CoreTimer v2.0

Handbook

Microsemi.

Table of Contents

Preface

About this Document

This handbook provides details about the CoreTimer APB slave module that provides access to an interrupt-generating, programmable decrementing counter.

Intended Audience

Designers using Libero® System-on-Chip (SoC) or Libero Integrated Design Environment (IDE)

Introduction

Overview

The CoreTimer module is an APB slave that provides access to an interrupt-generating, programmable decrementing counter. Figure 1 shows the top-level block diagram of CoreTimer.

Key Features

- Configurable 16-bit or 32-bit Timer
- Advanced Peripheral Bus (APB) slave interface for register access
- No additional clock required. Runs off the APB clock.
- Prescale provides Clock Division by up to 1,024
- Continuous or One-Shot operating modes
- Interrupt Generation

Core Version

This handbook supports CoreTimer version 2.0.

Supported Interfaces

CoreTimer is available with the following interfaces:

- APB interface for register access
- Interrupt Request (IRQ) interface

Supported Families

All Microsemi families and devices are supported by CoreTimer.

Device Utilization and Performance

CoreTimer has been implemented in several device families by Microsemi using the standard speed grades. Following is the summary of various implementation data listed in Table 1 and Table 2.

Table 1 CoreTimer Device Utilization and Performance (16-bit Counter)

Family	Tiles			Utilization		Performance MHz
	Sequential	Combinatorial	Total	Device	Total %	
Fusion	70	221	291	AFS600	2.11	110
SmartFusion	70	221	291	A2F500M3G	2.53	121
SmartFusion	70	99	169	M2S050	0.30	177
IGLOO®/e	71	244	315	AGL600V2/AGLE600V2	2.28	56
IGLOO+	71	244	315	AGLP125V2	10.10	87
IGLOO2	70	99	169	M2GL050	0.30	178
ProASIC®3/E	70	196	266	A3P600/A3PE60	1.92	123
ProASIC3L	70	194	264	A3P600L	1.91	88
RTAX-S™	70	112	182	RTAX1000S	1.50	85
Axcelerator®	72	120	192	AX2000	0.89	116
RTG4™	71	108	179	RT4G150	0.12	147

Note: Data in this table were achieved using the Verilog RTL with typical synthesis and layout settings. Top-level parameters/generics were set as: WIDTH =16, INTACTIVEH = 1.

Table 2 CoreTimer Device Utilization and Performance (32-bit Counter)

Family	Tiles			Utilization		Performance MHz
	Sequential	Combinatorial	Total	Device	Total %	
Fusion	118	286	404	AFS600	2.92	101
SmartFusion	118	286	404	A2F500M3G	3.51	114
SmartFusion	118	177	295	M2S050	0.52	120
IGLOO®/e	119	328	447	AGL600V2/AGLE600V2	3.23	47
IGLOO+	119	328	447	AGLP125V2	14.33	75
IGLOO2	118	177	295	M2GL050	0.52	122
ProASIC®3/E	118	269	387	A3P600/A3PE60	2.80	114
ProASIC3L	118	270	388	A3P600L	2.81	74
RTAX-S™	119	187	306	RTAX1000S	2.53	100
Axcelerator®	120	181	301	AX2000	1.40	98
RTG4™	121	133	254	RT4G150	0.17	147

Note: Data in this table were achieved using the Verilog RTL with typical synthesis and layout settings. Top-level parameters/generics were set as: WIDTH =32, INTACTIVEH = 1.

Functional Block Description

The width of the decrementing counter in the CoreTimer module can be statically configured as either 16 or 32 bits. Programmable registers provide a means to dynamically control the operation of the timer. If the interrupt is enabled, an interrupt is generated when the decrementing counter reaches zero.

CoreTimer supports two operating modes as follows:

- Continuous mode
- One-Shot Timer mode

Continuous Mode

This is the default mode. When zero is reached, the counter is reloaded with the start value, which is stored in a programmable register, and continues to count down. If the interrupt is enabled, this mode can be used to generate an interrupt on a constant interval.

One-Shot Timer Mode

The counter decrements from its high value and halts at zero. The timer must be reprogrammed to begin counting down again. This can be achieved by either clearing the Timer Mode bit in the Timer Control Register or writing a new value to the Load Register.

