Processor IP Virtex-4 FPGA Embedded Processor Block with PowerPC 405 Processor (v2.01b)

DS306 February 10, 2009 **Product Specification**

Introduction

This document describes the wrapper for the Virtex®-4 FPGA embeddesd processor block. For details regarding the Virtex-4 embedded block, see the *Embedded Processor Block in Virtex-4 FPGAs Reference Guide*.

Features

- Processor Local Bus (PLB) version 4.6 interfaces
- Dual Instruction-side and dual data-side PLB interfaces, with user-selectable address ranges and clock frequency translation
- Instruction-side and data-side On-Chip Memory (OCM) interfaces, with user-selectable address ranges
- PowerPC® 405 Auxiliary Processor Unit (APU) controller interface, with User-Defined Instruction decoding (UDI)

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex-4	
Version of Core	ppc405_virtex4	v2.01b
Resources Used		
	Min	Max
Slices	145	280
LUTs	125	420
FFs	225	390
Block RAMs	0	0
Special Features	PPC405_ADV	
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	N/A	
Design Tool Requirements		
Xilinx Implementation Tools	12.1	
Verification	ModelSim v6.2c and above	
Simulation	ModelSim v6.2c and above	
Synthesis	XST	
Support		
Provided by Xilinx, Inc.		

Functional Description

The embedded block in Virtex-4 FX FPGA devices contains the PowerPC 405 F6 processor core, on-chip memory (OCM) logic, and an APU controller. For a complete description of the PowerPC 405 Processor Block, see the *PowerPC 405 Processor Block Reference Guide*.

The Virtex-4 Embedded Processor wrapper adapts the configuration parameters and the I/O signals of the processor block for compatibility with the EDK design environment. Except for the PLB interfaces, the wrapper provides only connectivity of the processor block to the FPGA fabric with no intervening gate logic or storage elements. For PLB interfaces, the wrapper provides fabric-based logic to translate the PLB V3.4 interface signals on the processor block to PLB v4.6 signals used by various peripheral cores in the EDK system. The wrapper further replicates each of the I-side and D-side PLB interfaces on the processor block, presenting 2 parallel bus master interfaces for each side at the wrapper interface. This configuration allows for independent connection of the wrapper to a memory controller core, separately from the connection to shared peripherals, thus reducing bus contention.

Instruction Side PLB v4.6 Translation

The primary functions performed by the wrapper to translate the processor's Instruction Cache Unit (ICU) transfers from PLB v3.4 to PLB v4.6 consist of abort suppression and time-out sequencing.

The ICU issues only cache-line read transfers for 4-word or 8-word lines. The ICU may also abort transfers during normal operation. However, the Xilinx EDK system and its library of peripheral cores support a simplified subset of PLB v4.6 protocol that does not include aborted transfers. The Virtex-4 Embedded Processor wrapper suppresses any aborts issued by the ICU, preventing the abort condition from appearing on the IPLBn interface. To accomplish this, the wrapper always allows the targeted slave to respond to the read originally requested by the ICU (even if aborted during the same cycle as the request), but blocks the incoming data transfer signals (PLB_RdDAck) from reaching the processor. Until an aborted transfer has completed on the bus, the wrapper blocks any further requests from the ICU from going out onto the bus. If the ICU issues a pipelined request (before a prior read transfer has completed), and then aborts the request, the wrapper waits for the first transfer to complete before it begins blocking the data transfer resulting from the pipelined request. To accomplish this, the wrapper monitors all ICU read transfers to keep track of to which request each data transfer cycle belongs.

PLB v4.6 protocol introduces a new time-out error signal that is asserted by the PLB arbiter if no slave responds within 16 cycles of broadcasting a request. The PLB v3.4 ICU interface of the processor, however, expects all of its requests to result in the expected number of data transfer cycles. To satisfy the processor, the wrapper generates a dummy acknowledge (PLB_MAddAck) followed by a dummy sequence of the expected number of data read cycles, while asserting the bus error signal (PLB_MErr) back to the processor. Until the dummy read transfer sequence is completed, the wrapper blocks any further requests from the ICU from going out onto the bus. If the processor aborts a request, and that request also times out on the bus, the wrapper skips the dummy sequence and does not block any data received from the bus (abort and time-out cancel each other out).

