AN1852
Rev. 3, 11/2005

MSC8101 Software Initialization Overview

By Donald Simon

Using one specific reset scenario, this application note explains how an application on the MSC8101 begins execution. The last section of this application note contains troubleshooting suggestions for getting an application running on an MSC8101-based board. The MSC8101 device is identical to the MSC8103, except that the MSC8101 includes an enhanced filter coprocessor (EFCOP).

Further information can be found in the *MSC8101 Reference Manual*. This application note does not replace any information in the *MSC8101 Reference Manual*.

The MSC8101 reset initialization sequence is similar (but not identical) to that of the MPC8260 60x system bus. The system interface unit (SIU), communications processor module (CPM), and reset internal blocks of the MSC8101 are based on the corresponding blocks in the MPC8260.

CONTENTS

1 Reset Initialization Flow

The “Reset” and “Clocks” chapters in the *MSC8101 Reference Manual* show all possible reset causes and detailed timing information. The power-on reset case using the hardware reset configuration is the topic of this application note because it is the most common reset scenario. Reset configuration can come from the host port or the 60x system bus. The hardware reset configuration comes from the 60x system bus. **Figure 1** shows the power-on reset timing diagram using the 60x system bus.

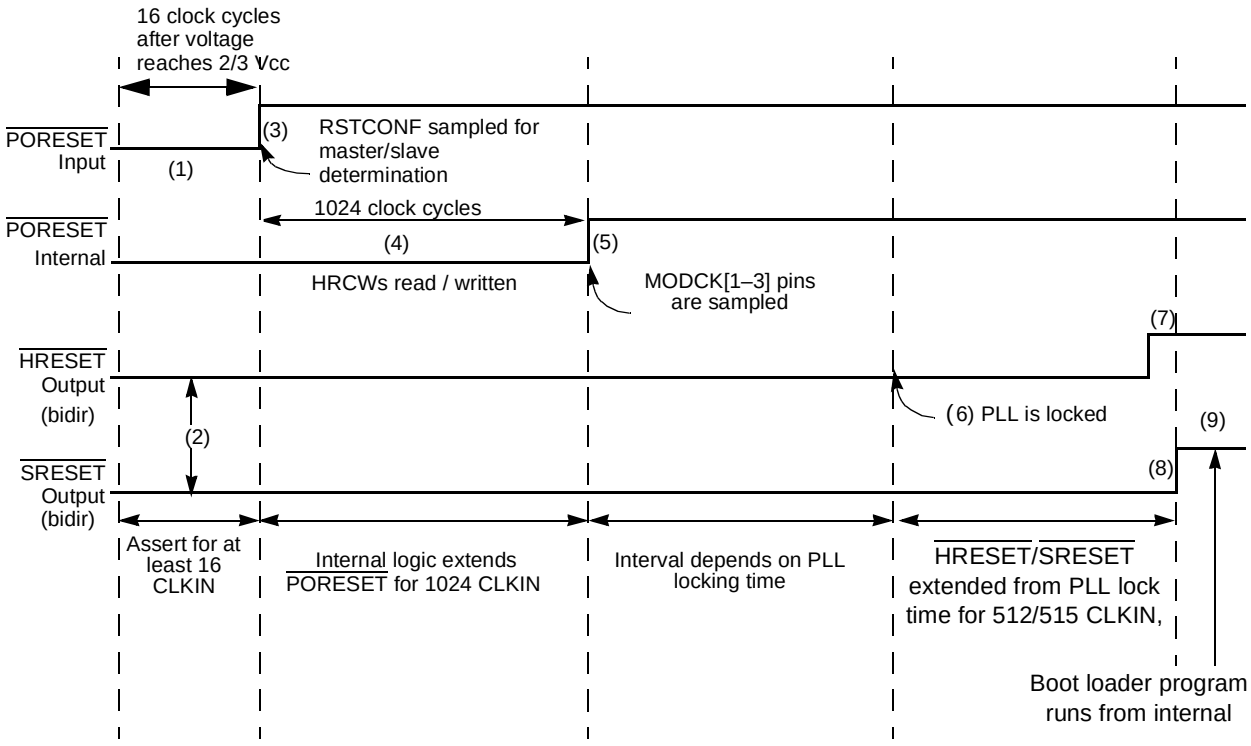

Figure 1. Timing Diagram for MSC8101 Power-On Reset Using Hardware Reset Configuration

The sequence depicted in **Figure 1** is described as follows.

