LogiCORE IP Mailbox (v1.00a)

DS776 September 21, 2010 **Product Specification**

Introduction

In a multiprocessor environment, the processors need to communicate data with each other. The easiest method is to set up inter-processor communication through a mailbox. Mailbox features a bi-directional communication channel between two processors. The Mailbox can be connected to the processor either through a PLB, AXI4-Lite, AXI4-Stream, or FSL interface. The PLB interface option is available for the MicroBlaze™ processor, PowerPC® processor, or any other PLBv46 master. The AXI4-Lite, AXI4-Stream, and FSL options are available for connection to any IP that supports them, for example MicroBlaze.

Features

- Supports AXI4-Lite, AXI4-Stream, PLB v4.6 and FSL independently on each of the ports
- Configurable depth of mailbox
- Configurable interrupt thresholds and maskable interrupts
- Configurable synchronous or asynchronous operation
- Bi-directional communication

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 27.				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				
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 Mailbox is used for bi-directional inter-processor communication. A mailbox is a link between two otherwise separate processor systems. Other multi-port IP blocks, such as a memory controller, may also be shared by the two sub systems.

In addition to sending the actual data between processors, the mailbox can be used to generate interrupts between the processors.

The Mailbox in a typical PLBv46 system is shown in the top-level block diagram in Figure 1. The same system partitioning is also used for AXI4-Lite interface option. FSL and AXI4-Stream options will have the Mailbox interface connected directly to a master with no bus inbetween.

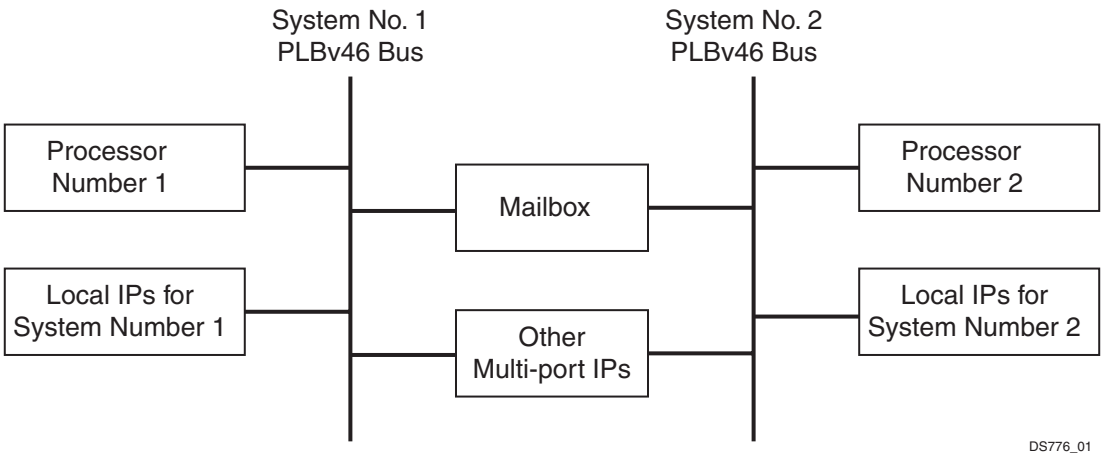

Figure 1: **Mailbox in a PLBv46 System**

Mailbox I/O Signals

The Mailbox has two interfaces that are used to connect to the rest of the system. Both interfaces can be independently configured to use the PLBv46, AXI4-Lite, AXI4-Stream, or FSL interface. The signal descriptions are included in five tables:

1. The PLB signals are described in Table 1.
2. The AXI4-Lite signals are described in Table 2.
3. The AXI4-Stream signals are described in Table 3.
4. The FSL signals are described in Table 4.
5. The common signals are described in Table 5.

All signals in Table 1 through Table 4 apply to both interface sides; <x> denotes the interface number, which may be 0 or 1.

