

MPC184 Security Processor Device Errata

This document details all known silicon errata for the MPC184 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	Updated Errata 1
4	Added Errata 6
5	Added Errata 7
6	Updated Errata 2-4
7	Added Clarification 8

Errata No. 1: PKEU ROM Failure

Detailed Description and Projected Impact:

The MPC184 PKEU performs complex Montgomery Multiplication and Exponentiation through a combination of hardware and ROM microcode. First silicon on the MPC184 has a manufacturing defect in the ROM which prevents normal operation.

Work-around:

None.

Projected Solution:

This errata was caused by a one time manufacturing error which effected the first wafer lot (MPC184VF) only. Rev.A (MPC184VFA) silicon and all subsequent revisions have fully functional PKEUs.

Errata No. 2: MPC8xx Bus Arbitration Error

Detailed Description and Projected Impact:

The MPC184 8xx Bus Master Interface Block takes two cycles to claim the 8XX bus after grant is active and bus_busy is not active. During this time, a higher priority master (most likely inside the MPC860) could get the grant, see that bus_busy is inactive, and drive the bus at the same time as the MPC184. This can lead to the MPC184 driving data at the same time as other masters, which can lead to corrupted data, time-outs, and system crashes.

Work-around:

The following sections are excerpts from the MPC860 User's Manual. Please refer to sections 11.4.2 and 14.4.6 for additional information.

The SIU module configuration register (SIUMCR) contains bits that configure the following features in the SIU:

- External bus arbitration
- External master support
- Debug and test port configuration
- System interface pin configuration
- Parity support

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	EARB	EARP			—				DSHW	DBGC		DBPC		—	FRC	DLK
Reset	n	000_0000_0								n		n		000		
R/W	R/W															
Addr	(IMMR & 0xFFFF0000) + 0x000															
Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Field	OPAR	PNCS	DPC	MPRE	MLRC		AEME	SEME	BSC	GB5E	B2DD	B3DD	—			
Reset	0000_0000_0000_0000															
R/W	R/W															
Addr	(IMMR & 0xFFFF0000) + 0x002															

Figure 11-3. SIU Module Configuration Register (SIUMCR)

Table 11-3 describes SIUMCR fields.

Table 11-3. SIUMCR Field Descriptions

Bits	Name	Description
0	EARB	External arbitration. For more information, see Section 14.4.6, "Arbitration Phase." The default value depends on the reset configuration; see Section 12.3.1.1, "Hard Reset Configuration Word." 0 Internal arbitration is performed. 1 External arbitration is assumed.
1–3	EARP	External arbitration request priority. Defines the priority of the external master's arbitration request relative to requests by internal modules. Valid when EARB is cleared. 000 = lowest priority and 111 = highest (however, the internal UPM-based refresh cycles always have a higher priority and will preempt any external master if the internal arbiter is used). See Figure 14-21.

Setting the EARP bits to 111 configures the MPC860 to treat the MPC184 as the highest priority master (except for the UPM for SDRAM refresh). This greatly reduces, but doesn't not completely eliminate, the potential of bus collisions.

Projected Solution:

Fixed in the MPC184VFB.

Errata No. 3: PCI Base Address Registers

Detailed Description and Projected Impact:

The MPC184 implements PCI Base Address Registers 0-3. Base Address Register 0 covers the MPC184 Controller and Channel Registers, and Base Address Register 3 covers the 8KB of General Purpose RAM. Both of these regions are considered well-behaved memory, and are therefore designated as prefetchable. Base Address Registers 1 and 2 cover the Execution Units, which include FIFOs and certain registers which are not well-behaved memory, and are therefore designated non-prefetchable. Splitting the prefetchable address space between non-contiguous base address registers is contrary to the logic many system BIOS use when auto configuring the system's global PCI address map. This errata does not effect performance, but does complicate initial configuration.

Work-around:

The system BSP can be written so that the address mapping created by the system BIOS is overwritten with an address map tolerant of the MPC184's prefetchable and non-prefetchable address spaces. The simplest way to do this is to consider Base Address Register 0 non-prefetchable. This will create a contiguous non-prefetchable address space (BAR 0-2) covering the controller, channels, and execution units, while preserving prefetching capability for the gpRAM via Base Address Register 3.

Projected Solution:

Beginning with MPC184VFB, each of the 4 PCI Base Address Registers will be modified to operate independently, and to be configurable to any address.

Errata No. 4: Planned Reversal of Interrupt Default Settings

Detailed Description and Projected Impact:

The MPC184'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:

The default setting of MPC184 interrupt registers will be masked beginning with MPC184VFB.

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

Detailed Description and Projected Impact:

The MPC184VF 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 MPC184 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 MPC184 User's Manual in each execution unit's Interrupt Status Register.

Errata No. 6: Potential MDEU Snooping Error

Detailed Description and Projected Impact:

The MPC184 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 the Descriptors chapter of the MPC184 User’s Manual. When performing ‘out-snooping’ as required for IPSec out-bound processing, it is possible for the MPC184’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 MPC184 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 MPC184, and an examination of MPC184 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.

A large number of concentrated test cases were run on the MPC184 for SDES-HMAC-SHA-1 with out-snooping with large data sizes (4-8KB). No failures were observed in the MPC184 in PCI mode at 66MHz. The same tests were re-run at 33MHz, and intermittent failures were observed at approximately the same rate (.1%) as were seen in the system which first reported this errata. At 33MHz, the PCI bridge/memory appears ‘fast’ to the MPC184, which provides some correlation between the systems. At 33MHz, no failures were observed in SDES-HMAC-SHA-1 for data sizes <4KB, and no failures were observed for 3DES-HMAC-SHA-1 at any data size.

Work-around:

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 will not see this errata, even with SDES-HMAC-SHA-1.

Projected Solution:

A hardware solution is under investigation.

Errata No. 7: PKEU Data Size Limitation:

The MPC184 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. 8: Clarification: PKEU Input Formatting

Detailed Description and Projected Impact:

The MPC184 PKEU is designed to operate on big numbers, represented in strings up to 2048 bits long. The 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 big 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 MPC184 is connected to a little endian processor via the 32 bit PCI interface, the data should be represented in memory as shown below so that the least significant 64-bit Dword is fetched first.

Address	Data
0x0000	3456789a bcdef0f1
0x0008	00000012 abcdef12

In applications in which the MPC184 is connected to a big endian processor via the 32 bit PCI interface, or the 8xx bus interface, the data representation depends on the byte swapping capabilities of the PCI interface or bridge. If little endian to big endian byte swapping is performed automatically, the data should be as shown above. If byte swapping is not performed by the bridge or memory control, 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 MPC184 device driver contains example code for performing the Dword and byte swapping necessary to transform data for use by the MPC184 PKEU. 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 MPC184 PKEU represents minimal overhead compared to the high rate of acceleration offered by the PKEU. The MPC184 device driver provides the necessary code to prepare data for use by the MPC184 PKEU. There is no plan for changes to the MPC184 silicon to remove this data transformation requirement.

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.

MPC184CE
Rev. 7
10/2004