

Mutex v2.1

LogiCORE IP Product Guide

Vivado Design Suite

PG117 (v2.1) November 21, 2019



Table of Contents

Chapter 1: Introduction	4
Features.....	4
IP Facts.....	5
Chapter 2: Overview	6
Feature Summary.....	6
Licensing and Ordering.....	7
Chapter 3: Product Specification	8
Standards.....	8
Performance.....	8
Resource Use.....	9
Port Descriptions.....	9
Register Space.....	11
Chapter 4: Designing with the Core	13
Clocking.....	13
Resets.....	13
Protocol Description.....	13
Chapter 5: Design Flow Steps	14
Customizing and Generating the Core.....	14
Constraining the Core.....	18
Simulation.....	19
Synthesis and Implementation.....	19
Appendix A: Upgrading	20
Migrating to the Vivado Design Suite	20
Upgrading in the Vivado Design Suite.....	20
Appendix B: Debugging	21
Finding Help on Xilinx.com.....	21

Debug Tools.....	22
Simulation Debug.....	23
Hardware Debug.....	24
Interface Debug.....	24
Appendix C: Application Software Development.....	25
Device Drivers.....	25
Appendix D: Additional Resources and Legal Notices.....	26
Xilinx Resources.....	26
Documentation Navigator and Design Hubs.....	26
References.....	26
Revision History.....	27
Please Read: Important Legal Notices.....	27

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

IP Facts

LogiCORE™ IP Facts Table	
Core Specifics	
Supported Device Family ¹	UltraScale+™, UltraScale Architecture, Zynq®-7000, 7 series
Supported User Interfaces	AXI4-Lite
Resources	Performance and Resource Use web page
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 Flows³	
Design Entry	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide .
Synthesis	Vivado Synthesis
Support	
Release Notes and Known Issues	Master Answer Record: 54409
All Vivado IP Change Logs	Master Vivado IP Change Logs: 72775
Xilinx Support web page	

Notes:

- For a complete list of supported devices, see the Vivado® IP catalog.
- Standalone driver details can be found in <Install Directory>/Vitis/<release>/data/embeddedsw/doc/xilinx_drivers.htm.
BareMetal driver support information is available from the [BareMetal Mutex Driver Page](#).
- For the supported versions of third-party tools, see the [Xilinx Design Tools: Release Notes Guide](#).

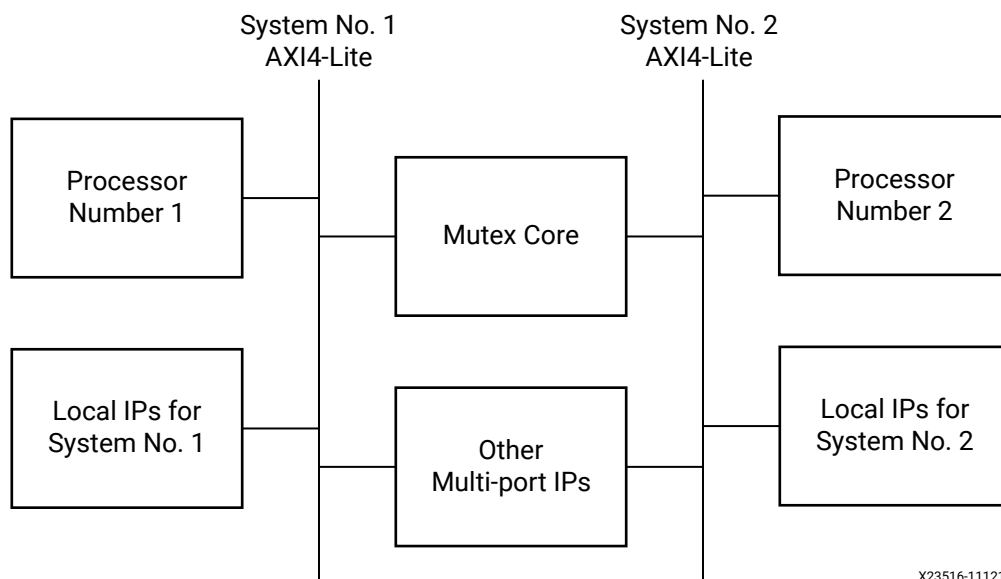
Overview

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 core in a typical AXI4-Lite system is shown in the top-level block diagram in the following figure.

