Freescale Semiconductor
Errata

MPC185CE
Rev. 9, 10/2004

XPC185 Security Processor Device Errata

This document details all known silicon errata for the XPC185 and its derivatives. Table 1 provides a revision history for this document.

Table 1. Document Revision History

Revision	Substantive Changes
0	Initial release
1	Added Errata 2-4
2	Added Errata 5
3	Added work-around to Errata 3, updated Errata 1
4	Added Errata 6
5	Updated Errata 2,4
6	Added Errata 7
7	Added Errata 8
8	Updated Errata 3, 6
9	Added Clarification 9

© Freescale Semiconductor, Inc., 2004. All rights reserved.

Errata No. 1: PKEU ROM Failure

Detailed Description and Projected Impact:

The XPC185 PKEU performs complex Montgomery Multiplication and Exponentiation through a combination of hardware and ROM microcode. First silicon on the XPC185 has a manufacturing defect in the ROM which prevents normal operation.

Work-around:

None. Both PKEUs on the XPC185 are effected by this defect.

Projected Solution:

This errata was caused by a one time manufacturing error which effected the first wafer lot (XPC185VF) only. Rev.A (XPC185VFA) silicon and all subsequent revisions have fully functional PKEUs.

Errata No. 2: Global Software Reset improperly resetting 60x Bus Interface Unit

Detailed Description and Projected Impact:

The 60x Interface module should not be reset via the Master Control Register’s SW Reset bit. This will cause the Controller to lose track of the 60x Bus Status, and could result in corrupted data being driven on the bus during the a Global SW Reset.

Work-around:

Reset individual device modules individually, and use Asynchronous Hardware Reset for a truly global reset. Any use of Global SW Reset should be removed for the user’s code to avoid this errata.

Projected Solution:

This errata has been fixed in Rev.B (XPC185VFB) silicon and all subsequent revisions.

Errata No. 3: XPC185 PLL losing lock

Detailed Description and Projected Impact:

The XPC185 has an internal noise issue which causes the internal PLL to lose synchronization, and immediately lock up. This generally leads to the host CPU generating a Machine Check Error, and the board on which the 185 is operating to lock up as well. Under certain I/O switching conditions, the XPC185 PLL loses lock, causing the device to hang. This is indicative of a noise issue.

Work-around:

The 185 design team continues to work on a design fix that will restore full operating range, but in the interim, the following operating conditions have been found to improve operation to the point that software development, and some board debug, can be accomplished.

The 185 is designed to operate on 60x buses running at both 2.5v and 3.3v. The suggested work-arounds are different for each operating voltage.

For 2.5v systems (MPC185 with MPC74xx)

Run I/O power as low as possible (2.25-2.35v)

Maintain 1.5v core power

The internal PLL should work at these conditions, but if it locks up, the user can use the PLL By-Pass (Pin D11 pulled low) to disable the internal PLL and run at the 60x bus <66MHz.

If the 185 does lock up, the CPU will likely generate a Machine Check Error.

To clear the error, a full system reset (hardware reset) will be required to restart.

For 3.3v systems (MPC185 with PowerQUICC 2, or MPC74xx)

Use PLL By-Pass mode (Pin D11 pulled low) with external clock at 33MHz.

Maintain 1.5v core, 3.3v I/O

If the board still locks up, try lowering I/O voltage to 3.0v (3.3v - 10%). Lowering I/O voltage in 3.3v systems doesn’t have as much benefit as doing the same in 2.5v systems, however this is worth trying if not a significant effort.

If the 185 does lock up, the CPU will likely generate a Machine Check Error.

To clear the error, a full system reset (hardware reset) will be required to restart.

Update:

XPC185VFB and VFC silicon show significantly improved stability in normal test conditions, resulting in the elimination of 185 lock-ups. Recent testing has determined that although the 185’s PLL is more stable under normal conditions, it still has significant jitter under test cases with worst case switching noise. This jitter can be reduced by running the 185’s PLL at 1.8v (the PLL has its own Vdd, APLL), however the PLL voltage must be tightly controlled to avoid reliability issues.

