CoreAPB3 v4.1

Handbook

Table of Contents

Introduction

Core Overview

CoreAPB3 is a bus component that provides an advanced microcontroller bus architecture (AMBA®) 3 advanced peripheral bus (APB) fabric for interconnecting between an APB master and up to 16 APB slaves. The slaves may be AMBA 2 or AMBA 3 compatible. Unlike AMBA 2 APB slaves, AMBA 3 APB slaves provide **ready** and **error** signals.

In this handbook, APB3 is used as an abbreviation for AMBA 3 APB.

CoreAPB3 supports a single APB3 master. For example, Core8051s or CoreABC. Alternatively, CoreAPB3 could be mastered by the CoreAHBtoAPB3 bridging core in a system where an Advanced High-Performance Bus (AHB) or AHB-Lite master interacts with some APB peripherals. CoreAHBtoAPB3 is used to bridge between CoreAHB or CoreAHBLite and CoreAPB3.

CoreAPB3 supports a number of configuration options and these allow the core to be adapted to suit a variety of systems. The data and address bus widths can be adjusted to suit the APB3 master. When the master address bus is less than 32 bits, a few options are available related to indirect addressing. The address space is evenly divided among 16 slave regions or slots, and access to each slot can be enabled or disabled. It is also possible to assign one or more slots to a “combined region”. This mechanism can be used to access, through a single slave interface, a resource whose size exceeds 1/16th of the address space. Slots allocated to a combined region do not have to be contiguous in the memory map and this means that it is possible to access a number of resources that are dispersed in the memory map through a single slave interface. This feature is most useful in a SmartFusion or SmartFusion2 device when a master based in the field programmable gate array (FPGA) fabric is required to access a number of resources in the microcontroller subsystem (MSS) through a single slave interface.

A block diagram of CoreAPB3 is shown in Figure 1. An example system using CoreAPB3 is shown in Figure 2.

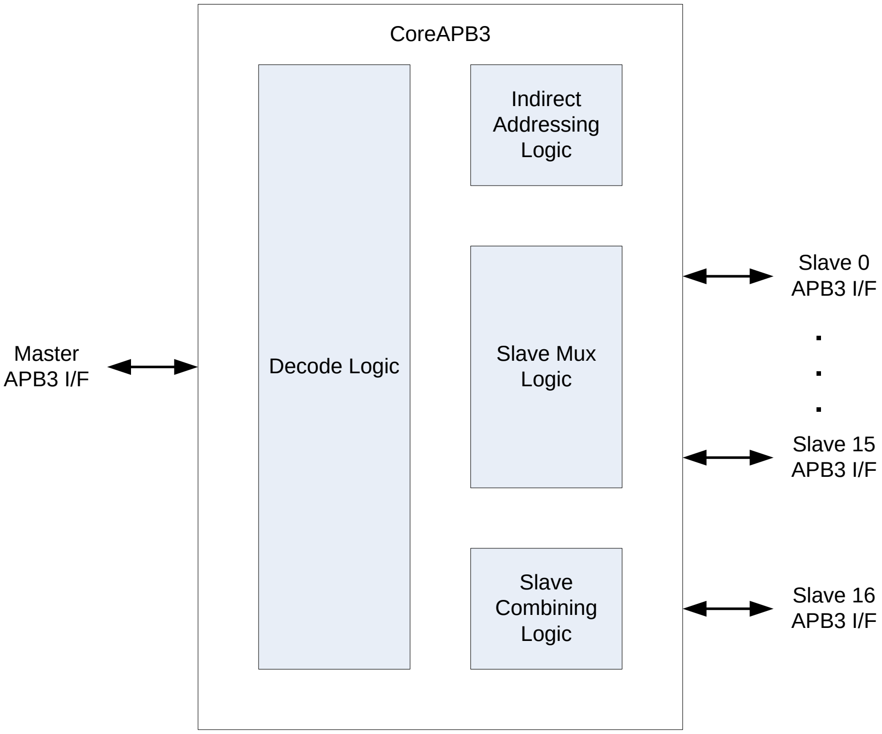


Figure 1. CoreAPB3 Block Diagram

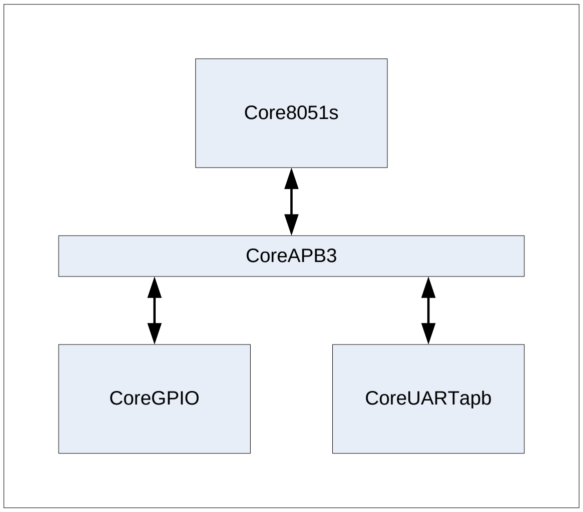


Figure 2. CoreAPB3 Example System

Key Features

- Supports up to 16 APB slaves
- Master data bus widths of 8, 16 or 32 bits are supported
- Supports master address bus widths ranging from 12 bits to 32 bits
- Indirect addressing allows a master with an address bus width less than 32 bits to address a memory space of up to 4 GB (2^{32})
- It is possible to “combine” several of the 16 slave slots to allow access to all of these slots through a single slave interface

Supported Microsemi FPGA Families

- SmartFusion2
- SmartFusion
- Fusion
- IGLOO, IGLOOe, IGLOO^{PLUS}
- ProASIC3, ProASIC3E, ProASIC3L
- ProASIC^{PLUS}
- Axcelerator
- RTAX-S
- IGLOO®2
- RTG4™

Core Version

This handbook supports CoreAPB3 version 4.1.

