AN2065
Order by AN2065

Freescale Semiconductor, Inc.

Application Note

MPC8xx True Little-Endian Mode

The MPC8xx implements three endian modes:

- Big-Endian (BE)—useful in big-endian systems
- True little-endian (TLE)—useful in little-endian systems
- Modified little-endian (LE)—used in systems where compatibility with 60x processors running in Modified little-endian mode is necessary

This document compares the MPC8xx true little-endian mode with the modified little-endian mode, describes the MPC8xx hardware involved in endian mode support and provides guidelines for software development in true little-endian mode.

1.1 True Little-Endian vs. Little-Endian Mode

The modified little-endian mode achieves the effect of little-endian byte ordering by modifying the low-order three bits of the effective address (EA), shown in Table 1, (this is known as munging).

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

© Motorola Inc., 2001. All rights reserved.

freescale™ semiconductor

For More Information On This Product,
Go to: www.freescale.com

Table 1. Modified Little-Endian EA Modification

Data Length (Bytes)	EA Modification
1	XOR with 0b111
2	XOR with 0b110
4	XOR with 0b100
8	(no change)

The EA modification makes it appear to the processor that individual aligned scalars are stored little-endian, while in fact they are stored big-endian but in different bytes within double words from the order in which they are stored in big-endian mode. This way, little-endian application programs and operating systems can be easily ported to other systems. However, if the storage used by a processor running in little-endian mode is accessed by another agent (processor, bridge,...) designed to operate in little-endian mode the results are not correct, as illustrated in Figure 1.

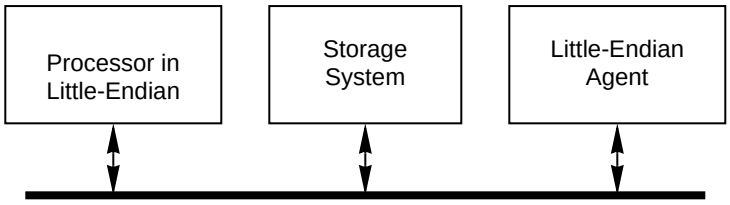

Figure 1. Little-Endian System

Let's assume the content of one of the main processor registers is 0x11223344. This 4 bytes of data is stored at the effective address 0b0000. The address modified according to the Table 1 is 0b100, so the content of the storage system becomes:

0000: xx xx xx xx 11 22 33 44

If the main processor reads a byte from effective address 0b0001 (modified to 0b0110) the result (0x33) is consistent with the little-endian mode of operation.

However, if the little-endian agent reads a word from the address 0b0000 the result (xx xx xx xx) is not the one expected.

To overcome this limitation the MPC8xx implements a true little-endian (TLE) mode. Dedicated hardware is used to swap the byte lanes according to the length of the data which is transferred while the effective address is not modified. This way data is stored in little-endian mode.

Following the same example in TLE mode the result of storing the word 0x11223344 at address 0b0000 will be:

0000: 44 33 22 11 xx xx xx xx.

If the main processor reads a byte from effective address 0b0001 the result will be 0x33; if the little-endian agent reads a word from the address 0b0000 the result will be 0x11223344, both results being correct and consistent with the little-endian mode of operation.

Freescale Semiconductor, Inc.

1.2 MPC8xx Endian Modes Related Hardware

The hardware found in Table 2 supports the various MPC8xx endian modes.

Table 2. Endian Modes Related Hardware

Hardware Resource	Controlled by:	Value
Big-Endian Mode		
SDMA byte lane swapper	BO field of FCR	1x
Modified Little-Endian Mode		
Load-Store address munging (3 bits)	MSR_{LE}	1
Sequencer address munging (3 bits)	MSR_{LE}	1
SDMA address munging (3 bits)	BO field of FCR	01
LCD address munging (3 bits)	$LCHCR_{BO}$	0
True Little-Endian Mode		
Load-Store address munging (2 bits)	$DCCST_{LES}$	1
SIU address de-munging (2 bits) (only for cache initiated accesses)	$DCCST_{LES}$	1
SIU byte lane swapper (only for cache initiated accesses)	$DCCST_{LES}$	1
SDMA byte lane swapper	BO field of FCR	1x

Freescale Semiconductor, Inc.

