Virtex-5 FPGA Embedded Processor Block with PPC440 Processor (v1.01a)

DS621 September 22, 2008 **Product Specification**

Introduction

This document describes the wrapper for the Virtex®-5 FPGA Embedded Processor Block. For details regarding the Virtex-5 Embedded Block, see the *Embedded Processor Block in Virtex-5 FPGAs Reference Guide*.

Features

- PowerPC® 440x5 dual-issue, superscalar 32-bit embedded processor developed by IBM
- 32 KB instruction cache, 32 KB data cache
- Memory Management Unit (MMU)
- Crossbar interconnect with 9 inputs and 2 outputs (128 bits wide), implemented in hardware
- 128-bit Processor Local Bus (PLB) version 4.6 interfaces
- High-speed memory controller interface
- Four DMA controllers with LocalLink channel interfaces
- Auxiliary Processor Unit (APU) controller and interface for connecting FPU or custom coprocessor

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex-5	
Version of core	ppc440_virtex5	v1.00a
Resources Used		
	Min	Max
Slices	0	9
LUTs	0	9
FFs	0	8
Block RAMs	0	0
Special Features	PPC440 Processor Block	
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs & application notes	None	
Additional Items	None	
Design Tool Requirements		
Xilinx Implementation Tools	ISE® 10.1 or later	
Verification	N/A	
Simulation	ModelSim SE/PE, NCsim	
Synthesis	XST	
Support		
Provided by Xilinx, Inc.		

Functional Description

The embedded block in Virtex-5 FXT FPGA devices contains the PowerPC 440 processor and other modules that allow system designers to improve the performance and reduce the fabric resource utilization of FPGA designs. To improve memory access among the processor and other master devices in the system, the embedded block contains a high bandwidth crossbar switch. The crossbar accepts transfer requests from the processor's instruction and data cache units, from 2 slave PLB interfaces and from 4 DMA controllers, all built into the embedded block. These transfers can be directed in parallel to a high-speed memory controller interface and to a PLB master interface.

For a complete description of the Virtex-5 FPGA Embedded Block, see the *Embedded Processor Block in Virtex-5 FPGAs Reference Guide*.

The Virtex-5 Embedded Processor Block wrapper provides connectivity of the embedded block to the FPGA fabric with no intervening gate logic or storage elements. The purpose of the wrapper is to adapt the configuration parameters and some of the I/O signals of the embedded block for compatibility with the EDK design environment.

I/O Signals

The I/O signals on the Embedded Processor Block wrapper are the same as on the embedded block, except as listed in Table 1. For details on the embedded block I/O signals, refer to the *Embedded Processor Block in Virtex-5 FPGAs Reference Guide*.

Table 1: **Embedded Processor Block Wrapper I/O Signal Exceptions**

Signal Name	Interface	Signal Direction	Description
PPCMPLBMSIZE [0:1]	MPLB	Output	Driven to constant "10" (128-bits)
PPCMPLBUABUS [0:31]	MPLB	Output	Driven to all-zeros
PPCS0PLBMIRQ [0:C_SPLB0_NUM_ MASTERS-1]	SPLB0	Output	Driven to all-zeros if C_SPLB0_PROPAGATE_MIRQ = 0 (default); driven by PPCS0PLBMIRQ [0:3] output of hard block if C_SPLB0_PROPAGATE_MIRQ = 1
PLBPPCS0UABUS [0:31]	SPLB0	Input	Unconnected
PPCS1PLBMIRQ [0:C_SPLB1_NUM_ MASTERS-1]	SPLB1	Output	Driven to all-zeros if C_SPLB1_PROPAGATE_MIRQ = 0 (default); driven by PPCS1PLBMIRQ [0:3] output of hard block if C_SPLB1_PROPAGATE_MIRQ = 1
PLBPPCS1UABUS [0:31]	SPLB1	Input	Unconnected
SPLB0_Error [0:3]	(debug)	Output	Driven by PPCS0PLBMIRQ [0:3] output of hard block
SPLB1_Error [0:3]	(debug)	Output	Driven by PPCS1PLBMIRQ [0:3] output of hard block
DBGC440DEBUGHALT	(debug)	Input	OR'ed with inverse of DBGC440DEBUGHALTNEG to produce processor block DBGC440DEBUGHALT input ('0' if unconnected)
DBGC440DEBUGHALTNEG	(debug)	Input	Inverse OR'ed with DBGC440DEBUGHALT to produce processor block DBGC440DEBUGHALT input ('1' if unconnected)