Supported Interfaces

CoreAPB3 has a single APB mirrored master interface that must be connected to an APB3 master and 16 APB mirrored slave interfaces that can be connected to APB peripherals.

Microsemi recommends using SmartDesign to connect and configure CoreAPB3 when creating a system design.

Interface Description

Configuration Parameters

The register transfer level (RTL) code for CoreAPB3 has parameters for configuring the core. When working with the core in SmartDesign, a configuration GUI is used to set the values of these parameters. The CoreAPB3 configuration parameters are described in Table 1.

Table 1 CoreAPB3 Configuration Parameters

Parameter Name	Valid Range	Default	Description
APB_DWIDTH	8, 16 or 32	32	APB master data bus width. This parameter must be set to match the width of the APB3 master connected to the CoreAPB3.
MADDR_BITS	12, 16, 20, 24, 28, 32	32	Number of address bits driven by the master. The width of the master address port on CoreAPB3 is 32 bits but not all of these bits may be driven by the master. Bits that are not driven should be tied low. This will happen automatically when the core is used within SmartDesign.
UPR_NIBBLE_POSN	2 – 8	7	Controls the position of the upper 4 bits (upper nibble) of the master address in the address bus output to the slaves. This parameter is only relevant when the master address bus width is less than 32 bits (MADDR_BITS < 32). Relative to the master address, it is only possible to “left shift” the upper nibble to higher order bits in the slave address. It is not possible to “right shift” the upper nibble to appear on lower order bits in the slave address. Settings are as follows: 2: master address upper nibble appears on bits [11:8] of slave address 3: master address upper nibble appears on bits [15:12] of slave address 4: master address upper nibble appears on bits [19:16] of slave address 5: master address upper nibble appears on bits [23:20] of slave address 6: master address upper nibble appears on bits [27:24] of slave address 7: master address upper nibble appears on bits [31:28] of slave address 8: master address upper nibble does not appear in slave address at all.
IADDR_OPTION	0 – 17	0	Controls the source of higher level address bits when indirect addressing is used. This parameter is only relevant when the master address bus width is less than 32 bits (MADDR_BITS < 32). Settings are as follows: 0: Indirect addressing is not in use 1: Upper address bits sourced from IADDR input 2: Upper address bits sourced from indirect address register(s) residing in slave slot 0. 3: Upper address bits sourced from indirect address register(s) residing in slave slot 1.

Parameter Name	Valid Range	Default	Description
			4: Upper address bits sourced from indirect address register(s) residing in slave slot 2. 5: Upper address bits sourced from indirect address register(s) residing in slave slot 3. 6: Upper address bits sourced from indirect address register(s) residing in slave slot 4. 7: Upper address bits sourced from indirect address register(s) residing in slave slot 5. 8: Upper address bits sourced from indirect address register(s) residing in slave slot 6. 9: Upper address bits sourced from indirect address register(s) residing in slave slot 7. 10: Upper address bits sourced from indirect address register(s) residing in slave slot 8. 11: Upper address bits sourced from indirect address register(s) residing in slave slot 9. 12: Upper address bits sourced from indirect address register(s) residing in slave slot 10. 13: Upper address bits sourced from indirect address register(s) residing in slave slot 11. 14: Upper address bits sourced from indirect address register(s) residing in slave slot 12. 15: Upper address bits sourced from indirect address register(s) residing in slave slot 13. 16: Upper address bits sourced from indirect address register(s) residing in slave slot 14. 17: Upper address bits sourced from indirect address register(s) residing in slave slot 15.
SC_0	0 or 1	0	0: Slot 0 is not assigned to the combined region 1: Slot 0 is assigned to the combined region. The slot 0 interface is not available for connection.
SC_1	0 or 1	0	0: Slot 1 is not assigned to the combined region 1: Slot 1 is assigned to the combined region. The slot 1 interface is not available for connection.
…	…	…	…
SC_15	0 or 1	0	0: Slot 15 is not assigned to the combined region 1: Slot 15 is assigned to the combined region. The slot 15 interface is not available for connection.
APBSLOT0ENABLE	0 or 1	1	0: Disables slave 0 1: Enables slave 0
APBSLOT1ENABLE	0 or 1	1	0: Disables slave 1 1: Enables slave 1
…	…	…	…
APBSLOT15ENABLE	0 or 1	1	0: Disables slave 15 1: Enables slave 15

Ports

The ports present on CoreAPB3 are listed in Table 2.

Table 2 CoreAPB3 Ports

Port Name	Type	Description
IADDR[31:0]	Input	Indirect address input. This port can be used as the source of higher address bits The value on this port is only relevant when the MADDR_BITS configuration parameter is set to a value less than 32 and the IADDR_OPTION configuration parameter is set to 1. The IADDR[31:0] port will only be displayed on the CoreAPB3 symbol in SmartDesign when MADDR_BITS < 32 and IADDR_OPTION = 1. In the configuration GUI, these settings corresponds to setting the "Number of address bits driven by master" option to a value less than 32 and setting the "Indirect Addressing" option to "Indirect address sourced from IADDR input port."
PRESETN	Input	APB reset, active low asynchronous reset
PCLK	Input	APB clock signal
PSEL	Input	APB select from master
PENABLE	Input	APB enable from master
PWRITE	Input	APB write indication from master
PADDR[31:0]	Input	APB address bus from master
PWDATA[31:0]	Input	APB write data from master. Depending on the data bus width configuration, it's possible that only the lower 8 or 16 bits of this bus may be in use.
PRDATA[31:0]	Output	APB read data output to master. Depending on the data bus width configuration, it's possible that only the lower 8 or 16 bits of this bus may be in use.
PREADY	Output	APB ready indication output to master
PSLVERR	Output	APB slave error indication to master
PENABLES	Output	APB enable to all slaves
PWRITES	Output	APB write indication to all slaves
PADDRS[31:0]	Output	APB address bus to all slaves
PWDATAS[31:0]	Output	APB write data to all slaves
PSELS0	Output	APB select signal to slave 0
PSELS1	Output	APB select signal to slave 1
...	…	…
PSELS15	Output	APB select signal to slave 15
PSELS16	Output	APB select signal to slave 16. This signal forms part of the "combined region" slave interface and is only relevant when the core has been appropriately configured by assigning some of the slave slots to the "combined region".

