Freescale Semiconductor, Inc.
Application Note

Document Number: AN5013
Rev. 1, 10/2014

Kinetis MK24FN1M to MK24FN256 Migration Guide

1 Purpose and overview

This document describes the details of migrating from Kinetis MK24FN1MF-120 MHz to MK24FN256-120 MHz microcontrollers. Migrating between the two devices can require hardware and/or software changes in some cases. Changes that might be required are described in this document.

Contents

1.1 Part numbering and mask set information

Table 1 lists all of the Kinetis MK24FN1M 120 MHz rev 1.1 and MK24FN256 part numbers we are comparing with this document.

Table 1. Part number migration

Part Number Origin	Part Number Destination
MK24FN1M0VDC12 (MK24FN1M) 121 M	MK24FN256VDC12 (MK24FN256)

2 General description

This document is focused on migration between 120 MHz MK24FN1M and 120 MHz MK24FN256 devices. This document describes migration between Kinetis MK24FN1M-120 MHz with up to 1 MB of Flash and Kinetis MK24FN256-120 with up to 256 KB of Flash devices. For simplicity, the document will refer to these devices as **MK24FN1M** and **MK24FN256**.

This document is focused on addressing the changes in functionality between these two Kinetis MCUs.

“Tips and Tricks” are included in this migration guide to aid you in the programming and control of the Kinetis MCUs.

2.1 Tips and tricks for Kinetis MCUs

2.1.1 FPU

This is how to enable the hardware FPU module on Kinetis.

```
__enable_FPU:
                movw    r1, #60808
                movt    r1, #57344 // CPACR base
                ldr     r0, [r1]
                orr.w   r0, r0, #VFP_Enable
                str     r0, [r1]
                bx      lr


// where:
#define VFP_Enable       0x00F00000
```

3 MCU and module compatibility summary

This document is divided into five sections as follows. Table 2 provides a summary of the modules and classification of each change.

- Section 4. Unchanged modules
- Section 5. Removed modules
- Section 6. Updated modules
- Section 7. Enhanced modules
- Section 8. Modules with different number of instantiations.

There are no new modules for the MK24FN256 devices. All of the modules have a functional equivalent available in the MK24FN1M microcontroller.

Section 5, “Removed modules" are of note. Unpredictable results may occur if a module that is present on the MK24FN1M is written to on the MK24FN256.

Section 6, “Updated modules" outlines the modules that have been updated to use newer/different versions. The overall functionality provided will be similar; however, changes will be required in software and possibly hardware changes may be required to use the updated features. Since a minor pin-out change has been made it is covered in this section.

Section 7, “Enhanced modules" outlines the modules which have undergone minor changes. Software and hardware changes may be required to use the enhanced features.

Section 8, “Different instantiations" describes modules where the modules themselves have not changed, but there is a fewer number of these modules included in the MK24FN256 MCU.

3.1 Table of modules and classification

Table 2 lists of all of the modules on the MK24FN256. Each row is annotated with the disposition of the module listed. Notes are provided where needed.

Table 2. Modules list and classification

Module	Unchanged	Removed	Updated/ modified	Enhanced	Different instantiations	Notes
ARM Cortex-M4	X					
FPU	X					
MPU		X				Removed
NVIC	X					# of interrupts
AWIC	X					# of interrupts
Debug	X					
ETM, ETB		X				No ETM, ETB trace
SIM			X			Errata #e7534 SUBFAMID fixed
SMC	X					

Table 2. Modules list and classification (continued)