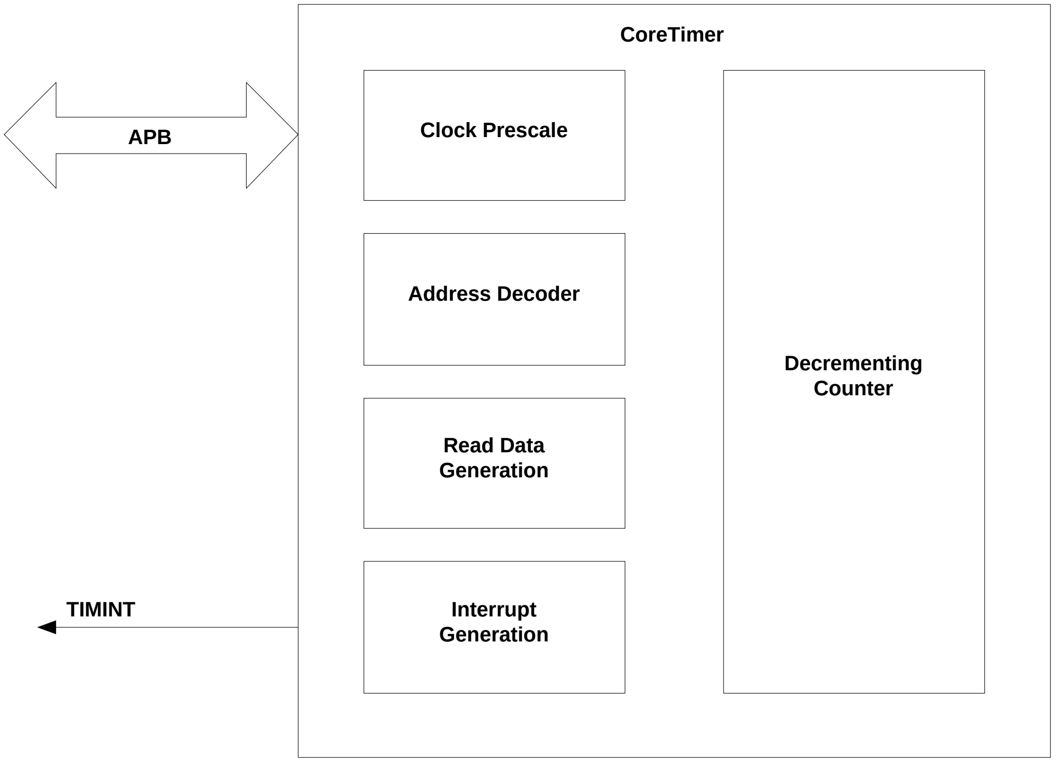


Figure 1 CoreTimer Block Diagram

Operation

Operation Overview

The timer is loaded by writing to the Load Register and then, if enabled, counts down to zero. When the counter is already running, writing to the Load Register will cause the counter to immediately restart at the new value.

When zero is reached, an interrupt is generated if the interrupt is enabled. The interrupt can be cleared by writing to the Interrupt Clear Register. If One-Shot Timer mode is selected, the counter halts on reaching zero until One-Shot Timer mode is deselected or a new load value is written. Otherwise, after reaching zero, the timer reloads the count value from the Load Register and continues to decrement. In Continuous mode, the counter effectively generates a periodic interrupt.

Continuous or One-Shot Timer mode is selected by the Timer Mode bit in the Timer Control Register. At any point, the current counter value can be read from the Current Value Register.

The counter is enabled by a bit in the Timer Control Register. At reset, the counter is disabled, the interrupt is cleared, and the Load Register is set to zero. The mode is set to Continuous, and the prescale value is set to divide the clock by two.

A prescale unit is used to provide a clock enable pulse for the decrementing counter. The prescaler is driven by the APB clock (PCLK) and can be programmed via the Prescale Setting Register to provide an enable pulse every 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1,024 periods of PCLK.

Interrupt Generation and Clearing

An interrupt is generated when the counter reaches zero and is only cleared when the Interrupt Clear Register, TimerIntClr, is written to. A register holds the value until the interrupt is cleared.

The interrupt can be masked by writing 0 to the Interrupt Enable bit in the Timer Control Register. Both the raw interrupt status (prior to masking) and the final interrupt status (after masking) can be read from status registers.

Clocking

The counter in CoreTimer is clocked with PCLK, but a clock enable signal produced by the prescaler is used to enable the counter to operate from a lower effective frequency than that at which PCLK is running.