1. The power-on reset pin ($\overline{\text{PORESET}}$) starts the initialization process on the MSC8101. $\overline{\text{PORESET}}$ should be asserted externally for at least 16 clock cycles after external core power reaches at least 2/3 V_{DD}.
2. The MSC8101 asserts both the $\overline{\text{HRESET}}$ and $\overline{\text{SRESET}}$ pins. $\overline{\text{HRESET}}$ and $\overline{\text{SRESET}}$ are bidirectional pins and require open collector drivers if they are driven externally.
3. $\overline{\text{RSTCONF}}$ is sampled on the rising edge of $\overline{\text{PORESET}}$ for MSC8101 master/slave determination. If $\overline{\text{RSTCONF}}$ is found high, there is only one MSC8101 in the system and the Hard Reset Configuration Word (HRCW) is cleared (all zeros). If $\overline{\text{RSTCONF}}$ is found low, eight HRCWs are read (corresponding to one master and seven slaves). Refer to **Section 2,** *Hard Reset Configuration Word* in this application note for information on how the HRCWs are read.
4. Internally, the MSC8101 holds $\overline{\text{PORESET}}$ low for 1024 CLKIN cycles. This $\overline{\text{PORESET}}$ is an internal signal.
5. When the internal $\overline{\text{PORESET}}$ line goes high, the MODCK[1–3] pins are sampled. These three pin values and MODCK_H in the HRCW define the core, CPM, and bus clock frequencies.
6. The Phase-Lock Loop (PLL) is locked.
7. The MSC8101 deasserts $\overline{\text{HRESET}}$.
8. Three cycles later, the MSC8101 deasserts $\overline{\text{SRESET}}$.
9. The bootloader program runs from internal ROM starting at address 0xF80000.
10. The bootloader then jumps to an address (vector) as defined in the address table. The vector is determined by multiplying the ISB bits in the HRCW by four and adding this to the starting address in the address table (that is, 0xFE000110). The specified vector location must contain the address of the first instruction to be executed from external memory.

All operations during the power-on reset sequence up until the internal bootloader executes are performed by hardware using the external system clock. No code (that is, internal firmware) is executed during this period. The MSC8101 does not read the status of the MODCK[1–3] pins until after reading the HRCW.

2 Hard Reset Configuration Word

The HRCW defines many of the attributes of the MSC8101, including the location of the internal memory map (ISB) and clocking options (MODCK_H). The fields in the HRCW are distributed to various register fields in the MSC8101 to define initial operation. Therefore, setting up the HRCW correctly is critical in getting the MSC8101 operational. Software debuggers must allow you to set various fields in the configuration word before the application is downloaded to the MSC8101 target board.

If $\overline{\text{RSTCONF}}$ is sampled high when $\overline{\text{PORESET}}$ goes high, there is only one MSC8101 in the system and the configuration word uses the default of all zeros. If $\overline{\text{RSTCONF}}$ is sampled low, the configuration master reads eight configuration words, regardless of the number of MSC8101 processors in the system. The MSC8101 slave $\overline{\text{RSTCONF}}$ inputs are enabled through one of the A[0–6] address lines. Refer to the *MSC8101 Reference Manual* for details.

The sequence for reading the configuration words is as follows:

1. The MSC8101 asserts Chip Select 0 ($\overline{\text{CS0}}$) and reads the master configuration word, which is four bytes. These bytes are located at addresses 0x00, 0x08, 0x10, and 0x18.
2. Next, seven MSC8101 slave HRCWs are read the same way as in step 1. MSC8101 slaves 1–7 start at addresses 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0, respectively.

After every slave configuration word read, the MSC8101 master puts the configuration word back on the data bus as a word (32 bits), then toggles the corresponding address pin, starting with address line A0 and ending with A6. The process is repeated for seven MSC8101 slaves. On the rising edge of the MSC8101 slave $\overline{\text{RSTCONF}}$ line, the slave MSC8101 latches in the configuration word.

Figure 2 shows an MSC8101 timing diagram with multiple MSC8101s in the system. Notice the $\overline{\text{PORESET}}$ line going high. It is not shown in **Figure 2**, but $\overline{\text{RSTCONF}}$ is tied low corresponding to an MSC8101 master. Also notice $\overline{\text{CS0}}$ pulsing low eight times and address lines A[0–6] pulsing low one at a time.

