UG0926
User Guide PolarFire SoC FPGA
MSS Simulation

NOTE: PDF files are intended to be viewed on the printed page; links and cross-references in this PDF file may point to external files and generate an error when clicked. **View the online help included with software to enable all linked content.**

Microsemi
a Microchip company

Revision History

Revision 1.0

First publication of this document.

Table of Contents

1 – Introduction

The PolarFire SoC FPGA's Microcontroller Subsystem (MSS) is modeled with Microsemi's AMBA Bus Functional Model (BFM) and it is limited at FIC itself. For information about the supported instructions and syntax of the BFM commands, refer to the Microsemi DirectCore AMBA BFM User's Guide.

Simulation can be useful in the following situation.

- Verifying the connectivity and addressing of peripherals in the Fabric that are connected to the MSS using the Fabric Interface Controllers (FICs).
- Generation of M2F and F2M interrupt.

2 – Creating a Project

Use the MSS stand-alone Configuration tool and Libero SoC to create MSS-based designs. Some steps of design creation are listed here; however, we recommend you refer to the MSS Standalone Configurator User Guide for more information.

1. Create the MSS configurator using the pfsoc_mss application by either creating a new configuration (.cfg) file or opening an existing one.
2. Configure the MSS subsystem with the required FIC interface and other necessary modules.
3. Generate the MSS component file(.cxz).

After you finish with the MSS stand-alone configuration, import the MSS subsystem in Libero, and then design the entire system, as follows:

1. Open the Libero SoC tool.
2. Create the project.
3. Invoke system builder to create your MSS block.
4. Import the MSS component file.
5. Design your entire system using MSS, AXI4 interconnect, fabric slaves, and fabric masters.

A typical system should resemble the ones in the following figures.