The interval between clock enable pulses can be adjusted via the prescale field in the Prescale Setting Register. It is possible to generate a clock enable pulse every 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1,024 periods of PCLK.

Interface Description

I/O Signals

The port signals for the CoreTimer macro are showed in Figure 2 and listed in Table 3.

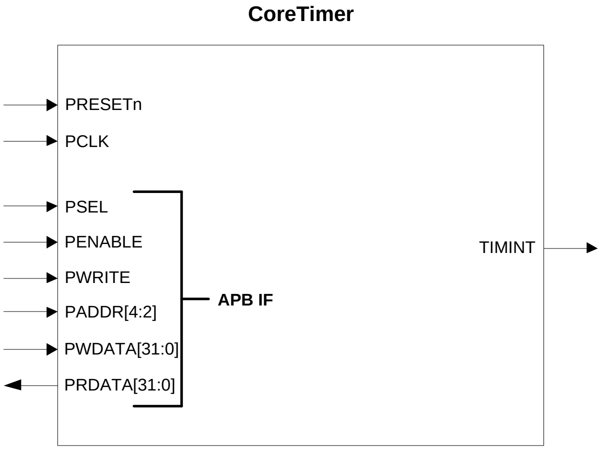


Figure 2 CoreTimer I/O Signal Diagram

Table 3 CoreTimer Connections

Port Name	Type	Description
Required Connections		
APB Interface		
PCLK	Input	APB System Clock; Reference clock for all internal logic
PRESETn	Input	APB active low asynchronous/synchronous reset. (Reset is synchronous when CoreTimer is instantiated on an RTG4 family device. Reset is asynchronous for all other device families.)
PADDR[4:2]	Input	Addresses the CoreTimer APB slave interface registers.
PSEL	Input	APB Slave Select; select signal to registers for APB reads and writes.
PENABLE	Input	APB Strobe. This signal indicates the second cycle of an APB transfer.
PWRITE	Input	APB Write/Read. If high, a write occurs when an APB transfer takes place. If low, a read takes place.
PWDATA[31:0]	Input	APB write data
PRDATA[31:0]	Output	APB read data

Port Name	Type	Description
		Optional Connections
TIMINT	Output	Interrupt output for timer. This signal indicates that an interrupt has been generated by the counter having decremented to zero. The polarity of this signal is configured at the time of instantiation via the INTACTIVEH parameter. This is normally connected to one of the interrupt source inputs of the Interrupt Controller.

Verilog/VHDL Parameters

CoreTimer has parameters (Verilog) or generics (VHDL) for configuring the RTL code as described in Table 4. All the parameters and generics are integer types.

Table 4 CoreTimer Configurable Options

Parameter	Valid Range	Description	Default
WIDTH	16 or 32	Sets the width of the counter.	32
INACTIVEH	0 or 1	TIMINT interrupt level. Active low when INTACTIVEH set to 0. Active high when INTACTIVEH set to 1.	1

Register Map and Descriptions

The CoreTimer registers are listed in Table 5.

Table 5 CoreTimer Registers

Offset	Register Name	Type	Width	Reset Value	Description
0x00	TimerLoad	Read/Write	16 or 32	0x0000 or 0x00000000	Load value
0x04	TimerValue	Read	16 or 32	0xFFFF or 0xFFFFFFFF	Current value
0x08	TimerControl	Read/Write	3	0x0	Control register
0x0C	TimerPrescale	Read/Write	4	0x0	Clock prescale setting
0x10	TimerIntClr	Write	–	–	Interrupt clear
0x14	TimerRIS	Read	1	0x0	Raw interrupt status
0x18	TimerMIS	Read	1	0x0	Masked interrupt status

Load Register – TimerLoad

Table 6 Bit Assignments for the TimerLoad Register

Bit(s)	Name	Type	Description
31:0	Load Value	Read/Write	The TimerLoad register contains the value from which the count is decremented. When the register is written to, the counter is loaded with the value written and begins to decrement, provided that the TimerEnable bit is set in the TimerCtrl register. When CoreTimer is configured to operate in continuous mode (TimerMode bit set to 0 in the TimerCtrl register), the counter will be reloaded with the value contained in this register when the current count reaches zero. The TimerLoad register is either 16 or 32 bits wide depending on the configuration of CoreTimer (Timer width controlled via WIDTH parameter at the time of instantiation).

