

CoreAPB3 v3.0

Handbook

Actel Corporation, Mountain View, CA 94043

© 2010 Actel Corporation. All rights reserved.

Printed in the United States of America

Part Number: 50200211-0

Release: February 2010

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Actel.

Actel makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability or fitness for a particular purpose. Information in this document is subject to change without notice. Actel assumes no responsibility for any errors that may appear in this document.

This document contains confidential proprietary information that is not to be disclosed to any unauthorized person without prior written consent of Actel Corporation.

Trademarks

Actel, IGLOO, Actel Fusion, ProASIC, Libero, Pigeon Point and the associated logos are trademarks or registered trademarks of Actel Corporation. All other trademarks and service marks are the property of their respective owners.

Table of Contents

Introduction	5
Key Features	6
Supported Actel FPGA Families	6
Core Version	7
Supported Interfaces.....	7
Utilization and Performance.....	7
Design Description	9
Verilog/VHDL Parameters.....	9
I/O Signals.....	10
Memory Map (programmer's view)	12
Design Details.....	15
Functional Blocks	15
APB Interface Timing	15
Tool Flows	17
Licensing	17
SmartDesign	17
Simulation Flows	18
Synthesis in Libero IDE.....	19
Place-and-Route in Libero IDE	20
System Operation	21
Usage with Core8051s.....	21
Usage with CoreABC	21
Ordering Information.....	24
Ordering Codes.....	24
Product Support	25
Customer Service.....	25
Actel Technical Support	25

Introduction

CoreAPB3 is a bus component that provides an advanced microcontroller bus architecture (AMBA3) advanced peripheral bus (APB3) fabric supporting up to 16 APB slaves. CoreAPB3 can be used with an APB3 Master that does not have built-in APB address decoding, such as Core8051s or CoreABC, to access APB (AMBA 2 or AMBA 3) peripherals; alternatively, CoreAPB3 can be used in conjunction with the CoreAHBtoAPB3 bridge to connect an AHB or AHB-Lite bus fabric such as CoreAHBLite to access APB peripherals from an AMBA high-performance bus (AHB) or AHB-Lite Master such as an ARM[®] Cortex[™]-M1. A single APB3 Master is connected to CoreAPB3. The master's PSEL and PADDR signals are used within CoreAPB3 to decode the appropriate PSELS slave select signals (only one can be active at any one time). This address decoding is dependent on the RANGESIZE hardware parameter/generic. Two possible addressing modes are available: direct addressing mode and indirect addressing mode. When direct addressing mode is used, all 16 APB slots are always of equal size, and that size may be configured as any value in powers of 2 from $2^{**}8 = 256$ locations to $2^{**}20 = 1048576$ locations by setting RANGESIZE appropriately in the IP configuration dialog in the SmartDesign canvas within Libero[®] Integrated Design Environment (IDE). When indirect addressing mode is used, slot 0 can access a 32-bit (4 GB) address space.

CoreAPB3 multiplexes the PRDATAS read data buses, PREADY signals, and PSLVERRS signals from the APB peripherals to send to the APB3 master. The CoreAPB3 block diagram is shown in Figure 1. A typical application using CoreAPB3 is shown in Figure 2.

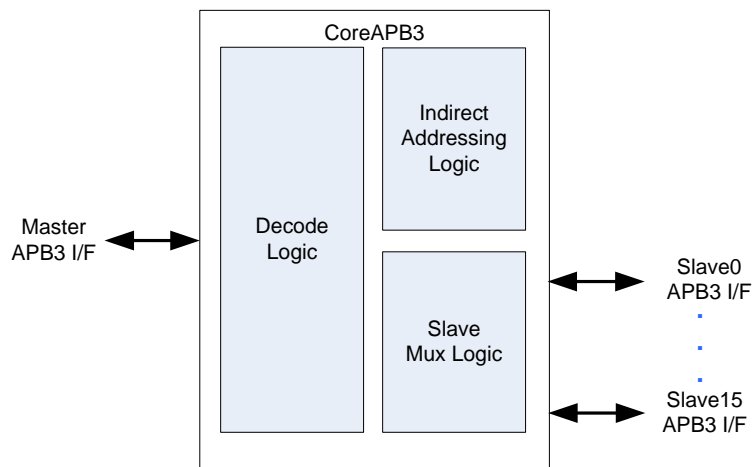


Figure 1 CoreAPB3 Block Diagram

CoreAPB3 has two possible addressing modes:

1. A direct addressing mode is used to directly access all 16 possible peripherals based on the PADDR bus and PSEL signal from the APB3 master. This mode is the same as has been used in versions of CoreAPB3 prior to version 3.0.
2. An indirect addressing mode is used to indirectly write to or read from a peripheral on slot 0 that requires a 32-bit (4 GB) address space. This mode has the following requirements:

- Slot 1 is unavailable for a peripheral. Slot 1 is used for storing an indirect addressing register.
- When the peripheral in slot 0 is selected by the master, the lower bits of PADDR (specifically, bits $\log_2(\text{RANGESIZE}) - 1$ down to 0) are sent to the peripheral on slot 0 along with the upper bits of the indirect addressing register (specifically, bits 32 down to $\log_2(\text{RANGESIZE})$) to create an effective 32-bit address.
- Slots 2 through 15 are accessed the same as in the direct addressing mode.