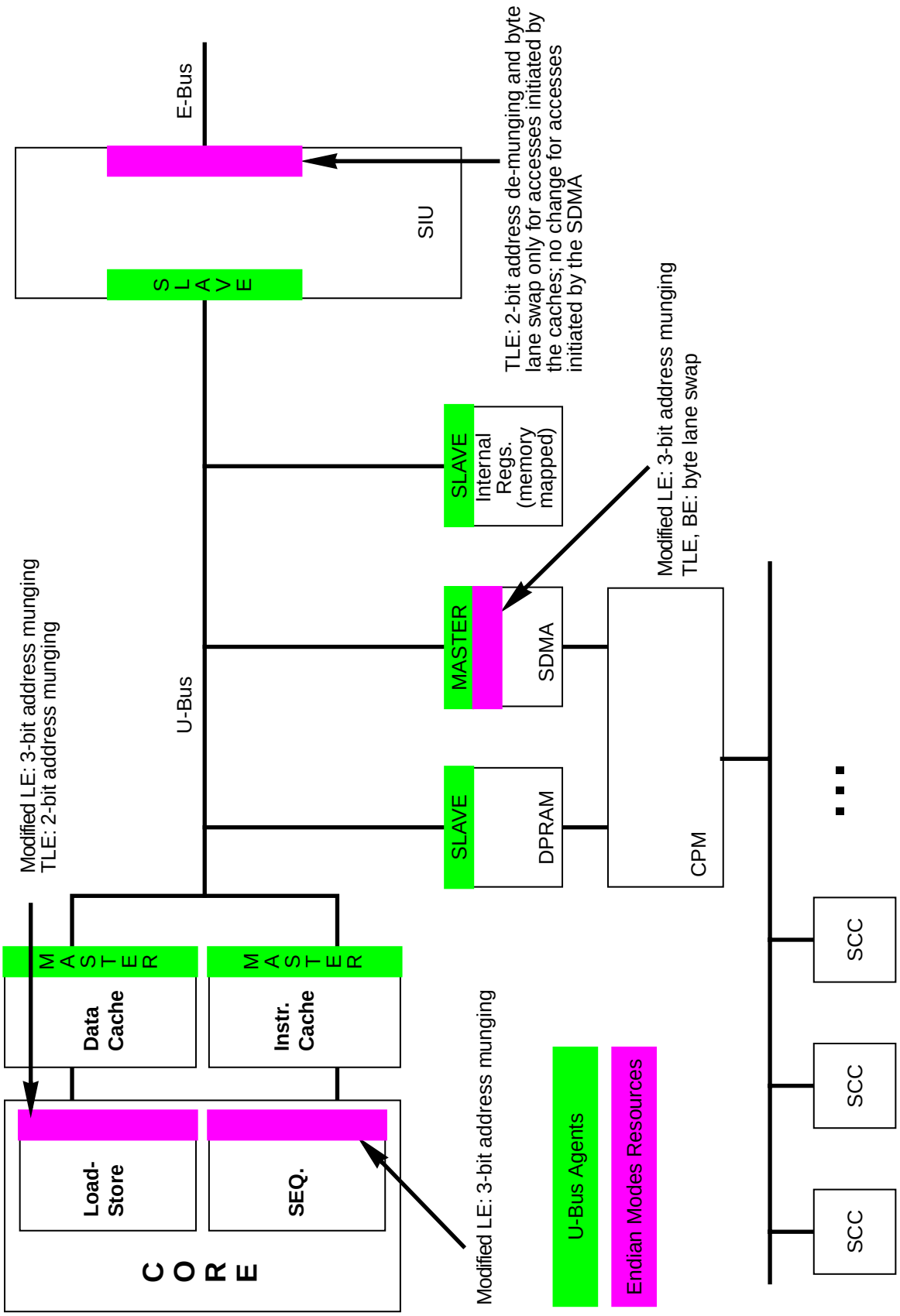

Figure 2. Endian Modes Support Hardware

Freescale Semiconductor, Inc.

1.2.1 Big-Endian Mode

The MPC8xx runs in BE mode if programmed as described in Table 3.

Table 3. Big-Endian Mode Programming

Bit/Field	Value
MSR_{LE}	0
$DCCST_{LES}$	0
FCR_{BO}	1x
$LCHCR_{BO}$	1

In BE mode the SDMA swaps the byte lanes from the order in which the bytes are delivered by the serial interfaces to big-endian order.

The CPM forwards the bytes received from one of the serial interfaces in the order in which they are received. To store them in the external memory in a natural order (first byte received at address ADDR, second byte at address ADDR+1,...) it is necessary to swap the byte lanes according to Figure 3.