PMC	X					
LLWU	X					# of wakeup sources
MCM			X			AXBS priority
AXBS – Cross Bar		X	X			Removed AXBS registers, priority order swap
Peripheral Bridge	X					Reduced # of modules
DMAMUX	X					
eDMA			X			Errata fix #e6933
EWM	X					
WDOG	X					
FLASH Memory			X			Same type of memory reduced to 256K
Flash Memory Controller			X			FTFE -> FTFA
SRAM	X					
System Register File	X					
VBAT Register File	X					
EzPORT			X			No FLEXMem CMD No sector write
Flexbus		X				
MCG				X		Enable IRC48M clock
IRC48M				X		Fixed IRC48 clock enable
System Oscillator	X					
RTC OSC	X					
CMT		X				Not required
MMCAU		X				
RNGA		X				
CRC	X					
ADC0	X					Pinout changed
ADC1	X					
CMP0	X					CMP0 remains
CMP1	X					CMP1 remains
CMP2		X			X	Removed
6-bit DAC					X	Supports CMP0
12-bit DAC0	X					

Table 2. Modules list and classification (continued)

12-bit DAC1		X			X	Removed
VREF	X					
PDB			X			To support reduced analog peripherals
FTM0	X					
FTM1	X					
FTM2	X					
FTM3	X					
PIT	X					
LPTMR	X					
RTC	X					
USB 2.0 FS OTG	X					
USBDCD	X					
USB Regulator	X					
FLEXCAN		X				Removed
SPI0	X				X	
SPI1	X				X	
SPI2	X				X	
I^2C0	X					
I^2C1	X					
I^2C2	X					
UART0			X			LON functionality removed
UART1	X					
UART2	X				X	
UART3	X				X	
UART4	X				X	
SDHC		X				
I^2S0	X					
GPIO	X					
PADS			X			3V pads – no passive filter except NMI & reset
PAD controls			X			8 high drive strength

4 Unchanged modules

Since the majority of the modules are remaining the same, we will begin with those modules. Although these modules are unchanged themselves, other system changes affect them and may have some software or hardware impact. Only the modules that have such a possibility are mentioned here.

The registers and SIM clock gate control bits for these modules remain in the same bit positions and same address in the memory map. The user can expect 100% software and hardware compatibility within the MCU as noted while minor operational differences are noted in the discussion. Note that when internal interface signals connect to pins on the MCU there will be the expected PAD interface changes in drive strength and slew rate as described since the pads on the MK24FN256 are 3V pads.

4.1 Nested Vector Interrupt Controller (NVIC)

The NVIC control registers change simply because the modules that were removed on the MK24FN256 no longer have active bits. These registers are documented in the ARM documentation; NVIC_ISER, NVIC_ICER, NVIC_ISPR, NVIC_ICPR, NVIC_IABR, NVIC_IP, NVIC_STIR.

4.1.1 NVIC software impact

Attempts to enable or disable interrupts or clear pending interrupts for the removed modules will have no impact. However, it is a good idea to set the unused vectors to a default interrupt handler.

4.1.2 NVIC hardware impact

There is no hardware impact.

4.2 DMAMUX changes

The features, functionality, and memory map for the DMAMUX are unchanged. The DMA request sources have been modified to change some of the DMA request options to reserved corresponding to modules that have been removed.

4.2.1 Hardware considerations

The connections between the on-chip modules and the DMAMUX are internal connections within the processor. The changes to the interconnects between modules and the DMAMUX have no board impact.

4.2.2 Software considerations

The DMAMUX_CHCFGn[SOURCE] fields should not be configured to select any of the reserved/unused DMA request source numbers or undefined operation can result.

4.3 Power Management Controller (PMC)

There were no changes to the PMC.

4.3.1 PMC software impact

There is no software impact.

4.3.2 PMC hardware impact

There is no hardware impact.

4.4 Real-Time Clock (RTC)

The RTC has been updated but there is no software and hardware impact if the RTC clock functionality common to both MCUs is used. If this module is not used there are requirements on how to handle the RTC interface pins.

Table 3. Unused interfaces

PINS	Recommendation if unused
VBAT	Float
EXTAL32	VSS
XTAL32	Float
RTC_WAKEUP_B	Float

5 Removed modules

The modules that have been removed from the MK24FN1M for the MK24FN256 are provided in this section. Registers in the peripheral modules and clock gate control bits in the SIM are now reserved. Care must be taken not to write to reserved memory locations or a bus error may be generated. A bus error is a non-maskable interrupt that should be handled.