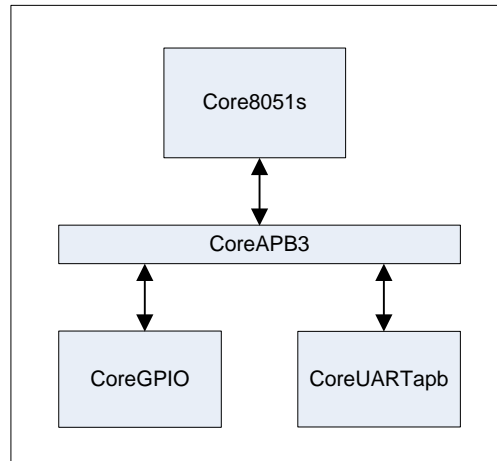


Figure 2 CoreAPB3 Typical Application

Key Features

- Supports up to 16 slave devices (APB2 or APB3)
- Automatic connection to AHB or AHB-Lite bridge (CoreAHBtoAPB3) and APB slaves, as well as APB3 slaves
- Provides configurable address decoding for APB slots
- Optional indirect addressing mode to access a 32-bit address (4 GB) peripheral

Supported Actel FPGA Families

This version of CoreAPB3 supports the following families:

- IGLOO®, IGLOOe, IGLOO PLUS
- ProASIC®3, ProASIC3E, ProASIC3L
- Fusion
- ProASIC^{PLUS}®
- Axcelerator®
- RTAX-S
- RTAX-DSP
- RTSX-SU
- SX-A

Core Version

This handbook supports CoreAPB3 version 3.0.

Supported Interfaces

CoreAPB3 is available with 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 (up to 16).

Actel recommends using SmartDesign to connect and configure CoreAPB3 in a processor-based system.

Utilization and Performance

CoreAPB3 has been implemented in several Actel device families. A summary of the implementation data for CoreAPB3 with RANGESIZE=256 is listed in Table 1, Table 2, and Table 3.

Table 1 CoreAPB3 Device Utilization and Performance (minimum configuration, 1 slave, direct addressing)

Family	Tiles			Utilization		Performance MHz
	Sequential	Combinatorial	Total	Device	Total %	
IGLOO/e, IGLOO PLUS	0	38	38	M1AGL1000V2	<1%	77 MHz
ProASIC3/E	0	38	38	M1A3P1000	<1%	>100 MHz
ProASIC3L	0	38	38	M1A3P1000L	<1%	>100 MHz
Fusion	0	38	38	M1AFS1500	<1%	>100 MHz
ProASIC	0	38	38	APA1000	<1%	40 MHz
Axcelerator	0	39	39	AX1000	<1%	>100 MHz
RTAX-S/DSP	0	39	39	RTAX1000S	<1%	>100 MHz
RTSX-SU	0	39	39	RTSX72SU	<1%	40 MHz
SX-A	0	39	39	A54SXA72A	<1%	52 MHz

Note: Data in this table were achieved using typical synthesis and layout settings

Table 2 CoreAPB3 Device Utilization and Performance (maximum configuration, 16 slaves, direct addressing)

Family	Tiles			Utilization		Performance MHz
	Sequential	Combinatorial	Total	Device	Total %	
IGLOO/e, IGLOO PLUS	0	591	591	M1AGL1000V2	<3%	35 MHz
ProASIC3/E	0	591	591	M1A3P1000	<3%	83 MHz
ProASIC3L	0	591	591	M1A3P1000L	<3%	67 MHz
Fusion	0	591	591	M1AFS1500	<2%	71 MHz
ProASIC ^{PLUS}	0	625	625	APA1000	<2%	22 MHz
Axcelerator	0	250	250	AX1000	<2%	>100 MHz
RTAX-S/DSP	0	250	250	RTAX1000S	<2%	>100 MHz
RTSX-SU	0	285	285	RTSX72SU	<5%	23 MHz
SX-A	0	285	285	A54SXA72A	<5%	36 MHz

Note: Data in this table were achieved using typical synthesis and layout settings

Table 3 CoreAPB3 Device Utilization and Performance (maximum configuration, 15 slaves¹, indirect addressing)

Family	Tiles			Utilization		Performance MHz
	Sequential	Combinatorial	Total	Device	Total %	
IGLOO/e, IGLOO PLUS	24	666	690	M1AGL10000V2	<3%	27 MHz
ProASIC3/E	24	666	690	M1A3P1000	<3%	68 MHz
ProASIC3L	24	666	690	M1A3P1000L	<3%	52 MHz
Fusion	24	666	690	M1AFS1500	<2%	53 MHz
ProASIC ^{PLUS}	24	715	739	APA1000	<2%	20 MHz
Axcelerator	24	316	340	AX1000	<2%	>100 MHz
RTAX-S/DSP	24	316	340	RTAX1000S	<2%	>100 MHz
RTSX-SU	24	353	377	RTSX72SU	<9%	21 MHz
SX-A	24	353	377	A54SXA72A	<9%	35 MHz

Note: Data in this table were achieved using typical synthesis and layout settings

¹ Slave 1 is unavailable due to its use to store the indirect addressing register for Slave 0.

Design Description

Verilog/VHDL Parameters

CoreAPB3 has parameters (Verilog) or generics (VHDL) for configuring the RTL code, described in Table 4. All parameters and generics are integer types.

