CoreAXItoAHBL v2.2

Handbook

Microsemi

Table of Contents

Introduction

Core Overview

The CoreAXItoAHBL IP core is an advanced extensible interface (AXI) slave and an advanced high-performance bus lite (AHB-Lite) master. This provides an interface (bridge) between the AXI domain and advanced high-performance bus (AHB) domain. CoreAXItoAHBL allows an AXI bus system to be connected to an AHBLite bus.

The CoreAXItoAHBL bridge enables the AXI master to communicate with the AHB slave. The AXI slave interface and AHB-Lite master interface allow the AXI bus and AHB bus to access the read/write buffer memory.

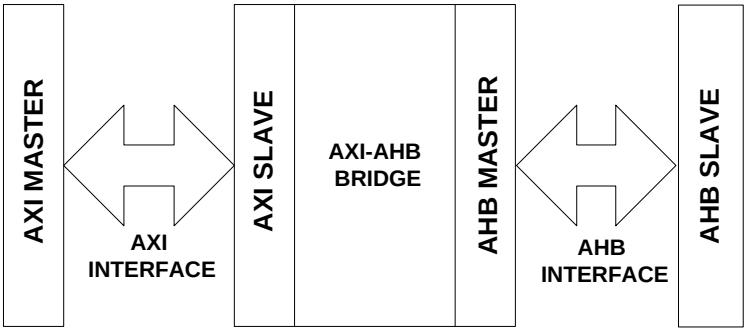

Figure 1 CoreAXItoAHBL Bridge Block Diagram

Core Version

This handbook applies to CoreAXItoAHBL version 2.2.

Supported Families

- SmartFusion®2
- IGLOO®2

Key Features

CoreAXItoAHBL is a highly configurable core and has the following features:

- Provides an interface (bridge) between the advanced extensible interface (AXI) domain and advanced high-performance bus (AHB) domain
- Makes alternate AXI write transaction and AXI read transactions possible
- Supports AXI data bus width of 64-bits, maximum burst size upto 8 bytes and maximum number of beats/transfer of 16
- Supports AHB data bus width of 32-bits
- Provides ERROR/OKAY response for every AXI master transaction
- Provides output signal to indicate whether the incoming AXI read/write address is aligned or unaligned address

Utilization and Performance

Utilization and performance data is listed in Table 1 for the SmartFusion2 (M2S150S) device family. The data listed in this table is indicative only. The overall device utilization and performance of the core is system dependent.

Table 1 Device Utilization and Performance

Family	AXI CLK to AHB CLK sync/async	AXI_AWIDTH	AXI_DWIDTH	AHB_AWIDTH	AHB_DWIDTH	Logic Elements				Frequency (MHz)
						Sequential	Combinatorial	Total	%	
SmartFusion2	0	32	64	32	32	1,804	2,095	3,899	3.4	155
SmartFusion2	1	32	64	32	32	1,804	2,095	3,899	3.4	155

Note: The data in this table is achieved using typical synthesis and layout settings. Frequency (in MHz) was set to 100 and speed grade was -1.

Functional Block Description

The CoreAXItoAHBL Bridge appears as a slave on the AXI and is a master on the AHBL. Read and write transactions on the AXI are converted into corresponding transfers on the AHBL.

The HCLK and ACLK clocks are configurable to synchronous or asynchronous depending on the parameter/generic. This block performs the cross-clock-domain control logic, when the AHB clock and AXI clock are asynchronous.

CoreAXItoAHBL consists of the following four major functional blocks:

- write memory buffer
- read memory buffer
- write/read AXI slave controller
- write/read AHB-Lite master controller

A basic block diagram of the design for CoreAXItoAHBL is shown in Figure 2.