5.1 Trace ETM and ETB

On the MK24FN256 the Embedded Trace Macrocell (ETM) and the Coresight Embedded Trace Buffer (ETB) are removed.

5.2 Cross bar switch (AXBS) now AXBS-lite

The MK24FN1M includes a full cross bar controller to manage the extra masters that are present on this device, such as the SDHC and Flexbus. The MK24FN256 removes the cross bar registers resulting in a change to the method of setting the XBAR priority of the Masters on the cross bar. On the MK24FN256 the masters have a fixed priority and the MCM can be used to change the arbitration mode from fixed priority arbitration to round robin mode.

Additionally, the number of masters and slaves changed, resulting in changes to the Miscellaneous Control Module Registers reset settings.

5.2.1 AXBS software considerations

The number of slaves and masters on the crossbar and their numbers have changed on the MK24FN256.

5.2.1.1 The MCM register MCM_PLASC reset state

For MK24FN1M the reset state is 0x001F.

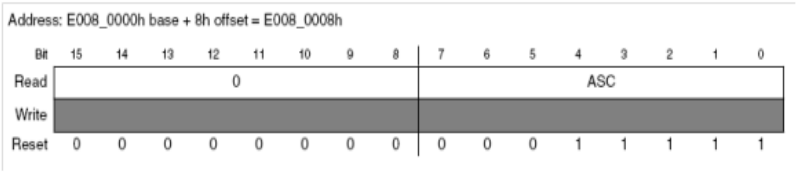

Figure 1. MK24FN1M MCM_PLASC reset state = 0x001F

The MCM register PLASC reset state on the MK24FN256 is equal to 0x000F.

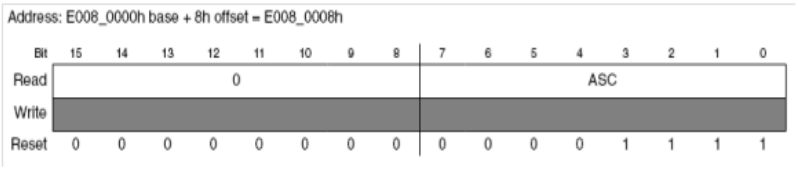

Figure 2. MK24FN256 MCM_PLASC reset state = 0x000F

5.2.1.2 The MCM_PLAMC reset state has changed

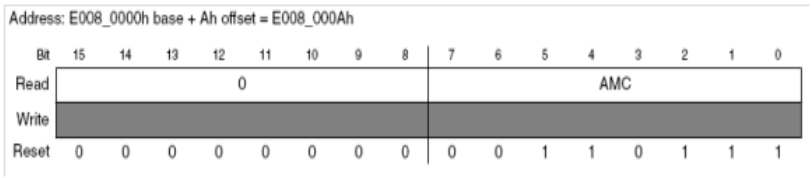

Figure 3. MK24FN1M MCM_PLAMC reset state = 0x0037

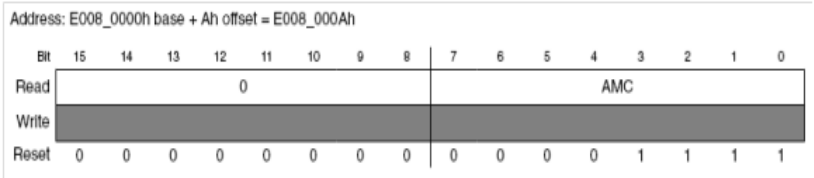

Figure 4. MK24FN256 MCM_PLAMC reset state = 0x000F

5.2.1.3 AXBS software fixed priority arbitration

The AXBS and AXBS-lite crossbar switches both come out of reset in fixed priority mode.

For the MK24FN1M, the AXBS crossbar switch comes out of reset with a fixed priority of from highest to lowest M0, M1, M2, M3, M4 (M5 -> M7 unused).

For the MK24FN256, the AXBS-lite crossbar switch comes out of reset with the same fixed priority of from highest to lowest M0, M1, M2, M3 (M4 -> M7 unused).