Data Side PLB v4.6 Translation

The primary function performed by the wrapper in translating the Data Cache Unit (DCU) transfers of the processor from PLB v3.4 to PLB v4.6 consists of time-out sequencing. Because the DCU does not abort any transfers during normal operation, no abort suppression is implemented. The wrapper derives the bus error signal (PLB_MErr) to the processor by logically OR-ing the individual PLB_MRdErr and PLB_MWrErr signals introduced by the PLB v4.6 interface.

It is not necessary for the wrapper to monitor all DCU data transfer cycles to perform time-out sequencing. Instead, it monitors the PLB_MBusy signal from the bus to determine when it is safe to begin generating a dummy read or write data transfer sequence in response to a time-out condition on the bus. As on the I-side, the wrapper blocks any further requests from the DCU from going out onto the bus until the dummy data transfer sequence is completed.

PLB Port Replication

The PowerPC 405 Processor Wrapper has two I-side, plus two D-side, PLB interfaces to help reduce bus contention. IPLB0 and DPLB0 are intended for **shared** peripheral connections, where more widely varying latencies may be tolerated. IPLB1 and DPLB1 are intended for **point-to-point** connection to a memory controller. The wrapper steers each request from the processor ICU onto one of the two IPLB interfaces, and similarly for the D-side, based on the address of the request. A base/high address parameter pair defines a single, binary-aligned address range for the IPLB1 interface. These address parameters are normally set automatically by EDK tools to match the address range of the connected memory controller. The IPLB1 interface may only connect to a bus that has exactly one slave. (If the interface of the slave has more than one address range associated with it, such as on a bridge component, no more than one range may be valid.) Any ICU request that falls within this address range is directed to IPLB1; otherwise it goes to IPLB0. A similar base/high address pair defines the response range for the DPLB1 interface, with all out-of-range DCU requests being steered to DPLB0. If the ISOCM or DSOCM interfaces are used, their address ranges take precedence; such requests do not appear on the PLB interface of the processor.

If the processor ICU issues a pipelined request, and the address maps to the same IPLB interface as the transfer currently in progress, the pipelined request is allowed to go out onto the bus immediately. However, if the address of the pipelined request maps to the opposite IPLB interface, the wrapper holds it back from the bus until the current transfer completes (PLB_MBusy goes low). This is necessary to prevent data transfers from occurring out-of-order with respect to their requests. A similar interlock governs pipelined requests from the DCU onto the DPLB interfaces.

Each of the 4 PLB interfaces may run at a different clock frequency. The processor block has one clock input (PLBCLK) used to synchronize both its ICU and DCU. This processor PLBCLK must be clocked using the fastest of all the connected PLB bus clocks. The wrapper automatically determines which of the connected PLB buses has the highest frequency, based on clock frequency information expressed in the design (CLK_FREQ_HZ values in the MHS). If frequency information is not present in the design, then the C_FASTEST_PLB_CLOCK parameter must be specified if either the IPLB1 or DPLB1 interface is used, otherwise an error will result. Any PLB bus that runs at a slower frequency must have a clock period that is an integer multiple of the fastest PLB clock, and must be rising-edge-aligned. The wrapper automatically synchronizes all slower PLB interfaces to the selected PLBCLK. In a typical application, the PLB buses connecting IPLB1 and DPLB1 to a memory controller may run at 2 or 3 times the frequency of the shared peripheral buses connected to IPLB0 and DPLB0. The CPU clock (CPMC405CLOCK) may also typically run 2 or 3 times faster than the IPLB1/DPLB1 clock.

PowerPC 405 Processor/Virtex-4 Parameters

Table 1 lists the parameters on the wrapper. Some signal are passed directly to the processor hard block as tie-off input signals, while others are used to modify the connectivity of the wrapper I/O signal interface.