Port Name	Type	Description
PRDATAS0[31:0]	Input	APB read data from slave 0. Depending on the data bus width configuration, it's possible that only the lower 8 or 16 bits of this bus may be in use.
PRDATAS1[31:0]	Input	APB read data from slave 1. Depending on the data bus width configuration, it's possible that only the lower 8 or 16 bits of this bus may be in use.
…	…	…
PRDATAS15[31:0]	Input	APB read data from slave 15. Depending on the data bus width configuration, it's possible that only the lower 8 or 16 bits of this bus may be in use.
PRDATAS16[31:0]	Input	APB read data from slave 16. Depending on the data bus width configuration, it's possible that only the lower 8 or 16 bits of this bus may be in use. This data bus forms part of the "combined region" slave interface and is only relevant when the core has been appropriately configured by assigning some of the slave slots to the "combined region".
PREADYS0	Input	APB ready signal from slave 0.
PREADYS1	Input	APB ready signal from slave 1.
…	…	…
PREADYS15	Input	APB ready signal from slave 15.
PREADYS16	Input	APB ready signal from slave 16. This signal forms part of the "combined region" slave interface and is only relevant when the core has been appropriately configured by assigning some of the slave slots to the "combined region".
PSLVERRS0	Input	APB error indication signal from slave 0.
PSLVERRS1	Input	APB error indication signal from slave 1.
…	…	…
PSLVERRS15	Input	APB error indication signal from slave 15.
PSLVERRS16	Input	APB error indication signal from slave 16. This signal forms part of the "combined region" slave interface and is only relevant when the core has been appropriately configured by assigning some of the slave slots to the "combined region".

Tool Flows

Licensing

CoreAPB3 is licensed in two ways, Obfuscated or register transfer level (RTL).

Obfuscated

Complete RTL code is provided for the core, allowing the core to be instantiated with SmartDesign. Simulation, synthesis, and layout can be performed within Libero System-on-Chip (SoC). The RTL code for the core is obfuscated.

RTL

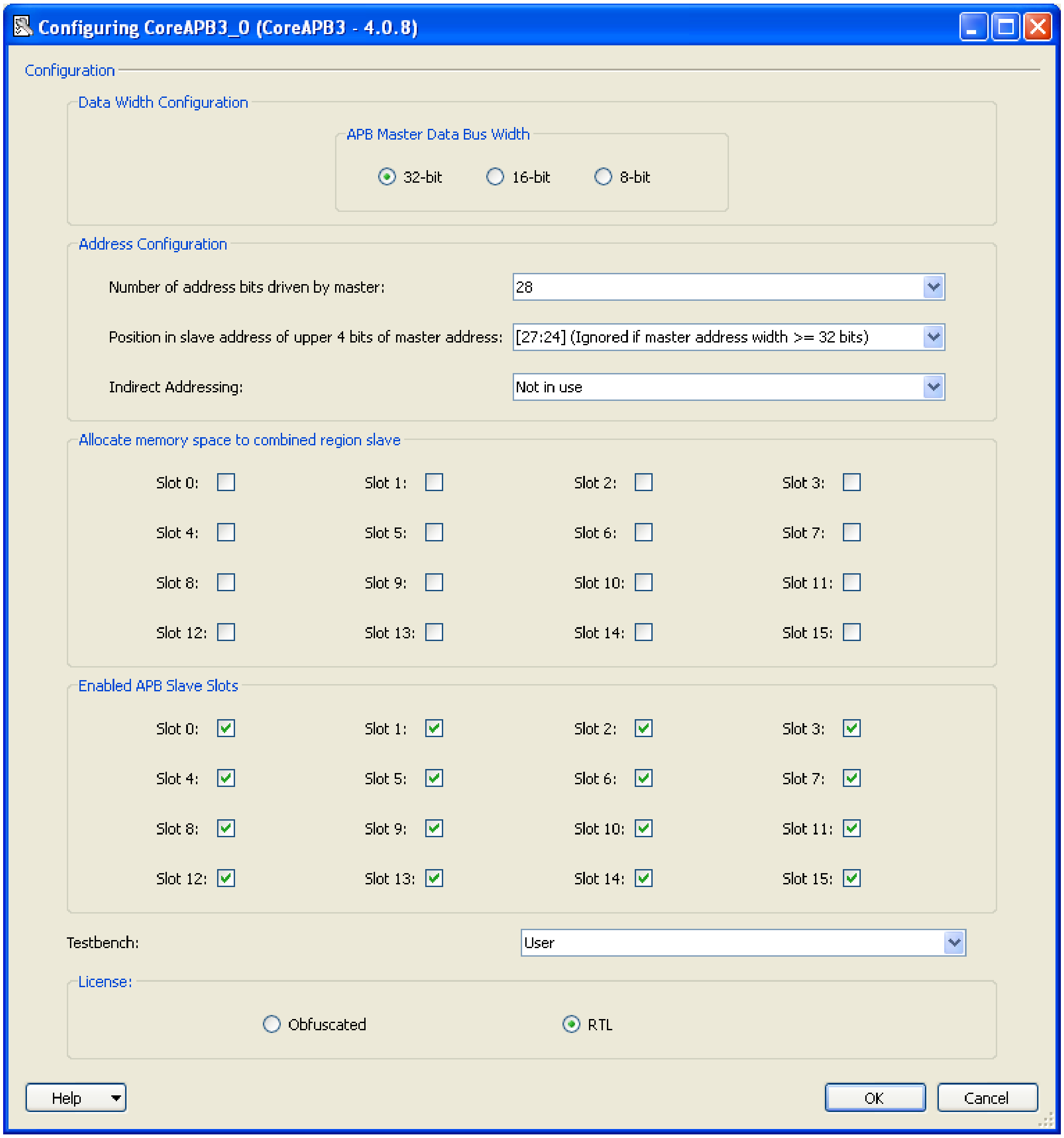
Complete RTL source code is provided for the core.

