LogiCORE IP Mutex v2.0

Product Guide for Vivado Design Suite

PG117 March 20, 2013

XILINX®

XILINX®

Table of Contents

Appendix A: Migrating

Appendix B: Debugging

Appendix C: Application Software Development

Appendix D: Additional Resources

Introduction

In a multi-processor environment, the processors share common resources. The Mutex core provides a mechanism for mutual exclusion to enable one process to gain exclusive access to a particular resource.

The Mutex core contains a configurable number of mutexes. Each of these can be associated with a 32-bit user configuration register to store arbitrary data.

Features

- Supports AXI4-Lite
- Configurable number of AXI4-Lite interfaces from 0 to 8
- Configurable asynchronous or synchronous interface operation
- Configurable USER register
- Configurable number of mutexes
- Configurable CPUID width
- Configurable enhanced security through hardware identification support

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	Zynq™-7000, Kintex™-7, Virtex®-7, Artix™-7
Supported User Interfaces	AXI4-Lite
Resources	See Table 2-3.
Provided with Core	
Design Files	Vivado: RTL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	N/A
Simulation Model	VHDL Behavioral
Supported S/W Driver(2)	Standalone
Tested Design Flows	
Design Entry	Vivado™ Design Suite
Simulation	Mentor Graphics Questa® SIM Vivado Simulator
Synthesis	Vivado Synthesis
Support	
Provided by Xilinx @ www.xilinx.com/support	

Notes:

1. For a complete listing of supported devices, see the Vivado IP Catalog.
2. Standalone driver details can be found in the SDK directory (<*install_directory*>/doc/usenglish/xilinx_drivers.htm). Linux OS and driver support information is available from //wiki.xilinx.com.
3. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

The Mutex core contains a configurable number of mutexes. Each mutex can be associated with a 32-bit user configuration register to store arbitrary data.

In a multi-processor environment, the processors share common resources. The mutex provides a mechanism for mutual exclusion to enable one process to gain exclusive access to a particular resource.

The Mutex in a typical AXI4-Lite system is shown in the top-level block diagram in Figure 1-1.

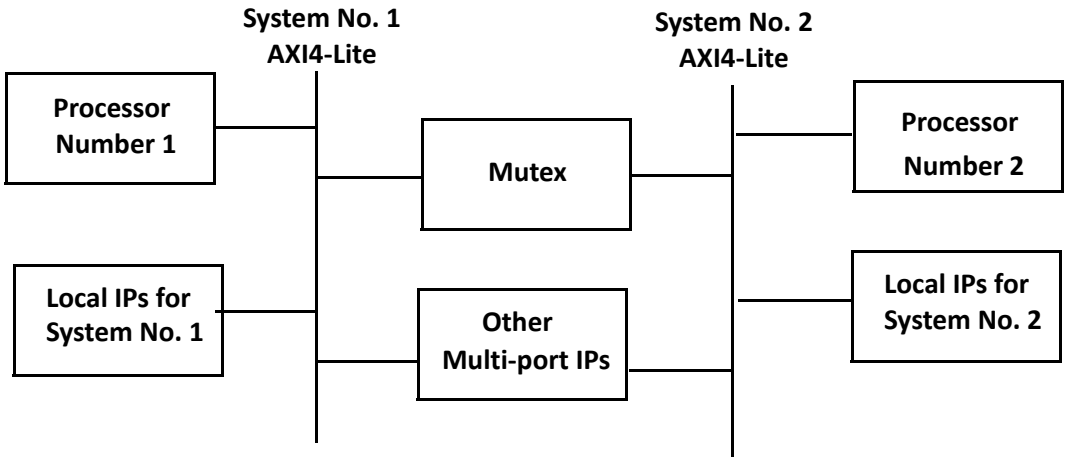

Figure 1-1: **Mutex in an AXI4-Lite System**

Feature Summary

Bus Interfaces

The Mutex has two bus interfaces to access the internal resources, usually connected to different processors in a multi-processor system.

Registers

The Mutex provides several types of registers, available with AXI4-Lite interfaces, to lock and release the mutex, and provide a user configuration register:

- Mutex registers, which provides the possibility to lock and release the mutex.
- User configuration.

Protection

- The Mutex provides hardware tamper-proof protection of mutex access, preventing any processor except the intended one from modifying a mutex.