Table 1: **Embedded Processor Block Wrapper Parameters**

Parameter	Description	Default	Type
C_IPLB1_ADDR_BASE	Base address of the slave (memory controller) connected to the IPLB1 interface. (2) (3) Any PLB accesses by the processor ICU to addresses outside the range defined by C_IPLB1_ADDR_BASE and C_IPLB1_ADDR_HIGH will be directed to the IPLB0 interface.	0xFFFFFFFF	std_logic_vector (0 to 31)
C_IPLB1_ADDR_HIGH	High address of the slave connected to the IPLB1 interface. (2) (3)	0x00000000	std_logic_vector (0 to 31)
C_DPLB1_ADDR_BASE	Base address of the slave (memory controller) connected to the DPLB1 interface. (2) (3) Any PLB accesses by the processor DCU to addresses outside the range defined by C_DPLB1_ADDR_BASE and C_DPLB1_ADDR_HIGH will be directed to the DPLB0 interface.	0xFFFFFFFF	std_logic_vector (0 to 31)
C_DPLB1_ADDR_HIGH	High address of the slave connected to the DPLB1 interface. (2) (3)	0x00000000	std_logic_vector (0 to 31)
C_IPLB0_DWIDTH	Data bus width of PLB bus connected to IPLB0. (2) (3)	64	integer
C_IPLB0_NATIVE_DWIDTH	Master size of IPLB0 on PLB bus (ignored by wrapper)	64 (constant)	integer
C_IPLB1_DWIDTH	Data bus width of PLB bus connected to IPLB1(1) [Note 1]	64	integer
C_IPLB1_NATIVE_DWIDTH	Master size of IPLB1 on PLB bus (ignored by wrapper)	64 (constant)	integer
C_DPLB0_DWIDTH	Data bus width of PLB bus connected to DPLB0 (1)	64	integer
C_DPLB0_NATIVE_DWIDTH	Master size of DPLB0 on PLB bus (ignored by wrapper)	64 (constant)	integer
C_DPLB1_DWIDTH	Data bus width of PLB bus connected to DPLB1(1)	64	integer
C_DPLB1_NATIVE_DWIDTH	Master size of DPLB1 on PLB bus (ignored by wrapper)	64 (constant)	integer
C_FASTEST_PLB_CLOCK	Specifies which PLB interface clock runs at the highest frequency, and is to be connected to the PLBCLK input to the processor block (synchronizes both the ICU and DCU). (4)	"DEFAULT"	string ("IPLB0" or "IPLB1" or "DPLB0" or DPLB1")
C_IDCR_BASEADDR	Bits [0:5] drive the input port TIEDCRADDR. Bits [6:9] must be all zeros. (2)	0x3FF	std_logic_vector (0 to 9)
C_IDCR_HIGHADDR	Bits [0:5] must be the same as C_IDCR_BASEADDR[0:5]. Bits [6:9] must be all ones. (2)	0x000	std_logic_vector (0 to 9)
C_MMU_ENABLE	Drives the input pin TIEC405MMUEN	1	integer

Table 1: **Embedded Processor Block Wrapper Parameters *(Cont'd)***

Parameter	Description	Default	Type
C_DETERMINISTIC_ MULT	Drives the input pin TIEC405DETERMINISTICMULT	0	integer
C_DISABLE_ OPERAND_ FORWARDING	Drives the input pin TIEC405DISOPERANDFWD	1	integer
C_PLBSYNCBYPASS	Drives the input pin CPMC405SYNCBYPASS	1	integer
C_APU_CONTROL	Drives the input port TIEAPUCONTROL. (See Table 3)	0xDE00	std_logic_vector (0 to 15)
C_APU_UDI1	Drives the input port TIEAPUUDI1. (See Table 4)	0xA18983	std_logic_vector (0 to 23)
C_APU_UDI2	Drives the input port TIEAPUUDI2. (See Table 4)	0xA38983	std_logic_vector (0 to 23)
C_APU_UDI3	Drives the input port TIEAPUUDI3. (See Table 4)	0xA589C3	std_logic_vector (0 to 23)
C_APU_UDI4	Drives the input port TIEAPUUDI4. (See Table 4)	0xA789C3	std_logic_vector (0 to 23)
C_APU_UDI5	Drives the input port TIEAPUUDI5. (See Table 4	0xA98C03	std_logic_vector (0 to 23)
C_APU_UDI6	Drives the input port TIEAPUUDI6. (See Table 4)	0xAB8C03	std_logic_vector (0 to 23)
C_APU_UDI7	Drives the input port TIEAPUUDI7. (See Table 4)	0xAD8C43	std_logic_vector (0 to 23)
C_APU_UDI8	Drives the input port TIEAPUUDI8. (See Table 4)	0xAF8C43	std_logic_vector (0 to 23)
C_PVRHIGH	Drives the input pins TIEPVRBIT8, TIEPVRBIT9, TIEPVRBIT10, TIEPVRBIT11.	0x0	std_logic_vector (0 to 3)
C_PVRLOW	Drives the input pins TIEPVRBIT28, TIEPVRBIT29, TIEPVRBIT30, TIEPVRBIT31.	0x0	std_logic_vector (0 to 3)
C_GENERATE_PLB_ TIMESPECS	Enables generation of timing constraints for proper synchronization of DPLB rdDAck input to the processor block and for relaxed multi-cycle timing constraints on any slower PLB buses.	1	integer