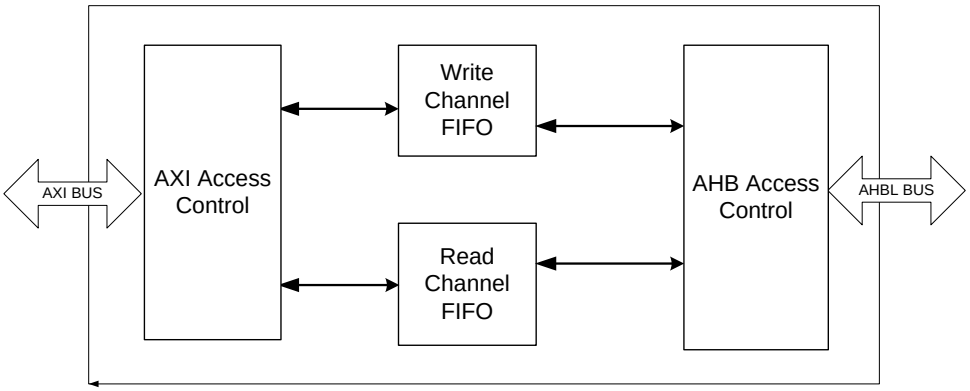

Figure 2 CoreAXItoAHBL Block Diagram

AXI Access Control

The AXI Access Control block is the AXI slave interface of the bridge. This block contains write/read finite state machine (FSM) which converts the AXI write/read transactions. The AXI master drives write transaction when the AWREADY is asserted. When AWVALID/ARVALID and AWREADY/ARVALID are asserted, the AXI master triggers the AXI Access Control FSM. During AXI write transaction, the FSM asserts write enable to the Write Channel FIFO. The AXI master also drives the address, burst length, and size control information to the AHB Access Control block.

This block depending on the AXI bursts, length, and size calculates the corresponding AHB transactions of the bursts, size, and length on the AHB side. It also provides the write strobe information to the AHB Access Control block. The Write Response from the AHB master is latched and sent to the AXI master.

It also generates the read enable to the Read Channel FIFO. The read data received from the Read Channel FIFO is driven to the AXI master through the AXI Access Control.

Write Channel FIFO Control

The Write Channel FIFO Control block is 16 deep x 72 wide asynchronous FIFO. It stores the write data from the AXI Access Control and the write strobe data in the Write Channel FIFO. The AHB Access Control block generates the read enable to the FIFO and reads the data when the write to the Write Channel FIFO has completed.

Read Channel FIFO Control

The Read Channel FIFO Control block is 16 deep x 64 wide asynchronous FIFO. It stores the write data from the AHB Access Control. The AXI Access Control block generates the read enable to the FIFO and reads the data.

AHB Access Control

The AHB Access Control block is the AHB master interface of the bridge. This block generates the AHB write/read transactions on the AHB bus when the HREADY is asserted. The AHB address is incremented based on the size and burst calculated.

The AHB Access Control block controls the read enable to the Write Channel FIFO and the write enable to the Read Channel FIFO. For write transactions on the AHB Slave, the write enable is asserted to the Write Channel FIFO. Similarly, for the read transactions the read enable is asserted to the Read Channel FIFO. The data reads from the AHB slave through the AHB master interface of the bridge is stored in the Read Channel FIFO.

Clock Domains

The CoreAXItoAHBL bridge consists of the following two clock domains:

- The AXI clock
- The AHB clock

The AXI Access Control block works on the AXI clock and the AHB Access Control block works on the AHB clock. The synchronization is achieved through the use of the following asynchronous FIFO's:

- Write Channel FIFO
- Read Channel FIFO.

AXI-AHB Interface Support

AHB Address (HADDR) Generation

Since the AXI master issues only the start address for read/write transactions, HADDR is required to be generated for the subsequent read/write beats of the burst transfer. When a valid read/write request is issued by the AXI interface, the start address of the transfer is registered. For subsequent beats, the address (HADDR) is generated depending on the type (ARBURST/AWBURST), size (ARSIZE/AWSIZE), and length (ARLEN/AWLEN) of burst type.

AXI Burst Size: Translation of AXI Interface → AHB Interface

The AXI-AHB bridge supports maximum 8 bytes (64-bit) data width size. The AXI master should generate only 64-bit transactions (ARSIZE and AWSIZE is 3'011) and use WSTRB signal if the size of the transfer is smaller than 8 bytes.