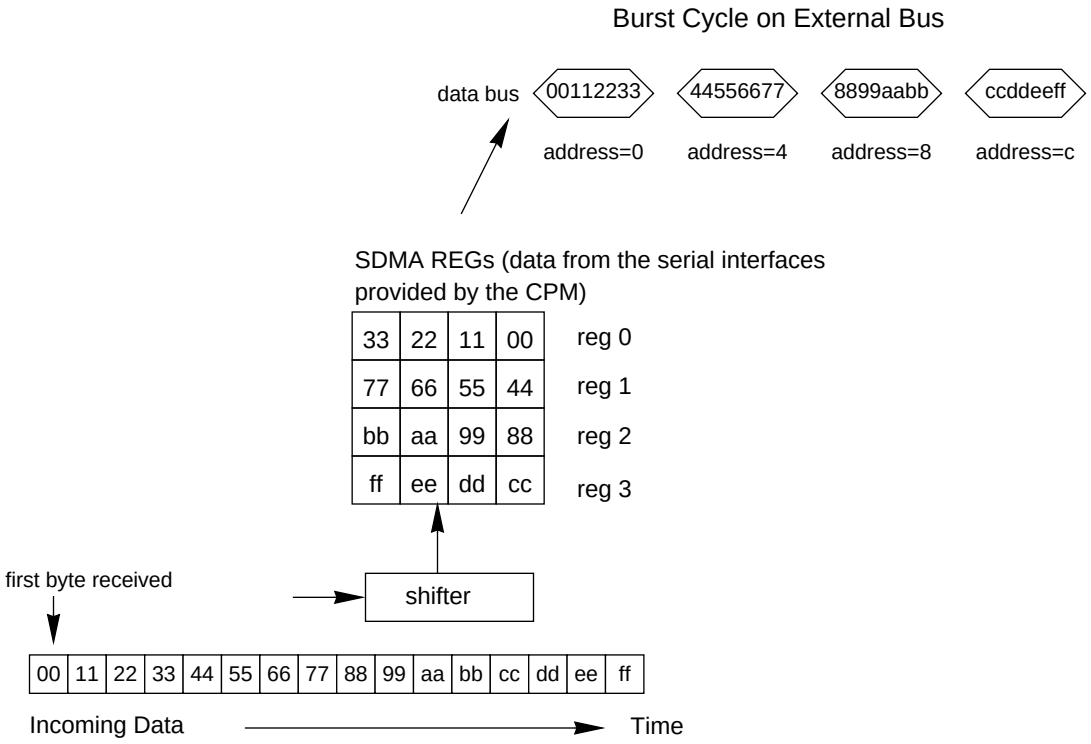

Figure 3. Data Path from Serial Interfaces to Memory

If the CPM has to transfer to the DMA less than one word, the data will be right-aligned (in Figure 3 00 for byte transfers and 11 00 for half word transfers). Table 4 shows the BE/TLE data path between CPM and 32-bit memory.

Freescale Semiconductor, Inc.

Table 4. Big-Endian/True Little-Endian Data Path between CPM and 32-bit Memory

Read/Write Type	Big-Endian Addr.	U-bus Addr	External Bus Addr	Data in the Register				U-bus and E-bus Format			
				0	1	2	3	0	1	2	3
Word	0	0	0	11	12	13	14	14	13	12	11
Half word	0	0	0			21	22	22	21		
Half word	2	2	2			31	32			32	31
Byte	0	0	0				‘a’	‘a’			
Byte	1	1	1				‘b’		‘b’		
Byte	2	2	2				‘c’			‘c’	
Byte	3	3	3				‘d’				‘d’

Data lanes: 0-D0–7, 1-D8–15, 2-D16–23, 3-D24–31

1.2.2 True Little-Endian Mode

The MPC8xx runs in TLE mode if programmed as described in Table 5.

Table 5. True LE Mode Programming

Bit/Field	Value
MSR_{LE}	0
$DCCST_{LES}$	1
FCR_{BO}	1x
$LCHCR_{BO}$	1

In TLE mode the following address/data changes are performed:

- The load-store unit munges the two least significant bits of the address according to Table 6.

Table 6. Address Munging in TLE mode

Data Length (Bytes)	EA Modification
1	XOR with 0b11
2	XOR with 0b10
4	(no change)

- The instruction cache fetch addresses are not modified because they are always 4 bytes long.
- The internal slaves (DPRAM and on-chip memory mapped registers) are addressed using the munged address as described in Table 6.

Freescale Semiconductor, Inc.

- For instruction and data caches access to the external bus (E-Bus) the SIU de-munges the address restoring its initial value and swaps the byte lanes according to the size of the data transferred. Table 7 describes this process.

  The little-endian address is the fetch or load-store address as computed by the sequencer or load-store unit. The U-Bus address is the little-endian address munged by the load-store unit. The external bus address is the U-Bus address de-munged by the SIU (equal to the little-endian address).

  The data in the register is the data as stored in a general-purpose core register. The U-Bus and caches format is identical to the data in the register format. Finally, the E-Bus format is the U-Bus and caches format swapped by the SIU.

Table 7. True Little-Endian Program/Data Path between Register and 32-bit Memory

Fetch/Load Store Type	Little-Endian Addr	U-bus and Caches Addr	External Bus Addr	Data in the Register				U-bus and Caches Format				E-bus Format			
				0	1	2	3	0	1	2	3	0	1	2	3
Word	0	0	0	11	12	13	14	11	12	13	14	14	13	12	11
Half word	0	2	0			21	22			21	22	22	21		
Half word	2	0	2			31	32	31	32					32	31
Byte	0	3	0				‘a’				‘a’	‘a’			
Byte	1	2	1				‘b’			‘b’			‘b’		
Byte	2	1	2				‘c’		‘c’					‘c’	
Byte	3	0	3				‘d’	‘d’							‘d’

Data lanes: 0-D0–7, 1-D8–15, 2-D16–23, 3-D24–31

- The SDMA swaps the byte lanes as in big-endian mode. The data format in the external memory is the same in big- and true little-endian modes:
  - ADDR: first byte received by a serial interface
  - ADDR+1: second byte...

1.2.3 Modified Little-Endian Mode

The MPC8xx runs in modified LE mode if programmed as described in Table 8.

Table 8. Modified-LE Mode Programming

Bit/Field	Value
MSR_{LE}	1
$DCCST_{LES}$	0
FCR_{BO}	01
$LCHCR_{BO}$	0

Freescale Semiconductor, Inc.

In modified LE the three U-Bus masters (data cache, instruction cache and SDMA) will munge the three least significant bits of the address as a function of the size of the data transferred, according to Table 1.

1.3 True Little-Endian Mode Transfer Examples

The examples in this section assume the memory has a 32-bit port width.

1.3.1 Core Registers to/from External Memory

The code from Table 9 illustrates such transfers.

