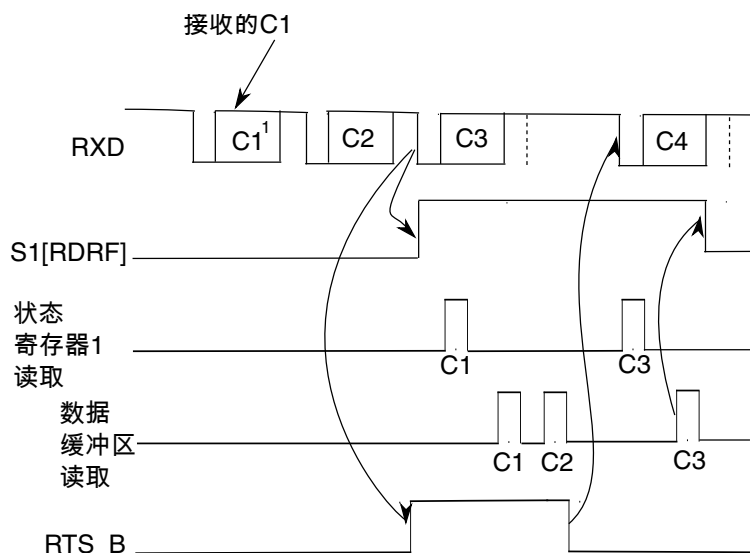


图 45-163. 接收器硬件流量控制时序图

45.5.2.8 红外解码器

红外解码器将接收到的字符从 IrDA 格式转换为接收器使用的 NRZ 格式。解码器还有一个 16-RT 时钟计数器，可以滤除噪声并在接收到 1 时发出指示。

45.5.2.8.1 起始位检测

如果已清除 S2[RXINV]，则已接收字符的第一个上升沿与起始位对应。红外解码器将复位其计数器。此时，接收器还会开始其起始位检测进程。在检测到起始位后，接收器会将其位时间与此起始位时间同步。对于剩余字符接收，红外解码器的计数器与接收器的位时间计数器将相互独立地进行计数。

45.5.2.8.2 噪声过滤

在红外解码器计数器前半段检测到的任何其他上升沿将被解码器忽略。解码器可能无法检测到小于一个 RT 时钟的任何脉冲，无论是在计数前半段还是后半段发现的，均是如此。

45.5.2.8.3 低位检测

在解码器计数的后半段，上升沿将被解码为 0，并发送至接收器。同时解码器计数器复位。

45.5.2.8.4 高位检测

在前一个上升沿之后的 16-RT 时钟，如果未发现上升沿，则解码器会向接收器发送 1。

如果下一个位是 0，其到达时间较晚，则会根据 [低位检测](#) 检测到低位。发送到接收器的值从 1 变为 0。随后，如果在接收器位时间采样周期外出现噪声脉冲，则延迟 0 将不会记录为噪声。

45.5.2.9 波特率公差

发送器件可在低于或高于接收器波特率的波特率下运行。累积的位时间偏差可导致三个停止位数据样本 (RT8、RT9 和 RT10) 之一超出实际停止位之外。如果 RT8、RT9 和 RT10 样本并非都是相同的逻辑值，则将产生噪声误差。如果接收器时钟发生偏差，即大部分 RT8、RT9 和 RT10 停止位样本值为逻辑 0，则将产生成帧错误。

当接收器对传入的帧进行采样时，它在此帧的任何有效下降沿上重新与 RT 时钟进行同步。在帧中进行重新同步将纠正发送位时间与接收位时间之间的偏差。

45.5.2.9.1 慢速数据公差

下表显示在不导致噪声错误或帧错误的情况下，慢速接收的帧可容许的偏差。慢速停止位从 RT8 而非 RT1 开始，但可及时抵达停止位数据样本 RT8、RT9 和 RT10。

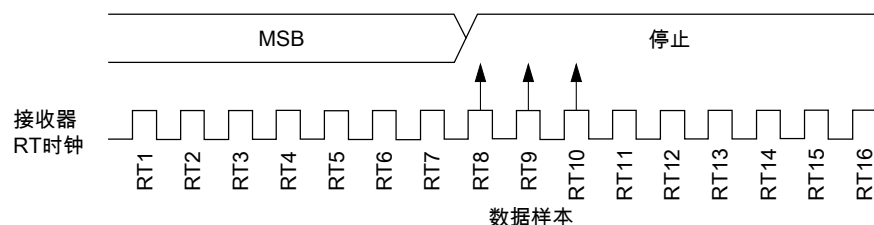


图 45-164. 慢速数据

对于 8 位数据字符，停止位的数据采样将使用接收器 154 RT 周期 (9 位时间 × 16 RT 周期 + 10 RT 周期)。

对于图 45-164 所示的未对齐字符，当发送器计数为 147 RT 周期 (9 位时间 × 16 RT 周期 + 3 RT 周期) 时，接收器计数到 154 RT 周期。

在无误差的情况下，慢速 8 位数据字符的接收器计数与发送器计数之间的最大百分比差异为：

$$((154 - 147) \div 154) \times 100 = 4.54\%$$

对于 9 位数据字符，停止位的数据采样将使用接收器 170 RT 周期 (10 位时间 × 16 RT 周期 + 10 RT 周期)。

对于图 45-164 所示的未对齐字符，当发送器计数为 163 RT 周期 (10 位时间 × 16 RT 周期 + 3 RT 周期) 时，接收器计数到 170 RT 周期。

在无误差的情况下，慢速 9 位数据字符的接收器计数与发送器计数之间的最大百分比差异为：

$$((170 - 163) \div 170) \times 100 = 4.12\%$$

45.5.2.9.2 快速数据容差

下图显示了快速接收帧允许的最大偏差。快速停止位在 RT10 而非 RT16 处停止，但仍在 RT8、RT9 和 RT10 处采样。

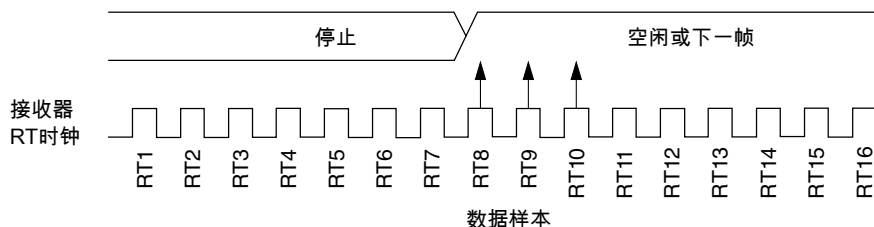


图 45-165. 快速数据

对于 8 位数据字符，停止位的数据采样将使用接收器 154 RT 周期 (9 位时间 × 16 RT 周期 + 10 RT 周期)。

对于图 45-165 所示的未对齐字符，当发送器件计数为 160 RT 周期 (10 位时间 × 16 RT 周期) 时，接收器计数到 154 RT 周期。

在无误差的情况下，快速 8 位数据字符的接收器计数与发送器计数之间的最大百分比差异为：

$$((154 - 160) \div 154) \times 100 = 3.90\%$$

对于 9 位数据字符，停止位的数据采样将使用接收器 170 RT 周期 (10 位时间 × 16 RT 周期 + 10 RT 周期)。

对于图 45-165 所示的未对齐字符，当发送器件计数为 176 RT 周期 (11 位时间 × 16 RT 周期) 时，接收器计数到 170 RT 周期。

在无误差的情况下，快速 9 位数据字符的接收器计数与发送器计数之间的最大百分比差异为：

$$((170 - 176) \div 170) \times 100 = 3.53\%$$

45.5.2.10 接收器唤醒

C1[WAKE]决定 UART 如何退出待机状态，以处理传入的消息。C1[WAKE]能使空闲线路唤醒或地址标志唤醒。

因为不允许多接收器系统，所以当 C7816[ISO_7816E]置位/使能时，不支持接收器唤醒。

45.5.2.10.1 空闲输入线路唤醒(C1[WAKE] = 0)

在此唤醒方法中，未同步接收器输入信号处于空闲状态将会清除 C2[RWU]并唤醒 UART。初始帧或每个消息的帧包含寻址信息。所有接收器都会评估寻址信息，进行过消息编址的接收器处理接下来的帧。任何未进行过消息编址的接收器可置位其 C2[RWU]并返回待机状态。C2[RWU]保持置位，接收器也保持待机状态，直到另一个空闲字符出现在未同步接收器输入信号上。

空闲线路唤醒需要消息至少由一个空闲字符隔开并且无消息包含空闲字符。

当 C2[RWU]为 1 且 S2[RWUID]为 0 时，唤醒接收器的空闲字符不会置位 S1[IDLE]，或接收数据寄存器已满标志 S1[RDRF]。接收器将唤醒并等待存储在接收数据缓冲区中下一条消息的第一个数据字符。当 S2[RWUID]和 C2[RWU]置位且 C1[WAKE]清零时，任何空闲情况均会置位 S1[IDLE]并生成中断 (如已使能)。

当 C7816[ISO_7816E]置位/使能时，不支持空闲输入线路唤醒。

45.5.2.10.2 地址标志唤醒(C1[WAKE] = 1)

在这种唤醒方法中，帧停止位之前的位中的逻辑 1 会清除 C2[RWU]并唤醒 UART。停止位之前的位中的逻辑 1 将帧标记为包含寻址信息的地址帧。所有接收器都会评估寻址信息，进行过消息编址的接收器处理接下来的帧。任何未进行过消息编址的接收器可置位其 C2[RWU]并返回待机状态。C2[RWU]保持置位，接收器也保持待机状态，直到另一个地址帧出现在未同步接收器输入信号上。

停止位之前的位中的逻辑 1 在接收到停止位后清除接收器的 C2[RWU], 并将接收到的数据放入接收器数据缓冲区中。注意, 如果匹配地址操作已使能 (例如, C4[MAEN1] 或 C4[MAEN2] 已置位), 则只有在比较结果匹配时接收到的帧才会被传输至接收缓冲区中。

地址标志唤醒允许消息包含空闲字符, 但需要保留停止位之前的位, 以用于地址帧中。

如果模块处于待机模式且没有任何项会触发唤醒 UART, 则没有错误标志被置位, 即使接收的数据行中检测到无效的错误的标志时也是如此。

当 C7816[ISO_7816E]置位/使能时, 不支持地址标志唤醒。

45.5.2.10.3 匹配地址操作

当 C4[MAEN1] 或 C4[MAEN2]置位时, 匹配地址操作使能。在该功能中, RX 引脚收到一个帧, 其地址标志位为逻辑 1, 则认为该帧是一个地址, 并与关联的 MA1 或 MA2 寄存器比较。只有在比较结果相匹配时, 该帧才会传输到接收缓冲器, 并且 S1[RDRF]置位。如果地址比较结果匹配, 则接收到的所有后续的地址标志位为逻辑 0 的帧均被认为是与该地址相关联的数据, 并传输至接收数据缓冲器中。如果标志地址匹配不成功, 则不会传输至接收数据缓冲器, 且所有后接帧 (地址标志位为逻辑 0) 都会被丢弃。如果 C4[MAEN1]和 C4[MAEN2]均为 0, 则接收器正常工作, 接收到的所有数据都会传输至接收数据缓冲器。

MA1 和 MA2 寄存器具有相同的匹配地址操作方式。注意, 当奇偶校验位针对 8 位和 9 位格式使能时, 地址标志的位置和奇偶校验位相同。

- 如果 C4[MAEN1]和 C4[MAEN2]二者仅断言一个, 则只会将标记的地址与关联的匹配寄存器进行比较, 并且在匹配成功时才会将数据传输至接收数据缓冲器。
- 如果 C4[MAEN1]和 C4[MAEN2]同时断言, 则会将标记的地址与两个匹配寄存器进行比较, 并且只会与任一寄存器相匹配时传输数据。

当 C7816[ISO_7816E]置位/使能时, 不支持地址匹配操作。

45.5.3 波特率生成

波特率发生器中的 13 位系数计数器和 5 位微调计数器派生接收器和发送器的波特率。写入 SBR[12:0]的值范围为 1 至 8191, 它决定模块时钟分频因子。SBR 位位于 UART 波特率寄存器 BDH 和 BDL 中。波特率时钟与模块时钟同步并驱动接收器。微调计数器为波特率时钟增加了微延时, 使得波特率的微调值与系统的波特率匹配。发送器由波特率时钟进行 16 分频后驱动。接收器的采样率为每位时间 16 个样本。

波特率生成受两个误差来源的影响:

- 模块时钟的整数分频可能不能提供准确的目标频率。可通过微调计数器来减少这个误差。
- 与模块时钟同步可能导致相移。

表 45-165 列出了可用的波特分频因子微调值。

$$\text{UART 波特率} = \text{UART 模块时钟} / (16 \times (\text{SBR}[12:0] + \text{BRFD}))$$

下表列出了一些达到目标波特率的示例，模块时钟频率为 10.2 MHz，进行或不进行微调。

表 45-165. 波特率 (示例: 模块时钟 = 10.2 MHz)

位 SBR (小数)	位 BRFA	BRFD 值	接收器 时钟 (Hz)	发送器 时钟 (Hz)	目标波特率	错误 (%)
17	00000	0	600,000.0	37,500.0	38,400	2.3
16	10011	19/32=0.59375	614,689.3	38,418.08	38,400	0.047
33	00000	0	309,090.9	19,318.2	19,200	0.62
33	00110	6/32=0.1875	307,344.6	19,209.04	19,200	0.047
66	00000	0	154,545.5	9659.1	9600	0.62
133	00000	0	76,691.7	4793.2	4800	0.14
266	00000	0	38,345.9	2396.6	2400	0.14
531	00000	0	19,209.0	1200.6	1200	0.11
1062	00000	0	9604.5	600.3	600	0.05
2125	00000	0	4800.0	300.0	300	0.00
4250	00000	0	2400.0	150.0	150	0.00
5795	00000	0	1760.1	110.0	110	0.00

表 45-166. 波特率微调

BRFA	波特率分频因子 (BRFD)
0 0 0 0 0	0/32 = 0
0 0 0 0 1	1/32 = 0.03125
0 0 0 1 0	2/32 = 0.0625
0 0 0 1 1	3/32 = 0.09375
0 0 1 0 0	4/32 = 0.125
0 0 1 0 1	5/32 = 0.15625
0 0 1 1 0	6/32 = 0.1875
0 0 1 1 1	7/32 = 0.21875
0 1 0 0 0	8/32 = 0.25
0 1 0 0 1	9/32 = 0.28125
0 1 0 1 0	10/32 = 0.3125

下一页继续介绍此表...

表 45-166. 波特率微调 (继续)

BRFA	波特率分频因子 (BRFD)
01011	11/32 = 0.34375
01100	12/32 = 0.375
01101	13/32 = 0.40625
01110	14/32 = 0.4375
01111	15/32 = 0.46875
10000	16/32 = 0.5
10001	17/32 = 0.53125
10010	18/32 = 0.5625
10011	19/32 = 0.59375
10100	20/32 = 0.625
10101	21/32 = 0.65625
10110	22/32 = 0.6875
10111	23/32 = 0.71875
11000	24/32 = 0.75
11001	25/32 = 0.78125
11010	26/32 = 0.8125
11011	27/32 = 0.84375
11100	28/32 = 0.875
11101	29/32 = 0.90625
11110	30/32 = 0.9375
11111	31/32 = 0.96875

45.5.4 数据格式 (非 ISO-7816)

每个数据字符都包含在一个帧中，帧包含一个起始位和一个停止位。其余数据格式取决于 C1[M]、C1[PE]、S2[MSBF]、BDH[SBNS]和 C4[M10]。

45.5.4.1 8 位配置

清除 C1[M]会针对 8 位数据字符配置 UART，也就是说，会在 D 中寄存器映射 8 位。包含 8 个数据位的帧总共有 10 位（如果 BDH[SBNS] = 1，则为 11 位）。8 个数据位中的最高有效位可用作地址标志，用来唤醒接收器。如果最高有效位如此使用，那么它会作为地址或数据指示，而剩下的 7 位则作为实际数据。C1[PE]置位之后，8 个数据位自动算做无校验位。参见下表。

表 45-167. 8 位数据格式的配置

UART_C1[PE]	开始位	数据位	地址位	奇偶校验位	STOP 位
0	1	8	0	0	1
0	1	7	1	0	1
1	1	7	0	1	1

注

在上述表格的最后一列，停止位的编号在 BDH[SBNS]置位之后会变成 2。

45.5.4.2 9 位配置

当 C1[M]置位且清除 C4[M10] 以及 BDH[SBNS]时，UART 配置用于 9 位数据字符。如果 C1[PE]使能，则第 9 位是 C3[T8/R8]，或者内部生成的无校验位。这样会生成总共包含 11 位的帧。如果第 9 个数据位被选定为 C3[T8]，则它在传输后将保持不变，可重复使用且无需重写，除非需要更改该值。当第 9 个数据位用作地址标记时，此功能可能非常有用。

当 C1[M]和 C4[M10]置位，且清除 BDH[SBNS]时，UART 配置用于 9 位数据字符，但帧由总共 12 位构成。12 位包含开始位和停止位、9 个数据字符位以及第 10 个内部数据位。注意，如果 C4[M10]已置位，则 C1[PE]也必须置位。在这种情况下，第 10 位为内部生成的无校验位。第 9 位可用作地址标记或第 9 个数据位。

参见下表。

表 45-168. 9 位数据格式的配置

C1[PE]	UC1[M]	C1[M10]	开始位	数据位	地址位	奇偶校验位	STOP 位
0	0	0	请参见 8 位配置				
0	0	1	无效配置				
0	1	0	1	9	0	0	1
0	1	0	1	8	1	0	1
0	1	1	无效配置				
1	0	0	请参见 8 位配置				
1	0	1	无效配置				
1	1	0	1	8	0	1	1
1	1	1	1	9	0	1	1
1	1	1	1	8	1	1	1

注

在上述表格的最后一列，停止位的编号在 BDH[SBNS]置位之后会变成 2。

注

除非是在 9 位模式下且 M10 已置位，否则不要在没有校验使能时使用地址标记唤醒。

45.5.4.3 时序示例

采用 NRZ 标记/空间数据格式的配置时序示例如下图所示。时序示例显示了以下子章节中的所有配置以及 LSB 和 MSB 第一变化。本节介绍选择单个停止位模式时可用的数据格式。

45.5.4.3.1 8 位格式，奇偶校验禁用

最高有效位可用于地址标记唤醒。

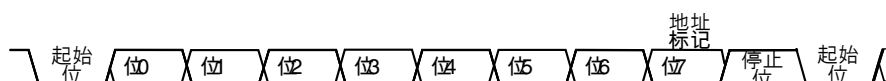


图 45-166. 8 位数据，LSB 开头

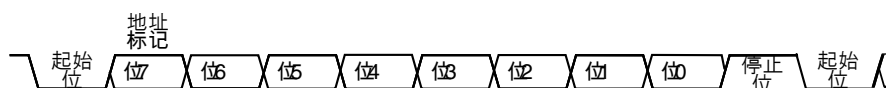


图 45-167. 8 位数据，MSB 开头

45.5.4.3.2 8 位格式，奇偶校验使能



图 45-168. 7 位数据，带 LSB 开头和奇偶校验

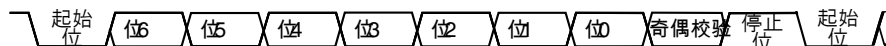


图 45-169. 7 位数据，带 MSB 开头和奇偶校验

45.5.4.3.3 9 位格式，奇偶校验禁用

最高有效位可用于地址标记唤醒。

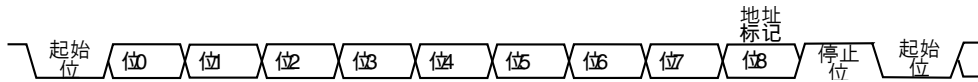


图 45-170. 9 位数据, LSB 优先

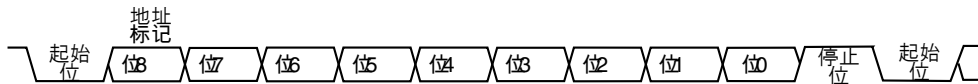


图 45-171. 9 位数据, MSB 优先

45.5.4.3.4 9 位格式, 奇偶校验使能



图 45-172. 8 位数据, LSB 优先和奇偶校验

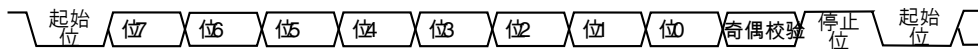


图 45-173. 8 位数据, MSB 优先和奇偶校验

45.5.4.3.5 非存储器映射的第十个奇偶校验位

存储器映射最高有效位可用于地址标记唤醒。

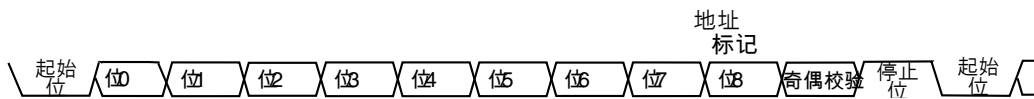


图 45-174. 9 位数据, LSB 优先和奇偶校验

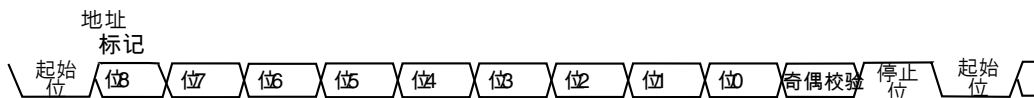


图 45-175. 9 位数据, MSB 优先和奇偶校验

45.5.5 单线操作

通常, UART 使用两个引脚进行发送和接收。在单线操作中, RXD 引脚与 UART 断开连接, 并且 UART 使用半双工串行连接。UART 使用 TXD 引脚来接收和发送。

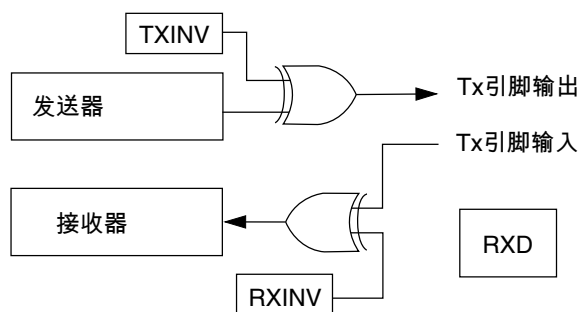


图 45-176. 单线操作 (C1[LOOPS] = 1, C1[RSRC] = 1)

通过置位 C1[LOOPS]和接收器源字段 C1[RSRC]使能单线操作。置位 C1[LOOPS]将禁用从未同步接收器输入信号至接收器的路径。置位 C1[RSRC]可将接收器输入连接至 TXD 引脚驱动器输出。必须使能发送器和接收器 (C2[TE] = 1 和 C2[RE] = 1)。当 C7816[ISO_7816EN]置位时，无需置位 C2[TE]和 C2[RE]。

45.5.6 循环操作

在循环操作中，发送器输出送入接收器输入。未同步接收器输入信号与 UART 断开。

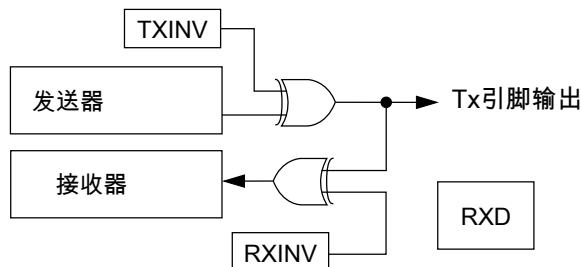


图 45-177. 循环操作 (C1[LOOPS] = 1, C1[RSRC] = 0)

通过置位 C1[LOOPS]和清除 C1[RSRC]使能循环操作。置位 C1[LOOPS]将禁用从未同步接收器输入信号至接收器的路径。清除 C1[RSRC]位，将发送器输出连接到接收器输入。必须使能发送器和接收器 (C2[TE] = 1 和 C2[RE] = 1)。当 C7816[ISO_7816EN]置位时，无需置位 C2[TE]和 C2[RE]。

45.5.7 ISO-7816/智能卡支持

UART 提供支持常用于连接智能卡的 ISO-7816 协议机制。ISO-7816 协议是一个 NRZ、单线、半双工接口。TxD 引脚用于开漏模式，因为数据信号用于发送和接收。ISO-7816 标准内存在多个子协议。UART 支持 T = 0 和 T = 1 协议。该模块还提供自动初始字符检测和配置功能，从而支持正向约定和反向约定数据格式。除了

用于辅助软件的一般中断之外，还提供 7816 专用的多个中断。此外，该模块还提供自动 NACK 响应，并且已设置对故障数据包的自动重新传输。另外还支持各类可编程超时和保护带时间。

术语基本时间单位(ETU)常用于 ISO-7816 中。此概念用于指系统(UART)运行的频率以及发送和接收数据的频率。一个 ETU 代表发送或接收单个位所花的时间。例如，一个标准的 7816 数据包（不包括任何保护时间或 NACK 元素）为 10 个 ETU（起始位、8 个数据位和一个奇偶校验位）。保护时间和等待时间同样也是以 ETU 衡量。

注

ISO-7816 规范还包含某些保留的配置选项。为了维持最大灵活性以支持后续的 7816 增强版或不完全符合规范的器件，UART 允许使用这些选项。此外，UART 还提供超过 7816 规范明显允许的选项灵活性的配置选项。未能正确配置 UART 可能会导致出现非预期行为或与 ISO-7816 规范不兼容。

45.5.7.1 初始字符

在 ISO-7816 (T = 0 模式) 中，可将 UART 配置为使用 C7816[INIT]检测下一个有效初始字符，ISO-7816 称之为 TS 字符。当检测到初始字符时，如果 IE7816[INITDE]已置位，则 UART 将产生一个中断。此外，UART 将根据初始字符自动调整 S2[MSBF]、C3[TXINV]和 S2[RXINV]。对应的初始字符以及得出的寄存器设置如下表所列。

表 45-169. 初始字符自动设置

初始字符 (位 1-10)	初始字符 (16 进制)	MSBF	TXINV	RXINV
LHHL LLL LLH 反向约定	3F	1	1	1
LHHL HHH LLH 正向约定	3B	0	0	0

C7816[INIT]置位之前，必须将 S2[MSBF]、C3[TXINV]和 S2[RXINV]复位为默认值。置位 C7816[INIT]后，接收器将搜索所有接收到的数据中是否存在第一个有效初始字符。检测到正向约定初始字符不会更改 S2[MSBF]、C3[TXINV]和 S2[RXINV]，而检测到反向约定初始字符将导致这些字段自动置位。所有接收到的数据（非有效初始字符）将被忽略，所有因无效数据导致的标志位将不会被断言。当 C7816[ANACK]置位时，如果接收到的是无效初始字符，则会返回 NACK，并且将会按照设置生成 RXT 中断。

45.5.7.2 协议 T = 0

选择 T = 0 协议时，使用相对复杂的错误检测方案。数据字符将按照下图所示进行格式化。此方案还用于响应复位和外设引脚选择(PPS)格式。



图 45-178. ISO-7816 T = 0 数据格式

和 UART 所支持的其他协议类似，此协议下，数据字符也包含一个起始位。但在这个协议下，却有两个停止位而不是通常的一个停止位。除了标准的偶数奇偶校验之外，接收器还可以在第一个停止位周期的后半部分生成并返回 NACK。NACK 的长度必须至少为一个时间周期(ETU)且不超过两个时间周期、尝试重新发送字符之前，发送器必须在检测到错误信号之后等待至少两个时间周期(ETU)。

假定 UART 和器件 (智能卡) 事先知道用于接收和发送的是哪个器件。在除 C2[TE] 和 C2[RE]之外的模式下，UART 未提供任何特殊机制来控制接收和发送。此模式还支持初始字符检测功能。

45.5.7.3 协议 T = 1

选择 T = 1 协议时，不使用 NACK 错误检测方案。相反，会根据字符使用奇偶校验位，或根据块使用 CRC 或 LRC，即对于每组字符而言。在此模式下，虽然停止位与下一个起始位之间可能存在其他无效位周期,但允许单个停止位使用该数据格式。数据字符将按照下图所示进行格式化。

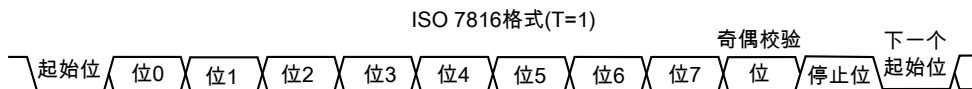


图 45-179. ISO 7816 T=1 数据格式

所传输的最小数据单位是块。块由若干个数据字符组成，其大小因块类型而有所不同。UART 不提供用于解码块类型的机制。作为块的一部分，LRC 或 CRC 包含在内。UART 不会计算发送块的 CRC 或 LRC，也不会验证接收块的 CRC 或 LRC 的有效性。7816 协议要求传输发起方和智能卡 (器件) 轮流发送和接收块。如果 UART 检测到块内的最后一个字符已发送，它会自动清除 C2[TE]、C3[TXDIR]并进入接收模式。因此，软件必须使用要发送的下一个数据对发送缓冲区进行编程，在

确定已收到接收块的最后一个字符后，软件将使能 C2[TE]并置位 C3[TXDIR]。UART 检测到发送块的最后一个字符已发送（如果 TL7816[TLEN] = 0）并且其他四个字符也已发送。其他四个字符的组成如下：三个字符位于 TL7816[TLEN]递减（序言）之前，一个字符位于 TL7816[TLEN] = 0 之后，即最后一个结语字符。

45.5.7.4 等待时间和保护时间参数

ISO-7816 规范定义了多个等待时间和保护时间参数。UART 允许灵活配置以及对这些设置进行违规检测。复位时，等待时间(IS7816[WT])默认为 9600 个 ETU，而保护时间(GT)默认为 12 个 ETU。这些值由 WP7816、WN7816 和 WF7816 寄存器中的参数控制。此外，C7816[TTYTYPE]的值也参与计算。用于计算每个等待时间和保护时间值的 ETU 数公式如表 45-170 所示。

等待时间(WT)是指智能卡器件发送字符的前沿与 UART 或器件发送的上一字符的前沿之间允许的最长时间。相应地，字符等待时间(CWT)是指相同模块内的两个字符前沿之间允许的最长时间。模块等待时间(BWT)是指智能卡器件接收的最后一个模块的前沿与智能卡器件发送的第一个字符的前沿之间允许的最长时间。

保护时间(GT)是指两个连续字符前沿之间允许的最长时间。字符保护时间(CGT)是指相同方向（即，发送或接收）的两个连续字符前沿之间允许的最短时间。模块保护时间(BGT)是指相反方向（即，发送后接收或者接收后发送）的两个连续字符前沿之间允许的最短时间。

当 C7816[TTYTYPE] = 1 或 C7816[ISO_7816E] = 0 或已经按照计数器说明接收或发送新的数据字起始位时，GT 和 WT 计数器将复位。当 C7816[TTYTYPE] = 0 或 C7816[ISO_7816E] = 0 或按照计数器说明接收或发送新的数据字起始位时，CWT、CGT、BWT 和 BGT 计数器将复位。当 C7816[TTYTYPE] = 1 时，部分计数器值需要假定 UART 首先开始时传输第一个数据。当 7816E 禁用、从 C7816[TTYTYPE] = 0 转换为 C7816[TTYTYPE] = 1 或退出复位时，需要此假设。在此情况下，假定上一个不存在传输为已接收到传输。

UART 将会自动处理 GT、CGT 和 BGT，从而使 UART 在对应的保护时间到期之前不会发送数据包。

表 45-170. 等待和保护时间计算

参数	复位值[ETU]	C7816[TTYTYPE] = 0 [ETU]	C7816[TTYTYPE] = 1 [ETU]
等待时间(WT)	9600	$WI \times 480$	不使用
字符等待时间(CWT)	不使用	不使用	$2^{(CWI1)} + CWI2$
模块等待时间(BWT)	不使用	不使用	$(11 + (BWI \times 960 \times GTFD)) * (WTX + 1)$
保护时间(GT)	12	GTN 不等于 255	不使用

下一页继续介绍此表...

表 45-170. 等待和保护时间计算 (继续)

参数	复位值[ETU]	C7816[TTYPE] = 0 [ETU]	C7816[TTYPE] = 1 [ETU]
		12 + GTN GTN 等于 255 12	
字符保护时间(CGT)	不使用	不使用	GTN 不等于 255 12 + GTN GTN 等于 255 11
模块保护时间(BGT)	不使用	不使用	16 + BGI

注

- 用户必须确保使用以下公式设定的字符等待时间(CWT)至少等于 12。小于 12 的值无效, 并且将导致 CWT 意外中断。
- 16 位等待时间整数 WI 通过级联 {WP7816A_T0[WI_H]、WP7816B_T0[WI_L]} 而成。
- 16 位模块等待时间整数 BWI 通过级联 {WP7816A_T1[BWI_H]、WP7816B_T1[BWI_L]} 而成。

45.5.7.5 ATR 持续时间计数器

ISO-7816 规范定义了特定时间 (以 etu 为单位), 在此时间内终端必须接收 ATR (复位响应), 在此时间外终端必须通过启动停用序列来中止卡会话。

UART 在硬件中通过 ATR 持续时间(ATD)计数器支持它, 可使用 AP7816a_T0 和 AP7816b_T0 寄存器对该计数器进行编程。加载到 ADT (ATR 持续时间) 计数器的值由级联的寄存器字段指定, 如下所示; $ADT = \{AP7816a_T0[ADTI_H], AP7816a_T0[ADTI_L]\}$ 。该计数器在检测到 TS 字符时开始计数, 后者在置位 IS7816[INITD]标志位时进行检测。一旦完成 ATR 进程, 必须通过将 0 写入 AP7816x_T0 寄存器来禁用 ATD 计数器, 以防止错误出现 ATD 持续时间中断 IS7816[ATD]。请注意, 仅在 T = 0 模式下支持此功能。

注

ADT 计数器在检测到完整 TS 字符时开始计数。此时必须注意, 自 TS 字符的开始位起已经过 10 个 ETU。在编程 AP7816a_T0 和 AP7816b_T0 寄存器时, 用户必须将其考虑在内。

45.5.7.6 波特率生成

WF7816[GTFD]中的值不会影响时钟频率。SBR 和 BRFD 用于生成时钟频率。此时钟频率仅供 UART 使用，它对智能卡器件不可见。发送器时钟的运行频率是接收时钟频率的 1/16，以便接收器能够在 ETU 期间对收到的值采样 16 次。

45.5.7.7 ISO-7816 操作中的 UART 限制

由于 UART 模块的灵活性，在 ISO-7816 模式下运行时，有几个不受支持的功能和中断。这些限制在寄存器字段定义内进行了介绍。

45.5.8 红外接口

UART 提供发送窄脉冲到红外 LED 和接收窄脉冲并将其转换为串行位（发往 UART）的能力。IrDA 物理层规格定义了用于数据交换的半双工红外通信链路。完整标准包含最高 16 Mb/s 的数据速率。本设计只包含 2.4 Kb/s 和 115.2 Kb/s 之间的数据速率。

UART 有一个红外发送编码器和接收解码器。UART 发送由红外子模块编码的串行位数据，为每个零比特发送窄脉冲。未针对每个 1 位发送脉冲。在接收数据时，使用红外光电二极管检测红外脉冲并通过红外接收解码器（MCU 外部）将其转换成 CMOS 电平。窄脉冲随后被红外接收解码器拉伸，以便返回将由 UART 接收的串行位流。发送脉冲和预期接收脉冲的极性可以反相，从而形成与外部 IrDA 收发器模块（使用低电平有效脉冲）的直接连接。

红外子模块从 UART 获得时钟源。在红外子模块中选择这两个时钟的其中一个，以在发送期间产生 3/16、1/16、1/32 或 1/4 的窄脉冲。

45.5.8.1 红外发送编码器

红外发送编码器将发送移位寄存器中的数据串行位转换为 TXD 信号。发送比特 0 的窄脉冲且无比特 1 脉冲。窄脉冲在位中间发送，持续时间为位时间的 1/32、1/16、3/16 或 1/4。清除 C3[TXINV]时，发送比特 0 的窄高脉冲，置位 C3[TXINV]时，发送比特 0 的窄低脉冲。

45.5.8.2 红外接收解码器

红外接收数据块将 RXD 信号数据转换到接收移位寄存器。每个比特 0 预期会收到窄脉冲，每个比特 1 预期不会收到脉冲。清除 S2[RXINV]时，比特 0 预期会收到窄高脉冲，置位 S2[RXINV]时，比特 0 预期会收到窄低脉冲。接收解码器符合 IrDA 串行红外物理层规格所定义的边沿抖动要求。

45.6 复位

复位为特定值的所有寄存器如[存储器映射和寄存器](#)所示。

45.7 系统级中断源

UART 可以发送多种中断信号。下表列出了 UART 生成的中断源。UART 中断源的本地使能如此表所述。与每个中断的具体操作相关的详细信息包含在[存储器映射和寄存器](#)的各个子章节中。但是，出于其操作的复杂性，[RXEDGIF 说明](#)也概述了与 RXEDGIF 中断有关的其他详细信息。表中所列出的任何 UART 中断请求均可用于使 CPU 退出等待模式。

表 45-171. UART 中断源

中断源	标志	本地使能	DMA 选择位
发送器	TDRE	TIE	TDMAS = 0
发送器	TC	TCIE	-
接收器	IDLE	ILIE	-
接收器	RDRF	RIE	RDMAS = 0
接收器	LBKDIF	LBKDIE	LBKDDMAS = 0
接收器	RXEDGIF	RXEDGIE	-
接收器	OR	ORIE	-
接收器	NF	NEIE	-
接收器	FE	FEIE	-
接收器	PF	PEIE	-
接收器	RXUF	RXUFE	-
发送器	TXOF	TXOFE	-
接收器	WT	WTWE	-
接收器	CWT	CWTE	-
接收器	BWT	BWTE	-
接收器	INITD	INITDE	-
接收器	TXT	TXTE	-
接收器	RXT	RXTE	-
接收器	GTV	GTVE	-

45.7.1 RXEDGIF 说明

在 RxD 引脚上检测到有效边沿时, S2[RXEDGIF]将置位。因此,只能在两种有线模式下检测到有效边沿。仅当 S2[RXEDGIF]置位时才能生成 RXEDGIF 中断。如果在 S2[RXEDGIF]置位之前未使能 RXEDGIE, 则不会生成中断。

45.7.1.1 RxD 边沿检测触发

边沿可通过软件编程为下降沿触发或上升沿触发。可使用 S2[RXINV]选择边沿触发极性。要检测下降沿, 请将 S2[RXINV]编程为 0。要检测上升沿, 请将 S2[RXINV]编程为 1。

在检测边沿之前使用同步逻辑。在检测边沿之前, RxD 输入端的接收数据必须位于无效逻辑电平。当 RxD 输入信号在一个模块时钟周期中被视为逻辑 1 (无效电平) 并且在下一个周期中被视为逻辑 0 (有效电平), 会检测到下降沿。当该输入信号在一个模块时钟周期中被视为逻辑 0 并且在下一个周期中被视为逻辑 1, 会检测到上升沿。

45.7.1.2 清除 RXEDGIF 中断请求

即使 RxD 输入仍保持断言状态, 将逻辑 1 写入 S2[RXEDGIF]也会立即清除 RXEDGIF 中断请求。如果在尝试通过写入 1 来清除 S2[RXEDGIF]时, 在 RxD 上检测到其他有效边沿, 则 S2[RXEDGIF]将保持置位。

45.7.1.3 退出低功耗模式

在低功耗模式 (等待和停止) 下, 接收输入有效边沿检测电路仍处于活动状态。如果中断未屏蔽 (S2[RXEDGIF] = 1), 则接收输入端的有效边缘将使 CPU 退出低功耗模式。

45.8 DMA 操作

在发送器中, 可以将 S1[TDRE] 配置为断言一次 DMA 传输请求。在接收器中, 可以将 S1[RDRF]、和 S2[LBKDIF]配置为断言一次 DMA 传输请求。下表介绍为 DMA 操作配置每个标志位所需的配置字段设置。

表 45-172. DMA 配置

标志位	请求使能位	DMA 选择位
TDRE	TIE = 1	TDMAS = 1
RDRF	RIE = 1	RDMAS = 1
LBKDIF	LBKDIE = 1	LBKDDMAS = 1

当配置用于 DMA 请求的某个标志时，如果该标志置位，那么其相关的 DMA 请求就会变为有效。如果将 S1[RDRF]配置为 DMA 请求，则读取 S1（后跟读取 D）的清除机制不会清除相关联的标志位。DMA 请求将保持断言状态，直到收到已完成 DMA 传输的指示为止。收到此指示后，将清除标志位和相关 DMA 请求。如果 DMA 操作未能移除导致 DMA 请求的情况，则将发出另一个请求。

45.9 应用信息

本节介绍 UART 应用信息。

45.9.1 发送/接收数据缓冲区操作

UART 具有独立的接收和发送缓冲区。这些缓冲区的大小因所使用模块的不同而不同。所使用缓冲区大小为通过 PFIFO[TXFIFOSIZE]和 PFIFO[RXFIFOSIZE]的固定常量。此外，所提供的传统支持允许 FIFO 结构用作深度 1。这是模块的默认/复位行为，可使用 PFIFO[RXFE]和 PFIFO[TXFE]位调整。此外，还提供了单独的发送和接收水印(watermark)级。