Table 2 AXI Burst Size and Corresponding AHB Burst Size

AXI-Master ARSIZE / AWSIZE	(AXI) Bytes in Transfer	AHB –Master HSIZE[1:0] (depends on WSTB's)	Size of AHB Transfer
000	1	00	Byte
001	2	00	Byte
		01	Halfword
010	4	00	Byte
		01	Halfword
		10	Word
011	8	00	Byte
		01	Halfword
		10	Word

AXI Burst Length: Translation of AXI Interface → AHB Interface

The AXI-AHB bridge supports maximum 16 numbers of beats/transfers in a burst. The AXI transaction is translated to multiple single undefined length burst AHBL transactions. Number of single undefined length burst transfers on AHBL side depends on the length of the AXI transactions. Table 3 describes the burst length translation from AXI to AHB.

Table 3 AXI Burst Length and Corresponding AHB Burst Length

AXI-Master ARLEN / AWLEN	No of Data Transfer	AHB –Master HBURST[2:0]	Beats in AHB Transfer
0000	1	SINGLE	SINGLE, NON_SEQ transfer
Other length beat bursts	2 to 16	INCR	undefined length burst on AHB interface

AXI Write Strobe: Translation of AXI Interface → AHB Interface

The AXI-AHB bridge expects the AXI master to drive valid byte lanes corresponding to the AXI write address as shown in Table 4.

Table 4 AXI Write Strobes and Corresponding AHB Write Transactions

AXI-Master WRITE transaction WSTRB [7:0] – on AXI Address 0x00000000	AHBL Transaction ADDRESS	AHBL Transaction SIZE
11111111	0x0000_0000	4-byte (32-bit)
	0x0000_0004	4-byte (32-bit)
00001111	0x0000_0000	4-byte(32-bit)

AHBL Slave Size (32-Bit)

The AXI-to-AHBL bridge supports only 32-bit AHBL slaves. When a read/write transaction request of size 32-bit or less is generated on the AXI master, the same is translated into AHBL read/write transaction request for 32-bit AHBL slave.

But when read/write transaction request of size 64-bit is generated on the AXI-master interface, the read/write transaction is split into at least two transactions of size 32-bit on AHBL slave side.

READ / WRITE Response

- Write (OKAY/ERROR) response to AXI Master Write transaction request is given after the last beat write of the write request.
- Read (OKAY/ERROR) response to AXI Master Read transaction request is given for every beat of the read transaction.

Unaligned Address Output Generation

- The incoming AXI AWADDR[2:0] and ARADDR[2:0] address signal must be aligned with the transfer size boundaries given by AWSIZE[2:0] and ARSIZE[2:0] respectively.
- For example: (Double-Word transfer size) must be aligned with (Double-Word address boundaries).

That means, If AWADDR[2:0] == 3’b000, then AWSIZE must be equal to 3’b011 for the address to be Double-word aligned, else it is unaligned.

Similarly, if AWADDR[2:0] == 3’b100, then AWSIZE must be equal to 3’b010 for the address to be Word aligned, else it is unaligned.

The UNALIGNED_ADDR_OUT output is asserted when the address is unaligned, else it remains de-asserted.

Licensing

CoreAXItoAHBL requires a register transfer level (RTL) license to be used and instantiated.

RTL

Complete RTL source code is provided for the core and testbenches.

SmartDesign

CoreAXItoAHBL is preinstalled in the SmartDesign IP Deployment design environment. An example instantiated view is shown in Figure 3. The core can be configured using the configuration GUI within SmartDesign, as shown in Figure 4.

For information on using SmartDesign to instantiate and generate cores, refer to the *Using DirectCore in Libero® SoC User Guide* or consult the *Libero SoC online help*.

