XILINX® LogiCORE™

OPB to PLBV46 Bridge (v1.01a)

DS726 April 24, 2009 Product Specification

Introduction

The On-Chip Peripheral Bus (OPB) to Processor Local Bus (PLB v4.6) Bridge module translates OPB transactions into PLBV46 transactions. It functions as a slave on the OPB side and a master on the PLBV46 side. The OPB to PLBV46 Bridge is necessary in systems where an OPB master device, such as a DMA engine or an OPB based coprocessor, requires access to high performance system resources on the PLBv46 bus.

The Xilinx® OPB to PLBV46 Bridge design allows customers to tailor the bridge to suit their application by setting certain parameters to enable and disable features. The parameterizable features of the design are discussed in Design Parameters.

Features

The Xilinx OPB to PLBV46 Bridge is a soft IP core with the following features:

- Bridge Functions
  - Uses 16-word deep posted write buffer to decouple OPB and PLBV46 transactions.
  - Uses 16-word deep read prefetch buffer to eliminate bridge related system lockup issues.
- PLBV46 Master interface
  - 32-bit native device width
  - Communicates with 32-bit, 64-bit, and 128-bit PLBV46 slaves
  - Non-burst transfers of 1 to 4 bytes
  - Uses fixed length, burst signaling of up to 16, 32-bit words.

LogiCORE™ IP Facts		
Core Specifics		
See EDK Supported Device Families.		
Version of core	opb_plbv46_bridge	v1_01_a
Resources Used		
	Min	Max
I/O	390	390
LUTs	467	898
FFs	630	720
Block RAMs	0	0
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	See Tools for requirements.	
Verification		
Simulation		
Synthesis		
Support		
Provided by Xilinx, Inc.		

Features (contd)

- OPB Slave interface
    - 32-bit OPB Slave interface that responds to byte enable transfers only. (Does not support dynamic bus sizing or non-byte enable transactions.)
    - Decodes up to four separate address ranges
    - PLBV46 and OPB clock periods may have a 1:1 or 1:2 synchronous relationship.
    - Utilizes read prefetch and OPB retries to eliminate deadlock and increase PLB bus performance.
- Utilizes post write buffer to improve performance.

Functional Description

Overview

Figure 1 provides a high-level overview of the OPB to PLBV46 Bridge.

OPB transactions are received and decoded in the OPB slave and data sent or received to or from the appropriate buffer. The bridge controls the operation of the slave and implements the read prefetching and posted writes. As a result the bridge effectively de-couples the OPB and PLB buses to improve the PLB performance and eliminate the typical read lockup potential.

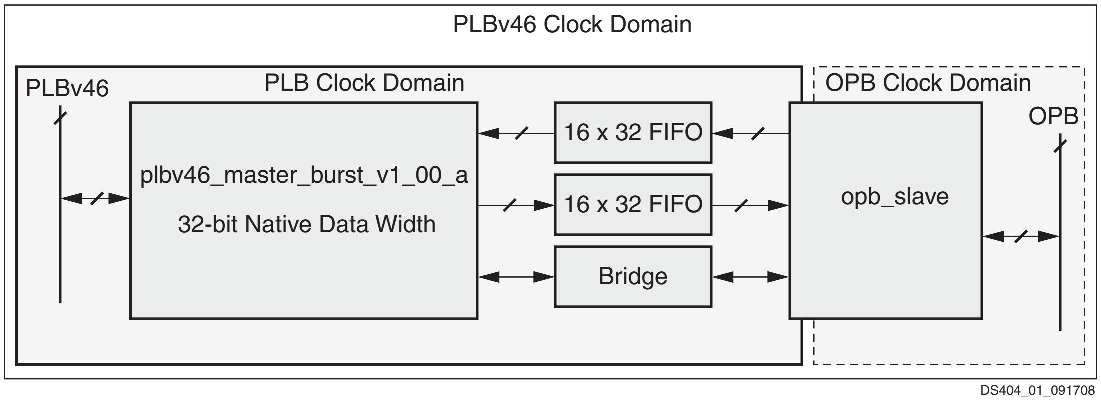

Figure 1: **OPB PLBV46 Bridge in 1:2 Clock Ratio Configuration**

Clocking

The bridge provides for a PLBV46:OPB clock *period* ratio of 1:1 or 1:2. The bridge implementation requires that the clocks be generated by one DCM. This insures that the rising edges of the PLBv46 and OPB clocks are aligned and that the necessary and proper period constraint is applied to signals that cross time domain boundaries.

Deadlock Prevention

Deadlock can occur when masters request their bridges to attempt read transactions at the same time and the OPB slave on the OPB to PLB bridge utilizes timeout suppression. It starts when the PLB to OPB bridge addrAcks a read transaction before it knows the OPB will be busy, thus tying up the PLB read bus. (There is no timeout on the PLBV46 once the address phase completes.) Simultaneously, an OPB master connects to the OPB to PLB bridge slave for a read and uses timeout suppression to block further access to the OPB until its read completes. However, its read will never complete because the PLB read bus is locked by the other bridge. The read attempts result in total locking of both buses.

The solution used by the OPB to PLB bridge involves decoupling the buses through the use of a posted write buffer and a read prefetch buffer. With this solution the system does not require the use of OPB timeout suppression and no inter-bridge communication is needed to eliminate the potential for deadlock.