可以通过多种方式确保数据块（一组字符）已完成传输。方法如下：

1. 将 TXFIFO[TXWATER]置位为 0。当发送缓冲区中无其他数据时，TDRE 有效。或者，发送移位寄存器也为空时，可使用 S1[TC]标志位表示。
2. 轮询 TCFIFO[TXCOUNT]。假设仅数据块中的数据已输入数据缓冲区中，当 TCFIFO[TXCOUNT] = 0 时，所有数据均已发送完毕或正在传输当中。
3. 可监控 S1[TC]。当 S1[TC]有效时，表示所有数据均已发送完毕，且移位寄存器中不存在当前正在发送的数据。

45.9.2 ISO-7816 初始化序列

本节概述了如何针对 ISO-7816 操作编程 UART。诸如智能卡上电或掉电步骤之类的信息以及何时进行这些操作不在本说明范围内。针对 ISO-7816 操作设置 UART:

1. 选择波特率。将此值写入 UART 波特率寄存器(BDH/L)以启动波特率发生器。请记住，当波特率为 0 时，波特率发生器被禁用。如果不同时写入 BDL，写入 BDH 不起作用。根据 7816 规范，初始（默认）波特率设置应为 $F_i = 372$ 且 $D_i = 1$ 并且最大频率为 5 MHz。换言之，设置 BDH、BDL 和 C4 寄存器后，提供给智能卡器件的传输频率必须为时钟频率的 $1/372$ 且不得超过 5 MHz。
2. 写入以设置 $BDH[LBKDIE] = 0$ 。
3. 写入 C1 以配置字长、奇偶校验和其他配置字段 (LOOPS、RSRC)) 并设置 $C1[M] = 1$ 、 $C1[PE] = 1$ 且 $C1[PT] = 0$ 。
4. 写入以设置 $S2[RWUID] = 0$ 且 $S2[LBKDE] = 0$ 。
5. 写入以设置 $MODEM[RXRTSE] = 0$ 、 $MODEM[TXRTSPOL] = 0$ 、 $MODEM[TXRTSE] = 0$ 且 $MODEM[TXCTSE] = 0$ 。
6. 写入以设置所需的中断使能字段 ($C3[ORIE]$ 、 $C3[NEIE]$ 、 $C3[PEIE]$ 和 $C3[FEIE]$)
7. 写入以设置 $C4[MAEN1] = 0$ 且 $C4[MAEN2] = 0$ 。
8. 写入 C5 寄存器并按照应用所需配置 DMA 控制寄存器。
9. 写入以设置 $C7816[INIT] = 1$ 、 $C7816[TTYPE] = 0$ 且 $C7816[ISO_7816E] = 1$ 。按照需要设置 $C7816[ONACK]$ 和 $C7816[ANACK]$ 。
10. 写入 IE7816 以按照需要设置中断使能参数。
11. 写入 ET7816 并按照需要设置。
12. 写入以设置 $C2[ILIE] = 0$ 、 $C2[RE] = 1$ 、 $C2[TE] = 1$ 、 $C2[RWU] = 0$ 且 $C2[SBK] = 0$ 。按照需要设置中断使能 $C2[TIE]$ 、 $C2[TCIE]$ 和 $C2[RIE]$ 。

此时，UART 将开始监听初始字符。确定之后，它将自动调整 $S2[MSBF]$ 、 $C3[TXINV]$ 和 $S2[RXINV]$ 。软件必须接收和处理复位响应。处理复位响应时，软件必须写入以设置 $C2[RE] = 0$ 且 $C2[TE] = 0$ 。随后，软件应调整 7816 特定以及 UART 通用参数，以匹配和配置在复位响应期间接收到的数据。编程设置新的设置之后（包括新的波特率和 $C7816[TTYPE]$ ），可以按照需要重新使能 $C2[RE]$ 和 $C2[TE]$ 。

45.9.2.1 发送程序 (C7816[TTYTYPE] = 0)

如果所选协议为 C7816[TTYTYPE] = 0，我们假设软件已经提前知道了发送方和接收方。因此，未提供自动发送/接收控制机制。软件必须监控 S1[TDRE]或配置中断，并相应地提供要发送的其他数据。此外，当轮到 UART 发送信息时，软件应置位 C2[TE] = 1 并控制 TXDIR。为便于监控，建议仅发送数据，直到下一个接收/发送切换加载到发送 FIFO/缓冲器为止。

45.9.2.2 传送程序(C7816[TTYTYPE] = 1)

当选择的协议为 C7816[TTYTYPE] = 1 时，将以数据块形式传送数据。开始传送之前，软件必须将数据块的信息字段部分的大小（字节数）写入 TLEN。如果正为数据块传送 CRC，则 TLEN 中的值必须比信息字段的大小大 1。随后，软件必须置位 C2[TE] = 1 和 C2[RE] = 1。接着，软件必须监控 S1[TDRE]/中断并将序言、信息和结语字段写入传送缓冲区。TLEN 将会自动递减，除了序言字节和最后的结语字节外。当最后的结语字节传送完毕之后，UART 会将 C2[TE]和 C3[TXDIR]自动清零，并且 UART 将会自动开始捕捉对传送的数据块的响应。软件检测到已接收响应之后，必须视需要根据紧急程度重复传送流程，以确保不超出数据块等待时间和字符等待时间。

45.9.3 初始化序列（非 ISO-7816）

要发起 UART 发送：

1. 配置 UART。
 - a. 选择波特率。将此值写入 UART 波特率寄存器(BDH/L)以启动波特率发生器。请记住，当波特率为 0 时，波特率发生器被禁用。如果不同时写入 BDL，写入 BDH 不起作用。
 - b. 写入 C1 以配置字长、奇偶校验和其他配置位 (LOOPS、RSRC、M、WAKE、ILT、PE 以及 PT)。写入 C4、MA1 和 MA2 进行配置。
 - c. 通过写入 C2 (TIE、TCIE、RIE、ILIE、TE、RE、RWU 以及 SBK)、S2 (MSBF 和 BRK13) 以及 C3 (ORIE、NEIE、PEIE 和 FEIE)，按照需要使能发送器、中断、接收器和唤醒。从发送移位寄存器移出前同步码或空闲字符。
2. 每个字节的发送步骤。
 - a. 通过读取 S1 或响应 TDRE 中断来监测 S1[TDRE]。也可以监测直接使用 TCFIFO[TXCOUNT]的发送缓冲区中空白空间的数量。

- b. 如果已置位 TDRE 标志位，或发送缓冲区中存在空白，则将待发送的数据写入(C3[T8]/D)。直至发送缓冲区中存在数据时，才会发生新的传送。
3. 对后面的每个发送，重复步骤 2。

注

在常规操作期间，当移位寄存器已从发送缓冲区加载待发送的下一个数据且发送缓冲区中包含的数据字数量少于或等于 TWFIFO[TXWATER]中的数值时，S1[TDRE]置位。此传输在上一帧停止位开始后的 9/16 的位时间开始。

要以最小空闲线路时间分离带前同步码的消息，请在两个消息之间使用该序列。

1. 将第一条信息的最后一个数据字写入 C3[T8]/D。
2. 等待 S1[TDRE]达到高值，当 TWFIFO[TXWATER] = 0 时表示最后一帧已传输到发送移位寄存器。
3. 通过清除和随后置位 C2[TE]来排列前同步码。
4. 将第二条信息的第一个和随后的数据字写入 C3[T8]/D。

45.9.4 溢出(OR)标志位含义

为保持灵活性，溢出标志位(OR)的工作方式略有不同，具体取决于操作模式。有一些需要认真考虑。本节阐述了行为及其产生的影响。无论使用何种模式，如果在置位[OR]时收到数据字，都会阻止 S1[RDRF]和 S1[IDLE]进行认定。如果以前已认定 S1[RDRF]或 S1[IDLE]，它们将保持断言状态，直到被清除为止。

45.9.4.1 溢出操作

S1[OR]的断言表示已发生一个重要事件。此认定表示接收到的数据已丢失，因为数据缓冲区中没有足够的空间来储存它们。因此，S1[OR]置位之后，数据缓冲区中不能再存储其他数据，直到 S1[OR] 被清除。这样就确保了应用程序可以处理溢出状况。

在大多数应用中，因为丢失数据的总量已知，应用会试图让系统返回到已知状态。在清除 S1[OR]之前，所有接收到的数据会被丢弃。为此，软件执行以下操作。

1. 从接收数据缓冲区中移除数据。可通过从数据缓冲区读取数据并进行处理（如果 FIFO 中的数据在溢出事件发生时仍然有价值），或使用 CFIFO[RXFLUSH]清除缓冲区来完成此操作。

- 清除 S1[OR]。注意，如果利用 CFIFO[RXFLUSH]来清除数据，则清除 S1[OR]会导致认定 SFIFO[RXUF]。这是因为，唯一可以清除 S1[OR]的方式需要从 FIFO 中读取额外的信息。在清除 OR 标志位之前，应先小心地禁用 SFIFO[RXUF]中断，然后在 OR 标志位被清除之后清除 SFIFO[RXUF]。

注意，在某些应用中，如果响应溢出事件的速度够快，仍可以恢复丢失的数据。例如，认定 C7816[ISO_7816E]，C7816[TTYTYPE]=1 且 C7816[ONACK]=1 时，此应用也许能够合理地确定设备是否会重新发送丢失的数据。在这种情况下，可能不需要刷新接收器数据缓冲区。而且，如果清除 S1[OR]，则可能重新发送丢失的数据，因而可能会恢复。

认定 LIN 分隔检测(LBKDE)之后，S1[OR]与其他模式相比具有明显不同的行为。如果检测到 LIN 分隔字符，且认定 S2[LBKDIF]之后、接收到首个数据字符之前未清除对应的标志位 S2[LBKDIF]，则无论数据缓冲区中实际存在多少可用空间，S1[OR]都将置位。此行为旨在让软件获得充足的时间从数据缓冲区读取 LIN 分隔字符，从而确保实际检测到分隔字符。分隔字符检查用于某些老旧的方案，并因为这些遗留原因而受到支持。不需要执行此检查的应用无需通过检查存储值来确保它是一个分隔字符，因此可直接清除 S2[LBKDIF]。

45.9.5 溢出 NACK 考虑因素

当使能 C7816[ISO_7816E]且 C7816[TTYTYPE] = 0 时，借助 7816 协议的重新传输特性，可避免在数据缓冲区溢出的情况下丢失数据。利用 C7816[ONACK]，可对模块进行编程，以便在出现溢出事件时发出 NACK。假设智能卡设备已配置重新传输特性，因此会重新传输丢失的数据。虽然很有用，但存在一种可能需要特别考量的编程影响。必须在完全接收数据字之前确定和执行 NACK 传输的需求。在接收 NACK 期间，应用代码会读取数据缓冲区，因而有足够的空间来存储被 NACK 的数据字。即使在 NACK 传输结束后数据缓冲区内拥有了空间，收到的数据也始终会因溢出而被丢弃，ET7816[RXTHRESHOLD] 数值也会递增 1。但是，如果已有足够的空间来写入收到的 NACK 数据，则会阻止 S1[OR]并且不会认定。

45.9.6 匹配地址寄存器

例如，两个匹配地址寄存器允许广播具备第二个匹配地址功能，或允许串行总线具有通用调用地址。

45.9.7 调制解调器的特性

本节介绍调制解调器的特性。

45.9.7.1 利用 RTS 准备接收

为防止接收器数据缓冲区溢出，接收器可使用 RTS 信号向另一个 UART 表明，它已准备好接收数据。其他 UART 可在其 CTS 信号被认定之后发送数据。这种信号交换符合 TIA-232-E 标准的要求。如果通信链接所需的电压电平与 UART 的 RTS 和 CTS 信号的电压电平不匹配，则必须使用收发器。

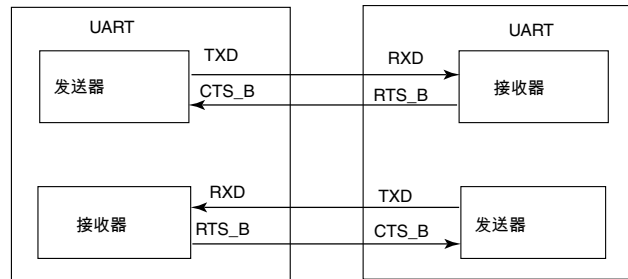


图 45-180. 准备接收

无论发送器的 RTS 信号用于硬件流量控制、收发器驱动器使能，还是完全不用，发送器的 CTS 信号都可用于硬件流量控制。

45.9.7.2 利用 RTS 使能收发器驱动器

RS-485 是一种多压降通信协议，其中，UART 收发器的驱动器为三态，UART 驱动时除外。发送器可利用 RTS 信号来使能收发器的驱动器。RTS 的极性可以与收发器的驱动器使能信号的极性相匹配。请参见下图。

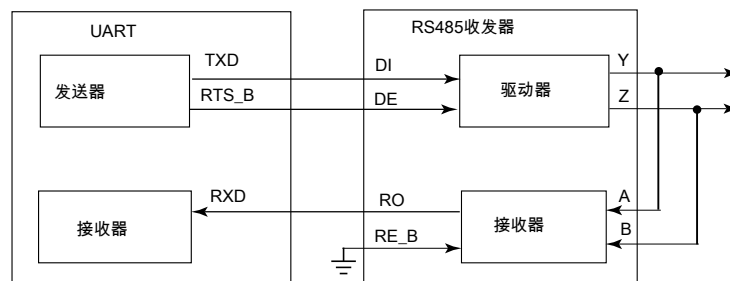


图 45-181. 利用 RTS 使能收发器驱动器

在图中，接收器使能信号认定。该连接的另一种选项是将 RTS_B 同时连接到 DE 和 RE_B。驱动时，收发器的接收器禁用。在此期间，上拉电阻可以把 RXD 拉至一个非浮点值。在单线模式下操作 UART 可以进一步优化该选项，由此腾出 RXD 引脚做其他用途。

45.9.8 IrDA 最小脉宽

IrDA 指定最小脉宽为 $1.6 \mu\text{s}$ 。UART 硬件不具备限制/强制脉宽大于或等于 $1.6 \mu\text{s}$ 的机制。但是，如果将波特率配置为 115.2 kbit/s，将窄脉宽配置为位时间的 3/16，则脉宽为 $1.6 \mu\text{s}$ 。

45.9.9 正在清除 7816 等待定时器 (WT、BWT、CWT) 中断

如果与 IS7816[WT]、IS7816[BWT]及 IS7816[CWT]关联的 7816 等待定时器中断已被清除但仍不符合等待时间要求，则 7816 等待定时器中断将自动重新断言。这种行为与 UART 上的大多数其它中断相似。在大多数情况下，如果造成中断触发的条件在清除中断后仍然存在，会重新断言中断。例如，考虑以下情形：

1. 按照程序，IS7816[WT]在 9600 个无响应周期后断言。
2. 9600 次无响应周期会导致 WT 中断认定。
3. 在第 9700 个周期时，IS7816[WT]会被中断服务程序清除。
4. 在 MT 中断清除之后，智能卡保持不响应。在第 9701 个周期时，会重新认定 WT 中断。

如果清除中断旨在让其不重新认定，则中断服务程序必须在清除中断之前先移除或清除最初造成中断认定的条件。有几种清除此条件的方式，包括确保发生可导致等待定时器重置的事件，例如，传输另一个数据包。

45.9.10 旧版和反向兼容性考虑因素

近期版本的 UART 增加了几项新功能。只要可行，均保留反向兼容性。但是，在某些情况下，这并不可行，或者并不需要。本章介绍了近期增强版本与过去版本在操作上的几点不同。如果使用之前版本的应用代码，则为了顾及以下几点，必须进行审核和更改。此处未列出的项目也可能需要加以考虑，这具体由应用代码决定。

1. 已使用多种保留寄存器和寄存器位，例如 MSFB 和 M10。
2. 当使用无效的地址空间时，此模块现在会生成一个错误。
3. 除非有文件另行说明，否则在某些情况下即使 S1[OR]已置位，也可以断言 S1[IDLE]。
4. S1[OR] 只有在数据缓冲区 (FIFO) 空间不足时才会置位。以前，数据缓冲区的大小始终固定为一，即使数据缓冲区仍有空间，只要 S1[RDRF]置位，S1[OR]标志位也会置位。虽然对于 S1[RDRF]清除机制保持不变，但是让 OR 标志位断言取决于 RDRF 事件而非数据缓冲区是否已满仍会大幅降低缓冲区的有用性（当缓冲区大小大于一时）。
5. 以前，当 C2[RWU] 置位（且 WAKE = 0），IDLE 标志位可重新断言导致中断的每个位，并要求主机处理器重新断言 C2[RWU]。此行为已被更改。现在，当

C2[RWU]已置位 (且 WAKE = 0)，在检测到空闲位之前必须至少先检测到一个非空闲位。

第 46 章

低功耗通用异步接收器/发送器 0 (LPUART0)

46.1 此模块的芯片实现细节

46.1.1 LPUART0 概述

LPUART0 模块支持基本的 UART 和 DMA 接口功能，以及 x4 至 x32 波特率过采样。

当其使用的时钟保持使能时，该模块可以在停止和 VLPS 模式下保持运行。

此模式支持 LIN 从机工作模式。

注

USB DP/DM 引脚可以根据 SIM_MISCCTRL[UARTSELONUSB]，配置为 UART TX/RX。详情请参见 USBFSOTG 一章中的“[USB 上的 UART 性能](#)”一节。

46.2 简介

46.2.1 特性

LPUART 模块特性包括：

- 全双工标准不归零(NRZ)格式
- 可编程波特率（13 位模数分频器），支持 4x 至 32x 的可配置过采样率
- 发送与接收波特率可与总线时钟异步运行：
 - 波特率可配置为不受总线时钟频率影响
 - 支持在停止模式下工作

- 中断、DMA 或轮询操作：
 - 发送数据寄存器为空，且传输完成
 - 接收数据寄存器已满
 - 接收溢出、奇偶错误、帧错误和噪声错误
 - 空闲接收器检测
 - 接收引脚上的有效边沿
 - 断点检测支持 LIN
 - 接收数据匹配
- 硬件奇偶生成和校验
- 可编程 8 位、9 位或 10 位字符长度
- 可编程 1 位或 2 位停止位
- 3 种接收器唤醒方法：
 - 空闲线路唤醒
 - 地址标志唤醒
 - 接收数据匹配
- 自动地址匹配，以减少 ISR 开销：
 - 地址标志匹配
 - 空闲线路地址匹配
 - 地址匹配开始、地址匹配结束
- 可选 13 位分隔字符生成/11 位分隔字符检测
- 可配置空闲长度检测，支持 1、2、4、8、16、32、64 或 128 个空闲字符
- 可选择发送器输出和接收器输入极性
- 硬件流控制支持请求发送(RTS)和清除发送(CTS)信号
- 带可编程脉冲宽度的可选择 IrDA 1.4 归零反相(RZI)格式

46.2.2 操作模式

46.2.2.1 停止模式

LPUART 能在停止模式下保持运行，只要异步传送与接收时钟保持使能。LPUART 可以产生一个中断或 DMA 请求，以实现从停止模式的唤醒。

46.2.2.2 等待模式

当 DOZEEN 位置位时，可以把 LPUART 配置为在等待模式下唤醒。传送端和接收端将完成当前字的传送/接收。

46.2.2.3 调试模式

LPUART 在调试模式下保持运行。

46.2.3 信号说明

信号	说明	I/O
LPUART_TX	发送数据。此引脚一般为输出，但在单线模式（发射器禁用或发射方向配置为接收数据）下为输入（三态）。	I/O
LPUART_RX	接收数据。	I
LPUART_CTS	清除发送。	I
LPUART_RTS	请求发送。	O

46.2.4 功能框图

下图显示 LPUART 的传送器部分。

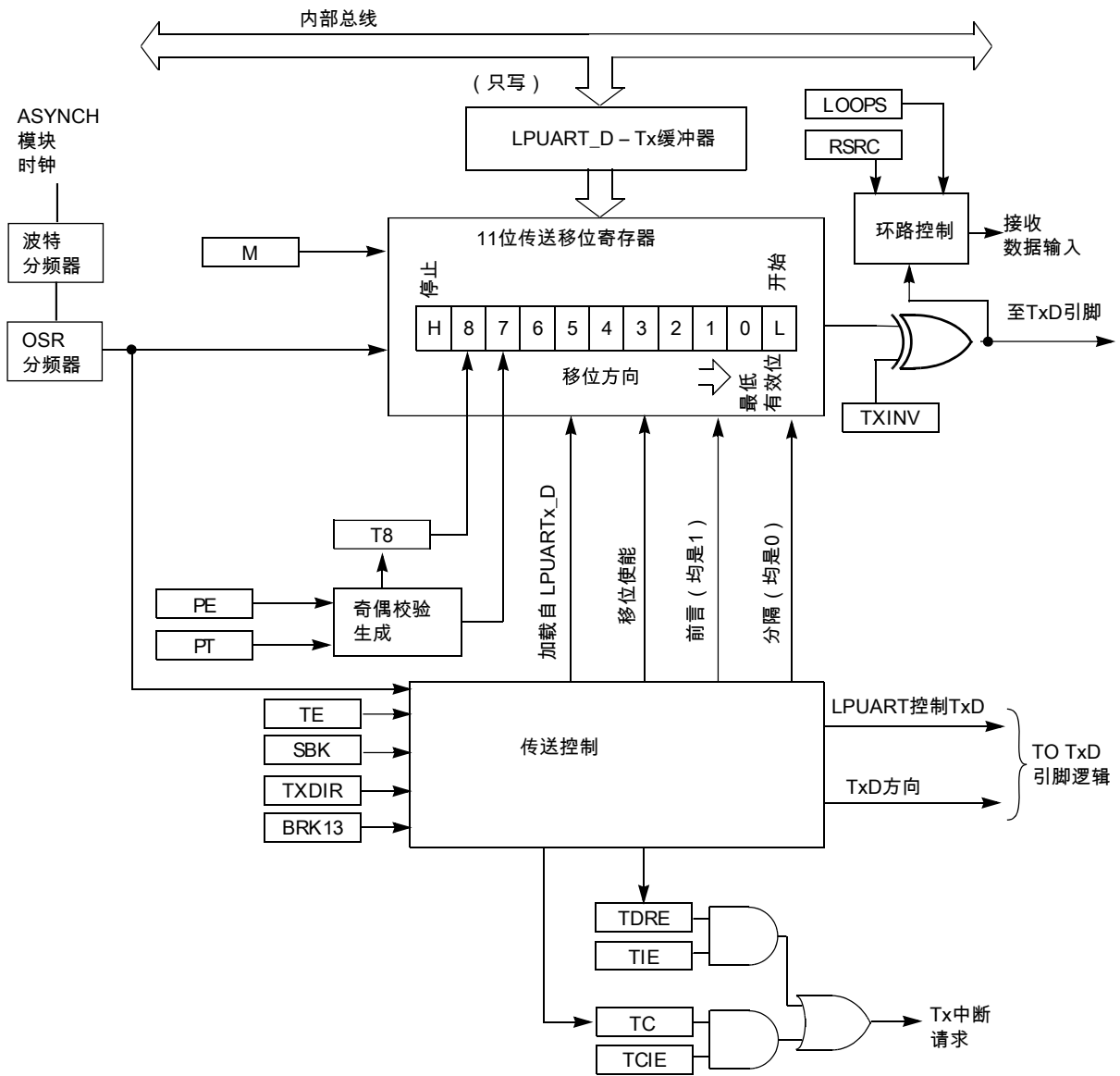


图 46-1. LPUART 传送器结构框图

下图显示 LPUART 的接收器部分。

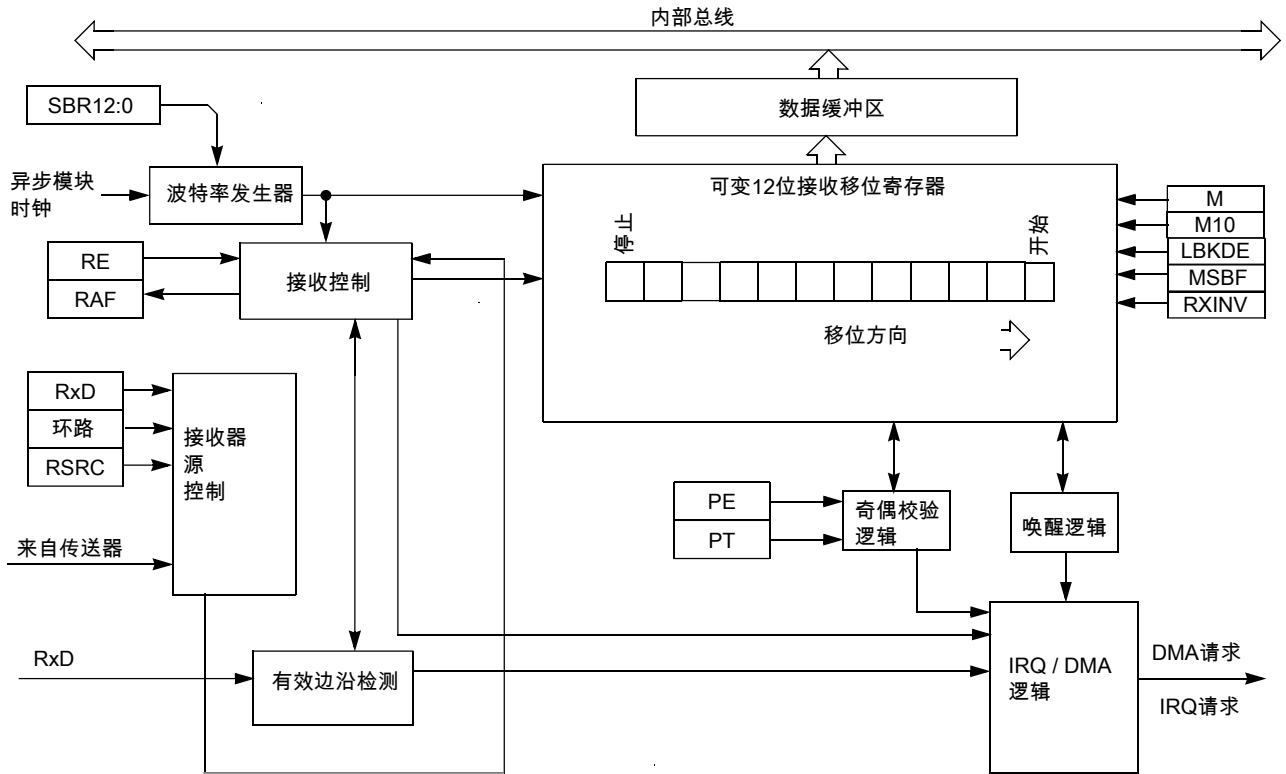


图 46-2. LPUART 接收器结构框图

46.3 寄存器定义

LPUART 含有寄存器可以控制波特率，选择 LPUART 选项，报告 LPUART 状态，并用于传送/接收数据。访问有效寄存器映射外的地址将生成总线错误。

LPUART 存储器映射

绝对地址（十六进制）	寄存器名称	宽度（单位：位）	访问	复位值	小节/页
4002_A000	LPUART 波特率寄存器 (LPUART0_BAUD)	32	R/W	0F00_0004h	46.3.1/1144
4002_A004	LPUART 状态寄存器 (LPUART0_STAT)	32	R/W	00C0_0000h	46.3.2/1146
4002_A008	LPUART 控制寄存器 (LPUART0_CTRL)	32	R/W	0000_0000h	46.3.3/1150
4002_A00C	LPUART 数据寄存器 (LPUART0_DATA)	32	R/W	0000_1000h	46.3.4/1154
4002_A010	LPUART 匹配地址寄存器 (LPUART0_MATCH)	32	R/W	0000_0000h	46.3.5/1156
4002_A014	LPUART 调制解调器 IrDA 寄存器 (LPUART0_MODIR)	32	R/W	0000_0000h	46.3.6/1157

46.3.1 LPUART 波特率寄存器 (LPUARTx_BAUD)

地址: 4002_A000h 基准 + 0h 偏移 = 4002_A000h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R				OSR					0		0	MATCFG					
W	MAEN1	MAEN2	M10					TDMAE		RDMAE				BOTHEDGE	RESYNCDIS		
复位	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	SBNS			SBR													
W	LBKDIE	FXEDGIE															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

LPUARTx_BAUD 字段描述

字段	描述
31 MAEN1	使能匹配地址模式 1 0 正常工作。 1 为 MATCH[MA1]使能自动地址匹配或数据匹配模式。
30 MAEN2	使能匹配地址模式 2 0 正常工作。 1 为 MATCH[MA2]使能自动地址匹配或数据匹配模式。
29 M10	10 位模式选择 M10 位会导致第 10 位成为串行传输的一部分。只有在发送器和接收器均禁用时才能更改此位。 0 接收器和发送器使用 8 位或 9 位数据字符。 1 接收器和发送器使用 10 位数据字符。
28-24 OSR	过采样率 此字段配置接收器的过采样率，范围在 4x (00011)和 32x (11111)之间。写入无效过采样率 (例如，不在 4x 和 32x 之间的值) 会取默认过采样率 16 (01111)。只有在发送器和接收器均禁用时才能更改 OSR 字段。注意，过采样率 = OSR + 1。
23 TDMAE	使能发送器 DMA TDMAE 配置发送数据寄存器空标志 LPUART_STAT[TDRE]，以生成 DMA 请求。 0 DMA 请求禁用。 1 DMA 请求使能。
22 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

LPUARTx_BAUD 字段描述 (继续)

字段	描述
21 RDMAE	<p>使能接收器已满 DMA</p> <p>RDMAE 配置接收器数据寄存器满标志 LPUART_STAT[RDRF]，以生成 DMA 请求。</p> <p>0 DMA 请求禁用。 1 DMA 请求使能。</p>
20 保留	<p>此字段为保留字段。 此只读字段为保留字段且值始终为 0。</p>
19-18 MATCFG	<p>匹配配置</p> <p>配置使用的匹配寻址模式。</p> <p>00 地址匹配唤醒 01 空闲匹配唤醒 10 匹配打开和匹配关闭 11 数据匹配时使能 RWU，匹配打开/关闭用于发送器 CTS 输入</p>
17 BOTHEDGE	<p>双边沿采样</p> <p>使能在波特率时钟两个边沿对接收的数据进行采样，实际上会使接收器在给定过采样率的条件下对输入数据进行采样的次数增加一倍。对于 x4 和 x7 之间的过采样比，此位必须置位，对于较高的过采样率，则是可选项。在接收器禁用时才能更改此位。</p> <p>0 接收器利用波特率时钟的上升沿对输入数据采样。 1 接收器利用波特率时钟的下降沿对输入数据采样。</p>
16 RESYNCDIS	<p>禁用再同步</p> <p>置位时，如果检测到数据一后跟数据零转换，将禁用接收数据字的再同步功能。在接收器禁用时才能更改此位。</p> <p>0 支持在接收数据字中再同步 1 禁用再同步</p>
15 LBKDIE	<p>LIN 断点检测中断使能</p> <p>LBKDIE 使能 LIN 断点检测标志 LBKDIF，以生成中断请求。</p> <p>0 来自 LPUART_STAT[LBKDIF]的硬件中断禁用 (使用轮询)。 1 LPUART_STAT[LBKDIF]标志为 1 时，请求硬件中断。</p>
14 RXEDGIE	<p>RX 输入有效边沿中断使能</p> <p>使能接收输入有效边沿，RXEDGIF，以生成中断请求。在 RXEDGIE 置位时更改 CTRL[LOOP]或 CTRL[RSRC] 可能导致 RXEDGIF 置位。</p> <p>0 来自 LPUART_STAT[RXEDGIF]的硬件中断禁用 (使用轮询)。 1 LPUART_STAT[RXEDGIF]标志为 1 时，请求硬件中断。</p>
13 SBNS	<p>停止位数选择</p> <p>SBNS 决定数据字符是一个停止位还是两个停止位。只有在发送器和接收器均禁用时才能更改此位。</p> <p>0 一个停止位。 1 两个停止位。</p>
SBR	波特率模数分频因子。

下一页继续介绍此表...

LPUARTx_BAUD 字段描述 (继续)

字段	描述
	SBR[12:0]中的 13 位为波特率生成器设定模数分频比。当 SBR 为 1 - 8191 时，波特率等于"波特时钟 / ((OSR+1) × SBR)"。13 位波特率设置[SBR12:SBR0]只能在发送器和接收器均禁用时更新 (LPUART_CTRL[RE]和 LPUART_CTRL[TE]均为 0)。

46.3.2 LPUART 状态寄存器 (LPUARTx_STAT)

地址: 4002_A000h 基准 + 4h 偏移 = 4002_A004h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W	w1c	w1c										w1c	w1c	w1c	w1c	w1c
复位	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MA1F	MA2F	0													
W	w1c	w1c														
复位	0	0	0													

LPUARTx_STAT 字段描述

字段	描述
31 LBKDIF	LIN 断点检测中断标志 当 LIN 断点检测电路使能且检测到 LIN 断点字符时 ,LBKDIF 置位。将 LBKDIF 清零的方法是将 1 写入该位。 0 未检测到 LIN 断点字符。 1 已检测到 LIN 断点字符。
30 RXEDGIF	LPUART_RX 引脚有效边沿中断标志

下一页继续介绍此表...

LPUARTx_STAT 字段描述 (继续)

字段	描述
	<p>当 LPUART_RX 引脚上出现一个有效边沿 (RXINV = 0 时为下降沿, RXINV = 1 时为上升沿) 时, RXEDGIF 置位。将 RXEDGIF 清零的方法是将 1 写入该位。</p> <p>0 接收引脚上未出现有效边沿。 1 接收引脚上已出现有效边沿。</p>
29 MSBF	<p>MSB 优先</p> <p>置位此位会反转在线上发送和接收的位的顺序。此位不会影响位的极性、奇偶校验位的位置或起始位或停止位的位置。只有在发送器和接收器均禁用时才能更改此位。</p> <p>0 LSB (bit0) 是在起始位之后发送的第一位。另外, 在起始位之后接收的第一位标为 bit0。 1 MSB (bit9、bit8、bit7 或 bit6) 是在起始位之后发送的第一位, 具体取决于 CTRL[M]、CTRL[PE]和 BAUD[M10]的设置。另外, 在起始位之后接收的第一位标为 bit9、bit8、bit7 或 bit6, 具体取决于 CTRL[M]和 CTRL[PE]的设置。</p>
28 RXINV	<p>接收数据反转</p> <p>置位该位会反转接收数据输入的极性。</p> <p>注: 置位 RXINV 可在所有情况下反转 LPUART_RX 输入: 数据位、起始位和停止位、断点和空闲。</p> <p>0 接收数据不反转。 1 接收数据反转。</p>
27 RWUID	<p>接收唤醒空闲检测</p> <p>对于空闲字符时的 RWU, RWUID 控制唤醒接收器的空闲字符是否置位 IDLE 位。对于地址匹配唤醒, RWUID 控制 IDLE 位是否在地址不匹配时置位。在接收器禁用时才能更改此位。</p> <p>0 在接收待机状态(RWU = 1)期间, 检测到空闲字符时, IDLE 位不置位。在地址匹配唤醒过程中, IDLE 位不会在地址不匹配时置位。 1 在接收待机状态(RWU = 1)期间, 检测到空闲字符时, IDLE 位置位。在地址匹配唤醒过程中, IDLE 位会在地址不匹配时置位。</p>
26 BRK13	<p>断点字符生成长度</p> <p>BRK13 选择较长的发送断点字符长度。帧传输错误检测不受该位状态的影响。在发送器禁用时才能更改此位。</p> <p>0 断点字符以 10 位时间 (如 M = 0, 则 SBNS = 0) 或 11 位时间 (如 M = 1, 则 SBNS = 0; M = 0, 则 SBNS = 1) 或 12 位时间 (如 M = 1, 则 SBNS = 1; M10 = 1, 则 SNBS = 0) 或 13 位时间 (如 M10 = 1, 则 SNBS = 1) 的长度发送。 1 断点字符以 13 位时间 (如 M = 0, 则 SBNS = 0) 或 14 位时间 (如 M = 1, 则 SBNS = 0; M = 0, 则 SBNS = 1) 或 15 位时间 (如 M = 1, 则 SBNS = 1; M10 = 1, 则 SNBS = 0) 或 16 位时间 (如 M10 = 1, 则 SNBS = 1) 的长度发送。</p>
25 LBKDE	<p>LIN 断点检测使能</p> <p>LBKDE 选择较长的断点字符检测长度。LBKDE 置位时, 接收数据不会存入接收数据缓冲器。</p> <p>0 断点字符以 10 位时间 (如 M = 0, 则 SBNS = 0) 或 11 位时间 (如 M = 1, 则 SBNS = 0; M = 0, 则 SBNS = 1) 或 12 位时间 (如 M = 1, 则 SBNS = 1; M10 = 1, 则 SNBS = 0) 或 13 位时间 (如 M10 = 1, 则 SNBS = 1) 的长度检测。 1 断点字符以 11 位时间 (如 M = 0, 则 SBNS = 0) 或 12 位时间 (如 M = 1, 则 SBNS = 0; M = 0, 则 SBNS = 1) 或 14 位时间 (如 M = 1, 则 SBNS = 1; M10 = 1, 则 SNBS = 0) 或 15 位时间 (如 M10 = 1, 则 SNBS = 1) 的长度检测。</p>

下一页继续介绍此表...

LPUARTx_STAT 字段描述 (继续)

字段	描述
24 RAF	<p>接收器有效标志</p> <p>当接收器检测到一个有效起始位的开始时，RAF 置位；当接收器检测到空闲线路时，RAF 被自动清零。</p> <p>0 LPUART 接收器空闲，等待起始位。 1 LPUART 接收器有效 (LPUART_RX 输入非空闲)。</p>
23 TDRE	<p>发送数据寄存器空标志</p> <p>TDRE 将在发送数据寄存器(LPUART_DATA)为空时置位。要清除 TDRE，写入 LPUART 数据寄存器 (LPUART_DATA)。</p> <p>TDRE 不会受到正在传输的字符的影响，它是在每个发送字符的起始位更新。</p> <p>0 发送数据缓冲器已满。 1 发送数据寄存器空。</p>
22 TC	<p>发送完成标志</p> <p>当传输正在进行或者载入前同步码或断点字符时，TC 将清零。当发送缓冲器为空且无数据、前同步码或断点字符正在发送时，TC 置位。TC 置位后，发送数据输出信号变为空闲 (逻辑 1)。清除 TC 的方法是写入 LPUART_DATA 以发送新数据，清除并置位 LPUART_CTRL[TE]以对前同步码排队，向 LPUART_CTRL[SBK]写入 1 以对断点字符排队。</p> <p>0 发送器有效 (发送数据、前同步码或断点)。 1 发送器空闲 (发送活动完成)。</p>
21 RDRF	<p>接收数据寄存器满标志</p> <p>RDRF 置位。要清除 RDRF，读取 LPUART_DATA 寄存器。</p> <p>正在被接收的字符在整个字符被接收之前，不会导致 RDRF 发生变化。即使 RDRF 置位，也会继续接收该字符，直到因接收到整个字符而发生溢出条件时为止。</p> <p>0 接收数据缓冲器空。 1 接收数据缓冲器满。</p>
20 IDLE	<p>空闲线路标志</p> <p>一段活动时间后，当 LPUART 接收线路变为空闲且持续整个字符时间时，IDLE 置位。ILT 被清零时，接收器在起始位之后开始进行空闲位时间的计数。如果接收字符为全 1，那么这些位时间和停止位时间将计数到接收器检测空闲线路所需的全字符逻辑高电平时间 (10 至 13 位时间)。ILT 置位时，接收器在停止位之后才开始进行空闲位时间的计数。前一字符结束时的停止位和任何逻辑高电平时间不计数到接收器检测空闲线路所需的全字符逻辑高电平时间。</p> <p>要清除 IDLE，向 IDLE 标志写入逻辑 1。将 IDLE 清零后，要再次将其置位，必须等到将新字符存入接收缓冲器或 LIN 断点字符使 LBDIF 标志置位之后。即使接收线路保持闲置状态更长的时间，IDLE 也仅设置一次。</p> <p>0 未检测到空闲线路。 1 检测到空闲线路。</p>
19 OR	<p>接收器溢出标志</p> <p>当软件未能防止接收数据寄存器因数据溢出时，OR 将置位。在完全接收到导致缓冲器溢出的数据字的停止位之后，OR 位将立即置位，同时阻止所有其他错误标志 (FE、NF 和 PF) 置位。移位寄存器中的数据将丢失，但已在 LPUART 数据寄存器中的数据不会受影响。如果 LBDIF 使能且检测到 LIN 断点，要是 LBDIF 未在接收到下一个数据字符之前清除，OR 字段将置位。</p> <p>在 OR 标志置位时，即使存在足够的空间，也不会有额外的数据存入数据缓冲器。要清除 OR，向 OR 标志写入逻辑 1。</p>

下一页继续介绍此表...