Figure 1: Mutex Core in an AXI4-Lite System



X23516-111219

Feature Summary

Bus Interfaces

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

Registers

The Mutex core provides several types of registers, available with AXI4-Lite interfaces:

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

Protection

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

Licensing and Ordering

This Xilinx® LogiCORE™ IP module is provided at no additional cost with the Xilinx Vivado® Design Suite under the terms of the [Xilinx End User License](#).

Information about other Xilinx® LogiCORE™ IP modules is available at the [Xilinx Intellectual Property](#) page. For information about pricing and availability of other Xilinx® LogiCORE IP modules and tools, contact your [local Xilinx sales representative](#).

License Checkers

If the IP requires a license key, the key must be verified. The Vivado® design tools have several license checkpoints for gating licensed IP through the flow. If the license check succeeds, the IP can continue generation. Otherwise, generation halts with an error. License checkpoints are enforced by the following tools:

- Vivado Synthesis
- Vivado Implementation
- `write_bitstream` (Tcl command)



IMPORTANT! IP license level is ignored at checkpoints. The test confirms a valid license exists. It does not check IP license level.

Product Specification

Standards

This Mutex adheres to the AXI4-Lite standard defined in the Arm® AMBA® AXI and ACE Protocol Specification ([IH10022E](#)).

Performance

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

Maximum Frequencies

For details about performance, visit [Performance and Resource Utilization](#).

Latency and Throughput

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

Table 1: 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 Use

For full details about performance and resource use, visit the [Performance and Resource Use web page](#).

Port Descriptions

The Mutex core 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 the following table, where <x> denotes the interface number (0 to 7).

AXI4-Lite Interface Ports

Table 2: I/O Signal Description for AXI4-Lite 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.

Table 2: I/O Signal Description for AXI4-Lite Interface (cont'd)

Port	Signal Name	Interface	I/O	Initial State	Description
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.
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 remains 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.

Notes:

1. This signal is not used. The Mutex core 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. When configured with multiple mutexes each reserves a 256 byte address range, that is, registers for mutex #0 is located between 0x0 and 0xFF, mutex #1 between 0x100 and 0x1FF and so on. The following table shows all the Mutex core registers and their addresses offsets for each available Mutex.

Table 3: 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 can manipulate the mutex. It is only the owner of the mutex that can 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 5: Write Data Register Bit Definitions](#).

CPUID is a unique identification value assigned by the tools to the software that executes on each processor. Because the CPUID is only assigned to software created from within the Vitis™ unified software platform, 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 required to lock and release a mutex (for a free mutex, the MUTEX register is zero):

- Write $\langle \text{CPUID} \ \& \ 1 \rangle$ to the MUTEX register. If the mutex is free, the lock bit is set to one and the CPUID field is updated with the new CPUID. If `C_ENABLE_HW_PROT` is enabled, the HWID is also stored for enhanced protection. If the mutex is already 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 $\langle \text{CPUID} \ \& \ 0 \rangle$ to the MUTEX register. If `C_ENABLE_HW_PROT` is enabled, the HWID is also taken into account. The mutex automatically sets 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.

Reserved		CPUID		Lock
31	9	8	1	0

Table 5: Write Data Register Bit Definitions

Bit(s)	Name	Core Access	Reset Value	Description
31-9	Reserved	N/A	-	Reserved for future use.
8-1	CPUID	R/W	-	Unique processor ID number.
0	LOCK	R/W	0	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 the following table.

USER	
31	0

Table 7: Mutex Read Data Register Bit Definitions

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

Designing with the Core

This section 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 this same clock signal.

Resets

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

Protocol Description

See the [Arm[®] AMBA[®] AXI and ACE Protocol Specification \(IH10022E\)](#).

Design Flow Steps

This section describes customizing and generating the core, constraining the core, and the simulation, synthesis, and implementation steps that are specific to this IP core. More detailed information about the standard Vivado[®] design flows and the IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* ([UG994](#))
- *Vivado Design Suite User Guide: Designing with IP* ([UG896](#))
- *Vivado Design Suite User Guide: Getting Started* ([UG910](#))
- *Vivado Design Suite User Guide: Logic Simulation* ([UG900](#))

Customizing and Generating the Core

This section includes information about using Xilinx[®] tools to customize and generate the core in the Vivado[®] Design Suite.

If you are customizing and generating the core in the Vivado IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* ([UG994](#)) for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values do change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl console.

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* ([UG896](#)) and the *Vivado Design Suite User Guide: Getting Started* ([UG910](#)).