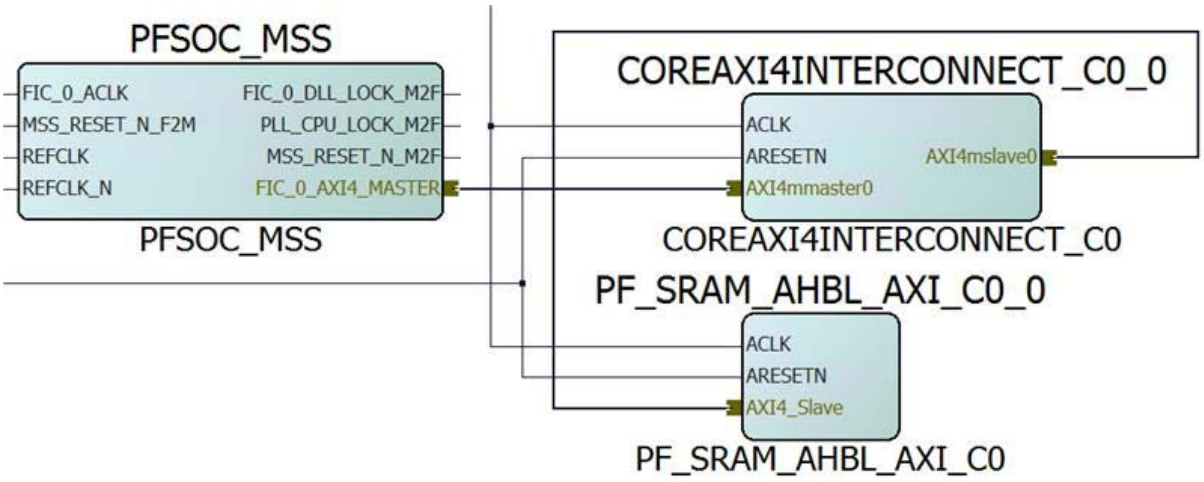

Figure 1 · Example 1: Typical Smart Design Block

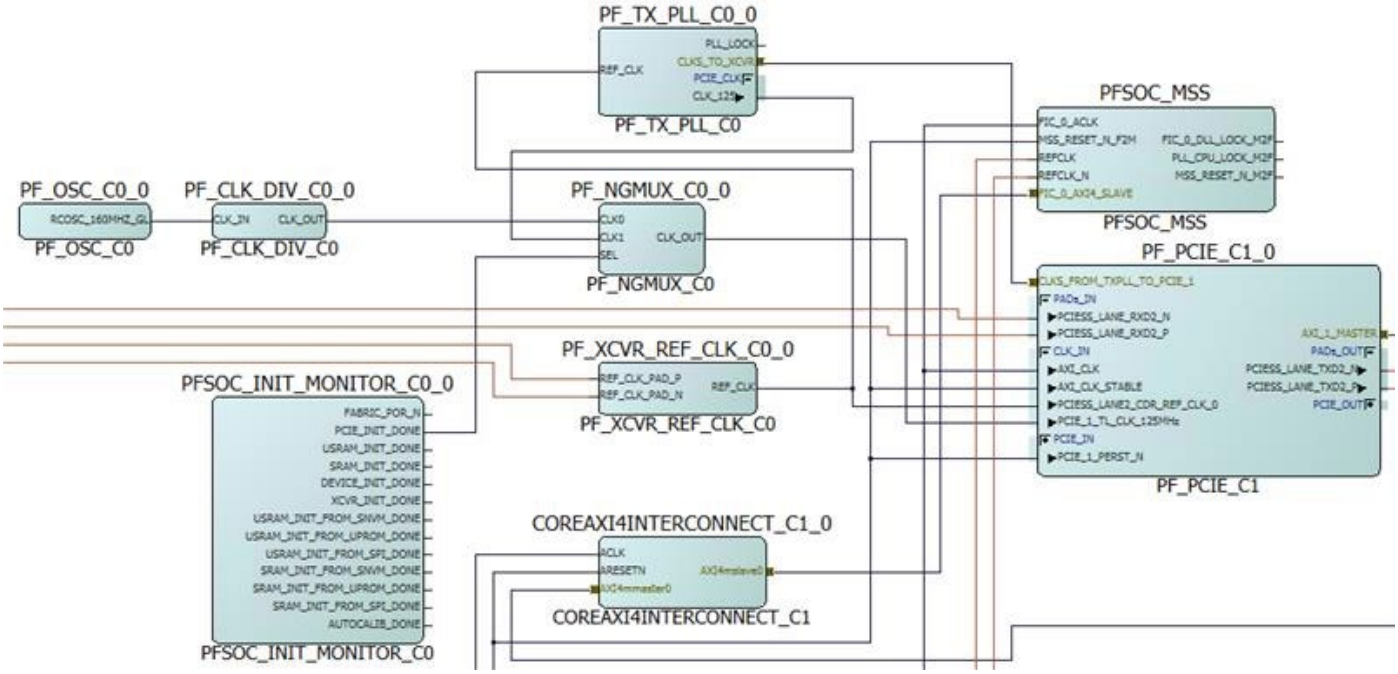

Figure 2 · Example 2: Typical Smart Design Block

6. After designing the entire system, check the DRC and generate the system.

Libero generates three BFM files for three master FIC interfaces in a simulation folder (see Figure 3).