LPUARTx_STAT 字段描述 (继续)

字段	描述
	0 未溢出。 1 接收溢出 (新 LPUART 数据丢失)。
18 NF	噪声标志 接收器中使用的高级过采样技术在接收到的每个位中取三个样本。在帧中的任何位时间内，如果任一样本与其余样本不一致，则将检测到该字符的噪声。每当接收到从 LPUART_DATA 读取的下一个字符时，如果在该字符内检测到噪声，NF 将置位。要清除 NF，向 NF 写入逻辑一。 0 未检测到噪声。 1 在 LPUART_DATA 中的接收字符中检测到噪声。
17 FE	帧错误标志 每当接收到从 LPUART_DATA 读取的下一个字符时，如果在预期为停止位的情况下检测到逻辑 0，FE 将置位。要清除 PE，向 PE 写入逻辑一。 0 未检测到帧传输错误。这并不保证帧传输正确。 1 帧传输错误
16 PF	奇偶校验错误标志 在奇偶校验使能(PE = 1)且接收字符中的奇偶位与预期奇偶值不一致的时候，每当接收到从 LPUART_DATA 读取的下一个字符时，PF 将置位。要清除 PF，向 PF 写入逻辑一。 0 无奇偶校验错误。 1 奇偶校验错误。
15 MA1F	匹配 1 标志 每当从 LPUART_DATA 读取的下一个字符与 MA1 匹配时，MA1F 置位。要清除 MA1F，向 MA1F 写入逻辑一。 0 接收的数据不等于 MA1 1 接收的数据等于 MA1
14 MA2F	匹配 2 标志 每当从 LPUART_DATA 读取的下一个字符与 MA2 匹配时，MA2F 置位。要清除 MA2F，向 MA2F 写入逻辑一。 0 接收的数据不等于 MA2 1 接收的数据等于 MA2
保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。

46.3.3 LPUART 控制寄存器 (LPUARTx_CTRL)

该读/写寄存器控制 LPUART 系统的各种可选特性。只有在发送器和接收器均禁用时才能更改此寄存器。

地址: 4002_A000h 基准 + 8h 偏移 = 4002_A008h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R				0	IDLECFG											
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPUARTx_CTRL 字段描述

字段	描述
31 R8T9	接收位 8/发送位 9 当 LPUART 针对 9 位或 10 位数据格式配置时，R8 为接收到的第九个数据位。在读取 9 位或 10 位数据时，先读取 R8，然后读取 LPUART_DATA。 当 LPUART 针对 10 位数据格式配置时，T9 为接收到的第十个数据位。当写入 10 位数据时，先写 T9，然后写 LPUART_DATA。如果 T9 不需要更改以前的值，例如用于生成地址标志或奇偶校验时，则无需在每次写入 LPUART_DATA 时写入 T9。
30 R9T8	接收位 9/发送位 8 当 LPUART 针对 10 位数据格式配置时，R9 为接收到的第十个数据位。在读取 10 位数据时，先读取 R9，然后读取 LPUART_DATA 当 LPUART 针对 9 位或 10 位数据格式配置时，T8 为接收到的第九个数据位。当写入 9 位或 10 位数据时，先写 T8，然后写 LPUART_DATA。如果 T8 不需要更改以前的值，例如用于生成地址标志或奇偶校验时，则无需在每次写入 LPUART_DATA 时写入 T8。
29 TXDIR	单线模式下的 LPUART_TX 引脚方向 当 LPUART 针对单线半双工工作模式 (LOOPS = RSRC = 1) 进行配置时，该位决定 LPUART_TX 引脚的数据方向。当 TXDIR 清零时，在接收器开始从 LPUART_TX 引脚接收数据之前，发送器将接收当前字符(若有)。 0 LPUART_TX 引脚在单线模式下为输入。 1 LPUART_TX 引脚在单线模式下为输出。
28 TXINV	发送数据反转

下一页继续介绍此表...

LPUARTx_CTRL 字段描述 (继续)

字段	描述
	<p>置位该位会反转发送数据输出的极性。</p> <p>注：设置 TXINV 可在所有情况下反转 LPUART_TX 输出：数据位、起始位和停止位、断点和空闲。</p> <p>0 发送数据不反转。 1 发送数据反转。</p>
27 ORIE	<p>溢流中断启用</p> <p>该位使能溢出标志(OR)以生成硬件中断请求。</p> <p>0 OR 中断禁用；使用轮询。 1 OR 置位时，请求硬件中断。</p>
26 NEIE	<p>噪声错误中断使能</p> <p>该位使能噪声标志(NF)以生成硬件中断请求。</p> <p>0 NF 中断禁用；使用轮询。 1 NF 置位时，请求硬件中断。</p>
25 FEIE	<p>帧传输错误中断使能</p> <p>该位使能帧传输错误标志(FE)以生成硬件中断请求。</p> <p>0 FE 中断禁用；使用轮询。 1 FE 置位时，请求硬件中断。</p>
24 PEIE	<p>奇偶校验错误中断使能</p> <p>该位使能奇偶校验错误标志(PF)以生成硬件中断请求。</p> <p>0 PF 中断禁用；使用轮询。 1 PF 置位时，请求硬件中断。</p>
23 TIE	<p>发送中断使能</p> <p>使能 STAT[TDRE]以生成中断请求。</p> <p>0 来自 TDRE 的硬件中断禁用；使用轮询。 1 TDRE 标志为 1 时，请求硬件中断。</p>
22 TCIE	<p>发送完成中断使能</p> <p>TCIE 使能发送完成标志 TC 以生成中断请求。</p> <p>0 来自 TC 的硬件中断禁用；使用轮询。 1 TC 标志为 1 时，请求硬件中断。</p>
21 RIE	<p>接收器中断使能</p> <p>使能 STAT[RDRF]以生成中断请求。</p> <p>0 来自 RDRF 的硬件中断禁用；使用轮询。 1 RDRF 标志为 1 时，请求硬件中断。</p>
20 ILIE	<p>空闲线路中断使能</p> <p>ILIE 使能空闲线路标志 STAT[IDLE]，以生成中断请求。</p>

下一页继续介绍此表...

LPUARTx_CTRL 字段描述 (继续)

字段	描述
	<p>0 来自 IDLE 的硬件中断禁用；使用轮询。</p> <p>1 IDLE 标志为 1 时，请求硬件中断。</p>
19 TE	<p>发送器使能</p> <p>使能 LPUART 发送器。通过先清除、然后置位 TE，也可用 TE 使空闲前同步码进入队列。当 TE 清零时，此寄存器位将读为 1，直到发送器完成当前字符并且 LPUART_TX 引脚三态为止。</p> <p>0 发送器被禁用。</p> <p>1 发送器已使能。</p>
18 RE	<p>接收器使能</p> <p>使能 LPUART 接收器。当 RE 写为 0 时，此寄存器位将读为 1，直到接收器完成当前字符（若有）的接收为止。</p> <p>0 接收器被禁用。</p> <p>1 接收器已使能。</p>
17 RWU	<p>接收器唤醒控制</p> <p>此字段可以设置为将 LPUART 接收器置于待机状态。发生 RWU 事件时，即是说，CTRL[WAKE]清零时为 IDLE 事件，CTRL[WAKE]置位且 STAT[RWUID]清零时为地址匹配事件，RWU 将自动清零。</p> <p>注：如果通道当前未处于空闲状态，RWU 只能在 CTRL[WAKE] = 0（空闲时唤醒）条件下置位。这种情况可由 STAT[RAF]决定。如果标志被设为唤醒 IDLE 事件且通道已经处于空闲状态，LPUART 有可能会丢弃数据。这是因为在允许 IDLE 重新置位之前，在检测到 IDLE 之前，必须收到这些数据，或者检测到 LIN 断点。</p> <p>0 接收器正常工作。</p> <p>1 LPUART 接收器处于待机状态，等待唤醒条件。</p>
16 SBK	<p>发送断点</p> <p>先将 1 写入 SBK 然后再将 0 写入，可让一个断点字符在发送数据流中排队。只要 SBK 置位，就可将长度为逻辑 0 的 10 至 13 或 13 至 16（若 LPUART_STATBRK13置位）位时间的其他断点字符加入队列。根据 SBK 置位和清零相对于当前发送信息的时序，可在软件将 SBK 清零前，将第二个断点字符加入队列。</p> <p>0 发送器正常工作。</p> <p>1 将断点字符加入发送队列。</p>
15 MA1IE	<p>匹配 1 中断使能</p> <p>0 MA1F 中断禁用</p> <p>1 MA1F 中断已使能</p>
14 MA2IE	<p>匹配 2 中断使能</p> <p>0 MA2F 中断已禁用</p> <p>1 MA2F 中断已使能</p>
13–11 保留	<p>此字段为保留字段。</p> <p>此只读字段为保留字段且值始终为 0。</p>
10–8 IDLECFG	<p>空闲配置</p> <p>配置在 IDLE 标志置位之前必须接收的空闲字符数。</p> <p>000 1 个空闲字符</p> <p>001 2 个空闲字符</p>

下一页继续介绍此表...

LPUARTx_CTRL 字段描述 (继续)

字段	描述
	010 4 个空闲字符 011 8 个空闲字符 100 16 个空闲字符 101 32 个空闲字符 110 64 个空闲字符 111 128 个空闲字符
7 LOOPS	环路模式选择 当 LOOPS 置位时, LPUART_RX 引脚从 LPUART 上断开, 发送器输出从内部连接到接收器输入。要使用循环功能, 则必须使能发送器和接收器。 0 正常工作 - LPUART_RX 和 LPUART_TX 使用不同的引脚。 1 环路模式或单线模式, 发送器输出内部连接到接收器输入 (见 RSRC 位)。
6 DOZEEN	使能休眠 0 使能 LPUART 休眠模式。 1 禁用 LPUART 休眠模式。
5 RSRC	接收器源选择 除非 LOOPS 字段置位, 否则, 此字段无意义或作用。当 LOOPS 置位时, RSRC 字段决定接收器移位寄存器输入源。 0 如果 LOOPS 置位且 RSRC 被清除, 则选择内部环回模式, LPUART 不使用 LPUART_RX 引脚。 1 单线 LPUART 模式, 其中, LPUART_TX 引脚连接到发送器输出和接收器输入。
4 M	9 位或 8 位模式选择 0 接收器和发送器使用 8 位数据字符。 1 接收器和发送器使用 9 位数据字符。
3 WAKE	接收器唤醒方法选择 决定哪个条件能在 RWU=1 时唤醒 LPUART : <ul style="list-style-type: none"> • 地址标志位于接收的数据字符最高有效位位置, 或者 • 接收引脚输入信号上的空闲条件。 0 针对空闲线路唤醒配置 RWU。 1 用地址标志唤醒配置 RWU。
2 ILT	空闲线路类型选择 决定接收器何时开始将逻辑 1 计为空闲字符位。计数在有效起始位后或停止位后开始。如果计数在起始位后开始, 则在停止位之前的逻辑 1 字符串可能导致空闲字符错误识别。在停止位后开始计数可避免错误空闲字符识别, 但需要正确同步的发送。 注: 在 LPUART 被设为 ILT = 1 的情况下, 逻辑 0 将在接收到停止位之后自动移位, 从而使空闲计数再复位。 0 空闲字符位计数开始于起始位之后。 1 空闲字符位计数开始于停止位之后。
1 PE	奇偶校验使能 使能硬件奇偶生成和校验。奇偶校验使能后, 正好在停止位之前的位将被当作奇偶校验位。

下一页继续介绍此表...

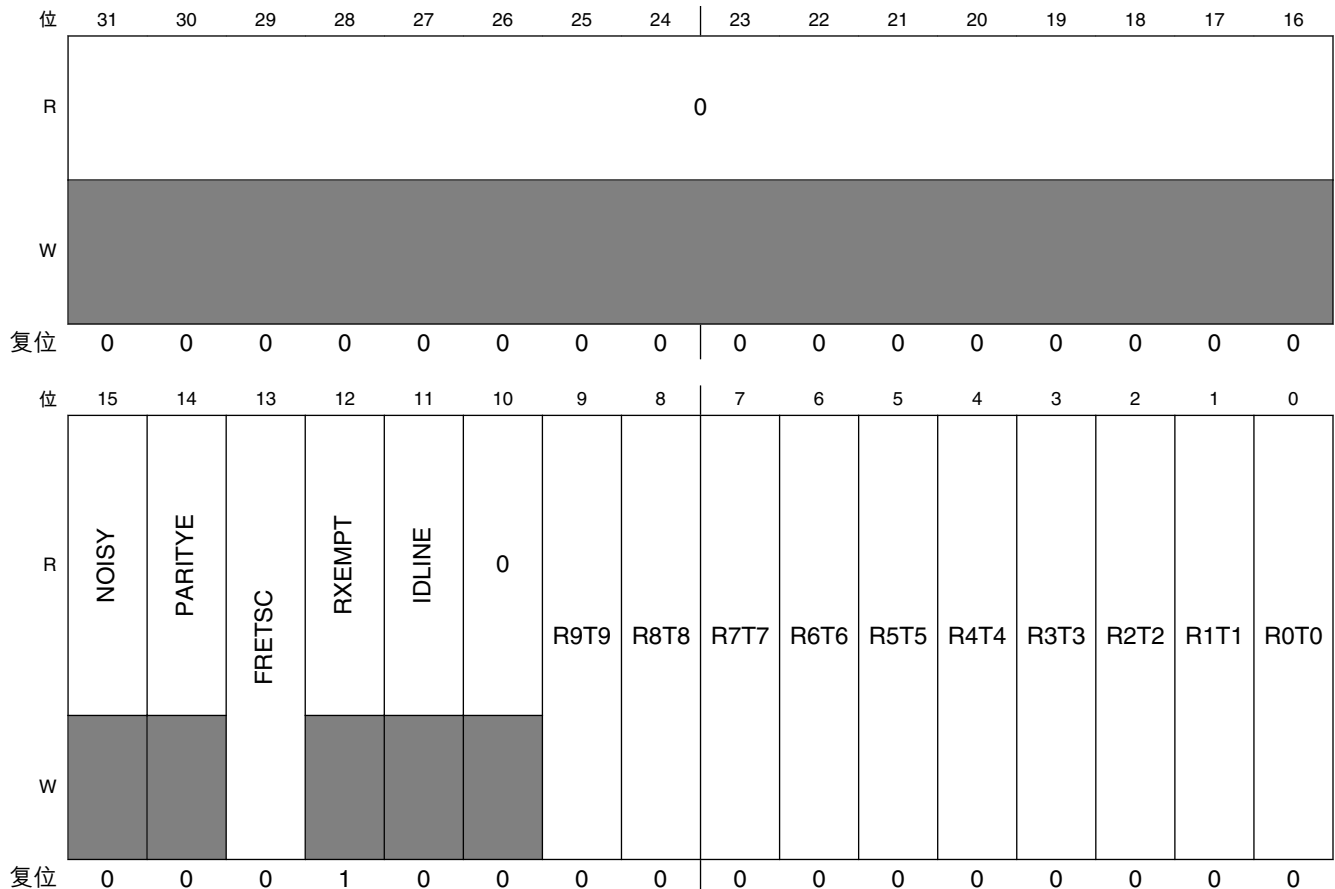
LPUARTx_CTRL 字段描述 (继续)

字段	描述
	0 无硬件奇偶生成或校验。 1 使能奇偶校验。
0 PT	奇偶校验类型 奇偶校验使能时(PE = 1)，该位选择偶数或奇数校验。奇数校验是指数据字符中 1 的总数 (包括奇偶校验位) 为奇数。偶数校验是指数据字符中 1 的总数 (包括奇偶校验位) 为偶数。 0 偶数校验。 1 奇数校验。

46.3.4 LPUART 数据寄存器 (LPUARTx_DATA)

此寄存器实际上是两个独立的寄存器。读操作返回只读接收数据缓冲区的内容，写操作将内容写入只写发送数据缓冲区。部分 LPUART 状态标志的标志自动清除机制还涉及到对该寄存器的读写操作。

地址: 4002_A000h 基准 + Ch 偏移 = 4002_A00Ch



LPUARTx_DATA 字段描述

字段	描述
31-16 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
15 NOISY	DATA[R9:R0]中所含当前已接收数据字在接收时存在噪声。 0 接收到的数据字无噪声。 1 接收到的数据存在噪声。
14 PARITYE	DATA[R9:R0]中所含当前已接收数据字在接收时存在奇偶检验错误。 0 接收到的数据字无奇偶校验错误。 1 接收到的数据字存在奇偶校验错误。
13 FRETSC	帧错误/发送特殊字符 对于读取操作，指示 DATA[R9:R0]中所含当前已接收数据字在接收时存在帧错误。对于写入操作，指示将发送一个断点或空闲字符而非 DATA[T9:T0]中所含内容。T9 用于指示断点字符 (为 0 时) 或空闲字符 (为 1 时)，DATA[T8:T0]的内容应为零。 0 接收到的数据字在读取时无帧错误，写入时将发送一个正常字符。 1 接收到的数据字存在帧错误，发送时将发送一个空闲或断点字符。
12 RXEMPT	接收缓冲器空 当接收缓冲器中无数据时置位。此字段不会考虑接收移位寄存器中的数据。 0 接收缓冲器含有有效数据。 1 接收缓冲器为空，读取返回的数据无效。
11 IDLINE	空闲线路 指示接收器线路在接收 DATA[9:0]中的字符之前处于空闲状态。与 IDLE 标志不同，此位可在首次使能接收器时针对接收到的第一个字符进行设置。 0 接收器在接收此字符之前未处于空闲状态。 1 接收器在接收此字符之前处于空闲状态。
10 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
9 R9T9	读取接收数据缓冲区 9 或写入发送数据缓冲区 9。
8 R8T8	读取接收数据缓冲区 8 或写入发送数据缓冲区 8。
7 R7T7	读取接收数据缓冲区 7 或写入发送数据缓冲区 7。
6 R6T6	读取接收数据缓冲区 6 或写入发送数据缓冲区 6。
5 R5T5	读取接收数据缓冲区 5 或写入发送数据缓冲区 5。
4 R4T4	读取接收数据缓冲区 4 或写入发送数据缓冲区 4。
3 R3T3	读取接收数据缓冲区 3 或写入发送数据缓冲区 3。
2 R2T2	读取接收数据缓冲区 2 或写入发送数据缓冲区 2。

下一页继续介绍此表...

LPUARTx_DATA 字段描述 (继续)

字段	描述
1 R1T1	读取接收数据缓冲区 1 或写入发送数据缓冲区 1。
0 R0T0	读取接收数据缓冲区 0 或写入发送数据缓冲区 0。

46.3.5 LPUART 匹配地址寄存器 (LPUARTx_MATCH)

地址: 4002_A000h 基准 + 10h 偏移 = 4002_A010h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0						MA2										0						MA1										
W	0						0										0						0										
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPUARTx_MATCH 字段描述

字段	描述
31-26 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
25-16 MA2	匹配地址 2 在最高有效位置位且相关联的 BAUD[MAEN]位置位时,将 MA1 和 MA2 寄存器与输入数据地址进行比较。如果匹配,将把以下数据传输至数据寄存器。如果匹配失败,以下数据将被丢弃。软件只能在相关联的 BAUD[MAEN]位清零时写入 MA 寄存器。
15-10 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
MA1	匹配地址 1 在最高有效位置位且相关联的 BAUD[MAEN]位置位时,将 MA1 和 MA2 寄存器与输入数据地址进行比较。如果匹配,将把以下数据传输至数据寄存器。如果匹配失败,以下数据将被丢弃。软件只能在相关联的 BAUD[MAEN]位清零时写入 MA 寄存器。

46.3.6 LPUART 调制解调器 IrDA 寄存器 (LPUARTx_MODIR)

MODEM 寄存器控制用于设置调制解调器配置的选项。

地址: 4002_A000h 基准 + 14h 偏移 = 4002_A014h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														IREN	TNP
W	[Shaded]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TXCTSSRC	TXCTSC	RXRTSE	TXRTSPOL	TXRTSE	TXCTSE		
W	[Shaded]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPUARTx_MODIR 字段描述

字段	描述
31–19 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
18 IREN	红外使能 使能/禁用红外调制/解调。 0 IR 已禁用。 1 IR 已使能。
17–16 TNP	发送器窄脉冲 使 LPUART 发送 1/OSR、2/OSR、3/OSR 或 4/OSR 窄脉冲。 00 1/OSR. 01 2/OSR. 10 3/OSR. 11 4/OSR.
15–6 保留	此字段为保留字段。 此只读字段为保留字段且值始终为 0。
5 TXCTSSRC	发送 CTS 源 配置 CTS 输入源。 0 CTS 输入为 LPUART_CTS 引脚。 1 CTS 输入为反相接收器匹配结果。
4 TXCTSC	发送 CTS 配置 配置是在每个字符开始，还是在发送器空闲时检查 CTS 状态。

下一页继续介绍此表...

LPUARTx_MODIR 字段描述 (继续)

字段	描述
	0 在每个字符开始时对 CTS 输入采样。 1 在发送器空闲时对 CTS 输入采样。
3 RXRTSE	接收器请求-发送使能 允许 RTS 输出控制发送器件的 CTS 输入，以防止接收器溢出。 注：请勿同时置位 RXRTSE 和 TXRTSE。 0 接收器对 RTS 不起作用。 1 如果接收器数据寄存器已满或者检测到的起始位会导致接收器数据寄存器变满，则 RTS 解除置位。如果接收器数据寄存器未变满并且未检测到会导致接收器数据寄存器变满的起始位，则 RTS 置位。
2 TXRTSPOL	发送器请求-发送极性 控制发送器 RTS 的极性。TXRTSPOL 不影响接收器 RTS 的极性。除非 TXRTSE 置位，否则，RTS 将在有效低电平状态下保持取反。 0 发送器 RTS 为有效低电平。 1 发送器 RTS 为有效高电平。
1 TXRTSE	发送器请求-发送使能 在发送前后控制 RTS。 0 发送器对 RTS 不起作用。 1 当将字符存入空发送器数据缓冲器时，RTS 将在起始位发送前一位时间断言。RTS 将在发送器数据缓冲器和移位寄存器中的所有字符（包括最后停止位）完全发送后一位时间解除置位。
0 TXCTSE	发送器清除-发送使能 TXCTSE 控制发送器的工作模式。TXCTSE 的设置可以独立于 TXRTSE 和 RXRTSE 的状态。 0 CTS 对发送器不起作用。 1 使能清除-发送工作模式。发送器每次准备好发送字符时都会检查 CTS 的状态。如果 CTS 断言，则发送字符。如果 CTS 解除断言，信号 TXD 将保持标记状态，发送将延迟到 CTS 断言为止。在发送字符过程中，CTS 中的变化不会影响字符的发送。

46.4 功能说明

LPUART 支持全双工、异步、NRZ 串行通信，还包括波特率生成器、发送器和接收器块。尽管发送器和接收器使用相同的波特率发生器，但它们是独立运行的。以下介绍 LPUART 的各个模块。

46.4.1 生成波特率

波特率发生器中的 13 位系数计数器派生接收器和发送器的波特率。写入 SBR[12:0] 的值范围为 1 至 8191，决定着异步 LPUART 波特时钟的波特时钟分频因子。SBR 位位于 LPUART 波特率寄存器 BDH 和 BDL 中。波特率时钟驱动接收器，发送器则由波特率时钟通过过采样率分频来驱动。根据过采样比，接收器的采样率为每位时间 4 至 32 个样本。

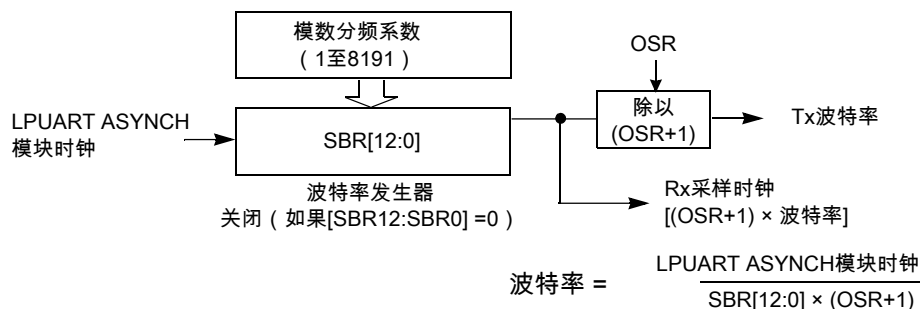


图 46-15. LPUART 波特率生成

波特率生成受两个误差来源的影响：

- 异步 LPUART 波特时钟的整数除法可能未提供准确的目标频率。
- 与异步 LPUART 波特时钟同步可能导致相移。

46.4.2 传送器功能说明

本节介绍 LPUART 传送器的总体结构框图，以及发送分隔和空闲字符的特殊功能。

传送器输出(LPUART_TX)空闲状态默认处于逻辑高电平，CTRL[TXINV]在复位后被清零。传送器输出通过 CTRL[TXINV]置位反相。传送器通过 CTRL[TE]位置位使能。这将把作为闲置状态的一个全角字符帧的报头字符排入队列。然后，发送器继续空闲，直到发送数据缓冲区中有数据可用。程序通过对 LPUART 数据寄存器进行写入操作而将数据存储到传送数据缓冲区。

LPUART 传送器的中心元件是传送移位寄存器，它有 10 位至 13 位长，具体取决于 CTRL[M]、BAUD[M10]和 BAUD[SBNS]控制位的设置。本节其余部分假设 CTRL[M]、BAUD[M10]和 BAUD[SBNS]被清零，选择常规 8 位数据模式。在 8 位数据模式下，移位寄存器占用一个起始位、八个数据位和一个停止位。当传送移位寄存器可载入新字符时，在传送数据寄存器中等待的值会被传输到移位寄存器，与波特率时钟同步，并且传送数据寄存器空(STAT[TDRE])状态标志置位，指示可将另一个字符写入 LPUART_DATA 处的传送数据缓冲区。

LPUART_TX 引脚传出一个停止位后，如果传送数据缓冲区中没有新字符在等待，那么传送器会置位发送完成标志并进入空闲模式，且 LPUART_TX 为高电平，等待发送更多字符。

将 0 写入 CTRL[TE]不会立即禁用传送器。虽然传送器不会开始传送其他字符，但首先必须完成当前正在进行的传送活动(可能包括数据字符、空闲字符或分隔字符)。

46.4.2.1 发送分隔字符和进入队列的空闲字符

LPUART_CTRL[SBK]发送分隔字符，其原始用途是引起旧式电传打字接收机的注意。分隔字符是逻辑 0 的全角字符时间，包括起始位和停止位在内共 10 位至 12 位时间。可通过置位 LPUART_STAT[BRK13]使能更长的 13 位时间的分隔字符。通常，程序将等待 LPUART_STAT[TDRE]变为置位状态，以指示消息的最后一个字符已移至传送移位器，向 LPUART_CTRL[SBK]位写入 1，然后写入 0。一旦移位器可用，此操作就将要发送的分隔字符排入队列。当排入队列的分隔字符移入移位器，与波特率时钟进行同步时，如果 LPUART_CTRL[SBK]保持为 1，则将把另一个分隔字符排入队列。如果接收设备是另一个同样的 LPUART，在所有数据位中分隔字符都将作为 0 被接受，并且发生成帧错误 (LPUART_STAT[FE]=1)。

在位 13 置位且数据位清零的情况下，还可通过向 LPUART_DATA 寄存器写入来传送分隔字符。支持将分隔字符传送作为正常数据流的一部分，同时还允许 DMA 传送分隔字符。

使用空闲线路唤醒后，消息之间需要一个空闲（逻辑 1）的全角字符时间，以唤醒任何处于睡眠状态的接收器。通常，程序将等待 LPUART_STAT[TDRE]变为设置状态，以指示消息的最后一个字符已移至发送移位器，向 LPUART_CTRL[TE]位写入 0，然后写入 1。一旦移位器可用，此操作就将要发送的空闲字符排入队列。LPUART_CTRL[TE]清零时，只要移位器中的字符还未完成，LPUART 传送器绝不会真正释放对 LPUART_TX 引脚的控制。

在位 13 置位且数据位置位的情况下，还可通过向 LPUART_DATA 寄存器写入来传送空闲字符。支持将空闲字符传送作为正常数据流的一部分，同时还允许 DMA 传送分隔字符。

分隔字符的长度受 LPUART_STAT[BRK13]、LPUART_CTRL[M]、LPUART_BAUD[M10]和 LPUART_BAUD[SNBS]位的影响，如下所示：

表 46-16. 分隔字符长度

BRK13	M	M10	SNBS	断点字符长度
0	0	0	0	10 位时间
0	0	0	1	11 位时间
0	1	0	0	11 位时间

下一页继续介绍此表...

表 46-16. 分隔字符长度 (继续)

BRK13	M	M10	SBNS	断点字符长度
0	1	0	1	12 位时间
0	X	1	0	12 位时间
0	X	1	1	13 位时间
1	0	0	0	13 位时间
1	0	0	1	13 位时间
1	1	0	0	14 位时间
1	1	0	1	14 位时间
1	X	1	0	15 位时间
1	X	1	1	15 位时间

46.4.2.2 硬件流量控制

传送器将通过 CTS 的值来控制是否传输。如果已使能清除-发送操作，则在 CTS 被断言时传输字符。如果 CTS 在传输中解除置位，但字符仍留在接收器数据缓冲区中，则将发送移位寄存器中的字符并让 LPUART_TX 保留标记状态，直至 CTS 重新置位。

如果清除-发送操作被禁用，则传送器会忽略 CTS 的状态。

即使同一 UART 接收器的 RTS 信号被禁用，传送器的 CTS 信号也能使能。

46.4.2.3 收发器驱动器使能

传送器可将 LPUART_RTS 用作驱动外部收发器的使能信号。详情参见[用 LPUART_RTS 使能收发器驱动器](#)。若请求-发送操作使能，当一个字符被放入一个空的发送数据缓冲区，LPUART_RTS 会在开始位被传送前置位，置位时间点和开始位之间间隔为传送一个比特位的时间。LPUART_RTS 在传送器数据缓冲区存有字符期间全程保持置位。在所有传送器数据缓冲区和转换寄存器中的字符全部发送后，包括最后一个停止位，LPUART_RTS 解除断言，时间间隔为一个比特位的传送时间。发送分隔字符也会置位 LPUART_RTS，并且与将字符存入发送器数据缓冲器内一样拥有相同的置位和解除置位时序。

仅当发送器使能时才会断言发送器的 LPUART_RTS 信号。但是传送器的 LPUART_RTS 信号不受其 LPUART_CTS 信号影响。LPUART_RTS 将保持置位状态，即使在数据传输过程中禁用发送器也是如此。

46.4.3 接收器功能说明

在本节中，以接收器功能框图为指导来说明接收器的整体功能。接下来将更详细地介绍用于重新构建接收器数据的数据采用技术。最后将介绍接收器唤醒功能的不同方式。

可通过设置 LPUART_STAT[RXINV] 反转接收器输入。可通过设置 LPUART_CTRL[RE] 位启用接收器。字符帧由一个逻辑 0 起始位、八至十个数据位 (MSB 或 LSB 优先) 以及一个或两个逻辑 1 停止位组成。有关 9 位或 10 位数据模式的信息，请参见 [8 位、9 位和 10 位数据模式](#)。有关此讨论的剩余部分，假设 LPUART 配置为正常的 8 位数据模式。

将停止位接收到接收移位器中后，如果接收数据寄存器未满，数据字符将传输到此接收数据寄存器，并设置接收数据寄存器已满 (LPUART_STAT[RDRF]) 状态标志。如果已设置 LPUART_STAT[RDRF]，说明此接收数据寄存器 (缓冲器) 已满，将设置溢出 (OR) 状态标志，且新数据将丢失。由于 LPUART 接收器为双缓冲，设置 LPUART_STAT[RDRF] 之后，在必须读取接收数据缓冲器中的数据以避免接收器溢出之前，程序有一个全字符时间。

当某个程序检测到接收数据寄存器已满 (LPUART_STAT[RDRF] = 1) 时，它通过读取 LPUART_DATA 从此接收数据寄存器中获取数据。有关标志清除的详情，请参见 [中断和状态标志](#)。

46.4.3.1 数据采样技术

LPUART 接收器支持可配置过采样速率，具体为采样波特率时钟的 4 倍至 32 倍。接收器以过采样速率与波特率之积开始对逻辑电平进行采样，以搜索 LPUART_RX 串行数据输入引脚上的下降沿。下降沿被定义为三个连续的逻辑 1 样本后的逻辑 0 样本。过采样波特率时钟将位时间分为 4 至 32 段，范围为 1 至 OSR (其中，OSR 为所配置过的采样比率)。定位到一个下降沿后，将在 $(OSR/2)$ 、 $(OSR/2)+1$ 和 $(OSR/2)+2$ 上再采三个样本，以确保这是真实的起始位，而不仅仅是噪声。如果这三个样本至少有两个为 0，接收器将视为已经与接收字符同步。如果在接收器被置位为已同步之前检测到另一个下降沿，则接收器从第一段开始重启采样。

然后此接收器将在 $(OSR/2)$ 、 $(OSR/2)+1$ 和 $(OSR/2)+2$ 对各个位时间 (包括起始位和停止位) 进行采样，以确定该位的逻辑电平。此逻辑电平被认为是相关的位时间内所采的大部分样本的逻辑电平。如果在某个字符帧中，任何位时间 (包括起始位和停止位) 中的任何样本未与此位的逻辑电平保持一致，则当接收的字符传输到接收数据缓冲器时，将设置噪声标志 (LPUART_STAT[NF])。

当 LPUART 接收器配置为在波特率时钟的两个边沿均采样时，每个接收位中的段数实际上会翻倍（从 1 至 $OSR \times 2$ ）。然后在 OSR 、 $OSR+1$ 和 $OSR+2$ 对起始位和数据位采样。对于 4 倍至 7 倍的采样速率，必须在时钟的两个沿均使能采样；对于更高的过采样速率，此为可选项。

下降沿检测逻辑继续寻找下降沿。如果检测到边沿，采样时钟将重新与位时间进行同步（除非再同步被禁用）。当存在噪声或不匹配的波特率时，这种方式可提高接收器的可靠性。它不能改善最差情况分析，因为某些字符在字符帧中的任何地方都没有多余的下落沿。

发生帧错误时，如果接收的字符不是断点间隔字符，将用三个逻辑 1 样本填充用于搜索下降沿的采样逻辑，这样就几乎马上可以检测到新的起始位。

46.4.3.2 接收器唤醒操作

接收器唤醒和接收器地址匹配是一种硬件机制，使 LPUART 接收器可忽略目标为不同接收器的消息中的字符。

在接收器唤醒中，所有接收器都会评估每条消息的第一个字符，一旦确定此消息针对不同接收器，它们将向相关接收器写入逻辑 1 以唤醒控制位

(LPUART_CTRL[RWU])。设置 RWU 位和 LPUART_S2[RWUID] 位时，将禁止设置与此接收器相关的状态标志（除了闲置位 IDLE 以外），从而消除处理不重要的消息字符所需的软件开销。在一条消息的末尾或下一条消息的开头，所有接收器将自动强制将 LPUART_CTRL[RWU] 设置为 0，使及时唤醒所有接收器以查看下一条消息的第一个字符。

在接收器地区匹配过程中，地址匹配在硬件中完成，LPUART 接收器会忽略不符合地区匹配要求的所有字符。

表 46-17. 接收器唤醒选项

RWU	MA1 MA2	MATCFG	WAKE:RWUID	接收器唤醒
0	0	X	X	正常操作
1	0	00	00	在空闲线路唤醒接收器，IDLE 标志未置位
1	0	00	01	在空闲线路唤醒接收器，IDLE 标志置位
1	0	00	10	在地址标记处唤醒接收器
1	1	11	X0	在数据匹配时唤醒接收器
0	1	00	X0	地址标记地址匹配，IDLE 标志未针对已丢弃字符置位

下一页继续介绍此表...