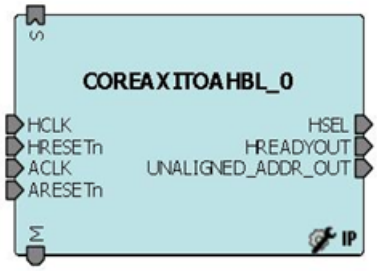

Figure 3 SmartDesign CoreAXItoAHBL Instance View

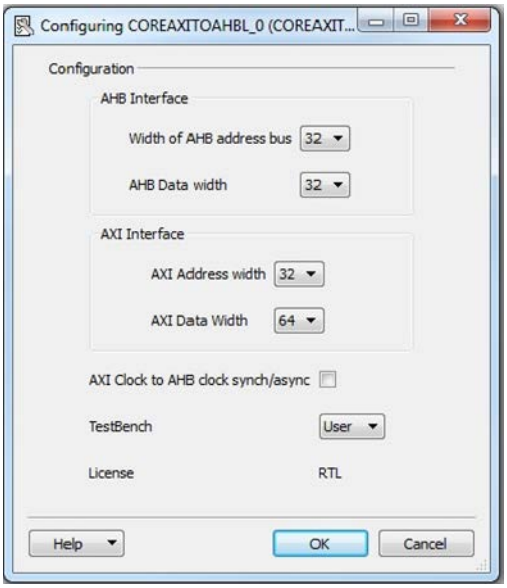

Figure 4 SmartDesign CoreAXItoAHBL Configuration Dialog Box

Simulation Flows

The User Testbench for CoreAXItoAHBL is included in all releases.

To run simulations, select the User Testbench flow within the SmartDesign and click **Save and Generate on the Generate pane**. The User Testbench is selected through the Core Testbench Configuration GUI.

When the SmartDesign generates the Libero SoC project, it installs the User Testbench files.

To run the User Testbench, set the design root to the CoreAXItoAHBL instantiation in the Libero SoC design hierarchy pane and click **Simulation** in the Libero SoC design flow window. This invokes ModelSim® and automatically runs the simulation.

User Testbench

An example of User Testbench is included with CoreAXItoAHBL.

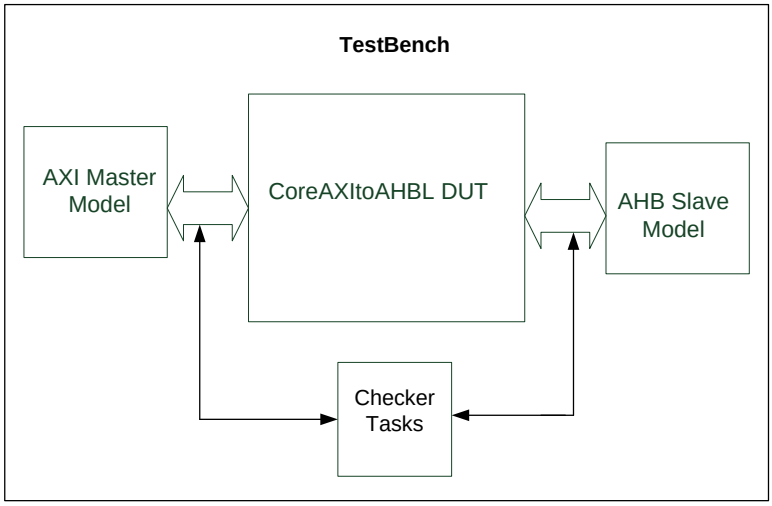

Figure 5 User Testbench

As shown in Figure 5, the User Testbench instantiates a Microsemi® DirectCore CoreAXItoAHBL design under test (DUT). The CoreAXItoAHBL Testbench environment consists of the following components:

- **The AXI master model:** The AXI master model drives the write/read AXI transactions to the DUT. These transactions are converted by the DUT in corresponding AHB write/read transactions.
- **The AHB slave model:** In case of AXI write transactions, the AHB slave model accepts the write transactions from the DUT and stores the write data into corresponding address location of internal memory. In case of AXI reads, the AHB slave model responds back with read data from the addressed location.
- **The checker:** The checker model checks and determines whether or not the transaction is successful and displays the result.

Synthesis in Libero SoC

Click **Synthesis** in Libero SoC. The synthesis window appears and displays the Synplicity® project. Set **Synplicity** to use the Verilog 2001 standard, if the verilog is being used. To run synthesis, select **Run**.

Place-and-Route in Libero SoC

Click **Layout** in the Libero SoC to invoke **Designer**. CoreAXItoAHBL does not require any special place-and-route settings.

Signal Descriptions for CoreAXItoAHBL

Signal descriptions for CoreAXItoAHBL are defined in Table 5.