SmartDesign

CoreAPB3 is available through the Libero SoC IP Catalog. It can be downloaded from a remote web-based repository and installed into your local vault, ready for use. Once installed in Libero SoC, the core can be instantiated, configured, connected, and generated using the SmartDesign tool.

Configuring CoreAPB3 in SmartDesign

Figure 3 shows the CoreAPB3 configuration GUI.



Configuring CoreAPB3_0 (CoreAPB3 - 4.0.8)

Configuration

Data Width Configuration

APB Master Data Bus Width

● 32-bit ○ 16-bit ○ 8-bit

Address Configuration

Number of address bits driven by master: 28

Position in slave address of upper 4 bits of master address: [27:24] (Ignored if master address width >= 32 bits)

Indirect Addressing: Not in use

Allocate memory space to combined region slave

Slot 0: □ Slot 1: □ Slot 2: □ Slot 3: □
Slot 4: □ Slot 5: □ Slot 6: □ Slot 7: □
Slot 8: □ Slot 9: □ Slot 10: □ Slot 11: □
Slot 12: □ Slot 13: □ Slot 14: □ Slot 15: □

Enabled APB Slave Slots

Slot 0: ☑ Slot 1: ☑ Slot 2: ☑ Slot 3: ☑
Slot 4: ☑ Slot 5: ☑ Slot 6: ☑ Slot 7: ☑
Slot 8: ☑ Slot 9: ☑ Slot 10: ☑ Slot 11: ☑
Slot 12: ☑ Slot 13: ☑ Slot 14: ☑ Slot 15: ☑

Testbench: User

License:

○ Obfuscated ● RTL

Figure 3. CoreAPB3 Configuration GUI

The configuration options displayed in the configuration GUI correspond with the configuration parameters listed in Table 1.

For some of the configuration options, tooltips will pop up when the mouse pointer hovers over the option in the configuration GUI. These tooltips explain a little more about the related options. The following paragraphs describe the configuration options available for CoreAPB3 with reference to the configuration GUI.

Data Bus Width Configuration

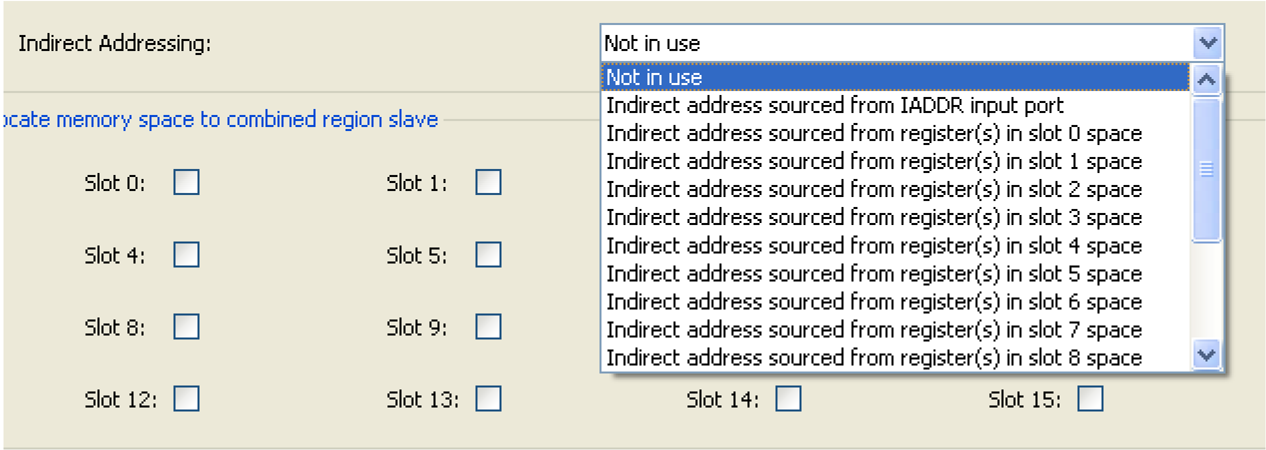
The data bus width can be set to 32, 16 or 8 bits. All of the data bus ports are 32 bits wide but the upper bits of these ports are not used when the data bus width is set to 16 or 8 bits. When the core is used in SmartDesign, any unused data bus inputs will be tied low automatically and unused outputs will be left unconnected.

Address Configuration

The number of address bits driven by the master connected to CoreAPB3 can be set to 12, 16, 20, 24, 28 or 32 bits. The width of the master address port is fixed at 32 bits and any unused upper bits of the address port will be tied low automatically when working with CoreAPB3 within SmartDesign.

The upper four address bits driven by the master are decoded to produce the select signals for the 16 slave slots, regardless of the master address bus width. If the master address bus width is 32 bits then the address appearing on the PADDRS[31:0] port that connects to all of the slaves will simply mirror the master address. If the master address bus width is less than 32 bits, then the lower (MADDR_BITS – 4) bits of the (32 bit) address to the slaves will be sourced from the master address and the value on the upper bits will be determined by the “Position in slave address of upper 4 bits of master address” and the “Indirect Addressing” settings.

The “Indirect Addressing” setting can be used to specify a source for the upper bits of the address to the slaves when the master address is less than 32 bits. The screenshot in Figure 4 shows some possible settings for “Indirect Addressing”.