Table 9. Examples of Transfers between Core Registers and External Memories

Instruction	Register Value after the Execution of the Instruction (hex)	E-Bus Address (hex)	E-Bus Data (hex)	Transfer Size
xor r30,r30,r30				
oris r30,r30,0x1234				
ori r30,r30,0x5678	r30=12345678			
xor r31,r31,r31				
oris r31,r31,0x000f				
ori r31,r31,0xf000	r31=000ff000			
stw r30,0x0000(r31)	r30=12345678	000ff000	78563412	Word
sth r30,0x0010(r31)	r30=12345678	000ff010	**7856**5678	Half word
sth r30,0x0022(r31)	r30=12345678	000ff022	7856**7856**	Half word
stb r30,0x0030(r31)	r30=12345678	000ff030	**78**345678	Byte
stb r30,0x0041(r31)	r30=12345678	000ff041	78**78**7812	Byte
stb r30,0x0052(r31)	r30=12345678	000ff052	7878**78**34	Byte
stb r30,0x0063(r31)	r30=12345678	000ff063	78783**78**	Byte
lwz r29,0x0100(r31)	r29=**10203040**	000ff100	40302010	Word
lhz r28,0x0110(r31)	r28=0000**1314**	000ff110	**1413**1211	Half word
lhz r27,0x0122(r31)	r27=0000**2122**	000ff122	2423**2221**	Half word
lbz r26,0x0130(r31)	r26=00000**34**	000ff130	**34**333231	Byte
lbz r25,0x0141(r31)	r25=00000**43**	000ff141	44**43**4241	Byte
lbz r24,0x0152(r31)	r24=00000**52**	000ff152	5453**52**51	Byte
lbz r23,0x0163(r31)	r23=00000**61**	000ff163	64636**61**	Byte

Freescale Semiconductor, Inc.

1.3.2 Core Registers to/from On-Chip Memory-Mapped Storage (DPRAM and Registers)

The addresses of the on-chip memory-mapped registers and locations in DPRAM must be redefined according to Table 10.

Table 10. Internal Addresses in True Little-Endian Mode

Register/DPRAM Field Size (Bytes)	Address in True Little-Endian Mode
1	Address in big-endian mode XOR 0b11
2	Address in big-endian mode XOR 0b10
4	(no change)

As an example, as defined in the *MPC860 PowerQUICC User Manual*, the RTCSC register could be located at 0xFF000220 (assuming the Internal Space Base Address is oxFF000000). This register is 16 bits wide and according to Table 10, the software must access it at address 0xFF000222. The core will munge 0xFF000222 to a U-bus address of 0xFF00220.

Accessing bytes or half words in larger on-chip memory mapped storage elements, Figure 4, is consistent with the little-endian conventions.

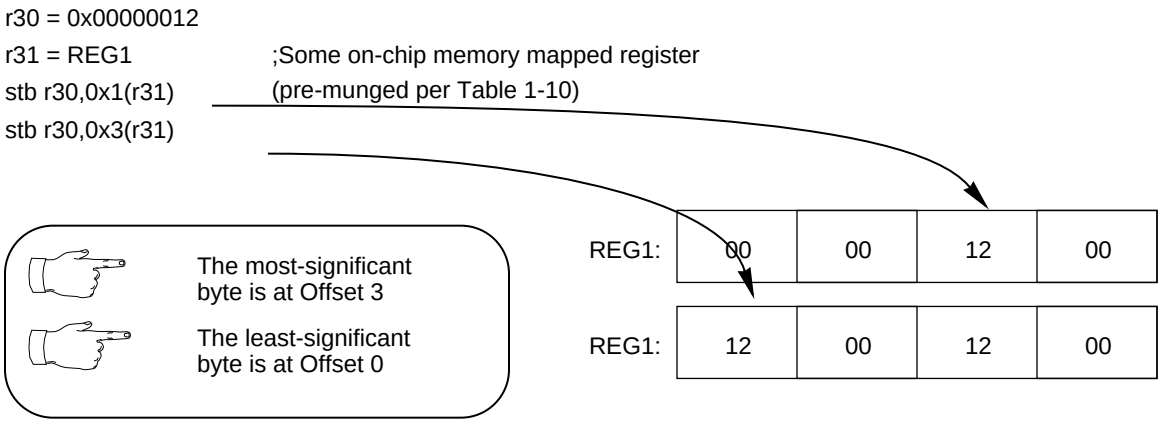

Figure 4. Accessing Bytes in Larger On-Chip Memory Mapped Storage Elements

Freescale Semiconductor, Inc.

The code from Table 11 illustrates possible cases of transfers to/from on-chip memory mapped storage (DPRAM and on-chip memory mapped registers).

Table 11. Transfers between Core Registers and On-Chip Memory-Mapped Storage

Instruction	Core Register Value after the Execution of the Instruction (hex)	U-Bus Address (hex)	U-Bus Data (hex)	Transfer Size	On-Chip Memory-Mapped Register Value after the Execution of the Instruction (hex)
stw r30,0x000**c**(r31)	r30=**12345678**	fff0010**c**	12345678	Word	12345678
sth r30,0x001**4**(r31)	r30=1234**5678**	fff0011**6**	1234**5678**	Half word	0000**5678**
sth r30,0x001**e**(r31)	r30=1234**5678**	fff0011**c**	**5678**1234	Half word	**5678**0000
stb r30,0x002**4**(r31)	r30=12345**678**	fff0012**7**	12345**78**	Byte	00000**78**
stb r30,0x002**d**(r31)	r30=12345**678**	fff0012**e**	3456**78**12	Byte	0000**78**00
stb r30,0x003**6**(r31)	r30=12345**678**	fff0013**5**	56**78**1234	Byte	00**78**0000
stb r30,0x003**f**(r31)	r30=12345**678**	fff0013**c**	**78**123456	Byte	**78**000000
re-initialization code removed					
lwz r10,0x000**c**(r31)	r10=**12345678**	fff0010**c**	12345678	Word	0x**12345678**
lhz r9,0x001**4**(r31)	r9=0000**5678**	fff0011**6**	1234**5678**	Half word	0x1234**5678**
lhz r8,0x001**e**(r31)	r8=0000**1234**	fff0011**c**	**1234**5678	Half word	0x**1234**5678
lbz r7,0x002**4**(r31)	r7=00000**78**	fff0012**7**	12345**78**	Byte	0x12345**78**
lbz r6,0x002**d**(r31)	r6=00000**56**	fff0012**e**	1234**56**78	Byte	0x1234**56**78
lbz r5,0x003**6**(r31)	r5=00000**34**	fff0013**5**	12**34**5678	Byte	0x12**34**5678
lbz r4,0x003**f**(r31)	r4=00000**12**	fff0013**c**	**12**345678	Byte	0x**12**345678