MCU performance is the same whether using the MK24FN1M or the MK24FN256 since the two MCUs have the same default fixed priority.

5.2.1.4 AXBS software round robin arbitration mode

For both the MK24FN1M and the MK24FN256, the AXBS and AXBS-lite crossbar switches can be programmed for round robin arbitration after any reset. However, the method to change to round robin arbitration mode is different for the MK24FN256 and the MK24FN1M. The differences are detailed in Table 4.

Table 4. Differences in method to change to round robin arbitration mode

For MK24FN1M AXBS - change to round robin arbitration mode by:
> - Setting AXBS_CRS0[ARB] = 01b;
> - Setting AXBS_CRS1[ARB] = 01b;
> - Setting AXBS_CRS2[ARB] = 01b;
> - Setting AXBS_CRS3[ARB] = 01b;
> - Setting AXBS_CRS4[ARB] = 01b;
For MK24FN256 AXBS-lite change to round robin arbitration mode by:
> - Setting the MCM_PLACR[ARB] to ‘1b

6 Updated modules

The modules that have been updated or changed on the MK24FN256 are provided in this section.

6.1 System integration module (SIM)

The SIM’s memory map does not change but there are a number of bit definitions within the registers that have changed.

6.1.1 SIM software considerations

In the SOPT2 register SDHCSRC, FBSL and PTD7PAD bit field are removed from MK24FN256. The bit field PLLFLLSEL is the same as the MK24FN1M, but there is a functional difference on the MK24FN256. If the IRC48 clock is selected [PLLFLLSEL = ‘11b], then the IRC48M is enabled regardless of the state of the IRC_EN in the USB_CLK_RECOVER_IRC_EN register bit field.

- In the SDID register the SUBFAMID bit fields are corrected. MK24FN1M read 0x2 which is an errata on this bit field #e7534. MK24FN256 correctly reads 0x4 for K24.
- In the SCGC1 register theUART4 and UART5 bit fields are removed.
- In the SCGC2 register the DAC0 and DAC1 bit fields are removed.

- In the SCGC3 register the SPI2 bit fields are removed.
- In the SCGC4 register the VREF, UART3, UART2 and CMT bit fields are removed.
- In the SCGC6 register the DAC0, RNGA and FLEXCAN0 bit fields are removed.
- In the SCGC7 register the MPU and FLEXBUS bit fields are removed.
- In the CLKDIV1 register the OUTDIV3 bit fields is removed.
- In the FCFG1 register the NVMSIZE, EESIZE and DEPART bit fields are removed and the PFSIZE bit field will be always set to ‘1001b.
- In the FCFG2 register the PFLASH and MAXADDR1 bit fields are removed.

The reserved bit fields in the SIM clock gating registers should always be written to a 0.

6.1.2 Hardware considerations

There is no hardware impact.

6.2 Multipurpose clock generator (MCG)

The MCG changes are limited to the IRC48 clock control. The bit field functionality OSCSEL in the register MCG_C7 is the same.

6.2.1 MCG software considerations

If the MCG Control register selects IRC48 MHz clock MCG_C7[OSCSEL]=10, then the IRC48M is enabled regardless of the state of the IRC_EN in the USB_CLK_RECOVER_IRC_EN register bit field.

6.3 EzPORT

The EzPORT module is changed due the change in Flash memory.

6.3.1 EzPORT software considerations

Since there is no FLEXMEM on the MK24FN256 the commands to write to the FLEXMEM memory block have been removed.

The command to program the flash has changed internally but the user will see no effect from this change. The difference is due to the size of the flash memory buffer that is programmed with each program command. On MK24FN1M the flash program command would write 64 bits (8 bytes)while the MK24FN256 flash command will program 32 bits (4 bytes).

6.3.2 EzPORT hardware considerations

When programming the Flash through EzPORT the programming size change is not evident. However, the programming time will be greater since the MCU is doing a program every 4 bytes instead of programming 8 bytes or more at a time.