Indirect Addressing: Not in use

Not in use
Indirect address sourced from IADDR input port
Indirect address sourced from register(s) in slot 0 space
Indirect address sourced from register(s) in slot 1 space
Indirect address sourced from register(s) in slot 2 space
Indirect address sourced from register(s) in slot 3 space
Indirect address sourced from register(s) in slot 4 space
Indirect address sourced from register(s) in slot 5 space
Indirect address sourced from register(s) in slot 6 space
Indirect address sourced from register(s) in slot 7 space
Indirect address sourced from register(s) in slot 8 space

ocate memory space to combined region slave

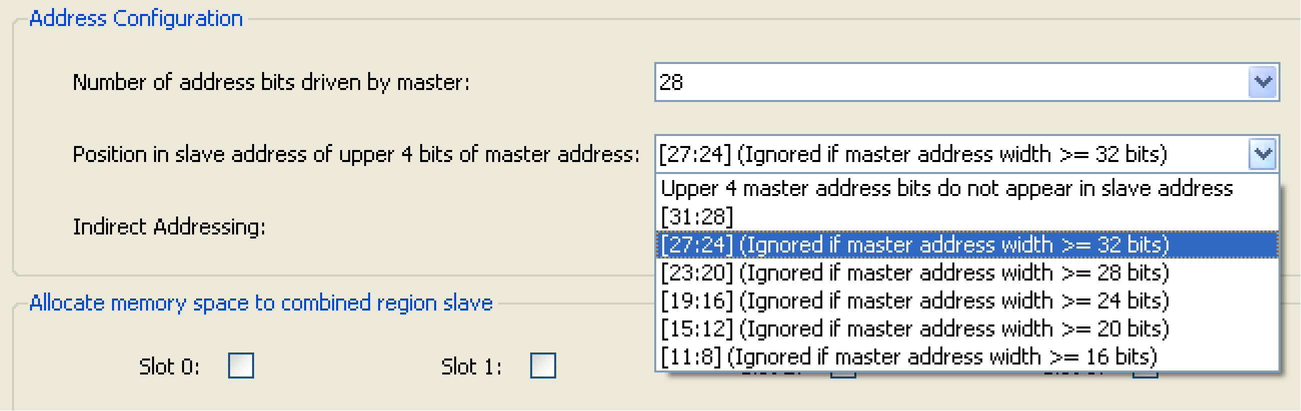
Slot 0: □ Slot 1: □
Slot 4: □ Slot 5: □
Slot 8: □ Slot 9: □
Slot 12: □ Slot 13: □ Slot 14: □ Slot 15: □

Figure 4. Indirect Addressing Options

As illustrated, upper, indirect address bits in the 32 bit address to slaves can be sourced from the IADDR input port of CoreAPB3. Alternatively, any of the slave slots can be used to hold indirect register(s) that a master can write and read to control indirect addressing. If a slave slot is used to implement indirect address registers then that slot is no longer available for connection to an APB peripheral. If the “Not in use” setting is selected for “Indirect Addressing” then any upper bits of the slave address that are not driven by the master are tied low.

Even though the IADDR input port is 32 bits wide, and indirect address register(s) can hold a 32 bit value, not all of these bits will be used. The lower bits of the slave address will always be sourced from the master address. When indirect address bits do feature in the slave address, there will be a one to one mapping for these indirect bits in terms of bit positions between the indirect address and the slave address.

The “Position in slave address of upper 4 bits of master address” setting can be used to manipulate the address output to the slaves by providing the option to left shift the four most significant address bits driven by the master to a higher order position in the slave address. This setting is only relevant when the master address width is less than 32 bits. Figure 5 shows the possible settings for the “Position in slave address of upper 4 bits of master address” option.



Address Configuration

Number of address bits driven by master: 28

Position in slave address of upper 4 bits of master address: [27:24] (Ignored if master address width >= 32 bits)

Indirect Addressing:

- Upper 4 master address bits do not appear in slave address
- [31:28]
- [27:24] (Ignored if master address width >= 32 bits)
- [23:20] (Ignored if master address width >= 28 bits)
- [19:16] (Ignored if master address width >= 24 bits)
- [15:12] (Ignored if master address width >= 20 bits)
- [11:8] (Ignored if master address width >= 16 bits)

Allocate memory space to combined region slave

Slot 0: □ Slot 1: □

Figure 5. Option to Left Shift Upper 4 bits of Master Address

Shifting the position of the upper four master address bits in the slave address may be useful when some further decoding of the slave address takes place and the master wants to access a number of resources on the other side of this additional decoding without necessarily having to adjust an indirect address value between accesses.

The effect on the slave address of left shifting of the upper four master address bits can be illustrated by considering a master that drives 12 address bits. In this case there are seven possible formats for the (32 bit) slave address and these are shown in Figure 6. The horizontal bars in this figure represent the different possible addresses and the numbers above each bar indicate bit positions in the addresses.