1.3.3 Serial Channels to/from Internal/External Memory

Data received from the serial channels and transferred to the external or internal (DPRAM) memories by the SDMA is stored in the same order as received (in both true little-endian and big-endian modes). The same transfers occur in opposite direction for data transmitted by the serial channels. Figure 5 and Table 12 illustrates such transfers.

Freescale Semiconductor, Inc.

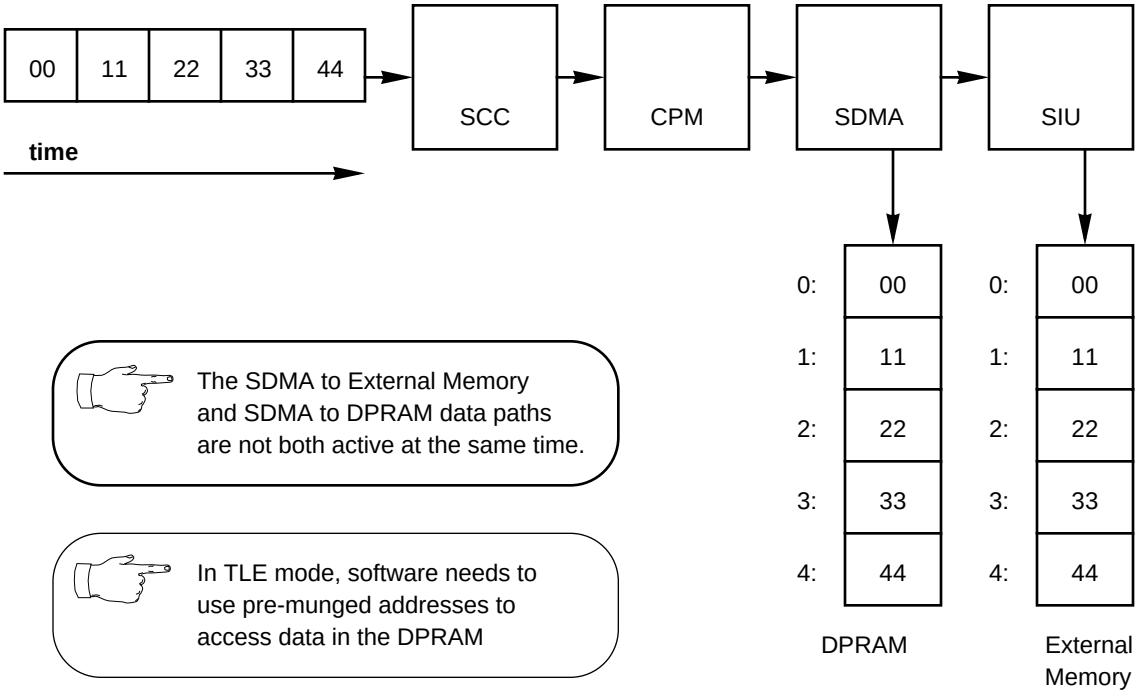

Figure 5. Data Path from Serial Channels to Internal/External Storage Serial Channels to/from Core Registers

Table 12. Transfers between Serial Channels and Internal/External Storage

Data Received by the SCC (hex) (Byte 35 is received first)	Data Transferred by the CPM to the SDMA (hex)	Address Transferred by the CPM to the SDMA (hex)	Transfer Size	U-bus and E-Bus Address (hex)	U-bus and E-Bus Data (hex)
38373635	38373635	fff02810	Word	fff02810	35363738
3635	3837**3635**	fff02810	Half word	fff02810	**3536**3738
3635	3837**3635**	fff02812	Half word	fff02812	3738**3536**
35	38373**635**	fff02810	Byte	fff02810	**35**363738
35	38373**635**	fff02811	Byte	fff02811	36**35**3837
35	38373**635**	fff02812	Byte	fff02812	3738**35**36
35	38373**635**	fff02813	Byte	fff02813	38373**635**

Freescale Semiconductor, Inc.

Data from the serial channels can be accessed by the core using either the DPRAM or the external memory as a temporary storage; refer to Figure 6.