Table 1: **Embedded Processor Block Wrapper I/O Signal Exceptions**

Signal Name	Interface	Signal Direction	Description
TIEC440ENDIANRESET	(tie-off)	Input	Tied per parameter C_ENDIAN_RESET
TIEC440PIR	(tie-off)	Input	Tied per parameter C_PIR
TIEC440USERRESET	(tie-off)	Input	Tied per parameter C_USER_RESET
TIEC440ICURDFETCH PLBPRIO	(tie-off)	Input	Tied per parameter C_ICU_RD_FETCH_PLB_PRIO
TIEC440ICURDSPEC PLBPRIO	(tie-off)	Input	Tied per parameter C_ICU_RD_SPEC_PLB_PRIO
TIEC440ICURDTOUCH PLBPRIO	(tie-off)	Input	Tied per parameter C_ICU_RD_TOUCH_PLB_PRIO
TIEC440DCURDLD CACHEPLBPRIO	(tie-off)	Input	Tied per parameter C_DCU_RD_LD_CACHE_PLB_PRIO
TIEC440DCURDNON CACHEPLBPRIO	(tie-off)	Input	Tied per parameter C_DCU_RD_NONCACHE_PLB_PRIO
TIEC440DCURDTOUCH PLBPRIO	(tie-off)	Input	Tied per parameter C_DCU_RD_TOUCH_PLB_PRIO
TIEC440DCURDURGENT PLBPRIO	(tie-off)	Input	Tied per parameter C_DCU_RD_URGENT_PLB_PRIO
TIEC440DCUWRFLUSH PLBPRIO	(tie-off)	Input	Tied per parameter C_DCU_WR_FLUSH_PLB_PRIO
TIEC440DCUWRSTORE PLBPRIO	(tie-off)	Input	Tied per parameter C_DCU_WR_STORE_PLB_PRIO
TIEC440DCUWRURGENT PLBPRIO	(tie-off)	Input	Tied per parameter C_DCU_WR_URGENT_PLB_PRIO
TIEDCRBASEADDR	(tie-off)	Input	Tied per parameter C_IDCR_BASEADDR(0 to 1)
TIEC440ERPNRESET	(tie-off)	Input	Tied to 0x0

Design Parameters

Table 2 lists the Embedded Processor Block wrapper parameters. Some parameters are passed directly to the embedded hard block either as instance attributes or as tie-off input signals. Others are used to modify the connectivity of the wrapper I/O signal interface.