Table 4 CoreAPB3 Parameters/Generics Descriptions

Parameter Name	Valid Range	Default	Description
APB_DWIDTH	8, 16, 32	32	APB master data bit width: 8, 16, or 32. This parameter must be set to match the width of the APB3 master you have connected to CoreAPB3.
RANGESIZE	256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576	256	APB slave slot size from 256 to 1048576 in powers of 2 (i.e., 2^{**8} to 2^{**20}).
IADDR_ENABLE	0 or 1	0	Indirect address enable 0 – Direct addressing (Legacy mode): 16 possible slots, range size determined by the RANGESIZE parameter. 1 – Indirect addressing: slot 1 is used to store an address register, and slot 0 uses the indirect address register to set the upper bits of a 32-bit PADDR bus to allow a full 32-bit address range for slot 0; slots 2-15 are the same as in Direct Addressing mode. <i>Note: If this parameter is set to 1, slot 1 is unavailable to be connected to slave 1 since it is used to store the indirect addressing registers.</i>
APBSLOT0ENABLE	0 or 1	1	Enables or disables slave 0 for APB master 0 – Disables slave 0 for APB master 1 – Enables slave 0 for APB master (default)
APBSLOT1ENABLE	0 or 1	1	Enables or disables slave 1 for APB master 0 – Disables slave 1 for APB master 1 – Enables slave 1 for APB master (default) If this parameter is set to 1, slot 1 is unavailable to be connected to slave 1 since it is used to store the indirect addressing registers.
APBSLOT2ENABLE	0 or 1	1	Enables or disables slave 2 for APB master 0 – Disables slave 2 for APB master 1 – Enables slave 2 for APB master (default)
...

Parameter Name	Valid Range	Default	Description
APBSLOT15ENABLE	0 or 1	1	Enables or disables slave 15 for APB master 0 – Disables slave 15 for APB master 1 – Enables slave 15 for APB master (default)

I/O Signals

The port signals for the CoreAPB3 macro are illustrated in Figure 3 and defined in Table 5.

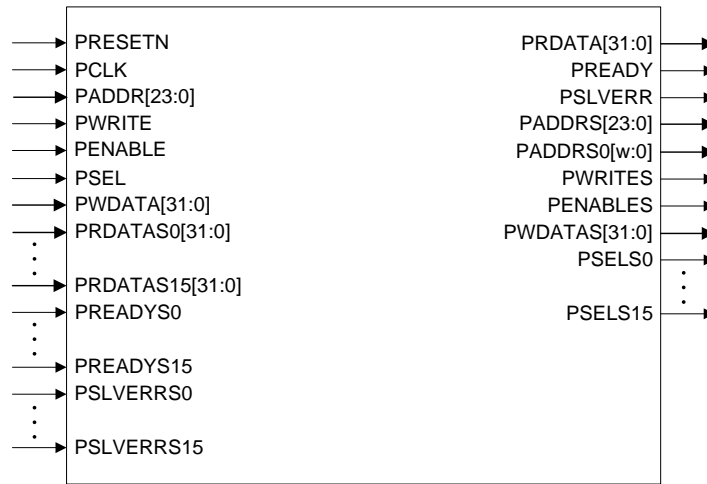


Figure 3 CoreAPB3 I/O Signal Diagram

Table 5 CoreAPB3 I/O Signal Descriptions

Port Name	Type	Description
APB3 Reset and Clock		
PRESETN	In	APB active Low asynchronous reset. <i>Note: This signal is only used if the parameter IADDR_ENABLE = 1.</i>
PCLK	In	APB clock: All APB signals are synchronous to the rising edge of this clock signal. <i>Note: This signal is only used if the parameter IADDR_ENABLE = 1.</i>
APB3 Master Interface		
PADDR[23:0]	In	APB master address bus. This port is used to address registers and message data for the APB3 master within CoreAPB3 destined for APB3 slaves.
PWRITE	In	APB write/read for APB3 master. If High, a write will occur when an APB transfer to CoreAPB3 takes place. If Low, a read from CoreAPB3 will take place.
PENABLE	In	APB enable for APB3 master. If High, the APB3 master is enabling access to one of the APB3 slots (slaves) during the second (or greater if PREADY is Low) cycles.
PSEL	In	APB select for APB3 master. If High, the APB3 master is selecting one of the APB3 slots (slaves).

Port Name	Type	Description
PWDATA[31:0]	In	APB write data from APB3 master to selected slave. Even though the bit-width of this port is 32, the effective bit-width is dictated by the setting of the APB_DWIDTH parameter setting. For example, if APB_DWIDTH = 16, only bits 15:0 from this port would be used internally and bits 31:16 would be ignored.
PRDATA[31:0]	Out	APB read data to APB3 master from selected slave.
PREADY	Out	APB ready signal for APB3 master. This signal is normally High, however, a selected slave can extend a transfer by holding this signal Low and then bringing this signal back High when the transfer has completed. This output is a conditioned multiplexed version of the individual PREADY* inputs.
PSLVERR	Out	APB slave error signal for APB3 master. This signal is normally Low; however, a selected slave can drive this signal High indicating an error during the APB transfer. This output is a conditioned multiplexed version of the individual PSLVERRS* inputs.
APB3 Slave Signals		
PADDRS[23:0]	Out	APB shared slave address bus. This port is used to address registers and message data from the APB3 master to APB3 slaves. This port is shared by slaves 1 to 15.
PADDRS0[w:0]	Out	APB slave 0 address bus. This port is used to address registers and message data from the APB3 master to APB3 slave 0. The 'w' in the port name represents the width of this port. If the parameter IADDR_ENABLE = 0, this port is 24 bits wide, i.e., w=23; if the parameter IADDR_ENABLE = 1, this port is 32 bits wide, i.e., w=31.
PWRITES	Out	APB shared write/read for APB3 slaves. If High, a write will occur when an APB transfer to the selected APB3 slave takes place. If Low, a read from the selected APB3 slave will take place. This signal is shared by all slaves.
PENABLES	Out	APB shared enable for APB3 slaves. If High, the APB3 master is enabling access to the selected APB3 slot (slave) during the second (or greater if PREADY is Low) cycles. This signal is shared by all slaves.
PWDATAS[31:0]	Out	APB shared write data from APB3 master to selected slave. This bus is shared by all slaves.
PSELS0	Out	APB select for APB3 slave 0. If High, the APB3 master is selecting APB3 slot 0 (slave 0).
...
PSELS15	Out	APB select for APB3 slave 15. If High, the APB3 master is selecting APB3 slot 15 (slave 15).
PRDATAS0[31:0]	In	APB read data from slave 0 to APB3 master.
...
PRDATAS15[31:0]	In	APB read data from slave 15 to APB3 master.
PREADYS0	In	APB ready signal from slave 0 to APB3 master.
...
PREADYS15	In	APB ready signal from slave 15 to APB3 master.