Licensing and Ordering Information

This Xilinx® LogiCORE™ IP module is provided at no additional cost with the Xilinx Vivado™ Design Suite tools under the terms of the Xilinx End User License. Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Product Specification

Standards

The Mutex adheres to the AXI4-Lite standard defined in the ARM® *AMBA® AXI and ACE Protocol Specification* [Ref 1].

Performance

The frequency and latency of the Mutex are optimized for use with MicroBlaze™. This means that the frequency targets are aligned to MicroBlaze targets.

Maximum Frequencies

Table 2-1 lists clock frequencies for the target families. The maximum achievable clock frequency can vary. The maximum achievable clock frequency and all resource counts can be affected by the tool flow, other tool options, additional logic in the FPGA, different versions of the Xilinx tools, and other factors.

Table 2-1: **Maximum Frequencies**

Architecture	Speed grade	Max Frequency
Artix™-7	-3	225
Kintex™-7	-3	320
Virtex-7	-3	320

Latency and Throughput

The latency and throughput of accesses to the Mutex depends on the bus interface. The latency for each interface when reading or writing, as well as the throughput, is shown in Table 2-2, according to the parameter settings affecting the measurements.

Table 2-2: **Latency and Throughput**

Bus Interface	Read Latency (clock cycles)	Write Latency (clock cycles)	Throughput (clock cycles/word)
Synchronous (C_ASYNC_CLKS = 0)			
AXI4-Lite	3	3	6
Asynchronous (C_ASYNC_CLKS = 1)			
AXI4-Lite	3	3	10

Resource Utilization

Because the Mutex core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When the Mutex core is combined with other designs in the system, the utilization of FPGA resources and timing of the Mutex design will vary from the results reported here. These values are generated from a minimal MicroBlaze system with the UART Lite and Mutex as the only peripherals.

The Mutex resource utilization for various parameter combinations measured with the Kintex-7 FPGA as the target device are detailed in Table 2-3.

Table 2-3: **xc7k325t-2-ffg900 FPGA Resource Estimates**

Parameter Values (other parameters at default value)						Device Resources			Performance
C_ASYNC_CLKS	C_NUM_PLB	C_NUM_AXI	C_ENABLE_USER	C_ENABLE_HW_PROT	C_NUM_MUTEX	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
0	1	0	0	0	1	21	34	46	333
0	1	0	0	1	1	21	35	42	337
0	1	0	1	1	1	30	84	81	333
0	1	0	0	0	16	21	30	52	333
0	1	0	0	1	16	18	30	47	335
0	1	0	1	1	16	34	61	94	324
0	2	0	1	1	16	60	127	151	321
0	8	0	1	1	16	200	409	470	327
0	0	1	1	1	16	35	64	97	335
0	0	2	1	1	16	65	128	143	326
0	0	8	1	1	16	176	410	451	325
1	2	0	1	1	16	70	136	144	336
1	8	0	1	1	16	229	442	454	322
1	0	2	1	1	16	59	132	153	329
1	0	8	1	1	16	184	424	479	329

Port Descriptions

The Mutex supports AXI4-Lite interfaces, and the number of interfaces is independently configured from 0 to 8. All interfaces are individually configured and contain the signals listed in Table 2-4, where <x> denotes the interface number (0 to 7).

Table 2-4: **I/O Signal Description for AXI-4Lite Interface**

Port	Signal Name	Interface	I/O	Initial State	Description
System Signals					
P43	S<x>_AXI_ACLK	System	I	-	AXI Clock
P44	S<x>_AXI_ARESETN	System	I	-	AXI Reset, active-Low
AXI Write Address Channel Signals					
P45	S<x>_AXI_AWADDR[C_S <x>_AXI_ADDR_WIDTH-1:0]	AXI	I	-	AXI Write address. The write address bus gives the address of the write transaction.
P46	S<x>_AXI_AWVALID	AXI	I	-	Write address valid. This signal indicates that valid write address is available.
P47	S<x>_AXI_AWREADY	AXI	O	0	Write address ready. This signal indicates that the slave is ready to accept an address.
AXI Write Channel Signals					
P48	S<x>_AXI_WDATA[C_S<x>_ AXI_DATA_WIDTH - 1: 0]	AXI	I	-	Write data
P49	S<x>_AXI_WSTB[C_S<x>_ AXI_DATA_WIDTH/8-1:0][1]	AXI	I	-	Write strobes. This signal indicates which byte lanes to update in memory(1)
P50	S<x>_AXI_WVALID	AXI	I	-	Write valid. This signal indicates that valid write data and strobes are available.
P51	S<x>_AXI_WREADY	AXI	O	0	Write ready. This signal indicates that the slave can accept the write data.
AXI Write Response Channel Signals					
P52	S<x>_AXI_BRESP[1:0]	AXI	O	0x0	Write response. This signal indicates the status of the write transaction. 00 - OKAY 10 - SLVERR 11 - DECERR
P53	S<x>_AXI_BVALID	AXI	O	0	Write response valid. This signal indicates that a valid write response is available.