Table 2: **Embedded Processor Block Wrapper Design Parameters**

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Unique processor ID	C_PIR[28:31]	(any 4-bit value)	0b1111	std_logic_vector
Reset value for Endian storage byte ordering	C_ENDIAN_RESET	0 = Big Endian 1 = Little Endian	0	std_logic
Reset value for user defined storage attributes: Tattribute[4:7]	C_USER_RESET[0:3]	(any 4-bit value)	0b0000	std_logic_vector
Interrupt mask for crossbar-related interrupts (initial value of DCR "IMASK")	C_INTERCONNEC_IMASK[0:31]	[Note 4]	0xFFFF_FFFF	bit_vector
Arbitration priority for all CPU fetch requests	C_ICU_RD_FETCH_PLB_PRIO[0:1]	0b00 = lowest through 0b11 = highest	0b00	std_logic_vector
Arbitration priority for all speculative CPU fetch requests	C_ICU_RD_SPEC_PLB_PRIO[0:1]	0b00 = lowest through 0b11 = highest	0b00	std_logic_vector
Arbitration priority for CPU fetch requests initiated by icbt instructions	C_ICU_RD_TOUCH_PLB_PRIO[0:1]	0b00 = lowest through 0b11 = highest	0b00	std_logic_vector
Arbitration priority for CPU cacheable load requests	C_DCU_RD_LD_CACHE_PLB_PRIO[0:1]	0b00 = lowest through 0b11 = highest	0b00	std_logic_vector
Arbitration priority for CPU non-cacheable load requests	C_DCU_RD_NONCACHE_PLB_PRIO[0:1]	0b00 = lowest through 0b11 = highest	0b00	std_logic_vector
Arbitration priority for CPU load requests initiated by dcbt instructions	C_DCU_RD_TOUCH_PLB_PRIO[0:1]	0b00 = lowest through 0b11 = highest	0b00	std_logic_vector
Arbitration priority for a CPU load request associated with an "urgent" state in which two or more CPU data cache operations are pending, waiting for a previous request to be serviced	C_DCU_RD_URGENT_PLB_PRIO[0:1]	0b00 = lowest through 0b11 = highest	0b00	std_logic_vector
Arbitration priority for CPU write requests initiated by flush instructions	C_DCU_WR_FLUSH_PLB_PRIO[0:1]	0b00 = lowest through 0b11 = highest	0b00	std_logic_vector

Table 2: **Embedded Processor Block Wrapper Design Parameters *(Contd)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Arbitration priority for CPU write requests initiated by store instructions	C_DCU_WR_STORE_PLB_PRIO[0:1]	`0b00` = lowest through `0b11` = highest	`0b00`	std_logic_vector
Arbitration priority for a CPU write request associated with an "urgent" state in which two or more CPU data cache operations are pending, waiting for a previous request to be serviced	C_DCU_WR_URGENT_PLB_PRIO[0:1]	`0b00` = lowest through `0b11` = highest	`0b00`	std_logic_vector
Arbitration priority for read/write requests initialed by DMA controller #0 (initial value of DCR "CFG_PLBS0", field "DMA0_PRI")	C_DMA0_PLB_PRIO[0:1]	`0b00` = lowest through `0b11` = highest	`0b00`	bit_vector
Arbitration priority for read/write requests initialed by DMA controller #1 (initial value of DCR "CFG_PLBS0", field "DMA1_PRI")	C_DMA1_PLB_PRIO[0:1]	`0b00` = lowest through `0b11` = highest	`0b00`	bit_vector
Arbitration priority for read/write requests initialed by DMA controller #2 (initial value of DCR "CFG_PLBS1", field "DMA2_PRI")	C_DMA2_PLB_PRIO[0:1]	`0b00` = lowest through `0b11` = highest	`0b00`	bit_vector
Arbitration priority for read/write requests initialed by DMA controller #3 (initial value of DCR "CFG_PLBS1", field "DMA3_PRI")	C_DMA3_PLB_PRIO[0:1]	`0b00` = lowest through `0b11` = highest	`0b00`	bit_vector
Base address (word-aligned) of DCR register block internal to the embedded block	C_IDCR_BASEADDR[0:9]	`0b00_0000_0000`, `0b01_0000_0000`, `0b10_0000_0000`, `0b11_0000_0000`	`0b11_1111_1111` [Note 1]	std_logic_vector
High address of DCR register block internal to the embedded block	C_IDCR_HIGHADDR[0:9]	C_IDCR_BASEADDR + `0b00_1111_1111`	`0b00_0000_0000` [Note 1]	std_logic_vector
Enables generation of timing constraints for proper synchronization of SPLB MBusy output signals to the PLB clock [Note 9]	C_GENERATE_PLB_TIMESPECS	0 = disable 1 = enable	1	integer
APU Controller				