Current Value Register – TimerValue

Table 7 Bit Assignments for the TimerValue Register

Bit(s)	Name	Type	Description
31:0	Current Value	Read-only	The TimerValue register contains the current value of the decrementing counter. The TimerValue register is either 16 or 32 bits wide depending on the configuration of CoreTimer.

Timer Control Register – TimerControl

Table 8 Bit Assignments for the TimerControl Register

Bit(s)	Name	Type	Description
31:3	–	–	Unused; reads zero.
2	Timer Mode	Read/Write	Selects timer operation mode: 0: Continuous operation (default) 1: One-shot count
1	Interrupt Enable	Read/Write	Interrupt enable bit: 0: Timer interrupt disabled (default) 1: Timer interrupt enabled
0	Timer Enable	Read/Write	Enable bit for timer: 0: Timer disabled (default) 1: Timer enabled

Prescale Setting Register – TimerPrescale

This register contains a single 4-bit field that determines the effective clock rate for the timer counter, based on PCLK. Table 9 lists the bit assignments for the TimerPrescale register.

Table 9 Bit Assignments for the TimerPrescale Register

Bits	Name	Type	Function
31:4	–	–	Unused; reads zero.
3:0	Prescale	Read/Write	Prescale field. Determines effective clock rate for the counter based on PCLK: 0000 = divide by 2 (default) 0001 = divide by 4 0010 = divide by 8 0011 = divide by 16 0100 = divide by 32 0101 = divide by 64 0110 = divide by 128 0111 = divide by 256 1000 = divide by 512 1001 = divide by 1,024 Others = divide by 1,024

Interrupt Clear Register – TimerIntClr

Table 10 Bit Assignments for the TimerIntClr Register

Bit(s)	Name	Type	Description
31:0	Interrupt Clear	Write-only	Writing any value to this register clears (de-asserts) the TIMINT interrupt output from CoreTimer.

Raw Interrupt Status Register – TimerRIS

This register indicates the raw interrupt status from the counter. This value is ANDed with the Timer Interrupt Enable bit from the Timer Control register to create the masked interrupt, which is passed to the interrupt output pin. Table 11 lists the bit assignments for the TimerRIS register.

Table 11 Bit Assignments for the TimerRIS Register

Bit(s)	Name	Type	Function
31:1	–	–	Unused; reads zero.
0	Raw Timer Interrupt	Read	Raw interrupt status from the counter: 0: Raw interrupt not pending 1: Raw Interrupt pending

Interrupt Status Register – TimerMIS

This register indicates the masked interrupt status from the counter. This value is the logical AND of the raw interrupt status with the Timer Interrupt Enable bit from the Timer Control Register and is the same value that is passed to the interrupt output pin. Table 12 lists the bit assignments for the TimerMIS register.

Table 12 Bit Assignments for theTimerMIS Register

Bit(s)	Name	Type	Function
31:1	–	–	Unused; reads zero.
0	Timer Interrupt	Read	Enabled interrupt status from the counter: 0: Interrupt not pending 1: Interrupt pending

Timing Diagrams

APB Interface

Figure 3 and Figure 4 show the typical write cycle and read cycle timing relationships relative to the system clock.

