CoreGPIO v3.1

Handbook

Table of Contents

Introduction

Core Overview

Core GPIO provides an Advanced Peripheral Bus (APB) register-based interface to up to 32 general purpose inputs and 32 general purpose outputs. The input logic contains a simple three-stage synchronization circuit, and the output is also set synchronously. Each bit can be set to either fixed configuration or register-based configuration via top-level parameters, including input type, interrupt type / enable, and output enable.

Key Features

CoreGPIO v3.1 has the following key features:

- AMBA 2 APB support, forward compatibility with AMBA 3 APB
- 8-, 16-, or 32-bit APB data width
- 1 to 32 bits of I/O, for all APB-width configurations
- Fixed or configurable interrupt generation
  - Negative edge
  - Positive edge
  - Both edges
  - Level High
  - Level Low
- Parameter-configurable for single-interrupt signal or up to 32-bit-wide interrupt bus
- Fixed or configurable I/O type (input, output, or both)
- Configurable output enable (internal or external implementation)

Supported Families

Currently all devices in the following families will be supported:

- IGLOO®
- IGLOOe
- IGLOO nano
- IGLOO PLUS
- ProASIC®3
- ProASIC3 nano
- ProASIC3L
- Fusion
- ProASIC PLUS®
- Axcelerator®
- RTAX-S
- SmartFusion®
- SmartFusion®2
- IGLOO®2
- RTG4™

Core Version

This handbook applies to CoreGPIO v3.1.

Supported Interfaces

CoreGPIO is available with the APB slave interface and must be connected to an APB master interface. Microsemi recommends that you use SmartDesign in the Libero® Integrated Design Environment (IDE) or Libero System-on-Chip (SoC) Project Manager to instantiate, configure, connect and generate CoreGPIO in a processor-based system, using ARM® Cortex®-M1, Core8051s, or CoreABC.

Device Utilization and Performance

A summary of utilization and performance data is shown in Table 1 and Table 2.

Table 1 CoreGPIO Utilization and Performance Data (minimum configuration)

Family	Tiles			Utilization		Performance (MHz)
	Sequential	Combinatorial	Total	Device	Total	
IGLOO/e/PLUS	128	332	460	AGL600V5	3%	79
ProASIC3/E/L	128	332	460	A3P600	3%	154
Fusion	128	332	460	AFS600	3%	154
ProASIC^{PLUS}	128	443	571	APA150	9%	78
Axcelerator	128	173	301	AX250	7%	233
RTAX-S	128	173	301	RTAX250S	7%	176
SmartFusion	32	33	65	A2F500M3G	0.56%	250
SmartFusion2	32	13	37	M2S050	0.07%	609
RTG4	32	13	37	RT4G150	0.02%	611

Notes:

1. *Data in this table were achieved using typical synthesis and layout settings.*
2. *Minimum configuration consists of the following parameter values: IO_NUM = 8, APB_WIDTH = 8, OE_TYPE = 0, INT_BUS = 0, FIXED_CONFIG_(0...7) = 1, IO_TYPE_(0...7) = 0, IO_INT_TYPE_(0...7) = 7*

Table 2 CoreGPIO Utilization and Performance Data (maximum configuration)

Family	Tiles			Utilization		Performance
	Sequential	Combinatorial	Total	Device	Total	
IGLOO/e/PLUS	512	1,297	1,809	AGL600V5	13%	79
ProASIC3/E/L	512	1,297	1,809	A3P600	13%	153
Fusion	512	1,297	1,809	AFS600	13%	153
ProASIC^{PLUS}	512	1,743	2,255	APA150	36%	78
Axcelerator	512	662	1,174	AX250	27%	229
RTAX-S	512	662	1,174	RTAX250S	27%	173
SmartFusion	768	1450	2218	A2F500M3G	19.25%	124
SmartFusion2	548	832	863	M2S050	1.53%	234

Family	Tiles			Utilization		Performance
	Sequential	Combinatorial	Total	Device	Total	
RTG4	512	921	1009	RT4G150	0.66%	182

Notes:

1. *Data in this table were achieved using typical synthesis and layout settings.*
2. *Maximum configuration consists of the following parameter values: IO_NUM = 32, APB_WIDTH = 8, OE_TYPE = 1, INT_BUS = 1, FIXED_CONFIG (0...31) = 0.*

Functional Block Diagram

Figure 1 illustrates a single-bit block diagram (this is replicated up to 32 times, depending on the number of I/Os).