On the rising edge of $\overline{\text{PORESET}}$, $\overline{\text{RSTCONF}}$ is sampled low, meaning that this MSC8101 is the master. $\overline{\text{CS0}}$ is asserted eight times corresponding to eight HRCW reads (four bytes per HRCW read). The first time $\overline{\text{CS0}}$ pulses low, the HRCW for the master is read. The second time $\overline{\text{CS0}}$ pulses low, the second HRCW is read. When $\overline{\text{CS0}}$ pulses high, the MSC8101 master asserts address line A0. This allows the MSC8101 slave to read its HRCW from the master. The third time $\overline{\text{CS0}}$ pulses low, the process repeats for the second MSC8101 slave using address A1. The process repeats for a total of seven slaves.

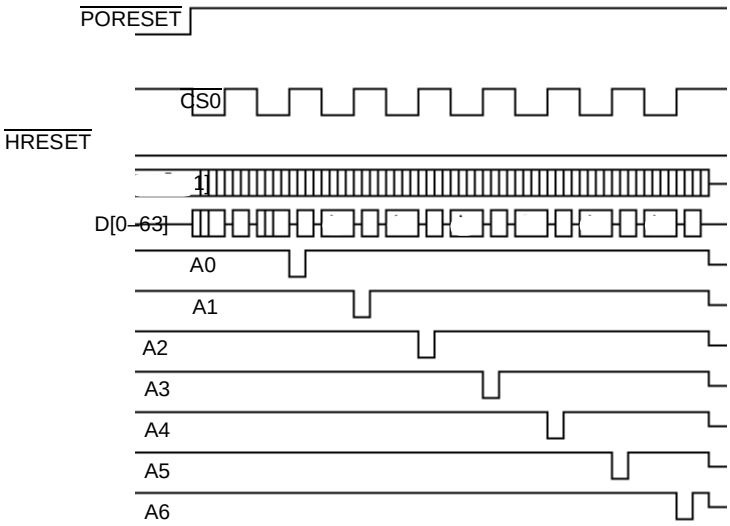

Figure 2. Power-On Reset With Multiple MSC8101 Devices

Signal Names	Notes
PORESET	Shows where power is applied to the MSC8101
CS0	There are eight clock pulses. During the first clock pulse, the master MSC8101 HRCW is read. During the seven clock pulses that follow, the master MSC8101 reads the seven slave MSC8101 HRCWs. During the low clock pulse, a single HRCW is read. During the high clock pulse, the master MSC8101 writes the HRCW to the data bus and asserts one address line.
HRESET	By the time HRESET is deasserted, the HRCW(s) and MODCK[1–3] pins are read, and the PLL is locked.
A[0–6]	MSC8101 address lines

2.1 HRCW Bit Definitions

Figure 3. Hard Reset Configuration Word

	Bit 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	EARB	EXMC	IRQ7 INT	EBM	BPS		SCDIS	ISPS	IRPC		DPPC		NMI OUT	ISB		
Type	R/W															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
	—	BBD	MMR		—		TCPC		BC1PC		—	DLLDIS	MODCK_H			—
Type	R/W															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 1 lists the bit definitions in the HRCW. This information is also in the “Reset” chapter in the *MSC8101 Reference Manual*. MODCK_H in the HRCW and the MODCK[1–3] pins set the initial MSC8101 operating mode. For details, refer to the Clocks chapter in the *MSC8101 Reference Manual.*