Port Name	Type	Description
PSLVERRS0	In	APB slave error signal from slave 0 to APB3 master.
...
PSLVERRS15	In	APB slave error signal from slave 15 to APB3 master.

Note: All signals are active High (logic 1) unless otherwise noted.

Memory Map (programmer's view)

The memory maps for the two IADDR_ENABLE parameter settings are shown in Table 6 and Table 7. The RANGESIZE parameter affects the accessible address range for the APB peripherals.

If used with a peripheral that requires a full 32-bit PADDR memory space, IADDR_ENABLE = 1 should be selected; in this case, the full 32-bit address APB slave interface should be connected to the slot 0 mirrored slave interface of CoreAPB3, so that an APB3 master can access the large memory space peripheral by setting up the indirect address register within slot 1 of CoreAPB3.

Memory Map (IADDR_ENABLE=0)

Table 6 Memory Map (IADDR_ENABLE=0)

Resource	Address Range
Slave0	0 x RANGESIZE to (1 x RANGESIZE) - 1
Slave1	1 x RANGESIZE to (2 x RANGESIZE) - 1
Slave2	2 x RANGESIZE to (3 x RANGESIZE) - 1
Slave3	3 x RANGESIZE to (4 x RANGESIZE) - 1
Slave4	4 x RANGESIZE to (5 x RANGESIZE) - 1
Slave5	5 x RANGESIZE to (6 x RANGESIZE) - 1
Slave6	6 x RANGESIZE to (7 x RANGESIZE) - 1
Slave7	7 x RANGESIZE to (8 x RANGESIZE) - 1
Slave8	8 x RANGESIZE to (9 x RANGESIZE) - 1
Slave9	9 x RANGESIZE to (10 x RANGESIZE) - 1
Slave10	10 x RANGESIZE to (11 x RANGESIZE) - 1
Slave11	11 x RANGESIZE to (12 x RANGESIZE) - 1
Slave12	12 x RANGESIZE to (13 x RANGESIZE) - 1
Slave13	13 x RANGESIZE to (14 x RANGESIZE) - 1
Slave14	14 x RANGESIZE to (15 x RANGESIZE) - 1
Slave15	15 x RANGESIZE to (16 x RANGESIZE) - 1

Memory Map (IADDR_ENABLE=1)

Table 7 Memory Map (IADDR_ENABLE=1)

Resource	Address Range
Slave0	0 x RANGESIZE to (1 x RANGESIZE) -1
Indirect Addressing registers (uses Slave 1 address space)	
IADDR[7:0]	(1 x RANGESIZE) + 0

Resource	Address Range
IADDR[15:8] IADDR[23:16] IADDR[31:24]	(1 x RANGESIZE) + 4 (1 x RANGESIZE) + 8 (1 x RANGESIZE) + 12 The IADDR[31:0] registers default to zero (logic 0) on reset. Note that addresses from (1 x RANGESIZE) + 13 to (2 x RANGESIZE) - 1 are reserved (unused).
Slave2	2 x RANGESIZE to (3 x RANGESIZE) - 1
Slave3	3 x RANGESIZE to (4 x RANGESIZE) - 1
Slave4	4 x RANGESIZE to (5 x RANGESIZE) - 1
Slave5	5 x RANGESIZE to (6 x RANGESIZE) - 1
Slave6	6 x RANGESIZE to (7 x RANGESIZE) - 1
Slave7	7 x RANGESIZE to (8 x RANGESIZE) - 1
Slave8	8 x RANGESIZE to (9 x RANGESIZE) - 1
Slave9	9 x RANGESIZE to (10 x RANGESIZE) - 1
Slave10	10 x RANGESIZE to (11 x RANGESIZE) - 1
Slave11	11 x RANGESIZE to (12 x RANGESIZE) - 1
Slave12	12 x RANGESIZE to (13 x RANGESIZE) - 1
Slave13	13 x RANGESIZE to (14 x RANGESIZE) - 1
Slave14	14 x RANGESIZE to (15 x RANGESIZE) - 1
Slave15	15 x RANGESIZE to (16 x RANGESIZE) - 1

Indirect Addressing Operation

When the IADDR_ENABLE parameter is set to 1, indirect addressing is supported for slave 0. The effective size of the IADDR register is equal to $32 - \log_2(\text{RANGESIZE})$. The lower $\log_2(\text{RANGESIZE})$ bits of the IADDR register are unused and are read-only, returning only zeroes when read; data written to the lower $\log_2(\text{RANGESIZE})$ bits of the IADDR register are ignored.

In order to set the address for slave 0, the APB master needs to write to the IADDR register bits. The manner in which data is written to the IADDR register depends on the data width of the APB master, which must be configured by setting the APB_DWIDTH parameter to 8, 16, or 32.

If the APB_DWIDTH parameter is:

- 8-bits wide, each byte of the IADDR register must be set separately by writing to addresses (1 x RANGESIZE) + 0, (1 x RANGESIZE) + 4, (1 x RANGESIZE) + 8, and (1 x RANGESIZE) + 12, to set IADDR[7:0], IADDR[15:8], IADDR[23:16], and IADDR[31:24], respectively.
- 16-bits wide, each halfword of the IADDR register must be set separately by writing to addresses (1 x RANGESIZE) + 0 and (1 x RANGESIZE) + 4 to set IADDR[15:0] and IADDR[31:16], respectively.
- 32-bits wide, the whole word of the IADDR register must be set by writing to address (1 x RANGESIZE) + 0 to set IADDR[31:0].