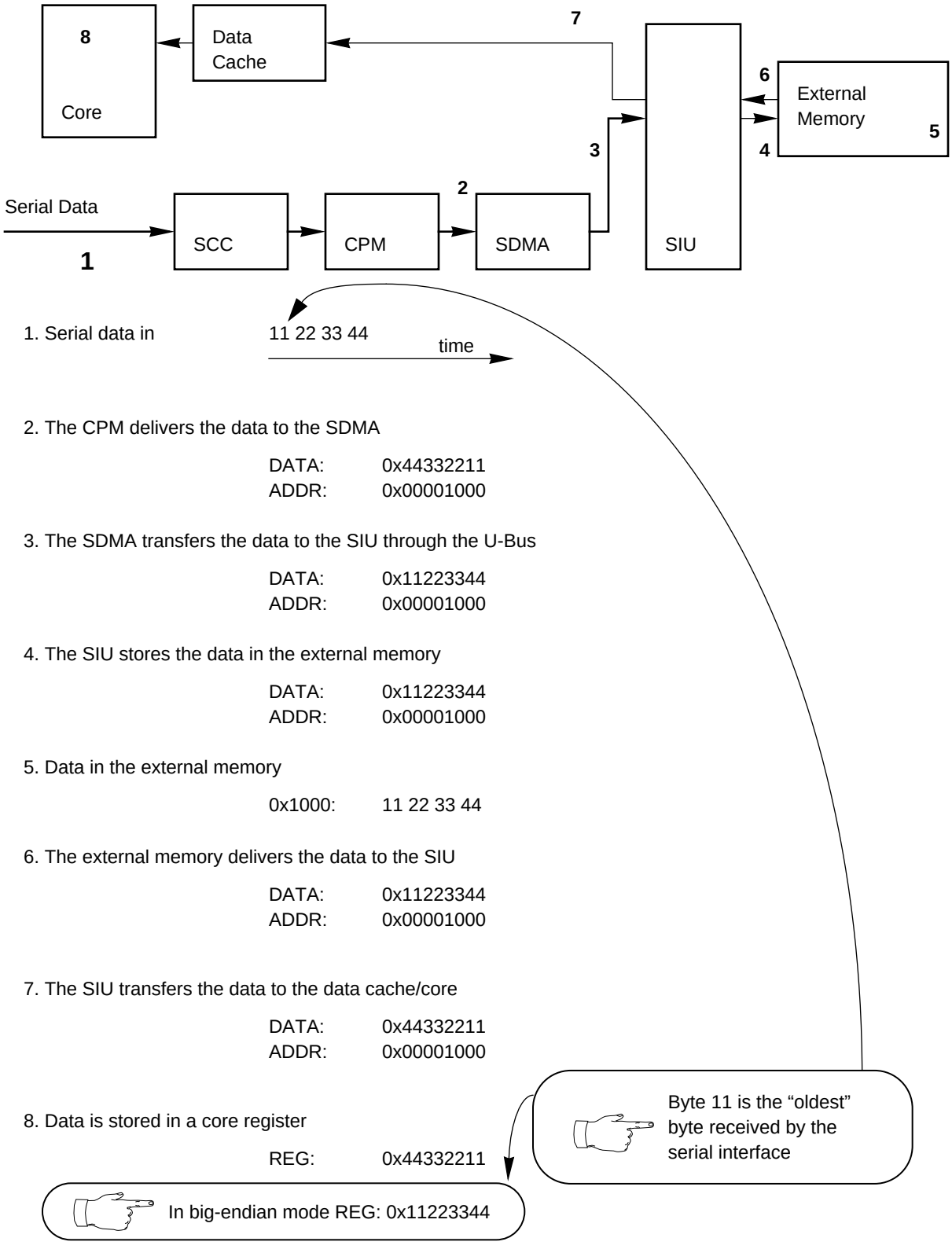

Figure 6. Data Flow between Serial Channels and Core Registers

Freescale Semiconductor, Inc.

1.4 Entering True Little-Endian Mode

To enter true little-endian mode a “Set Little-Endian Swap Mode” command must be issued.

Because of the pipeline structure of the MPC8xx the “Set Little-Endian Swap Mode” command issued to the DCCST register might take effect only after executing one or several more instruction which follows it. Because of this, after the **mtspr DCCST,reg** the **isync** instruction must be placed. If executed in big-endian mode, this instruction will synchronize the core and the next instruction will be executed in little-endian mode. If **cmpi 0,r1,0x4c** is executed instead of **isync** it means that the processor is already in little-endian mode and no further synchronization is necessary.

Table 13 provides a possible sequence of code.

Table 13. Possible Core-Dependent Code Sequence to Enter TLE Mode

Address	Big-Endian Interpretation	Little-Endian Interpretation	Remarks
ADDR - 4:	whatever		Instructions executed in big-endian mode
ADDR:	xori r0,r0,0x0		
ADDR + 4:	addis r0,r0,0x0500		
ADDR + 8:	mtspr DCCST,r0		
ADDR + 12:	isync (opcode: 0x4c00012c)	cmpi 0,r1, 0x4c (opcode: 0x2c01004c)	Instruction executed in either big-endian or little-endian mode
ADDR + 16:		whatever	Instructions executed in little-endian mode

The programmer must ensure that this code will not be executed after switching to little-endian mode.

Sometimes it is preferable to have a routine which brings the MPC8xx to true little-endian mode disregarding in which mode the core runs. For example such a routine can be placed at interrupt vector 0x100 where the core might branch after hardware reset (in big-endian mode) or after software reset (already in little-endian mode).

Table 14 provides an example of such a routine (assuming the opcodes are stored in little-endian mode in the program memory).

Table 14. Possible Core-Independent Routine to Enter TLE Mode

Address	Instruction	Flow in Big-Endian (Executed after Hard Reset)	Flow in True Little-Endian (Executed after Soft Reset)
0x00000100:	.long 0x48000020	subfic r0,r0, 0x48	b $+0x20
0x00000104:	.long 0x3c05003c	addis r0,r0,0x0500	Not executed
0x00000108:	.long 0xa68b187c	mtspr DCCST,r0	Not executed
0x0000010c:	.long 0x2c01004c	isync or cmpi 0,r1,0x4c	Not executed
0x00000110:	.long 0x2c01004c	isync # no-operation	Not executed
0x00000114:	.long 0x2c01004c	isync # no-operation	Not executed

Freescale Semiconductor, Inc.