Table 2-4: **I/O Signal Description for AXI-4Lite Interface *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P54	S<x>_AXI_BREADY	AXI	I	-	Response ready. This signal indicates that the master can accept the response information.
AXI Read Address Channel Signals					
P55	S<x>_AXI_ARADDR[C_S<x>_AXI_ADDR_WIDTH-1:0]	AXI	I	-	Read address. The read address bus gives the address of a read transaction.
P56	S<x>_AXI_ARVALID	AXI	I	-	Read address valid. This signal indicates, when HIGH, that the read address is valid and will remain stable until the address acknowledge signal, S<x>_AXI_ARREADY, is High.
P57	S<x>_AXI_ARREADY	AXI	O	1	Read address ready. This signal indicates that the slave is ready to accept an address.
AXI Read Data Channel Signals					
P58	S<x>_AXI_RDATA[C_S<x>_AXI_DATA_WIDTH -1:0]	AXI	O	0x0	Read data
P59	S<x>_AXI_RRESP[1:0]	AXI	O	0x0	Read response. This signal indicates the status of the read transfer. 00 - OKAY 10 - SLVERR 11 - DECERR
P60	S<x>_AXI_RVALID	AXI	O	0	Read valid. This signal indicates that the required read data is available and the read transfer can complete
P61	S<x>_AXI_RREADY	AXI	I	-	Read ready. This signal indicates that the master can accept the read data and response information

1. This signal is not used. The Mutex assumes that all byte lanes are active.

Register Space

Each interface of the Mutex core can access all mutexes. Only one interface at the time can access any of the mutexes. For example, while one interface is accessing any of the mutexes, all other AXI interfaces are blocked. Interface arbitration has fixed priority for AXI 0-7 in descending order. For example, S0_AXI has the highest priority, and S7_AXI the lowest. Table 2-5 shows all the Mutex registers and their addresses offsets.

Table 2-5: **Mutex Registers**

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
BASEADDR + 0x0	MUTEX	R/W	0	Mutex register for mutex ownership
BASEADDR + 0x4	USER	N/A	0	USER configuration register.
BASEADDR + 0x8 to 0xFC	Reserved			Reserved for future use

Mutex Register (MUTEX)

The MUTEX register contains one mandatory and two optional bit fields. The LOCK bit is required because this bit determines if the mutex is in the locked or released state. CPUID is usually included to control access of who may manipulate the mutex. It is only the owner of the mutex that may release it. For extra safety, an optional HWID field is also available. The HWID bits are not user-accessible and are handled implicitly in the background. HWID contains which port the AXI master is attached. This guarantees that no other processor can fake the CPUID and gain access over the mutex. Bit assignment in the MUTEX register is described in Table 2-7.

CPUID is a unique identification value assigned by the tools to software that executes on each processor. Because CPUID is only assigned to software created from within SDK, any other master that accesses the mutex must be manually assigned a unique number that does not interfere with the others. Examples of this are external processors and hardware IPs other than the MicroBlaze™ processor. Each processor has its allocated CPUID listed in xparameters.h.

Mutex lock and release process

The steps needed to lock and release a mutex (for a free mutex, the MUTEX register is zero):

- Write <CPUID & 1> to the MUTEX register. If the mutex is free, the lock bit will be set to one and the CPUID field will be update with the new CPUID. If C_ENABLE_HW_PROT is enabled, the HWID is also stored for enhanced protection. Should the mutex already be locked, the access is ignored.
- Read back the MUTEX register to verify that the mutex has been locked by the current CPU by comparing the value with the written CPUID. If not, retry step 1 until ownership has been granted.
- Manipulate the shared resource that is protected by the mutex.
- Release the mutex by writing <CPUID & 0> to the mutex register. If C_ENABLE_HW_PROT is enabled, the HWID is also taken into account. The mutex will automatically set the MUTEX register to zero.

