LogiCORE IP Mutex (v1.00a)

DS775 September 21, 2010 **Product Specification**

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 and PLB v4.6
- Configurable number of PLB interfaces from 0 to 8
- 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	Spartan®-3, Spartan-3E, Spartan-6, Spartan-3A/3AN/3A DSP, Virtex®-4, Virtex-5, Virtex-6				
Supported User Interfaces	AXI-Lite, PLB				
Resources Used	LUTs	FFs	DSP Slices	F_{MAX} (MHz)	Block RAMs
	See Table 10.				0
Provided with Core					
Documentation	Product Specification				
Design Files	VHDL				
Example Design	Not Provided				
Test Bench	Not Provided				
Constraints File	Not Provided				
Simulation Model	Mentor Graphics ModelSim v6.5c and above				
Tested Design Tools					
Design Entry Tools	XPS 12.3				
Simulation	Mentor Graphics ModelSim v6.5c and above				
Synthesis Tools	ISE® 12.3				
Support					
Provided by Xilinx, Inc.					

1. For a complete listing of supported devices, see the release notes for this core.

Functional Description

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.

I/O Signals

The Mutex supports both PLB and AXI4-Lite interfaces simultaneously, the number of interfaces of each type are independently configured from 0 to 8. All interfaces are individually configured and contain the signals listed in Table 1 for PLB and Table 2 for AXI4-Lite, where <x> denotes the interface number (0 to 7).

Table 1: **I/O Signal Description for the PLB Interface**

Port	Signal Name	Interface	I/O	Initial State	Description
System Signals					
P1	SPLB<x>_Clk	System	I	-	PLB clock
P2	SPLB<x>_Rst	System	I	-	PLB reset, active high
PLB Interface Signals					
P3	PLB<x>_ABus[0:31]	PLB	I	-	PLB address bus
P4	PLB<x>_PAValid	PLB	I	-	PLB primary address valid
P5	PLB<x>_masterID[0: C_SPLB<x>_MID_WIDTH - 1]	PLB	I	-	PLB current master identifier
P6	PLB<x>_RNW	PLB	I	-	PLB read not write
P7	PLB<x>_BE[0:(C_SPLB<x>_ DWIDTH/8) - 1]	PLB	I	-	PLB byte enables
P8	PLB<x>_size[0:3]	PLB	I	-	PLB size of requested transfer
P9	PLB<x>_type[0:2]	PLB	I	-	PLB transfer type
P10	PLB<x>_wrDBus[0:C_SPLB<x>_ DWIDTH - 1]	PLB	I	-	PLB write data bus
Unused PLB Interface Signals					
P11	PLB<x>_UABus[0:31]	PLB	I	-	PLB upper address bits
P12	PLB<x>_SAValid	PLB	I	-	PLB secondary address valid
P13	PLB<x>_rdPrim	PLB	I	-	PLB secondary to primary read request indicator
P14	PLB<x>_wrPrim	PLB	I	-	PLB secondary to primary write request indicator
P15	PLB<x>_abort	PLB	I	-	PLB abort bus request
P16	PLB<x>_busLock	PLB	I	-	PLB bus lock
P17	PLB<x>_MSize[0:1]	PLB	I	-	PLB data bus width indicator
P18	PLB<x>_lockErr	PLB	I	-	PLB lock error
P19	PLB<x>_wrBurst	PLB	I	-	PLB burst write transfer

Table 1: **I/O Signal Description for the PLB Interface *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P20	PLB<x>_rdBurst	PLB	I	-	PLB burst read transfer
P21	PLB<x>_wrPendReq	PLB	I	-	PLB pending bus write request
P22	PLB<x>_rdPendReq	PLB	I	-	PLB pending bus read request
P23	PLB<x>_wrPendPri[0:1]	PLB	I	-	PLB pending write request priority
P24	PLB<x>_rdPendPri[0:1]	PLB	I	-	PLB pending read request priority
P25	PLB<x>_reqPri[0:1]	PLB	I	-	PLB current request priority
P26	PLB<x>_TAttribute[0:15]	PLB	I	-	PLB transfer attribute
PLB Slave Interface Signals					
P27	SI<x>_addrAck	PLB	O	0	Slave address acknowledge
P28	SI<x>_SSize[0:1]	PLB	O	0	Slave data bus size
P29	SI<x>_wait	PLB	O	0	Slave wait
P30	SI<x>_rearbitrate	PLB	O	0	Slave bus rearbitrate
P31	SI<x>_wrDAck	PLB	O	0	Slave write data acknowledge
P32	SI<x>_wrComp	PLB	O	0	Slave write transfer complete
P33	SI<x>_rdDBus[0:C_SPLB<x>_ DWIDTH - 1]	PLB	O	0	Slave read data bus
P34	SI<x>_rdDAck	PLB	O	0	Slave read data acknowledge
P35	SI<x>_rdComp	PLB	O	0	Slave read transfer complete
P36	SI<x>_MBusy[0:C_SPLB<x>_ NUM_MASTERS - 1]	PLB	O	0	Slave busy
P37	SI<x>_MWrErr[0:C_SPLB<x>_ NUM_MASTERS - 1]	PLB	O	0	Slave write error
P38	SI<x>_MRdErr[0: C_SPLB<x>_NUM_MASTERS - 1]	PLB	O	0	Slave read error
Unused PLB Slave Interface Signals					
P39	SI<x>_wrBTerm	PLB	O	0	Slave terminate write burst transfer
P40	SI<x>_rdWdAddr[0:3]	PLB	O	0	Slave read word address
P41	SI<x>_rdBTerm	PLB	O	0	Slave terminate read burst transfer
P42	SI<x>_MIRQ[0: C_SPLB<x>_NUM_MASTERS - 1]	PLB	O	0	Master interrupt request