Notes:

1. 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.
2. The size of the address range (HIGH - BASE + 1) must be a power of 2, and BASE must be a multiple of that size (aligned). Default values for base/high address pair are to insure that the parameters get set to a valid address range if the interface is connected; otherwise the tools will generate an error.
3. These parameters are calculated and automatically assigned by the EDK XPS tools during the system creation process, if not explicitly specified by the user. Users may choose to set the address range wider than that of the connected slave in order to reduce the size and delay of the address decode logic, provided no other slaves fall within the specified range.
4. Automatically assigned by the EDK XPS tools during the system creation process, if possible. Otherwise, must be specified by the user if either the IPLB1 or DPLB1 interface is used.

I/O Signals

The I/O signals on the wrapper are the same as on the processor block, except for static *tie-off* inputs, which are set by configuration parameters, as well as by the signal exceptions listed in Table 2. For details on the embedded block I/O signals, see the *PowerPC405 Processor Block Reference Guide*.

Table 2: **Embedded Processor Block Wrapper Signal Exceptions**

Wrapper Port Name	Processor Block Connection	I/O Type	Function
IPLB*n*_PLB_Clk DPLB*n*_PLB_Clk	PLBCLK	I	Clock for each PLB bus interface. The fastest of these clocks (C_FASTEST_PLB_CLOCK) connects to the processor block's PLBCLK input, which synchronizes the ICU and DCU. All other bus clocks must run at an integer multiple of the fastest clock's period, and must be rising-edge-aligned.
IPLB*n*_PLB_Rst DPLB*n*_PLB_Rst	N/A	I	Resets only the wrapper logic associated with each PLB interface.
IPLB*n*_PLB_MAddrAck DPLB*n*_PLB_MAddrAck	PLBC405ICUADDRACK PLBC405DCUADDRACK	I	Indicates a PLB slave acknowledges the current bus access request.
IPLB*n*_PLB_MSSize[0:1] DPLB*n*_PLB_MSSize[0:1]	PLBC405ICUSSIZE1 PLBC405DCUSSIZE1	I	Indicates the native bus width (size) of the PLB slave that accepted the request.
IPLB*n*_PLB_MRdDAck DPLB*n*_PLB_MRdDAck	PLBC405DCURDDACK	I	Indicates the read-data bus contains valid data that the processor must latch during the current clock cycle.
IPLB*n*_PLB_MRdDBus [0:*width*-1] DPLB*n*_PLB_MRdDBus [0:*width*-1]	PLBC405DCURDDBUS[0:63]	I	Read data bus. Only the upper 64-bits are sampled when connected to a 128-bit PLB bus.
IPLB*n*_PLB_MRdWdAddr[0:3] DPLB*n*_PLB_MRdWdAddr[0:3]	PLBC405ICURDWDADDR[1:3] PLBC405DCURDWDADDR[1:3]	I	Indicates which word (or double-word) of a cache-line read transfer is present on the read-data bus.
IPLB*n*_PLB_MWrDAck	N/A	I	Ignored.
DPLB*n*_PLB_MWrDAck	PLBC405DCUWRDACK	I	Indicates that data on the DCU write-data bus is being accepted by the PLB slave.
IPLB*n*_PLB_MBusy DPLB*n*_PLB_MBusy	PLBC405ICUBUSY PLBC405DCUBUSY	I	Indicates the PLB slave is busy performing an operation requested by the processor.
IPLB*n*_PLB_MRdErr DPLB*n*_PLB_MRdErr	PLBC405ICUERR PLBC405DCUERR	I	Indicates an error was detected by the PLB slave during a read transfer. Processor PLBC405DCUERR input is the logical-OR of DPLBn_PLB_MRdErr and DPLBn_PLB_MWrErr.
IPLB*n*_PLB_MWrErr	N/A	I	Ignored