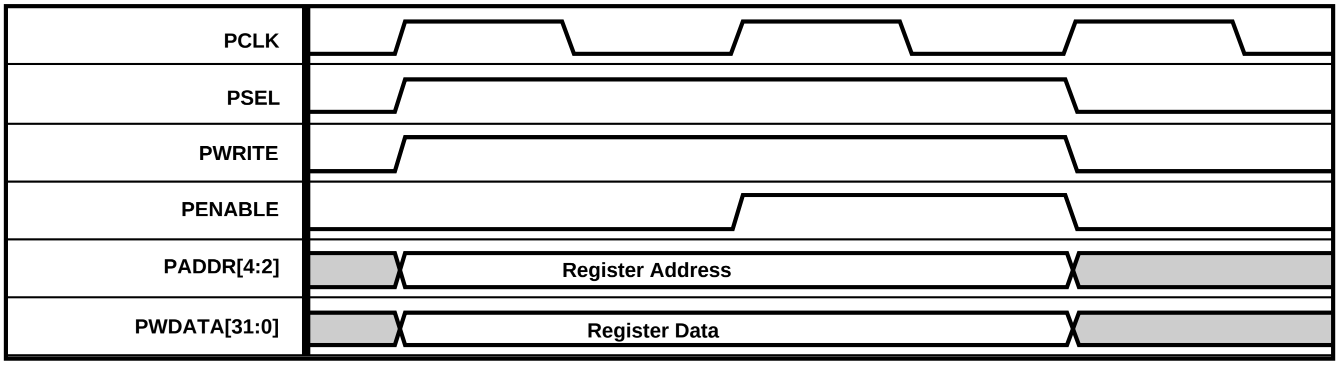


Figure 3 Data Write Cycle

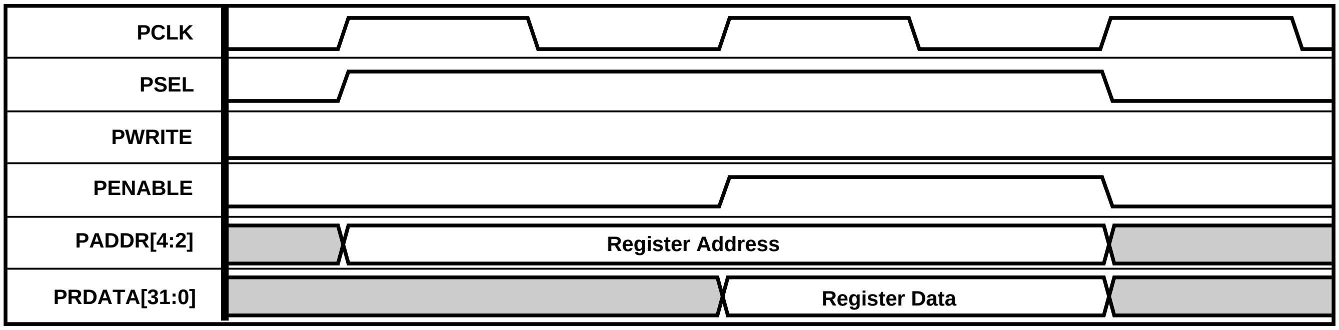


Figure 4 Data Read Cycle

Tool Flows

License

No license is required to use CoreTimer.

RTL

Complete unobfuscated RTL code is provided for the core.

Note: This allows the core to be instantiated with SmartDesign. Simulation, Synthesis, and Layout can be performed within Libero (SoC or IDE).

SmartDesign

CoreTimer (Figure 5) is preinstalled in the SmartDesign IP Deployment design environment. The core can be configured using the configuration GUI in SmartDesign, as shown in Figure 6.

For information on using SmartDesign to instantiate and generate cores, refer to the Using DirectCore in Libero® IDE User's Guide

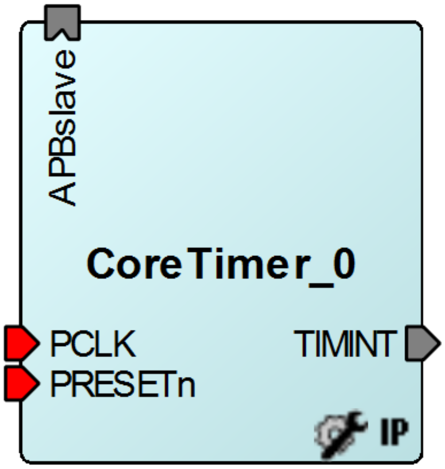
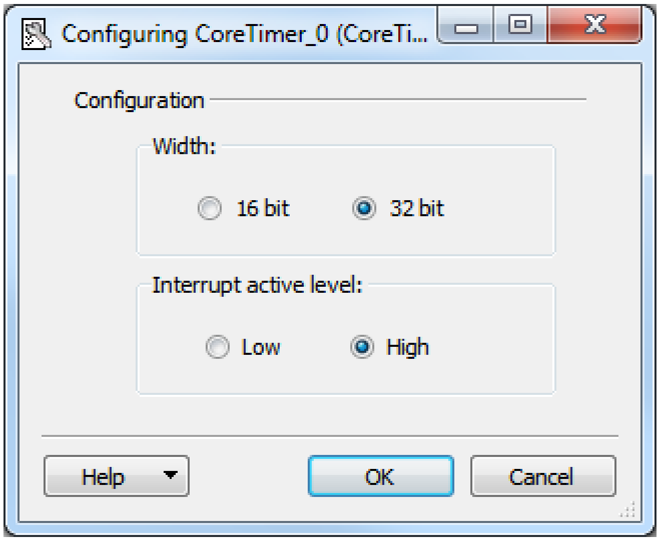


Figure 5 CoreTimer Full I/O View



Configuring CoreTimer_0 (CoreTi...