表 46-17. 接收器唤醒选项 (继续)

RWU	MA1 MA2	MATCFG	WAKE:RWUID	接收器唤醒
0	1	00	X1	地址标记地址匹配, IDLE 标志针对已丢弃字符置位
0	1	01	X0	空闲线路地址匹配
0	1	10	X0	地址匹配开启, 地址匹配关闭, IDLE 标志未针对已丢弃字符置位
0	1	10	X1	地址匹配开启, 地址匹配关闭, IDLE 标志针对已丢弃字符置位

46.4.3.2.1 空闲线路唤醒

清除唤醒后, 将为接收器配置闲置线路唤醒。在此模式中, 当接收器检测到一个全字符时间时, 将自动清除 LPUART_CTRL[RWU]。LPUART_CTRL[M]和 LPUART_BAUD[M10]控制位选择 8 位至 10 位数据模式, 而 LPUART_BAUD[SBNS]位选择 1 位或 2 位停止位数字以确定构建全字符时间所需的闲置位时间数, 由于有起始位和停止位, 因此通常是 10 至 13 位时间。

当 LPUART_CTRL[RWU] 为 1 且 LPUART_STAT[RWUID]为 0 时, 用于唤醒接收器的闲置状态不会设置 LPUART_STAT[IDLE]标志。接收器将唤醒并等待下一条消息的第一个数据字符, 以 LPUART_STAT[RDRF] 标志并生成中断 (如果启用)。当 LPUART_STAT[RWUID]为 1 时, 不论 LPUART_CTRL[RWU]为 0 还是 1, 任何闲置状态都会设置 LPUART_STAT[IDLE]标志并生成中断 (如果启用)。

闲置线路类型 (LPUART_CTRL[ILT]) 控制位可选择两种方式检测闲置线路。清除 LPUART_CTRL[ILT]后, 闲置位计数器从起始位之后开始计数, 以便字符末尾的的停止位和任何逻辑 1 都计入闲置的全角字符时间。设置 LPUART_CTRL[ILT]后, 闲置位计数器在停止位时间之后才开始计数, 因此闲置检测不受上一条消息的最后一个字符中的数据影响。

46.4.3.2.2 地址标记唤醒

设置 LPUART_CTRL[WAKE]后, 会针对地址标记唤醒配置接收器。在此模式下, LPUART_CTRL[RWU]会在接收器在接收到字符的最高有效位检测到逻辑 1 时自动清零。

地址标记唤醒允许报文包含闲置字符, 但需要保留 MSB 以用于地址帧中。在接收到停止位和置位 LPUART_STAT[RDRF]标志之前, 地址帧的 MSB 的逻辑 1 清除 LPUART_CTRL[RWU] 位。在这种情况下, 即使在大部分字符时间里接收器都处于休眠状态, MSB 置位的字符被接收。

46.4.3.2.3 数据匹配唤醒

当 LPUART_CTRL[RWU]置位且 LPUART_BAUD[MATCFG]等于 11 时，接收器被配置为数据匹配唤醒模式。在此模式下，当接收器检测到与 MATCH[MA1]字段匹配的字符时（若 BAUD[MAEN1]置位），或者当检测到与 MATCH[MA2]匹配的字符时（若 BAUD[MAEN2]置位），LPUART_CTRL[RWU]将自动清零。

46.4.3.2.4 地址匹配操作

当 LPUART_BAUD[MAEN1]或 LPUART_BAUD[MAEN2]位置位且 LPUART_BAUD[MATCFG]等于 00 时，地址匹配操作使能。在该功能中，LPUART_RX 引脚收到一个字符，如果正好位于停止位之前的位位置中为逻辑 1，则认为该字符是一个地址，并与关联的 MATCH[MA1]或 MATCH[MA2]字段比较。如果比较结果相匹配，该字符才会传输到接收缓冲器，并且 LPUART_STAT[RDRF]置位。如果停止位之前的位位置为逻辑 0，则接收到的所有后续字符视为与该地址相关联的数据，并传输至接收数据缓冲器。如果标记的地址匹配不成功，则不会传输至接收数据缓冲器，且所有后续字符（停止位之前的位位置为逻辑 0）也会被丢弃。如果 LPUART_BAUD[MAEN1]和 LPUART_BAUD[MAEN2]位均为负，则接收器正常工作，接收到的所有数据都会传输至接收数据缓冲器。

MATCH[MA1]和 MATCH[MA2]字段的地址匹配操作的工作方式相同。

- 如果 LPUART_BAUD[MAEN1]和 LPUART_BAUD[MAEN2]二者仅置位一个，则只会将标记的地址与关联的匹配寄存器进行比较，并且在匹配时才会将数据传输至接收数据缓冲器。
- 如果 LPUART_BAUD[MAEN1]和 LPUART_BAUD[MAEN2]同时置位，则会将标记的地址与两个匹配寄存器进行比较，并且只会在与任一寄存器相匹配时传输数据。

46.4.3.2.5 空闲匹配操作

当 LPUART_BAUD[MAEN1]或 LPUART_BAUD[MAEN2]位置位且 LPUART_BAUD[MATCFG]等于 01 时，空闲匹配操作使能。在此功能下，在空闲线路条件之后由 LPUART_RX 引脚接收的第一字符将被认为是一个地址，并将与关联的 MA1 或 MA2 寄存器比较。如果比较结果相匹配，该字符才会传输到接收缓冲器，并且 LPUART_STAT[RDRF]置位。所有后续字符将被认为是与该地址关联的数据，并会在检测到下一个空闲线路条件之前，传输至接收数据缓冲器。如果地址匹配不成功，则不会传输至接收数据缓冲器，并且会丢弃下一个空闲条件之前的所有后续帧。如果 LPUART_BAUD[MAEN1]和 LPUART_BAUD[MAEN2]位均为负，则接收器正常工作，接收到的所有数据都会传输至接收数据缓冲器。

MA1 和 MA2 寄存器具有相同的空闲匹配操作方式。

- 如果 LPUART_BAUD[MAEN1]和 LPUART_BAUD[MAEN2]二者仅置位一个，则只会将空闲线路之后的第一个字符与关联的匹配寄存器进行比较，并且在匹配时才会将数据传输至接收数据缓冲器。
- 如果 LPUART_BAUD[MAEN1]和 LPUART_BAUD[MAEN2]同时置位，则会将空闲线路之后的第一个字符与两个匹配寄存器进行比较，并且只会与任一寄存器相匹配时传输数据。

46.4.3.2.6 匹配开启匹配关闭操作

当 LPUART_BAUD[MAEN1]和 LPUART_BAUD[MAEN2]均置位且 LPUART_BAUD[MATCFG]等于 10 时，匹配开启、匹配关闭操作使能。在此功能下，由 LPUART_RX 引脚接收且与 MATCH[MA1]匹配的一个字符将被接收并传输到接收缓冲器，LPUART_STAT[RDRF]置位。所有后续字符均被认为是数据并会被传输至接收数据缓冲器，直到接收到与 MATCH[MA2]寄存器相匹配的字符为止。与 MATCH[MA2]相匹配的字符以及所有后续字符均会被丢弃，这种情况将持续下去，直到接收到与 MATCH[MA1]相匹配的另一个字符为止。如果 LPUART_BAUD[MAEN1]和 LPUART_BAUD[MAEN2]位均为负，则接收器正常工作，接收到的所有数据都会传输至接收数据缓冲器。

注

匹配开启、匹配关闭操作需要同时置位 LPUART_BAUD[MAEN1]和 LPUART_BAUD[MAEN2]。

46.4.3.3 硬件流量控制

为支持硬件流量控制，可以将接收器编程为自动解除置位和置位 LPUART_RTS。

- 在传输完成之前，LPUART_RTS 保持置位状态，即使在数据传输过程中禁用发送器也是如此。详情请参见[用 LPUART_RTS 使能收发器驱动器](#)。
- 如果接收器请求发送功能已使能，若接收器数据寄存器中的字符数已满，或者检测到将导致接收器数据寄存器满的起始位，接收器将自动解除置位 LPUART_RTS。
- 当接收器数据寄存器中的字符数未满足并且未检测到将导致接收器数据寄存器满的起始位时，接收器将置位 LPUART_RTS。如果已置位 STAT[RDRF]，则不会有任何影响。
- 即使已解除置位 LPUART_RTS，接收器也会继续接收字符，直到接收器数据缓冲区溢出为止。
- 如果接收器请求发送功能已禁用，则接收器 LPUART_RTS 将保持解除置位状态。

46.4.3.4 红外解码器

红外解码器将接收到的字符从 IrDA 格式转换为接收器使用的 NRZ 格式。解码器还有一个 OSR 过采样波特率时钟计数器，可以滤除噪声并在收到 1 时发出指示。

46.4.3.4.1 起始位检测

如果已清除 STAT[RXINV]，则已接收字符的第一个下降沿与起始位对应。红外解码器将复位其计数器。此时，接收器还会开始其起始位检测进程。在检测到起始位后，接收器会将其位时间与此起始位时间同步。对于剩余字符接收，红外解码器的计数器与接收器的位时间计数器将相互独立地进行计数。

46.4.3.4.2 噪声过滤

在红外解码器计数器前半段检测到的任何其他上升沿将被解码器忽略。解码器可能无法检测到小于一个过采样波特时钟的任何脉冲，无论是在计数前半段还是后半段发现的，均是如此。

46.4.3.4.3 低位检测

在解码器计数的后半段，上升沿将被解码为 0，并发送至接收器。同时解码器计数器复位。

46.4.3.4.4 高位检测

在上一个上升沿后面的 OSR 过采样波特率时钟内，如果未发现上升沿，则解码器会向接收器发送 1。

如果下一个位是 0，其到达时间较晚，则会根据 [低位检测](#) 检测到低位。发送到接收器的值从 1 变为 0。随后，如果在接收器位时间采样周期外出现噪声脉冲，则延迟 0 将不会记录为噪声。

46.4.4 其他 LPUART 功能

以下各节介绍其他 LPUART 功能。

46.4.4.1 8 位、9 位和 10 位数据模式

LPUART 发送器和接收器可通过设置 LPUART_CTRL[M]配置为在 9 位数据模式下工作，也可通过设置 LPUART_CTRL[M10]配置为在 10 位数据模式下工作。在 9 位模式下，有第 9 数据位；在 10 位模式下，则有第 10 数据位。对于发送数据缓冲器，这些位存储于 LPUART_CTRL[T8]和 LPUART_CTRL[T9]中。对于接收器，这些位存储于 LPUART_CTRL[R8]和 LPUART_CTRL[R9]中。它们也可通过对 LPUART_DATA 寄存器的 16 位或 32 位访问操作来访问。

为了一致地写入发送数据缓冲区，应先写入 LPUART_CTRL[T8]和 LPUART_CTRL[T9]，再写入 LPUART_DATA[7:0]。对于对 LPUART_DATA 寄存器进行的 16 位和 32 位写入操作，所有 10 个发送位都会同时写入发送数据缓冲器。

如果要作为新字符的第 9 位和第 10 位发送的位值与上一个字符相同，则无需再次写入 LPUART_CTRL[T8]和 LPUART_CTRL[T9]。当数据从发送数据缓冲器传输到发送移位器时，将在数据从 LPUART_DATA[7:0]传输到移位器的同时复制 LPUART_CTRL[T8]和 LPUART_CTRL[T9]中的值。

9 位数据模式通常与奇偶校验配合使用以实现八位数据加第九位的奇偶性，或与地址标志唤醒配合使用以使第九数据位可作为唤醒位。10 位数据模式通常与奇偶校验和地址标志唤醒配合使用，以使第 9 位用作唤醒位，第 10 位用作奇偶校验位。在自定义协议中，第 9 位与"或者"或"第 10 位也可作为软件控制标志。

46.4.4.2 空闲长度

空闲字符指其中起始位、所有数据位和停止位均处于标记位置的字符。CTRL[ILT] 寄存器可以配置为从一起起始位（任何数据位和停止位均向空闲字符检测计数）或从前一停止位开始检测空闲字符。

在检测到空闲线路条件之前，必须接收的空闲字符数也可通过 CTRL[IDLECFG] 字段配置。此字段配置在 STAT[IDLE] 标志置位、STAT[RAF] 标志清零且以下一个接收到的字符置位 DATA[IDLINE] 标志之前必须接收的空闲字符数。

空闲线路唤醒和空闲匹配操作也会受到 CTRL[IDLECFG] 字段的影响。当地址匹配或匹配开启/关闭操作使能时，STAT[RWUID] 位置位会导致被丢弃的任何字符被当作空闲字符处理。

46.4.4.3 循环模式

设置 LPUART_CTRL[LOOPS] 后，相同寄存器中的 LPUART_CTRL[RSRC] 位选择循环模式(LPUART_CTRL[RSRC] = 0)或单线模式(LPUART_CTRL[RSRC] = 1)。循环模式有时用于检查软件（与外部系统中的连接无关）以帮助隔离系统问题。在此模式下，发送器输出从内部连接到接收器输入，且 LPUART_RX 引脚未被 LPUART 使用。

46.4.4.4 单线操作

LPUART_CTRL[LOOPS] 置位后，相同寄存器中的 RSRC 位选择循环模式(LPUART_CTRL[RSRC] = 0) 或单线模式(LPUART_CTRL[RSRC] = 1)。单线模式实施半双工串行连接。接收器从内部连接到发送器输出和 LPUART_TX 引脚(LPUART_RX 引脚未使用)。

在单线模式中，LPUART_CTRL[TXDIR] 位控制 LPUART_TX 引脚上串行数据的方向。清除 LPUART_CTRL[TXDIR] 后，LPUART_TX 引脚成为接收器的输入，且发送器暂时与 LPUART_TX 引脚断开，使外部设备可向此接收器发送串行数据。LPUART_CTRL[TXDIR] 置位后，LPUART_TX 引脚成为由发送器驱动的输出，内部回路循环被禁用，结果，接收器将无法收到发送器发出的字符。

46.4.5 红外接口

LPUART 提供发送窄脉冲到红外 LED 和接收窄脉冲并将其转换为串行位（发往 LPUART）的能力。IrDA 物理层规格定义了用于数据交换的半双工红外通信链路。完整标准包含最高 16 Mb/s 的数据速率。本设计只包含 2.4 Kb/s 和 115.2 Kb/s 之间的数据速率。

LPUART 有一个红外发送编码器和接收解码器。LPUART 发送由红外子模块编码的串行位数据，每个零比特均被转换为窄脉冲。未发送每个 1 位的脉冲。在接收数据时，使用红外光电二极管检测红外脉冲并通过红外接收解码器（LPUART 外部）将其转换成 CMOS 电平。然后，窄脉冲随后被红外子模块拉伸，以便返回为将由 LPUART 接收的串行位流。发送脉冲和预期接收脉冲的极性可以反相，从而形成与外部 IrDA 收发器模块（使用高电平有效脉冲）的直接连接。

红外子模块从 LPUART 获得时钟源。在红外子模块中选择这两个时钟的其中一个，以在发送期间产生 1/OSR、2/OSR、3/OSR 或 4/OSR 的窄脉冲。

46.4.5.1 红外发送编码器

红外发送编码器转换从发送移位寄存器到 LPUART_TX 信号的数据串行位。发送比特 0 的窄脉冲且无比特 1 脉冲。窄脉冲在位开始时发送,持续一个位时的 1/OSR、2/OSR、3/OSR 或 4/OSR。清除 LPUART_CTRL[TXINV]时,发送比特 0 的窄低脉冲,置位 LPUART_CTRL[TXINV]时,发送比特 0 的窄高脉冲。

46.4.5.2 红外接收解码器

红外接收区块转换从 LPUART_RX 信号到接收移位寄存器的数据。每个 0 位预期会收到窄脉冲,每个 1 未预期不会收到脉冲。清除 LPUART_STAT[RXINV]时,0 位预期会收到窄低脉冲,置位 LPUART_STAT[RXINV]时,0 位预期会收到窄高脉冲。接收解码器符合 IrDA 串行红外物理层规格所定义的边沿抖动要求。

46.4.6 中断和状态标志

LPUART 发送器有两个状态标志,可选择性地生成硬件中断请求。发送数据寄存器为空 (LPUART_STAT[TDRE]) 时,发送数据缓冲器中有用于向 LPUART_DATA 写入另一个发送字符的空间。如果已设置发送中断使能 (LPUART_CTRL[TIE]) 位,设置 LPUART_STAT[TDRE]置位时,就会请求硬件中断。发送完成 (LPUART_STAT[TC]) 表示发送器已完成所有数据、报头和隔离字符的发送并处于空闲状态,同时 LPUART_TX 处于无效状态。此标志通常用于带调制解调器的系统,用于确定安全关闭此调制解调器的时间。如果已设置发送完成中断使能 (LPUART_CTRL[TCIE]) 位,设置 LPUART_STAT[TC]时将请求硬件中断。如果已清除相应的 LPUART_CTRL[TIE] or LPUART_CTRL[TCIE]本地中断掩码,则可使用软件轮询而非硬件中断以监控 LPUART_STAT[TDRE]和 LPUART_STAT[TC]状态标志。

当某个程序检测到接收数据寄存器已满 (LPUART_STAT[RDRF] = 1) 时,它通过读取 LPUART_DATA 以从此接收数据寄存器中获取数据。LPUART_STAT[RDRF]标志通过读取 LPUART_DATA 清零。

IDLE 状态标志包括一种逻辑,可防止此标志在 LPUART_RX 线路长时间空闲时重复置位。IDLE 可通过在 LPUART_STAT[IDLE]标志中写入 1 来清零。将 LPUART_STAT[IDLE]清零后,要再次将其置位,必须等到接收器接收到至少一个新字符且 LPUART_STAT[RDRF]置位。

如果接收的字符中检测到导致 LPUART_STAT[RDRF]置位的相关错误,将在 LPUART_STAT[RDRF]置位的同时置位错误标志,即噪声标志 (LPUART_STAT[NF])、成帧错误 (LPUART_STAT[FE]) 和奇偶错误标志 (LPUART_STAT[PF])。溢出情况下这些标志不置位。

当一个新字符准备好从接收移位器转移到接收数据缓冲区时，如果 LPUART_STAT[RDRF] 已经置位，则溢出 (LPUART_STAT[OR]) 标志而非数据将置位，同时丢失任何相关的 NF、FE 或 PF 条件。

若接收的字符匹配 MATCH[MA1]"与"或者"或"MATCH[MA2]内容，则 LPUART_STAT[MA1F]"与"或者"或"LPUART_STAT[MA2F]标志与 LPUART_STAT[RDRF]同时置位。

LPUART_RX 串行数据输入引脚上的有效边沿随时都可导致 LPUART_STAT[RXEDGIF]标志置位。将 LPUART_STAT[RXEDGIF]标志清零的方法是将 1 写入该标志。此功能需要使能接收器(LPUART_CTRL[RE] = 1)。

第 47 章

FlexIO

47.1 此模块的芯片实现细节

47.1.1 FlexIO 在本器件上的配置

本节概述了 FlexIO 的特性和模块配置。

此器件包含一个带 8 个引脚、4 个计数器和 4 个移位器的 FlexIO 模块。

此器件中的 FlexIO 模块支持以下其他特性：

- 支持仅采用串行通信协议的仿真

47.1.2 FlexIO 触发器选项

FlexIO 有一个可选的触发输入源，受 FlexIO_TIMCTLn[TRGSEL]（4 位字段）控制，用于启动计数器"与"或者"或"重新加载计数器。可用选项如下表所示。

表 47-1. FlexIO 触发器选项

FlexIO_TIMCTLn[TRGSEL] (4 位字段)	所选源
0000	外部触发器
0001	CMP0
0010	保留
0011	保留
0100	PIT 通道 0 输出
0101	PIT 通道 1 输出
0110	PIT 通道 2 输出
0111	PIT 通道 3 输出
1000	TPM0 溢出

下一页继续介绍此表...

表 47-1. FlexIO 触发器选项 (继续)

FlexIO_TIMCTLn[TRGSEL] (4 位字段)	所选源
1001	TPM1 溢出
1010	TPM2 溢出
1011	保留
1100	RTC 警报
1101	RTC 秒钟
1110	LPTMR 输出
1111	保留

47.2 简介

47.2.1 概述

FlexIO 是一个非常灵活的可配置的模块，提供各种功能，包括：

- 支持各种串行通信协议的仿真
- 灵活的 16 位定时器支持各种触发、复位、使能和禁用条件

这些功能由 FlexIO 提供，具有如下特点：

- 低软件/CPU 开支：与软件实现的位带操作相比，开支更少，与专用外设 IP 相比，开支更多。
- 空间优化/高功效实施方案：相比于为某种协议定制的专用外设，更加高效。

47.2.2 特性

FlexIO 模块广泛支持各种协议，包括但不限于以下协议：

- UART
- I2C
- SPI
- I2S
- PWM/波形生成

关键特性：

- 32 位移位寄存器阵列，具有发送、接收和数据匹配模式
- 双缓冲移位器工作模式支持连续数据传输

- 移位器级联可支持更大传输大小
- 自动开始/停止位生成
- 中断、DMA 或轮询发送/接收操作
- 可编程的波特率不受总线时钟频率的影响，并支持在停止模式下异步工作
- 高度灵活的 16 位定时器支持各种内部或外部触发、复位、使能和禁用条件

47.2.3 结构框图

下方的框图是对 FlexIO 定时器和移位器配置的总体概述。

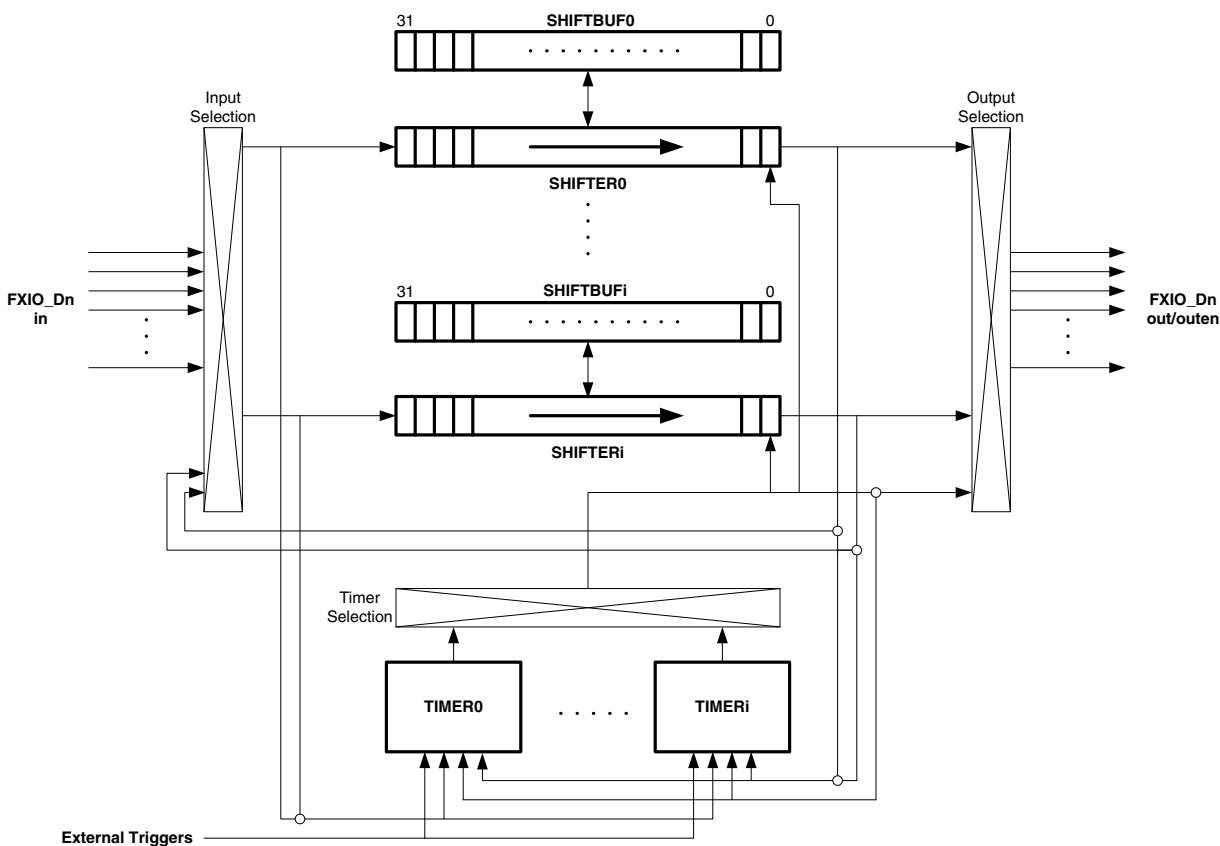


图 47-1. FlexIO 框图

47.2.4 操作模式

FlexIO 模块支持下表描述的芯片工作模式。

表 47-2. FlexIO 模块支持的芯片工作模式

芯片模式	FlexIO 操作
运行	正常操作
停止/等待	在休眠使能位(CTRL[DOZEN])置位且 FlexIO 采用在停止/等待模式下保持运行的外部或内部时钟源的情况下, 可以继续工作。
低漏电停止	休眠使能(CTRL[DOZEN])位被忽略, FlexIO 将等待所有定时器完成任何挂起操作后, 才确认进入低漏电模式。
调试	在调试使能位(CTRL[DBG E])置位的条件下, 可以继续工作。

47.2.5 FlexIO 信号说明

信号	说明	I/O
FXIO_Dn (n=0...7)	双向 FlexIO 移位器和定时器引脚输入/输出	I/O

47.3 存储器映射和寄存器

FLEXIO 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4005_F000	版本 ID 寄存器 (FLEXIO_VERID)	32	不适用	0101_0000h	47.3.1/1178
4005_F004	参数寄存器 (FLEXIO_PARAM)	32	不适用	参见章节	47.3.2/1179
4005_F008	FlexIO 控制寄存器 (FLEXIO_CTRL)	32	不适用	0000_0000h	47.3.3/1180
4005_F00C	引脚状态寄存器 (FLEXIO_PIN)	32	不适用	0000_0000h	47.3.4/1181
4005_F010	移位器状态寄存器 (FLEXIO_SHIFTSTAT)	32	w1c	0000_0000h	47.3.5/1182
4005_F014	移位器错误寄存器 (FLEXIO_SHIFTEERR)	32	w1c	0000_0000h	47.3.6/1182
4005_F018	定时器状态寄存器 (FLEXIO_TIMSTAT)	32	w1c	0000_0000h	47.3.7/1183
4005_F020	移位器状态中断使能 (FLEXIO_SHIFTSIEN)	32	不适用	0000_0000h	47.3.8/1184
4005_F024	移位器错误中断使能 (FLEXIO_SHIFTEIEN)	32	不适用	0000_0000h	47.3.9/1184
4005_F028	定时器中断使能寄存器 (FLEXIO_TIMIEN)	32	不适用	0000_0000h	47.3.10/1185
4005_F030	移位器状态 DMA 使能 (FLEXIO_SHIFTSDEN)	32	不适用	0000_0000h	47.3.11/1185
4005_F080	移位器控制 N 寄存器 (FLEXIO_SHIFTCTL0)	32	不适用	0000_0000h	47.3.12/1186
4005_F084	移位器控制 N 寄存器 (FLEXIO_SHIFTCTL1)	32	不适用	0000_0000h	47.3.12/1186
4005_F088	移位器控制 N 寄存器 (FLEXIO_SHIFTCTL2)	32	不适用	0000_0000h	47.3.12/1186

下一页继续介绍此表...

FLEXIO 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4005_F08C	移位器控制 N 寄存器 (FLEXIO_SHIFTCTL3)	32	不适用	0000_0000h	47.3.12/ 1186
4005_F100	移位器配置 N 寄存器 (FLEXIO_SHIFTCFG0)	32	不适用	0000_0000h	47.3.13/ 1188
4005_F104	移位器配置 N 寄存器 (FLEXIO_SHIFTCFG1)	32	不适用	0000_0000h	47.3.13/ 1188
4005_F108	移位器配置 N 寄存器 (FLEXIO_SHIFTCFG2)	32	不适用	0000_0000h	47.3.13/ 1188
4005_F10C	移位器配置 N 寄存器 (FLEXIO_SHIFTCFG3)	32	不适用	0000_0000h	47.3.13/ 1188
4005_F200	移位器缓冲器 N 寄存器 (FLEXIO_SHIFTBUF0)	32	不适用	0000_0000h	47.3.14/ 1189
4005_F204	移位器缓冲器 N 寄存器 (FLEXIO_SHIFTBUF1)	32	不适用	0000_0000h	47.3.14/ 1189
4005_F208	移位器缓冲器 N 寄存器 (FLEXIO_SHIFTBUF2)	32	不适用	0000_0000h	47.3.14/ 1189
4005_F20C	移位器缓冲器 N 寄存器 (FLEXIO_SHIFTBUF3)	32	不适用	0000_0000h	47.3.14/ 1189
4005_F280	移位器缓冲器 N 位交换寄存器 (FLEXIO_SHIFTBUFBIS0)	32	不适用	0000_0000h	47.3.15/ 1190
4005_F284	移位器缓冲器 N 位交换寄存器 (FLEXIO_SHIFTBUFBIS1)	32	不适用	0000_0000h	47.3.15/ 1190
4005_F288	移位器缓冲器 N 位交换寄存器 (FLEXIO_SHIFTBUFBIS2)	32	不适用	0000_0000h	47.3.15/ 1190
4005_F28C	移位器缓冲器 N 位交换寄存器 (FLEXIO_SHIFTBUFBIS3)	32	不适用	0000_0000h	47.3.15/ 1190
4005_F300	移位器缓冲器 N 字节交换寄存器 (FLEXIO_SHIFTBUFBYS0)	32	不适用	0000_0000h	47.3.16/ 1190
4005_F304	移位器缓冲器 N 字节交换寄存器 (FLEXIO_SHIFTBUFBYS1)	32	不适用	0000_0000h	47.3.16/ 1190
4005_F308	移位器缓冲器 N 字节交换寄存器 (FLEXIO_SHIFTBUFBYS2)	32	不适用	0000_0000h	47.3.16/ 1190
4005_F30C	移位器缓冲器 N 字节交换寄存器 (FLEXIO_SHIFTBUFBYS3)	32	不适用	0000_0000h	47.3.16/ 1190
4005_F380	移位器缓冲器 N 位字节交换寄存器 (FLEXIO_SHIFTBUFBBS0)	32	不适用	0000_0000h	47.3.17/ 1190
4005_F384	移位器缓冲器 N 位字节交换寄存器 (FLEXIO_SHIFTBUFBBS1)	32	不适用	0000_0000h	47.3.17/ 1190
4005_F388	移位器缓冲器 N 位字节交换寄存器 (FLEXIO_SHIFTBUFBBS2)	32	不适用	0000_0000h	47.3.17/ 1190
4005_F38C	移位器缓冲器 N 位字节交换寄存器 (FLEXIO_SHIFTBUFBBS3)	32	不适用	0000_0000h	47.3.17/ 1190
4005_F400	定时器控制 N 寄存器 (FLEXIO_TIMCTL0)	32	不适用	0000_0000h	47.3.18/ 1191

下一页继续介绍此表...

FLEXIO 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4005_F404	定时器控制 N 寄存器 (FLEXIO_TIMCTL1)	32	不适用	0000_0000h	47.3.18/ 1191
4005_F408	定时器控制 N 寄存器 (FLEXIO_TIMCTL2)	32	不适用	0000_0000h	47.3.18/ 1191
4005_F40C	定时器控制 N 寄存器 (FLEXIO_TIMCTL3)	32	不适用	0000_0000h	47.3.18/ 1191
4005_F480	定时器配置 N 寄存器 (FLEXIO_TIMCFG0)	32	不适用	0000_0000h	47.3.19/ 1193
4005_F484	定时器配置 N 寄存器 (FLEXIO_TIMCFG1)	32	不适用	0000_0000h	47.3.19/ 1193
4005_F488	定时器配置 N 寄存器 (FLEXIO_TIMCFG2)	32	不适用	0000_0000h	47.3.19/ 1193
4005_F48C	定时器配置 N 寄存器 (FLEXIO_TIMCFG3)	32	不适用	0000_0000h	47.3.19/ 1193
4005_F500	定时器比较 N 寄存器 (FLEXIO_TIMCMP0)	32	不适用	0000_0000h	47.3.20/ 1195
4005_F504	定时器比较 N 寄存器 (FLEXIO_TIMCMP1)	32	不适用	0000_0000h	47.3.20/ 1195
4005_F508	定时器比较 N 寄存器 (FLEXIO_TIMCMP2)	32	不适用	0000_0000h	47.3.20/ 1195
4005_F50C	定时器比较 N 寄存器 (FLEXIO_TIMCMP3)	32	不适用	0000_0000h	47.3.20/ 1195

47.3.1 版本 ID 寄存器 (FLEXIO_VERID)

地址: 4005_F000h 基准 + 0h 偏移 = 4005_F000h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								FEATURE															
W	MAJOR								MINOR								FEATURE															
复位	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXIO_VERID 字段描述

字段	描述
31-24 MAJOR	主要版本号 本只读字段返回模块规格的主要版本号。
23-16 MINOR	次要版本号 本只读字段返回模块规格的次要版本号。
FEATURE	功能规格编号 本只读字段返回功能集编号。

下一页继续介绍此表...

FLEXIO_VERID 字段描述 (继续)

字段	描述
0x0000	已执行的标准功能。
0x0001	支持状态、逻辑和并行模式。

47.3.2 参数寄存器 (FLEXIO_PARAM)

地址: 4005_F000h 基准 + 4h 偏移 = 4005_F004h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TRIGGER								PIN								TIMER								SHIFTER							
W	TRIGGER								PIN								TIMER								SHIFTER							
复位	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0

FLEXIO_PARAM 字段描述

字段	描述
31-24 TRIGGER	触发器编号 已设置的外部触发器数目。
23-16 PIN	引脚编号 FlexIO 模块拥有的外部引脚数目。
15-8 TIMER	定时器编号 FlexIO 拥有的定时器数量。
SHIFTER	移位器编号 FlexIO 模块拥有的移位器数量。

47.3.3 FlexIO 控制寄存器 (FLEXIO_CTRL)

地址: 4005_F000h 基准 + 8h 偏移 = 4005_F008h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DOZEN	DBGE	0													
W	DOZEN	DBGE	[Reserved]													
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													FASTACC	SWRST	FLEXEN
W	[Reserved]													FASTACC	SWRST	FLEXEN
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXIO_CTRL 字段描述

字段	描述
31 DOZEN	使能休眠 在休眠模式下，禁用 FlexIO 操作。在低漏电停止模式下，此字段将忽略，FlexIO 始终禁用。

下一页继续介绍此表...

FLEXIO_CTRL 字段描述 (继续)

字段	描述
	0 在休眠模式下, FlexIO 使能。 1 在休眠模式下, FlexIO 禁用。
30 DBGE	调试模式使能 在调试模式下, 使能 FlexIO 操作。 0 在调试模式下, FlexIO 禁用。 1 在调试模式下, FlexIO 使能。
29-3 保留	此只读字段为保留字段且值始终为 0。
2 FASTACC	快速访问 使能对 FlexIO 寄存器的快速寄存器访问, 但要求 FlexIO 时钟的频率至少为总线时钟频率的两倍。 0 配置用于对 FlexIO 进行正常的寄存器访问 1 配置用于对 FlexIO 进行快速寄存器访问
1 SWRST	软件复位 FlexIO 控制寄存器不受软件复位的影响, FlexIO 中的所有其他逻辑均受软件复位的影响, 且寄存器访问被忽略, 直到此位被清除。此位保持置位直到被软件清除, 且在 FlexIO 时钟域清除复位。 0 软件复位禁用 1 软件复位已使能, 除控制寄存器以外的所有 FlexIO 寄存器均复位。
0 FLEXEN	FlexIO 使能 0 FlexIO 模块禁用。 1 FlexIO 模块使能。

47.3.4 引脚状态寄存器 (FLEXIO_PIN)

地址: 4005_F000h 基准 + Ch 偏移 = 4005_F00Ch

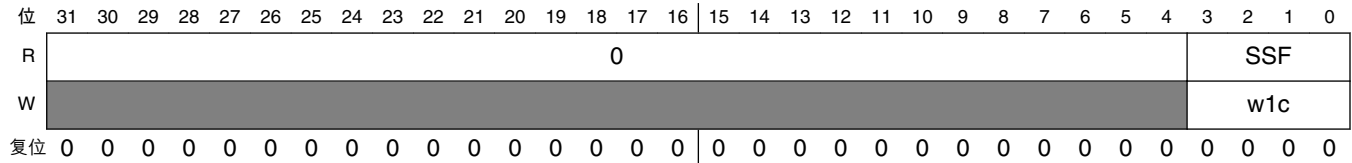
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																PDI															
W																	PDI															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXIO_PIN 字段描述

字段	描述
31-8 保留	此只读字段为保留字段且值始终为 0。
PDI	引脚数据输入 在每个 FlexIO 引脚上返回输入数据。

47.3.5 移位器状态寄存器 (FLEXIO_SHIFTSTAT)

地址: 4005_F000h 基准 + 10h 偏移 = 4005_F010h

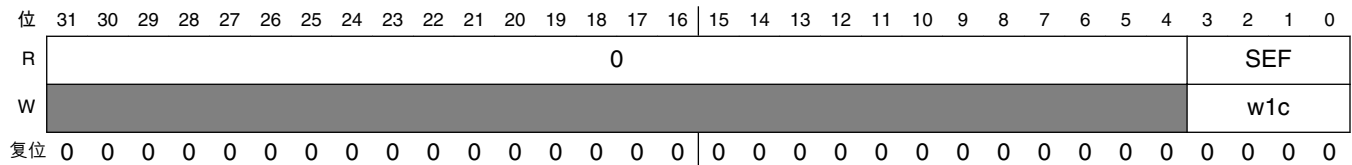


FLEXIO_SHIFTSTAT 字段描述

字段	描述
31-4 保留	此只读字段为保留字段且值始终为 0。
SSF	<p>移位器状态标志位</p> <p>发生下列事件之一时，会更新移位器状态标志位：</p> <p>当 SMOD=接收时，SHIFTBUF 与来自寄存器（SHIFTBUF 已满）的数据一起加载时状态标志位置位，在读取 SHIFTBUF 寄存器时清除状态标志位。</p> <p>当 SMOD=发送时，在 SHIFTBUF 数据传输至移位器（SHIFTBUF 为空）或初始配置为 MOD=发送时状态标志位置位，在写入 SHIFTBUF 寄存器时清除状态标志位。</p> <p>当 SMOD=匹配存储时，在 SHIFTBUF 和移位器匹配时状态标志位置位，在读取 SHIFTBUF 寄存器时清除状态标志位。</p> <p>当 SMOD=匹配继续时，返回 SHIFTBUF 和移位器之间的当前匹配结果。</p> <p>也可通过向适用于所有模式的标志位中写入逻辑 1 来清除状态标志位，匹配继续除外。</p> <p>0 状态标志位被清除 1 状态标志位置位</p>

47.3.6 移位器错误寄存器 (FLEXIO_SHIFTErr)

地址: 4005_F000h 基准 + 14h 偏移 = 4005_F014h



FLEXIO_SHIFTErr 字段描述

字段	描述
31-4 保留	此只读字段为保留字段且值始终为 0。
SEF	<p>移位器错误标志位</p> <p>发生下列事件之一时，移位器错误标志位置位：</p>

下一页继续介绍此表...

FLEXIO_SHIFTErr 字段描述 (继续)

字段	描述
	<p>当 SMOD=接收时，表示移位器在从 SHIFTBUF (SHIFTBUF 溢出) 读取之前的数据之前，已做好向 SHIFTBUF 中存储新数据的准备，或者表示接收的起始或停止位与期望值不匹配。</p> <p>当 SMOD=发送时，表示在将新数据写入 SHIFTBUF (SHIFTBUF 下溢) 之前，移位器已做好从 SHIFTBUF 加载新数据的准备。</p> <p>当 SMOD=匹配存储时，表示在从 SHIFTBUF (SHIFTBUF 溢出) 读取之前的匹配数据之前，匹配事件已发生。</p> <p>当 SMOD=匹配继续时，错误标志位在 SHIFTBUF 和移位器之间发生匹配时置位。</p> <p>可通过向标志位写入 1 来清除。当 SMOD=匹配继续时，也可在读取 SHIFTBUF 寄存器时清除。</p> <p>0 移位器错误标志位被清除 1 移位器错误标志位位置位</p>

47.3.7 定时器状态寄存器 (FLEXIO_TIMSTAT)

地址: 4005_F000h 基准 + 18h 偏移 = 4005_F018h

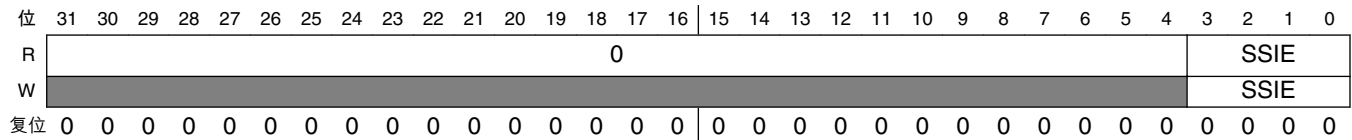
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TSF															
W																	w1c															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

FLEXIO_TIMSTAT 字段描述

字段	描述
31-4 保留	此只读字段为保留字段且值始终为 0。
TSF	<p>定时器状态标志位</p> <p>定时器状态标志位的置位情况由定时器模式决定，且可通过向标志位写入逻辑 1 来清除。</p> <p>在 8 位计数器模式中，当高 8 位计数器等于 0 且递减时，定时器状态标志位置位，这也导致计数器会重新加载比较寄存器中的值。</p> <p>在 8 位 PWM 模式中，当高 8 位计数器等于 0 且递减时，定时器状态标志位置位，这也导致计数器会重新加载比较寄存器中的值。</p> <p>在 16 位计数器模式中，当 16 位计数器等于 0 且递减时，定时器状态标志位置位，这也导致计数器会重新加载比较寄存器中的值。</p> <p>0 定时器状态标志位被清除 1 定时器状态标志位位置位</p>

47.3.8 移位器状态中断使能 (FLEXIO_SHIFTSIEN)

地址: 4005_F000h 基准 + 20h 偏移 = 4005_F020h

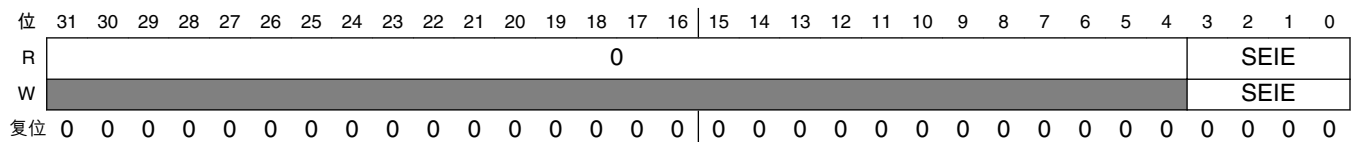


FLEXIO_SHIFTSIEN 字段描述

字段	描述
31-4 保留	此只读字段为保留字段且值始终为 0。
SSIE	移位器状态中断使能 当对应的 SSF 置位时，使能中断生成。 0 移位器状态标志中断已禁用 1 移位器状态标志中断使能

47.3.9 移位器错误中断使能 (FLEXIO_SHIFTEIEN)

地址: 4005_F000h 基准 + 24h 偏移 = 4005_F024h



FLEXIO_SHIFTEIEN 字段描述

字段	描述
31-4 保留	此只读字段为保留字段且值始终为 0。
SEIE	移位器错误中断使能 当对应的 SEF 置位时，使能中断生成。 0 移位器错误标志中断已禁用 1 移位器错误标志中断使能

47.3.10 定时器中断使能寄存器 (FLEXIO_TIMIEN)

地址: 4005_F000h 基准 + 28h 偏移 = 4005_F028h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0																																TEIE
W																																	TEIE
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

FLEXIO_TIMIEN 字段描述

字段	描述
31–4 保留	此只读字段为保留字段且值始终为 0。
TEIE	<p>定时器状态中断使能</p> <p>当对应的 TSF 置位时，使能中断生成。</p> <p>0 定时器状态标志中断已禁用 1 定时器状态标志中断使能</p>

47.3.11 移位器状态 DMA 使能 (FLEXIO_SHIFTSDEN)

地址: 4005_F000h 基准 + 30h 偏移 = 4005_F030h

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0																																SSDE
W																																	SSDE
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

FLEXIO_SHIFTSDEN 字段描述

字段	描述
31–4 保留	此只读字段为保留字段且值始终为 0。
SSDE	<p>移位器状态 DMA 使能</p> <p>当对应的 SSF 置位时，使能 DMA 请求生成。</p> <p>0 移位器状态标志位 DMA 请求禁用 1 移位器状态标志位 DMA 请求使能</p>

47.3.12 移位器控制 N 寄存器 (FLEXIO_SHIFTCTLn)

Address: 4005_F000h base + 80h offset + (4d × i), where i=0d to 3d

	位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R		0						TIMSEL		TIMPOL	0						PINCFG	
W		保留						TIMSEL		TIMPOL	保留						PINCFG	
复位		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R		0					PINSEL			PINPOL	0				SMOD			
W		保留					PINSEL			PINPOL	保留				SMOD			
复位		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

FLEXIO_SHIFTCTLn 字段描述

字段	描述
31-26 保留	此只读字段为保留字段且值始终为 0。
25-24 TIMSEL	定时器选择 选择要用于控制逻辑/位移寄存器和生成移位时钟的定时器。

下一页继续介绍此表...