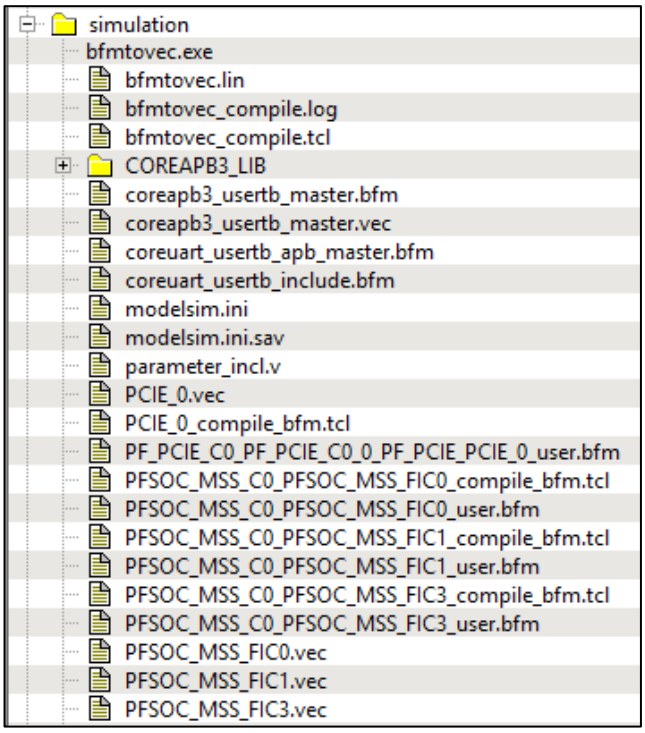

Figure 3 · BFM Files in the Simulation Folder

7. Add the supported BFM instructions in these BFM files to perform the simulation.

3 – Simulation Flow

FIC interface

PolarFire SoC FPGA provides multiple Fabric Interface Controllers (FIC) to enable connectivity between user logic in the FPGA fabric and the Microprocessor Subsystem (MSS). FIC is part of the MSS, and acts as a bridge between MSS and the fabric.

BFM Commands

All BFM commands used in PolarFire PCI BFM and SmartFusion2 FPGA High Speed Serial Interface Simulation can be used to simulate the MSS.

The following shows a typical BFM example.

```
procedure main;
memmap FPR_BASE_ADDR 0x2060000000;
int u;
int l;
write64        w FPR_BASE_ADDR 0x0   0x10102020 0xaaaaaaaa
read64         w FPR_BASE_ADDR 0x0
readcheck64    w FPR_BASE_ADDR 0x0   0x10102020 0xaaaaaaaa
readstore64 x FPR_BASE_ADDR 0x0 u l
print "Lower 32bits = %h", l
print "Upper 32bits = %h", u
readmask64     x FPR_BASE_ADDR 0x0   0x10102020 0xaaaaaaaa 0xFFFFFFFF 0x0000FFFF
readstore64 x FPR_BASE_ADDR 0x0 u l
print "Lower 32bits = %h", l
print "Upper 32bits = %h", u

writemult64    w FPR_BASE_ADDR 0x0  0xFFFFFFFF 0xEEEEEEEE 0xAAAAAAAA 0xBBBBBBBB 0xCCCCCCCC
0xDDDDDDDD 0x01010101 0x02020202 0x03030303  0xBADCAD00
readmult64      w FPR_BASE_ADDR 0x0  5
readmultchk64      w FPR_BASE_ADDR 0x0  0xFFFFFFFF 0xEEEEEEEE 0xAAAAAAAA 0xBBBBBBBB
0xCCCCCCCC 0xDDDDDDDD 0x01010101 0x02020202 0x03030303  0xBADCAD00
return
```

After adding a testbench to your design, you can perform MSS simulation with these BFM files by launching the Per-Synth simulation (see Figure 4).