Table 2: **Embedded Processor Block Wrapper Signal Exceptions *(Cont'd)***

Wrapper Port Name	Processor Block Connection	I/O Type	Function
DPLB*n*_PLB_MWrErr	PLBC405DCUERR	I	Indicates an error was detected by the PLB slave during a write transfer. Processor PLBC405DCUERR input is the logical-OR of DPLB*n*_PLB_MRdErr and DPLB*n*_PLB_MWrErr.
IPLB*n*_PLB_MRdBTerm DPLB*n*_PLB_MRdBTerm	N/A	I	Ignored
IPLB*n*_PLB_MRearbitrate DPLB*n*_PLB_MRearbitrate	N/A	I	Ignored
IPLB*n*_PLB_MWrBTerm DPLB*n*_PLB_MWrBTerm	N/A	I	Ignored
IPLB*n*_PLB_MTimeout DPLB*n*_PLB_MTimeout	(various)	I	The PLB arbiter asserts this signal during the 17th clock cycle after the assertion of PLB_PAValid if no response is received from any slave. The wrapper then generates the expected data transfer signals back to the processor block while asserting PLBC405ICUERR or PLBC405DCUERR.
IPLB*n*_M_request DPLB*n*_M_request	C405PLBDCUREQUEST	O	Indicates the processor is making an access request.
IPLB*n*_M_RNW	N/A	O	Always driven to 1.
DPLB*n*_M_RNW	C405PLBDCURNW	O	Specifies whether the data-access request is a read or a write.
IPLB*n*_M_ABus[0:31] DPLB*n*_M_ABus[0:31]	C405PLBICUABUS[0:29] C405PLBDCUABUS[0:31]	O	Specifies the memory address of the access request.
DPLB*n*_M_TAttribute[0]	C405PLBDCUWRITETHRU	O	Indicates the value of the write-through storage attribute for the target address.
DPLB*n*_M_TAttribute[3]	C405PLBDCUGUARDED	O	Indicates the value of the guarded storage attribute for the target address.
IPLB*n*_M_TAttribute[4] DPLB*n*_M_TAttribute[4]	C405PLBICUU0ATTR C405PLBDCUU0ATTR	O	Indicates the value of the user-defined storage attribute "U0" for the target address.
IPLB*n*_M_TAttribute[9] DPLB*n*_M_TAttribute[9]	C405PLBDCUCACHEABLE	O	Indicates the value of the cacheability storage attribute for the target address.
IPLB*n*_M_TAttribute[0-3, 5-8, 10-15] DPLB*n*_M_TAttribute[1, 2, 5-8, 10-15]	N/A	O	Always 0.

Table 2: **Embedded Processor Block Wrapper Signal Exceptions *(Cont'd)***

Wrapper Port Name	Processor Block Connection	I/O Type	Function
IPLB*n*_M_BE[0:15]	N/A	O	Always driven to all-ones. (The ICU only issues cache-line transfers.)
DPLB*n*_M_BE[0:15]	C405PLBDCUBE[0:7]	O	Specifies which bytes are transferred during single-unit transfers. 8-bit value from processor is mirrored when connected to a 128-bit PLB bus.
IPLB*n*_M_priority[0:1] DPLB*n*_M_priority[0:1]	C405PLBICUPRIORITY[0:1] C405PLBDCUPRIORITY[0:1]	O	Indicates the priority of the access request.
IPLB*n*_M_wrDBus[0:*width*-1]	N/A	O	Always driven to all-zeros.
DPLB*n*_M_wrDBus[0:*width*-1]	C405PLBDCUWRDBUS[0:63]	O	The DCU write-data bus. 64-bit value from processor is mirrored when connected to a 128-bit PLB bus.
IPLB*n*_M_abort DPLB*n*_M_abort	N/A	O	Always driven to 0. Aborts by the processor's ICU (C405PLBICUABORT) are suppressed by the wrapper by masking the data transfer signals returned by the slave. The processor DCU issues aborts (C405PLBDCUABORT) only during reset, and are ignored in the wrapper.
IPLB*n*_M_busLock DPLB*n*_M_busLock	N/A	O	Always 0.
IPLB*n*_M_lockErr DPLB*n*_M_lockErr	N/A	O	Always 1.
IPLB*n*_M_MSize[0:1] DPLB*n*_M_MSize[0:1]	N/A	O	Always "01" (64-bit master).
IPLB*n*_M_rdBurst DPLB*n*_M_rdBurst	N/A	O	Always 0.
IPLB*n*_M_type[0:2] DPLB*n*_M_type[0:2]	N/A	O	Always "000".
IPLB*n*_M_UABus[0:31] DPLB*n*_M_UABus[0:31]	N/A	O	Always `0x00000000`.
IPLB*n*_M_wrBurst DPLB*n*_M_wrBurst	N/A	O	Always 0.
DBGC405DEBUGHALT	DBGC405DEBUGHALT	I	OR'ed with inverse of DBGC405DEBUGHALTNEG to produce processor block DBGC405DEBUGHALT input ('0' if unconnected)
DBGC405DEBUGHALTNEG	DBGC405DEBUGHALT	I	Inverse OR'ed with DBGC405DEBUGHALT to produce processor block DBGC405DEBUGHALT input ('1' if unconnected)