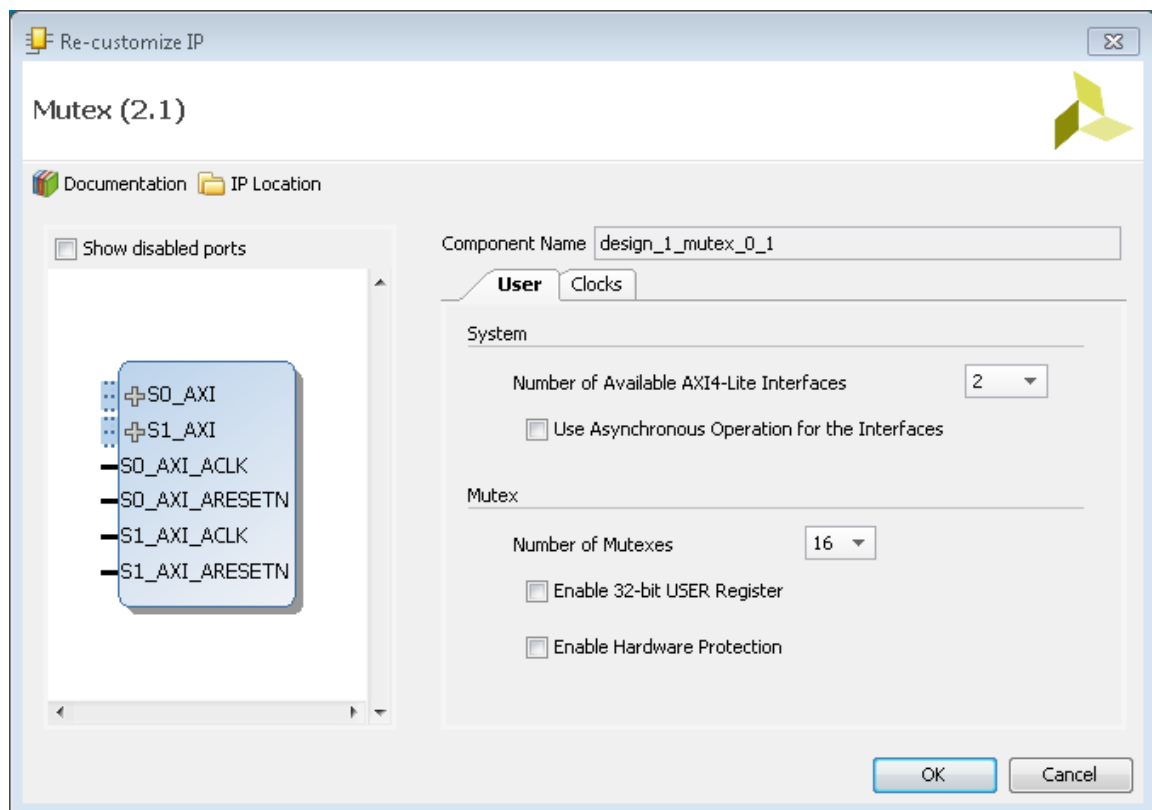
Figures in this chapter are illustrations of the Vivado IDE. The layout depicted here might vary from the current version.

There are two tabs in Vivado IDE core configuration screen, User and Clocks. The User screen is divided in two categories: System and Mutex. When using the Vivado IP integrator feature, the addresses are auto-generated.

User Tab Configuration

The User tab configuration screen is shown in the following figure.