FLEXIO_SHIFTCTLn 字段描述 (继续)

字段	描述
23 TIMPOL	定时器极性 0 在移位时钟上升沿移位 1 在移位时钟下降沿移位
22-18 保留	此只读字段为保留字段且值始终为 0。
17-16 PINCFG	移位器引脚配置 00 移位器引脚输出禁用 01 移位器引脚开漏或双向输出使能 10 移位器引脚双向输出数据 11 移位器引脚输出
15-11 保留	此只读字段为保留字段且值始终为 0。
10-8 PINSEL	移位器引脚选择 选择移位器输入或输出要使用的引脚。
7 PINPOL	移位器引脚极性 0 引脚的有效电平高 1 引脚的有效电平低
6-3 保留	此只读字段为保留字段且值始终为 0。
SMOD	移位器模式 配置移位器的模式。 000 禁用。 001 接收模式。在定时器过期时捕捉当前的移位器内容并加载至 SHIFTBUF。 010 发送模式。在定时器过期时将 SHIFTBUF 内容加载至移位器。 011 保留。 100 匹配存储模式。在定时器过期时，比较移位器数据和 SHIFTBUF 内容。 101 匹配连续模式。连续比较移位器数据与 SHIFTBUF 内容。 110 保留。 111 保留。

47.3.13 移位器配置 N 寄存器 (FLEXIO_SHIFTCFGn)

Address: 4005_F000h base + 100h offset + (4d × i), where i=0d to 3d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0							
W	[Reserved]															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								INSRC	0	0	SSTOP	0	SSTART		
W	[Reserved]								INSRC	[Reserved]	[Reserved]	SSTOP	[Reserved]	SSTART		
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXIO_SHIFTCFGn 字段描述

字段	描述
31–21 保留	此只读字段为保留字段且值始终为 0。
20–16 保留	此只读字段为保留字段且值始终为 0。
15–9 保留	此只读字段为保留字段且值始终为 0。
8 INSRC	输入源 为移位器选择输入源。 0 引脚 1 移位器 N+1 输出

下一页继续介绍此表...

FLEXIO_SHIFTCFGn 字段描述 (继续)

字段	描述
7 保留	此只读字段为保留字段且值始终为 0。
6 保留	此只读字段为保留字段且值始终为 0。
5-4 SSTOP	<p>移位器停止位</p> <p>当 SMOD=发送时，如果选择的定时器已使能停止位，则此字段允许自动插入停止位。</p> <p>当 SMOD=接收或匹配存储时，如果选择的定时器已使能停止位，则此字段允许自动检查停止位。</p> <p>00 停止位禁止用于发送器/接收器/匹配存储</p> <p>01 保留用于发送器/接收器/匹配存储</p> <p>10 发送器在存储时输出停止位数值 0，接收器/匹配存储在停止位非 0 时置位错误标志位</p> <p>11 发送器在存储时输出停止位数值 1，接收器/匹配存储在停止位非 1 时置位错误标志位</p>
3-2 保留	此只读字段为保留字段且值始终为 0。
SSTART	<p>移位器起始位</p> <p>当 SMOD=发送时，如果选择的定时器已使能起始位，则此字段允许自动插入起始位。</p> <p>当 SMOD=接收或匹配存储时，如果选择的定时器已使能起始位，则此字段允许自动检查起始位。</p> <p>00 起始位禁止用于发送器/接收器/匹配存储，发送器在使能时加载数据</p> <p>01 起始位禁止用于发送器/接收器/匹配存储，发送器在首次移位时加载数据</p> <p>10 发送器在首次移位加载数据之前输出起始位数值 0，接收器/匹配存储在起始位非 0 时置位错误标志位</p> <p>11 发送器在首次移位加载数据之前输出起始位数值 1，接收器/匹配存储在起始位非 1 时置位错误标志位</p>

47.3.14 移位器缓冲器 N 寄存器 (FLEXIO_SHIFTBUF_n)

Address: 4005_F000h base + 200h offset + (4d × i), where i=0d to 3d

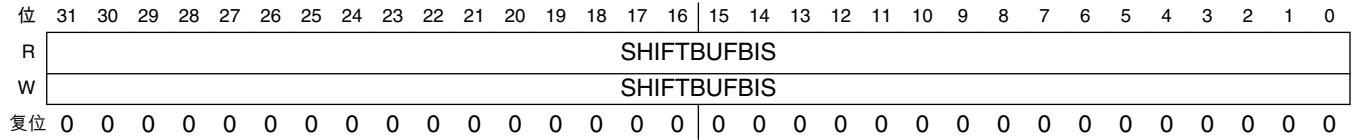
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SHIFTBUF																															
W	SHIFTBUF																															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXIO_SHIFTBUF_n 字段描述

字段	描述
SHIFTBUF	<p>移位缓冲器</p> <p>根据 SMOD 设置，移位缓冲器数据可用于多种功能：</p> <p>如果 SMOD=Receive，移位器数据将在定时器过期时传输至 SHIFTBUF。</p> <p>如果 SMOD=Transmit，SHIFTBUF 数据将在定时器开始之前传输至移位器。</p> <p>如果 SMOD=Match Store/Continuous，SHIFTBUF[31:16]将含有要与移位器内容匹配的数据。匹配要么采用连续检查方式（即匹配连接模式），要么在定时器过期时检查（匹配存储模式）。SHIFTBUF[15:0]可用于屏蔽匹配结果（1=屏蔽，0=无屏蔽）。在匹配存储模式中，只要发生匹配事件，移位器数据[31:16]写入 SHIFTBUF[31:16]。</p>

47.3.15 移位器缓冲器 N 位交换寄存器 (FLEXIO_SHIFTBUFBISn)

Address: 4005_F000h base + 280h offset + (4d × i), where i=0d to 3d

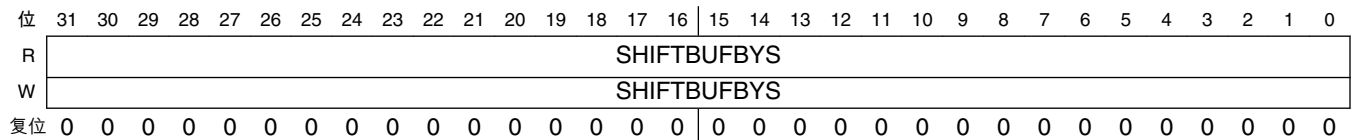


FLEXIO_SHIFTBUFBISn 字段描述

字段	描述
SHIFTBUFBIS	移位缓冲器 SHIFTBUF 寄存器的别名，除了对此寄存器的读/写均会交换位。读取返回 SHIFTBUF[0:31]。

47.3.16 移位器缓冲器 N 字节交换寄存器 (FLEXIO_SHIFTBUFBYSn)

Address: 4005_F000h base + 300h offset + (4d × i), where i=0d to 3d

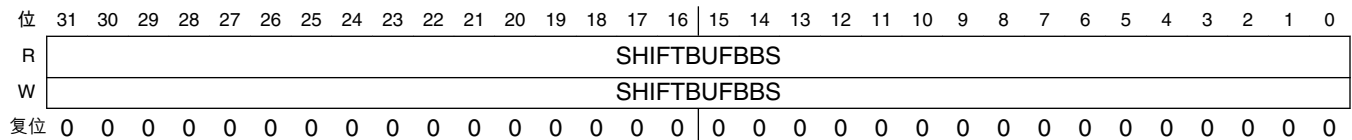


FLEXIO_SHIFTBUFBYSn 字段描述

字段	描述
SHIFTBUFBYS	移位缓冲器 SHIFTBUF 寄存器的别名，除了对此寄存器的读/写均会交换字节。读取返回{ SHIFTBUF[7:0]、SHIFTBUF[15:8]、SHIFTBUF[23:16]、SHIFTBUF[31:24]}。

47.3.17 移位器缓冲器 N 位字节交换寄存器 (FLEXIO_SHIFTBUFBBSn)

Address: 4005_F000h base + 380h offset + (4d × i), where i=0d to 3d



FLEXIO_SHIFTBUFBBSn 字段描述

字段	描述
SHIFTBUFBBS	移位缓冲器

FLEXIO_SHIFTBUFBBSn 字段描述 (继续)

字段	描述
	SHIFTBUF 寄存器的别名，除了对此寄存器的读/写均会在每个字节内交换位。读返回{ SHIFTBUF[24:31]、SHIFTBUF[16:23]、SHIFTBUF[8:15]、SHIFTBUF[0:7] }。

47.3.18 定时器控制 N 寄存器 (FLEXIO_TIMCTLn)

Address: 4005_F000h base + 400h offset + (4d × i), where i=0d to 3d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				TRGSEL				TRGPOL	TRGSRC	0				PINCFG	
W	0				TRGSEL				TRGPOL	TRGSRC	0				PINCFG	
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				PINSEL				PINPOL	0				TIMOD		
W	0				PINSEL				PINPOL	0				TIMOD		
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXIO_TIMCTLn 字段描述

字段	描述
31-28 保留	此只读字段为保留字段且值始终为 0。
27-24 TRGSEL	<p>触发器选择</p> <p>TRGSEL 的有效数值将取决于 FLEXIO_PARAM 寄存器。</p> <ul style="list-style-type: none"> 当 TRGSRC = 1 时, N 的有效值取决于 FLEXIO_PARAM 寄存器的引脚、TIMER、SHIFTER 字段。 当 TRGSRC = 0 时, N 的有效值取决于 FLEXIO_PARAM 寄存器的 TRIGGER 字段。 <p>有关外部触发选项, 请参见芯片配置章节。</p> <p>内部触发器选项配置如下:</p> <p>{N,00} 引脚 2N 输入 {N,01} 移位器 N 状态标志位 {N,10} 引脚 2N+1 输入 {N,11} 定时器 N 触发器输出</p>
23 TRGPOL	<p>触发器极性</p> <p>0 触发器的有效电平高 1 触发器的有效电平低</p>
22 TRGSRC	<p>触发源</p> <p>0 已选择外部触发器 1 已选择内部触发器</p>
21-18 保留	此只读字段为保留字段且值始终为 0。
17-16 PINCFG	<p>定时器引脚配置</p> <p>00 定时器引脚输出禁用 01 定时器引脚开漏或双向输出使能 10 定时器引脚双向输出数据 11 定时器引脚输出</p>
15-11 保留	此只读字段为保留字段且值始终为 0。
10-8 PINSEL	<p>定时器引脚选择</p> <p>选择定时器输入或输出要使用的引脚。</p>
7 PINPOL	<p>定时器引脚极性</p> <p>0 引脚的有效电平高 1 引脚的有效电平低</p>
6-2 保留	此只读字段为保留字段且值始终为 0。
TIMOD	<p>定时器模式</p> <p>在 8 位计数器模式中, 计数器和比较寄存器的低 8 位用于配置定时器移位时钟的波特率, 高 8 位用于配置移位器位计数。</p> <p>在 8 位 PWM 模式中, 计数器和比较寄存器的低 8 位用于配置定时器移位时钟的高时段, 高 8 位用于配置定时器移位时钟的低时段。利用另一个定时器或外部信号来配置移位器位计数。</p> <p>在 16 位计数器模式中, 计数器和比较寄存器的全部 16 位用于配置移位时钟或移位器位计数的波特率。</p>

下一页继续介绍此表...

FLEXIO_TIMCTLn 字段描述 (继续)

字段	描述
00	定时器禁用。
01	双 8 位计数器波特/位模式。
10	双 8 位计数器 PWM 模式。
11	单 16 位计数器模式。

47.3.19 定时器配置 N 寄存器 (FLEXIO_TIMCFGn)

当 N 被 4 整除 (例如定时器 0) 时, 保留利用定时器 N-1 使能或禁用来使能或禁用定时器的选项。

Address: 4005_F000h base + 480h offset + (4d × i), where i=0d to 3d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						TIMOUT	0	TIMDEC	0	TIMRST					
W	保留						TIMOUT	保留	TIMDEC	保留	TIMRST					
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TIMDIS			0	TIMENA			0	TSTOP			0	TSTART		0
W	保留	TIMDIS			保留	TIMENA			保留	TSTOP			保留	TSTART		保留
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXIO_TIMCFGn 字段描述

字段	描述
31-26 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

FLEXIO_TIMCFGn 字段描述 (继续)

字段	描述
25-24 TIMOUT	<p>定时器输出</p> <p>配置定时器输出的初始状态，以及是否受定时器复位影响。</p> <p>00 使能时，定时器输出为逻辑 1，且不受定时器复位影响</p> <p>01 使能时，定时器输出为逻辑 0，且不受定时器复位影响</p> <p>10 使能时以及定时器复位时，定时器输出为逻辑 1</p> <p>11 使能时以及定时器复位时，定时器输出为逻辑 0</p>
23-22 保留	此只读字段为保留字段且值始终为 0。
21-20 TIMDEC	<p>定时器递减</p> <p>配置定时器递减源和移位时钟源</p> <p>00 FlexIO 时钟上的递减计数器，移位时钟等于定时器输出。</p> <p>01 触发器输入（两个边沿）的递减计数器，移位时钟等于定时器输出。</p> <p>10 引脚输入（两个边沿）的递减计数器，移位时钟等于引脚输入。</p> <p>11 触发器输入（两个边沿）的递减计数器，移位时钟等于触发器输入。</p>
19 保留	此只读字段为保留字段且值始终为 0。
18-16 TMRST	<p>定时器复位</p> <p>配置将使得定时器计数器（以及可选的定时器输出）复位的条件。在 8 位计数器模式中，定时器复位只会复位配置波特率的低 8 位。在所有其他模式中，定时器复位将复位计数器的全部 16 位。</p> <p>000 定时器永不复位</p> <p>001 保留</p> <p>010 定时器在定时器引脚等于定时器输出时复位</p> <p>011 定时器在定时器触发器等于定时器输出时复位</p> <p>100 定时器在定时器引脚的上升沿复位</p> <p>101 保留</p> <p>110 定时器在触发器的上升沿复位</p> <p>111 定时器在触发器的上升沿或下降沿复位</p>
15 保留	此只读字段为保留字段且值始终为 0。
14-12 TIMDIS	<p>定时器禁用</p> <p>配置条件，使得定时器禁用和停止递减。</p> <p>000 定时器永不禁用</p> <p>001 定时器在定时器 N-1 禁用时禁用</p> <p>010 定时器在定时器比较时禁用</p> <p>011 定时器在定时器比较和触发器低电平时禁用</p> <p>100 定时器在引脚上升沿或下降沿禁用</p> <p>101 假设触发器为高电平，则定时器在引脚上升沿或下降沿禁用</p> <p>110 定时器在触发器下降沿禁用</p> <p>111 保留</p>
11 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

FLEXIO_TIMCFGn 字段描述 (继续)

字段	描述
10–8 TIMENA	<p>定时器使能</p> <p>配置条件，使得定时器使能和开始递减。</p> <p>000 定时器始终使能</p> <p>001 定时器在定时器 N-1 使能时使能</p> <p>010 定时器在触发器高电平时使能</p> <p>011 定时器在触发器高电平和引脚高电平时使能</p> <p>100 定时器在引脚上升沿使能</p> <p>101 定时器在引脚上升沿和触发器高电平时使能</p> <p>110 定时器在触发器上升沿使能</p> <p>111 定时器在触发器上升沿或下降沿使能</p>
7–6 保留	此只读字段为保留字段且值始终为 0。
5–4 TSTOP	<p>定时器停止位</p> <p>停止位可在定时器比较（每个字之间）或定时器禁用时添加。当停止位使能时，配置的移位器会在定时器禁用时输出停止位的内容。如果停止位在定时器禁用时使能，则直到移位时钟的下一个上升沿之前，定时器都保持禁用。如果配置用于定时器比较和定时器禁用，则仅会在定时器禁用时插入一个停止位。</p> <p>00 停止位禁用</p> <p>01 停止位在定时器比较时使能</p> <p>10 停止位在定时器禁用时使能</p> <p>11 停止位在定时器比较和定时器禁用时使能</p>
3–2 保留	此只读字段为保留字段且值始终为 0。
1 TSTART	<p>定时器起始位</p> <p>当起始位使能时，配置的移位器会在定时器使能时输出起始位的内容，而定时器计数器会在移位时钟的首个上升沿从比较基础器重新加载。</p> <p>0 起始位禁用。</p> <p>1 起始位使能。</p>
0 保留	此只读字段为保留字段且值始终为 0。

47.3.20 定时器比较 N 寄存器 (FLEXIO_TIMCMPn)

Address: 4005_F000h base + 500h offset + (4d × i), where i=0d to 3d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CMP															
W																	CMP															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXIO_TIMCMPn 字段描述

字段	描述
31-16 保留	此只读字段为保留字段且值始终为 0。
CMP	<p>定时器比较值</p> <p>定时器比较数值在定时器首次使能、定时器复位以及定时器递减至 0 时被加载至定时器计数器。在双 8 位计数器波特/位模式下,低 8 位将波特率分频器配置为等于 $(CMP[7:0] + 1) * 2$。高 8 位将每个字中的位数配置为等于 $(CMP[15:8] + 1) / 2$。在双 8 位计数器 PWM 模式下,低 8 位将输出的高时段配置为 $(CMP[7:0] + 1)$,高 8 位将输出的低时段配置为 $(CMP[15:8] + 1)$。在 16 位计数器模式中,比较值可用于生成波特率分频器(如果移位时钟源是定时器输出),使其等于 $(CMP[15:0] + 1) * 2$。当移位时钟源是引脚或触发器输入时,比较寄存器用于设置每个字中的位数,等于 $(CMP[15:0] + 1) / 2$。</p>

47.4 功能说明

47.4.1 移位器操作

移位器负责缓冲并把数据移进或移出 FlexIO。移位、加载和存储事件时序由分配给该移位器的定时器通过 SHIFTCTL[TIMSEL]寄存器控制。移位器可以支持 DMA、中断或轮询操作。以下框图显示了移位器微架构的详细情况。

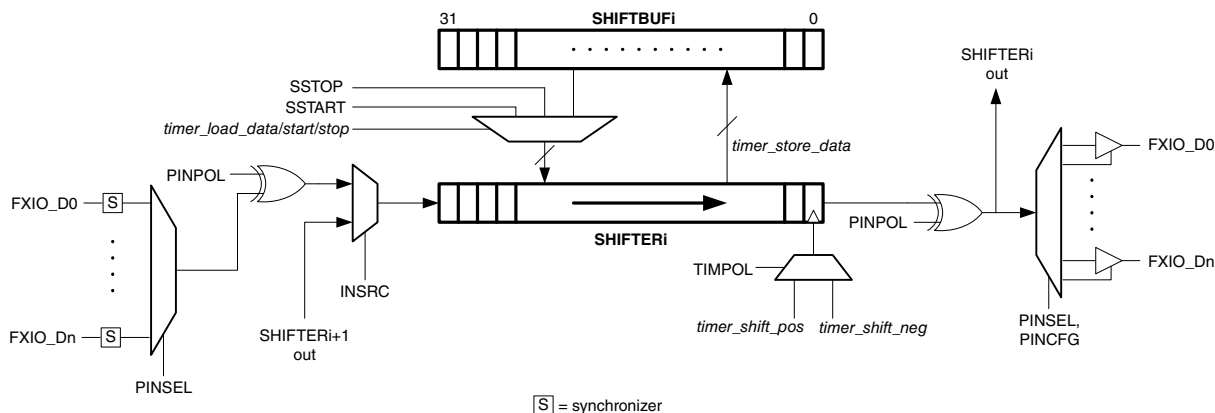


图 47-58. 移位器微架构

47.4.1.1 发送模式

针对发送模式配置 (SHIFTCTL[SMOD]=Transmit) 时,移位器将从 SHIFTBUF 寄存器加载数据,并在分配的定时器显示发生加载事件时移出数据。通过配置移位器或定时器中的 SHIFTCFG[SSTART]、TIMCFG[TSTART]或 SHIFTCFG[SSTOP]、

TIMCFG[TSTOP]寄存器，也可以在 SHIFTBUF 数据前/后自动加载可选的启动/停止位。请注意，当移位器初始配置为发送模式时，如果使能了一个停止位，则移位器会立即加载停止位。

当数据已经从 SHIFTBUF 寄存器载入移位器或者当移位器初始配置为发送模式时，移位器状态标志 (SHIFTSTAT[SSF]) 和任意已经使能的中断或 DMA 请求将置位。当新数据被写入 SHIFTBUF 寄存器时，该标志会清零。

当尝试从空的 SHIFTBUF 寄存器加载数据时 (缓冲器下溢)，移位器错误标志 (SHIFTErr[SEF]) 和使能的任意中断将置位。将对应的错误位置 1 将清除此位。

47.4.1.2 接收模式

当针对接收模式配置 (SHIFTCTL[SMOD]=Receive) 时，如果分配的定时器显示存在存储事件，移位器会移入数据并把数据存入 SHIFTBUF 寄存器。通过配置移位器和定时器中的 SHIFTCFG[SSTART]、TIMCFG[TSTART]或 SHIFTCFG[SSTOP]、TIMCFG[TSTOP]寄存器，可以在对移位器数据采样之前/之后使能开始/停止位检查。

当数据已经从移位器存入 SHIFTBUF 寄存器时，移位器状态标志 (SHIFTSTAT[SSF]) 和使能的任意中断或 DMA 请求标志位会置位。当已经完成从 SHIFTBUF 寄存器读取数据时，该标志会清零。

当试图把数据存进已经满的 SHIFTBUF 寄存器 (缓冲器溢出) 时，或者当开始/停止位检查发生错误时，移位器错误标志 (SHIFTErr[SEF]) 和使能的任何中断都会置位。将对应的错误位置 1 将清除此位。

47.4.1.3 匹配存储模式

当处于匹配存储模式 (SHIFTCTL[SMOD]=Match Store) 时，如果分配的定时器产生了一个存储触发事件，移位器会移入数据，检查是否存在匹配结果，并把匹配数据存入 SHIFTBUF 寄存器。通过配置移位器和定时器中的 SHIFTCFG[SSTART]、TIMCFG[TSTART]或 SHIFTCFG[SSTOP]、TIMCFG[TSTOP]寄存器，可以在对移位器数据采样之前/之后使能开始/停止位检查。最多可以比较 16 位数据，用 SHIFTBUF[31:16]配置待匹配的数据，用 SHIFTBUF[15:0]可以屏蔽匹配的结果。

当发生匹配并且匹配数据已经从移位器存入 SHIFTBUF 寄存器时，移位器状态标志 (SHIFTSTAT[SSF]) 和使能的任意中断或 DMA 请求标志位会置位。当已经完成从 SHIFTBUF 寄存器读取匹配数据时，该标志会清零。

当试图把匹配数据存进已经满的 SHIFTBUF 寄存器 (缓冲器溢出) 时，或者当开始/停止位检查发生错误时，移位器错误标志 (SHIFTErr[SEF]) 和使能的任何中断都会置位。将对应的错误位置 1 将清除此位。

47.4.1.4 匹配连续模式

当处于连续匹配模式($\text{SHIFTCTL}[\text{SMOD}] = \text{Match Continuous}$)时，每当分配的定时器发生移位事件时，移位器会移入数据，并连续检查是否存在匹配结果。最多可以比较 16 位数据，用 $\text{SHIFTBUF}[31:16]$ 配置待匹配的数据，用 $\text{SHIFTBUF}[15:0]$ 可以屏蔽匹配结果。

当发生匹配时，移位器状态标志($\text{SHIFTSTAT}[\text{SSF}]$)和使能的任意中断或 DMA 请求标志位会置位。只要移位器数据与 SHIFTBUF 寄存器之间不再匹配，此标志就会自动清零。

当发生匹配时，移位器错误标志($\text{SHIFTErr}[\text{SEF}]$)和使能的任意中断会置位。当有来自 SHIFTBUF 寄存器的读取操作时，或者当该寄存器中将对应的错误位置 1 将清除此位。

47.4.2 定时器操作

FlexIO 16 位定时器控制位移寄存器的加载、移位和存储，计数器加载比较寄存器的内容并在每个 FLEXIO 时钟周期上递减，直到减小到 0。它们可以执行通用定时器功能，例如生成一个时钟、选择输出或者 PWM 波形。定时器可以配置为在一定条件下使能，以响应触发器、引脚或移位器条件；始终递减或仅在触发器或引脚边沿递减；响应触发器或引脚条件复位；在触发器或引脚条件下或在出现定时器比较时禁用。定时器可选择性的包括一个开始条件“与”或者“或”停止条件。

每个定时器都可以独立运行，但也可以配置为与之前的定时器同时使能或者禁用，例如，定时器 1 可以与定时器 0 同时使能或者禁用。或者，一个定时器的输出可以用于触发任何其他的定时器使能或者禁止。每个定时器使用的触发器都独立配置，且可配置为定时器输出、移位器状态标识位、引脚输入或外部触发输入（请参考芯片配置章节了解外部触发器连接的详情）。触发器配置与引脚配置相互独立，且可配置用于输入、输出数据或输出使能。

应在置位定时器模式 (TIMOD) 之前配置定时器配置寄存器 (TIMCFGn)。当 TIMOD 配置用于所需的模式，在检测到定时器使能 (TIMENA) 配置的条件时，则发生以下事件。

- 定时器计数器将加载比较寄存器的当前数值，并按照 TIMDEC 的配置递减。
- 定时器输出将根据 TIMOUT 配置进行置位。
- 该定时器控制的发送移位器将根据 SSTART 的配置，输出其起始位值，或者加载移位缓冲区的移位寄存器并输出首个位。

之后定时器将生成定时器输出和定时器移位时钟，这由 TIMOD 和 TIMDEC 字段决定。移位时钟等于定时器输出（当 TIMDEC=00 或 01）或等于递减时钟（当 TIMDEC=10 或 11）。当 TIMDEC 配置为从引脚或触发器递减时，定时器在上升和下降沿上都将递减。

如果定时器按照 TIMRST 字段的配置被配置为复位时，则定时器计数器将再次加载比较寄存器的当前数值，而定时器输出则如同 TIMEOUT 的配置受到复位的影响。

如果定时器的起始位使能，则在定时器开始递减之后，定时器计数器会在移位时钟的第一个上升沿重新加载比较寄存器。如果在第一个上升沿之前，移位时钟不具备下降沿（例如，当 TIMEOUT=1）时，则配置为在下降沿移位、在第一个移位加载的移位器将不会正确加载。

当配置用于 8 位计数器模式时，只要低 8 位计数器递减到 0，定时器输出会切换，低 8 位计数器寄存器将从比较寄存器重新加载，而高 8 位计数器将递减。对于 8 位 PWM 模式，低 8 位计数器只有在输出高电平时才会递减，而高 8 位计数器只有在输出低电平时才会递减。只要低或高 8 位计数器递减到 0，定时器输出会切换。

当定时器递减到 0 时，发生比较事件，具体取决于定时器的模式。对于 8 位计数器或 PWM 模式，计数器的两个部分都必须等于 0，且上半部分必须递减，以便发生定时器比较事件，在 16 位模式下则整个计数器必须等于 0 且递减。定时器比较事件将导致定时器状态标志位置位，定时器计数器加载定时器比较寄存器的内容，定时器输出切换，已配置的任何发送移位寄存器加载以及已配置的任何接收移位寄存器开始存储。

当定时器配置为在每次比较增加一个停止位时，则发生以下附加事件。

- 该定时器控制的发送移位器将输出其停止位值（如果已通过 SSTOP 配置）。
- 该定时器控制的接收移位器将根据 SSTOP 的配置，在其移位缓冲器中存储移位寄存器的内容。
- 比较之后，在移位时钟的首个上升沿，定时器计数器将重新加载比较寄存器的当前值。

当定时器配置为每次比较插入一个停止位时，则必须将发送移位器配置为在第一个移位时加载。

检测到由定时器禁用配置的条件（TIMDIS）时，会发生以下事件。

- 定时器计数器将重新加载比较寄存器的当前数值，并按照 TIMDEC 的配置递减。
- 定时器输出将被清除。
- 该定时器控制的发送移位器将输出其停止位值（如果已通过 SSTOP 配置）。
- 该定时器控制的接收移位器将根据 SSTOP 的配置，在其移位缓冲器中存储移位寄存器的内容。

如果定时器停止位已使能，则定时器计数器将继续递减，直到检测到移位时钟的下一个上升沿，而此时定时器计数器将终止。可在与定时器禁用条件相同的周期中检测到定时器使能条件（如果定时器停止位已禁用），或在禁用条件后移位时钟的首个上升沿上检测到（如果停止位已使能）。当定时器处于停止状态下时，接收移位寄存器的停止位使能会将移位寄存器的内容存储到移位缓冲器中，并验证配置的移位边沿上输入数据的状态。如果定时器禁用和移位时钟的下一个上升沿之间没有配置边沿，则不会发生最终的存储和校验。

47.4.3 引脚操作

各定时器和移位器的引脚可以配置为使用任意极性的任何 FlexIO 引脚。各定时器和移位器均可配置为输入、输出数据、输出使能或双向输出。配置为输出使能的引脚可以用作开漏（反相极性，因为输出使能置位会导致该引脚上输出逻辑零），或者用于控制双向输出上的使能。可以配置任何定时器或移位器来控制引脚的输出使能，这种情况下，双向输出数据由另一个定时器或移位器驱动。

在将引脚配置为输入时（其中包括一个配置为引脚输入的定时器触发器），输入信号先同步到 FlexIO 时钟，然后，该信号由定时器或移位器使用。在使用外部引脚输入来产生输出或控制移位器时，这会带来较小的延迟，范围在 0.5 至 1.5 个 FlexIO 时钟周期之间。这会使最大建立时间设为 1.5 个 FlexIO 时钟周期。

如果一个输入由一个以上的定时器或移位器使用时，则会发生一次同步，以确保使用该输入的所有定时器和移位器都能在同一周期看到任何边沿。

注意，FlexIO 引脚也采用内部连接方式，如果将 FlexIO 移位器或定时器配置为在未使用的引脚上输出数据，则会形成一个内部连接，允许其他移位器和定时器将该引脚用作输入。这样允许将移位器输出用来触发定时器，或者将定时器输出移入移位器。该路径也同步到 FlexIO 时钟，因此会导致 1 个周期的延迟。

因此，在将一个引脚输入用作定时器触发器、定时器时钟或移位器数据输入时，会发生以下同步延迟：

1. 外部引脚为 0.5 – 1.5 个 FlexIO 时钟周期
2. 内部驱动引脚为 1 个 FlexIO 时钟周期

有关针对特定应用的输出有效时间和输入建立时间等时序考虑因素（SPI 主机、SPI 从机、I2C 主机、I2S 主机、I2S 从机），请参见“FlexIO 应用信息”一节。

47.5 应用信息

本节提供各种 FlexIO 模块应用的示例。

47.5.1 UART 发送

用一个定时器、一个移位器和一个引脚（如果支持 CTS 则要两个引脚）即可支持 UART 发送。开始和停止位插入采用自动处理方式，用 DMA 控制器可以支持多路传输。定时器状态标志可以用于指示发送各个字的停止位的时间。

断点和空闲字符要求软件实现，在发送断点或空闲字符之后，须更改 SSTART 和 SSTOP 字段以发送所需状态，并且要发送的数据必须等于 0xFF 或 0x00。要支持 2 位停止位，就要求用软件把停止位插入数据流（并增加要发送的位数）。请注意，在执行对 SHIFTBUF_n 的写入操作时（或者，如果要先发送 MSB，则是 SHIFTBUFBIS），寄存器的其余部分将保持不变，从而使地址标记位或其他停止位不受影响。

FlexIO 不支持校验位的自动插入。

表 47-61. UART 发送配置

寄存器	值	备注
SHIFTCFG _n	0x0000_0032	将起始位配置为 0，停止位配置为 1。
SHIFTCTL _n	0x0003_0002	在上升沿用定时器 0 配置发送，同时在引脚 0 上输出数据。通过置位 PINPOL 可以使输出数据反相，或者通过置位 PINPOL=0x1 和 PINCFG=0x1，也可支持开漏。
TIMCMP _n	0x0000_0F01	配置 8 位传输，波特率为 FlexIO 时钟/4。使 TIMCMP[15:8] = (number of bits x 2) - 1。使 TIMCMP[7:0] = (baud rate divider / 2) - 1。
TIMCFG _n	0x0000_2222	配置起始位、停止位、触发器低电平使能、比较禁用。使 TIMEN=0x3，即可支持 CTS。
TIMCTL _n	0x01C0_0001	以移位器 0 状态标志作为反相内部触发器源，配置双 8 位计数器。使 PINSEL=0x1（引脚 1）且 PINPOL=0x1，则可支持 CTS。
SHIFTBUF _n	要发送的数据	可将发送数据写入 SHIFTBUF[7:0]，以启动一次 8 位传输，用移位器状态标志指示什么时候可以利用中断或 DMA 请求来写数据。通过写入 SHIFTBUFBBS[7:0] 寄存器，则可支持 MSB 首次传输。

47.5.2 UART 接收

用一个定时器、一个移位器和一个引脚（如果支持 RTS 则需要两个定时器和两个引脚）即可支持 UART 接收。开始和停止位验证采用自动处理方式，用 DMA 控制器可以支持多路传输。定时器状态标志可以用于指示接收各个字停止位的时间。

FlexIO 不支持对接收到的数据进行三重采样，数据只会各位中间采样一次。另一个定时器可用于对输入数据进行毛刺滤波，也可用于一个用户可编程长度的空闲状态。分隔字符会导致错误标志置位，移位器缓冲器寄存器将返回 0x00。

FlexIO 不支持自动验证校验位。

表 47-62. UART 接收器配置

寄存器	值	备注
SHIFTCFGn	0x0000_0032	将起始位配置为 0，停止位配置为 1。
SHIFTCTLn	0x0080_0001	在时钟下降沿用定时器 0 配置接收，其中，在引脚 0 输入数据。通过置位 PINPOL 可以使输入数据反相。
TIMCMPn	0x0000_0F01	配置 8 位传输，波特率为 FlexIO 时钟/4。使 TIMCMP[15:8] = (number of bits x 2) - 1。使 TIMCMP[7:0] = (baud rate divider / 2) - 1。
TIMCFGn	0x0204_2422	配置起始位、停止位、引脚上升沿使能、比较禁用。使能与接收的数据再同步，其中，TIMOUT=0x2 且 TIMRST=0x4。
TIMCTLn	0x0000_0081	用反相引脚 0 输入，配置双 8 位计数器。
SHIFTBUFn	要接收的数据	可以从 SHIFTBUFBYS[7:0] 读取接收的数据，移位器状态标志用于指示何时可以使用中断或 DMA 请求读取数据。通过从 SHIFTBUFBYS[7:0] 寄存器读取，可以支持 MSB 优先传输。

采用 RTS 配置的 UART 接收器使用第二个定时器来生成 RTS 输出。从移位缓冲器寄存器中读取数据时，如果检测到起始位，RTS 将置位，如果数据从移位寄存器中被读走，RTS 将被清 0。在 RTS 被置位时，不会检测到起始位，只会简单忽略接收到的数据。

表 47-63. 采用 RTS 配置的 UART 接收器

寄存器	值	备注
SHIFTCFGn	0x0000_0032	将起始位配置为 0，停止位配置为 1。
SHIFTCTLn	0x0080_0001	在时钟下降沿用定时器 0 配置接收，其中，在引脚 0 输入数据。通过置位 PINPOL 可以使输入数据反相。
TIMCMPn	0x0000_0F01	配置 8 位传输，波特率为 FlexIO 时钟/4。使 TIMCMP[15:8] = (number of bits x 2) -

下一页继续介绍此表...

表 47-63. 采用 RTS 配置的 UART 接收器 (继续)

寄存器	值	备注
		1. 使 TIMCMP[7:0] = (baud rate divider / 2) - 1。
TIMCFGn	0x0204_2522	配置起始位、停止位、在触发器处于低位时引脚上升沿使能、比较禁用。使能与接收的数据再同步，其中，TIMOUT=0x2 且 TIMRST=0x4。
TIMCTLn	0x03C0_0081	用反相引脚 0 输入，配置双 8 位计数器。触发器位于内部，使用反向引脚 1 输入。
TIMCMP(n+1)	0x0000_FFFF	从不比较。
TIMCFG(n+1)	0x0030_6100	在定时器 N 使能时使能，在触发器下降沿禁用。触发器递减，以确保不会比较。
TIMCTL(n+1)	0x0143_0083	配置 16 位计数器和引脚 1 上的输出。触发器位于内部，使用移位器 0 标志位。
SHIFTBUFn	要接收的数据	可以从 SHIFTBUFBS[7:0] 读取接收的数据，移位器状态标志用于指示何时可以使用中断或 DMA 请求读取数据。通过从 SHIFTBUFBS[7:0] 寄存器读取，可以支持 MSB 优先传输。

47.5.3 SPI 主机

可使用两个定时器、两个移位器和四个引脚支持 SPI 主机模式。支持 CPHA=0 或 CPHA=1，用 DMA 控制器还可支持传输。对于 CPHA=1，片选可以保持置位以便进行多次传输，定时器状态标志可以用作传输结束标志。

停止位用于确保在从机选择失效与下次传输之间至少有 1 个时钟周期。通过内核或 DMA 向传送缓冲器进行的写入操作用于启动每次传输。

受同步延迟影响，串行输入数据的建立时间为 1.5 个 FlexIO 时钟周期，因此，最大波特率为 FlexIO 时钟频率/4。

表 47-64. SPI 主机(CPHA=0)配置

寄存器	值	备注
SHIFTCFGn	0x0000_0000	起始和停止位禁用。
SHIFTCTLn	0x0083_0002	在时钟下降沿用定时器 0 配置传送，其中，数据在引脚 0 上输出。
SHIFTCFG(n+1)	0x0000_0000	起始和停止位禁用。
SHIFTCTL(n+1)	0x0000_0101	在时钟上升沿用定时器 0 配置接收，其中，数据在引脚 1 上输入。
TIMCMPn	0x0000_3F01	配置 32 位传输，波特率为 FlexIO 时钟/4。使 TIMCMP[15:8] = (位数 x 2) - 1。使 TIMCMP[7:0] = (波特率分频器/2) - 1。

下一页继续介绍此表...

表 47-64. SPI 主机(CPHA=0)配置 (继续)

寄存器	值	备注
TIMCFGn	0x0100_2222	配置开始位、停止位、触发器高电平时使能、比较时禁用, 初始时钟状态为逻辑 0。置位 PINPOL, 使输出移位时钟反相。
TIMCTLn	0x01C3_0201	用引脚 2 输出 (移位时钟) 配置双 8 位计数器, 移位器 0 标志作为反向触发器。
TIMCMP(n+1)	0x0000_FFFF	从不比较。
TIMCFG(n+1)	0x0000_1100	定时器 0 使能时使能, 定时器 0 禁用时禁用。
TIMCTL(n+1)	0x0003_0383	用反相引脚 3 输出 (作为从机选择) 配置 16 位计数器 (从不比较)。
SHIFTBUFn	要传送的数据	可将传送数据写入 SHIFTBUF, 用移位器状态标志指示什么时候可以利用中断或 DMA 请求来写入数据。通过写入 SHIFTBUFBBS 寄存器, 则可支持 MSB 首次传输。
SHIFTBUF(n+1)	要接收的数据	可以从 SHIFTBUFBYS 读取接收的数据, 移位器状态标志用于指示何时可以使用中断或 DMA 请求读取数据。通过从 SHIFTBUFBYS 寄存器读取, 可以支持 MSB 首次传输。

表 47-65. SPI 主机(CPHA=1)配置

寄存器	值	备注
SHIFTCFGn	0x0000_0021	开始位在首次移位时加载数据。
SHIFTCTLn	0x0003_0002	在时钟上升沿用定时器 0 配置传送, 其中, 数据在引脚 0 上输出。
SHIFTCFG(n+1)	0x0000_0000	起始和停止位禁用。
SHIFTCTL(n+1)	0x0080_0101	在时钟下降沿用定时器 0 配置接收, 其中, 数据在引脚 1 上输入。
TIMCMPn	0x0000_3F01	配置 32 位传输, 波特率为 FlexIO 时钟/4。使 TIMCMP[15:8] = (位数 x 2) - 1。使 TIMCMP[7:0] = (波特率分频器/2) - 1。
TIMCFGn	0x0100_2222	配置开始位、停止位、触发器高电平时使能、比较时禁用, 初始时钟状态为逻辑 0。置位 PINPOL, 使输出移位时钟反相。使 TIMDIS=3, 如此, 只要传送缓冲器中有数据, 从机选择就能保持置位。
TIMCTLn	0x01C3_0201	用引脚 2 输出 (移位时钟) 配置双 8 位计数器, 移位器 0 标志作为反向触发器。
TIMCMP(n+1)	0x0000_FFFF	从不比较。
TIMCFG(n+1)	0x0000_1100	定时器 0 使能时使能, 定时器 0 禁用时禁用。
TIMCTL(n+1)	0x0003_0383	用反相引脚 3 输出 (作为从机选择) 配置 16 位计数器 (从不比较)。
SHIFTBUFn	要传送的数据	可将传送数据写入 SHIFTBUF, 用移位器状态标志指示什么时候可以利用中断或

下一页继续介绍此表...