Table 14. Possible Core-Independent Routine to Enter TLE Mode (Continued)

0x00000118:	.long 0x2c01004c	isync # no-operation	Not executed
0x0000011c:	.long 0x2c01004c	isync # no-operation	Not executed
0x00000120:	# Reset Routine in True Little-Endian format		

1.5 Performance Issues

MPC8xx performance is the same in big- and true little-endian modes.

In modified little-endian mode the performance is slightly affected because the stream hit mechanism is less efficient than in big- and true little-endian modes, as illustrated in Figure 7 and Figure 8.

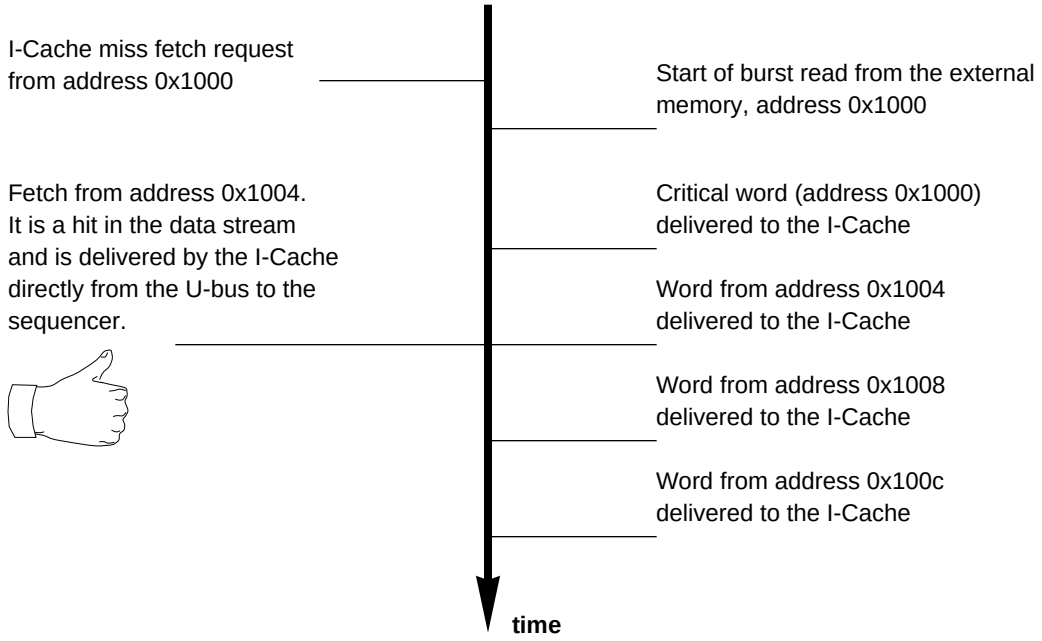

Figure 7. Stream Hit Mechanism in Big- or True Little-Endian Modes

Freescale Semiconductor, Inc.

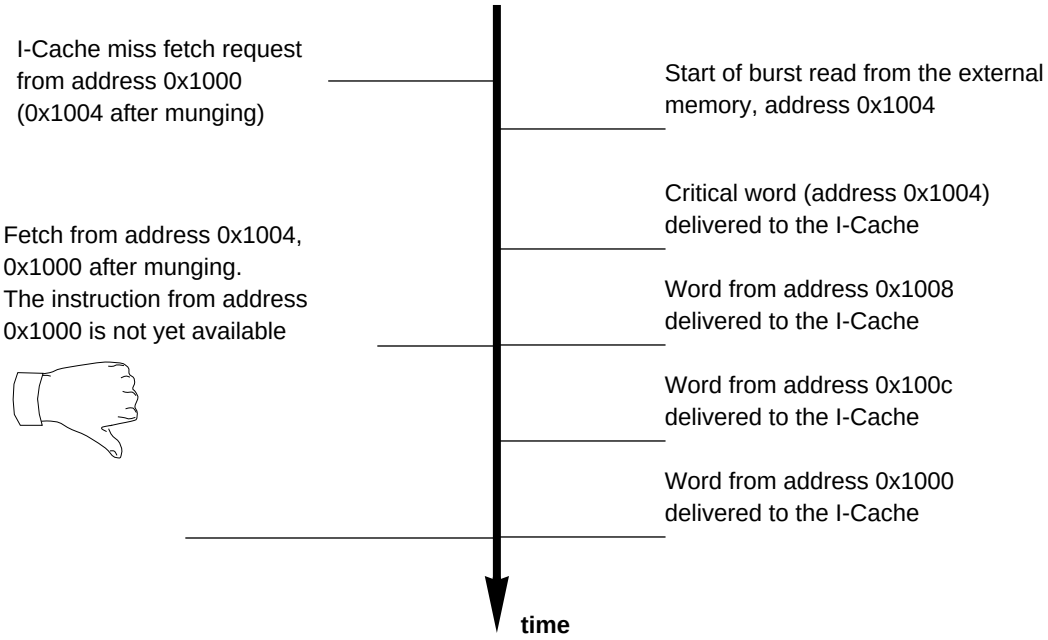

Figure 8. Stream Hit Mechanism In Modified Little-Endian Mode

The actual performance penalty depends upon the application (stream hit-ratio, memories access time).