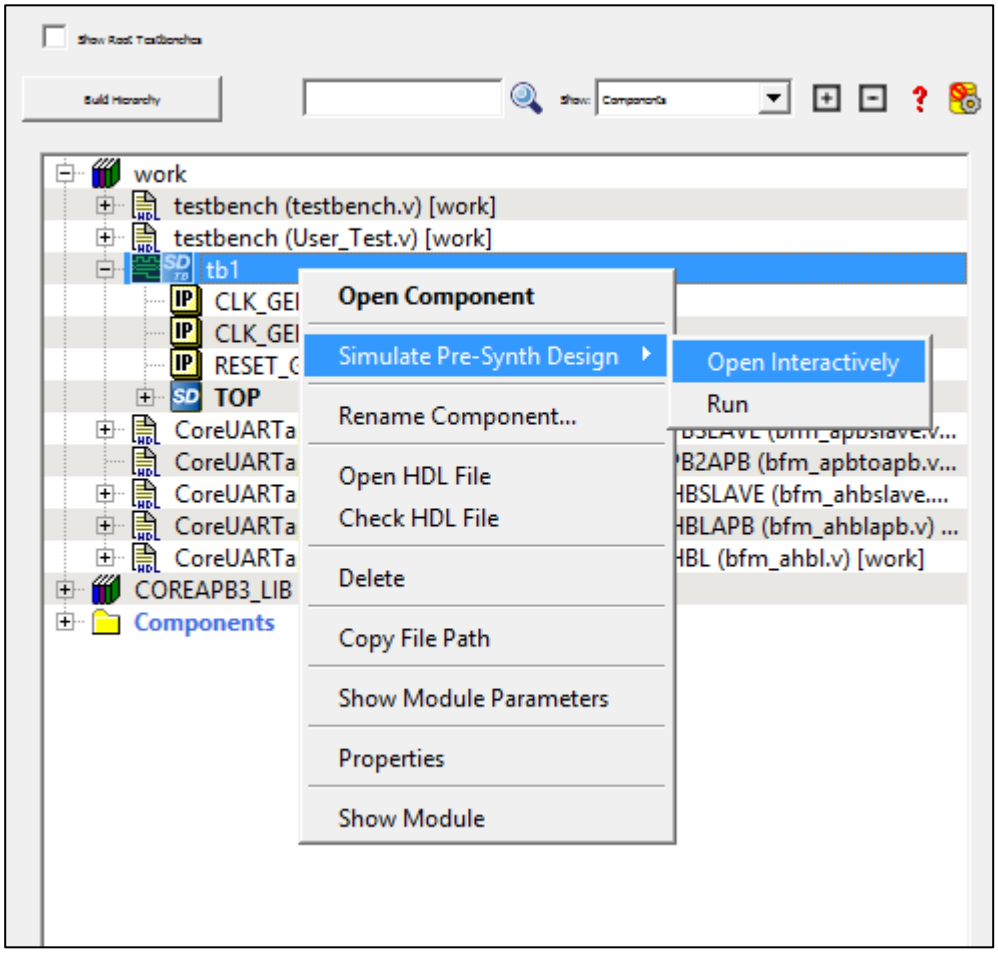

Figure 4 · Simulation Launching

The simulation transcript window displays the BFM transactions, as shown in the following figures.

```
# PFSOC_FIC_0_BFM: Data Write 20600f0010 cccccccddddddd
# PFSOC_FIC_0_BFM:25:readmult64 x 00000020 600f0000 10 at 568 ns
# PFSOC_FIC_0_BFM: Data Write 20600f0018 0101010102020202
# PFSOC_FIC_0_BFM: Data Write 20600f0020 03030303badcad00
# PFSOC_FIC_0_BFM: Data Read 20600f0000 ffffffffeeeeeee at 895.179000ns
# PFSOC_FIC_0_BFM: Data Read 20600f0008 aaaaaaaabbbbbbbb at 905.181000ns
# PFSOC_FIC_0_BFM: Data Read 20600f0010 ccccccccdddddddd at 915.183000ns
# PFSOC_FIC_0_BFM:26:readmultchk64 x 00000020  600f0000 ffffffff ... at 919 ns
# PFSOC_FIC_0_BFM: Data Read 20600f0018 0101010102020202 at 925.185000ns
# PFSOC_FIC_0_BFM: Data Read 20600f0020 03030303badcad00 at 935.187000ns
VSIM 2> run
# PFSOC_FIC_0_BFM: Data Read 20600f0000 ffffffffeeeeeee MASK:ffffffffffffffff at 1085.217000ns
# PFSOC_FIC_0_BFM: Data Read 20600f0008 aaaaaaaabbbbbbbb MASK:ffffffffffffffff at 1095.219000ns
# PFSOC_FIC_0_BFM: Data Read 20600f0010 ccccccccdddddddd MASK:ffffffffffffffff at 1105.221000ns
# PFSOC_FIC_0_BFM:31:return
# PFSOC_FIC_0_BFM: Data Read 20600f0018 0101010102020202 MASK:ffffffffffffffff at 1115.223000ns
# PFSOC_FIC_0_BFM: Data Read 20600f0020 03030303badcad00 MASK:ffffffffffffffff at 1125.225000ns
################################################################################
#
# FIC_0 BFM Simulation Complete - 7 Instructions - NO ERRORS
#
################################################################################
```