Table 3: **APU Control Parameter Fields**

Field Name	C_APU_CONTROL Bit (TIEAPUCONTROL)	APU Controller Configuration Register Bit
LdStDecDis	0	5
UDIDecDis	1	6
ForceUDINonB	2	7
FPUDecDis	3	8
FPUCArithDis	4	9
FPUConvIDis	5	10
FPUEstimIDis	6	11
ForceFPUNonB	7	15
StoreWBOK	8	16
LdStPrivOp	9	17
ForceAlign	10	20
LETrap	11	21
BETrap	12	22
BESteer	13	23
APUDiv	14	24
FCMEn	15	31

Table 4: **UDI Control Parameter Fields**

Field Name	C_APU_UDI Bit (TIEAPUUDI)	APU Controller Configuration Register Bit
PriOpCodeSel	0	0
ExtOpCode	1:11	1:11
PrivOp	12	12
RaEn	13	13
RbEn	14	14
GPRWrite	15	15
XerOVEn	16	16
XerCAEn	17	17
CRFieldEn	18:20	18:20
Type	21:22	26:27
UDIEn	23	31

Parameter - Port Dependencies

If either the IPLB1 or DPLB1 bus interface is connected, then either the clock frequencies of all connected PLB buses must be known in the design (CLK_FREQ_HZ values in the MHS) or the user must specify parameter C_FASTEST_PLB_CLOCK.

Timing Constraints

When any of the PLB buses operates at a lower clock frequency than the fastest PLB clock, the wrapper automatically generates multi-cycle timing constraints for the I/O signals that cross clock domains. This prevents paths traversing the slower bus from being artificially over-constrained by the PLBCLK period of the processor.

Multi-cycle timespecs are generated only if the frequencies of all PLB buses are defined in the design ("CLK_FREQ_HZ" values in MHS) and C_GENERATE_PLB_TIMESPECS = 1. The names of all generated timespecs begin with "TS_*instancename*_".

Regardless of whether multi-cycle timespecs are generated, the wrapper always generates TPTHRU timing groups consisting of all signals that cross domains, which can be referenced in user timing constraints, if needed. The generated TPTHRU groups are named "*instancename*_MCYCLE_*busname*", for each connected PLB bus interface, where *busname* is IPLB0, IPLB1, DPLB0 or DPLB1.

If the highest frequency DPLB bus runs at the same frequency as the CPU clock of the processor (CPMC405CLOCK), the wrapper generates a timespec to constrain the PLBC405DCURDDACK input of the processor block so that all transitions arrive during the first half of each cycle, provided C_GENERATE_PLB_TIMESPECS = 1. This is done to correct an internal synchronization problem in the Virtex-4 processor block.

Reference Documents

1. UG018 *PowerPC 405 Processor Block Reference Guide*

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
9/26/08	1.0	Initial Xilinx release.
4/24/09	1.1	Replaced references to supported device families and tool name(s) with hyperlink to PDF file.
2/10/10	1.2	Incorporated CR545435; added supported device families and tool name(s) in LogiCORE Table.

Notice of Disclaimer