Table 2: **Embedded Processor Block Wrapper Design Parameters *(Contd)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Initializes 17 of the bits of the APU Control DCR (see Table 3)	C_APU_ CONTROL[0:16]	(see table)	0b000100000000 00000	bit_ vector
Initializes 24 of the bits of UDI #0 through UDI #15 configuration register (see Table 4)	C_APU_UDI_0[0:23] through C_APU_UDI_15[0:23]	(see table)	0x000000	bit_ vector
Memory Controller Interface				
Base address of the memory connected to the PPC440MC interface	C_PPC440MC_ADDR _BASE[0:31]	[Note 2]	0xFFFFFFFF	std_logic_ vector
High address of the memory connected to the PPC440MC interface [Note 2]	C_PPC440MC_ADDR _HIGH[0:31]	[Note 2]	0x00000000	std_logic_ vector
Mask used to determine if there is a row conflict between this transaction and previous transaction (initial value of DCR "MI_ROWCONFLICT_MASK")	C_PPC440MC _ROW_CONFLICT_ MASK[0:31]		0x00000000	bit_ vector
Mask used to determine if there is a bank conflict between this transaction and previous transaction (initial value of DCR "MI_BANKCONFLICT_MASK")	C_PPC440MC_ BANK_CONFLICT_ MASK[0:31]		0x00000000	bit_ vector
Control and configuration for the memory controller interface (initial value of DCR "MI_CONTROL")	C_PPC440MC_ CONTROL[0:31]	[Note 4]	0x0000008f	bit_ vector
Secondary arbitration priority for all instruction fetches requested by the CPU for the MC interface (initial value of DCR "ARB_XBM", field "440ICUR")	C_PPC440MC_ PRIO_ICU	0-4 [Note 4, 5]	4	integer
Secondary arbitration priority for all data writes requested by the CPU for the MC interface (initial value of DCR "ARB_XBM", field "440DCUW")	C_PPC440MC_ PRIO_DCUW	0-4 [Note 4, 5]	3	integer

Table 2: **Embedded Processor Block Wrapper Design Parameters *(Contd)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Secondary arbitration priority for all data reads requested by the CPU for the MC interface (initial value of DCR "ARB_XBM", field "440DCUR")	C_PPC440MC _PRIO_DCUR	0-4 [Note 4, 5]	2	integer
Secondary arbitration priority for all transactions requested by SPLB1, DMA2 or DMA3 for the MC interface (initial value of DCR "ARB_XBM", field "PLBS1")	C_PPC440MC_ PRIO_SPLB1	0-4 [Note 4, 5]	0	integer
Secondary arbitration priority for all transactions requested by SPLB0, DMA0 or DMA1 for the MC interface (initial value of DCR "ARB_XBM", field "PLBS0")	C_PPC440MC_ PRIO_SPLB0	0-4 [Note 4, 5]	1	integer
MC interface arbitration mode (initial value of DCR "ARB_XBM", field "MODE")	C_PPC440MC_ARB_ MODE	0 = Least Recently Used (LRU) 1 = round-robin 2= fixed priority	0	integer
Maximum number of quad-words per burst through crossbar to MC interface (used to derive the initial value of DCR "CFG_PLBS0" and "CFG_PLBS1", fields "THRMIB" and "THWMIB")	C_PPC440MC_ MAX_BURST	1, 2, 4, 8, 16	8	integer
MPLB Interface				
MPLB Address bus width (ignored by wrapper)	C_MPLB_AWIDTH	32 [Note 2]	32	integer
MPLB Data bus width (ignored by wrapper)	C_MPLB_DWIDTH	128 [Note 2]	128	integer
Master size of MPLB on PLB bus (ignored by wrapper)	C_MPLB_NATIVE_ DWIDTH	128 (constant)	128	integer
Secondary arbitration priority for all instruction fetches requested by the CPU for the MPLB interface (initial value of DCR "ARB_XBC", field "440ICUR")	C_MPLB_PRIO_ICU	0-4 [Note 4, 6]	4	integer