If the 'wrong' processor attempts to free the mutex with C_ENABLE_HW_PROT active and with the correct CPUID the operation is ignored because both the HWID and CPUID must

match for the operation to be successful. Also, the operation is ignored if the 'right' processor writes the wrong CPUID.

Table 2-6: **Write Data Register**

Reserved		CPUID		Lock
0	22	23	30	31

Table 2-7: **Write Data Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 22	Reserved	N/A	0	Reserved for future use.
23 - 30	CPUID	R/W	-	Unique processor ID number.
31	LOCK	R/W	-	Lock status: 0 = free, 1 = Mutex currently owned by CPUID.

Mutex User Configuration Register (USER)

The USER configuration is used to store a 32-bit value associated with a mutex. It can contain any arbitrary information. Bit assignment in the USER register is described in Table 2-9.

Table 2-8: **User Configuration Register (USER)**

USER	
0	31

Table 2-9: **Mutex Read Data Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 31	USER	R/W	-	User configuration register

Designing with the Core

General Design Guidelines

This chapter includes guidelines and additional information to facilitate designing with the core.

Clocking

The `Sn_AXI_ACLK` (n = 0 - 7) input should normally be connected to the same clock as the interconnect.

With synchronous operation (`C_ASYNC_CLKS` = 0), the clock inputs used must all be connected to the same clock signal in the above case.

Resets

The `Sn_AXI_ARESETN` (n = 0 - 7) input should normally be connected to the same reset as the interconnect.

Protocol Description

See the AMBA® AXI4 Interface standard for a description of the AXI4-Lite protocol (ARM® *AMBA AXI and ACE Protocol Specification, Version 2.0 ARM IHI 0022D* [Ref 1]).

Customizing and Generating the Core

This chapter includes information about using Xilinx tools to customize and generate the core in the Vivado™ Design Suite environment.

GUI

The Mutex parameters are divided in two categories: System and Mutex. When using Vivado™ IP integrator feature, the addresses are auto-generated.

The configuration screen is shown in Figure 4-1.

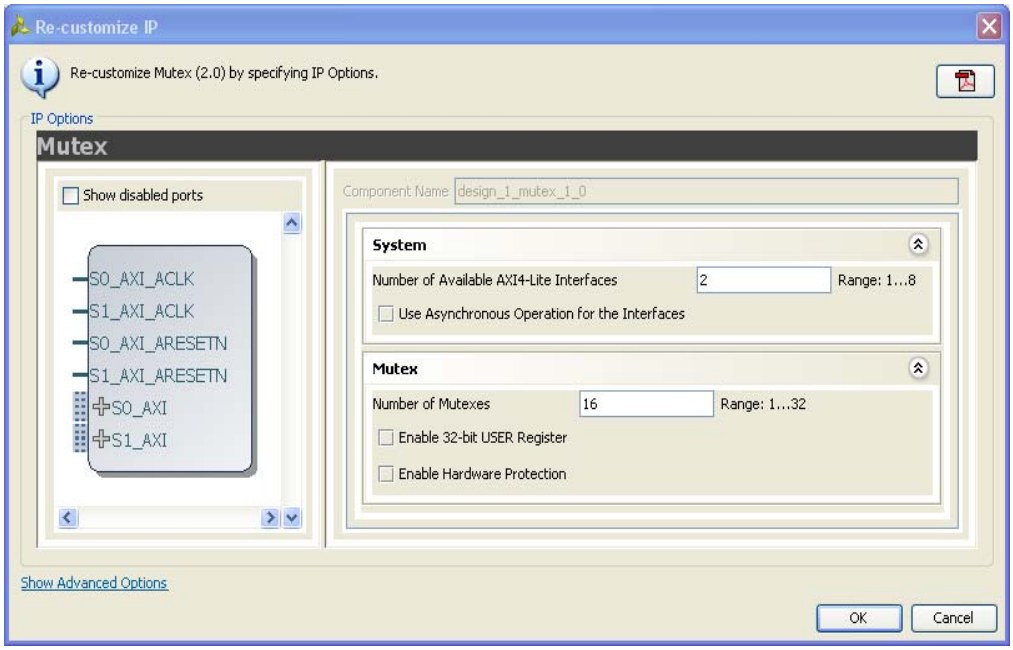

Figure 4-1: **Configuration Screen**

- **Number of Available AXI4-Lite Interfaces** - Sets the number of available bus interfaces, typically one interface per connected processor.
- **Use Asynchronous Operation for the Interfaces** - Enables asynchronous operation, when the clocks of the used interfaces are not identical.
- **Number of Mutexes** - Defines how many individual mutexes are available.