With the higher PLL voltage, PLL jitter is reduced, but not eliminated, and because this jitter can have a positive or negative effect on timing on each clock edge, the worst case jitter can cause timing failures in all but the most carefully designed boards. Higher I/O voltages (3.3v) as used in PowerQUICC II systems cause more jitter than 2.5v I/O.

As a consequence of the 185's PLL jitter, Freescale is issuing the following revised 185 AC timings. These timings show a new maximum operating frequency of 66MHz in PowerQUICC II systems.

XPC185 users are requested to contact their Freescale sales/FAE organization to schedule calls with the 185 Product Team to discuss the impact of this errata on their programs.

AC Timing Characteristics:

Table 1 shows the AC timing specifications for use with a PowerQUICC II. All timings assume a 40-pF load.

Table 2. AC Electrical Characteristics - PowerQUICC II

Condition	Name	Min	Max	Unit
Power supply voltage—Core	V_{DD}	1.45	1.65	V
Power supply voltage—I/O	V_{DDQ}	2.3	3.2	V
Power supply voltage—PLL	V_{PLL}	1.75	1.85	V
Clock frequency	F_{clock}	—	66	MHz
Clock cycle time	t_{KHKH}	12	—	nS
Clock-to-signal valid delay	t_{KHQV}	—	7.4	nS
Clock-to-signal hold	t_{KHQX}	-0.6	—	
Input setup time to clock-bused signals	t_{DVKH}	4.3	—	nS
Input hold time clock	t_{KHDX}	2.2	—	nS

Table 2 shows the AC timing specifications for use with an MPC107 (Tsi107) or other 60x bridge/memory controller. All timings assume a 15-pF load.

Table 3. AC Electrical Characteristics - 60x Bridge/Memory Controller

Condition	Name	Min	Max	Unit
Power supply voltage—Core	V_{DD}	1.45	1.65	V
Power supply voltage—I/O	V_{DDQ}	2.3	2.6	V
Power supply voltage—PLL	V_{PLL}	1.75	1.85	V
Clock frequency	F_{clock}	—	100	MHz
Clock cycle time	t_{KHKH}	10	—	nS
Clock-to-signal valid delay	t_{KHQV}	—	5.1	nS
Clock-to-signal hold	t_{KHQX}	-0.05	—	
Input setup time to clock-bused signals	t_{DVKH}	2.75	—	nS
Input hold time clock	t_{KHDX}	0.85	—	nS

Projected Solution:

This errata will be fixed in Rev.D (XPC185VFD) silicon, scheduled for Q105 sampling.

Errata No. 4: Planned Reversal of Interrupt Default Settings

Detailed Description and Projected Impact:

The XPC185's interrupt registers reset to an unmasked state. While not an errata, having interrupts masked following reset simplifies the user's task during system debug and initial bring-up.

Work-around:

All interrupt registers can be manually masked after reset, but prior to intensive debug. As the system becomes more stable, interrupt sources can be selectively unmasked, and in normal operation, most if not all interrupts will be unmasked.

Projected Solution:

A new bit has been added to the MPC185VFB Interrupt Mask Register. This change has been made in Rev.B (XPC185VFB) silicon and all subsequent revisions. MPC185 User's Manual has been updated to reflect this change. The revised Interrupt Mask Register and bit definitions are shown in Figure 1.

	0	1	2	3	4	5	11	12	13	14	15
Field	CHA_2		CHA_1		A-Err	Reserved		CHA_4		CHA_3	
Definition	Err	Dn	Err	Dn				Err	Dn	Err	Dn
Reset	0x0000										
R/W	R/W										
Addr	0x 01008										

	16	31
Field	Reserved	
Definition		
Reset	0x0000	
R/W	R/W	
Addr	0x 01008	

	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Field	PKEU_2		PKEU_1		Reserved		RNG		Reserved		AFEU		MDEU_2		MDEU_1	
Definition	Err	Dn	Err	Dn			Err	Dn			Err	Dn	Err	Dn	Err	Dn
Reset	0x0000															
R/W	R/W															
Addr	0x 01008															