Table 1. Hard Reset Configuration Word Bit Descriptions

Name	Reset	Description	Settings
EARB 0	0	**External Arbitration** Defines the initial value for ACR[EARB].	0 No external arbitration is assumed 1 External arbitration is assumed
EXMC 1	0	**External MEMC** Defines the initial value of BR0[EMEMC].	0 No external memory controller is assumed 1 External memory controller is assumed
IRQ7 INT 2	0	**$\overline{\text{IRQ7}}$ or $\overline{\text{INT\_OUT}}$ Selection**	0 $\overline{\text{IRQ7}}$/$\overline{\text{INT\_OUT}}$ pin is $\overline{\text{IRQ7}}$ 1 $\overline{\text{IRQ7}}$/$\overline{\text{INT\_OUT}}$ pin is $\overline{\text{INT\_OUT}}$
EBM 3	0	**External PowerPC Bus Mode** Defines the initial value of BCR[EBM].	0 Single-master bus mode 1 Multimaster bus mode
BPS 4–5	0	**Boot Port Size** Defines the initial value of BR0[PS], the port size for memory controller bank 0.	00 64-bit port size 01 8-bit port size 10 16-bit port size 11 32-bit port size
SCDIS 6	0	**SC140 Disabled** Enables/disables the SC140.	0 SC140 enabled 1 SC140 disabled
ISPS 7	0	**Internal Space Port Size** Defines the initial value of BCR[ISPS].	0 64-bit data bus 1 32-bit data bus
IRPC 8–9	0	**Interrupt Pin Configuration** Defines the initial value of SIUMCR[IRPC], which defines the configuration of the Reserved/BADDR[29–31]/IRQ[2, 3, 5] pins	00 Pins are Reserved 01 Pins are IRQ2, IRQ3, and IRQ5 10 Pins are BADDR[29–31] 11 Value is reserved
DPPC 10–11	0	**Data Parity Pin Configuration** Defines the initial value of SIUMCR[DPPC], which specifies the configuration of the eight external Data Parity (DP[0–7]) pins. The additional arbitration lines (EXT_BR2, EXT_BG2, EXT_DBG2, EXT_BR3, EXT_BG3, and EXT_DBG3) are operational only when ACR[EARB] = 0. Setting EARB (to choose external arbiter) combined with programming DPPC to 11 deactivates these lines	00 DP0 is Reserved; DP[1–7] are $\overline{\text{IRQ[1–7]}}$ 01 Pins are DP[0–7] 10 DP0 is Reserved; DP1 is $\overline{\text{IRQ1}}$; DP[2–3] are Reserved; DP[4–5] are DREQ[3–4]; DP[6–7] are DACK[3–4] 11 DP0 is $\overline{\text{EXT\_BR2}}$; DP1 is $\overline{\text{EXT\_BG2}}$; DP2 is $\overline{\text{EXT\_DBG2}}$; DP3 is $\overline{\text{EXT\_BR3}}$; DP4 is $\overline{\text{EXT\_BG3}}$; DP5 is EXT_DBG3; DP6 is $\overline{\text{IRQ6}}$; DP7 is $\overline{\text{IRQ7}}$
NMI OUT 12	0	**NMI Out to External Handler** Defines whether a non-maskable interrupt ($\overline{\text{NMI}}$) event is handled by the SC140 core or an external host.	0 $\overline{\text{NMI}}$ is serviced by SC140 core 1 $\overline{\text{NMI}}$ is routed to external pin $\overline{\text{NMI\_OUT}}$ and serviced by the external host
ISB 13–15	0	**Initial Internal Space Base Select** Defines the initial value of IMMR[0–14] and determines the base address of the internal memory space. The SC140 core internal address space spans from 0x00000000–0x00FFFFFF (16 MB). Therefore, it is not advisable to map the IMMR in this space, since the SC140 core will not be able to access the SIU and CPM registers.	000 0xF0000000 001 0xF0F00000 010 0xFF000000 011 0xFFF00000 100 Reserved. Do not use. 101 0x00F00000 110 0x0F000000 111 0x0FF00000
— 16	0	Reserved. Write to zero for future compatibility.	
BBD 17	0	**Bus Busy Disable** Defines the initial value of SIUMCR[BBD].	0 $\overline{\text{ABB}}$/$\overline{\text{IRQ2}}$ is $\overline{\text{ABB}}$, $\overline{\text{DBB}}$/$\overline{\text{IRQ3}}$ is $\overline{\text{DBB}}$. 1 $\overline{\text{ABB}}$/$\overline{\text{IRQ2}}$ is $\overline{\text{IRQ2}}$, $\overline{\text{DBB}}$/$\overline{\text{IRQ3}}$ is $\overline{\text{IRQ3}}$.

Table 1. Hard Reset Configuration Word Bit Descriptions (Continued)

Name	Reset	Description	Settings
MMR 18–19	0	**Mask Masters Requests** Defines the initial value of SIUMCR[MMR].	00 No masking on bus request lines 01 Reserved 10 Reserved 11 All external bus requests masked (boot master is the internal core)
— 20–21	0	Reserved. Write to zero for future compatibility.	
TCPC 22–23	0	**Transfer Code Pin Configuration** Defines the initial value of SIUMCR[TCPC] after reset. Determines whether the MODCK/TC/BNKSEL pins support Transfer Code (TC[0–2]) or SRAM Bank Select (BNKSEL[0–2]) signals.	00 Selects TC[0–2] 01 Reserved 10 Selects BNKSEL[0–2] 11 Reserved
BC1PC 24–25	0	**BC1PC Value** Defines the initial value of SIUMCR[BC1PC].	00 BCTL1 01 BCTL1 10 Reserved 11 Reserved
— 26	0	Reserved. Write to zero for future compatibility.	
DLLDIS 27	0	**DLL Disable** Defines whether the DLL mechanism is disabled.	0 No DLL bypass 1 DLL bypass
MODCK_H 28–30	0	**MODCK High Order Bits** High-order bits of the MODCK bus, which determine the clock reset configuration.	See the *MSC8101 Technical Data* sheet for details.
— 31	0	Reserved. Write to zero for future compatibility.	