Table 1: **PLBv46 I/O Signal Description**

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
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

Table 1: **PLBv46 I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
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: **AXI4-Lite I/O Signal Description**

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)

Table 2: **AXI4-Lite I/O Signal Description *(Cont'd)***

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

Table 3: **AXI4-Stream I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
System Signals					
P62	S<x>_AXIS_ACLK	System	I	-	AXI Clock
P63	M<x>_AXIS_ACLK	System	I	-	AXI Clock
AXI Slave Channel Signals					
P64	S<x>_AXIS_TDATA[C_S<x>_AXIS_DATA_WIDTH - 1: 0]	AXIS	I	-	Data
P65	S<x>_AXIS_TLAST	AXIS	I	-	Last data flag, indicates that this is the last word.
P66	S<x>_AXIS_TVALID	AXIS	I	-	Data valid. This signal indicates that valid data and last flag are available.
P67	S<x>_AXIS_TREADY	AXIS	O	0	Data ready. This signal indicates that the slave can accept the data.
AXI Master Channel Signals					
P68	M<x>_AXIS_TDATA[C_M<x>_AXIS_DATA_WIDTH -1:0]	AXIS	O	0x0	Data
P69	M<x>_AXIS_TLAST	AXIS	O	0	Last data flag, indicates that this is the last word.
P70	M<x>_AXIS_TVALID	AXIS	O	0	Data valid. This signal indicates that valid data and last flag are available.
P71	M<x>_AXIS_TREADY	AXIS	I	-	Data ready. This signal indicates that the slave can accept the data.

Table 4: **FSL I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
FSL Master Interface Signals					
P72	FSL<x>_M_Clk	MFSL	I	N/A	This port provides the input clock to the FSL master interface of the mailbox when used in the asynchronous FIFO mode (C_ASYNC_CLKS = 1). All transactions on the master interface use this clock when implemented in the asynchronous mode
P73	FSL<x>_M_Data	MFSL	I	0	The data input to the FSL master interface of the mailbox
P74	FSL<x>_M_Control	MFSL	I	0	Unused for mailbox
P75	FSL<x>_M_Write	MFSL	I	0	Input signal that controls the write enable signal of the FSL master interface of the FIFO. When set to 1, the value of FSL<x>_M_Data is pushed into the mailbox FIFO on a rising clock edge.
P76	FSL<x>_M_Full	MFSL	O	N/A	Output signal on the FSL master interface of the FIFO indicating that the FIFO is full.
FSL Slave Interface Signals					
P77	FSL<x>_S_Clk	SFSL	I	N/A	This port provides the input clock to the FSL slave interface on the mailbox when used in the asynchronous FIFO mode (C_ASYNC_CLKS = 1). All transactions on the slave interface use this clock when implemented in the asynchronous mode

Table 4: **FSL I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P78	FSL<x>_S_Data	SFSL	O	N/A	The data output bus onto the FSL slave interface of the mailbox
P79	FSL<x>_S_Control	SFSL	O	N/A	Unused for mailbox
P80	FSL<x>_S_Read	SFSL	I	0	Input signal on the FSL slave interface that controls the read acknowledge signal of the FIFO. When set to 1, the value of FSL<x>_S_Data is popped from the FIFO on a rising clock edge.
P81	FSL<x>_S_Exists	SFSL	O	N/A	Output signal on the FSL slave interface indicating that FIFO contains valid data.