Table 2: **Embedded Processor Block Wrapper Design Parameters *(Contd)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Secondary arbitration priority for all data writes requested by the CPU for the MPLB interface (initial value of DCR "ARB_XBC", field "440DCUW")	C_MPLB_PRIO_ DCUW	0-4 [Note 4, 6]	3	integer
Secondary arbitration priority for all data reads requested by the CPU for the MPLB interface (initial value of DCR "ARB_XBC", field "440DCUR")	C_MPLB_PRIO_ DCUR	0-4 [Note 4, 6]	2	integer
Secondary arbitration priority for all transactions requested by SPLB1, DMA2 or DMA3 for the MPLB interface (initial value of DCR "ARB_XBC", field "PLBS1")	C_MPLB_PRIO_ SPLB1	0-4 [Note 4, 6]	0	integer
Secondary arbitration priority for all transactions requested by SPLB0, DMA0 or DMA1 for the MPLB interface (initial value of DCR "ARB_XBC", field "PLBS0")	C_MPLB_PRIO_ SPLB0	0-4 [Note 4, 6]	1	integer
MPLB interface arbitration mode (initial value of DCR "ARB_XBC", field "MODE") [Note 10]	C_MPLB_ARB_ MODE	0 = Least Recently Used (LRU) 1 = round-robin 2= fixed priority	0	integer
Allow MBusy to block MPLB when Tattribute[7] ("Sync") is asserted (initial value of DCR "ARB_XBC", field "SYNCTATTR")	C_MPLB_SYNC_ TATTRIBUTE	0 = disable Sync 1 = enable Sync	0	integer
Maximum number of quad-words per burst through crossbar to MPLB interface (used to derive the initial value of DCR "CFG_PLBS0" and "CFG_PLBS1", fields "THRPLBM" and "THWPLBM")	C_MPLB_MAX_ BURST	1, 2, 4, 8, 16	8	integer
Allow locked transfers on MPLB (initial value of DCR "CFG_PLBM", field "LOCKXFER")	C_MPLB_ALLOW_ LOCK_XFER	0 = disallow, 1 = allow	1	integer

Table 2: **Embedded Processor Block Wrapper Design Parameters *(Contd)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Allow read address pipelining on MPLB (initial value of DCR "CFG_PLBM", field "RPIPE") [Note 10]	C_MPLB_READ_ PIPE_ENABLE	0 = disallow, 1 = allow	1	integer
Allow write address pipelining on MPLB (initial value of DCR "CFG_PLBM", field "WPIPE")	C_MPLB_WRITE_ PIPE_ENABLE	0 = disallow, 1 = allow	1	integer
Allow posted writes on MPLB and SPLB interfaces (initial value of DCRs "CFG_PLBM", "CFG_PLBS0" and "CFG_PLBS1"; field "WPOST")	C_MPLB_WRITE_ POST_ENABLE	0 = disallow, 1 = allow	1	integer
Point-to-Point interconnect mode on MPLB (ignored by wrapper)	C_MPLB_P2P	0 [Note 2, 7]	0	integer
SPLB0 Interface				
SPLB0 Address bus width (ignored by wrapper)	C_SPLB0_AWIDTH	32 [Note 2]	32	integer
SPLB0 Data bus width (ignored by wrapper)	C_SPLB0_DWIDTH	128 [Note 2]	128	integer
Slave size of SPLB0 on PLB bus (ignored by wrapper)	C_SPLB0_NATIVE_ DWIDTH	128 (constant)	128	integer
SPLB0 support for burst transfers (ignored by wrapper)	C_SPLB0_ SUPPORT_BURSTS	1 (constant)	1	integer
Include C_SPLB0_RNG*_MPLB ranges in SPLB0 decode	C_SPLB0_USE_ MPLB_ADDR	0 = exclude, 1 = include	0	integer
Number of valid C_SPLB0_RNG*_MPLB ranges	C_SPLB0_NUM_ MPLB_ADDR_RNG	0-4	0	integer
Base address of SPLB0 access to MC interface (used to derive value of DCR "TMPL0_PLBS0_MAP")	C_SPLB0_RNG_MC_ BASEADDR[0:31]	[Note 3, 8]	0xFFFF_FFFF [Note 1]	std_logic_ vector
High address of SPLB0 access to MC interface (used to derive value of DCR "TMPL0_PLBS0_MAP")	C_SPLB0_RNG_MC_ HIGHADDR[0:31]	[Note 3, 8]	0x0000_0000 [Note 1]	std_logic_ vector