Table 2: **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

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

Port	Signal Name	Interface	I/O	Initial State	Description
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.
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

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

Port	Signal Name	Interface	I/O	Initial State	Description
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 assumes that all byte lanes are active.

Parameters

The Mutex design is parameterized to tailor it for different systems. This allows the user to configure a design that utilizes 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 3. The generics, G2 through G9, are configured for the PLB interface, while generics G12 through G16, are configured for the AXI interface.

Table 3: **Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameter					
G1	Target FPGA family	C_FAMILY	spartan3, aspartan3, spartan3e, aspartan3e, spartan3a, aspartan3a, spartan3adsp, aspartan3adsp, spartan6, aspartan6, spartan6l, qsspartan6, qsspartan6l, virtex4, qrvirtex4, qvirtex4, virtex5, qrvirtex5, virtex6, virtex6l, qrvirtex6	virtex6	string
PLB Parameters					
G2	PLB Base Address	C_SPLB<x>_BASEADDR	Valid Address(1)	None(2)	std_logic_vector
G3	PLB High Address	C_SPLB<x>_HIGHADDR	Valid Address(3)	None(2)	std_logic_vector
G4	PLB least significant address bus width	C_SPLB<x>_AWIDTH	32	32	integer
G5	PLB data width	C_SPLB<x>_DWIDTH	32, 64, 128	32	integer
G6	Selects point-to-point or shared bus topology	C_SPLB<x>_P2P	0 = Shared Bus Topology 1 = Point-to-Point Bus Topology(4)	0	integer

Table 3: **Design Parameters *(Cont'd)***

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G7	PLB Master ID Bus Width	C_SPLB<x>_MID_ WIDTH	$\log_2$(C_SPLB_NUM_ MASTERS) with a minimum value of 1	1	integer
G8	Number of PLB Masters	C_SPLB<x>_NUM_ MASTERS	1 - 16	1	integer
G9	Support Bursts	C_SPLB<x>_SUPPORT_ BURSTS	0	0	integer
G10	Width of the Slave Data Bus	C_SPLB_NATIVE_ DWIDTH	32	32	integer
G11	Frequency of PLB interface	C_SPLB<x>_CLK_ FREQ_HZ	integer	100_000 _000	integer
AXI Parameters					
G12	AXI Base Address	C_S<x>_AXI_BASEADDR	Valid Address(1)	None(2)	std_logic_ vector
G13	AXI High Address	C_S<x>_AXI_HIGHADDR	Valid Address(3)	None(2)	std_logic_ vector
G14	AXI address bus width	C_S<x>_AXI_ADDR_ WIDTH	32	32	integer
G15	AXI data bus width	C_S<x>_AXI_DATA_ WIDTH	32	32	integer
G16	AXI interface type	C_S<x>_AXI_PROTOCOL	AXI4LITE	AXI4 LITE	string
Mutex Parameters					
G17	Specify if interfaces are synchronous or asynchronous	C_ASYNC_CLKS	0 - 1	0	Integer
G18	Number of PLB interfaces	C_NUM_PLB	0 - 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 shall be contained inside the core	C_NUM_MUTEX	1 - 32	16	Integer

Notes:

1. The user must set the values. The C_<interface>_BASEADDR must be a multiple of the range, where the range is C_<interface>_HIGHADDR - C_<interface>_BASEADDR + 1.
2. No default value will be specified to insure that the actual value is set, i.e., if the value is not set, a compiler error will be generated.
3. C_<interface>_HIGHADDR - C_<interface>_BASEADDR must be a power of 2 greater than equal to C_<interface>_BASEADDR + 0xFF.
4. Value of '1' is not supported in this core.

Parameter - Port Dependencies

The dependencies between the Mutex core design parameters and I/O signals are described in Table 4. In addition, when certain features is deselected, the related logic will no longer be a part of the design. The unused input and output signals are set to a specified value.