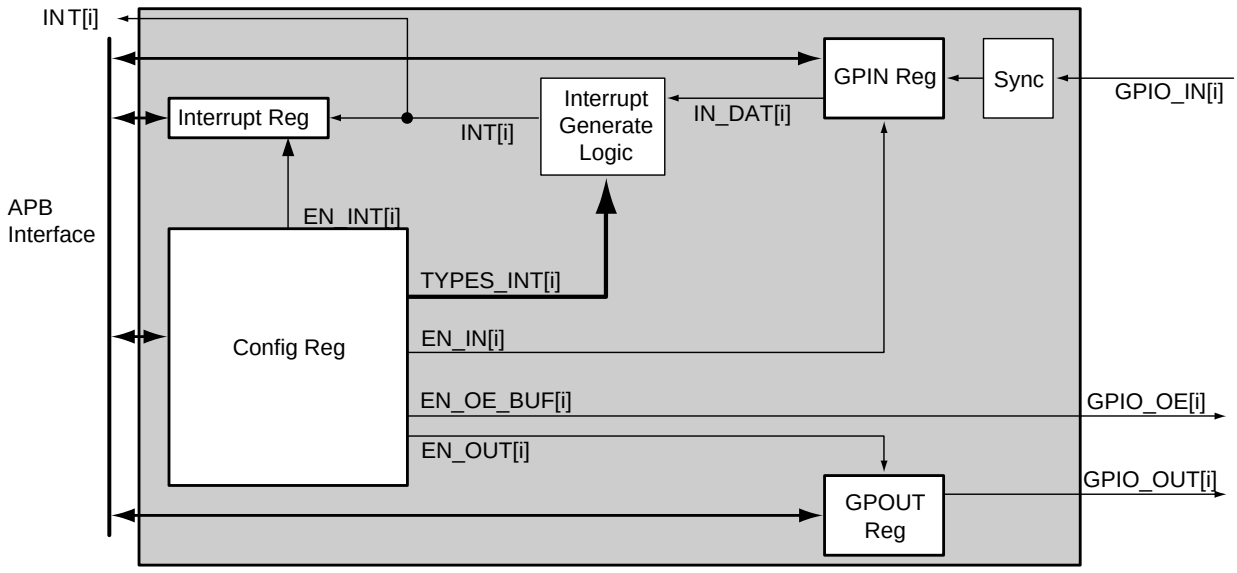

Figure 1 Single I/O Bit Block Diagram for CoreGPIO

1 - Tool Flows

SmartDesign

Configuration

CoreGPIO, available from the Libero web repository, can be seen and downloaded to your local vault via the SmartDesign IP catalog. For information on using SmartDesign to instantiate, configure, connect, and generate cores, refer to the Libero IDE or SoC Online Help.

Figure 2 shows the CoreGPIO configuration window, as well as cross-references to the corresponding top-level parameters.