Table 5: **Mailbox Common I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
FSL Common Interface Signals					
P82	FSL_Clk	System	I	N/A	This is the input clock to the mailbox when used in synchronous FIFO mode (C_ASYNC_CLKS = 0) and both interfaces are FSL based (C_INTERCONNECT_PORT_<x> = 3). The FSL_Clk is in this case used to clock the core, in all other cases are the internal mailbox clock automatically derived from either SPLB<x>_Clk, S<x>_AXI_ACLK or S<x>_AXIS_ACLK depending on the settings.'
P83	SYS_Rst	System	I	N/A	External system reset. This signal is only required when both interfaces are configured to be streaming interfaces (FSL or AXI4-Stream). If any PLB or AXI4-Lite interface is available this signal is optional.
P84	FSL_Rst	System	O	0	Output reset signal generated by the FSL reset logic. Any peripherals connected to the FSL bus may use this reset signal to operate the peripheral reset.
Common Signals					
P85	Interrupt_0	System	O	0	Interrupt signal that data is available at interface 0
P86	Interrupt_1	System	O	0	Interrupt signal that data is available at interface 1

Parameters

To allow the user to obtain a Mailbox that is uniquely tailored for the system, certain features can be parameterized in the Mailbox design. 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 Mailbox design are as shown in Table 6. The interface related generics, G3 through G19, are separately configured for each interface.

Table 6: **Mailbox 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, virtex6, virtex6l, qrvirtex6	virtex6	string
G2	Level of external reset	C_EXT_RESET_HIGH	0 or 1	1	integer
PLB Parameters					
G3	PLB Base Address	C_SPLB<x>_BASEADDR	Valid Address(1)	None(2)	std_logic_vector
G4	PLB High Address	C_SPLB<x>_HIGHADDR	Valid Address(3)	None(2)	std_logic_vector
G5	PLB least significant address bus width	C_SPLB<x>_AWIDTH	32	32	integer
G6	PLB data width	C_SPLB<x>_DWIDTH	32, 64, 128	32	integer
G7	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
G8	PLB Master ID Bus Width	C_SPLB<x>_MID_WIDTH	$\log_2$(C_SPLB_NUM_MASTERS) with a minimum value of 1	1	integer
G9	Number of PLB Masters	C_SPLB<x>_NUM_MASTERS	1 - 16	1	integer
G10	Support Bursts	C_SPLB<x>_SUPPORT_BURSTS	0	0	integer
G11	Width of the Slave Data Bus	C_SPLB_NATIVE_DWIDTH	32	32	integer
G12	Frequency of PLB interface	C_SPLB<x>_CLK_FREQ_HZ	integer	100_000_000	integer
AXI4-Lite Parameters					
G13	AXI Base Address	C_S<x>_AXI_BASEADDR	Valid Address(1)	None(2)	std_logic_vector
G14	AXI High Address	C_S<x>_AXI_HIGHADDR	Valid Address(3)	None(2)	std_logic_vector

Table 6: **Mailbox Design Parameters *(Cont'd)***

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G15	AXI address bus width	C_S<x>_AXI_ ADDR_WIDTH	32	32	integer
G16	AXI data bus width	C_S<x>_AXI_DATA_ WIDTH	32	32	integer
G17	AXI interface type	C_S<x>_AXI_ PROTOCOL	AXI4LITE	AXI4 LITE	string
AXI4-Stream Parameters					
G18	AXI data bus width	C_S<x>_AXIS_DATA _WIDTH	32	32	integer
G19	AXI data bus width	C_M<x>_AXIS_DATA _WIDTH	32	32	integer
Mailbox Parameters					
G20	Specify if interfaces are synchronous or asynchronous	C_ASYNC_CLKS	0 - 1	0	Integer
G21	Use BRAMs to implement FIFO	C_IMPL_STYLE	0 - 1	1	Integer
G22	FSL bus width	C_FSL_DWIDTH	32	32	Integer
G23	Select interface type that shall be used on port 0: 1 - PLBv46 2 - AXI4-Lite 3 - FSL 4 - AXI4-Stream	C_INTERCONNECT _PORT_0	1 - 4	0	Integer
G24	Select interface type that shall be used on port 1: 1 - PLBv46 2 - AXI4-Lite 3 - FSL 4 - AXI4-Stream	C_INTERCONNECT _PORT_1	1 - 4	0	Integer
G25	FIFO depth of mailbox	C_MAILBOX_ DEPTH	16 - 8192	16	Integer