表 47-65. SPI 主机(CPHA=1)配置 (继续)

寄存器	值	备注
		DMA 请求来写入数据。通过写入 SHIFTBUFBBS 寄存器, 则可支持 MSB 首次传输。
SHIFTBUF(n+1)	要接收的数据	可以从 SHIFTBUFBYS 读取接收的数据, 移位器状态标志用于指示何时可以使用中断或 DMA 请求读取数据。通过从 SHIFTBUFBYS 寄存器读取, 可以支持 MSB 首次传输。

47.5.4 SPI 从机

可使用一个定时器、两个移位器和四个引脚支持 SPI 从机模式。支持 CPHA=0 或 CPHA=1, 用 DMA 控制器还可支持传输。对于 CPHA=1, 片选可以保持置位以便进行多次传输, 定时器状态标志可以用作传输结束标志。

发送数据必须在外部从机选择置位之前写入发送缓冲器寄存器中, 否则移位器错误标志将置位。

受同步延迟影响, 串行输出数据的输出有效时间为 2.5 个 FlexIO 时钟周期, 因此, 最大波特率为 FlexIO 时钟频率/6。

表 47-66. SPI 从机(CPHA=0)配置

寄存器	值	备注
SHIFTCFGn	0x0000_0000	起始和停止位禁用。
SHIFTCTLn	0x0083_0002	在移位时钟下降沿用定时器 0 配置发送, 其中, 数据在引脚 0 上输出。
SHIFTCFG(n+1)	0x0000_0000	起始和停止位禁用。
SHIFTCTL(n+1)	0x0000_0101	在移位时钟上升沿用定时器 0 配置接收, 其中, 数据在引脚 1 上输入。
TIMCMPn	0x0000_003F	配置 32 位传输。使 TIMCMP[15:0] = (位数 x 2) - 1。
TIMCFGn	0x0120_6000	配置触发器上升沿使能, 初始时钟状态为逻辑 0, 并在引脚输入上递减。
TIMCTLn	0x06C0_0203	用引脚 2 输入 (移位时钟) 配置 16 位计数器, 以引脚 3 输入 (从机选择) 为反相触发器。
SHIFTBUFn	要传送的数据	可将传送数据写入 SHIFTBUF, 用移位器状态标志指示什么时候可以利用中断或 DMA 请求来写入数据。通过写入 SHIFTBUFBBS 寄存器, 则可支持 MSB 首次传输。
SHIFTBUF(n+1)	要接收的数据	可以从 SHIFTBUFBYS 读取接收的数据, 移位器状态标志用于指示何时可以使用中

表 47-66. SPI 从机(CPHA=0)配置

寄存器	值	备注
		断或 DMA 请求读取数据。通过从 SHIFTBUFBYS 寄存器读取, 可以支持 MSB 首次传输。

表 47-67. SPI 从机(CPHA=1)配置

寄存器	值	备注
SHIFTCFGn	0x0000_0001	将移位器配置为在首次移位和停止位禁用时加载。
SHIFTCTLn	0x0003_0002	在移位时钟上升沿用定时器 0 配置发送, 其中, 数据在引脚 0 上输出。
SHIFTCFG(n+1)	0x0000_0000	起始和停止位禁用。
SHIFTCTL(n+1)	0x0080_0101	在移位时钟下降沿用定时器 0 配置接收, 其中, 数据在引脚 1 上输入。
TIMCMPn	0x0000_003F	配置 32 位传输。使 TIMCMP[15:0] = (位数 x 2) - 1。
TIMCFGn	0x0120_6602	配置起始位、触发器上升沿使能、触发器下降沿禁用, 初始时钟状态为逻辑 0, 并在引脚输入上递减。
TIMCTLn	0x06C0_0203	用引脚 2 输入 (移位时钟) 配置 16 位计数器, 以引脚 3 输入 (从机选择) 为反相触发器。
SHIFTBUFn	要传送的数据	可将传送数据写入 SHIFTBUF, 用移位器状态标志指示什么时候可以利用中断或 DMA 请求来写入数据。通过写入 SHIFTBUFBBS 寄存器, 则可支持 MSB 首次传输。
SHIFTBUF(n+1)	要接收的数据	可以从 SHIFTBUFBYS 读取接收的数据, 移位器状态标志用于指示何时可以使用中断或 DMA 请求读取数据。通过从 SHIFTBUFBYS 寄存器读取, 可以支持 MSB 首次传输。

47.5.5 I2C 主机

可使用两个定时器、两个移位器和两个引脚支持 I2C 主机模式。一个定时器用于生成 SCL 输出, 一个定时器用于控制移位器。在接收发送器必须发送 0xFF 以三态输出时, 两个移位器用于每个字的发送和接收。FlexIO 将在每个字之后插入一个停止位, 以生成/验证 ACK/NACK。FlexIO 等待第一次写入到发送数据缓冲区, 然后才会使能 SCL 的生成。可使用 DMA 控制器支持数据传送, 发送下溢或接收溢出时, 移位器错误标志会置位。

第一个定时器生成适用于整个数据包的位时钟 (START 至重复的 START/STOP), 因此需采用数据包的时钟边沿总数来设置比较寄存器 (减去 1)。当引脚等于输出时, 定制器支持利用复位计数器实现时钟延展 (尽管这会导致时钟高时段和时钟低时段分别最少增加一个 FlexIO 时钟周期)。第二个定时器利用 SCL 输入引脚来控制发送/接收移位寄存器, 这样会导致需要额外两个 FlexIO 时钟周期的 SDA 数据保持时间。

为了传输每一个字, 需要对发送和接收移位器提供相关服务, 在接收时, 发送移位器必须发送 0xFF, 而接收移位器返回实际存在于 SDA 引脚中的数据。发送移位器会在 SCL 引脚的最后一个下降沿加载 1 个额外的字, 如果生成 STOP 信号, 则此字应为 0x00, 如果生成一个重复 START 信号, 则此字应为 0xFF。在主机-接收器传输的最后一个字期间, 应由软件置位发送 SSTOP 位, 以生成 NACK。

如果检测到 NACK, 则接收移位寄存器会认定一个错误中断, 但由软件负责生成 STOP 或重复 START 信号。如果在主机-发送期间检测到 NACK, 则中断例程应立即在发送移位器寄存器中写入 0x00 (如果生成 STOP) 或 0xFF (如果生成重复 START)。软件则应等待 SCL 的下一个上升沿, 然后禁用两个定时器。在等待重复 START 或 STOP 信号的设置延时后, 应禁用发送移位器。

受同步延迟影响, 发送输出的数据有效时间为 2 个 FlexIO 时钟周期, 因此, 最大波特率为 FlexIO 时钟频率/6。

有效的 I2C 主机数据延迟 2 个周期, 这是因为时钟输出在对发送/接收移位器计时之前, 会先经过同步器 (以确保某些 SDA 保持时间)。因为 SCL 输入与 FlexIO 时钟同步, 所以同步延迟为 1 个周期, 然后再用 1 个周期来生成输出。

表 47-68. I2C 主机配置

寄存器	值	备注
SHIFTCFGn	0x0000_0032	起始位使能 (逻辑 0) 和停止位使能 (逻辑 1)。
SHIFTCTLn	0x0101_0082	在时钟上升沿用定时器 1 配置发送, 其中, 引脚 0 上反相输出使能 (开漏输出)。
SHIFTCFG(n+1)	0x0000_0020	起始位禁用, 停止位使能 (逻辑 0), 以便进行 ACK/NACK 检测。
SHIFTCTL(n+1)	0x0180_0001	在时钟下降沿用定时器 1 配置接收, 其中, 数据在引脚 0 上输入。
TIMCMPn	0x0000_2501	配置 2 字传输, 波特率为 FlexIO 时钟/4。使 TIMCMP[15:8] = (字数 x 18) + 1。使 TIMCMP[7:0] = (波特率分频比/2) - 1。
TIMCFGn	0x0102_2222	配置起始位、停止位、触发器高电平使能、比较禁用, 在输出等于引脚时复位。初始时钟状态为逻辑 0, 且不受复位影响。
TIMCTLn	0x01C1_0101	用引脚 1 输出使能 (SCL 开漏) 配置双 8 位计数器, 移位器 0 标志作为反向触发器。
TIMCMP(n+1)	0x0000_000F	配置 8 位传输。使 TIMCMP[15:0] = (位数 x 2) - 1。

下一页继续介绍此表...

表 47-68. I2C 主机配置 (继续)

寄存器	值	备注
TIMCFG(n+1)	0x0020_1112	在定时器 0 使能时使能, 在定时器 1 禁用时禁用, 在每个字结尾使能起始位和停止位, 在引脚输入递减。
TIMCTL(n+1)	0x01C0_0183	用反相引脚 1 输入(SCL)配置 16 位计数器。
SHIFTBUFn	要传送的数据	可将传送数据写入 SHIFTBUFBS[7:0], 用移位器状态标志指示何时可以利用中断或 DMA 请求写入数据。
SHIFTBUF(n+1)	要接收的数据	可以从 SHIFTBUFBS[7:0]读取接收的数据, 移位器状态标志用于指示何时可以使用中断或 DMA 请求读取数据。

47.5.6 I2S 主机

可使用两个定时器、两个移位器和四个引脚支持 I2S 主机模式。一个定时器用于生成位时钟和控制移位器, 一个定时器用于生成帧同步。FlexIO 等待第一次写入到发送数据缓冲区, 然后才会使能位时钟和帧同步的产生。可使用 DMA 控制器支持数据传送, 当有发送下溢或接收溢出时, 移位器错误标志会置位。

位时钟频率是 FlexIO 时钟频率的偶整数分频比, 初始帧同步置位与第一个位时钟边缘同时出现。定时器使用起始位确保在第一个输出数据之前一个时钟周期生成帧同步。

受同步延迟影响, 接收器输入的建立时间为 1.5 个 FlexIO 时钟周期, 因此, 最大波特率为 FlexIO 时钟频率/4。

表 47-69. I2S 主机配置

寄存器	值	备注
SHIFTCFGn	0x0000_0001	首次移位时加载发送数据, 禁用停止位。
SHIFCTLn	0x0003_0002	在时钟上升沿用定时器 0 配置传送, 其中, 数据在引脚 0 上输出。
SHIFTCFG(n+1)	0x0000_0000	起始和停止位禁用。
SHIFCTL(n+1)	0x0080_0101	在时钟下降沿用定时器 0 配置接收, 其中, 数据在引脚 1 上输入。
TIMCMPn	0x0000_3F01	配置 32 位传输, 波特率为 FlexIO 时钟/4。使 TIMCMP[15:8] = (位数 × 2) - 1。使 TIMCMP[7:0] = (波特率分频器 / 2) - 1。
TIMCFGn	0x0000_0202	配置起始位, 触发器高电平时使能, 永不禁用。初始时钟状态为逻辑 1。
TIMCTLn	0x01C3_0201	用引脚 2 输出 (位时钟) 配置双 8 位计数器, 移位器 0 标志作为反向触发器。设置 PINPOL 以使输出移位时钟反转。

下一页继续介绍此表...

表 47-69. I2S 主机配置 (继续)

寄存器	值	备注
TIMCMP(n+1)	0x0000_007F	配置 32 位传输, 波特率为 FlexIO 时钟/4。使 TIMCMP[15:0] = (位数 x 波特率分频比) - 1。
TIMCFG(n+1)	0x0000_0100	在定时器 0 使能时使能, 并且永不禁用。
TIMCTL(n+1)	0x0003_0383	用反相引脚 3 输出配置 16 位计数器 (作为帧同步)。
SHIFTBUFn	要传送的数据	可将传送数据写入 SHIFTBUFBIS, 用移位器状态标志指示何时可以利用中断或 DMA 请求写入数据。通过写入 SHIFTBUF 寄存器, 可支持 LSB 优先传输。
SHIFTBUF(n+1)	要接收的数据	可以从 SHIFTBUFBIS 读取接收的数据, 移位器状态标志用于指示何时可以使用中断或 DMA 请求读取数据。通过从 SHIFTBUF 寄存器读取, 可以支持 LSB 优先传输。

47.5.7 I2S 从机

可使用两个定时器、两个移位器和四个引脚来支持 I2S 从机模式 (针对单个发送和单个接收, 也可以是其他发送和接收的组合)。

发送数据必须在外部帧同步信号到达之前写入发送缓冲器寄存器中, 否则移位器错误标志将置位。

受同步延迟影响, 串行输出数据的输出有效时间为 2.5 个 FlexIO 时钟周期, 因此, 最大波特率为 FlexIO 时钟频率/6。

I2S 从机的最大输出有效时间为 2.5 个周期, 这是因为时钟同步有最大 1.5 周期的延迟, 而输出数据有 1 个周期的延迟

表 47-70. I2S 从机配置

寄存器	值	备注
SHIFTCFGn	0x0000_0000	起始和停止位禁用。
SHIFTCTLn	0x0103_0002	在移位时钟上升沿用定时器 1 配置发送, 其中, 数据在引脚 0 上输出。
SHIFTCFG(n+1)	0x0000_0000	起始和停止位禁用。
SHIFTCTL(n+1)	0x0180_0101	在移位时钟下降沿用定时器 1 配置接收, 其中, 数据在引脚 1 上输入。
TIMCMPn	0x0000_007D	配置两个 32 位传输/帧。使 TIMCMP[15:0] = (位数 x 4) - 3。

下一页继续介绍此表...

表 47-70. I2S 从机配置 (继续)

寄存器	值	备注
TIMCFGn	0x0030_2400	配置引脚上升沿使能（反相帧同步）和比较禁用，初始时钟状态为逻辑 1，并在触发器输入上递减（位时钟）。
TIMCTLn	0x0440_0383	用反相引脚 3 输入（帧同步）配置 16 位计数器，以引脚 2 输入（位时钟）为触发器。
TIMCMP(n+1)	0x0000_003F	配置 32 位传输。使 TIMCMP[15:0] = (位数 x 2) - 1。
TIMCFG(n+1)	0x0020_3500	配置引脚上升沿使能（触发器高位）和比较禁用（触发器低位），初始时钟状态为逻辑 0，并在引脚输入上递减。
TIMCTL(n+1)	0x0340_0203	用引脚 2 输入（位时钟）配置 16 位计数器，以定时器 0 输出作为触发器。
SHIFTBUFn	要传送的数据	可将传送数据写入 SHIFTBUFBIS，用移位器状态标志指示何时可以利用中断或 DMA 请求写入数据。通过写入 SHIFTBUF 寄存器，可支持 LSB 优先传输。
SHIFTBUF(n+1)	要接收的数据	可以从 SHIFTBUFBIS 读取接收的数据，移位器状态标志用于指示何时可以使用中断或 DMA 请求读取数据。通过从 SHIFTBUF 寄存器读取，可以支持 LSB 优先传输。

第 48 章

集成芯片间声音 (I2S) /同步音频接口 (SAI)

48.1 此模块的芯片实现细节

48.1.1 实例化信息

此器件包含两个 I²S 模块。

根据器件的配置，模块功能包括：

- TX 数据线路/模块： 1
- RX 数据线路/模块： 1
- FIFO 大小 (字)： 8
- 最大字数/帧： 16
- 最大的位时钟分频器： 512

48.1.2 I²S/SAI 时钟

48.1.2.1 音频主机时钟

将接收器或发送器配置为内部生成的位时钟时，音频主机时钟 (MCLK) 用于生成位时钟。音频主机时钟还可作为引脚的输出或输入。发送器和接收器具有相同的音频主机时钟输入。

48.1.2.2 位时钟

I²S/SAI 发送器和接收器支持异步位时钟 (BCLK)，该时钟可通过音频主机时钟从内部生成或从外部提供。该模块还支持接收器和发送器之间或两个单独 I²S/SAI 外设之间的同步操作选项。

48.1.2.3 总线时钟

总线时钟供控制寄存器使用，用于生成同步中断和 DMA 请求。

48.1.2.4 I²S/SAI 时钟生成

每个 SAI 外设均可控制输入时钟选择、引脚方向和一个音频主机时钟的分频比。

MCLK 控制寄存器 (MCR[MICS]) 的 MCLK 输入时钟选择位选择 I²S/SAI 模块的 MCLK 分频器的时钟输入。

下表所示为此器件的输入时钟选择选项。

表 48-1. I2S0 和 I2S1MCLK 输入时钟选择

MCR[MICS]	时钟选择
00	系统时钟
01	OSCERCLK
10	不支持
11	MCGPLLCLK, MCGFLLCLK, or IRC48MCLK

该模块的 MCLK 分频器寄存器 (MDR) 用于配置 MCLK 分频比。

该模块 MCLK 控制寄存器 (MCR[MOE]) 的 MCLK 输出使能位控制 MCLK 引脚的方向。当 MOE 为 0 时，引脚为输入，当 MOE 为 1 时，引脚为时钟分频器的输出。

发送器和接收器可以分别的选择总线时钟或者是音频主机时钟来生成位时钟。每个模块发送配置 2 寄存器和接收配置 2 寄存器 (TCR2[MSEL]和 RCR2[MSEL]) 的时钟模式字段用于选择主机时钟。

下表所示为此器件模块实例的 TCR2[MSEL]和 RCR2[MSEL]字段设置。

表 48-2. I2S0 主机时钟设置

TCR2[MSEL]、RCR2[MSEL]	主机时钟
00	总线时钟
01	当 MOE 为 1 时: MCLK 通过 PLL、OSC 或系统时钟从内部生成。 当 MOE 为 0 时: I2S0_MCLK。
10	I2S1_MCLK
11	保留

表 48-3. I2S1 主机时钟设置

TCR2[MSEL], RCR2[MSEL]	主机时钟
00	总线时钟
01	当 MOE 为 1 时: MCLK 通过 PLL、OSC 或系统时钟从内部生成。 当 MOE 为 0 时: I2S0_MCLK。
10	I2S0_MCLK
11	保留

48.1.2.5 时钟门控和 I²S/SAI 初始化

可使用 SIM 寄存器中的其中一位来控制 I²S/SAI 模块的时钟。为了功耗最小化,可在任何复位后清除这些位,从而禁用相应模块的时钟。时钟使能位应在模块初始化例程开始时通过软件置位,以便在任何 I²S/SAI 寄存器初始化之前使能模块时钟。

48.1.3 I²S/SAI 在低功耗模式下的操作

48.1.3.1 停止和超低功耗模式

在停止模式下,在相应的停止使能位置位(分别为 TCSR[STOPE]与"或者"或"RCSR[STOPE])并且发送器"与"或者"或"接收器正在使用一个外部生成的位时钟或者在停止模式下保持工作的音频主机时钟的条件下,SAI 发送器"与"或者"或"接收器可以继续工作。SAI 发送器"与"或者"或"接收器可以生成一个异步中断,以从停止模式唤醒 CPU。

在 VLPS 模式下,如果是从运行模式进入 VLPS 模式,则模块的行为与在停止模式下相同。然而,如果是从 VLPR 模式进入 VLPS 模式,FIFO 可能受总线带宽限制的影响,在从停止模式唤醒之前下溢或溢出。在 VLPW 和 VLPR 模式下,模块受最大总线时钟频率的限制。

用内部生成的位时钟或者在停止模式下被禁用的音频主机时钟工作时:

在停止模式下,如果清零发送器停止使能(TCSR[STOPE])位,则发送器在完成当前发送帧之后禁用;如果清零接收器停止使能(RCSR[STOPE])位,则接收器在完成当前接收帧之后禁用。在当前帧结束、等待发送器和接收器禁用的时候,不能——不能确认——进入停止模式。

48.1.3.2 低漏电模式

在进入低漏电模式时，停止使能 (TCSR[STOPE]和 RCSR[STOPE]) 位将被忽略，并且 SAI 将在完成当前发送和接收帧之后禁用。在当前帧结束、等待发送器和接收器禁用的时候，不能（不能确认）进入停止模式。

48.2 简介

I²S（或 I2S）模块提供同步音频接口（SAI），支持带帧同步功能的全双工串行接口，例如 I²S、AC97、TDM、编解码器/DSP 接口。

48.2.1 特性

请注意，有部分特性不受所有 SAI 实例的支持；请参阅本章第一节中有关芯片特性的信息。

- 带独立位时钟和帧同步的发送器，支持 1 个数据线路
- 带独立位时钟和帧同步的接收器，支持 1 个数据线路
- 最大帧大小 16 字
- 字尺寸在 8 位与 32 位之间
- 可以为帧中的第一个字和其余字单独配置字尺寸
- 用于每个发送和接收通道的异步 8 x 32 位 FIFO
- 支持在发生 FIFO 错误之后正常重启
- 支持在发生 FIFO 错误之后自动重新启动，无需软件干预
- 支持将 8 位和 16 位数据打包到每个 32 位 FIFO 字

48.2.2 结构框图

下面的结构框图还展示了模块时钟。

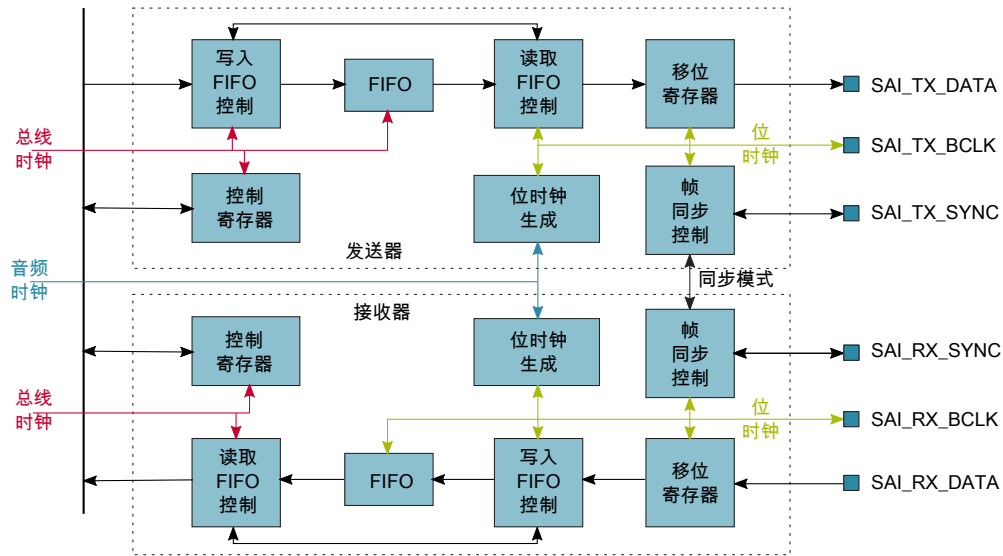


图 48-1. I²S/SAI 结构框图

48.2.3 工作模式

此模块可运行于以下功耗模式：运行模式、停止模式、低漏电模式和调试模式。

48.2.3.1 运行模式

在运行模式下，SAI 发送器和接收器正常工作。

48.2.3.2 停止模式

在停止模式下，在相应的停止使能位置位（分别为 TCSR[STOPE] 与“或者”或“RCSR[STOPE]）并且发送器”与“或者”或“接收器正在使用一个外部生成的位时钟或者在停止模式下保持工作的音频主机时钟的条件下，SAI 发送器”与“或者”或“接收器可以继续工作。SAI 发送器”与“或者”或“接收器可以生成一个异步中断，以从停止模式唤醒 CPU。

在停止模式下，如果清零发送器停止使能(TCSR[STOPE])位，则发送器在完成当前发送帧之后禁用；如果清零接收器停止使能(RCSR[STOPE])位，则接收器在完成当前接收帧之后禁用。在当前帧结束、等待发送器和接收器禁用的时候，不能——不能确认——进入停止模式。

48.2.3.3 低漏电模式

在进入低漏电模式时，停止使能（TCSR[STOPE]和RCSR[STOPE]）位将被忽略，并且SAI将在完成当前发送和接收帧之后禁用。在当前帧结束、等待发送器和接收器禁用的时候，不能（不能确认）进入停止模式。

48.2.3.4 调试模式

在调试模式下，SAI发送器"与"或者"或"接收器可以在调试使能位置位的条件下继续工作。当TCSR[DBG E]或RCSR[DBG E]位清零且进入调试模式时，SAI将在完成当前发送或接收帧之后禁用。发送器和接收器位时钟不受调试模式的影响。

48.3 外部信号

名称	函数	I/O
SAI_TX_BCLK	传送位时钟。从外部生成时，此位时钟为输入；从内部生成时，为输出。	I/O
SAI_TX_SYNC	发送帧同步。从外部生成时，此帧同步为位时钟同步采样的输入；从内部生成时，为位时钟同步生成的输出。	I/O
SAI_TX_DATA	发送数据。发送数据由位时钟同步生成，当未发送字时为三态。	O
SAI_RX_BCLK	接收位时钟。从外部生成时，此位时钟为输入；从内部生成时，为输出。	I/O
SAI_RX_SYNC	接收帧同步。从外部生成时，此帧同步为位时钟同步采样的输入；从内部生成时，为位时钟同步生成的输出。	I/O
SAI_RX_DATA	接收数据。接收数据由位时钟同步采样。	I
SAI_MCLK	音频主机时钟。从外部生成时，此主机时钟为输入；从内部生成时，为输出。	I/O

48.4 存储器映射和寄存器定义

对从偏移量 0x108 及以上的地址进行读取或写入访问时，会导致总线错误。

I2S 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4002_F000	SAI 发送控制寄存器 (I2S0_TCSR)	32	不适用	0000_0000h	48.4.1/1219
4002_F004	SAI 发送配置 1 寄存器 (I2S0_TCR1)	32	不适用	0000_0000h	48.4.2/1223
4002_F008	SAI 发送配置 2 寄存器 (I2S0_TCR2)	32	不适用	0000_0000h	48.4.3/1223
4002_F00C	SAI 发送配置 3 寄存器 (I2S0_TCR3)	32	不适用	0000_0000h	48.4.4/1225
4002_F010	SAI 发送配置 4 寄存器 (I2S0_TCR4)	32	不适用	0000_0000h	48.4.5/1226
4002_F014	SAI 发送配置 5 寄存器 (I2S0_TCR5)	32	不适用	0000_0000h	48.4.6/1228
4002_F020	SAI 发送数据寄存器 (I2S0_TDR0)	32	W (始终读 0)	0000_0000h	48.4.7/1229
4002_F040	SAI 发送 FIFO 寄存器 (I2S0_TFR0)	32	不适用	0000_0000h	48.4.8/1229
4002_F060	SAI 发送掩码寄存器 (I2S0_TMR)	32	不适用	0000_0000h	48.4.9/1230
4002_F080	SAI 接收控制寄存器 (I2S0_RCSR)	32	不适用	0000_0000h	48.4.10/1231
4002_F084	SAI 接收配置 1 寄存器 (I2S0_RCR1)	32	不适用	0000_0000h	48.4.11/1235
4002_F088	SAI 接收配置 2 寄存器 (I2S0_RCR2)	32	不适用	0000_0000h	48.4.12/1235
4002_F08C	SAI 接收配置 3 寄存器 (I2S0_RCR3)	32	不适用	0000_0000h	48.4.13/1237
4002_F090	SAI 接收配置 4 寄存器 (I2S0_RCR4)	32	不适用	0000_0000h	48.4.14/1238
4002_F094	SAI 接收配置 5 寄存器 (I2S0_RCR5)	32	不适用	0000_0000h	48.4.15/1240
4002_F0A0	SAI 接收数据寄存器 (I2S0_RDR0)	32	不适用	0000_0000h	48.4.16/1241
4002_F0C0	SAI 接收 FIFO 寄存器 (I2S0_RFR0)	32	不适用	0000_0000h	48.4.17/1241
4002_F0E0	SAI 接收掩码寄存器 (I2S0_RMR)	32	不适用	0000_0000h	48.4.18/1242
4002_F100	SAI MCLK 控制寄存器 (I2S0_MCR)	32	不适用	0000_0000h	48.4.19/1243
4002_F104	SAI MCLK 分频寄存器 (I2S0_MDR)	32	不适用	0000_0000h	48.4.20/1244
4003_0000	SAI 发送控制寄存器 (I2S1_TCSR)	32	不适用	0000_0000h	48.4.1/1219
4003_0004	SAI 发送配置 1 寄存器 (I2S1_TCR1)	32	不适用	0000_0000h	48.4.2/1223
4003_0008	SAI 发送配置 2 寄存器 (I2S1_TCR2)	32	不适用	0000_0000h	48.4.3/1223
4003_000C	SAI 发送配置 3 寄存器 (I2S1_TCR3)	32	不适用	0000_0000h	48.4.4/1225
4003_0010	SAI 发送配置 4 寄存器 (I2S1_TCR4)	32	不适用	0000_0000h	48.4.5/1226
4003_0014	SAI 发送配置 5 寄存器 (I2S1_TCR5)	32	不适用	0000_0000h	48.4.6/1228
4003_0020	SAI 发送数据寄存器 (I2S1_TDR0)	32	W (始终读 0)	0000_0000h	48.4.7/1229
4003_0040	SAI 发送 FIFO 寄存器 (I2S1_TFR0)	32	不适用	0000_0000h	48.4.8/1229
4003_0060	SAI 发送掩码寄存器 (I2S1_TMR)	32	不适用	0000_0000h	48.4.9/1230

下一页继续介绍此表...

I2S 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
4003_0080	SAI 接收控制寄存器 (I2S1_RCSR)	32	不适用	0000_0000h	48.4.10/1231
4003_0084	SAI 接收配置 1 寄存器 (I2S1_RCR1)	32	不适用	0000_0000h	48.4.11/1235
4003_0088	SAI 接收配置 2 寄存器 (I2S1_RCR2)	32	不适用	0000_0000h	48.4.12/1235
4003_008C	SAI 接收配置 3 寄存器 (I2S1_RCR3)	32	不适用	0000_0000h	48.4.13/1237
4003_0090	SAI 接收配置 4 寄存器 (I2S1_RCR4)	32	不适用	0000_0000h	48.4.14/1238
4003_0094	SAI 接收配置 5 寄存器 (I2S1_RCR5)	32	不适用	0000_0000h	48.4.15/1240
4003_00A0	SAI 接收数据寄存器 (I2S1_RDRO)	32	不适用	0000_0000h	48.4.16/1241
4003_00C0	SAI 接收 FIFO 寄存器 (I2S1_RFR0)	32	不适用	0000_0000h	48.4.17/1241
4003_00E0	SAI 接收掩码寄存器 (I2S1_RMR)	32	不适用	0000_0000h	48.4.18/1242
4003_0100	SAI MCLK 控制寄存器 (I2S1_MCR)	32	不适用	0000_0000h	48.4.19/1243
4003_0104	SAI MCLK 分频寄存器 (I2S1_MDR)	32	不适用	0000_0000h	48.4.20/1244

48.4.1 SAI 发送控制寄存器 (I2Sx_TCSR)

地址: 基址 基准 + 0h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TE	STOPE	DBGE	BCE	0	0	0	SR	0	0	0	WSF	SEF	FEF	FWF	FRF
W	TE	STOPE	DBGE	BCE	0	0	FR	SR	0	0	0	w1c	w1c	w1c	FWF	FRF
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W				WSIE	SEIE	FEIE	FWIE	FRIE							FWDE	FRDE
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCSR 字段描述

字段	描述
31 TE	<p>发送器使能</p> <p>使能/禁用发送器。当软件清除此字段时，发送器将保持使能，并且此位保持置位，直到当前帧结束为止。</p> <p>0 发送器被禁用。</p> <p>1 发送器使能，或者发送器已经被禁用并且尚未达到帧结束。</p>
30 STOPE	<p>停止模式使能</p> <p>配置发送器在停止模式下的操作。在所有低漏电停止模式下，此字段将被忽略，发送器将被禁用。</p> <p>0 发送器在停止模式下被禁用。</p> <p>1 发送器在停止模式下使能。</p>
29 DBGE	<p>调试模式使能</p> <p>在调试模式下使能/禁用发送器操作。发送位时钟不受调试模式的影响。</p> <p>0 在调试模式下，发送器将在完成当前帧之后被禁用。</p> <p>1 发送器在调试模式下使能。</p>
28 BCE	<p>位时钟使能</p> <p>独立于 TE，使能发送位时钟。每当 TE 置位时，此字段都会自动置位。当软件清除此字段时，发送位时钟将保持使能，并且此位保持置位，直到当前帧结束为止。</p>

下一页继续介绍此表...

I2Sx_TCSR 字段描述 (继续)

字段	描述
	0 发送位时钟被禁用。 1 发送位时钟使能。
27-26 保留	此只读字段为保留字段且值始终为 0。
25 FR	FIFO 复位 复位 FIFO 指针。读取此字段将始终返回零。只有当发送器被禁用或者 FIFO 错误标志置位时，才能复位 FIFO 指针。 0 无效。 1 FIFO 复位。
24 SR	软件复位 置位时，将复位包括 FIFO 指针在内的内部发送器逻辑。软件可见寄存器不受影响，状态寄存器除外。 0 无效。 1 软件复位。
23-21 保留	此只读字段为保留字段且值始终为 0。
20 WSF	字开始标志 指示检测到配置字的开始。向此字段写入逻辑 1，清除此标志。 0 未检测到字开始。 1 检测到字开始。
19 SEF	同步错误标志 指示在外部生成的帧同步中检测到错误。向此字段写入逻辑 1，清除此标志。 0 未检测到同步错误。 1 检测到帧同步错误。
18 FEF	FIFO 错误标志 指示使能发送 FIFO 发生了下溢。向此字段写入逻辑 1，清除此标志。 0 未检测到发送下溢。 1 检测到发送下溢。
17 FWF	FIFO 警告标志 指示使能发送 FIFO 为空。 0 无使能发送 FIFO 为空。 1 使能发送 FIFO 为空。
16 FRF	FIFO 请求标志 指示使能发送通道 FIFO 中的字数少于或等于发送 FIFO 深度。 0 尚未达到发送 FIFO 深度。 1 已经达到发送 FIFO 深度。
15-13 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

I2Sx_TCSR 字段描述 (继续)

字段	描述
12 WSIE	字开始中断使能 使能/禁用字开始中断。 0 禁用中断。 1 使能中断。
11 SEIE	同步错误中断使能 使能/禁用同步错误中断。 0 禁用中断。 1 使能中断。
10 FEIE	FIFO 错误中断使能 使能/禁用 FIFO 错误中断。 0 禁用中断。 1 使能中断。
9 FWIE	FIFO 警告中断使能 使能/禁用 FIFO 警告中断。 0 禁用中断。 1 使能中断。
8 FRIE	FIFO 请求中断使能 使能/禁用 FIFO 请求中断。 0 禁用中断。 1 使能中断。
7-5 保留	此只读字段为保留字段且值始终为 0。
4-2 保留	此只读字段为保留字段且值始终为 0。
1 FWDE	FIFO 警告 DMA 使能 使能/禁用 DMA 请求。 0 禁用 DMA 请求。 1 使能 DMA 请求。
0 FRDE	FIFO 请求 DMA 使能 使能/禁用 DMA 请求。 0 禁用 DMA 请求。 1 使能 DMA 请求。

48.4.2 SAI 发送配置 1 寄存器 (I2Sx_TCR1)

地址: 基址 基准 + 4h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TFW															
W																	TFW															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR1 字段描述

字段	描述
31-3 保留	此只读字段为保留字段且值始终为 0。
TFW	发送 FIFO 深度 为所有使能的发送通道配置深度等级。

48.4.3 SAI 发送配置 2 寄存器 (I2Sx_TCR2)

当 TCSR[TE]置位时，不得改变此寄存器。

地址: 基址 基准 + 8h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYNC	BCS	BCI	MSEL	BCP	BCD	0									
W	SYNC	BCS	BCI	MSEL	BCP	BCD										
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DIV							
W									DIV							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR2 字段描述

字段	描述
31-30 SYNC	同步模式 配置异步和同步操作模式。当配置为同步操作模式时，接收器必须配置为异步操作。 00 异步模式。 01 与接收器同步。
29 BCS	位时钟交换 此字段交换发送器使用的位时钟。当发送器配置为异步模式且此位置位时，发送器将由接收器位时钟 (SAI_RX_BCLK) 计时。这样允许发送器和接收器共享同一位时钟，但发送器会继续使用发送帧同步 (SAI_TX_SYNC)。

下一页继续介绍此表...

I2Sx_TCR2 字段描述 (继续)

字段	描述
	<p>当发送器配置为同步模式时，发送器 BCS 字段和接收器 BCS 字段必须设置为相同的值。当两个字段都置位时，发送器和接收器都将由发送器位时钟(SAI_TX_BCLK)计时，但会使用接收器帧同步(SAI_RX_SYNC)。</p> <p>0 使用正常位时钟源。 1 交换位时钟源。</p>
28 BCI	<p>位时钟输入</p> <p>当此字段置位且将内部生成的位时钟用于同步或异步模式时，发送器实际使用的位时钟会被管脚输出延迟延迟（发送器时钟从管脚输入进来，就如时钟是外部生成的一样）。其结果将缩短数据输入建立时钟，但会增加数据输出有效时间。</p> <p>当此位置位时，针对发送器应使用数据手册中提供的从机模式时序数据。在同步模式下，此位允许发送器使用数据手册提供的从机模式时序数据，而接收器则使用主机模式时序。当针对外部生成的位时钟配置时 同步时，此字段不起作用。</p> <p>0 无效。 1 对内部逻辑计时，就如位时钟是由外部生成的一样。</p>
27-26 MSEL	<p>MCLK 选择</p> <p>选择用于生成内部生成的位时钟的音频主机时钟选项。当针对外部生成的位时钟配置时，此字段不起作用。</p> <p>注：根据具体的器件，有些主机时钟选项可能不可用。有关各选项的意义，请参见芯片特性信息。</p> <p>00 选择的总线时钟。 01 选择的主机时钟(MCLK) 1 选项。 10 选择的主机时钟(MCLK) 2 选项。 11 选择的主机时钟(MCLK) 3 选项。</p>
25 BCP	<p>位时钟极性</p> <p>配置位时钟的极性。</p> <p>0 位时钟高电平有效，在上升沿驱动输出，在下降沿采样输入。 1 位时钟低电平有效，在下降沿驱动输出，在上升沿采样输入。</p>
24 BCD	<p>位时钟方向</p> <p>配置位时钟的方向。</p> <p>0 在从机模式下，位时钟通过外部方式产生。 1 在主机模式下，位时钟通过内部方式产生。</p>
23-8 保留	此只读字段为保留字段且值始终为 0。
DIV	<p>位时钟分频</p> <p>当配置为内部位时钟时，将音频主机时钟分频以生成位时钟。分度值为(DIV + 1) * 2。</p>

48.4.4 SAI 发送配置 3 寄存器 (I2Sx_TCR3)

地址: 基址 基准 + Ch 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R	0								0								TCE
W	0																TCE
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0												WDFL				
W	0												WDFL				
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

I2Sx_TCR3 字段描述

字段	描述
31–24 保留	此只读字段为保留字段且值始终为 0。
23–17 保留	此只读字段为保留字段且值始终为 0。
16 TCE	<p>发送通道使能</p> <p>使能发送操作对应的数据通道。通道必须在其 FIFO 被访问之前使能。但是在接收操作各帧结束时，更改此字段将立即生成 FIFO 请求和警告标志。</p> <p>0 发送数据通道 N 被禁用。 1 发送数据通道 N 使能。</p>
15–4 保留	此只读字段为保留字段且值始终为 0。
WDFL	<p>字标志配置</p> <p>配置字标志置位的起始字。写入的值必须比字编号小 1。例如，写入 0 会配置帧中的第一个字。当配置值大于 TCR4[FRSZ]时，字标志的起始位将不会置位。</p>

48.4.5 SAI 发送配置 4 寄存器 (I2Sx_TCR4)

当 TCSR[TE]置位时，不得改变此寄存器。

地址: 基址 基准 + 10h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			FCONT	0		FPACK		0				FRSZ			
W	[保留]			FCONT	[保留]		FPACK		[保留]				FRSZ			
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SYWD					0			MF	FSE	ONDEM	FSP	FSD
W	[保留]			SYWD					[保留]			MF	FSE	ONDEM	FSP	FSD
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR4 字段描述

字段	描述
31-29 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

I2Sx_TCR4 字段描述 (继续)

字段	描述
28 FCONT	FIFO 遇错继续 配置在什么时候, 在检测到 FIFO 错误时 SAI 将继续发送。 0 在遇到 FIFO 错误时, SAI 将在 FIFO 错误标志被清除之后, 从下一帧起始位继续执行。 1 在遇到 FIFO 错误时, SAI 将在 FIFO 警告标志被清除之后, 从导致 FIFO 错误置位的同一字继续执行。
27-26 保留	此只读字段为保留字段且值始终为 0。
25-24 FPACK	FIFO 包装模式 使能将 8 位数据或 16 位数据打包到每个 32 位 FIFO 字中。如果字长度大于 8 位或 16 位, 则只会从 FIFO 中载入前 8 位或前 16 位。每帧里的首字始终以一个 32 位 FIFO 字开始, 移位的首位必须在第一个包装字中配置。当使能 FIFO 包装时, FIFO 写指针只会在软件已经写入整个 32 位 FIFO 字时进一。 00 FIFO 包装被禁用。 01 保留 10 8 位 FIFO 包装使能 11 16 位 FIFO 包装使能
23-20 保留	此只读字段为保留字段且值始终为 0。
19-16 FRSZ	帧长度 配置每帧的字数。写入的值必须比帧中的字数小 1。例如, 如果每帧一个字, 则写入 0。支持的最大帧长度为 16 个字。
15-13 保留	此只读字段为保留字段且值始终为 0。
12-8 SYWD	同步宽度 以位时钟为单位配置帧同步长度。写入的值必须比位时钟编号小 1。例如, 若要使帧同步只在一个位时钟的长度内置位, 则写入 0。同步宽度的配置值不得长于帧的首字。
7-5 保留	此只读字段为保留字段且值始终为 0。
4 MF	MSB 优先 配置首先发送 LSB 还是 MSB。 0 先发送 LSB。 1 先发送 MSB。
3 FSE	帧同步过早 0 帧同步在帧首位置位。 1 帧同步在帧首位之前一位置位。
2 ONDEM	按需模式 置位时, 帧同步将内部生成, 仅当 FIFO 警告标志清零时, 才会生成帧同步。 0 内部帧同步连续生成。 1 内部帧同步在 FIFO 警告标志清零时生成。
1 FSP	帧同步极性 配置帧同步的极性。

下一页继续介绍此表...