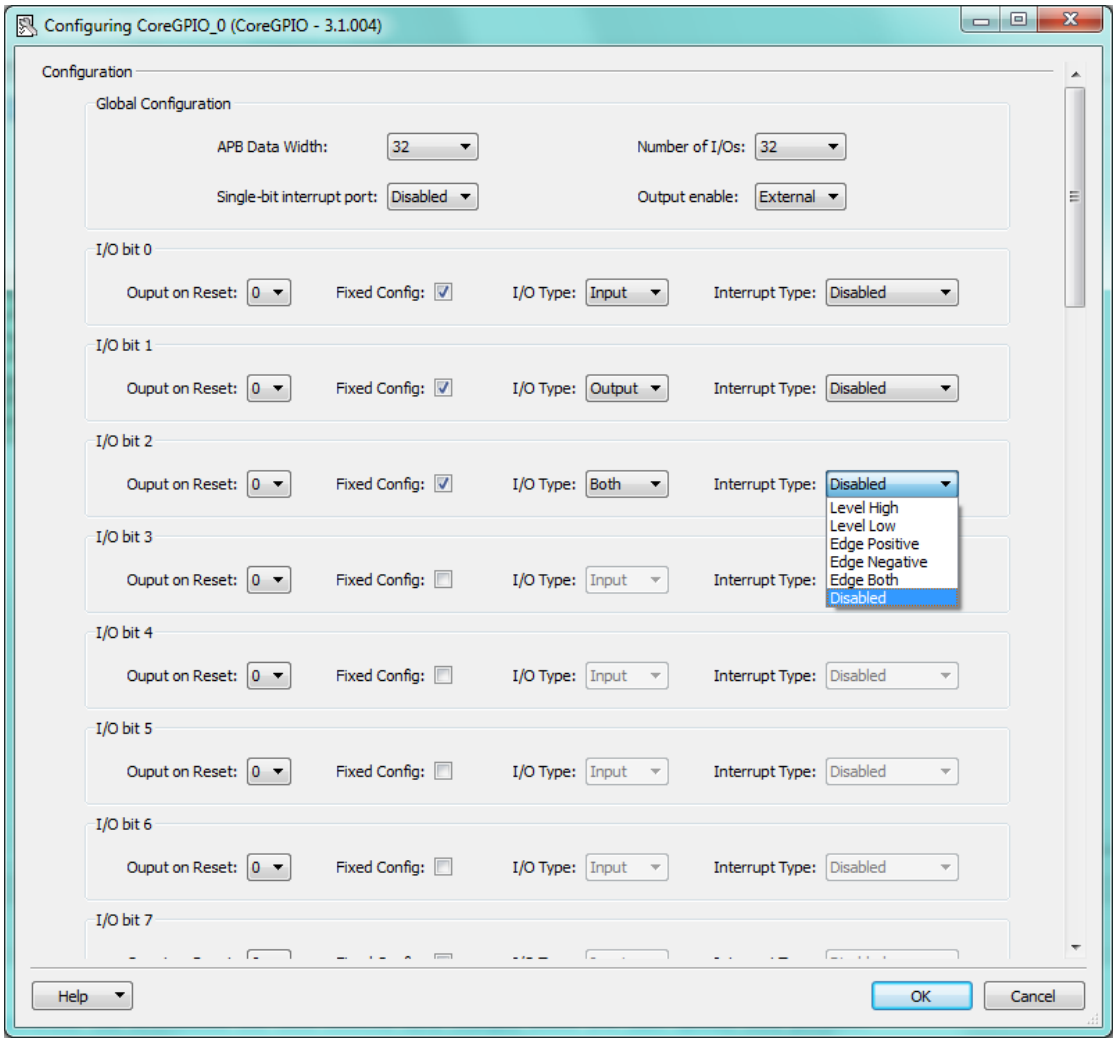

Figure 2 CoreGPIO Configuration Window

Simulation Flows

To run simulations, select the user testbench in SmartDesign through the CoreGPIO IP configuration GUI. Generate the design in SmartDesign. The appropriate testbench files are now installed.

To run the testbenches, set the design root to the CoreGPIO instance in the Libero IDE or SoC Design Explorer and click the Simulation icon in the Project Flow tab. This invokes ModelSim® and automatically runs the simulation.

Synthesis in Libero IDE or SoC

To run Synthesis on the core with parameters set in SmartDesign, set the design root to the SmartDesign design (wrapper) and click the **Synthesis** icon in the Project Manager. The Synthesis window appears, displaying the Synplicity® project. To perform synthesis, click the **Run** icon.

Place-and-Route in Libero IDE or Libero SoC

After setting the design root appropriately and running Synthesis, click the **Layout** icon in the Project Manager to invoke Designer. CoreGPIO requires no special place-and-route settings.

2 - Interface Description

Generics

Table 3 gives descriptions for the CoreGPIO generics.