Table 6: **Mailbox Design Parameters *(Cont'd)***

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G26	Read Clock period for interface 0 when asynchronous LUTRAM is used (in ps)	C_READ_CLOCK_ PERIOD_0	> 0 when enabled	0	Integer
G27	Read Clock period for interface 1when asynchronous LUTRAM is used (in ps)	C_READ_CLOCK_ PERIOD_0	> 0 when enabled	0	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 Mailbox core design parameters and I/O signals are described in Table 7. 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 7: **Mailbox Parameter-Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G6	C_SPLB<x>_DWIDTH	P7, P10, P33	-	Affects the number of bits in data bus
G8	C_SPLB<x>_MID_WIDTH	P5	G9	This value is calculated as: $\log_2$(C_SPLB<x>_NUM_MASTERS) with a minimum value of 1
G9	C_SPLB<x>_NUM_MASTERS	P36, P37, P38, P42	-	Affects the number of PLB masters
G15	C_S<x>_AXI_ADDR_WIDTH	P45, P55	-	Defines the width of the ports
G16	C_S<x>_AXI_DATA_WIDTH	P48, P49, P58	-	Defines the width of the ports
G18	C_S<x>_AXIS_DATA_WIDTH	P64	-	Defines the width of the ports
G19	C_M<x>_AXIS_DATA_WIDTH	P68	-	Defines the width of the ports
G22	C_FSL_DWIDTH	P73, P78	-	Affects the number of bits in data bus
I/O Signals				
P5	PLB<x>_masterID[0: C_SPLB<x>_MID_WIDTH - 1]	-	G8	Width of the PLB<x>_masterID varies according to C_SPLB<x>_MID_WIDTH
P7	PLB<x>_BE[0: (C_SPLB<x>_DWIDTH/8) -1]	-	G6	Width of the PLB<x>_BE varies according to C_SPLB<x>_DWIDTH
P10	PLB<x>_wrDBus[0: C_SPLB<x>_DWIDTH - 1]	-	G6	Width of the PLB<x>_wrDBus varies according to C_SPLB<x>_DWIDTH

Table 7: **Mailbox Parameter-Port Dependencies *(Cont'd)***

Generic or Port	Name	Affects	Depends	Relationship Description
P33	SI<x>_rdDBus[0: C_SPLB<x>_DWIDTH - 1]	-	G6	Width of the SI<x>_rdDBus varies according to C_SPLB<x>_DWIDTH
P36	SI<x>_MBusy[0: C_SPLB<x>_NUM_MASTERS - 1]	-	G9	Width of the SI<x>_MBusy varies according to C_SPLB<x>_NUM_MASTERS
P37	SI<x>_MWrErr[0: C_SPLB<x>_NUM_MASTERS - 1]	-	G9	Width of the SI<x>_MWrErr varies according to C_SPLB<x>_NUM_MASTERS
P38	SI<x>_MRdErr[0: C_SPLB<x>_NUM_MASTERS - 1]	-	G9	Width of the SI<x>_MRdErr varies according to C_SPLB<x>_NUM_MASTERS
P42	SI<x>_MIRQ[0: C_SPLB<x>_NUM_MASTERS - 1]	-	G9	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]	-	G15	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]	-	G16	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]	-	G16	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]	-	G15	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]	-	G16	Port width depends on the generic C_S<x>_AXI_DATA_WIDTH
P64	S<x>_AXIS_TDATA[C_S<x>_AXIS _DATA_WIDTH -1:0]	-	G18	Port width depends on the generic C_S<x>_AXIS_DATA_WIDTH
P68	M<x>_AXIS_TDATA[C_M<x>_AXIS _DATA_WIDTH -1:0]	-	G19	Port width depends on the generic C_M<x>_AXIS_DATA_WIDTH
P74	FSL<x>_M_Data		G22	Width of the FSL<x>_M_Data varies according to C_FSL_DWIDTH
P79	FSL<x>_S_Data		G22	Width of the FSL<x>_S_Data varies according to C_FSL_DWIDTH

Register Descriptions

Each interface of the Mailbox core has the same set of information registers. The information at each interface is not identical but rather localized for that interface since the communication is bi-directional.