Figure 2: User Tab Configuration



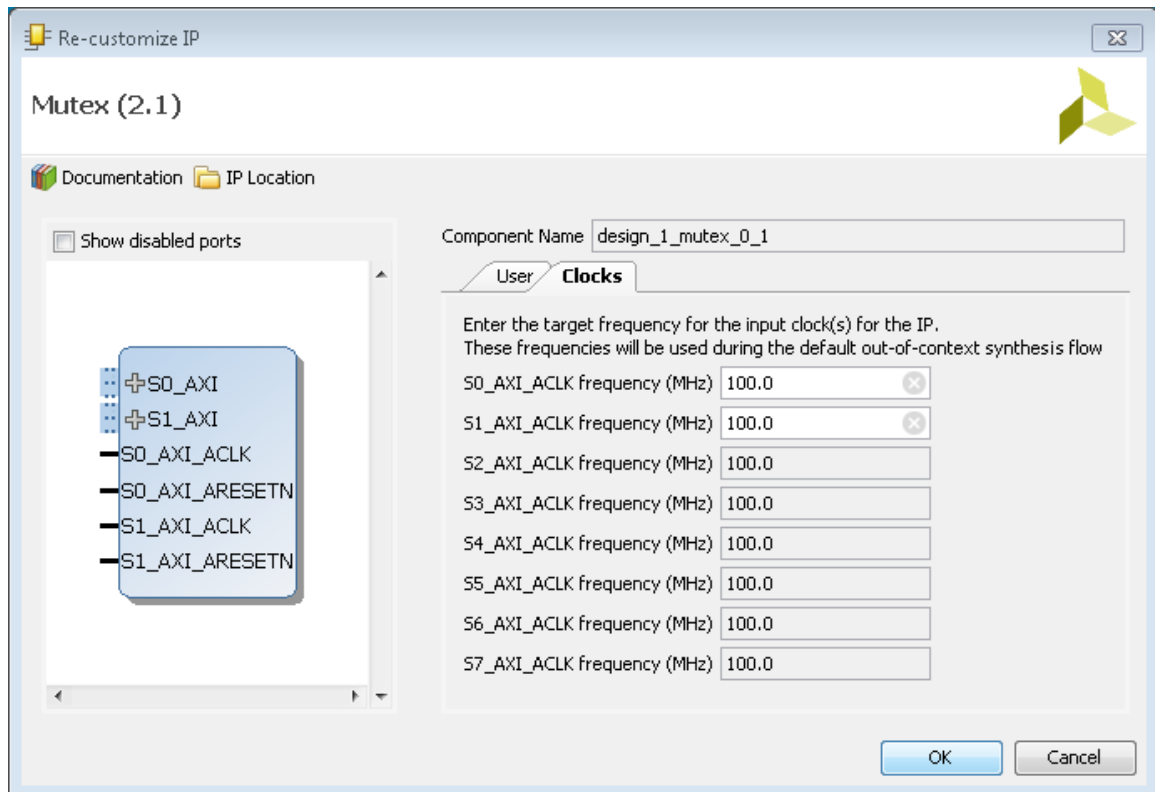
- **Number of Available AXI4-Lite Interfaces:** Sets the number of available bus interfaces, typically one interface per connected processor.
- **Use Asynchronous Operation for the Interfaces:** Enables asynchronous operation, when the clocks of the used interfaces are not identical.
- **Number of Mutexes:**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

Clocks Tab Configuration

The Clocks tab configuration screen is shown in the following figure.

Figure 3: Clocks Tab Configuration Screen



- **Sn_AXI_ACLK frequency (MHz)**: Sets the frequency for the input clock (n = 0–7).

Parameter Values

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 the following table.

Table 8: 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
G18	Number of synchronization FF for each clock domain crossing. All interfaces except S0_AXI have an additional latency of C_NUM_SYNC_FF S0_AXI clock cycles plus C_NUM_SYNC_FF clock cycles for the local Sx_AXI interface.	C_NUM_SYNC_FF	1-8	2	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

User Parameters

The following table shows the relationship between the fields in the Vivado® IDE and the user parameters (which can be viewed in the Tcl Console).

Table 9: User Parameters

Vivado IDE Parameter/Value	User Parameter/Value	Default Value
Number of Available AXI4-Lite Interfaces	C_NUM_AXI	2
Use Asynchronous Operation for the Interfaces	C_ASYNC_CLKS	0
Number of Mutexes	C_NUM_MUTEX	16
Enable 32-bit USER Register	C_ENABLE_USER	0
Enable Hardware Protection	C_ENABLE_HW_PROT	0
Sn_AXI_ACLK frequency (MHz) ¹	Sn_AXI_ACLK_FREQ_MHZ	100.0

Notes:

1. n=0-7

Constraining the Core

This section contains information about constraining the core in the Vivado® Design Suite.

Required Constraints

This section is not applicable for this IP core.

Device, Package, and Speed Grade Selections

This section is not applicable for this IP core.

Clock Frequencies

This section is not applicable for this IP core.

Clock Management

The Mutex core 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

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

Simulation

For comprehensive information about Vivado® simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation (UG900)*.

Synthesis and Implementation

For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP (UG896)*.

Upgrading

This appendix contains information about migrating a design from the ISE® Design Suite to the Vivado® Design Suite, and for upgrading to a more recent version of the IP core. For customers upgrading in the Vivado Design Suite, important details (where applicable) about any port changes and other impact to user logic are included.

Migrating to the Vivado Design Suite

For information about migrating to the Vivado Design Suite, see the *ISE to Vivado Design Suite Migration Guide* ([UG911](#)).

Upgrading in the Vivado Design Suite

This section provides information about any changes to the user logic or port designations that take place when you upgrade to a more current version of this IP core in the Vivado Design Suite.

Debugging

This appendix includes details about resources available on the Xilinx[®] Support website and debugging tools.