Table 3 CoreGPIO Generics

Parameter	Values	Default Value	Description
FAMILY	0–99	17	Will be set automatically to the device family selected in Libero IDE or SoC. 11 – Axcelerator 12 – RTAX-S 14 – ProASIC 15 – ProASIC3 16 – ProASIC3E 17 – Fusion 20 – IGLOO 21 – IGLOOe 22 – ProASIC3L 23 – IGLOOPLUS 18 – SmartFusion 19 – SmartFusion2 24 – IGLOO2 25 – RTG4
APB_WIDTH	8, 16, 32	32	APB data width
IO_NUM	1–32	32	Number of GPIOs
OE_TYPE	0 or 1	0	If 0, output buffering is implemented outside CoreGPIO. The user is responsible for instantiating tristate buffers outside of the core. If 1, output buffering (if enabled) is implemented inside the core. When GPIO_OE[i] is 0, GPIO_OUT is high impedance (Z).
FIXED_CONFIG_x	0 or 1	0	If 0, configuration for bit x (0-31) is set via APB-accessible register CONFIG_x (see the "Register Map" section). If 1, configuration for bit x (0-31) is set via "IO_INT_TYPE_x" (described below) and "IO_TYPE_x".
IO_INT_TYPE_x	0-5	0	Interrupt types selected according to the following scheme: 0 – Level High 1 – Level Low 2 – Edge Positive 3 – Edge Negative 4 – Edge Both 7 – Disabled Note that selecting one type will synthesize out logic for other types. For example, Level High will remove and/or gates for edge detect.

Parameter	Values	Default Value	Description
IO_TYPE_x	0-2	0	If 0, bit x is of type input only. Output logic will be synthesized out. If 1, bit x is of type output only. Input logic will be synthesized out. If 2, bit x is of type input and output (both).
IO_VAL_x	0 or 1	0	Sets the output at reset for GPIO bit x.
INT_BUS	0 or 1	0	If 0, the INT_OR output is fixed at 0 (unused). If 1, the INT_OR output is set if any of the INT signals are set (OR operation).

Ports

Table 4 outlines the top-level signals for CoreGPIO.

Table 4 CoreGPIO Ports

Name	Type	Description
APB Bus Signals		
PCLK	Input	APB System Clock – Reference clock for all internal logic
PRESETN	Input	APB active low asynchronous reset
PWDATA [APB_WIDTH-1:0]	Input	APB write data
PRDATA [APB_WIDTH-1:0]	Output	APB read data
PADDR[7:0]	Input	APB address bus
PENABLE	Input	APB strobe – Indicates the second cycle of an APB transfer
PSEL	Input	APB slave select
PWRITE	Input	APB write/read select signal
PREADY	Output	APB 3 ready signal for future APB 3 compliance; tied internally High
PSLVERR	Output	APB 3 transfer error signal for future APB 3 compliance; tied internally Low
GPIO Signals		
GPIO_IN [IO_NUM-1:0]	Input	GPIO input
GPIO_OUT[IO_NUM-1:0]	Output	GPIO output
GPIO_OE[IO_NUM-1:0]	Output	GPIO output enable
INT[IO_NUM-1:0]	Output	Interrupt mask; can be connected directly to processor (for example, Cortex-M1)
INT_OR	Output	Provides an OR'ed version (single wire) of the interrupt mask provided on INT[IO_NUM-1:0]

Notes:
1. *Unless otherwise noted, all of the signals above are active High.*

3 - Register Map

Overview

Table 5 through Table 7 describe the CoreGPIO Register map

Table 5 CoreGPIO Register Address Map (APB_WIDTH = 8)