Table 5 CoreAXItoAHBL I/O Signals

Port Name	Width	Direction	Description
AHBL Slave Interface Ports			
HCLK	1	In	AHB clock. All the AHB signals inside the block are clocked on the rising edge.
HRESETn	1	In	AHB Reset. The signal is active low. Asynchronous assertion and synchronous de-assertion. This is used to reset AHB registers in the block.
HSEL	1	Out	AHBL slave select –Indicates that the current transfer is intended for the selected slave
HADDR	AHB_AWIDTH	Out	AHBL address – 32 bit address on the AHBL interface
HWRITE	1	Out	AHBL write – When high, indicates that the current transaction is a write. When low, indicates that the current transaction is a read.
HREADYOUT	1	Out	When HIGH, the HREADYOUT signal indicates that a transfer has finished on the bus. This signal can be driven LOW to extend a transfer.
HTRANS	2	Out	AHBL transfer type – Indicates the transfer type of the current transaction b00: IDLE b01: BUSY b10: NONSEQUENTIAL b11: SEQUENTIAL
HSIZE	3	Out	AHBL transfer size – Indicates the size of the current transfer (8/16/32/64 bit transactions only) bx00: 8-bit (byte) transaction bx01: 16-bit (half word) transaction bx10: 32-bit (word) transaction bx11: 64-bit(double word) transaction
HWDATA	AHB_DWIDTH	Out	AHBL write data – Write data from the AHBL master to the AHBL slave
HBURST	3	Out	Size and type of burst from the AHB master
HREADYIN	1	In	AHBL ready input – Used to monitor the AHBL bus ready

Port Name	Width	Direction	Description
HRESP	1	In	AHBL response status – When driven high at the end of a transaction indicates that the transaction is completed with errors. When driven low at the end of a transaction indicates that the transaction is completed successfully.
HRDATA	AHB_DWIDTH	In	AHBL read data – Read data from the AHBL slave to the AHBL master
HMASTLOCK	1	Out	Indicates that the current transfer is a part of locked sequence. 0: unlock transfer 1: lock transfer (Not supported for bridge)
AXI Master Interface Ports			
Global Signal Ports (Clocks)			
ACLK	1	In	AXI clock – All the AXI signals inside the block are clocked on the rising edge.
ARESETn	1	In	AXI reset signal – The signal is active low. The signal is asynchronously asserted and synchronously de-asserted.. This is used to reset all the AXI registers in the Block
AXI Write ADDRESS CHANNEL			
AWID	4	In	Write Address ID – This Signal is the Identification tag for the write address group of signals.
AWADDR	AXI_AWIDTH	In	Write address – Gives the address of the first transfer in a write burst The associated control signals are used to determine the addresses of the remaining transfers in the burst.
AWLEN	4	In	Burst length – Gives the exact number of transfers in a burst This information determines the number of data transfers associated with the address.
AWSIZE	3	In	Burst size – Indicates the size of each transfer in the burst
AWBURST	2	In	Burst type – Coupled with the size information, details how the address for each transfer within the burst is calculated
AWLOCK	2	In	Lock type – Provides additional information about the atomic characteristics of the transfer (Not supported for bridge)
AWVALID	1	In	Write address valid – Indicates that valid write address and control information are available: 1: address and control available 0: address and control not available
AWREADY	1	Out	Write address ready – Indicates that the slave is ready to accept an address and associated control signals: 1: slave ready 0: slave not ready

Port Name	Width	Direction	Description
AXI Write DATA CHANNEL			
WID	4	In	Write Data ID tag – The Identification tag for the write data transfer The WID must match the AWID value of the write transaction.
WDATA	AXI_DWIDTH	In	Write data bus is 64 bits wide.
WSTRB	8	In	Write strobes. – Indicates which byte lanes to update in the memory There is one write strobe for each 8 bits of the write data bus. WSTRB[n] corresponds to WDATA [(8 × n) + 7 :(8 × n)].
WLAST	1	In	Write last – Indicates the last transfer in a write burst
WVALID	1	In	Write valid – Indicates that valid write data and strobes are available: 1: write data and strobes available 0: write data and strobes unavailable.
WREADY	1	Out	Write ready – Indicates that the slave can accept the write data: 1: slave ready 0: slave not ready
AXI Write RESPONSE CHANNEL			
BREADY	1	In	Response ready – Indicates that the master can accept the response information 1: master ready 0: master not ready
BID	4	Out	Response ID – The Identification tag for the write response The BID must match the AWID value of the write transaction to which the slave is responding.
BRESP	2	Out	Write response – Indicates the status of the write transaction The allowable responses are 00: OKAY 01: EXOKAY 10: SLVERR DECERR is not supported.
BVALID	1	Out	Write response valid – Indicates that a valid write response is available: 1: write response available. 0: write response not available.
AXI Read ADDRESS CHANNEL			
ARID	4	In	Read Address ID – The identification tag for the read address group of signals
ARADDR	AXI_AWIDTH	In	Read address – Gives the initial address of a read burst transaction Only the start address of the burst is provided.
ARLEN	4	In	Burst length – Gives the exact number of transfers in a burst This information determines the number of data transfers associated with the address.