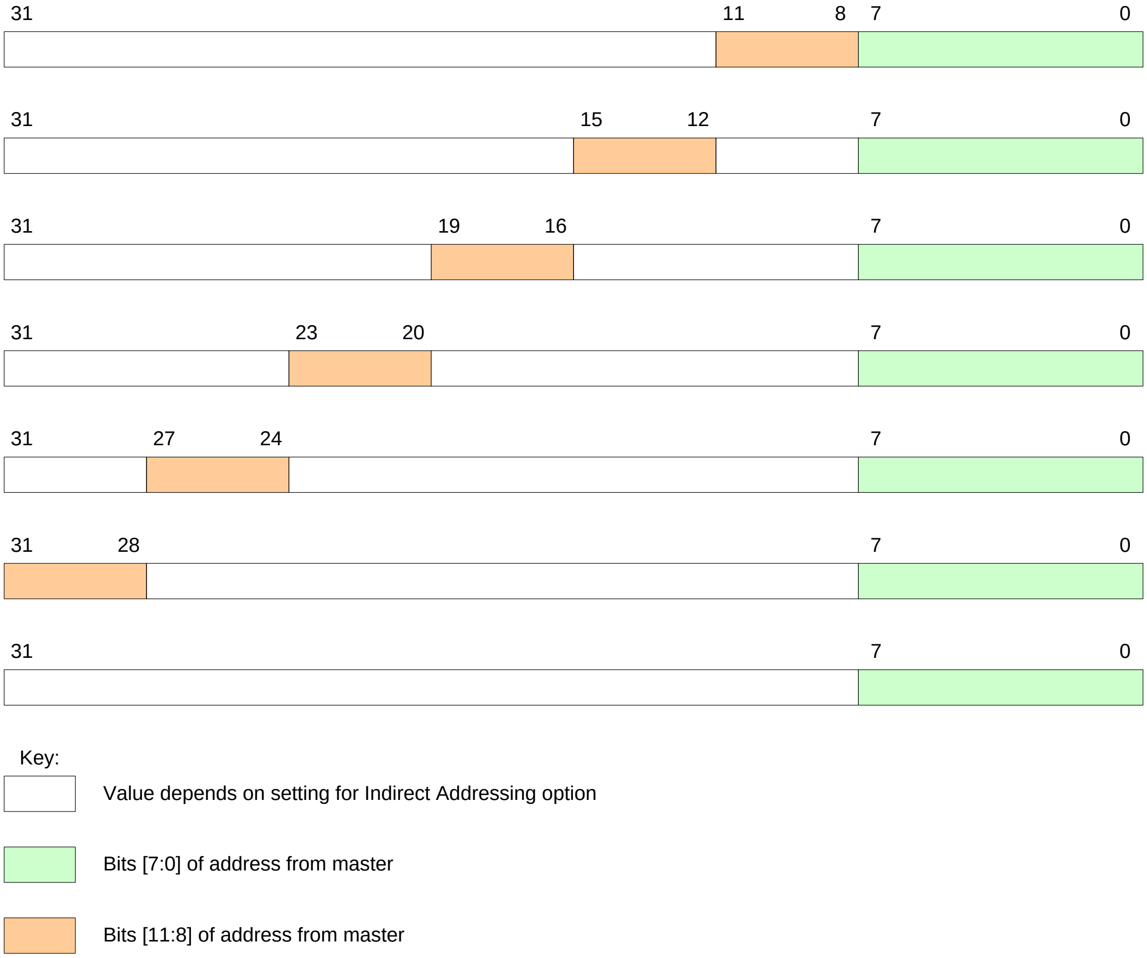
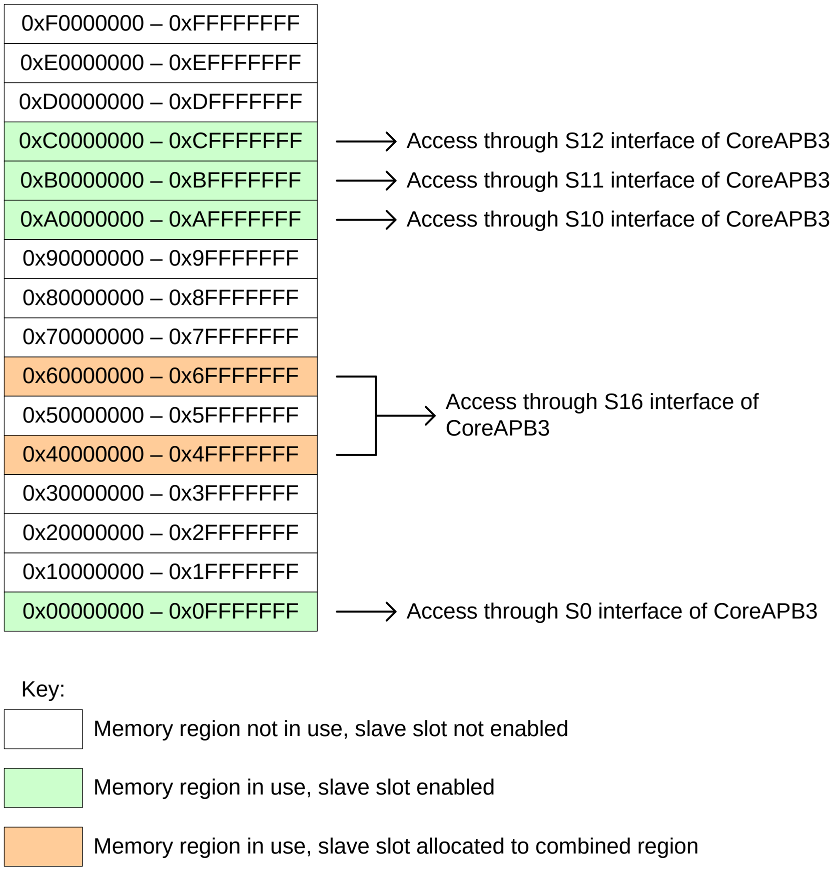

31
11
8
7
0
31
15
12
7
0
31
19
16
7
0
31
23
20
7
0
31
27
24
7
0
31
28
7
0
31
7
0
Key:
Value depends on setting for Indirect Addressing option
Bits [7:0] of address from master
Bits [11:8] of address from master

Figure 6. Slave Address Formats for 12 bit Master Address

Combining Slave Slots

One or more slave slots can be assigned to a “combined region” by checking the checkboxes in the “Allocate memory space to combined region slave” region of the configuration GUI. When some slots have been allocated to a combined region, an additional slave interface, labeled S16, appears for connection on the CoreAPB3 symbol in SmartDesign. Combining slave slots provides a means to access a region larger than the size of a slot through a single slave interface. If slots are combined, they do not necessarily have to be contiguous in the memory space. If a slave slot is allocated to the combined region then its corresponding interface is no longer available for connection separately.

Figure 7 shows an example memory map where slot combining is used. In this example an APB master (with a 32 bit address bus) can access slaves based at addresses 0x00000000, 0xA0000000, 0xB0000000 and 0xC0000000 through individual slave interfaces of CoreAPB3 as illustrated in the figure. Additionally, any access from the master with an address in the ranges 0x40000000 – 0x4FFFFFFF or 0x60000000 – 0x6FFFFFFF will result in an access on the S16 slave interface of CoreAPB3.