Figure 5 · MSS FIC Simulation Log – FIC_0 as Master

```
# BFM:26:readmult64 x 600f0000 10 at 1185 ns
# BFM: Data Write 600f0018 0101010102020202
# FIC_0: 64 Bit Write 600f0000=ffffffffeeeeeee at 1235 ns
# FIC_0: 64 Bit Write 600f0008=aaaaaaaabbbbbbbb at 1245 ns
# FIC_0: 64 Bit Write 600f0010=ccccccccdddddddd at 1255 ns
# FIC_0: 64 Bit Write 600f0018=0101010102020202 at 1265 ns
# FIC_0: 64 Bit Write 600f0020=03030303badcad00 at 1275 ns
# BFM: Data Write 600f0020 03030303badcad00
# FIC_0: 64 Bit Read 600f0000=ffffffffeeeeeee at 1605 ns
# FIC_0: 64 Bit Read 600f0008=aaaaaaaabbbbbbbb at 1615 ns
# FIC_0: 64 Bit Read 600f0010=ccccccccdddddddd at 1625 ns
# FIC_0: 64 Bit Read 600f0018=0101010102020202 at 1635 ns
# FIC_0: 64 Bit Read 600f0020=03030303badcad00 at 1645 ns
# BFM: Data Read 600f0000 ffffffffeeeeeee at 1865.000000ns
# BFM: Data Read 600f0008 aaaaaaaabbbbbbbb at 1875.000000ns
# BFM: Data Read 600f0010 ccccccccdddddddd at 1885.000000ns
# BFM:27:readmultchk64 x 600f0000 ffffffff ... at 1895 ns
# BFM: Data Read 600f0018 0101010102020202 at 1895.000000ns
# BFM: Data Read 600f0020 03030303badcad00 at 1905.000000ns
# FIC_0: 64 Bit Read 600f0000=ffffffffeeeeeee at 1995 ns
# FIC_0: 64 Bit Read 600f0008=aaaaaaaabbbbbbbb at 2005 ns
# FIC_0: 64 Bit Read 600f0010=ccccccccdddddddd at 2015 ns
# FIC_0: 64 Bit Read 600f0018=0101010102020202 at 2025 ns
# FIC_0: 64 Bit Read 600f0020=03030303badcad00 at 2035 ns
# BFM: Data Read 600f0000 ffffffffeeeeeee MASK:ffffffffffffffff at 2255.000000ns
# BFM: Data Read 600f0008 aaaaaaaabbbbbbbb MASK:ffffffffffffffff at 2265.000000ns
# BFM: Data Read 600f0010 ccccccccdddddddd MASK:ffffffffffffffff at 2275.000000ns
# BFM:33:return
# BFM: Data Read 600f0018 0101010102020202 MASK:ffffffffffffffff at 2285.000000ns
# BFM: Data Read 600f0020 03030303badcad00 MASK:ffffffffffffffff at 2295.000000ns
################################################################################
#
# PCIE1 BFM Simulation Complete - 7 Instructions - NO ERRORS
#
################################################################################
```

Figure 6 · MSS FIC Simulation Log – FIC_0 as Slave

Interrupts

F2H Interrupts

The MSS simulation model acknowledges the assertion of F2H interrupts. (Fabric to MSS)

There are 64 F2H interrupt ports. MSS acknowledges them by printing a message on reception of valid active high interrupt (see Figure 7).