- **Enable 32-bit USER Register** - The USER register can be used to store arbitrary data. Usually it stores the address to the shared resource controlled by the mutex.
- **Enable Hardware Protection** - When hardware protection is enabled, HWID is used to complement the CPUID for enhanced security. The HWID consists of the AXI interface number and AXI transaction ID for the processor that has locked the mutex. The HWID is not user-accessible and is thus tamper proof.

GUI Parameters

The Mutex design is parameterized to tailor it for different systems. This allows you to configure a design that uses the resources required by the system only and that operates with the best possible performance. The features that can be parameterized in the Mutex design are shown in Table 4-1.

Table 4-1: **Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameter					
G1	Target FPGA family	C_FAMILY	Supported architectures	virtex7	string
Mutex Parameters					
G17	Specify if interfaces are synchronous or asynchronous	C_ASYNC_CLKS	0 - 1	0	Integer
G19	Number of AXI4-Lite interfaces	C_NUM_AXI	0 - 8	0	Integer
G20	If the 32-bit USER register associated with a mutex should be available	C_ENABLE_USER	0 - 1	32	Integer
G21	Number of bits used for the CPUID field	C_OWNER_ID_WIDTH	8	8	Integer
G22	If hardware protection of a mutex should be enabled besides the CPUID (if available)	C_ENABLE_HW_PROT	0 - 1	0	Integer
G23	Number of mutexes that are contained inside the core	C_NUM_MUTEX	1 - 32	16	Integer

Constraining the Core

This chapter contains information about constraining the core in the Vivado™ Design Suite environment.

Required Constraints

There are no required constraints for this core.

Device, Package, and Speed Grade Selections

There are no device, package or speed grade requirements for this core.

Clock Frequencies

There are no specific clock frequency requirements for this core.

Clock Management

The Mutex can either be fully synchronous with all clocked elements clocked by the same physical clock, or asynchronous with different clocks on the connected bus interfaces.

With an asynchronous configuration, the parameter C_ASYNC_CLKS (Use Asynchronous Operation for the Interfaces) must be set manually.

To operate properly when connected to MicroBlaze™, the corresponding bus interface clock must be the same as the MicroBlaze `Clk`.

Clock Placement

There are no specific clock placement requirements for this core.

Banking

There are no specific banking rules for this core.

Transceiver Placement

There are no transceiver placement requirements for this core.

I/O Standard and Placement

There are no specific I/O standards and placement requirements for this core.

Appendix A

Migrating

This appendix describes migrating from older versions of the IP to the current IP release.

For information on migrating to the Vivado™ Design Suite, see the *Vivado Design Suite Migration Methodology Guide* [Ref 2].

Debugging

This appendix includes details about resources available on the Xilinx Support website and debugging tools. In addition, this appendix provides a step-by-step debugging process to guide you through debugging the Mutex core.

The following topics are included in this appendix:

Finding Help on Xilinx.com

To help in the design and debug process when using the Mutex, the Xilinx Support web page (www.xilinx.com/support) contains key resources such as product documentation, release notes, answer records, information about known issues, and links for opening a Technical Support WebCase.

Documentation

This product guide is the main document associated with the Mutex. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page (www.xilinx.com/support) or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Design Tools tab on the Downloads page (www.xilinx.com/download). For more information about this tool and the features available, open the online help after installation.

Known Issues

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core are listed below, and can also be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use keywords such as

- Product name
- Tool message(s)
- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Master Answer Record for the Mutex

AR 54409

Contacting Technical Support

Xilinx provides technical support at www.xilinx.com/support for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support:

1. Navigate to www.xilinx.com/support.
2. Open a WebCase by selecting the WebCase link located under Support Quick Links.