Table 2: **Embedded Processor Block Wrapper Design Parameters *(Contd)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Base address of SPLB0 access to MPLB interface (used to derive value of DCR "TMPL0_PLBS0_MAP" if C_SPLB0_USE_MPLB_ADDR = 1)	C_SPLB0_RNG0_MPLB_ BASEADDR[0:31] through C_SPLB0_RNG3_MPLB_ BASEADDR[0:31]	[Note 3, 8]	0xFFFF_FFFF [Note 1]	std_logic_ vector
High address of SPLB0 access to MC interface (used to derive value of DCR "TMPL0_PLBS0_MAP" if C_SPLB0_USE_MPLB_ADDR = 1)	C_SPLB0_RNG0_MPLB_HIGHADDR[0:31] through C_SPLB0_RNG3_MPLB_HIGHADDR[0:31]	[Note 3, 8]	0x0000_0000 [Note 1]	std_logic_ vector
Number of masters connected to SPLB0]	C_SPLB0_NUM_ MASTERS	1-4 [Note 2]	1	integer
Width of MasterID bus on SPLB0	C_SPLB0_MID_WIDTH	1-2 [Note 2]	1	integer
Allow locked transfers on SPLB0 (initial value of DCR "CFG_PLBS0", field "LOCKXFER")	C_SPLB0_ALLOW_ LOCK_XFER	0 = disallow, 1 = allow	1	integer
Allow read address pipelining on SPLB0 (initial value of DCR "CFG_PLBS0", field "RPIPE")	C_SPLB0_READ_ PIPE_ENABLE	0 = disallow, 1 = allow	1	integer
Propagate MIRQ signals from crossbar onto SPLB0 bus	C_SPLB0_ PROPAGATE_MIRQ	0 = disable, 1 = enable	0	integer
Point-to-Point interconnect mode on SPLB0. Currently used only to detect whether the SPLB0 interface is connected.	C_SPLB0_P2P	0 = shared bus 1= point-to-point -1 = unconnected [Note 2]	-1	integer
SPLB1 Interface				
SPLB1 Address bus width (ignored by wrapper)	C_SPLB1_AWIDTH	32 [Note 2]	32	integer
SPLB1 Data bus width (ignored by wrapper)	C_SPLB1_DWIDTH	128 [Note 2]	128	integer
Slave size of SPLB1 on PLB bus (ignored by wrapper)	C_SPLB1_NATIVE_ DWIDTH	128 (constant)	128	integer
SPLB1 support for burst transfers (ignored by wrapper)	C_SPLB1_SUPPORT _BURSTS	1 (constant)	1	integer

Table 2: **Embedded Processor Block Wrapper Design Parameters *(Contd)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Include C_SPLB1_RNG*_MPLB ranges in SPLB1 decode	C_SPLB1_USE_ MPLB_ADDR	0 = exclude, 1 = include	0	integer
Number of valid C_SPLB1_RNG*_MPLB ranges	C_SPLB1_NUM_ MPLB_ADDR_RNG	0-4	0	integer
Base address of SPLB1 access to MC interface (used to derive value of DCR "TMPL0_PLBS1_MAP")	C_SPLB1_RNG_MC_ BASEADDR[0:31]	[Note 3, 8]	0xFFFF_FFFF [Note 1]	std_logic_ vector
High address of SPLB1 access to MC interface (used to derive value of DCR "TMPL0_PLBS1_MAP")	C_SPLB1_RNG_MC_ HIGHADDR[0:31]	[Note 3, 8]	0x0000_0000 [Note 1]	std_logic_ vector
Base address of SPLB1 access to MPLB interface (used to derive value of DCR "TMPL0_PLBS1_MAP" if C_SPLB1_USE_MPLB_AD DR = 1)	C_SPLB1_RNG0_MP LB_BASEADDR[0:31] through C_SPLB1_RNG3_MP LB_BASEADDR[0:31]	[Note 3, 8]	0xFFFF_FFFF [Note 1]	std_logic_ vector
High address of SPLB1 access to MC interface (used to derive value of DCR "TMPL0_PLBS1_MAP" if C_SPLB1_USE_MPLB_AD DR = 1)	C_SPLB1_RNG0_MP LB_HIGHADDR[0:31] through C_SPLB1_RNG3_MP LB_HIGHADDR[0:31]	[Note 3, 8]	0x0000_0000 [Note 1]	std_logic_ vector
Number of masters connected to SPLB1]	C_SPLB1_ NUM_MASTERS	1-4 [Note 2]	1	integer
Width of MasterID bus on SPLB1	C_SPLB1_MID_ WIDTH	1-2 [Note 2]	1	integer
Allow locked transfers on SPLB1 (initial value of DCR "CFG_PLBS1", field "LOCKXFER")	C_SPLB1_ALLOW_ LOCK_XFER	0 = disallow, 1 = allow	1	integer
Allow read address pipelining on SPLB1 (initial value of DCR "CFG_PLBS1", field "RPIPE")	C_SPLB1_READ_ PIPE_ENABLE	0 = disallow, 1 = allow	1	integer
Propagate MIRQ signals from crossbar onto SPLB1 bus	C_SPLB1_PROPAGAT E_ MIRQ	0 = disable, 1 = enable	0	integer