I2Sx_TCR4 字段描述 (继续)

字段	描述
	0 帧同步高电平有效。 1 帧同步低电平有效。
0 FSD	帧同步方向 配置帧同步的方向。 0 在从机模式下，帧同步通过外部方式生成。 1 在主机模式下，帧同步通过内部方式生成。

48.4.6 SAI 发送配置 5 寄存器 (I2Sx_TCR5)

当 TCSR[TE]置位时，不得改变此寄存器。

地址: 基址 基准 + 14h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	WNW			0	WOW			0	FBT			0																			
W		WNW				WOW				FBT																						
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

I2Sx_TCR5 字段描述

字段	描述
31-29 保留	此只读字段为保留字段且值始终为 0。
28-24 WNW	N 字宽 配置每个字中的位数，适用于每个字，帧中的首字除外。写入的值必须比每字位数小 1。不支持小于 8 位的字度。
23-21 保留	此只读字段为保留字段且值始终为 0。
20-16 WOW	首字宽 配置每帧首字的位数。写入的值必须比首字中的位数小 1。如果每帧只有一个字，则不支持小于 8 位的字宽。
15-13 保留	此只读字段为保留字段且值始终为 0。
12-8 FBT	首位移位 配置帧中的每个字中首个发送位的位索引。如果配置为 MSB 优先，被发送的下一位的索引将比当前发送位小 1。如果配置为 LSB 优先，被发送的下一位的索引将比当前发送位大 1。配置为 MSB 优先时，写入的值必须大于或等于字宽。配置为 LSB 优先时，写入的值必须小于或等于 31 字宽。
保留	此只读字段为保留字段且值始终为 0。

48.4.7 SAI 发送数据寄存器 (I2Sx_TDRn)

地址: 基址 + 20h 偏移 + (4d × i), 其中 i=0d 到 0d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	TDR																															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TDRn 字段描述

字段	描述
TDR	发送数据寄存器 在访问通道的发送数据寄存器之前, 必须置位相应的 TCR3[TCE]位。当发送 FIFO 未满时写入此寄存器, 结果会将写入的数据推入发送数据 FIFO。当发送 FIFO 已满时, 对此寄存器的写入操作将被忽略。

48.4.8 SAI 发送 FIFO 寄存器 (I2Sx_TFRn)

读取和写入指针的 MSB 用于区分 FIFO 的满和空两种状态。如果读取和写入指针相同, 则 FIFO 为空。如果除 MSB 以外, 读取和写入指针相同, 则 FIFO 为满。

地址: 基址 + 40h 偏移 + (4d × i), 其中 i=0d 到 0d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
R	0	0											WFP						
W																WFP			
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
R	0											RFP							
W																RFP			
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

I2Sx_TFRn 字段描述

字段	描述
31 保留	此只读字段为保留字段且值始终为 0。
30-20 保留	此只读字段为保留字段且值始终为 0。
19-16 WFP	写入 FIFO 指针 发送数据通道的 FIFO 写入指针。
15-4 保留	此只读字段为保留字段且值始终为 0。
RFP	读取 FIFO 指针

下一页继续介绍此表...

I2Sx_TFRn 字段描述 (继续)

字段	描述
	发送数据通道的 FIFO 读取指针。

48.4.9 SAI 发送掩码寄存器 (I2Sx_TMR)

此寄存器搭载双缓冲器，并在下列情况下更新：

1. 当 TCSR[TE]先置位时
2. 每帧结束时。

这样就允许每帧中的屏蔽字可以逐帧更改。

地址: 基址 基准 + 60h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TWM															
W	0																TWM															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TMR 字段描述

字段	描述
31-16 保留	此只读字段为保留字段且值始终为 0。
TWM	<p>发送字掩码</p> <p>配置是否针对帧中的对应字屏蔽发送字 (发送数据引脚三态 , 且不从 FIFO 读取发送数据)。</p> <p>0 字 N 使能。 1 字 N 被屏蔽。屏蔽时, 发送数据引脚三态。</p>

48.4.10 SAI 接收控制寄存器 (I2Sx_RCSR)

地址: 基址 基准 + 80h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RE	STOPE	DBGE	BCE	0	0	0	SR	0	0	0	WSF	SEF	FEF	FWF	FRF
W	RE	STOPE	DBGE	BCE	0	0	FR	SR	0	0	0	w1c	w1c	w1c	FWF	FRF
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W				WSIE	SEIE	FEIE	FWIE	FRIE							FWDE	FRDE
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCSR 字段描述

字段	描述
31 RE	<p>接收器使能</p> <p>使能/禁用接收器。当软件清除此字段时，接收器将保持使能，并且此字段保持置位，直到当前帧结束为止。</p> <p>0 接收器使能。 1 接收器使能，或者接收器已经被禁用并且尚未达到帧结束。</p>
30 STOPE	<p>停止模式使能</p> <p>配置接收器在停止模式下的操作。在所有低漏电停止模式下，此位将被忽略，接收器将被禁用。</p> <p>0 接收器在停止模式下被禁用。 1 接收器在调试模式下被禁用。</p>
29 DBGE	<p>调试模式使能</p> <p>在调试模式下使能/禁用接收器操作。接收位时钟不受调试模式的影响。</p> <p>0 在调试模式下，接收器将在完成当前帧之后被禁用。 1 接收器在调试模式下使能。</p>
28 BCE	<p>位时钟使能</p> <p>独立于 RE，使能接收位时钟。每当 RE 置位时，此字段都会自动置位。当软件清除此字段时，接收位时钟将保持使能，并且此字段保持置位，直到当前帧结束为止。</p>

下一页继续介绍此表...

I2Sx_RCSR 字段描述 (继续)

字段	描述
	0 接收位时钟被禁用。 1 接收位时钟使能。
27-26 保留	此只读字段为保留字段且值始终为 0。
25 FR	FIFO 复位 复位 FIFO 指针。读取此字段将始终返回零。只有当接收器被禁用或者 FIFO 错误标志置位时，才能复位 FIFO 指针。 0 无效。 1 FIFO 复位。
24 SR	软件复位 复位包括 FIFO 指针在内的内部接收器逻辑。软件可见寄存器不受影响，状态寄存器除外。 0 无效。 1 软件复位。
23-21 保留	此只读字段为保留字段且值始终为 0。
20 WSF	字开始标志 指示检测到配置字的开始。向此字段写入逻辑 1，清除此标志。 0 未检测到字开始。 1 检测到字开始。
19 SEF	同步错误标志 指示在外部生成的帧同步中检测到错误。向此字段写入逻辑 1，清除此标志。 0 未检测到同步错误。 1 检测到帧同步错误。
18 FEF	FIFO 错误标志 指示使能接收 FIFO 已溢出。向此字段写入逻辑 1，清除此标志。 0 未检测到接收溢出。 1 检测到接收溢出。
17 FWF	FIFO 警告标志 指示使能接收 FIFO 已满。 0 使能接收 FIFO 均未满。 1 使能接收 FIFO 已满。
16 FRF	FIFO 请求标志 指示使能接收通道 FIFO 中的字数大于接收 FIFO 深度。 0 未达到接收 FIFO 深度。 1 已达到接收 FIFO 深度。
15-13 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

I2Sx_RCSR 字段描述 (继续)

字段	描述
12 WSIE	字开始中断使能 使能/禁用字开始中断。 0 禁用中断。 1 使能中断。
11 SEIE	同步错误中断使能 使能/禁用同步错误中断。 0 禁用中断。 1 使能中断。
10 FEIE	FIFO 错误中断使能 使能/禁用 FIFO 错误中断。 0 禁用中断。 1 使能中断。
9 FWIE	FIFO 警告中断使能 使能/禁用 FIFO 警告中断。 0 禁用中断。 1 使能中断。
8 FRIE	FIFO 请求中断使能 使能/禁用 FIFO 请求中断。 0 禁用中断。 1 使能中断。
7-5 保留	此只读字段为保留字段且值始终为 0。
4-2 保留	此只读字段为保留字段且值始终为 0。
1 FWDE	FIFO 警告 DMA 使能 使能/禁用 DMA 请求。 0 禁用 DMA 请求。 1 使能 DMA 请求。
0 FRDE	FIFO 请求 DMA 使能 使能/禁用 DMA 请求。 0 禁用 DMA 请求。 1 使能 DMA 请求。

48.4.11 SAI 接收配置 1 寄存器 (I2Sx_RCR1)

地址: 基址 基准 + 84h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																RFW															
W																	RFW															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR1 字段描述

字段	描述
31-3 保留	此只读字段为保留字段且值始终为 0。
RFW	接收 FIFO 深度 配置所有使能接收器通道的深度等级。

48.4.12 SAI 接收配置 2 寄存器 (I2Sx_RCR2)

当 RCSR[RE]置位时, 不得改变此寄存器。

地址: 基址 基准 + 88h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYNC	BCS	BCI	MSEL	BCP	BCD	0									
W	SYNC	BCS	BCI	MSEL	BCP	BCD										
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DIV							
W									DIV							
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR2 字段描述

字段	描述
31-30 SYNC	同步模式 配置异步和同步操作模式。当配置为同步操作模式时, 发送器必须配置为异步操作。 00 异步模式。 01 与发送器同步。
29 BCS	位时钟交换 此字段交换接收器使用的位时钟。当接收器配置为异步模式且此位置位时, 接收器将由发送器位时钟 (SAI_TX_BCLK) 计时。这样允许发送器和接收器共享同一位时钟, 但接收器会继续使用接收器帧同步 (SAI_RX_SYNC)。

下一页继续介绍此表...

I2Sx_RCR2 字段描述 (继续)

字段	描述
	<p>当接收器配置为同步模式时，发送器 BCS 字段和接收器 BCS 字段必须设置为相同的值。当两个字段都置位时，发送器和接收器都将由接收器位时钟(SAI_RX_BCLK)计时，但会使用发送器帧同步(SAI_TX_SYNC)。</p> <p>0 使用正常位时钟源。 1 交换位时钟源。</p>
28 BCI	<p>位时钟输入</p> <p>当此字段置位且将内部生成的位时钟用于同步或异步模式时，接收器实际使用的位时钟会被管脚输出延迟延迟（接收器时钟从管脚输入进来，就如时钟是外部生成的一样）。其结果将缩短数据输入建立时钟，但会增加数据输出有效时间。</p> <p>当此位置位时，针对接收器应使用数据手册中提供的从机模式时序数据。在同步模式下，此位允许接收器使用数据手册提供的从机模式时序数据，而发送器则使用主机模式时序。当针对外部生成的位时钟配置时 同步时，此字段不起作用。</p> <p>0 无效。 1 对内部逻辑计时，就如位时钟是由外部生成的一样。</p>
27-26 MSEL	<p>MCLK 选择</p> <p>选择用于生成内部生成的位时钟的音频主机时钟选项。当针对外部生成的位时钟配置时，此字段不起作用。</p> <p>注：根据具体的器件，有些主机时钟选项可能不可用。有关各选项的可用性及其对芯片特性的意义，请参见芯片特性信息。</p> <p>00 选择的总线时钟。 01 选择的主机时钟(MCLK) 1 选项。 10 选择的主机时钟(MCLK) 2 选项。 11 选择的主机时钟(MCLK) 3 选项。</p>
25 BCP	<p>位时钟极性</p> <p>配置位时钟的极性。</p> <p>0 位时钟高电平有效，在上升沿驱动输出，在下降沿采样输入。 1 位时钟低电平有效，在下降沿驱动输出，在上升沿采样输入。</p>
24 BCD	<p>位时钟方向</p> <p>配置位时钟的方向。</p> <p>0 在从机模式下，位时钟通过外部方式产生。 1 在主机模式下，位时钟通过内部方式产生。</p>
23-8 保留	此只读字段为保留字段且值始终为 0。
DIV	<p>位时钟分频</p> <p>当配置为内部位时钟时，将音频主机时钟分频以生成位时钟。分度值为 $(DIV + 1) * 2$。</p>

48.4.13 SAI 接收配置 3 寄存器 (I2Sx_RCR3)

地址: 基址 基准 + 8Ch 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R	0								0								RCE
W	RCE																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0												WDFL				
W	WDFL																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

I2Sx_RCR3 字段描述

字段	描述
31-24 保留	此只读字段为保留字段且值始终为 0。
23-17 保留	此只读字段为保留字段且值始终为 0。
16 RCE	接收通道使能 使能接收操作对应的数据通道。通道必须在其 FIFO 被访问之前使能。但是在发送操作各帧结束时，更改此字段将立即生成 FIFO 请求和警告标志。 0 接收数据通道 N 被禁用。 1 接收数据通道 N 使能。
15-4 保留	此只读字段为保留字段且值始终为 0。
WDFL	字标志配置 配置字标志置位的字。写入的值必须比字编号小 1 (例如，要为帧中第一个字配置，就要写入零)。当配置值大于帧大小字段时，字标志的起始位将不会置位。

48.4.14 SAI 接收配置 4 寄存器 (I2Sx_RCR4)

当 RCSR[RE]置位时，不得改变此寄存器。

地址: 基址 基准 + 90h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			FCONT	0		FPACK		0				FRSZ			
W	[Grayed]			FCONT	[Grayed]		FPACK		[Grayed]				FRSZ			
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SYWD					0			MF	FSE	ONDEM	FSP	FSD
W	[Grayed]			SYWD					[Grayed]			MF	FSE	ONDEM	FSP	FSD
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR4 字段描述

字段	描述
31-29 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

I2Sx_RCR4 字段描述 (继续)

字段	描述
28 FCONT	FIFO 遇错继续 配置在什么时候, 在检测到 FIFO 错误时 SAI 将继续接收。 0 在遇到 FIFO 错误时, SAI 将在 FIFO 错误标志被清除之后, 从下一帧起始位继续执行。 1 在遇到 FIFO 错误时, SAI 将在 FIFO 警告标志被清除之后, 从导致 FIFO 错误置位的同一字继续执行。
27-26 保留	此只读字段为保留字段且值始终为 0。
25-24 FPACK	FIFO 包装模式 使能将 8 位数据或 16 位数据打包到每个 32 位 FIFO 字中。如果字长度大于 8 位或 16 位, 则只会把前 8 位或前 16 位存入 FIFO 中。每帧里的首字始终以一个 32 位 FIFO 字开始, 移位的首位必须在第一个包装字中配置。当使能 FIFO 包装时, FIFO 读指针只会在软件已经读取整个 32 位 FIFO 字时进一。 00 FIFO 包装被禁用。 01 保留。 10 8 位 FIFO 包装使能 11 16 位 FIFO 包装使能
23-20 保留	此只读字段为保留字段且值始终为 0。
19-16 FRSZ	帧长度 配置每帧的字数。写入的值必须比帧中的字数小 1。例如, 如果每帧一个字, 则写入 0。支持的最大帧长度为 16 个字。
15-13 保留	此只读字段为保留字段且值始终为 0。
12-8 SYWD	同步宽度 以位时钟为单位配置帧同步长度。写入的值必须比位时钟编号小 1。例如, 若要使帧同步只在一个位时钟的长度内置位, 则写入 0。同步宽度的配置值不得长于帧的首字。
7-5 保留	此只读字段为保留字段且值始终为 0。
4 MF	MSB 优先 配置首先接收 LSB 还是 MSB。 0 先 LSB 接收。 1 先接收 MSB。
3 FSE	帧同步过早 0 帧同步在帧首位置位。 1 帧同步在帧首位之前一位置位。
2 ONDEM	按需模式 置位时, 帧同步将内部生成, 仅当 FIFO 警告标志清零时, 才会生成帧同步。 0 内部帧同步连续生成。 1 内部帧同步在 FIFO 警告标志清零时生成。
1 FSP	帧同步极性 配置帧同步的极性。

下一页继续介绍此表...

I2Sx_RCR4 字段描述 (继续)

字段	描述
	0 帧同步高电平有效。 1 帧同步低电平有效。
0 FSD	帧同步方向 配置帧同步的方向。 0 在从机模式下，帧同步通过外部方式生成。 1 在主机模式下，帧同步通过内部方式生成。

48.4.15 SAI 接收配置 5 寄存器 (I2Sx_RCR5)

当 RCSR[RE]置位时，不得改变此寄存器。

地址: 基址 基准 + 94h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	WNW				0	WOW				0	FBT				0																
W		WNW					WOW					FBT																				
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

I2Sx_RCR5 字段描述

字段	描述
31-29 保留	此只读字段为保留字段且值始终为 0。
28-24 WNW	N 字宽 配置每个字中的位数，适用于每个字，帧中的首字除外。写入的值必须比每字位数小 1。不支持小于 8 位的字度。
23-21 保留	此只读字段为保留字段且值始终为 0。
20-16 WOW	首字宽 配置每帧首字的位数。写入的值必须比首字中的位数小 1。如果每帧只有一个字，则不支持小于 8 位的字宽。
15-13 保留	此只读字段为保留字段且值始终为 0。
12-8 FBT	首位移位 配置帧中的每个字中首个接收位的位索引。如果配置为 MSB 优先，接收的下一位的索引将比当前接收位小 1。如果配置为 LSB 优先，接收的下一位的索引将比当前接收位大 1。配置为 MSB 优先时，写入的值必须大于或等于字宽。配置为 LSB 优先时，写入的值必须小于或等于 31 字宽。
保留	此只读字段为保留字段且值始终为 0。

48.4.16 SAI 接收数据寄存器 (I2Sx_RDRn)

读取此寄存器会在每次读取时引入一个额外的外设时钟等待状态。

地址: 基址 + A0h 偏移 + (4d × i), 其中 i=0d 到 0d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDR																															
W	RDR																															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RDRn 字段描述

字段	描述
RDR	接收数据寄存器 在访问通道的接收数据寄存器之前，必须置位相应的 RCR3[RCE]位。当接收 FIFO 不为空时从此寄存器读取，将返回接收 FIFO 顶部的值。当接收 FIFO 为空时，对此寄存器的读取操作将被忽略。

48.4.17 SAI 接收 FIFO 寄存器 (I2Sx_RFRn)

读取和写入指针的 MSB 用于区分 FIFO 的满和空两种状态。如果读取和写入指针相同，则 FIFO 为空。如果除 MSB 以外，读取和写入指针相同，则 FIFO 为满。

地址: 基址 + C0h 偏移 + (4d × i), 其中 i=0d 到 0d

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												WFP			
W													WFP			
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0											RFP			
W												RFP				
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RFRn 字段描述

字段	描述
31–20 保留	此只读字段为保留字段且值始终为 0。
19–16 WFP	写入 FIFO 指针 接收数据通道的 FIFO 写入指针。
15 保留	此只读字段为保留字段且值始终为 0。
14–4 保留	此只读字段为保留字段且值始终为 0。

下一页继续介绍此表...

I2Sx_RFRn 字段描述 (继续)

字段	描述
RFP	读取 FIFO 指针 接收数据通道的 FIFO 读取指针。

48.4.18 SAI 接收掩码寄存器 (I2Sx_RMR)

此寄存器搭载双缓冲器，并在下列情况下更新：

1. 当 RCSR[RE]先置位时
2. 每帧结束时

这样就允许每帧中的屏蔽字可以逐帧更改。

地址: 基址 基准 + E0h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																RWM															
W	0																RWM															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RMR 字段描述

字段	描述
31-16 保留	此只读字段为保留字段且值始终为 0。
RWM	接收字掩码 配置是否针对帧中对应字屏蔽接收字（接收到的数据被忽略，且不写入接收 FIFO）。 0 字 N 使能。 1 字 N 被屏蔽。

48.4.19 SAI MCLK 控制寄存器 (I2Sx_MCR)

MCLK 控制寄存器(MCR)控制音频主机时钟的时钟源和方向。

地址: 基址 基准 + 100h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DUF	MOE	0				MICS		0							
W	DUF	MOE					MICS									
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_MCR 字段描述

字段	描述
31 DUF	分频器更新标志 向 MCLK 分频器比提供实时更新状态。 0 MCLK 分频器比当前未更新。 1 MCLK 分频器比实时更新。当此标志保持置位时，将阻止对 MCLK 分频器比进一步的更新。
30 MOE	MCLK 输出使能 使能 MCLK 分频器，并将 MCLK 信号引脚配置为输出。当软件清除此字段时，它将保持置位，直到 MCLK 分频器完全被禁用为止。 0 MCLK 信号引脚被配置为旁路 MCLK 分频器的输入。 1 将 MCLK 信号引脚配置为 MCLK 分频器的输出，并使能 MCLK 分频器。
29-26 保留	此只读字段为保留字段且值始终为 0。
25-24 MICS	MCLK 输入时钟选择 选择输入 MCLK 分频器的时钟。在 MCLK 分频器使能时，此字段不能更改。有关与这些输入的连接，请参见芯片特性信息。 00 选择 MCLK 分频器输入时钟 0。 01 选择 MCLK 分频器输入时钟 1。 10 选择 MCLK 分频器输入时钟 2。 11 选择 MCLK 分频器输入时钟 3。
保留	此只读字段为保留字段且值始终为 0。

48.4.20 SAI MCLK 分频寄存器 (I2Sx_MDR)

MCLK 分频寄存器(MDR)配置 MCLK 分频比。尽管可以在 MCLK 分频器时钟使能时更改 MDR，但在 MCR[DUF]置位的情况下，将阻止对 MDR 进行额外的写操作。当 MCLK 分频时钟被禁用时，对 MDR 的写操作不会置位 MCR[DUF]。

地址: 基址 基准 + 104h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0												FRACT				DIVIDE																
W	0												FRACT				DIVIDE																
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_MDR 字段描述

字段	描述
31-20 保留	此只读字段为保留字段且值始终为 0。
19-12 FRACT	MCLK 小数 设置 MCLK 分频比，使得： $MCLK \text{ 输出} = MCLK \text{ 输入} * ((FRACT + 1) / (DIVIDE + 1))$ 。FRACT 的设置值必须等于或小于 DIVIDE 字段中的值。 注：在使用小数分频值时，MCLK 占空比不会始终是 50/50。请参见 音频主机时钟 。
DIVIDE	MCLK 分频 设置 MCLK 分频比，使得： $MCLK \text{ 输出} = MCLK \text{ 输入} * ((FRACT + 1) / (DIVIDE + 1))$ 。FRACT 的设置值必须等于或小于 DIVIDE 字段中的值。 注：在使用小数分频值时，MCLK 占空比不会始终是 50/50。请参见 音频主机时钟 。

48.5 功能说明

本节提供数据块的完整功能说明。

48.5.1 SAI 时钟

SAI 时钟包括：

- 音频主机时钟
- 位时钟
- 总线时钟

48.5.1.1 音频主机时钟

将接收器或发射器配置为内部生成的位时钟时，音频主机时钟用于生成位时钟。发送器和接收器可以在总线时钟和三个音频主机时钟之间独立选择，以生成位时钟。

每个 SAI 外设均可控制输入时钟选择、引脚方向和一个音频主机时钟的分频比。如果使用该音频主机时钟的 SAI 模块已使能，则输入时钟选择和引脚方向不能改变。当 SAI 使用该主机时钟时，MCLK 分频比可以改变，不过分频比更改需要几个周期。可对 MCR[DUF] 轮询以确定分频比更改完成的时间。

音频主机时钟生成和选择取决于具体的芯片。有关如何生成音频主机时钟的详情，请参见芯片特性时钟信息。下图所示为典型设置。

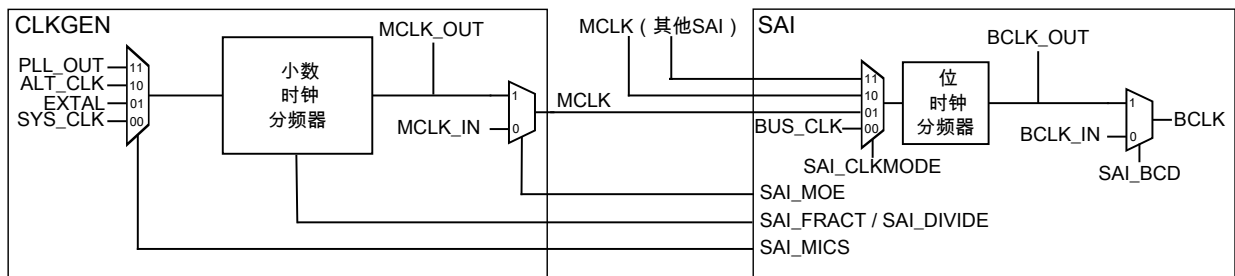


图 48-74. SAI 主机时钟生成

MCLK 小数时钟分频器使用输入时钟的两个时钟边沿生成分频后的时钟，该时钟频率接近于输出频率，但不会产生任何新的时钟边沿。将 FRACT 和 1DIVIDE 配置为相同值将导致 1 分频时钟，同时不支持将 FRACT 配置为高于 DIVIDE。如果将 FRACT 设为低于 50/50 的分频时，对于整数分频比，占空比范围为 66/33，对于较大的非整数分频比，该范围将接近 50/50。如果分频比为整数或半整数，则不存在周期性抖动或占空比偏差，否则分频器输出将在两个经过分频的频率之间振荡，这两个频率为分频器输入时钟频率的最接近整数或半整数除数。因为在生成分频时钟时使用了两个输入时钟边沿，所以最大抖动等于分频器输入时钟周期的一半。

48.5.1.2 位时钟

SAI 发送器和接收器支持异步自由运行位时钟 (BCLK)，该时钟可通过音频主机时钟从内部生成或从外部提供。此外还具有支持接收器和发送器之间进行同步位时钟和帧同步操作选项。

外部生成的位时钟必须：

- 在使能 SAI 发送器或接收器之前使能
- 在禁用 SAI 发送器或接收器并完成其当前帧之后禁用

如果 SAI 发送器或接收器在异步模式下使用外部生成的位时钟，并且该位时钟由 STOP 模式下禁用的 SAI 生成，则在进入 STOP 模式之前应由软件禁用发送器或接收器。当发送器或接收器处于同步模式时，此情况不适用，因为所有同步 SAI 为同时启用和禁用。

48.5.1.3 总线时钟

总线时钟供控制和配置寄存器使用，用于生成同步中断和 DMA 请求。

注

尽管没有指定具体的最小总线时钟频率，但总线时钟频率必须足够快（相对于位时钟频率），以确保 FIFO 运行，并且不产生发送器 FIFO 下溢或接收器 FIFO 溢出情况。

48.5.2 SAI 复位

SAI 将在系统复位时异步复位。SAI 有一个软件复位和一个 FIFO 复位。

48.5.2.1 软件复位

SAI 发送器包括一个软件复位，可以复位所有发送器内部逻辑，包括位时钟生成、状态标志和 FIFO 指针。它不会复位配置寄存器。在被软件清零之前，软件复位将保持置位。

SAI 接收器包括一个软件复位，可以复位所有接收器内部逻辑，包括位时钟生成、状态标志和 FIFO 指针。它不会复位配置寄存器。在被软件清零之前，软件复位将保持置位。

48.5.2.2 FIFO 复位

SAI 发送器包括一个 FIFO 复位，可以将 FIFO 写指针同步为与 FIFO 读指针相同的值。这样就会清空 FIFO 的内容，应在 TCSR[FEF]置位之后，FIFO 重新初始化且 TCSR[FEF]清零之前使用。FIFO 复位只会置位一个周期。

SAI 接收器包括一个 FIFO 复位，可以将 FIFO 读指针同步为与 FIFO 写指针相同的值。这样就会清空 FIFO 的内容，应在 RCSR[FEF]置位且已经从 FIFO 读取全新剩余数据之后，并且在 RCSR[FEF]清零之前使用。FIFO 复位只会置位一个周期。

48.5.3 同步模式

SAI 发送器和接收器可以相互同步运行。

48.5.3.1 同步模式

SAI 发送器和接收器可以配置为用同步位时钟和帧同步运行。

如果发送器和接收器都要使用发送器位时钟和帧同步：

- 必须将发送器配置异步工作模式，将接收器配置为同步工作模式。
- 在同步模式下，只有在同时使能发送器和接收器时，接收器才能使能。
- 建议首先禁用接收器并且最后使能。

如果发送器和接收器都要使用接收器位时钟和帧同步：

- 必须将接收器配置异步工作模式，将发送器配置为同步工作模式。
- 在同步模式下，只有在同时使能接收器和发送器时，接收器才能使能。
- 建议最后使能并且首先禁用接收器。

工作于同步模式下时，只会共享位时钟、帧同步和发送器/接收器使能。否则，即使要同时在发送器和接收器上统一配置配置寄存器，发送器和接收器也会独立工作。

48.5.4 帧同步配置

使能时，SAI 会连续发送"与"或者"或"接收数据帧。每个帧由固定数量的字构成，每个字由固定数量的位构成。在每个帧内部，可以屏蔽任意给定字，使接收器忽略该字，并使发送器在该字持续期间三态。

帧同步信号用于指示各帧的开始。有效的帧同步要求检测到一个上升沿（如果高电平有效）和下降沿（如果低电平有效），并且发送器或接收器不得忙于处理上一帧。在使能发送器或接收器之后的前四个位时钟周期，也会忽略（从机模式）或不生成（主机模式）有效的帧同步。

发送器和接收器帧同步可以用任意下列选项独立配置：

- 外部生成或内部生成
- 高电平有效或低电平有效
- 在帧中首位置位或者提前一位置位
- 在 1 位时钟与首字长之间的持续时间内置位
- 每帧的帧长度为 1 至 16 字
- 每字支持的字长为 8 至 32 位

- 首字长和剩余字长可以独立配置
- 字可以配置为优先发送/接收 MSB 或 LSB

在使能 SAI 发送器或接收器之后，不能更改这些配置选项。

48.5.5 数据 FIFO

每个发送和接收通道都包括一个大小为 8×32 位的 FIFO。通过 SAI 发送/接收数据寄存器访问 FIFO 数据。

48.5.5.1 数据对齐

使用首位移位配置字段（选择首位移位的位指数，即 0-31）可以在 32 位宽寄存器的任意位置对齐 FIFO 中的数据。

支持数据对齐和首位移位配置要求的示例详见图 48-75（LSB 优先配置）和图 48-76（MSB 优先配置）。

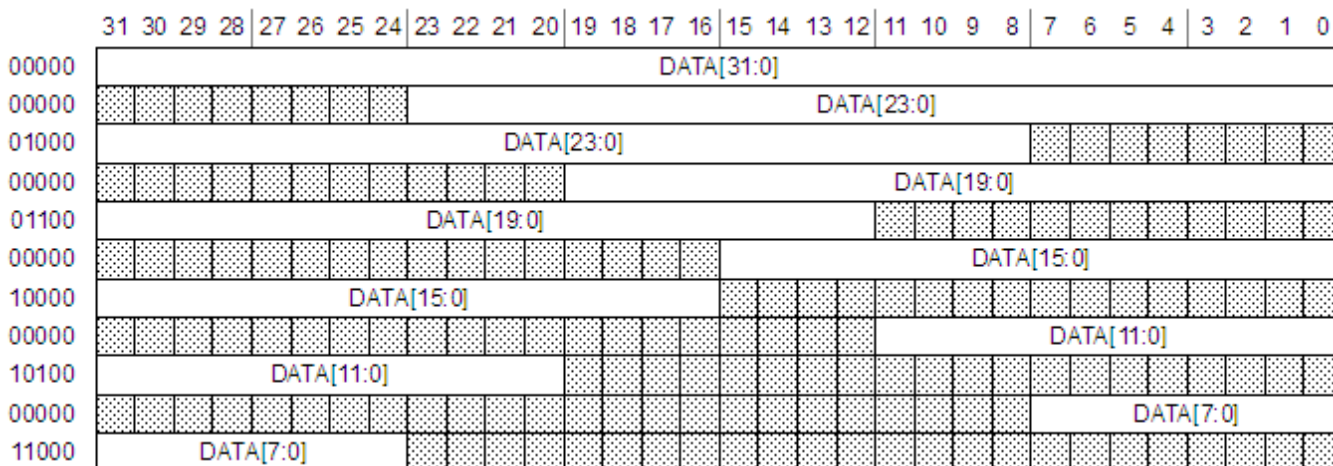


图 48-75. SAI 首位移位，LSB 优先

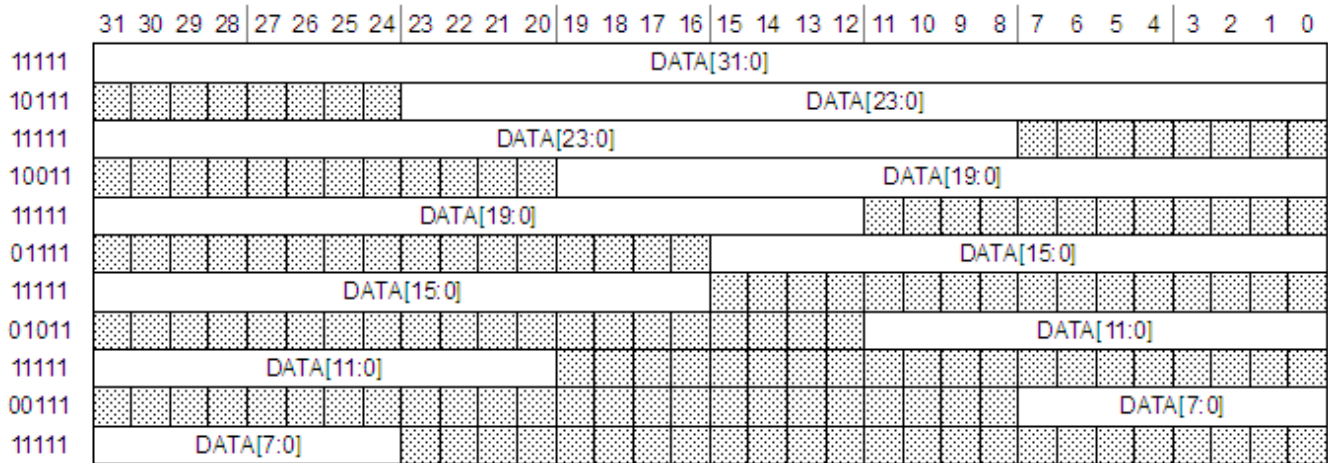


图 48-76. SAI 首位移位, MSB 优先

48.5.5.2 FIFO 指针

写入 TDR 时, 相应 TFR 的 WFP 在每次有效写入后递增。SAI 支持 8 位、16 位和 32 位 TDR 写入, 且 FIFO 指针将在每次单独写入后递增。注意 8 位写入应仅用于高达 8 位数据的传送, 16 位写入应仅用于高达 16 位数据的传送。

若相应的 TCR3[TCE]位清零或 FIFO 为满, 则 TDR 写入被忽略。若传送 FIFO 为空, 则 TDR 写入必须比下一个未屏蔽的字启动至少早三位时钟, 以避免 FIFO 下溢。

读取 RDR 时, 相应 RFP 的 RFR 在每次有效读取后递增。SAI 支持 8 位、16 位和 32 位 RDR 读取, 且 FIFO 指针将在每次单独读取后递增。注意 8 位读取应仅用于高达 8 位数据的接收, 16 位读取应仅用于高达 16 位数据的接收。

若相应的 RCR3[RCE]位清零或 FIFO 为空, 则 RDR 读取被忽略。若接收 FIFO 为满, 则 RDR 读取必须比下一个未屏蔽的字结束至少早三位时钟, 以避免 FIFO 下溢出。

48.5.5.3 FIFO 包装

FIFO 包装支持将多个 8 位或 16 位数据字存储在单个 32 位 FIFO 字中, 以用于发送器"与"或者"或"接收器。可通过调整每个字的位数和每个帧的字数来模拟 (例如, 每帧一个 32 位字与每帧两个 16 位字对比), FIFO 包装不需要每帧有多个字数, 且完全支持字屏蔽。FIFO 包装使能时, FIFO 指针仅在整个 32 位 FIFO 字被软件写入 (发送) 或读取 (接收) 时递增, 支持各帧内的不同字加载/存储在不同存储器区域的场景。

16 位 FIFO 包装使能以进行发送时，在各帧开始且第二个未屏蔽发送字后加载发送移位寄存器。发送的第一个字取自字节偏移\$0 中的 16 位字 (TCFG5[FBT]选择的第一个位必须在此 16 位字中进行配置)，发送的第二个字取自字节偏移\$2 中的 16 位字 (TCSR5[FBT][3:0]选择的第一个位)。一旦 16 位字发送后，发送器将发送逻辑 0，直至下一个字开始。

使能 16 位 FIFO 包装已进行接收时，接收移位寄存器存储在每次接收的第二个未屏蔽字后，若每帧接收的未屏蔽字数量为奇数，则存储在每帧结束时。接收的第一个字存储在字节偏移\$0 中的 16 位字 (RCFG5[FBT]选择的第一个位必须在此 16 位字中进行配置)，接收的第二个字存储在字节偏移\$2 中的 16 位字 (RCSR5[FBT][3:0]选择的第一个位)。一旦接收到 16 位字，接收器将忽略接收的数据，直至下一个字开始。

8 位 FIFO 包装与 16 位包装类似，区别在于有四字加载/存储在每个 32 位 FIFO 字中。第一个字加载/存储在字节偏移\$0 中，第二个字在字节偏移\$1，第三个字在字节偏移\$2，第四个字在字节偏移\$3。TCFG5[FBT]"与"或者"或"RCFG5[FBT]必须在字节偏移\$0 中进行配置。

48.5.6 字掩码寄存器

SAI 传送器和接收器各含有一个字掩码寄存器，即 TMR 和 RMR，可用作在帧中屏蔽任意字。由于字掩码寄存器为双缓冲，软件可在各帧结束前进行更新，用来在下一个帧屏蔽特定字。

TMR 根据每个选定字的长度将传送数据引脚分为三态，且传送 FIFO 不得根据屏蔽字读取。

RMR 会导致每个选定字的接收数据被丢弃，且不得写入接收 FIFO 中。

48.5.7 中断和 DMA 请求

SAI 传送器和接收器生成独立的中断和 DMA 请求，但支持相同的状态标志。生成传送器和接收器中断的异步版本，用来从停止模式中唤醒 CPU。

48.5.7.1 FIFO 请求标志

FIFO 请求标志根据 FIFO 条目数量和 FIFO 深度配置进行置位。

使能的传送 FIFO 条目数量少于或等于传送 FIFO 深度配置时，传送 FIFO 请求标志置位，使能的传送 FIFO 条目数量大于传送 FIFO 深度配置时，传送 FIFO 请求标志清零。

使能的接收 FIFO 条目数量大于接收 FIFO 深度配置时，接收 FIFO 请求标志置位，使能的接收 FIFO 条目数量小于或等于接收 FIFO 深度配置时，接收 FIFO 请求标志清零。

FIFO 请求标志可生成中断或 DMA 请求。

48.5.7.2 FIFO 警告标志

FIFO 警告标志根据 FIFO 条目数量置位。

使能后的传送 FIFO 条目数量为空时，传送警告标志置位；使能后的传送 FIFO 条目数量不为空时，传送警告标志清零。

使能后的接收 FIFO 条目数量为满时，接收警告标志置位；使能后的接收 FIFO 条目数量不为满时，接收警告标志清零。

FIFO 警告标志可生成中断或 DMA 请求。

48.5.7.3 FIFO 错误标志

使能后的传送 FIFO 下溢时，传送 FIFO 错误标志置位。标志置位后，所有的使能传送通道将传送 0，直至 TCSR[FEF]清零且下一个传送帧启动。TCSR[FEF]清零前，所有使能传送 FIFO 必须复位并使用新数据进行初始化。

TCR4[FCONT]置位时，在没有软件干预的情况下，FIFO 将在下溢后继续传送数据。为确保数据传送顺序正确，传送器将采用与帧中导致 FIFO 下溢同样的字数，但仅在新数据已写入传送 FIFO 之后。软件还应清除 TCSR[FEF]标志，但不得重新初始化传送 FIFO。

使能的接收 FIFO 下溢时 RCSR[FEF]置位。置位后，所有使能接收通道均会丢弃接收的数据，知道 RCSR[FEF]清零且下一个接收帧开始。RCSR[FEF]清零前所有使能接收 FIFO 均应为空。

RCR4[FCONT]置位时，在没有软件干预的情况下，FIFO 将在溢出后继续接收数据。为确保数据接收顺序正确，接收器将采用与帧中导致 FIFO 溢出同样的字数，但仅在新数据已写入接收 FIFO 之后。软件还应清除 RCSR[FEF]标志，但不得清空接收 FIFO。

FIFO 错误标志仅能生成中断。

48.5.7.4 同步错误标志

当配置为外部生成帧同步时同步错误标志（TCSR[SEF]或RCSR[SEF]）置位，且发送器或接收器忙于之前的帧时外部帧同步置位。外部帧同步置位被忽略，且同步错误标志置位。同步错误标志置位时，传送机或接收机在每个帧空闲或结束时继续检查帧同步置位

同步错误标志仅能生成中断。

48.5.7.5 字开始标志

根据字标志寄存器字段的配置，字开始标志在选定字的第二位时钟开始处置位。

字开始标志仅能生成中断。

第 49 章

通用输入/输出 (GPIO)

49.1 此模块的芯片实现细节

49.1.1 GPIO 信号数量

[可订购部件编号](#)对本文档中所支持的器件上的 GPIO 数量做了详细的说明。

8 个 GPIO 引脚支持高驱动力模式 - PTB0、PTB1、PTD4、PTD5、PTD6、PTD7、PTC3 和 PTC4。所有其他 GPIO 都只支持常规驱动选项。

PTA4 包含一个由 PORTA_PCR4[PFE]控件使能或禁用的无源输入滤波器。复位后此功能是默认禁止的。

49.2 简介

通用输入和输出(GPIO)模块可通过外设总线访问，还能通过零等待状态接口 (IOPORT)与处理器内核通信，实现最高的引脚性能。GPIO 寄存器支持 8 位、16 位或 32 位访问。

当引脚配置为用于 GPIO 功能时，GPIO 数据方向和输出数据寄存器控制每个引脚的方向和输出数据。假设引脚相应的端口控制和中断模块已使能，则当引脚配置为任意数字功能时，GPIO 输入数据寄存器显示每个引脚上的逻辑值。

位带操作以及置位，清零寄存器的设计可以方便的支持用户对某个比特位的置位，清零以及翻转操作。

49.2.1 特性

GPIO 模块的特性包括：

- 端口数据输入寄存器适用于所有数字引脚多路复用模式
- 端口数据输出寄存器具有相应的置位/清零/切换寄存器
- 端口数据方向寄存器

注

GPIO 模块的时钟源为系统时钟。

49.2.2 工作模式

下表介绍了 GPIO 模块的不同工作模式及其在这些模式下的行为。

表 49-1. 工作模式

工作模式	说明
运行	GPIO 模块正常运行。
等待	GPIO 模块正常运行。
停止	GPIO 模块被禁用。
调试	GPIO 模块正常运行。

49.2.3 GPIO 信号说明

表 49-2. GPIO 信号说明

GPIO 信号说明	说明	I/O
PORTA31–PORTA0	通用输入/输出	I/O
PORTB31–PORTB0	通用输入/输出	I/O
PORTC31–PORTC0	通用输入/输出	I/O
PORTD31–PORTD0	通用输入/输出	I/O
PORTE31–PORTE0	通用输入/输出	I/O

注

并非每个端口中的所有引脚都配置于每个器件上。有关设备上的可用 GPIO 端口数量信息，请参见信号复用一章。

49.2.3.1 详细信号说明

表 49-3. GPIO 接口详细信号说明

信号	I/O	说明	
PORTA31–PORTA0	I/O	通用输入/输出	
PORTB31–PORTB0		状态含义	置位：该引脚为逻辑 1。

下一页继续介绍此表...