Port Name	Width	Direction	Description
ARSIZE	3	In	Burst size – Indicates the size of each transfer in the burst
ARBURST	2	In	Burst type – Coupled with the size information, details how the address for each transfer within the burst is calculated
ARLOCK	2	In	Lock Type – Provides additional information about the atomic characteristics of the transfer (Not supported for bridge)
ARVALID	1	In	Read address valid – Indicates, when HIGH, that the read address and control information is valid 1: address and control valid 0: address and control not valid
ARREADY	1	Out	Read address ready – Indicates that the slave is ready to accept an address and associated control signals: 1: slave ready 0: slave not ready
AXI Read RESPONSE CHANNEL			
RREADY	1	In	Read ready – Indicates that the master can accept the read data and response information: 1: master ready 0 : master not ready
RID	4	Out	Read ID Tag – The ID tag of the read data group of signals The RID value is generated by the slave and must match the ARID value of the read transaction to which it is responding.
RDATA	AXI_DWIDTH	Out	Read data – Read data bus is 64 bits wide
RRESP	2	Out	Read Response – Indicates the status of the read transfer. The allowable responses are 00: OKAY 01: EXOKAY 10: SLVERR DECERR is not supported
RLAST	1	Out	Read Last – Indicates the last transfer in a read burst
RVALID	1	Out	Read Valid – Indicates that the required read data is available and the read transfer can complete 1: read data available 0: read data not available
Other Signals			
UNALIGNED_ADDR_OUT	1	Out	Unaligned Address Out – This signal indicates whether the incoming AXI address is aligned or unaligned. 1: unaligned address 0: aligned address

Core Parameters

CoreAXItoAHBL Configurable Options

There are a number of configurable options which are applied to CoreAXItoAHBL (as shown in Table 6). If a configuration other than the default is required, the configuration dialog box in the SmartDesign should be used to select appropriate values for the configurable options.

Table 6 CoreAXItoAHBL Configuration Options

Name	Valid Range	Description
FAMILY	19,24	Must be set to the required FPGA family: 19: SmartFusion2 24: IGLOO2
AHB_AWIDTH	32	A 32-bit System AHB address bus.
AHB_DWIDTH	32	Write/Read data bus on AHB side.
AXI_AWIDTH	32	A 32-bit System AXI address bus.
AXI_DWIDTH	64	Write/Read data bus on AXI side.
CLOCKS_ASYNC	0 or 1	Set to 1 when the AXI clock is asynchronous to the AHB clock. In general, this should be set whenever the two clocks are not on the same global network.

Register Map and Descriptions

CoreAXItoAHBL does not contain any registers.

Ordering Information

Ordering Codes

CoreAXItoAHBL can be ordered through your local Sales Representative. It should be ordered using the following number scheme: CoreAXItoAHBL-XX, where XX is listed in Table 7.

Table 7 Ordering Codes

XX	Description
RM	RTL for RTL source — multiple-use license

List of Changes

The following table lists critical changes that were made in each revision of the document.

Date	Change	Page
September 2014	Resolved SARs mentioned in Release Notes.	4
	Updated Utilization and Performance section.	
June 2014	Added a device family under Supported Families section.	3
	Added Utilization and Performance section.	4
February 2013	Initial v2.0 Handbook release.	N/A

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

Visit the Microsemi SoC Products Group Customer Support website for more information and support (http://www.microsemi.com/soc/support/search/default.aspx). Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on website.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/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 to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account 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.

The technical support email address is soc_tech@microsemi.com.

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 via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1(949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense and security, aerospace, 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; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.

50200397-2/09.14