Table 2: **Embedded Processor Block Wrapper Design Parameters *(Contd)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Point-to-Point interconnect mode on SPLB1. Currently used only to detect whether the SPLB1 interface is connected.	C_SPLB1_P2P	0 = shared bus 1= point-to-point -1 = unconnected [Note 2]	-1	integer
DMA0 Through DMA3 Interfaces				
Number of DMA channels used in the design. Used to initially set the following DCR fields: CFG_PLBS0:DMA0_EN (if P1), CFG_PLBS0:DMA1_EN (if P2), CFG_PLBS1:DMA2_EN (if P3), CFG_PLBS1:DMA3_EN (if =4),	C_NUM_DMA	0-4	0	integer
DMA #0 through DMA #3 transmit channel control (initial value of TX Channel Control Register)	C_DMA0_TXCHANNELCTRL[0:31] through C_DMA3_TXCHANNELCTRL[0:31]	[Note 4]	`0x0101_0000`	bit_ vector
DMA #0 through DMA #3 receive channel control (initial value of RX Channel Control Register)	C_DMA0_RX CHANNELCTRL[0:31] through C_DMA3_RXCHANNEL CTRL[0:31]	[Note 4]	`0x0101_0000`	bit_ vector
DMA #0 through DMA #3 control register (bits 2:7 set the initial value of DMA Control Register bits 26:31)	C_DMA0_CONTROL[0:7] through C_DMA3_ CONTROL[0:7]	[Note 4]	`0b0000_0000`	bit_vector
DMA #0 through DMA #3 Transmit IRQ coalescing clock divider ratio	C_DMA0_TXIRQ TIMER[0:9] through C_DMA3_TXIRQ TIMER[0:9]	`0b0000000000` = divide-by-1, through `0b1111111111` = divide-by-1024	`0b11_1111_1111`	bit_vector
DMA #0 through DMA #3 Receive IRQ coalescing clock divider ratio	C_DMA0_RXIRQ TIMER[0:9] through C_DMA0_RXIRQ TIMER[0:9]	0b0000000000 = divide-by-1, through 0b1111111111 = divide-by-1024	`0b11_1111_1111`	bit_ vector
DCR Interface				
Enable the auto-lock feature for the DCR indirect mode	C_DCR_AUTOLOCK_ ENABLE	0-1	1	integer

Table 2: **Embedded Processor Block Wrapper Design Parameters *(Contd)***

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Synchronization mode for the external MDCR interface	C_PPCDM_ ASYNCMODE	0=Synchronous, 1=Asynchronous	0	integer
Synchronization mode for the external SDCR interface	C_PPCDS_ASYNCM ODE	0=Synchronous, 1=Asynchronous	0	integer

Notes:

1. Default values for base/high address pair are to insure that the parameters are explicitly set by the user. If the user does not override the default values, the tools will generate an error.
2. These parameters are calculated and automatically assigned by the EDK XPS tools during the system creation process. Values for these parameters should not be specified by the user.
3. The size of an address range (HIGHADDR - BASEADDR + 1) must be a power of 2, and BASEADDR must be a multiple of that size.
4. Refer to the description of the corresponding DCR register in the *Embedded Processor Block in Virtex-5 FPGAs Reference Guide*.
5. The five C_PPC440MC_PRIO parameters must have mutually-unique values.
6. The five C_MPLB_PRIO parameters must have mutually-unique values.
7. Point-to-Point mode is not supported on the crossbar MPLB interface.
8. Each SPLB address range (if used) must be minimum 128MB.
9. If the Interconnect-to-PLB clock ratio is > 1:1 and your design uses any of the MBusy outputs on either SPLB interface (PPCS*PLBMBUSY), then the C_GENERATE_PLB_TIMESPECS parameter should remain enabled. Otherwise, an undetected timing violation may occur on the MBusy outputs.
10. If parameter C_MPLB_ARB_MODE is set to 1 (round-robin), parameter C_MPLB_READ_PIPE_ENABLE should not be set to 0. Otherwise, one crossbar master may monopolize the MPLB bus under certain conditions.

Table 3: **APU Control Register Initialization**

Field Name	Control Register Bits	C_APU_ CONTROL bits	Default
LD/ST Decode Disable	5	0	0
UDI Decode Disable	6	1	0
Force UDI Non-auton, late confirm	7	2	0
FPU Decode Disable	8	3	1
FPU Complex Arith. Disable	9	4	0
FPU Convert Disable	10	5	0
FPU Estimate/Select Disable	11	6	0
FPU Single Precision Disable	12	7	0
FPU Double Precision Disable	13	8	0
FPU FPSCR Disable	14	9	0
Force FPU Non-auton, late confirm	15	10	0
Store WriteBack OK	16	11	0
Ld/St Priv. Op	17	12	0
Force Align	20	13	0

Table 3: **APU Control Register Initialization** ***(Contd)***

Field Name	Control Register Bits	C_APU_ CONTROL bits	Default
LE Trap	21	14	0
BE Trap	22	15	0
FCM Enable	31	16	0

Table 4: **UDI Configuration Register Initialization**

Field Name	Control Register Bits	C_APU_UDI bits
Primary Op-code	0	0
Extended Op-code	1:11	1:11
Privilege Op	12	12
Ra Enable	13	13
Rb Enable	14	14
GPR Write	15	15
CR Enable	16	16
CRField[0:2]	18:20	17:19
Type	26:27	20:21
Wildcard	30	22
En	31	23

Table 5: **DCR Fields Initialized to Constant Values**

DCR Name	Field	Value
CFG_PLBM, CFG_PLBS0, CFG_PLBS1	LOCK_SESR	1
CFG_PLBM	XBAR_PRIORITY_ENA	1
CFG_PLBM	SL_ETERM_MODE	0
CFG_PLBS0, CFG_PLBS1	ADDRACK_DLY	1
TMPL_SEL_REG		0X3FFFFFFF
TMPL1_XBAR_MAP, TMPL2_XBAR_MAP, TMPL3_XBAR_MAP		0X00000000
TMPL1_PLBS0_MAP, TMPL2_PLBS0_MAP, TMPL3_PLBS0_MAP		0X00000000
TMPL1_PLBS1_MAP, TMPL2_PLBS1_MAP, TMPL3_PLBS1_MAP		0X00000000

Reference Documents

1. UG200 Embedded Processor Block in Virtex-5 FPGAs Reference Guide

Revision History

Date	Version	Revision
4/7/08	1.0	Initial Xilinx release.
9/22/08	1.1	Incorporated CR473090; corrected PDF properties, hyperlinks in Ref Doc section, trademark/registration symbol usage, legal footer; and made minor content edits.

Notice of Disclaimer

Xilinx is providing this design, code, or information (collectively, the “Information”) to you “**AS-IS**” with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.