6.4 eDMA changes

The eDMA module is the same basic module with the same features on both parts; however there is an erratum on the eDMA module that has been fixed on the MK24FN256 device.

The fixed erratum is e6933 in the KINETIS_1N83J errata document:

e6933: eDMA: Possible misbehavior of a preempted channel when using continuous link mode

Errata type: Errata

Description: When using continuous link mode (DMA_CR[CLM] = 1) with a high priority channel linking to itself, if the high priority channel preempts a lower priority channel on the cycle before its last read/write sequence, the counters for the preempted channel (the lower priority channel) are corrupted. When the preempted channel is restored, it runs past its "done" point instead of performing a single read/write sequence and retiring.

The preempting channel (the higher priority channel) will execute as expected.

Workaround: Disable continuous link mode (DMA_CR[CLM]=0) if a high priority channel is using minor loop channel linking to itself and preemption is enabled. The second activation of the preempting channel will experience the normal startup latency (one read/write sequence + startup) instead of the shortened latency (startup only) provided by continuous link mode.

6.4.1 Hardware considerations

The DMA changes only involve software. No board changes will be required because of these updates.

6.4.2 Software considerations

The erratum only affects the eDMA when it is used in a very specific situation. In order to encounter the original problem a DMA channel that is configured for continuous link mode to itself must preempt a lower priority channel. If there are no cases where a eDMA channel links to itself, then there is no effect on eDMA software or behavior.

If there is an eDMA channel that links to itself, then the recommended workaround (disabling continuous link mode) can be used on both MK24FN1M and MK24FN256. The software workaround is backwards compatible, so software can be left as-is on both devices. There is a potential for a small performance improvement if the continuous link mode feature is enabled on the MK24FN256 that implements the fix.

6.5 Analog interconnect changes

In order to allow for many of the analog modules to work together, the processor makes internal connections between some of the analog modules. There is also a PDB timer that works in conjunction with some of the analog peripherals. Form and function are the same in general, but because of the removal of the 12-bit DAC modules and two of the comparator instantiations (only CMP0 is available on the MK24FN256) some of the interconnections and PDB features have also been removed.

6.5.1 CMP input connection changes

The CMP0 input options are the same except that the 12-bit DAC1_OUT option on IN4 is removed because the 12-bit DAC module is no longer present. CMP0_IN4 is now the only option for CMP0 IN4.

6.5.2 CMP hardware considerations

The connections that have been removed are internal to the processor. No PCB changes should be needed to accommodate the CMP input connection changes. The board designer should be aware that DAC1_OUT is no longer an option on the pin associated with CMP0_IN4.

6.5.3 CMP software considerations

No software changes should be associated with this change.

6.6 PDB connection changes

The PDB feature set has been reduced to remove features associated with the DAC and two comparator instantiations that are no longer present on MK24FN256.

6.6.1 PDB output trigger changes

The PDB includes a pulse out trigger that can be used to trigger the comparator modules. Because the number of CMP module instantiations was reduced from three to two, the number of PDB pulse out triggers is also reduced from three to two.

6.6.2 PDB input trigger changes

The comparators can also be used as input triggers for the PDB, so the removal of CMP3 also affects the PDB input triggers. The CMP3 input trigger options in the PDB trigger 0b0010 and 0b0011 slots have been removed.

6.6.3 PDB DAC internal triggers

One of the features of the PDB on MK24FN1M is the ability to generate interval triggers to the DACs. One of the 12-bit DAC modules have been removed on MK24FN256, so the PDB on the MK24FN256 only one interval trigger can be used to generate DAC internal triggers.

6.6.4 PDB hardware considerations

The PDB features that have changed are all related to connections within the chip, so these changes do not impact hardware.

6.6.5 PDB software considerations

There are a number of registers associated with the PDB features that are unneeded on the MK24FN256 have been removed:

- The PDB_DACINTC1, PDB_DACINT1 registers that were used to control the DAC1 interval triggers have been removed.
- The PDB0_PO2DLY register that were used to control the pulse out signals for CMP3 has been removed.