PADDR[7:0]	Type	Reset Value (Hex)	Brief Description
0x00-0x7C (0x00, 0x04, 0x08, …, 0x7C)	R/W	0x00	8-bit configuration registers for all 32 bits; 1 register per bit.
0x80	W	0x00	Interrupt clear register 1 (bits 7:0)
0x84	W	0x00	Interrupt clear register 2 (bits 15:8)
0x88	W	0x00	Interrupt clear register 3 (bits 23:16)
0x8C	W	0x00	Interrupt clear register 4 (bits 31:24)
0x90	R	0x00	Input register 1 (bits 7:0)
0x94	R	0x00	Input register 2 (bits 15:8)
0x98	R	0x00	Input register 3 (bits 23:16)
0x9C	R	0x00	Input register 4 (bits 31:24)
0xA0	R/W	0x00	Output register 1 (bits 7:0)
0xA4	R/W	0x00	Output register 2 (bits 15:8)
0xA8	R/W	0x00	Output register 3 (bits 23:16)
0xAC	R/W	0x00	Output register 4 (bits 31:24)

Notes:

1. *Values shown in hexadecimal format; type designations: R = read only; R/W = read/write.*
2. *Lower 2 bits of PADDR are unconnected inside CoreGPIO.*

Table 6 CoreGPIO Register Address Map (APB_WIDTH = 16)

PADDR[7:0]	Type	Reset Value (Hex)	Brief Description
0x00-0x7C (0x00, 0x04, 0x08, …, 0x7C)	R/W	0x00	8-bit configuration registers for all 32 bits; 1 register per bit.
0x80	W	0x00	Interrupt clear register 1 (bits 15:0)
0x84	W	0x00	Interrupt clear register 2 (1bits 31:16)
0x90	R	0x00	Input register 1 (bits 15:0)
0x94	R	0x00	Input register 2 (bits 31:16)
0xA0	R/W	0x00	Output register 1 (bits 15:0)

PADDR[7:0]	Type	Reset Value (Hex)	Brief Description
Notes: *1. Values shown in hexadecimal format; type designations: R = read only; R/W = read/write.* *2. Lower 2 bits of PADDR are unconnected inside CoreGPIO.*			

Table 7 CoreGPIO Register Address Map (APB_WIDTH = 32)

PADDR[7:0]	Type	Reset Value (Hex)	Brief Description
0x00-0x7C			
(0x00, 0x04, 0x08, …, 0x7C)	R/W	0x00	8-bit configuration registers for all 32 bits; 1 register per bit.
0x80	W	0x00	Interrupt clear register 1 (bits 31:0)
0x90	R	0x00	Input register 1 (bits 31:0)
0xA0	R/W	0x00	Output register 1 (bits 31:0)
Notes: *1. Values shown in hexadecimal format; type designations: R = read only; R/W = read/write.* *2. Lower 2 bits of PADDR are unconnected inside CoreGPIO.*			

Configuration Registers

There are up to 32 8-bit configuration registers (depending on the IO_NUM parameter). Table 8 describes the CoreGPIO configuration register operation.

Table 8 Per-bit Configuration Register

Bits	Name	Function
7:5	INTTYPE	Sets the interrupt type for this particular bit: 000 – Level High 001 – Level Low 010 – Edge Positive 011 – Edge Negative 100 – Edge Both 101 to 111 – Invalid
4	Reserved	Unused
3	INTENABLE	Interrupt enable for this particular bit 1 – Enable interrupt generation 0 – Disable interrupt generation
2	OUTBUFF	Sets the output enable for this particular bit, whether via the GPIO_OE signal or implemented internally (see parameter "OE_TYPE"). 1 – Enables output 0 – Disables output
1	INREG	Input register enable 1 – Enables input register for this particular bit 0 – Disables input register for this particular bit
0	OUTREG	Output register enable

Bits	Name	Function
		1 – Enables output functionality for this particular bit 0 – Disables output functionality for this particular bit

Interrupt Registers

These are per-bit interrupt clear registers. Writing a 1 to any bit clears the interrupt bit register of the corresponding GPIO bit.

In 32-bit mode, all 32 interrupt bits are in a single 32-bit register located at address 0x80.

In 16-bit mode, 32 interrupt bits are split into two 16-bit registers located at addresses 0x80 and 0x84.

In 8-bit mode, 32 interrupt bits are split into four 8-bit registers located at addresses 0x80, 0x84, 0x88, and 0x8C.