Table 8 shows all the Mailbox registers and their addresses for the PLB and AXI4-Lite cases. Much of the information can be acquired for the FSL and AXI4-Stream cases with the use of FSL<x>_M_Full/FSL<x>_S_Exists or S<x>_AXIS_TREADY/M<x>_AXIS_TVALID respectively.

Table 8: **Mailbox Registers**

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
BASEADDR + 0x0	WRDATA	Write	N/A	Write Data address. Write only.
BASEADDR + 0x4	Reserved	N/A	N/A	Reserved for future use
BASEADDR + 0x8	RDDATA	Read	N/A	Read Data address. Read only
BASEADDR + 0xC	Reserved	N/A	N/A	Reserved for future use

Table 8: **Mailbox Registers**

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
BASEADDR + 0x10	STATUS	Read	0x1	Status flags for mailbox. Read only.
BASEADDR + 0x14	ERROR	Read	0x0	Error flags, clear on read. Read only.
BASEADDR + 0x18	SIT	-	-	Send Interrupt Threshold. Read/Write
BASEADDR + 0x1C	RIT	-	-	Receive Interrupt Threshold. Read/Write
BASEADDR + 0x20	IS	-	-	Interrupt Status register. Read/Write
BASEADDR + 0x24	IE	-	-	Interrupt Enable register. Read/Write
BASEADDR + 0x28	IP	-	-	Interrupt Pending register. Read only
BASEADDR + 0x2C	Reserved	-	-	Reserved for future use
BASEADDR + 0x30	Reserved	-	-	Reserved for future use
BASEADDR + 0x34	Reserved	-	-	Reserved for future use
BASEADDR + 0x38	Reserved	-	-	Reserved for future use
BASEADDR + 0x3C	Reserved	-	-	Reserved for future use

Write Data Register (WRDATA)

Writing to this register will result in the data transferred to the RDDATA register at the other interface. Trying to write while the full flag is set will result in an error and the FULL_ERROR bit will be set. The register is write only and a read request issued to WRDATA will be ignored. Bit assignment in the WRDATA register is described in Table 10..

Table 9: **Write Data Register**

WRDATA	
0	C_FSL_DWIDTH-1

Table 10: **Mailbox Write Data Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - C_FSL_DWIDTH - 1	WRDATA	Write	-	Write register to send data to the other interface

Mailbox Read Data Register (RDDATA)

Reading from this register will pop one value from the mail FIFO. Trying to read while the empty flag is set will result in an error and the EMPTY_ERROR bit will be set. The register is read only and a write request issued to RDDATA will be ignored. Bit assignment in the RDDATA register is described in Table 12.

Table 11: **Read Data Register**

RDDATA	
0	C_FSL_DWIDTH-1

Table 12: **Mailbox Read Data Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - C_FSL_DWIDTH - 1	RDDATA	Read	-	Read register to get data word sent from the other interface

Mailbox Status Register (STATUS)

The Mailbox Status Register contains the current status of the mailbox. The register is read only and a write request issued to STATUS will be ignored. Bit assignment in the STATUS register is described in Table 14

Table 13: **Status Register**

Reserved		RTA	STA	Full	Empty
0	27	28	29	30	31

Table 14: **Mailbox Status Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 27	Reserved			Reserved for future use
28	RTA	Read	0	Receive Threshold Active indicates the current FIFO status of this interface in the receive direction '0' = The receive FIFO level is less than or equal to the RIT threshold '1' = The receive FIFO level is greater than the RIT threshold
29	STA	Read	0	Send Threshold Active indicates the current FIFO status of this interface in the send direction '0' = The send FIFO level is greater than the SIT threshold '1' = The send FIFO level is less than or equal to the SIT threshold
30	Full	Read	'0'	Indicates the current status of this interface in the send direction '0' = There is room for more data '1' = The FIFO is full, any attempts to write data will be ignored and generate an error
31	Empty	Read	'1'	Indicates the current status of this interface in the receive direction '0' = There is data available '1' = The FIFO is empty, any attempts to read data will be ignored and generate an error