Freescale Semiconductor, Inc.

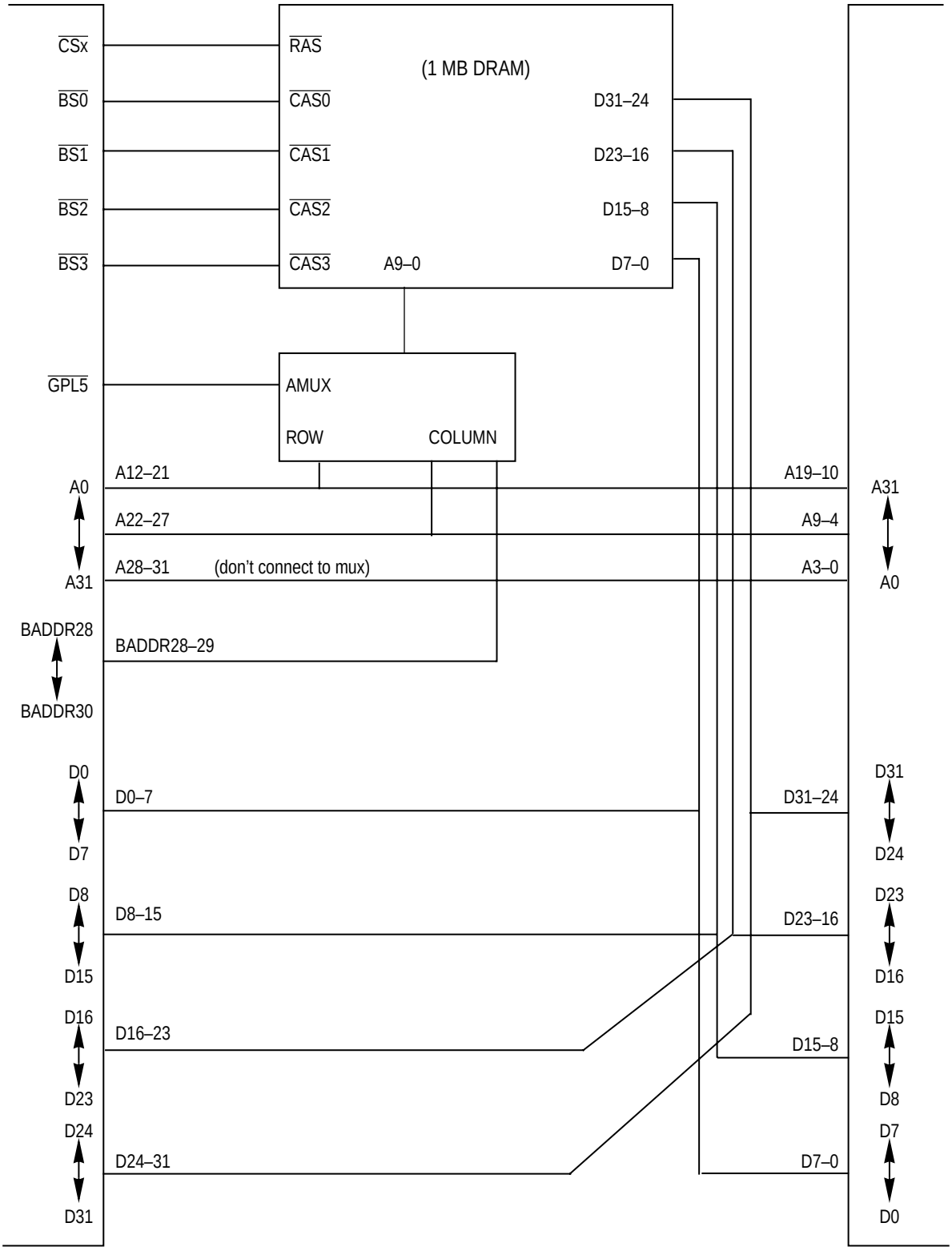

Figure 9. Shared DRAM and External Bus between MPC8xx (TLE mode) and Little-Endian External Master

Freescale Semiconductor, Inc.

Table 15. MPC8xx True Little-Endian External Bus

D(0-7)	LSB	Corresponds to BS0
D(8-15)		Corresponds to BS1
D(16-23)		Corresponds to BS2
D(24-31)	MSB	Corresponds to BS3

The appropriate byte lanes of the MPC8xx's external bus must be connected to the appropriate byte lanes of the little-endian external master (that is, MSB to MSB, LSB to LSB, etc.) Thus, when any size transfer is initiated (by the internal or external master), the appropriate byte-select signals will assert; refer to Table 16.

Table 16. MPC8xx TLE Data Bus Connections to External LE Master

MPC8xx (in TLE mode)	External LE master
D0	D7
D1	D6
D2	D5
D3	D4
D4	D3
D5	D2
D6	D1
D7	D0
D8	D15
D9	D14
D10	D13
D11	D12
D12	D11
D13	D10
D14	D9
D15	D8
D16	D23
D17	D22
D18	D21
D19	D20
D20	D19

Freescale Semiconductor, Inc.

Table 16. MPC8xx TLE Data Bus Connections to External LE Master (Continued)

D21	D18
D22	D17
D23	D16
D24	D31
D25	D30
D26	D29
D27	D28
D28	D27
D29	D26
D30	D25
D31	D24

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

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.

freescale™
semiconductor