Indirect Addressing Example 1

Consider a system with an 8-bit wide APB master (APB_DWIDTH = 8) that has peripherals using a slot size of 1024 locations (RANGESIZE = 1024) that is to write to a peripheral at address 0x40006000. Here are the steps that the APB master would take:

1. Write 0x60 to address $(1 \times \text{RANGESIZE}) + 4 = 1028$ to set $\text{IADDR}[15:8] = 0x60$. Writes to $\text{IADDR}[9:8]$ or to $\text{IADDR}[7:0]$ will be ignored since the lower $\log_2(\text{RANGESIZE}) = 10$ bits of the IADDR register are ignored.
2. Write 0x00 to address $(1 \times \text{RANGESIZE}) + 8 = 1032$ to set $\text{IADDR}[23:16] = 0x00$.
3. Write 0x40 to address $(1 \times \text{RANGESIZE}) + 12 = 1036$ to set $\text{IADDR}[31:24] = 0x40$.
4. Write data byte to $\text{PADDR}[11:0] = 0x000$ (slave 0 address 0x00), which CoreAPB3 passes to $\text{PADDRS0}[31:0] = (\text{IADDR}[31:8]$ concatenated with $\text{PADDR}[7:0]) = 0x40006000$.

Indirect Addressing Example 2

Consider a system with a 16-bit wide APB master ($\text{APB_DWIDTH} = 16$) that has peripherals using a slot size of 4096 locations ($\text{RANGESIZE} = 4096$) that is to write to a peripheral at address 0x12345678. Here are the steps that the APB master would take:

1. Write 0x5000 to address $(1 \times \text{RANGESIZE}) + 0 = 4096$ to set $\text{IADDR}[15:12] = 0x5$. Writes to $\text{IADDR}[11:0]$ will be ignored since the lower $\log_2(\text{RANGESIZE}) = 12$ bits of the IADDR register are ignored.
2. Write 0x1234 to address $(1 \times \text{RANGESIZE}) + 4 = 4100$ to set $\text{IADDR}[31:16] = 0x1234$.
3. Write data half-word to $\text{PADDR}[15:0] = 0x0678$ (slave 0 address 0x678), which CoreAPB3 passes to $\text{PADDRS0}[31:0] = (\text{IADDR}[31:12]$ concatenated with $\text{PADDR}[11:0]) = 0x12345678$.

Indirect Addressing Example 3

Consider a system with a 32-bit wide APB master ($\text{APB_DWIDTH}=32$) that has peripherals using a slot size of 65536 locations ($\text{RANGESIZE}=65536$) that is to write to a peripheral at address 0x64208000. Here are the steps that the APB master would take:

1. Write 0x64200000 to address $(1 \times \text{RANGESIZE}) + 0 = 65536$ to set $\text{IADDR}[31:16] = 0x6420$. Note that writes to $\text{IADDR}[15:0]$ will be ignored since the lower $\log_2(\text{RANGESIZE}) = 16$ bits of the IADDR register are ignored.
2. Write data word to $\text{PADDR}[19:0] = 0x08000$ (slave 0 address 0x8000), which CoreAPB3 passes to $\text{PADDRS0}[31:0] = (\text{IADDR}[31:16]$ concatenated with $\text{PADDR}[15:0]) = 0x64208000$.

Design Details

Functional Blocks

CoreAPB3, shown in Figure 1, consists of decoding logic used to determine which slave the master wants to access, optional indirect address register logic to allow connecting a full 32-bit addressable APB peripheral to slot 0, and multiplexing logic to send multiplexed versions of the peripheral read data busses, PREADY signals, and PSLVERR signals to the master.

APB Interface Timing

CoreAPB3 implements a master interface and slave interfaces that are compliant with the AMBA3 APB specification:

<http://www.arm.com/products/solutions/AMBAHomePage.html>

Tool Flows

Licensing

CoreAPB3 is licensed in two ways. Depending on your license type, tool flow functionality may be limited.

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 LiberoIDE. The RTL code for the core is obfuscated² and some of the testbench source files are not provided; they are precompiled into the compiled simulation library instead.

RTL

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

SmartDesign

CoreAPB3 is preinstalled in the SmartDesign IP Deployment design environment.

The core should 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 IDE User's Guide](#).

² Obfuscated means the RTL source files have had formatting and comments removed, and all instance and net names have been replaced with random character sequences.