Mailbox Error Register (ERROR)

The Mailbox Error Register contains the error flags for PLB and AXI4-Lite accesses from this interface. The error register will be cleared at read, this means that all bits are sticky and that they indicate any errors that occurred since last time the error register was read. The register is read only and a write request issued to ERROR will be ignored. Bit assignment in the ERROR register is described in Table 16.

Table 15: **Error Register**

Reserved		Full Error	Empty Error
0	29	30	31

Table 16: **Mailbox Error Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 29	Reserved			Reserved for future use
30	Full Error	Read	'0'	Indicates if there has been any attempts to write to the WRDATA register while the Full flag was asserted since the error register was last read '0' = No error has occurred '1' = One or more attempts to write while FSL link was full
31	Empty Error	Read	'0'	Indicates if there has been any attempts to read from the RDDATA register while the Empty flag was asserted since the error register was last read '0' = No error has occurred '1' = One or more attempts to read while FSL link was empty

Mailbox Send Interrupt Threshold Register (SIT)

The Mailbox Send Interrupt Threshold Register contains the interrupt threshold for this interface in the send direction. Depending on the send FIFO data level writing a new SIT can cause a rising edge on STA that can generate a STI interrupt if it is enabled in the IE register. Bit assignment in the SIT register is described in Table 18.

Table 17: **SIT Register**

	SIT	
0	32-Log2(C_MAILBOX_DEPTH)	31

Table 18: **Mailbox SIT Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
Log2(C_MAILBOX_DEPTH)	SIT	Read/Write	0	Lower Log2(C_MAILBOX_DEPTH) bits used, right justified to bit 31

Mailbox Receive Interrupt Threshold Register (RIT)

The Mailbox Receive Interrupt Threshold Register contains the interrupt threshold for this interface in the receive direction. Depending on the receive FIFO data level writing a new RIT can cause a rising edge on RTA that can generate a RTI interrupt if it is enabled in the IE register. Bit assignment in the RIT register is described in Table 20.

Table 19: **RIT Register**

	RIT	
0	32-Log2(C_MAILBOX_DEPTH)	31

Table 20: **Mailbox RIT Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
Log2(C_MAILBOX_DEPTH)	RIT	Read/Write	0	Lower Log2(C_MAILBOX_DEPTH) bits used, right justified to bit 31

Mailbox Interrupt Status Register (IS)

The Mailbox Interrupt Status Register contains the current interrupt status for this interface. There are three types of interrupts that can be generated. Mailbox Error interrupt are generated when any of the bits in the ERROR register is set. The other two interrupts are FIFO related: RTI is generated for a rising edge on the RTA bit in the STATUS register and STI that is generated for a rising edge on the STA STATUS register bit. RTI and STI are used to indicate that it is time to read from or write to the FIFOs to avoid any stalls in the data flow. Bit assignment in the IS register is described in Table 22.

Table 21: **IS Register**

Reserved		ERR	RTI	STI
0	28	29	30	31

Table 22: **Mailbox IS Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 28	Reserved			Reserved for future use
29	ERR	Read/Write	'0'	Mailbox Error Interrupt Status for this interface. Values for read: '0' = No interrupt event has occurred. '1' = A Mailbox error has occurred. Values for write: '0' = No change '1' = Acknowledge and clear the interrupt if it is active
30	RTI	Read/Write	'0'	Mailbox Receive Threshold Interrupt pending status for this interface. Values for read: '0' = No interrupt event has occurred. '1' = Data level in the receive FIFO has caused a RTI. Values for write: '0' = No change '1' = Acknowledge and clear the interrupt if it is active
31	STI	Read/Write	'0'	Mailbox Send Threshold Interrupt penging status for this interface. Values for read: '0' = No interrupt event has occurred. '1' = Data level in the send FIFO has caused a STI. Values for write: '0' = No change '1' = Acknowledge and clear the interrupt if it is active

Mailbox Interrupt Enable Register (IE)

The Mailbox Interrupt Enable Register contains the mask for the allowed interrupts on this interface. Bit assignment in the IE register is described in Table 24.