In addition to the registers that have been removed, the PDB_SC[TRGSEL] field should not be set to 0b0011. These values were used to select the CMP3 as PDB input trigger. This value is reserved on the MK24FN256. Configuring the PDB for this input trigger values will result in undefined operation.

6.7 Pad changes

The pads on the MK24FN1M were 5 V tolerant pads. The MK24FN256 has a 3V pads that have removed the passive filter from all pads except NMI and RESET. The drive strength of each pad has changed on the MK24FN256.

6.7.1 Pads hardware considerations

On MK24FN1M, all digital/GPIO pads have configurable drive strength (normal/high is 2mA/9mA). On MK24FN256 all digital/GPIO pads have normal drive strength ~5mA, and only 8 pads have high drive strength option ~20mA. Please see the reference manual and data sheet for details.

6.8 Pinout changes

There was a small analog connection changes from MK24FN1MF that may affect the customer's use case. All removed module connections to pins have been eliminated in the pinout.

6.8.1 Pinout hardware considerations

- The ADC0_DP2 is moving from ball map coordinate E2 to J1, ADC0_DM2 is moving from ball map coordinate F4 to J2.
- On MK24FN1M ADC0_DP2 shared functions with digital function PTE2, LLWU_P1, SPI_SCK and UART1_CTS. While on MK24FN256 ADC0_DP2 shares a dedicated analog input pad with ADC1_DP1.
- On MK24FN1M ADC0_DM2 shared functions with digital function PTE3, SPI1_SIN, SPI1_SOUT and UART1_RTS.
- While on MK24FN256 ADC0_DM2 shares a dedicated analog input pad with ADC1_DM1.

121 XFBGA	Pin Name	Default	ALT0
J1	ADC1_DP1/ ADC0_DP2	ADC1_DP1/ ADC0_DP2	ADC1_DP1/ ADC0_DP2
J2	ADC1_DM1/ ADC0_DM2	ADC1_DM1/ ADC0_DM2	ADC1_DM1/ ADC0_DM2

Figure 5. MK24FN256 signal multiplexing from RM chapter 10

6.8.2 Pinout software considerations

On MK24FN256 the pin mux control registers for PTE2 and PTE3 no longer affect the ADC0_DP2 and ADC0_DM2 analog input channels. Now the default and MUX = ALT0 pin function is ADC1_SE6a for PTE2_PCR[MUX = 0] and the default and MUX = ALT0 pin function is ADC1_SE5a for PTE3_PCR[MUX = 0].

6.9 Flash controller (FTFE to FTFA)

The flash controller is required to be updated. The MK24FN1M FTFE flash supports 1 Mbytes of flash with FlexRAM, while the MK24FN256 FTFA supports 256 Kbytes of program flash with no FlexRAM.

6.9.1 Memory map considerations

The FTFE and FTFA use different module acronyms, but other than that the memory map remains the same. For example, on a MK24FN1M device there is an FTFE_FSTAT register. The FTFE_FSTAT register becomes the FTFA_FSTAT register on a MK24FN256 device. All of the register addresses and fields are identical only the preceding acronym changes

The size of the Flash array and the IFR block has been reduced. The IFR block changes from 1024 bytes on MK24FN1M down to 256 bytes of write once flash space.

Address Range	Size (Bytes)	Field Description
0x000 – 0x3BF	960	Reserved
0x3C0 – 0x3FF	64	Program Once Field

Figure 6. MK24FN1M IFR memory

Address Range	Size (Bytes)	Field Description
0x00 – 0xBF	192	Reserved
0xC0 – 0xFF	64	Program Once Field

Figure 7. MK24FN256 IFR memory

6.9.2 Software impact

The biggest change when moving from the FTFE to the FTFA module is that the programming size for the flash changes. The FTFE module uses a program phrase command (FCMD = 0x7) to program 64 bits at a time, while FTFA module uses a program longword command (FCMD = 0x6) to program 32 bits at a time. Any development tool or application code that modifies the flash contents will need to be changed to use the new program size.