If the IP requires a license key, the key must be verified. The Vivado[®] design tools have several license checkpoints for gating licensed IP through the flow. If the license check succeeds, the IP can continue generation. Otherwise, generation halts with an error. License checkpoints are enforced by the following tools:

- Vivado Synthesis
- Vivado Implementation
- `write_bitstream` (Tcl command)



IMPORTANT! IP license level is ignored at checkpoints. The test confirms a valid license exists. It does not check IP license level.

Finding Help on Xilinx.com

To help in the design and debug process when using the core, the [Xilinx Support web page](#) contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support. The [Xilinx Community Forums](#) are also available where members can learn, participate, share, and ask questions about Xilinx solutions.

Documentation

This product guide is the main document associated with the core. This guide, along with documentation related to all products that aid in the design process, can be found on the [Xilinx Support web page](#) or by using the Xilinx[®] Documentation Navigator. Download the Xilinx Documentation Navigator from the [Downloads page](#). For more information about this tool and the features available, open the online help after installation.

Answer Records

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 can 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 Core

AR [54409](#).

Technical Support

Xilinx provides technical support on the [Xilinx Community Forums](#) for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To ask questions, navigate to the [Xilinx Community Forums](#).

Debug Tools

There are many tools available to address Mutex design issues. It is important to know which tools are useful for debugging various situations.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx® devices.

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)

See the *Vivado Design Suite User Guide: Programming and Debugging* ([UG908](#)).

Reference Boards

All Xilinx® development boards support the Mutex core. 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 Mentor Graphics Questa Advanced Simulator is described below. A similar approach can be used with other simulators.

- Check for the latest supported versions of Questa Advanced Simulator in the [Xilinx Design Tools: Release Notes Guide](#).
- 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 core, see the timing in the relevant specification:
 - For AXI4-Lite see the *Arm® AMBA® AXI and ACE Protocol Specification* ([IH10022E](#)).

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® debug feature is a valuable resource to use in hardware debug. The signal names mentioned in the following individual sections can be probed using the debug feature 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.
- If your outputs go to 0, check your licensing.

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 debug feature capture that the waveform is correct for accessing the AXI4-Lite interface.

Application Software Development

Device Drivers

The Mutex core is supported by the mutex driver, included with the Xilinx[®] Vitis[™] unified software platform.

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see [Xilinx Support](#).

Documentation Navigator and Design Hubs

Xilinx[®] Documentation Navigator (DocNav) provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open DocNav:

- From the Vivado[®] IDE, select **Help** → **Documentation and Tutorials**.
- On Windows, select **Start** → **All Programs** → **Xilinx Design Tools** → **DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In DocNav, click the **Design Hubs View** tab.
- On the Xilinx website, see the [Design Hubs](#) page.

Note: For more information on DocNav, see the [Documentation Navigator](#) page on the Xilinx website.

References

These documents provide supplemental material useful with this guide:

1. *Arm® AMBA® AXI and ACE Protocol Specification* [IH10022E](#).
2. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* ([UG994](#))
3. *Vivado Design Suite User Guide: Designing with IP* ([UG896](#))
4. *Vivado Design Suite User Guide: Getting Started* ([UG910](#))
5. *Vivado Design Suite User Guide: Logic Simulation* ([UG900](#))
6. *ISE to Vivado Design Suite Migration Guide* ([UG911](#))
7. *Vivado Design Suite User Guide: Programming and Debugging* ([UG908](#))

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
11/21/2019 Version 2.1	
General Updates	Replaced SDK with Vitis™ software platform
11/18/2015 Version 2.1	
General Updates	Added support for UltraScale+™
06/24/2015 Version 2.1	
General Updates	Moved performance and resource utilization data to the web
04/02/2015 Version 2.1	
General Updates	<ul style="list-style-type: none"> • Clarified address map when multiple mutexes are used. • Added C_NUM_SYNC_FF parameter to control number of synchronization FF.
03/20/2013 Version 1.0	
Initial release.	Initial release as a Product Guide; replaces PG089. There are no documentation changes for this release.

Please Read: Important Legal Notices

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 Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at <https://www.xilinx.com/legal.htm#tos>; 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 such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at <https://www.xilinx.com/legal.htm#tos>.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

Copyright

© Copyright 2013-2019 Xilinx, Inc. Xilinx, the Xilinx logo, Alveo, Artix, Kintex, Spartan, Versal, 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, PrimeCell, Mali, and MPCore are trademarks of Arm Limited in the EU and other countries. All other trademarks are the property of their respective owners.