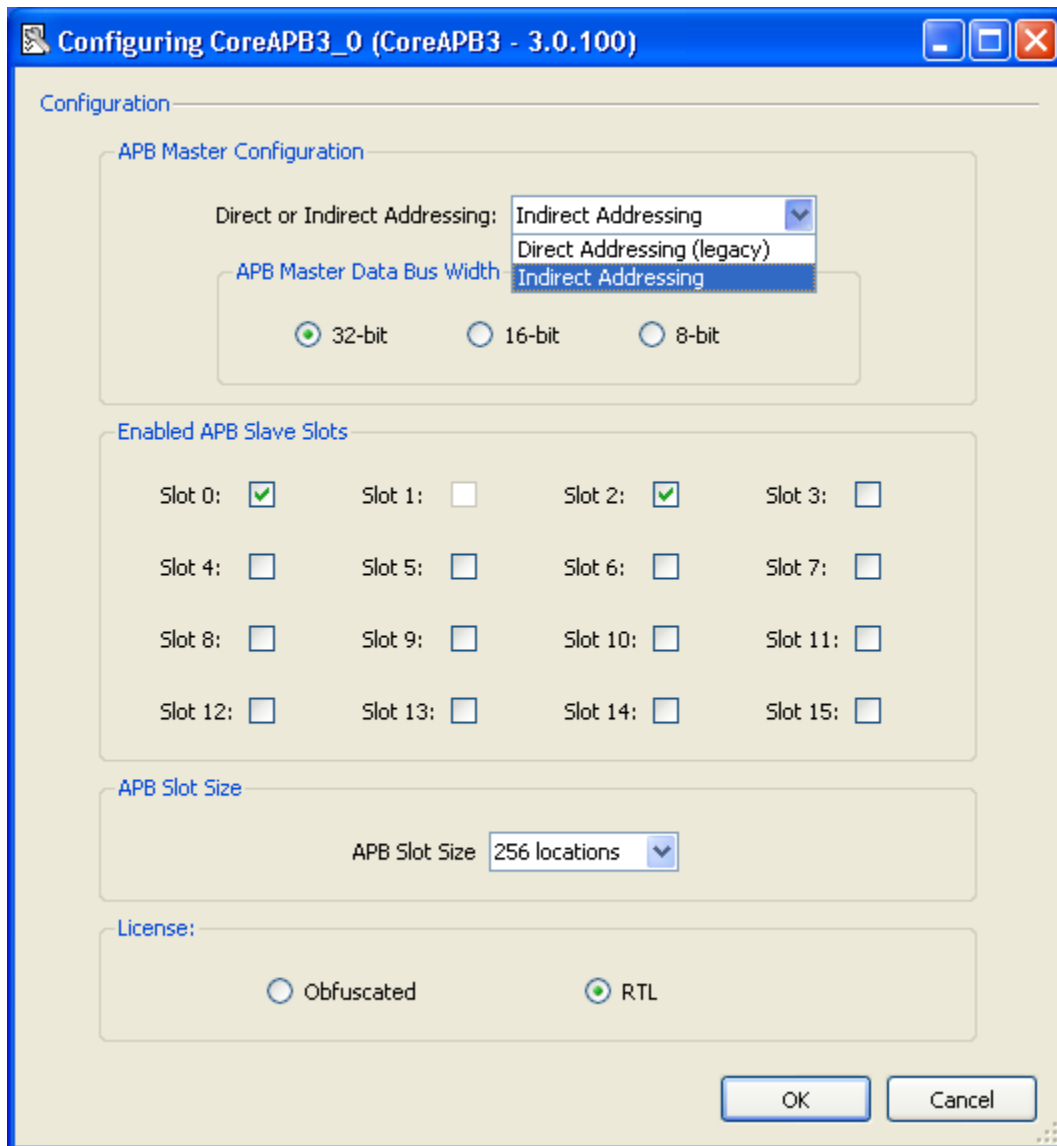


Figure 4 CoreAPB3 SmartDesign Configuration GUI

Simulation Flows

The user testbench for CoreAPB3 is included in all releases.

To run simulations, select the **User Testbench** flow within the SmartDesign CoreAPB3 configuration GUI, right click the canvas, and select **Generate Design**.

When SmartDesign generates the design files, it will install the user testbench files.

To run the user testbench, set the design root to the CoreAPB3 instantiation in the Libero IDE design hierarchy pane and click the **Simulation** icon in the Libero IDE Design Flow window. This will invoke ModelSim® and automatically run the simulation.

User Testbench

CoreAPB3's user testbench gives an example of how to use the core with a master and several slaves. As shown in Figure 5, the testbench instantiates a behavioral Actel DirectCore AMBA bus functional model (BFM) module to emulate using an APB3 master and several APB slaves.

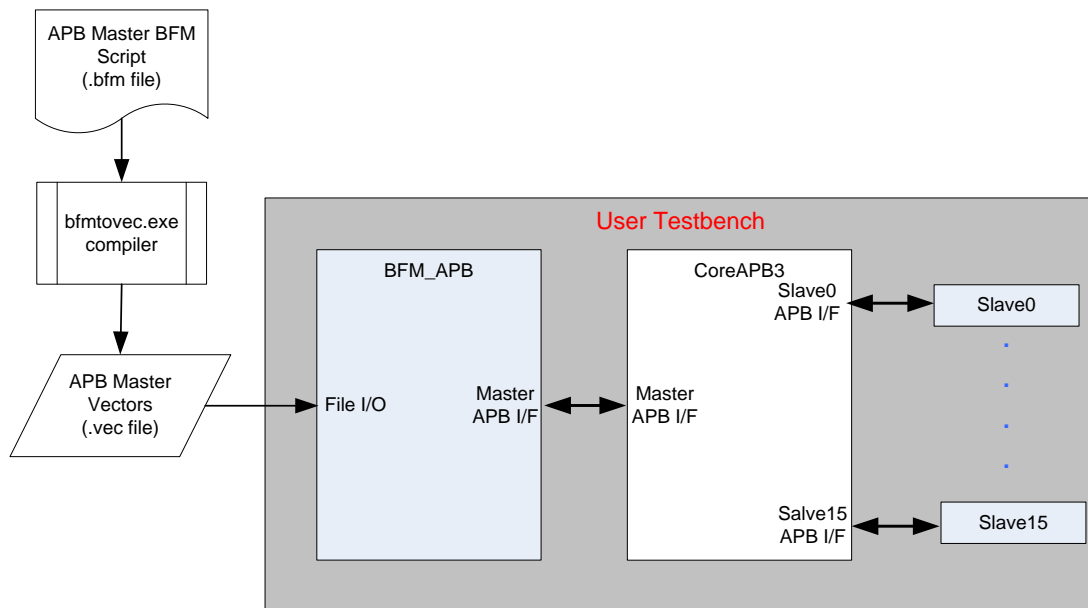


Figure 5 CoreAPB3 User Testbench

A BFM ASCII script source file (ending with the *.bfm suffix), with comments, is included with the user testbench. You can find the script source files in the following directory:

YourLiberoProjectDirectory/COREAPB3/mti/scripts, where “*YourLiberoProjectDir*” represents the path to your Libero IDE project that uses CoreAPB3. The BFM source file is for controlling the APB3 master and is named *coreapb3_usertb_apb_master.bfm*. This BFM source file is automatically recompiled each time the simulation is invoked from Libero IDE by the *bfmtovec.exe* executable, if running on a Windows® platform, or by the *bfmtovec.lin* executable, if running on a Linux platform. The output *.vec vector file, created by the *bfmtovec.exe* executable, is read in by the BFM modules for simulation in ModelSim.

You can alter the BFM script. Refer to the [Actel DirectCore AMBA BFM User's Guide](#) for more information.

The source code for the user testbench and BFM script is available with the CoreAPB3 Obfuscated and RTL releases. A compiled ModelSim simulation library containing the BFM modules is available with the CoreAPB3 Obfuscated release. An obfuscated RTL version of the BFM modules is available with the CoreAPB3 RTL release.

Synthesis in Libero IDE

After setting the design root appropriately for your design, click the **Synthesis** icon in Libero IDE. The Synthesis window appears, displaying the Synplify® project. Set Synplify to use the Verilog 2001 standard if Verilog is being used. To run Synthesis, click the **Run** icon.

Place-and-Route in Libero IDE

After setting the design root appropriately for your design, and after running Synthesis, click the **Layout** icon in the Libero IDE to invoke Designer. CoreAPB3 requires no special place-and-route settings.

System Operation

This chapter describes how to implement and integrate CoreAPB3 into your own design.

Usage with Core8051s

CoreAPB3 can be used with an APB3 master such as Core8051s. Figure 6 shows an example system using SmartDesign. Refer to the SmartDesign documentation on how to create your Core8051s-based design.

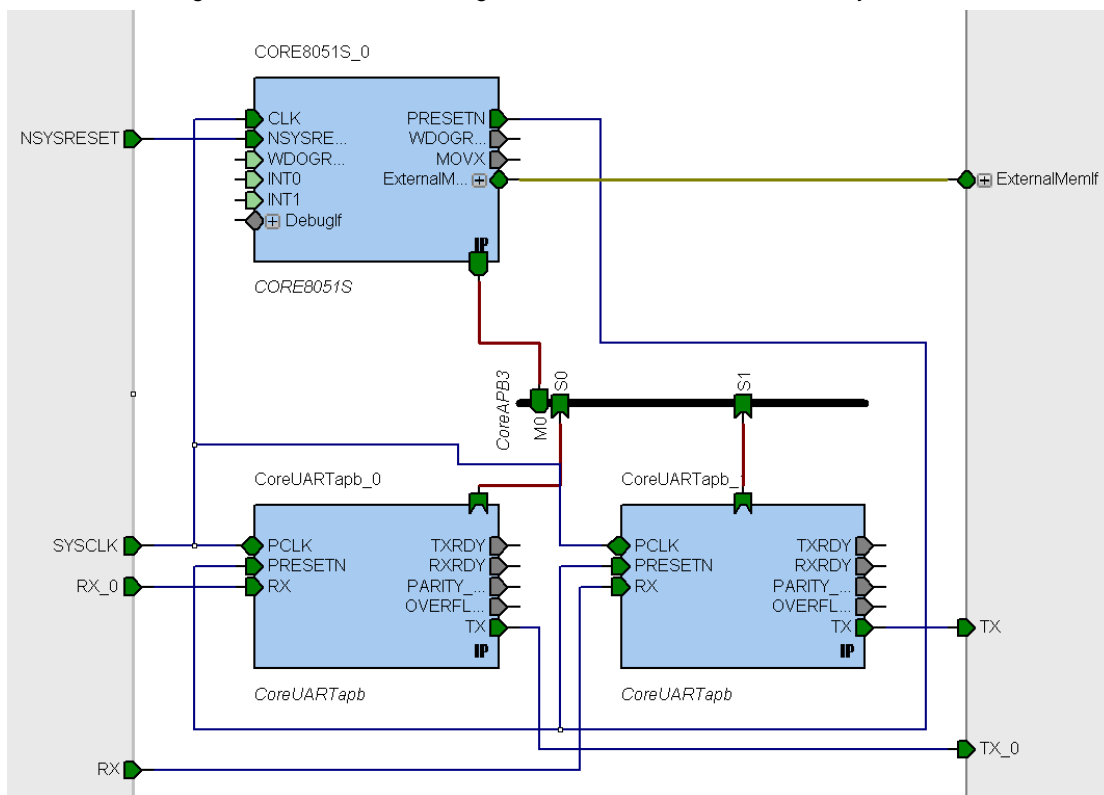


Figure 6 Example System using Core8051s and CoreAPB3

Usage with CoreABC

CoreAPB3 can be used with an APB3 master such as CoreABC. Figure 7 shows an example system using SmartDesign. Refer to the SmartDesign documentation on how to create your CoreABC-based design.