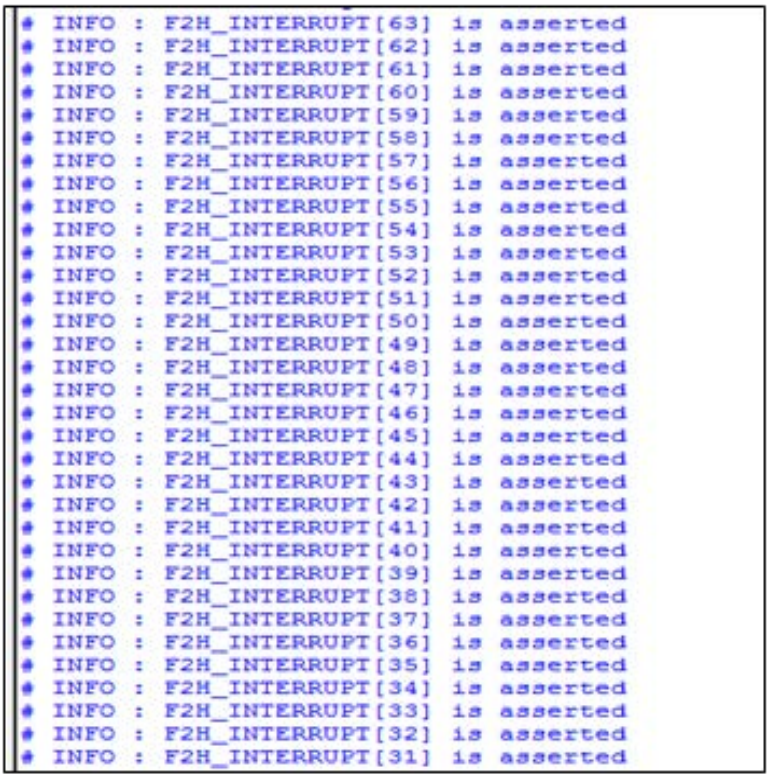

Figure 7 · F2H Interrupt Simulation Log (Valid Interrupts)

The interrupt inputs should be high for one clock of MSS clock; otherwise, the MSS model rejects the interrupt and prints a message about the interrupt level being too low (see Figure 8).

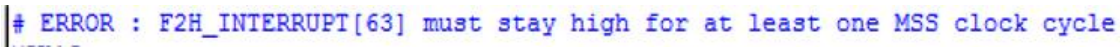

Figure 8 · F2H Interrupt Simulation Log (Invalid Interrupts)

H2F Interrupts

The MSS simulation model allows you to use text files to set and clear H2F interrupts (MSS to Fabric). To do this, add the following command in the run.do file, as follows:

vsim -L polarfire -L presynth -t 1ps -g H2F_MEMFILE=(path)/*.txt presynth.tb

Example: vsim -L polarfire -L presynth -t 1ps -g H2F_MEMFILE=E:/mss_sim/h2f_sim.txt presynth.tb

There are 16 H2F interrupts. Table 1 shows their allocation in MSS.

Table 1 · H2F Interrupt Allocation

H2F Line	Group
0	GPIO
1	MMUART,SPI,CAN
2	I2C
3	MAC0
4	MAC1
5	WATCHDOGS
6	Maintenance
7	SCB
8	G5C-Message
9	DDRC
10	G5C-DEVRST
11	RTC/USOC
12	TIMER
13	ENVM, QSPI
14	USB
15	MMC/SDIO

You use entries in the text file to set and clear an interrupt, as shown below.