Input Registers

Read-only for input configured ports. Disabling a bit in this register with the CONFIG_X[1] (INREG) bit will force the bit to 0 via a MUX, while keeping the incoming current value in the register.

In 32-bit mode, all 32 input bits are in a single 32-bit register located at address 0x90.

In 16-bit mode, 32 input bits are split into two 16-bit registers located at addresses 0x90 and 0x94.

In 8-bit mode, 32 input bits are split into four 8-bit registers located at addresses 0x90, 0x94, 0x98, and 0x9C.

Output Registers

The output registers are writeable/readable for output configured ports, and are logical "don't cares" for input configured ports. Disabling a bit in this register with the CONFIG_X[0] (OUTREG) bit will force the bit to 0 via a MUX, while keeping the previously written value in the output register.

In 32-bit mode, all 32 output bits are in a single 32-bit register located at address 0xA0.

In 16-bit mode, 32 output bits are split into two 16-bit registers located at addresses 0xA0 and 0xA4.

In 8-bit mode, 32 output bits are split into four 8-bit registers located at addresses 0xA0, 0xA4, 0xA8, and 0xAC.

4 - Testbench Operation and Modification

An example user testbench is included with CoreGPIO for both VHDL and Verilog. The testbench is provided as an obfuscated bus functional model (BFM), connected as shown in Figure 3 to a CoreGPIO block. You can examine and change the testbench by modifying the *.bfm file and generating a *.vec APB master vector file, as shown in Figure 3.

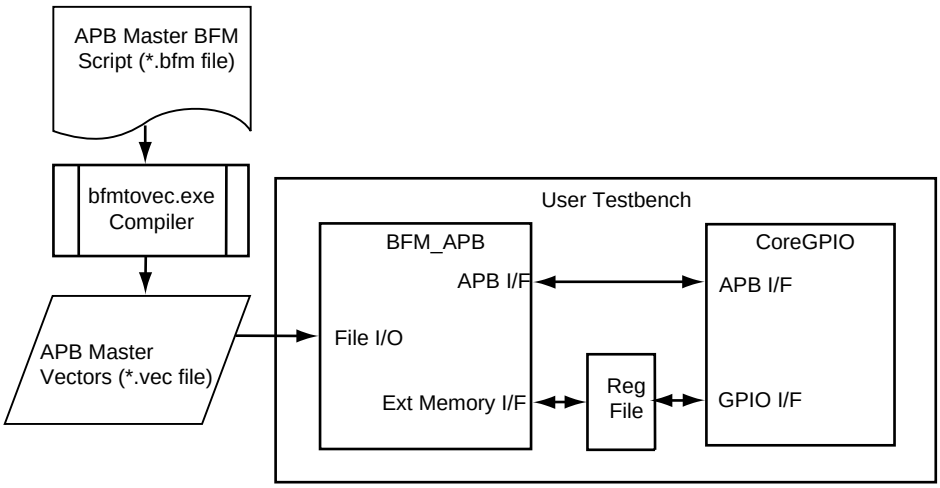

Figure 3 CoreGPIO User Testbench Block Diagram

The user testbench instantiates a Microsemi DirectCore AMBA BFM module to emulate an APB master that controls the operation of CoreGPIO via reads and writes to access internal registers. A BFM ASCII script source file with comments is included in the directory *<proj>/simulation*, where <proj> represents the path to your Libero IDE or SoC project.

The BFM source file, coregpio_usertb_apb_master.bfm, controls the APB master processor. This BFM source file is automatically recompiled each time the simulation is invoked from Libero IDE or SoC by the bfmtovec.exe executable, if running on a Windows® platform, or by the bfmtovec.lin executable, if running on a Linux platform. The coregpio_usertb_apb_master.vec vector file, created by the bfmtovec executable, is read in by the BFM module for simulation in ModelSim.

You can alter the BFM script, if desired. Refer to the Microsemi *DirectCore AMBA BFM User Guide* for more information.

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 Technical Support, visit http://www.microsemi.com/design-support/technical-support.

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 Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & 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**.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

50200183-1/03.15