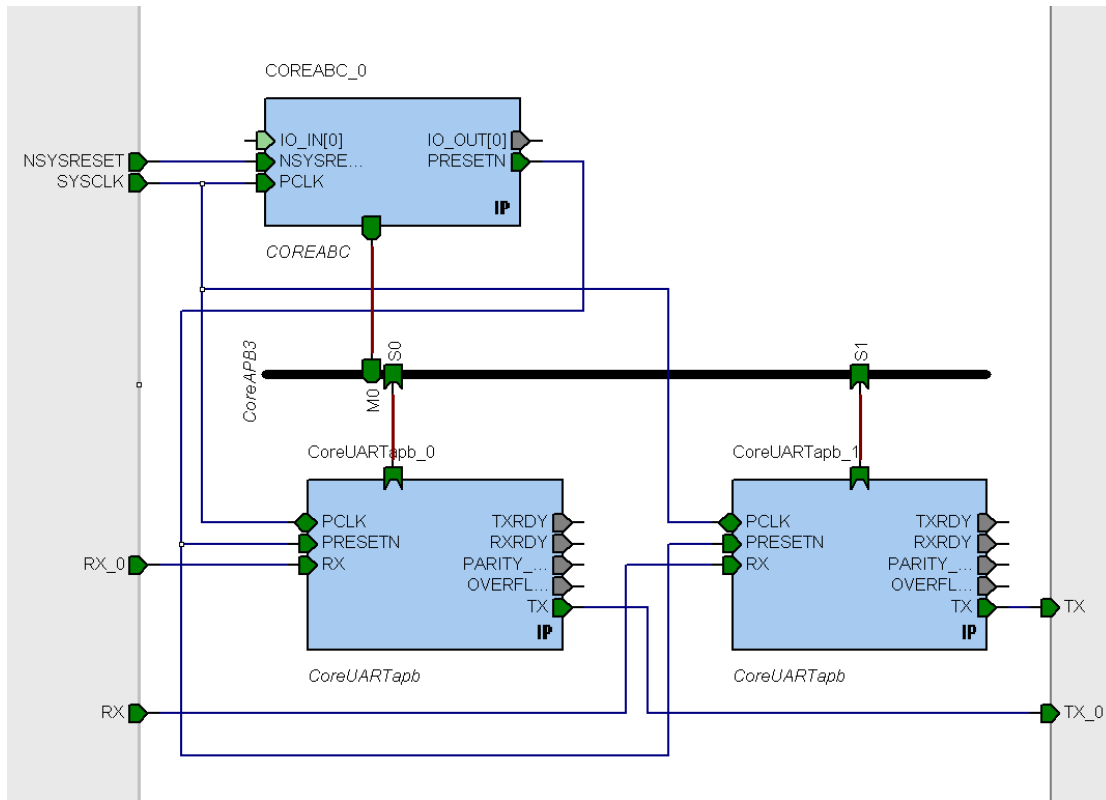


Figure 7 Example System using CoreABC and CoreAPB3

Ordering Information

Ordering Codes

CoreAPB3 can be ordered through your local Actel sales representative. It should be ordered using the following number scheme: CoreAPB3-XX, where XX is listed in Table 8.

Table 8 Ordering Codes

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

Note: CoreAPB3-OM is included free with a Libero IDE license.

Product Support

Actel backs its products with various support services including Customer Service, a Customer Technical Support Center, a web site, an FTP site, electronic mail, and worldwide sales offices. This appendix contains information about contacting Actel 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 Northeast and North Central U.S.A., call **650.318.4480**

From Southeast and Southwest U.S.A., call **650. 318.4480**

From South Central U.S.A., call **650.318.4434**

From Northwest U.S.A., call **650.318.4434**

From Canada, call **650.318.4480**

From Europe, call **650.318.4252** or **+44 (0) 1276 401 500**

From Japan, call **650.318.4743**

From the rest of the world, call **650.318.4743**

Fax, from anywhere in the world **650. 318.8044**

Actel Customer Technical Support Center

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

Actel Technical Support

Visit the [Actel Customer Support website](http://www.actel.com/support/search/default.aspx)

(<http://www.actel.com/support/search/default.aspx>) for more information and support.

Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the Actel web site.

Website

You can browse a variety of technical and non-technical information on Actel's [home page](http://www.actel.com/), at <http://www.actel.com/>.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. Several ways of contacting the Center follow:

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 tech@actel.com.

Phone

Our Technical Support Center answers all calls. The center retrieves information, such as your name, company name, phone number and your question, and then issues a case number. The Center then forwards the information to a queue where the first available application engineer receives the data and returns your call. The phone hours are from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. The Technical Support numbers are:

650.318.4460

800.262.1060

Customers needing assistance outside the US time zones can either contact technical support via email (tech@actel.com) or contact a local sales office. [Sales office listings](#) can be found at www.actel.com/company/contact/default.aspx.



Actel is the leader in low-power FPGAs and mixed-signal FPGAs and offers the most comprehensive portfolio of system and power management solutions. Power Matters. Learn more at <http://www.actel.com> .

Actel Corporation • 2061 Stierlin Court • Mountain View, CA 94043 • USA

Phone 650.318.4200 • Fax 650.318.4600 • Customer Service: 650.318.1010 • Customer Applications Center: 800.262.1060

Actel Europe Ltd. • River Court, Meadows Business Park • Station Approach, Blackwater • Camberley Surrey GU17 9AB • United Kingdom

Phone +44 (0) 1276 609 300 • Fax +44 (0) 1276 607 540

Actel Japan • EXOS Ebisu Building 4F • 1-24-14 Ebisu Shibuya-ku • Tokyo 150 • Japan

Phone +81.03.3445.7671 • Fax +81.03.3445.7668 • <http://jp.actel.com>

Actel Hong Kong • Room 2107, China Resources Building • 26 Harbour Road • Wanchai • Hong Kong

Phone +852 2185 6460 • Fax +852 2185 6488 • www.actel.com.cn