When opening a WebCase, include:

- Target FPGA including package and speed grade.
- All applicable Xilinx Design Tools and simulator software versions.
- Additional files based on the specific issue might also be required. See the relevant sections in this debug guide for guidelines about which file(s) to include with the WebCase.

Debug Tools

The main tools available to address Mutex design issues are the Vivado Lab tools.

Vivado Lab Tools

Vivado™ inserts logic analyzer and virtual I/O cores directly into your design. Vivado Lab Tools allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature represents the functionality in the Vivado IDE that is used for logic debugging and validation of a design running in Xilinx FPGAs in hardware.

The Vivado logic analyzer is used to interact with the logic debug LogiCORE IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

Reference Boards

All 7-Series Xilinx development boards support Mutex. These boards can be used to prototype designs and establish that the core can communicate with the system.

Simulation Debug

The simulation debug flow for Questa SIM is described below. A similar approach can be used with other simulators.

- Check for the latest supported versions of Questa SIM in the Xilinx Design Tools: Release Notes Guide. Is this version being used? If not, update to this version.
- If using Verilog, do you have a mixed mode simulation license? If not, obtain a mixed-mode license.
- Ensure that the proper libraries are compiled and mapped. In the Vivado Design Suite **Flow > Simulation Settings** can be used to define the libraries.
- Have you associated the intended software program for the MicroBlaze™ processor with the simulation? Use the command **Tools > Associate ELF Files** in Vivado Design Suite.
- When observing the traffic on the interfaces connected to the Mutex, see the timing in the relevant specification:
  - For AXI4-Lite, see the *AMBA® AXI and ACE Protocol Specification* [Ref 1].

Hardware Debug

Hardware issues can range from link bring-up to problems seen after hours of testing. This section provides debug steps for common issues. The Vivado Lab Tools are valuable resources to use in hardware debug. The signal names mentioned in the following individual sections can be probed using the Vivado Lab Tools for debugging the specific problems.

Many of these common issues can also be applied to debugging design simulations.

General Checks

Ensure that all the timing constraints for the core were properly incorporated from the example design and that all constraints were met during implementation.

- Does it work in post-place and route timing simulation? If problems are seen in hardware but not in timing simulation, this could indicate a PCB issue. Ensure that all clock sources are active and clean.
- If using MMCMs in the design, ensure that all MMCMs have obtained lock by monitoring the `LOCKED` port.

Interface Debug

AXI4-Lite Interfaces

Read from a register that does not have all 0s as a default to verify that the interface is functional. Output `Sn_AXI_ARREADY` asserts when the read address is valid, and output `Sn_AXI_RVALID` asserts when the read data/response is valid, where *n* is the interface number (0 or 1). If the interface is unresponsive, ensure that the following conditions are met:

- The `Sn_AXI_ACLK` input is connected and toggling.
- The interface is not being held in reset, and `Sn_AXI_ARESETN` is an active-Low reset.
- The common core reset is not active, and `SYS_Rst` is an active-High reset.
- If the simulation has been run, verify in simulation and/or the Vivado Lab Tools debugging tool capture that the waveform is correct for accessing the AXI4-Lite interface.

XILINX®

Appendix C

Application Software Development

Device Drivers

The Mutex is supported by the mutex driver, included with Xilinx Software Development Kit.

Appendix D

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

www.xilinx.com/support.

For a glossary of technical terms used in Xilinx documentation, see:

www.xilinx.com/company/terms.htm.

References

These documents provide supplemental material useful with this product guide:

1. ARM® AMBA® AXI and ACE Protocol Specification (IHI 0022D)
2. Vivado Design Suite Migration Methodology Guide (UG911)
3. IBM CoreConnect128-Bit Processor Local Bus, Architectural Specification (v4.6)
4. Vivado™ Design Suite user documentation
5. Vivado Design Suite User Guide, Designing with IP (UG896)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/20/2013	1.0	Initial release as a Product Guide; replaces PG089. There are no documentation changes for this release.

Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at http://www.xilinx.com/warranty.htm; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: http://www.xilinx.com/warranty.htm#critapps.

© Copyright 2013 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. AMBA, AMBA Designer, ARM, ARM1176JZ-S, CoreSight, Cortex, and PrimeCell are trademarks of ARM in the EU and other countries. All other trademarks are the property of their respective owners.