All the bits in the HRCW except the ones listed in **Table 2** are distributed to other registers in the MSC8101. These bits can be changed only by going through a power-on reset sequence.

Table 2. HRCW Bits Not Distributed to Other Registers

Field Name	HRCW Bit Numbers
SCDIS	6
NMI OUT	12
DLLDIS	27
MODCK_H	28–30

2.2 HRCW Example

The following example explains how the MSC8101 reads the HRCW.

1. On power up, the MSC8101 reads the HRCW by asserting $\overline{\text{CS0}}$. This corresponds to address 0xFF800000 on the MSC8101ADS board.
2. The MSC8101 master asserts $\overline{\text{CS0}}$ seven more times to read the HRCWs for the slaves.
3. A debugger is used to read the contents starting at 0xFF800000.

Refer to **Section 2**, *Hard Reset Configuration Word*, in this application note for an explanation of the byte ordering of the HRCW.

```
ff800000: 0c000000 00000000 32000000 00000000
ff800010: 0e000000 00000000 05000000 00000000
ff800020: ffffffff ffffffff ffffffff ffffffff
ff800030: ffffffff ffffffff ffffffff ffffffff
```

Taking every eighth byte starting at address 0xFF800000, the HRCW in this example is 0x0C320E05. These bits are shown in **bold** in the code example above and included as used in **Figure 4**.

Figure 4. Hard Reset Configuration Word Example

	Bit 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	EARB	EXMC	IRQ7 INT	EBM	BPS		SCDIS	ISPS	IRPC		DPPC		NMI OUT	ISB		
Value	0	0	0	0	1	1	0	0	0	0	1	1	0	0	1	0
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
	—	BBD	MMR		—		TCPC		BC1PC		—	DLLDIS	MODCK_H			—
Value	0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	0

Bits discussed in this application note are shown in **Table 3**.

Table 3. Selected MSC8101HRCW Bit Descriptions

Name	Value	Description
ISB 13–15	2	**Initial Internal Space Base Select** Places the base address of the internal memory space at address 0xFF000000. This address must be compatible with the memory mapping of the board used.
MODCK_H 28–30	5	**MODCK High-Order Bits** The PLL multiplication factors are determined by six mode clock bits. Three of these bits are dedicated external signals (MODCK[1–3]). The other three bits are in MODCK_H which is part of the HRCW. On the MSC8101ADS board, MODCK[1–3] are tied to three switches (DS9), which are negative logic. If the switches are set to 010 (on-off-on), the MODCK[1–3] bits have a value of 010. Concatenating MODCK_H with MODCK[1–3] gives a value of 101010. Refer to the *MSC8101ADS User's Manual* (shipped with the kit) for detailed information about the on-board switch functions. Refer to the *MSC8101 Technical Data* sheet for detailed clocking information.

3 Clock Configuration

In the MSC8101, the clocking options are determined by a total of six bits:

- MODCK[1–3] pins sampled on rising edge of $\overline{\text{PORESET}}$ internal (see **Figure 1**)
- MODCK_H in the HRCW (bits 28–30)

These bits set up the main PLL multiplication factor and the core, CPM, and 60x system bus frequencies. Notice that although MODCLK_H is read in with the HRCW, its value is influential only during the power-on reset sequence. If the system enters a hard reset sequence later, MODCK_H is read as "don't care". To reset MODCK_H, a power-on reset sequence must occur.

4 Internal Memory Map

The MSC8101 contains three address spaces: SC140 core, QBus, and the 60x system bus Internal Memory Map (IMM). The SC140 core address space is fixed. The base addresses for the QBus and 60x system bus are programmable through the QBUSBR/QBUSMR and IMMR registers, respectively.