Another change is that the default size of the flash protection regions reduces. For example, there are 32 P-flash protection regions on both the FTFE and FTFA where the total P-flash size is evenly distributed across the 32 regions. For an FTFA device with four times less P-flash, the size of the P-flash protection regions will be four times smaller.

The program flash section command (FCMD = 0x0B) has been removed along with the Program Swap command (FCMD = 0x46).

The ability to erase and program a block while executing (RWW) from flash has been eliminated since there is only one Flash block.

6.9.3 Hardware impact

There is no hardware impact.

6.10 Private Peripheral Bus (PPB)

The two embedded trace functions that were present on the MK24FN1M are no longer present on the MK24FN256.

0xE004_0000–0xE004_0FFF	Trace Port Interface Unit (TPIU)
0xE004_1000–0xE004_1FFF	Embedded Trace Macrocell (ETM)
0xE004_2000–0xE004_2FFF	Embedded Trace Buffer (ETB)
0xE004_3000–0xE004_3FFF	Embedded Trace Funnel
0xE004_4000–0xE007_FFFF	Reserved

Figure 8. PPB on MK24FN1M with Trace

ITM	System clock	—	—
ETM	System clock	TRACE clock	TRACE_CLKOUT
ETB	System clock	—	—
cJTAG, JTAGC	—	—	JTAG_CLK

Figure 9. Module clocks on MK24FN1M

0xE004_0000–0xE004_0FFF	Trace Port Interface Unit (TPIU)
0xE004_1000–0xE004_1FFF	Reserved
0xE004_2000–0xE004_2FFF	Reserved
0xE004_3000–0xE004_3FFF	Reserved
0xE004_4000–0xE007_FFFF	Reserved

Figure 10. PPB on MK24FN256 without Trace

Module	Bus interface clock	Internal clocks	I/O interface clocks
ITM	System clock	—	—
cJTAG, JTAGC	—	—	JTAG_CLK

Figure 11. Module clocks on MK24FN256

6.10.1 Software considerations

During JTAG debug the embedded trace cannot be used. The registers in the MCM that controlled the ETB have been removed.

E008_0014	ETB Counter Control register (MCM_ETBCC)	32	R/W	0000_0000h	17.2.5/375
E008_0018	ETB Reload register (MCM_ETBRL)	32	R/W	0000_0000h	17.2.6/376
E008_001C	ETB Counter Value register (MCM_ETBCNT)	32	R	0000_0000h	17.2.7/376
E008_0030	Process ID register (MCM_PID)	32	R/W	0000_0000h	17.2.8/377

Figure 12. MCM registers removed from the memory map of MK24FN256

6.11 Low Leakage Wakeup Unit (LLWU)

The LLWU has been updated on the MK24FN256 to remove the wakeup sources for modules that were removed.

6.11.1 Software considerations

These sources are now reserved and will not cause a wakeup from low power modes. The LLWU wakeup sources table on MK24FN256 now shows sources, CMP1 and CMP2, that were present on the MK24FN1M are no longer listed as sources since they were removed.

Input	Wakeup source		Input	Wakeup source
LLWU_P0	PTE1/LLWU_P0 pin		LLWU_P12	PTD0/LLWU_P12 pin
LLWU_P1	PTE2/LLWU_P1 pin		LLWU_P13	PTD2/LLWU_P13 pin
LLWU_P2	PTE4/LLWU_P2 pin		LLWU_P14	PTD4/LLWU_P14 pin
LLWU_P3	PTA4/LLWU_P3 pin [1]		LLWU_P15	PTD6/LLWU_P15 pin
LLWU_P4	PTA13/LLWU_P4 pin		LLWU_M0IF	LPTMR[2]
LLWU_P5	PTB0/LLWU_P5 pin		LLWU_M1IF	CMP0[2]
LLWU_P6	PTC1/LLWU_P6 pin		LLWU_M2IF	CMP1[2]
LLWU_P7	PTC3/LLWU_P7 pin		LLWU_M3IF	Reserved
LLWU_P8	PTC4/LLWU_P8 pin		LLWU_M4IF	Reserved
LLWU_P9	PTC5/LLWU_P9 pin		LLWU_M5IF	RTC Alarm[2]
LLWU_P10	PTC6/LLWU_P10 pin		LLWU_M6IF	Reserved
LLWU_P11	PTC11/LLWU_P11 pin		LLWU_M7IF	RTC Seconds[2]