表 49-3. GPIO 接口详细信号说明 (继续)

信号	I/O	说明
PORTC31–PORTC0 PORTD31–PORTD0 PORTE31–PORTE0		解除置位: 该引脚为逻辑 0。 有效电平: 输出时, 该信号发生在系统时钟的上升沿。就输入而言, 它可能在任何时刻发生, 且输入信号可能和系统时钟不同步。 无效电平: 输出时, 该信号发生在系统时钟的上升沿。就输入而言, 可能在任何时刻发生, 且输入信号可能和系统时钟不同步。
	时序	

注

并非每个端口内的所有引脚都会在设备上实施。有关器件的可用 GPIO 端口数量, 请参见信号复用的相关章节。

49.3 存储器映射和寄存器定义

在有效存储器映射外对 GPIO 存储器空间进行任何读写访问都将导致总线错误。

GPIO 存储器映射

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
400F_F000	端口数据输出寄存器 (GPIOA_PDOR)	32	R/W	0000_0000h	49.3.1/1256
400F_F004	端口置位输出寄存器 (GPIOA_PSOR)	32	W (始终读 0)	0000_0000h	49.3.2/1257
400F_F008	端口清零输出寄存器 (GPIOA_PCOR)	32	W (始终读 0)	0000_0000h	49.3.3/1257
400F_F00C	端口翻转输出寄存器 (GPIOA_PTOR)	32	W (始终读 0)	0000_0000h	49.3.4/1258
400F_F010	端口数据输入寄存器 (GPIOA_PDIR)	32	R	0000_0000h	49.3.5/1258
400F_F014	端口数据方向寄存器 (GPIOA_PDDR)	32	R/W	0000_0000h	49.3.6/1259
400F_F040	端口数据输出寄存器 (GPIOB_PDOR)	32	R/W	0000_0000h	49.3.1/1256
400F_F044	端口置位输出寄存器 (GPIOB_PSOR)	32	W (始终读 0)	0000_0000h	49.3.2/1257
400F_F048	端口清零输出寄存器 (GPIOB_PCOR)	32	W (始终读 0)	0000_0000h	49.3.3/1257
400F_F04C	端口翻转输出寄存器 (GPIOB_PTOR)	32	W (始终读 0)	0000_0000h	49.3.4/1258
400F_F050	端口数据输入寄存器 (GPIOB_PDIR)	32	R	0000_0000h	49.3.5/1258
400F_F054	端口数据方向寄存器 (GPIOB_PDDR)	32	R/W	0000_0000h	49.3.6/1259
400F_F080	端口数据输出寄存器 (GPIOC_PDOR)	32	R/W	0000_0000h	49.3.1/1256

下一页继续介绍此表...

GPIO 存储器映射 (继续)

绝对地址 (十六进制)	寄存器名称	宽度 (单位: 位)	访问	复位值	小节/页
400F_F084	端口置位输出寄存器 (GPIOC_PSOR)	32	W (始终读 0)	0000_0000h	49.3.2/1257
400F_F088	端口清零输出寄存器 (GPIOC_PCOR)	32	W (始终读 0)	0000_0000h	49.3.3/1257
400F_F08C	端口翻转输出寄存器 (GPIOC_PTOR)	32	W (始终读 0)	0000_0000h	49.3.4/1258
400F_F090	端口数据输入寄存器 (GPIOC_PDIR)	32	R	0000_0000h	49.3.5/1258
400F_F094	端口数据方向寄存器 (GPIOC_PDDR)	32	R/W	0000_0000h	49.3.6/1259
400F_F0C0	端口数据输出寄存器 (GPIOD_PDOR)	32	R/W	0000_0000h	49.3.1/1256
400F_F0C4	端口置位输出寄存器 (GPIOD_PSOR)	32	W (始终读 0)	0000_0000h	49.3.2/1257
400F_F0C8	端口清零输出寄存器 (GPIOD_PCOR)	32	W (始终读 0)	0000_0000h	49.3.3/1257
400F_F0CC	端口翻转输出寄存器 (GPIOD_PTOR)	32	W (始终读 0)	0000_0000h	49.3.4/1258
400F_F0D0	端口数据输入寄存器 (GPIOD_PDIR)	32	R	0000_0000h	49.3.5/1258
400F_F0D4	端口数据方向寄存器 (GPIOD_PDDR)	32	R/W	0000_0000h	49.3.6/1259
400F_F100	端口数据输出寄存器 (GPIOE_PDOR)	32	R/W	0000_0000h	49.3.1/1256
400F_F104	端口置位输出寄存器 (GPIOE_PSOR)	32	W (始终读 0)	0000_0000h	49.3.2/1257
400F_F108	端口清零输出寄存器 (GPIOE_PCOR)	32	W (始终读 0)	0000_0000h	49.3.3/1257
400F_F10C	端口翻转输出寄存器 (GPIOE_PTOR)	32	W (始终读 0)	0000_0000h	49.3.4/1258
400F_F110	端口数据输入寄存器 (GPIOE_PDIR)	32	R	0000_0000h	49.3.5/1258
400F_F114	端口数据方向寄存器 (GPIOE_PDDR)	32	R/W	0000_0000h	49.3.6/1259

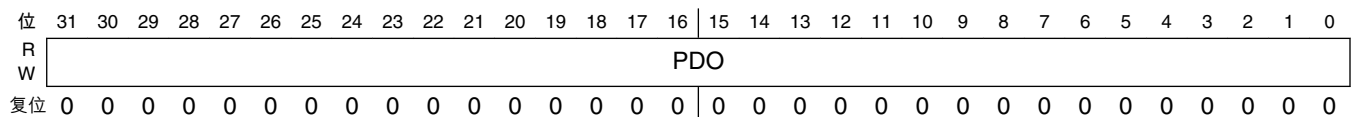
49.3.1 端口数据输出寄存器 (GPIOx_PDOR)

该寄存器配置在各通用输出引脚上驱动的逻辑电平。

注

请勿修改选定封装中未绑定引脚的配置寄存器。封装中未绑定的引脚将默认处于禁用状态，以实现最低功耗。

地址: 基址 基准 + 0h 偏移



GPIOx_PDOR 字段描述

字段	描述
PDO	<p>端口数据输出</p> <p>读取未绑定引脚的寄存器位将返回无效值。</p> <p>0 如果引脚配置为通用输出，则在引脚上驱动逻辑电平 0。 1 如果引脚配置为通用输出，则在引脚上驱动逻辑电平 1。</p>

49.3.2 端口置位输出寄存器 (GPIOx_PSOR)

该寄存器可以将 PDOR 字段置位。

地址: 基址 基准 + 4h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTSO																															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

GPIOx_PSOR 字段描述

字段	描述
PTSO	<p>端口置位输出</p> <p>如下所示对该寄存器进行写操作将更新 PDOR 中对应位的内容：</p> <p>0 PDORn 中的对应位不变。 1 PDORn 中的对应位设置为逻辑 1。</p>

49.3.3 端口清零输出寄存器 (GPIOx_PCOR)

该寄存器可以将 PDOR 字段清零。

地址: 基址 基准 + 8h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTCO																															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

GPIOx_PCOR 字段描述

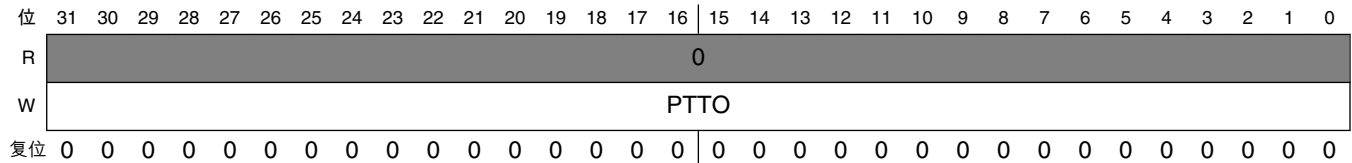
字段	描述
PTCO	<p>端口清零输出</p> <p>如下所示对该寄存器进行写操作将更新端口数据输出寄存器(PDOR)中对应位的内容：</p>

GPIOx_PCOR 字段描述 (继续)

字段	描述
0	PDORn 中的对应位不变。
1	PDORn 中的对应位清零为逻辑 0。

49.3.4 端口翻转输出寄存器 (GPIOx_PTOR)

地址: 基址 基准 + Ch 偏移



GPIOx_PTOR 字段描述

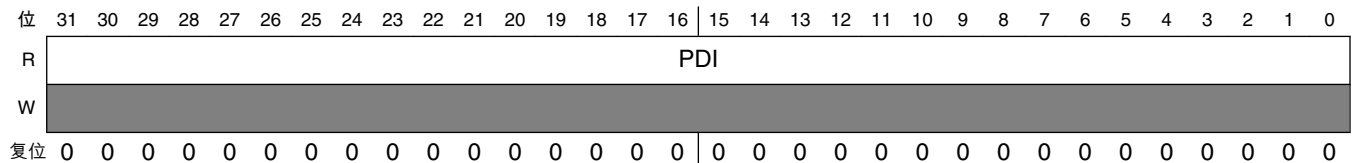
字段	描述
PTTO	<p>端口翻转输出</p> <p>如下所示对该寄存器进行写操作将更新 PDOR 中对应位的内容：</p> <p>0 PDORn 中的对应位不变。</p> <p>1 PDORn 中的对应位置为当前逻辑状态的反相电平。</p>

49.3.5 端口数据输入寄存器 (GPIOx_PDIR)

注

请勿修改选定封装中未绑定引脚的配置寄存器。封装中未绑定的引脚将默认处于禁用状态，以实现最低功耗。

地址: 基址 基准 + 10h 偏移



GPIOx_PDIR 字段描述

字段	描述
PDI	<p>端口数据输入</p> <p>读取器件的未绑定引脚将返回 0。读取未配置为数字功能的引脚将返回 0。如果对应的 PORT 模块未使能，那么 PDIR 中的对应位不更新。</p>

GPIOx_PDIR 字段描述 (继续)

字段	描述
0	引脚逻辑电平为逻辑 0，或者未配置为数字功能。
1	引脚逻辑电平为逻辑 1。

49.3.6 端口数据方向寄存器 (GPIOx_PDDR)

PDDR 将各个端口引脚配置为输入或输出。

地址: 基址 基准 + 14h 偏移

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDD																															
W	PDD																															
复位	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PDDR 字段描述

字段	描述
PDD	<p>端口数据方向</p> <p>将各个端口引脚配置为输入或输出。</p> <p>0 引脚配置为通用输入，用于 GPIO 功能。</p> <p>1 引脚配置为通用输出，用于 GPIO 功能。</p>

49.4 功能说明

49.4.1 通用输入

当引脚配置为数字功能，且对应的端口控制和中断模块已使能时，可通过端口数据输入寄存器得到该引脚的逻辑状态。

在启用端口控制和中断模块中的任何数字过滤器后，端口数据输入寄存器将返回同步引脚状态。输入引脚同步器与端口控制和中断模块保持同步，所以如果对应的端口控制和中断模块被禁用，同步器也会被禁用。当通用输入功能不需要某个端口时，此功能有助于降低功耗。

49.4.2 通用输出

当引脚配置为 GPIO 功能时，可通过端口数据输出寄存器和端口数据方向寄存器来控制每个引脚的逻辑状态。下表介绍了如何把引脚配置为输入/输出。

条件	结果
引脚已配置为 GPIO，并且已将对应的端口数据方向寄存器位清零。	引脚配置为输入。
引脚已配置为 GPIO，并且已将对应的端口数据方向寄存器位置位。	引脚配置为输出，并且引脚的逻辑状态与对应的端口数据输出寄存器相同。

为了让用户可以对通用输出模块实现高效的位操作，我们提供了端口置位、端口清零、以及端口翻转位寄存器，以便通过单次寄存器的写操作来置位、清零或切换一个端口内的一个或多个输出信号。

无需启用对应的端口控制和中断模块来操作端口数据方向寄存器和端口数据输出寄存器（包括置位/清零/跳变寄存器）的状态。

第 50 章

JTAG 控制器(JTAGC)

50.1 简介

JTAGC 数据块提供测试芯片功能和连接性的手段, 同时在不处于测试模式时对系统逻辑保持透明。采用边界扫描技术执行测试(如 IEEE 1149.1-2001 标准定义)。所有输入/输出 JTAGC 数据块的数据都按串行格式传输。

50.1.1 结构框图

以下是 JTAG 控制器(JTAGC)块的简化结构框图。更多有关 JTAGC 寄存器的信息, 请参考芯片特性的配置信息及 [寄存器说明](#)。

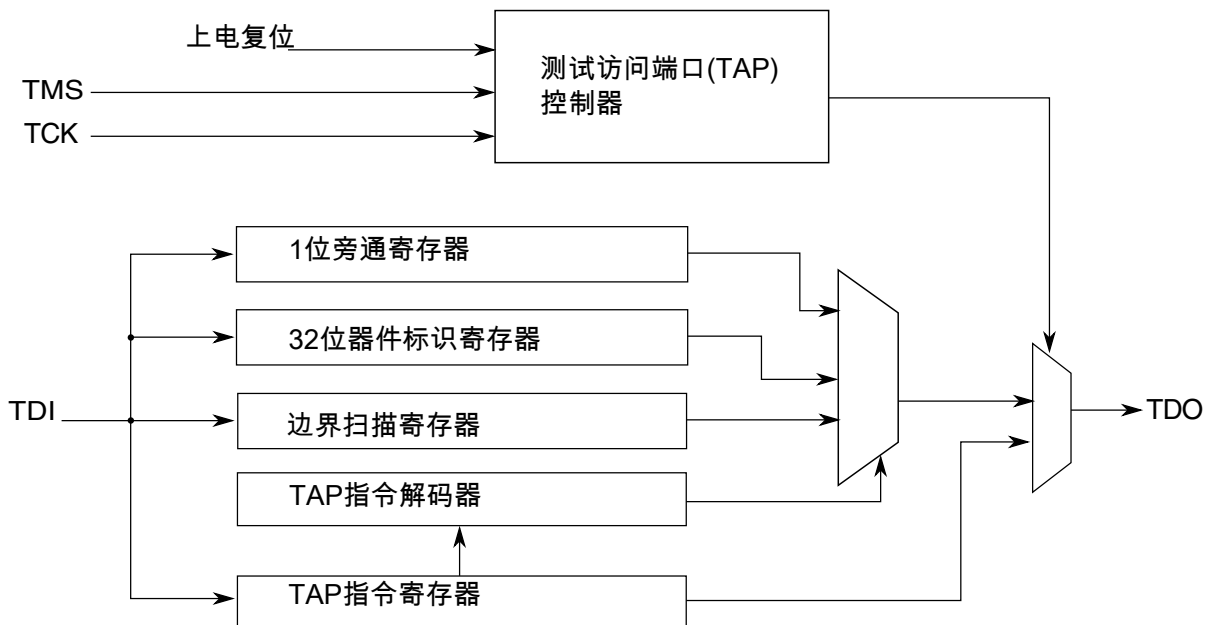


图 50-1. JTAG (IEEE 1149.1) 结构框图

50.1.2 特性

JTAGC 结构块符合 IEEE 1149.1-2001 标准，支持以下功能：

- IEEE 1149.1-2001 测试访问端口(TAP)接口
 - 4 个引脚 (TDI、TMS、TCK 和 TDO)
- 支持多个 IEEE 1149.1-2001 定义指令以及多种公共和专用器件特定指令的指令寄存器。参考表 50-3 查看受支持指令的列表。
- 旁通寄存器，边界扫描寄存器，以及器件标识寄存器。
- 控制数据寄存器、指令寄存器及关联电路运行的 TAP 控制器状态机。

50.1.3 操作模式

JTAGC 结构块将上电复位指示用作其主要复位信号。支持多种 IEEE 1149.1-2001 定义的测试模式及旁路模式。

50.1.3.1 复位

当上电复位被认定 或者 TMS 输入高电平并保持足够长时间，让 TCK 的连续上升边沿能够使得 TAP 控制器的状态机有序进入测试-逻辑-复位状态时，JTAGC 结构块置于复位状态。让 TCK 的五个连续上升边沿的 TMS 保持较高值，则无论 TAP 控制器当前处于何种状态，都保证可以进入测试-逻辑-复位状态。认定上电复位所需的值，会导致异步进入复位状态。处于复位状态时，发生以下操作：

- TAP 控制器将强制进入测试-逻辑-复位状态，从而禁用测试逻辑，并允许片上系统逻辑继续无阻碍地正常工作
- 另外，指令寄存器会加载 IDCODE 指令。

50.1.3.2 IEEE 1149.1-2001 定义测试模式

JTAGC 数据块支持多种 IEEE 1149.1-2001 定义的测试模式。通过在使能 JTAGC 的条件下，将相应的指令载入指令寄存器实现测试模式的选择。支持的测试指令包括 EXTEST、HIGHZ、CLAMP、SAMPLE 和 SAMPLE/PRELOAD。每条指令均定义了可在指令为最新状态时工作并与片上系统逻辑交互的数据寄存器集合。对于每条指令，仅使能一条测试数据寄存器路径，以在 TDI 和 TDO 之间转移数据。

使能边界扫描寄存器，以便在 EXTEST、SAMPLE 或 SAMPLE/PRELOAD 指令有效时在 TDI 和 TDO 之间实现串行访问。使能单位旁通寄存器移位级，以便在 BYPASS、HIGHZ、CLAMP 或保留指令有效时在 TDI 和 TDO 之间实现串行访问。各测试模式的功能详见 [JTAGC 数据块指令](#)。

50.1.3.3 旁通模式

在无需测试操作时，可以加载 BYPASS 指令，把 JTAGC 数据块置于旁通模式。处于旁通模式时，单位旁通移位寄存器用于提供最小长度串行路径，以在 TDI 和 TDO 之间转移数据。

50.2 外部信号说明

JTAGC 由一组连接片外开发工具并允许访问测试支持功能的信号组成。下表列出了 JTAGC 信号，下面的章节会予以说明。

表 50-1. JTAG 信号属性

名称	I/O	功能	复位状态	上/下拉
TCK	输入	测试时钟。	—	下拉
TDI	输入	测试数据输入	—	上拉
TDO	输出	测试数据输出	高 Z	—
TMS	输入	测试模式选择	—	上拉

50.2.1 TCK—测试时钟输入

测试时钟输入 (TCK) 是一种借助 TAP 来同步测试逻辑和控制寄存器访问的输入引脚。

50.2.2 TDI—测试数据输入

测试数据输入 (TDI) 是一种接收序列测试指令和数据的输入引脚。TDI 在 TCK 的上升沿采样。

50.2.3 TDO—测试数据输出

测试数据输出(TDO)是一种发送测试指令和数据序列输出的输出引脚。TDO 具备三态，只有在 TAP 控制器状态机处于 Shift-IR 和 Shift-DR 状态时才能有效驱动，请参见 [TAP 控制器状态机](#) 中的描述。

50.2.4 TMS—测试模式选择

测试模式选择(TMS)是一个输入引脚，用于排定 IEEE 1149.1-2001 测试控制状态机的序列。TMS 在 TCK 的上升沿采样。

50.3 寄存器说明

本节详细说明可通过 TAP 接口访问的 JTAGC 数据块寄存器，包括数据寄存器和指令寄存器。其中包括各位级的说明以及各寄存器的复位状态。这些寄存器未进行存储器映射，只能通过 TAP 访问。

50.3.1 指令寄存器

JTAGC 数据块使用一个 4 位指令寄存器，如下图所示。该指令寄存器允许将指令载入数据块，以选择要执行的测试指令或者要访问的测试指令数据寄存器，或者两者同时选择。指令在 TAP 控制器处于 Shift-IR 状态时通过 TDI 移入，并在 Update-IR 状态下在 TCK 的下降沿锁存。只能在 Update-IR 和 Test-Logic-Reset TAP 控制器状态下更改锁存的指令值。同步进入 Test-Logic-Reset 状态会导致 IDCODE 指令在 TCK 的下降沿加载。异步进入 Test-Logic-Reset 状态会导致 IDCODE 指令异步加载。在 Capture-IR TAP 控制器状态下，指令移位寄存器载入值 0001b，使该值在 TAP 控制器序列进入 Shift-IR 状态时成为寄存器的读取值。

	3	2	1	0
R	0	0	0	1
W	指令代码			
复位：	0	0	0	1

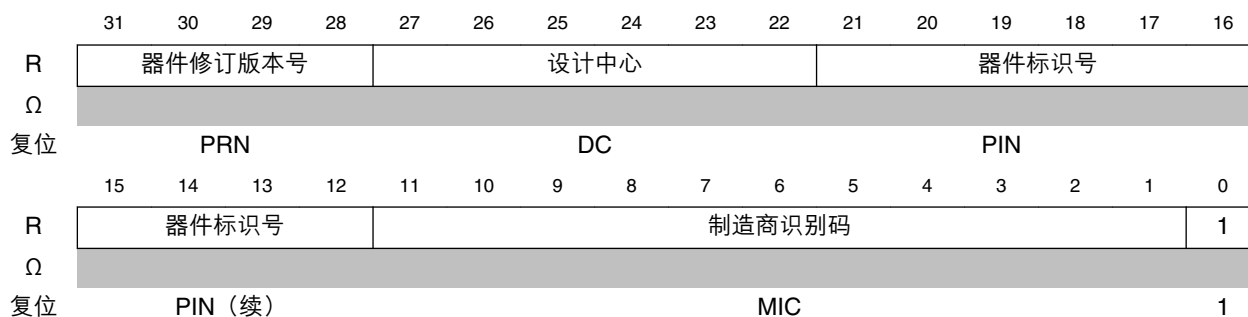
图 50-2. 指令寄存器

50.3.2 旁通寄存器

旁通寄存器是选定的单位移位寄存器路径，当 BYPASS、CLAMP、HIGHZ 或保留指令有效时，可在 TDI 与 TDO 之间进行串行数据传输。在进入 Capture-DR 状态之后，该单位移位寄存器设置为逻辑 0。因此，在选择旁通寄存器之后移出的首位始终为逻辑 0。

50.3.3 器件标识寄存器

器件标识(JTAG ID)寄存器（如下图所示）允许通过 TAP 确定器件的修订版本号、器件型号、制造商和负责器件设计的数据中心。选择器件标识寄存器，以用于在读取 IDCODE 指令有效时，在 TDI 与 TDO 之间进行串行数据传输。在器件标识寄存器被选择时进入 Capture-DR 状态，IDCODE 载入要在 Shift-DR 状态下在 TDO 时移出的移位寄存器中。在 Update-DR 状态下不会发生操作。



下表描述了器件标识寄存器的功能。

表 50-2. 器件标识寄存器字段说明

字段	说明
PRN	器件修订版本号。含有器件的修订版本号。值为 0x0。
DC	设计中心。表示设计中心。值为 0x26。
PIN	器件标识号。含有器件编号。
MIC	制造商识别码。含有简化版电子元件工业联合会(JEDEC) ID。值为 0x00E。
IDCODE ID	IDCODE 寄存器 ID。将此寄存器识别为器件标识寄存器而非旁通寄存器。始终置 1。

50.3.4 边界扫描寄存器

在 EXTEST、SAMPLE 或 SAMPLE/PRELOAD 指令有效时，边界扫描寄存器连接在 TDI 和 TDO 之间。它用于捕捉输入引脚数据、强迫输出引脚采用固定数值，并为双向引脚选择逻辑值和方向。边界扫描寄存器的每个位都代表一个独立的边界扫描寄存器单元，如 IEEE 1149.1-2001 标准和[边界扫描](#) 所述。边界扫描寄存器的大小和位序由器件决定，具体请参考器件 BSDL 文件。

50.4 功能说明

本节解释 JTAGC 功能说明。

50.4.1 JTAGC 复位配置

处于复位状态时，TAP 控制器将强制进入测试-逻辑-复位状态，从而禁用测试逻辑，并允许片上系统逻辑正常工作。另外，指令寄存器会加载 IDCODE 指令。

50.4.2 IEEE 1149.1-2001 (JTAG)测试接入端口

JTAGC 数据块用 IEEE 1149.1-2001 TAP 来访问寄存器。该端口可与 MCU 上的其他 TAP 控制器共享。该端口的所有权取决于当前加载的指令的值。

数据在 TDI 与 TDO 之间移位，只是所选寄存器以最低有效位开始，如下图所示。这种情况适用于指令寄存器、测试数据寄存器和旁通寄存器。

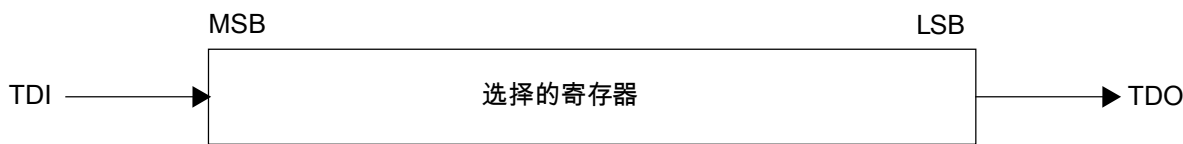
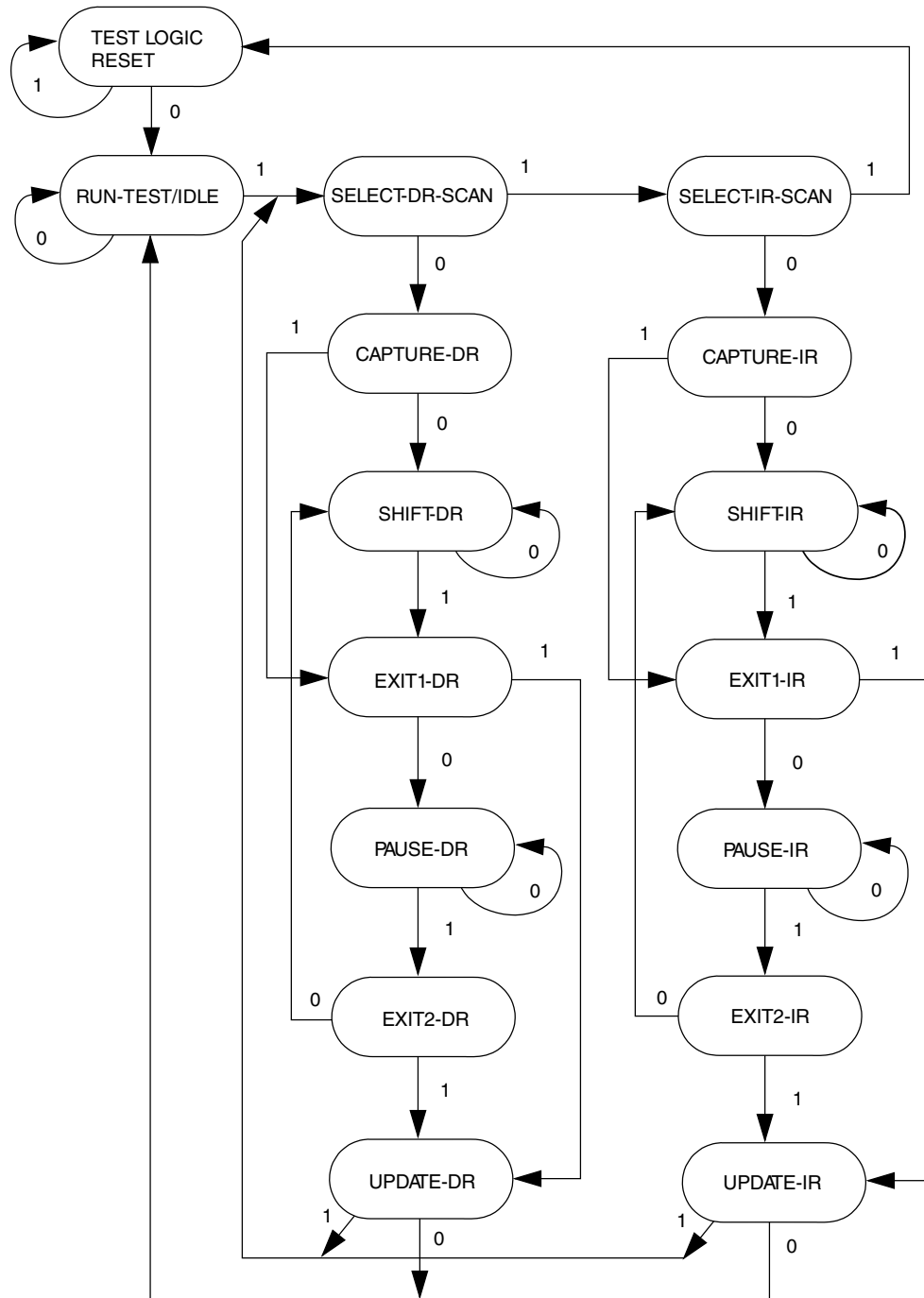


图 50-3. 通过寄存器进行数据移位

50.4.3 TAP 控制器状态机

TAP 控制器是一个同步状态机，负责解析 TMS 引脚上的逻辑值序列。下图展示了状态机的各种状态。每个状态旁边显示的值是在 TCK 信号上升沿采样的 TMS 信号的值。如下图所示，使 TMS 保持于逻辑 1，同时通过足够数量的上升沿对 TCK 计时，也会导致状态机进入测试-逻辑-复位状态。



本图中各状态转换旁边的值表示TMS在TCK上升沿时间的值。

图 50-4. IEEE 1149.1-2001 TAP 控制器有限状态机

50.4.3.1 使能 TAP 控制器

JTAGC TAP 控制器通过将 JTAGC 使能 设为逻辑值 1 来使能。

50.4.3.2 选择一个 IEEE 1149.1-2001 寄存器

在 JTAGC 使能的情况下，通过在指令寄存器中加载任意 JTAGC 数据块指令即可实现对 JTAGC 数据寄存器的访问。指令通过 Select-IR-Scan 路径移入，并在 Update-IR 状态下加载。此时，所有数据寄存器访问都是通过 Select-DR-Scan 路径来实施的。

Select-DR-Scan 路径用于读取或写入寄存器数据，其方法是在 Shift-DR 寄存器下移入数据 (LSB 优先)。在读取寄存器时，寄存器值将在 Capture-DR 状态下载入 IEEE 1149.1-2001 移位器。在写入寄存器时，值将在 Update-DR 状态下从 IEEE 1149.1-2001 移位器载入寄存器。在读取寄存器时，无需移出寄存器的全部内容。获得所需数量的位数时，可以终止移位。

50.4.4 JTAGC 数据块指令

JTAGC 数据块部署的是 IEEE 1149.1-2001 规定的指令 (列于芯片特性调试信息中)。详情请参见 IEEE 1149.1-2001 标准。

JTAGC 数据块部署的是 IEEE 1149.1-2001 规定的指令 (列于下表中)。本节将概要介绍每条指令；详情请参见 IEEE 1149.1-2001 标准。所有未定义的操作码均保留。

表 50-3. 4 位 JTAG 指令

指令	代码[3:0]	指令汇总
IDCODE	0000	选择器件标识寄存器，以便移位
SAMPLE/PRELOAD	0010	选择边界扫描寄存器，以便移位、采样和预载，而不影响功能运行
SAMPLE	0011	选择边界扫描寄存器，以便移位和采样，而不影响功能运行
EXTEST	0100	选择边界扫描寄存器并将预加载的值应用到输出引脚。 注：执行该指令会置位重置功能。
保留出厂调试	0101	仅供出厂调试使用
保留出厂调试	0110	仅供出厂调试使用
保留出厂调试	0111	仅供出厂调试使用
保留 ARM JTAG-DP	1000	该指令控制 ARM JTAG-DP 控制器。详情请参见 ARM JTAG-DP 文档。
HIGHZ	1001	选择旁通寄存器并三态所有输出引脚。 注：执行该指令会置位重置功能。
保留 ARM JTAG-DP	1010	该指令控制 ARM JTAG-DP 控制器。详情请参见 ARM JTAG-DP 文档。
保留 ARM JTAG-DP	1011	该指令控制 ARM JTAG-DP 控制器。详情请参见 ARM JTAG-DP 文档。
CLAMP	1100	选择旁通寄存器并将预加载的值应用到输出引脚。

下一页继续介绍此表...

表 50-3. 4 位 JTAG 指令 (继续)

指令	代码[3:0]	指令汇总
		注：执行该指令会置位重置功能。
保留 ARM JTAG-DP	1110	该指令控制 ARM JTAG-DP 控制器。详情请参见 ARM JTAG-DP 文档。
BYPASS	1111	为数据操作选择旁通寄存器

50.4.4.1 IDCODE 指令

IDCODE 选择 32 位器件标识寄存器，作为 TDI 和 TDO 之间的移位路径。该指令允许询问 MCU 以确定其版本号和其他器件标识数据。IDCODE 为在 JTAGC 数据块复位时置于指令寄存器中的指令。

50.4.4.2 SAMPLE/PRELOAD 指令

SAMPLE/PRELOAD 指令有两个功能：

- 指令的 SAMPLE 部分含有系统数据以及控制信号的样本，该样本位于 MCU 输入引脚端，刚好在输出引脚端边界扫描寄存器单元之前。在 SAMPLE/PRELOAD 指令有效时，该采样在 Capture-DR 状态下发生于 TCK 的上升沿。查看采样数据时，要在 Shift-DR 状态下通过边界扫描寄存器将其移到 TDO 输出端。数据捕获和移位操作对系统操作都是透明的。
- 指令的 PRELOAD 部分在选择 EXTEST 或 CLAMP 指令之前初始化边界扫描寄存器单元，以执行边界扫描测试。此目的通过在 Shift-DR 状态下将初始化数据移入边界扫描寄存器来实现。初始化数据将在 Update-DR 下在 TCK 的下降沿传输到边界扫描寄存器单元的并行输出端。这些数据通过 EXTEST 或 CLAMP 指令应用于外部输出引脚。系统操作不受影响。

50.4.4.3 SAMPLE 指令

SAMPLE 指令含有系统数据以及控制信号的样本，该样本位于 MCU 输入引脚端，刚好在输出引脚端边界扫描寄存器单元之前。在 SAMPLE 指令有效时，该采样在 Capture-DR 状态下发生于 TCK 的上升沿。查看采样数据时，要在 Shift-DR 状态下通过边界扫描寄存器将其移到 TDO 输出端。Update-DR 状态下无规定的操作。数据捕获和移位操作对系统操作都是透明的。

50.4.4.4 EXTEST 外部测试指令

EXTEST 选择边界扫描寄存器，作为 TDI 与 TDO 之间的移位路径。通过将边界扫描寄存器中的预加载数据驱动到系统输出引脚上，实现对片外电路和板级连接的测试。一般地，预加载数据会在选择 EXTEST 之前，通过 SAMPLE/PRELOAD 指令载入边界扫描寄存器中。EXTEST 会置位 MCU 的内部系统复位，以强制实现可预测的内部状态，同时执行外部边界扫描操作。

50.4.4.5 HIGHZ 指令

HIGHZ 选择旁通寄存器，作为 TDI 与 TDO 之间的移位路径。当 HIGHZ 有效时，所有输出驱动器都进入无效驱动状态（如高阻抗）。HIGHZ 也会为 MCU 置位内部系统复位，以强制进入可预测的内部状态。

50.4.4.6 CLAMP 指令

CLAMP 允许从边界扫描寄存器确定从 MCU 引脚驱动的信号状态，同时，旁通寄存器被选为 TDI 与 TDO 之间的串行路径。CLAMP 可使总移位路径减至 1 位（旁通寄存器）并通过边界扫描寄存器执行 EXTEST 型指令，从而增强测试效率。CLAMP 也会为 MCU 置位内部系统复位，以强制进入可预测的内部状态。

50.4.4.7 BYPASS 指令

BYPASS 选择旁通寄存器，在 TDI 与 TDO 之间形成一条单位移位寄存器路径。BYPASS 可在无需进行 MCU 测试操作时，缩短总移位路径，从而增强测试效率。这样就可以更加快速地从电路板上执行测试功能所需要的其他元件来回移动测试数据了。当 BYPASS 指令有效时，系统逻辑将正常运行。

50.4.5 边界扫描

通过边界扫描技术可以通过与各面板关联的移位寄存器级，控制和观察处于元件边界的信号。每个级都是更大的边界扫描寄存器单元的一部分，各面板单元以串行方式互联，在设计边界周围形成一个移位寄存器链。边界扫描寄存器由此移位寄存器链构成，当加载 EXTEST、SAMPLE 或 SAMPLE/PRELOAD 指令时，此寄存器将在 TDI 与 TDO 之间连接起来。移位寄存器链包含一个串行输入和串行输出以及时钟和控制信号。

50.5 初始化/应用信息

测试逻辑是一种静态逻辑设计，无论 TCK 处于高电平或者低电平，停止 TCK 时钟都不会造成任何数据丢失。但是，系统时钟不会从内部与 TCK 同步。使用测试逻辑和系统功能逻辑的混合操作需要实现外部同步。

要初始化 JTAGC 结构块并使能对寄存器的访问，则需遵从以下顺序：

1. 通过 TMS 控制的 TAP 控制器状态机转换将 JTAGC 置于复位状态
2. 为测试或将执行的操作加载正确的指令

附录 A

修订历史记录

下表列出本文的修订历史记录。

表 A-1. 修订历史记录

修订版本号	日期	重大变更
2	2015 年 12 月	初次公开发行版。
3	2016 年 5 月	增加 48-pin QFN 信息。



联系方式:

主页:
nxp.com

网络支持:
nxp.com/support

本文档中的信息仅供系统和软件实施方使用 NXP 产品。本文并未明示或暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。NXP 保留对本文档中所述任何产品进行更改的权利，恕不另行通知。

NXP 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用或使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性或附带性损害在内的所有责任。NXP 的数据表和/或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。All operating parameters, including “typicals”, must be validated for each customer application by customer's technical experts. NXP 未转让与其专利权及其他权利相关的许可。NXP 销售产品时遵循以下网址中包含的标准销售条款和条件: nxp.com/SalesTermsandConditions。

NXP, and the NXP logo, Freescale, the Freescale logo CodeTest, CodeWarrior, ColdFire, ColdFire+, Energy Efficient Solutions logo, and Kinetis, mobileGT, Processor Expert, Qorivva, and Symphony are trademarks of NXP B.V. Airfast, BeeKit, BeeStack, CoreNet, Flexis, MagniV, MXC, Platform in a Package, Ready Play, SafeAssure, SafeAssure logo, SMARTMOS, Tower, TurboLink, and Xtrinsic are trademarks. Vybrid is a trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. ARMnnn is the trademark of ARM Limited.

©2016 NXP B.V.

Document Number KS22P100M120SF0RM
Revision 3, May 2016