	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Field	AESU_2		AESU_1		DEU_2		DEU_1		GIE	TEA	DPE	APE	Reserved		KEU	
Definition	Err	Dn	Err	Dn	Err	Dn	Err	Dn							Err	Dn
Reset	0x0000															
R/W	R/W															
Addr	0x 01008															

Figure 1. Interrupt Mask Register

Table 4. Interrupt Mask, Status, and Clear Register Signals

Bits	Name	Reset Value	Description
Multiple	CH_Err_Dn	0	Each of the 4 channels has Error & Done bits. 0 No error detected. 1 Error detected. Indicates that execution unit status register must be read to determine exact cause of the error. 0 Not DONE. 1 DONE bit indicates that the interrupting channel or EU has completed its operation.
Multiple	EU_Err_Dn	0	Each of the execution units has Error & Done bits. 0 No error detected. 1 Error detected. Indicates that execution unit status register must be read to determine exact cause of the error. 0 Not DONE. 1 DONE bit indicates that the interrupting channel or EU has completed its operation.
5:11, 16:31, 36:37, 40:41, 60:61	Reserved	0	Reserved, set to zero.
4	A-Err	0	EU Assignment Error bit. This bit indicates that a static assignment of a EU was attempted on a EU which is currently in use. 0 No error detected. 1 EU Assignment Error detected.
56	GIE	0	Reserved in XPC185VF, XPC185VFA MPC185VFB and subsequent revisions only Global Interrupt Enable. Individual interrupt sources reflected by the Interrupt Status register are “enabled” at reset. This bit, which resets to “disabled”, allows the user to selectively mask individual interrupt sources in the Interrupt Mask Register before enabling the remaining unmasked interrupt sources. 0 MPC185VFB interrupts globally disabled 1 MPC185VFB interrupts enabled
57	TEA	0	Transfer Error Acknowledge. Set when the MPC185 as a master receives a Transfer Error Acknowledge. 0 No error detected. 1 TEA detected on 60x bus.
58	DPE	0	Data Parity Error. Set when the MPC185 detects a slave data parity error. 0 No error detected. 1 DPE detected on 60x bus.
59	APE	0	Address Parity Error (bit 59). Set when the MPC185 detects a slave address parity error. 0 No error detected. 1 APE detected on 60x bus.

Errata No. 5: Improper Key Size, Data Size Error Interrupt in Debug Mode

Detailed Description and Projected Impact:

The XPC185VF is designed to operate as a bus master, using descriptors to provide the crypto-channels and execution units with information such as lengths and pointers to items such as keys, context, and data. It is also possible to operate the XPC185 without descriptors, by directly writing key lengths to the key size registers, data lengths to the data size registers, etc. Operating this way is typically only done during debug. Directly writing any execution unit size register (example, DEU Key Size Register or DEU Data Size Register) causes that register to generate an illegal size error.

Work-around:

When operating in debug mode, key size and data size errors must be disabled in the appropriate execution unit's Interrupt Control Register (example, disable key size error, data size error in DEU Interrupt Control Error.)

Projected Solution:

Debug mode is rarely used, and the improper signalling of the illegal size error does not effect normal, descriptor based operation. There is no schedule to implement a fix for this errata. A warning will be added to the MPC185 User's Manual in each execution unit's Interrupt Status Register.

Errata No. 6: Unnecessary Transaction Repeat

Detailed Description and Projected Impact:

The XPC185VFA is designed to operate as a 60x bus master, using 60x bus signals to monitor the transaction status of other masters on the bus, taking bus ownership and relinquishing bus ownership as necessary. Under certain conditions, the XPC185VFA can lose track of its own last successful transaction, and repeat its last transaction the next time it is granted the bus. When this occurs, redundant data can be placed in EU key registers, or EU input FIFOs, cause the output of the 185 to be corrupted. The conditions which lead to this failure occur in systems in which the XPC185VFA is one of three or more masters on the 60x bus. Transactions initiated by a master (other than the 185), and cancelled by a different master through ARTRY cause the 185 to believe it needs to retry its last transaction the next time it takes ownership of the 60x bus.