Wait time	(Time to wait in number MSS PLL clock cycles, Hex)
Interrupt Value	(16-bit value, Hex)
Wait time	(Time to wait in number MSS PLL clock cycles, Hex)
Interrupt Value	(16-bit value, Hex)
….	

Example:

100	(Wait for 100 (256 in DEC) MSS PLL clock cycles)
FFFF	(Set all 16 interrupts)
1000	(Wait for 1000 (4096 in DEC) MSS clock cycles)
0000	(Clear all 16 interrupts)
…	

You can also clear interrupts by clearing an interrupt register bit in corresponding peripheral. These AXI transactions can be generated by a Master in fabric.

Table 2 · H2F Vnterrupt Clearing Addresses

H2F Line	Group	AXI Address and Data Bits to Clear an Interrupt	
0	GPIO	Reg	**g5soc_mss_regmap:GPIO:INTR**
		Physical address	0x2012 0080 0x2012 1080 0x2012 2080 0x2812 0080 0x2812 1080 0x2812 2080
		Data	Bit-0: To clear an interrupt the bit is written with '1'
1	MMUART	Reg	**g5soc_mss_regmap:MMUART:RTO**
		Physical address	0x2000 004C 0x2010 004C 0x2010 204C 0x2010 404C 0x2010 604C 0x2800 004C 0x2810 004C 0x2810 204C 0x2810 404C 0x2810 604C
		Data	Writing the RTO register clears this interrupt.
	SPI	Reg	**g5soc_mss_regmap:SPI:INT_CLEAR**
		Physical address	0x2010 800C 0x2010 900C 0x2810 800C 0x2810 900C
		Data	Bit-5: Write one to clear the interrupt Bit-4: Write one to clear the interrupt
	CAN		Support will be added in a future Libero release
2	I2C		Support will be added in a future Libero release
3	MAC0		Support will be added in a future Libero release
4	MAC1		Support will be added in a future Libero release
5	WATCHDOGS		Support will be added in a future Libero release

H2F Line	Group	AXI Address and Data Bits to Clear an Interrupt	
6	Maintenance		Support will be added in a future Libero release
7	SCB		Support will be added in a future Libero release
8	G5C-Message		Support will be added in a future Libero release
9	DDRC		Support will be added in a future Libero release
10	G5C-DEVRST		Support will be added in a future Libero release
11	RTC/USOC		Support will be added in a future Libero release
12	TIMER		Support will be added in a future Libero release
13	ENVM, QSPI		Support will be added in a future Libero release
14	USB		Support will be added in a future Libero release
15	MMC/SDIO		Support will be added in a future Libero release

4 – Limitations

- The FIC AXI BFM slave does not support 38-bit addressing. When addressed with a 38-bit address from the fabric master, the AXI transaction is not considered and a message appears in the console.
- Bits 2 to 15 of the H2F interrupts cannot be cleared through FIC AXI.

5 – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**
From the rest of the world, call **650.318.4460**
Fax, from anywhere in the world, **650.318.8044**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

For Microsemi SoC Products Support, visit http://www.microsemi.com/products/fpga-soc/design-support/ fpga-soc-support.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page at https://www.microsemi.com/product-directory/design-resources/1750-libero-soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions through our Web portal and receive answers back by email, fax, or phone. Also, if you have design problems, you can upload your design files to receive assistance. We constantly monitor the cases created from the web portal throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

Technical support can be reached at https://soc.microsemi.com/Portal/Default.aspx.

5-02-00591-8/05.22

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support at https://soc.microsemi.com/Portal/Default.aspx or contact a local sales office.

Visit About Us for sales office listings and corporate contacts.

Sales office listings can be found at https://www.microsemi.com/locations

Microsemi Headquarters One Enterprise, Aliso Viejo, CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: **sales.support@microsemi.com www.microsemi.com**

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at **www.microsemi.com.**

5-02-00926-0/7.20