0xF0000000 – 0xFFFFFFFF	
0xE0000000 – 0xEFFFFFFF	
0xD0000000 – 0xDFFFFFFF	
🟩 0xC0000000 – 0xCFFFFFFF	Access through S12 interface of CoreAPB3
🟩 0xB0000000 – 0xBFFFFFFF	Access through S11 interface of CoreAPB3
🟩 0xA0000000 – 0xAFFFFFFF	Access through S10 interface of CoreAPB3
0x90000000 – 0x9FFFFFFF	
0x80000000 – 0x8FFFFFFF	
0x70000000 – 0x7FFFFFFF	
🟧 0x60000000 – 0x6FFFFFFF	Access through S16 interface of CoreAPB3
0x50000000 – 0x5FFFFFFF	
🟧 0x40000000 – 0x4FFFFFFF	
0x30000000 – 0x3FFFFFFF	
0x20000000 – 0x2FFFFFFF	
0x10000000 – 0x1FFFFFFF	
🟩 0x00000000 – 0x0FFFFFFF	Access through S0 interface of CoreAPB3

Key:

□ Memory region not in use, slave slot not enabled

🟩 Memory region in use, slave slot enabled

🟧 Memory region in use, slave slot allocated to combined region

Figure 7. Example Memory Map Showing Use of Slot Combining

Enabling of Slave Slots

Checkboxes are provided in the configuration GUI to enable or disable each slave slot. The enable checkbox for any slot assigned to the combined region is grayed out since it is not possible to connect a slave to that slot. If a slot is disabled its corresponding interface does not appear for connection on the CoreAPB3 symbol in SmartDesign. The “combined region” slave interface, labeled S16, does not have an enable check box associated with it; instead this interface becomes available for connection if any slots have been assigned to the combined region.

Memory Map

Division of Address Space into Slots

In its simplest configuration, CoreAPB3 will divide the address space seen by the master into 16 slave slots. Table 3 lists the address ranges for the slave slots for a number of master address bus widths. Slave slots can be combined to create a portion of the memory map that is accessed through an additional, combined region slave interface. This interface will be labeled S16 on the CoreAPB3 symbol in SmartDesign. If indirect addressing and/or left shifting of the upper 4 bits of the master address in the slave address is in use, then it is possible to access a slave resource whose address lies outside the basic slot ranges listed in Table 3.

Table 3 CoreAPB3 Memory Map for Some Sample Master Address Bus Widths

Resource	Address Space		
	12 bit Master Address	**20 bit Master Address**	**32 bit Master Address**
Slave 0	0x000 – 0x0FF	0x00000 – 0x0FFFF	0x00000000 – 0x0FFFFFFF
Slave 1	0x100 – 0x1FF	0x10000 – 0x1FFFF	0x10000000 – 0x1FFFFFFF
Slave 2	0x200 – 0x2FF	0x20000 – 0x2FFFF	0x20000000 – 0x2FFFFFFF
Slave 3	0x300 – 0x3FF	0x30000 – 0x3FFFF	0x30000000 – 0x3FFFFFFF
Slave 4	0x400 – 0x4FF	0x40000 – 0x4FFFF	0x40000000 – 0x4FFFFFFF
Slave 5	0x500 – 0x5FF	0x50000 – 0x5FFFF	0x50000000 – 0x5FFFFFFF
Slave 6	0x600 – 0x6FF	0x60000 – 0x6FFFF	0x60000000 – 0x6FFFFFFF
Slave 7	0x700 – 0x7FF	0x70000 – 0x7FFFF	0x70000000 – 0x7FFFFFFF
Slave 8	0x800 – 0x8FF	0x80000 – 0x8FFFF	0x80000000 – 0x8FFFFFFF
Slave 9	0x900 – 0x9FF	0x90000 – 0x9FFFF	0x90000000 – 0x9FFFFFFF
Slave 10	0xA00 – 0xAFF	0xA0000 – 0xAFFFF	0xA0000000 – 0xAFFFFFFF
Slave 11	0xB00 – 0xBFF	0xB0000 – 0xBFFFF	0xB0000000 – 0xBFFFFFFF
Slave 12	0xC00 – 0xCFF	0xC0000 – 0xCFFFF	0xC0000000 – 0xCFFFFFFF
Slave 13	0xD00 – 0xDFF	0xD0000 – 0xDFFFF	0xD0000000 – 0xDFFFFFFF
Slave 14	0xE00 – 0xEFF	0xE0000 – 0xEFFFF	0xE0000000 – 0xEFFFFFFF
Slave 15	0xF00 – 0xFFF	0xF0000 – 0xFFFFF	0xF0000000 – 0xFFFFFFFF

Indirect Address Registers

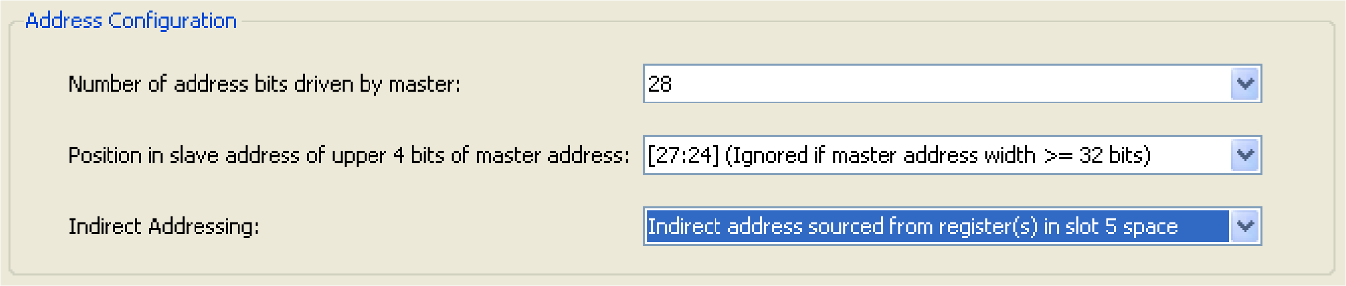
If the Indirect Addressing configuration option is set such that any of the slave slots are used to implement registers to hold the indirect address, then some accessible registers reside within CoreAPB3 itself. The number of registers and where they appear in the memory map as seen by the master connected to CoreAPB3 depend on the slave slot selected for the indirect address register(s) and also on the data width configuration.