Generally in a system, the CPU uses ARTRY to advise other masters that the transaction being requested must not continue, due to the CPU having a modified copy of the data in its caches. ARTRY causes the original master to cancel the transaction so the CPU can push the modified data to main memory. ARTRY is also used to cancel bus transactions which cannot be completed in a timely manner, such as CPU reads of memory locations in PCI address space. If a PCI bridge cannot source the data requested by the CPU immediately, it may advise the CPU to “try again later” through ARTRY, in order to free the bus for other transactions. Even in systems with only two obvious masters, such as a PowerQUICC 2 and XPC185VFA, if the PowerQUICC 2 supports PCI (rather than local bus), the internal PCI bridge acts as a 60x bus master and will also use ARTRY to signal “try again later” when it can’t source or sink PCI data quickly enough. Thus, the XPC185VFA is likely to be effected by the Unnecessary Transaction Repeat errata in all but the most basic systems.

Work-around:

The first step is to evaluate if this errata effects your system. Determine if the conditions under which other 60x devices assert ARTRY match the conditions described above.

If your system’s PCI bridge can be configured to hold the 60x bus until PCI data is sourced or sunk, rather than terminating with ARTRY, configure it to hold the bus.

If 3rd party ARTRYs cannot be prevented, the remaining option is to operate the XPC185VFA as a slave. Recall that the XPC185VFA has 32KB of on-chip gpRAM. Rather than creating descriptors which cause the 185 to fetch keys and data from external memory, the data to be encrypted or decrypted can be placed in gpRAM, along with the necessary keys and context. The descriptors which utilize the keys, context, and data can also be placed in gpRAM, or written directly to the crypto-channel’s descriptor buffer. When the XPC185VFA has completed the requested operation, it will signal DONE to the host via interrupt, allowing the host to retrieve the permuted data from the 185’s gpRAM. This method allows the 185 to act as a master internal to its own memory space, which enables flow control, and single pass, single descriptor operations, but places a greater burden on system resources to move data in and out of gpRAM.

Projected Solution:

This errata has been fixed in Rev.C (XPC185VFC) silicon and all subsequent revisions.

Errata No. 7: Potential MDEU Snooping Error

Detailed Description and Projected Impact:

The MPC185VF performs encryption and hashing operations in a single pass of the data (as required for operations such as IPSec) through a mechanism called ‘snooping”. A complete description of snooping can be found in Section 5.2.1 of the MPC185 User’s Manual. When performing ‘out-snooping’ as required for IPSec out-bound processing, it is possible for the MPC185’s MDEU to fail to snoop a word from the symmetric encryption unit and consequently generate an incorrect HMAC. It is assumed that the system will append the HMAC calculated by the MPC185 and transmit it. The IPSec tunnel termination point will detect the bad HMAC and drop the packet.

This errata was discovered during verification of a device using a security block derived from the MPC185, and an examination of MPC185 logic indicates the possibility of a similar failure exists. Simulation indicates the errata is more likely to occur under the following conditions:

Single DES with HMAC-SHA-1, very large data sizes (>4KB), and fast memory access. The combination of the fastest encryption algorithm with the slowest HMAC algorithm with fast memory access creates a situation in which the writes from the DEU Output FIFO can occur faster than the flow control signals from the MDEU snooping can regulate.

In the system in which this errata was discovered, the security block was writing to fast SRAM, and failures were only discovered when data size exceeded 4KB. Failures were still very intermittent (~.1% of test cases), with the most failures occurring near the maximum data size for the simulation, which was 8KB.

Work-around:

This failure has never been observed in the MPC185. A large number of concentrated test cases were run for SDES-HMAC-SHA-1 with out-snooping with large data sizes (4-8KB), and no failures were observed. If the potential for occasional transmission of packets with bad HMACs is unacceptable, the following work-around could be used:

Rather than using a single descriptor ‘0x2053_1C20 (SDES-CBC-Encrypt-HMAC-SHA-1)’ for outbound IPSec, processing could be split into two descriptors. The first descriptor 0x2050_0000 would perform SDES-CBC-Encrypt, and the second 0x31C0_0000 would perform HMAC-SHA-1. This will result in lower performance, however at the large data sizes most impacted by this errata, the overhead of a second descriptor should be insignificant.