Table 4: **Mutex Parameter-Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G5	C_SPLB<x>_DWIDTH	P7, P10, P33	-	Affects the number of bits in data bus
G7	C_SPLB<x>_MID_WIDTH	P5	G8	This value is calculated as: $\log_2$(C_SPLB<x>_NUM_MASTERS) with a minimum value of 1
G8	C_SPLB<x>_NUM_MASTERS	P36, P37, P38, P42	-	Affects the number of PLB masters
G14	C_S<x>_AXI_ADDR_WIDTH	P45, P55	-	Defines the width of the ports
G15	C_S<x>_AXI_DATA_WIDTH	P48, P49, P58	-	Defines the width of the ports
I/O Signals				
P5	PLB<x>_masterID[0: C_SPLB_MID_WIDTH - 1]	-	G7	Width of the PLB<x>_masterID varies according to C_SPLB<x>_MID_WIDTH
P7	PLB<x>_BE[0:(C_SPLB_ DWIDTH/8) -1]	-	G5	Width of the PLB<x>_BE varies according to C_SPLB<x>_DWIDTH
P10	PLB<x>_wrDBus[0:C_SPLB_ DWIDTH - 1]	-	G5	Width of the PLB<x>_wrDBus varies according to C_SPLB<x>_DWIDTH
P33	SI<x>_rdDBus[0:C_SPLB_ DWIDTH - 1]	-	G5	Width of the SI<x>_rdDBus varies according to C_SPLB<x>_DWIDTH
P36	SI<x>_MBusy[0: C_SPLB_NUM_MASTERS - 1]	-	G8	Width of the SI<x>_MBusy varies according to C_SPLB<x>_NUM_MASTERS
P37	SI<x>_MWrErr[0: C_SPLB_NUM_MASTERS - 1]	-	G8	Width of the SI<x>_MWrErr varies according to C_SPLB<x>_NUM_MASTERS
P38	SI<x>_MRdErr[0:C_SPLB_ NUM_MASTERS - 1]	-	G8	Width of the SI<x>_MRdErr varies according to C_SPLB<x>_NUM_MASTERS
P42	SI<x>_MIRQ[0: C_SPLB_NUM_MASTERS - 1]	-	G8	Width of the SI<x>_MIRQ varies according to C_SPLB<x>_NUM_MASTERS
P45	S<x>_AXI_AWADDR[C_S<x>_ AXI_ADDR_WIDTH-1:0]	-	G14	Port width depends on the generic C_S<x>_AXI_ADDR_WIDTH
P48	S<x>_AXI_WDATA[C_S<x>_ AXI_DATA_WIDTH-1:0]	-	G15	Port width depends on the generic C_S<x>_AXI_DATA_WIDTH
P49	S<x>_AXI_WSTB[C_S<x>_ AXI_DATA_WIDTH/8-1:0]	-	G15	Port width depends on the generic C_S<x>_AXI_DATA_WIDTH
P55	S<x>_AXI_ARADDR[C_S<x>_ AXI_ADDR_WIDTH -1:0]	-	G14	Port width depends on the generic C_S<x>_AXI_ADDR_WIDTH
P58	S<x>_AXI_RDATA[C_S<x>_ AXI_DATA_WIDTH -1:0]	-	G15	Port width depends on the generic C_S<x>_AXI_DATA_WIDTH

Register Descriptions

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 PLB and AXI interfaces are blocked. Interface arbitration has fixed priority where PLB 0-7 has higher priority than AXI 0-7 in descending order. For example, PLB0 has the highest priority, and S7_AXI the lowest.

Table 5 shows all the Mutex registers and their addresses offsets.

Table 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 accessible by the user and are handled implicitly in the background. HWID contains which port the PLB or AXI master is attached, and in the PLB case, also the master ID for the accessing master on that PLB bus. 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 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 EDK, 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™ and PowerPC® processors. 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 will be 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 6: **Write Data Register**

Reserved		CPUID		Lock
0	22	23	30	31

Table 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 9.

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

USER	
0	31

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

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

Design Implementation

Target Technology

The target technology is an FPGA listed in the Supported Device Family field of the LogiCORE IP Facts Table.

Device Utilization and Performance Benchmarks

Core Performance

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 Virtex-6 FPGA as the target device are detailed in Table 10.

Table 10: **Performance and Resource Utilization Benchmarks on the Virtex-6 FPGA (xc6vlx240t-ff1156-3)**

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

Support

Xilinx provides technical support for this LogiCORE 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*.

Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx ISE Design Suite Embedded Edition software under the terms of the Xilinx End User License. The core is generated using the Xilinx ISE Embedded Edition software (EDK).

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 modules and software, please contact your local Xilinx sales representative.

Reference Documents

1. IBM CoreConnect128-Bit Processor Local Bus, Architectural Specification (v4.6)
2. AMBA® AXI Protocol Version: 2.0 Specification (ARM IHI 0022C)

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
9/21/10	1.0	Initial release.

Notice of Disclaimer