As the name implies, the IMM address is an internal memory map address for the 60x system bus that is internal to the MSC8101. The IMM spans 128 KB. The IMM includes the CPM dual-port RAM address space. Since the IMMR resides inside the IMM, writing to the IMMR relocates the IMMR to a new address. The IMMR register setting determines the base address for all devices residing on the internal 60x system bus, including the dual-port RAM, DMA, CPM, memory controller, SIU, and the interrupt controller.

The Internal Space Base (ISB) bits (HCRW[13–15]) define the initial address for the base address of the IMM space. These three bits map into one of eight base addresses (refer to **Table 1**) that is written to bits 0–14 (also called ISB bits) in the IMMR register. The IMMR[ISB] bits can be updated to form a new IMM space base address.

5 Troubleshooting

This section lists hardware and software troubleshooting hints for starting and running an application on an MSC8101-based board. If the hardware is functional and reliable, start with the software and tool sections. If the software is reliable, start with the hardware and tools sections. It is best to begin with as many known working components as possible, then change one item at a time until the system is working reliably.

5.1 Hardware

Table 4 lists some of the most common hardware problems.

Table 4. Hardware Troubleshooting

Check	Comment
Power supply	Ensure that the power supply and/or regulators on the target board provides sufficient current to drive the MSC8101 and the target board.
MSC8101 voltage	Verify the voltage level of the MSC8101. This is critical for proper operation.
MSC8101 temperature	Power consumption varies with voltage and frequency. Precaution needs to be taken where required.
MSC8101 is held in reset until voltage reaches 2/3 V_{DD}	Assert $\overline{\text{PORESET}}$ externally for at least 16 clock cycles after external power reaches at least 2/3 V_{DD}. A test is to manually hold $\overline{\text{PORESET}}$ in reset during power up, then release it. If the problem goes away, $\overline{\text{PORESET}}$ was not held long enough during power up.
MSC8101 clock	Check the system clock frequency and clock circuit. If a crystal is used, ensure that the crystal meets all of the MSC8101 loading and drive requirements. A crystal is preferable over a crystal oscillator.
Address and data lines	Ensure that the order of the low-to-high bits is correct. This may require use of a logic analyzer. On power up, $\overline{\text{CS0}}$ should be asserted to read in the HRCW. If the defaults can be taken, hold RSTCONF high when $\overline{\text{PORESET}}$ deasserts during the power-on reset sequence.
Check the host computer parallel port interface configuration	For example, interfacing an IBM Thinkpad 600 to an MSC8101ADS board requires the following steps: **1.** Configure the parallel port as LPT2, even though the laptop only has one parallel port. **2.** Configure the parallel port in bidirectional mode. This is different from ECP or EPP mode.

5.2 Software

Table 5 lists the most common software problems.

Table 5. Software Troubleshooting

Check	Comment
Memory map	Verify that the linker file memory map matches the memory on the target board. The link map sets the location of RAM and stack.
Unknown exceptions	An unknown interrupt may occur. If the debugger cannot isolate which interrupt, place known code at each exception vector. If you are using a debugger, set a breakpoint in the start-up routine, and then run from the starting address of the start-up routine. This bypasses the reset exception. If the debugger stops at the breakpoint, then there is a problem in the exceptions.
Turn optimization off	Turn off optimization until the MSC8101 application is working reliably.
Program logic is not correct	Ensure that the memory map is correct (both linker file memory map and physical memory on the board). Another common problem is that the IMMR is not set up correctly. Most debuggers allow you to set up an internal memory map (IMM) address. Then the application typically points a structure to this same address. For example, if the debugger sets up the IMM to location 0x14700000, the application may have a variable called IMM that is a pointer to this address or a pointer to a structure at this address: `IMM = (t_SC1IMM *) (0x14700000);`

Note: Call your Freescale sales office or representative to determine an appropriate software environment for system development using the MSC8101.

5.3 Tools

Table 6 lists the most common tools problems.

Table 6. Tools Troubleshooting

Check	Comment
Memory test	Try reading and writing to RAM to determine whether you have communication with the board and whether RAM is being selected.
Interface from host to target board	Ensure that the host can communicate with the target system.
Debugger configuration set up properly	There are many registers to check. Start with the HRCW and IMMR.
If you are using a third-party board, does the debugger company have any experience with your board or something similar?	The first place to look is the manufacturer web site. If the information is not there, contact the manufacturer.
Application runs but data is not correct.	The MSC8101 only supports big-endian mode. Make sure that the compiler is set up to use big-endian mode.

NOTES:

NOTES:

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 München, 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

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

Document Order No.: AN1852
Rev. 3
11/2005