Configuration

Width:

○ 16 bit ● 32 bit

Interrupt active level:

○ Low ● High

Figure 6 CoreTimer Configurator

Simulation Flows

The User Testbench for CoreTimer is provided.

To run simulations, select the User Testbench flow within SmartDesign and click **Save & Generate** on the Generate pane. When SmartDesign generates the Libero project, it will install the user testbench files.

To run the user testbench, set the design root to the **CoreTimer instantiation** in the Libero design hierarchy pane and click the **Simulation** icon in the Libero Design Flow window. This will invoke ModelSim® and automatically run the simulation.

User Testbench

The simulation testbench for CoreTimer is shown in Figure 7 below. This testbench includes an instantiation of the CoreTimer macro along with a BFM-based APB Master. All BFM compilers are included for both the Linux and Windows operating systems, as well as the BFM vector source files. The user_tb.bfm file driving the APB master may be edited directly by the user if the simulation of a specific case is required.

The BFM (APB Master) performs APB writes and reads to the register interface of CoreTimer, to demonstrate the operating performance of CoreTimer in one-shot and continuous modes. The APB master registers the receipt of an interrupt from CoreTimer on the GP_IN port.

Note: The user testbench does not import the users' custom configuration parameters; only a single suite of predefined parameters are tested, which may be altered directly in the testbench.v/vhd file.

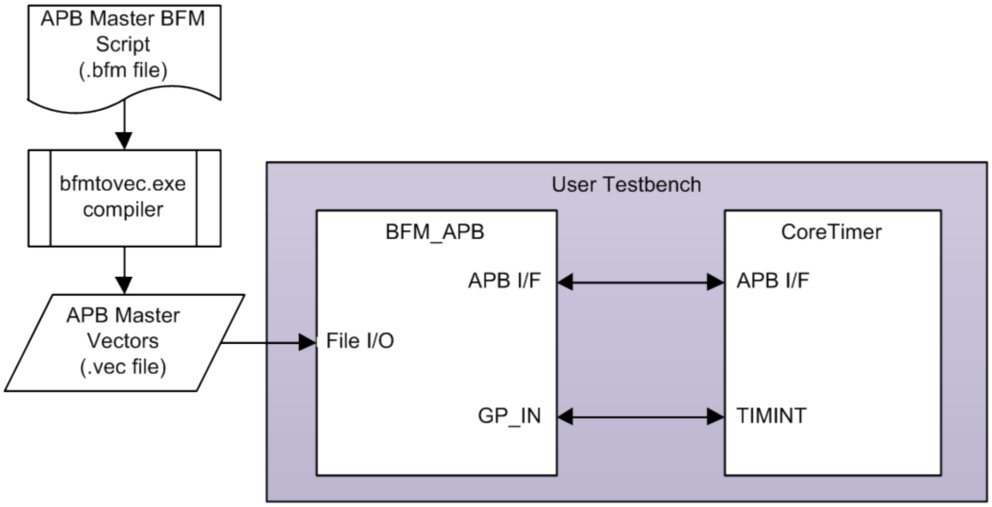


Figure 7 CoreTimer User Testbench

Synthesis in Libero SoC

Having set the design route appropriately, click the **Synthesis** icon in Libero SoC. The Synthesis window appears, displaying the Synplicity® project. Set Synplicity to use the Verilog 2001 standard if Verilog is being used. To run Synthesis, select the **Run** icon.

Place-and-Route in Libero SoC

Having set the design route appropriately and run Synthesis, click the **Layout** icon in the Libero SoC to invoke Designer. CoreTimer requires no special place-and-route settings.

List of Changes

The following table lists critical changes that were made in the current version of the document.

Date	Change (V2.0)	Page
June 2015	Migrated the CoreTimer handbook to the new specifications document structure.	NA
	Updated Table 1 and Table 2 for device utilization and performance details.	6 & 6
April 2015	Added support for SmartFusion, SmartFusion2, IGLOO2, and RTG4 device families.	NA

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/designsupport/fpga-soc-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 (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; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,600 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

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer’s responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided “as is, where is” and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.