If the data width is 32 bits then there is a single 32 bit indirect address register located at the base address of the slave slot chosen for the indirect address. If the data width is 16 bits then two 16 bit registers at offsets of 0x0 and 0x4 are used to hold the 32 bit indirect address. For a data width of 8 bits, four 8 bit registers at offsets 0x0, 0x4, 0x8 and 0xC are used to store the 32 bit indirect address. Table 4 shows the address offsets for indirect address registers for data widths of 32, 16 and 8 bits.

Table 4 Address Offsets for Indirect Address Registers

Data width	Offset	Register Description
32	0x0	Bits [31:0] of indirect address
16	0x0	Bits [15:0] of indirect address
	0x4	Bits [31:16] of indirect address
8	0x0	Bits [7:0] of indirect address
	0x4	Bits [15:8] of indirect address
	0x8	Bits [23:16] of indirect address
	0xC	Bits [31:24] of indirect address

Consider, as an example, that slave slot 5 is used to implement indirect address registers, by setting the Indirect Addressing configuration option as shown in Figure 8.


Address Configuration

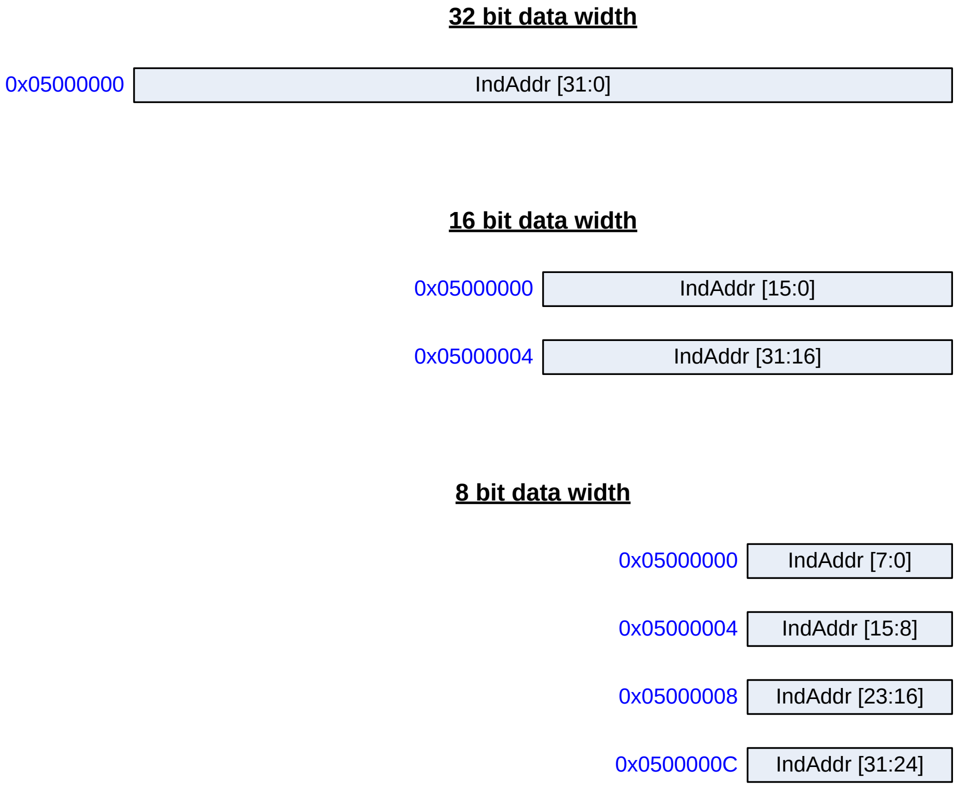
Number of address bits driven by master: 28

Position in slave address of upper 4 bits of master address: [27:24] (Ignored if master address width >= 32 bits)

Indirect Addressing: Indirect address sourced from register(s) in slot 5 space

Figure 8. Selecting Slot 5 for Indirect Address Registers

For the settings in Figure 8, the possible arrangements (depending on the data width setting) of the indirect address registers in the memory map as seen by the master of CoreAPB3 are shown in Figure 9.



Note:
1. IndAddr = Indirect Address
2. Addresses are shown in blue type

Figure 9. Indirect Address Register(s) Arrangement When Slot 5 Selected to Hold Indirect Address.

Note that although the indirect address register(s) hold a 32 bit address, at most only the upper 24 bits of this value will be used. The lower bits of the address output from CoreAPB3 to the slaves will be sourced from the master address. Have a look at Figure 6 on page 14 to see how the indirect address can contribute to the slave address. In that figure, portions of the address formats that are shown in white could be sourced from the indirect address register(s). When indirect address register(s) are providing part of the slave address, there is a direct mapping between the indirect address and the slave address in terms of bit positions. So, choosing the last address format shown in Figure 6 as an example, bits [31:8] of the slave address could be sourced from the indirect address. Bits [7:0] of the slave address would always be sourced from the master address.

Aside from the indirect address register(s), the white portions of the addresses in Figure 6 could be sourced from the IADDR input port or these white parts could simply be zeroed if the Indirect Addressing configuration option is set to “Not in use”.

The number of indirect address bits that come into play reduces as the master address width is increased. If the master address width is 32 bits, then indirect addressing, and the option to left shift the upper 4 bits of the master address, do not feature at all since the master can fully control all 32 bits of the address to the slaves. The slave address will simply mirror the master address in this case.

Ordering Information

Ordering Codes

CoreAPB3 can be ordered through your local Sales Representative. It should be ordered using the following number scheme: CoreAPB3-XX, where XX is listed in Table 5.

Table 5 Ordering Codes

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

List of Changes

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

Date	Change	Page
Revision 3 (December 2014)	Updated to suit v4.1 of CoreAPB3	N/A
Revision 2 (February 2013)	Updated to suit v4.0 of CoreAPB3	N/A
Revision 1 (February 2010)	Updated to suit v3.0 of CoreAPB3	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 (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 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**.