If it is known that higher layer software in the IPSec tunnel termination point will recover from a failed HMAC, and the remote possibility of an HMAC failure can be tolerated, the user can continue to use the higher performance, single descriptor method. While Freescale cannot guarantee HMAC failure will never occur at typical packet sizes (<2KB), all indications are that typical systems (no fast SRAM on the 60x bus) will not see this errata, even with SDES-HMAC-SHA-1.

Projected Solution:

A hardware solution is under investigation.

Errata No. 8: PKEU Data Size Limitation:

Detailed Description and Projected Impact:

The MPC185 PKEU ROM contains a large number of routines which can be used to perform modular arithmetic and elliptic curve based Public Key operations. A few less frequently used PKEU routines are sensitive to data size errors, and will result in the PKEU hanging without signalling an error. See the table below for complete list of PKEU errata, work - arounds, and resolutions.

Errata	Work Around	Projected Solution
Inversion routines F2M_INV and MOD_INV hang for modulus sizes <60 bytes	No workaround: If this function is used, it must be performed in software. Mainly affects RSA-CRT but not DSA (160 bits) or EC (< 480 bits)	Warning to be added to documentation.
PKEU hangs if the real modulus size is less than the value written to the data size register when using MOD_R2MODN and F2M_R2 routines. (ex. If modulus is 128 bytes, a data size of 129 bytes will cause the PKEU to hang rather than returning a data size error.)	Software driver must make sure not to write a wrong data size.	Warning to be added to documentation.
ECC routines (projective) return (Rinverse, Rinverse, 0) instead of currently defined (1,1,0) for point at infinity	Redefine point at infinity as Z=0 only, this must be ok with all software (applications).	Warning to be added to documentation.

Errata No. 9: Clarification: PKEU Input Formatting:

Detailed Description and Projected Impact:

The MPC185 PKEUs are designed to operate on big numbers, represented in strings up to 2048 bits long. Each PKEU's internal architecture is natively 64-bit little endian, which means that it expects data least significant word first. This leads to the non-intuitive requirement for input data (exponents, modulus, etc.) to be represented in memory as a ìbigî integer with the least significant bit aligned to the right.

Integer representation- 0x00000012 abcdef12 3456789a bcdef0f1

String representation- 0x12abcdef 12345678 9abcdef0 f1000000

In applications in which the MPC185 is connected to a big endian processor via the 64 bit 60x interface, data should be represented in memory as shown below so that the least significant 64-bit Dword is fetched first and the bytes within the Dwords are in big endian format.

Address	Data
0x0000	f1f0debc 9a785634
0x0008	12efcdab 12000000

Work Around:

The MPC185 device driver contains example code for performing the Dword and byte swapping necessary to transform data for use by the MPC185 PKEUs. The example code (shown below) can also be found in the device driver file pkhatest.c.

```
void CopyLongWordReverse (unsigned char *src, unsigned char *dst, unsigned int len)
{
 int i, j;
 unsigned char *source, *srcsave=NULL;
 unsigned char *dstend = dst + len;   /* len is in bytes */

 if (src == dst) {       /* move into same area */
   source = malloc (len);
   srcsave = source;
   bcopy (src, source, len);
   source += (len-8);
 } else {
   source = src + (len-8);
 }

 while (dst < dstend) {
   j=7;
   for (i=0; i<8; i++) {
     dst[i]=source[j];
     j--;
   }
   dst +=8;
   source -= 8;
 }
 if (srcsave != NULL)
   free (srcsave);
}
```

Projected Solution:

The required data transformation for using the MPC185 PKEUs represents minimal overhead compared to the high rate of acceleration offered by the PKEU. The MPC185 device driver provides the necessary code to prepare data for use by the MPC185 PKEU. There is no plan for changes to the MPC185 silicon to remove this data transformation requirement.

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:
www.freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130

Europe, Middle East, and Africa:
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)

Japan:
Freescale Semiconductor Japan Ltd.
Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-0047 Japan
0120-191014
+81-3-3440-3569

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
852-26668334

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Learn More: For more information about Freescale Semiconductor products, please visit
www.freescale.com

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2004.