Table 23: **IE Register**

Reserved		ERR	RTI	STI
0	28	29	30	31

Table 24: **Mailbox IE Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 28	Reserved			Reserved for future use
29	ERR	Read/Write	'0'	Mailbox Error Interrupt Enable for this interface '0' = ERR interrupt is disabled '1' = ERR interrupt is enabled
30	RTI	Read/Write	'0'	Mailbox Receive Threshold Interrupt Enable for this interface '0' = RTI interrupt is disabled '1' = RTI interrupt is enabled
31	STI	Read/Write	'0'	Mailbox Send Threshold Interrupt Enable for this interface '0' = STI interrupt is disabled '1' = STI interrupt is enabled

Mailbox Interrupt Pending Register (IP)

The Mailbox Interrupt Pending Register contains the currently pending interrupts from this interface. It is a read only register generated by performing bit-wise AND between the IS and IE registers. A write request issued to IP will be ignored. Bit assignment in the IP register is described in Table 26. All the bits in this register are ORed together to generate the interrupt output signal for this interface. When an interrupt has been serviced it is acknowledge by writing the corresponding bit to the IS Register.

Table 25: **IP Register**

Reserved		ERR	RTI	STI
0	28	29	30	31

Table 26: **Mailbox IP Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 28	Reserved			Reserved for future use
29	ERR	Read	'0'	Mailbox Error Interrupt Pending status for this interface '0' = No pending interrupt '1' = Pending interrupt for Mailbox errors
30	RTI	Read	'0'	Mailbox Receive Threshold Interrupt Pending status for this interface '0' = No pending interrupt '1' = Pending interrupt for data level in receive FIFO
31	STI	Read	'0'	Mailbox Send Threshold Interrupt Pending status for this interface '0' = No pending interrupt '1' = Pending interrupt for data level in send FIFO

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

Since the Mailbox 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 Mailbox core is combined with other designs in the system, the utilization of FPGA resources and timing of the Mailbox design will vary from the results reported here. These values are generated from a minimal dual MicroBlaze system, each with a UART Lite and a shared Mailbox as the only peripherals.

The Mailbox resource utilization for various parameter combinations measured with Virtex-6 as the target device are detailed in Table 27.

Table 27: **Performance and Resource Utilization Benchmarks on Virtex-6 (xc6vlx240t-ff1156-3)**

Parameter Values (other parameters at default value)				Device Resources				Performance
C_ASYNC_CLKS	C_INTERCONNECT_PORT_0	C_INTERCONNECT_PORT_1	C_MAILBOX_DEPTH	Slices	Slice Flip-Flops	LUTs	BRAMs	F_{MAX} (MHz)
0	1	1	16	122	194	324	0	326
0	1	1	64	158	225	440	0	323
0	1	1	2048	143	304	358	4	291
0	1	2	16	149	218	328	0	322
0	1	3	16	92	132	233	0	324
0	1	4	16	77	132	236	0	314
1	1	1	16	161	307	394	0	309
1	1	1	64	180	387	501	0	320
1	1	1	2048	200	450	566	4	314
1	1	2	16	135	307	368	0	311
1	1	3	16	98	213	259	0	321
1	1	4	16	86	213	256	0	320

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)
3. AMBA® 4 AXI4-Stream Protocol Version: 1.0 Specification (ARM IHI 0051A)

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