Figure 13. MK24FN256 LLWU wakeup sources

6.11.2 Hardware considerations

The input pins that are associated with the removed modules no longer operate as LLWU wakeup inputs. Specifically, the CMP2 input is removed.

6.12 Universal Asynchronous Receiver Transmitter (UART0)

The LON functionality of UART0 has been removed. The vector for LON functionality has been removed on MK24FN256.

0x0000_00B8	46	30	0	7	UART0	Single interrupt vector for UART LON sources

Figure 14. MK24FN1M vector 46

0x0000_00B8	46	30	0	7	—	—

Figure 15. MK24FN256 vector 46

7 Enhanced modules

The MCG, SIM, and USB modules have been enhanced on the MK24FN256. All enhancements involve the functionality of the IRC48M clock.

7.1 48 MHz Internal Reference Clock (IRC48M)

The enhancements to the MK24FN256 fix a problem in the implementation of the IRC48 enable on MK24FN1M. There is an issue on the MK24FN1M that is fixed on MK24FN256. IRC48MCLK is enabled via the USB control registers. IRC48MCLK will be reset by USB reset event and can lead to a system lockup when the IRC48MCLK is also used to clock the system. To avoid this scenario, usage of USB reset should be avoided.

7.1.1 IRC48M software considerations

On MK24FN1M the IRC48 clock could only be enabled one way. To enable the clock on MK24FN1M the IRC_EN bit in the USB0_CLK_RECOVER register is set to ‘1b.

On MK24FN1M, if the USB0_USBTRC0 register bit USBRESET is set then the IRC48 clock is turned off since the IRC_EN bit is cleared. If the clock was sourcing the CPU then the CPU would halt with no clock when the IRC_EN bit was cleared. If the IRC48 is only being used as the clock source to the USB device then all that is required to do is re-enable the IRC with the IRC_EN bit.

The following code is typical of this case:

```
/* USB Module Configuration */

/* Reset USB Module */

SIM_SCGC4 |= (SIM_SCGC4_USBOTG_MASK);          /* USB Clock Gating */

       USB0_USBTRC0 |= USB_USBTRC0_USBRESET_MASK;

       while (FLAG_CHK(USB_USBTRC0_USBRESET_SHIFT, USB0_USBTRC0)) {

       };
```

In the MK24FN256 the IRC48M clock is enabled in three ways. Please see the RM for more details of this functionality. The following wording is taken directly from the RM clocking section.

IRC48MCLK is enabled via the following control settings while operating in these clocking modes:

USB Control register enables—enabled when USB_CLK_RECOVER_IRC_EN[IRC_EN]=1 or MCG Control register selects IRC48 MHz clock—enabled when MCG_C7[OSCSEL]=10 or SIM Control register selects IRC48 MHz clock—enabled when SIM_SOPT2[PLLFLLSEL]=11.

7.1.2 IRC48M hardware considerations

There is no hardware impact.

8 Different instantiations

The modules with different instantiations include the 12-bit DAC, CMP and associated 6-bit DAC.

The MK24FN1M had more of each of these modules. The MK24FN256 has at least one of these modules.

The number of CMP modules is reduced to two.

The number of 6-bit DACs is reduced to two.

9 Revision history

Rev. number	Date	Substantive change(s)
0	8/2014	Initial release (NDA)
1	10/2014	Initial public release. Hardware and/or software changes required for migration are updated.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: AN5013
Rev. 1
10/2014