PLBv46 Master Burst IO Signals

Table 1: **PLBv46 Master Burst IO Signal Description**

Signal Name	Interface	Signal Type	Init Status	Description
PLB Clock and Reset				
MPLB_Clk	PLB Bus	I		PLB main bus clock. See table note 1.
MPLB_Rst	PLB Bus	I		PLB main bus reset. See table note 1.
Other System Signal				
MD_error	PLB Bus	O	’0’	Master Detected Error Status Output
PLB Request and Qualifier Signals				
M_request	PLB Bus	O	’0’	See Table note 2.
M_priority	PLB Bus	O	’0’	
M_buslock	PLB Bus	O	’0’	
M_RNW	PLB Bus	O	’0’	
M_BE(0:[**C_MPLB_DWIDTH**/8]-1)	PLB Bus	O	zeros	
M_Msize(0:1)	PLB Bus	O	"00"	
M_size(0:3)	PLB Bus	O	"0000"	
M_type(0:2)	PLB Bus	O	"000"	
M_ABus(0:**31**)	PLB Bus	O	zeros	
M_wrBurst	PLB Bus	O	’0’	
M_rdBurst	PLB Bus	O	’0’	
M_wrDBus(0:**C_MPLB_DWIDTH**-1)	PLB Bus	O	zeros	
PLB Reply Signals				

Table 1: **PLBv46 Master Burst IO Signal Description** ***(Contd)***

Signal Name	Interface	Signal Type	Init Status	Description
PLB_MSSize(0:1)	PLB Bus	I		Unused See table note 2
PLB_MaddrAck	PLB Bus	I		See table note 1.
PLB_Mrearbitrate	PLB Bus	I		
PLB_MTimeout	PLB Bus	I		
PLB_MRdErr	PLB Bus	I		
PLB_MWrErr	PLB Bus	I		
PLB_MRdDBus(0:**C_MPLB_DWIDTH**-1)	PLB Bus	I		
PLB_MRdDAck	PLB Bus	I		
PLB_MWrDAck	PLB Bus	I		
PLB_RdBTerm	PLB Bus	I		
PLB_MWrBTerm	PLB Bus	I		
PLB Signal Ports Included in the Design, but Unused Internally				
M_TAttribute(0 to 15)	PLB Bus	O	’0’	Unused. See table note 2.
M_lockerr	PLB Bus	O	’0’	
M_abort	PLB Bus	O	’0’	
M_UABus(0:**31**))	PLB Bus	O	zeros	
PLB_MBusy	PLB Bus	I		
PLB_MIRQ	PLB Bus	I		
PLB_RdWdAddr(0:3)	PLB Bus	I		
OPB Signals				
OPB_select	OPB	I		Slave select
OPB_RNW	OPB	I		Read=1, Write=0
OPB_BE(0:3)	OPB	I		Byte Enables
OPB_beXfer	OPB	I		Unused
OPB_hwXfer	OPB	I		Unused
OPB_fwXfer	OPB	I		Unused
OPB_dwXfer	OPB	I		Unused
OPB_seqAddr	OPB	I		Sequential Address (burst) indication
OPB_ABus(0:31)	OPB	I		Transaction address
OPB_DBus(0:31)	OPB	I		Write data arriving from the bus.

Table 1: **PLBv46 Master Burst IO Signal Description** ***(Contd)***

Signal Name	Interface	Signal Type	Init Status	Description
Sl_xferAck	OPB	O		Slave transfer acknowledge
Sl_beAck	OPB	O		Byte enable transfer acknowledge
Sl_errAck	OPB	O		Error Acknowledge
Sl_retry	OPB	O		Asserted high to force master off the bus.
Sl_DBus(0:31)	OPB	O		Read data sent back to bus
Sl_toutSup	OPB	O		Time out suppress
OPB System signals				
OPB_Clk	OPB	I		OPB Clock domain
OPB_Rst	OPB	I		OPB slave reset.

Notes:

1. This function and timing of this signal is defined in the IBM 128-Bit Processor Local Bus Architecture Specification Version 4.6.
2. Output ports that are not used are driven to constant logic levels that are consistent with the inactive state for the subject signal. Input ports that are required but not used are internally ignored by the design.
3. For Fixed Length Burst requests, the starting address for the request as specified by the IP2Bus_Mst_Addr(0:31) input must be aligned on an address boundary matching the C_MPLB_NATIVE_DWIDTH value.

OPB Slave Interface

Single Transaction Bridging

The OPB slave must complete a transaction before it will accept a *new* read or write transaction. It responds with the assertion of `Sl_retry` to each master request until the previous transaction is *successfully bridged*. A *new* read is a transaction to an address that is different than the one currently being prefetched. Every write is considered to be a *new* write irrespective of address. *Successfully bridged* means the plbv46_master_burst had an opportunity at delivering the transaction to the PLB slave and returned a status of success or failure. `Sl_toutSup` is not used to suppress the OPB bus timeout while the bridge attempts to perform a read transaction.

For write transactions, a successfully bridged transaction is complete when the posted write buffer has been emptied by the plbv46_master_burst.

For read transactions, a successfully bridged transaction is complete after the read prefetch is satisfied (either with data or an error indication), a new request with an address satisfying the original prefetch address matches, and the prefetch data is used partially or fully to satisfy the request.

If the prefetch buffer address does not receive a match in a specified time period, a prefetch match timeout error occurs, which results in a flush of the prefetch buffer (and any error status) and a return to accepting transactions. [1]

[1] The user must set the C_PREFETCH_TIMEOUT parameter to a value that balances between stalling access to the bridge and thrashing the read prefetch buffer.

The bridge does not support byte or halfword bursting on the OPB bus. Bursts must start on a word aligned address (address bits 30 to 31 = '00), end on a word aligned address, must and contain only full word data.

The bridge drives valid byte enables onto the PLBV46 only when it detects a single write or single read. It does this by examining OPB_seqAddr. When OPB_SeqAddr=0, it assumes a single transaction with byte enables. Xilinx OPB masters that use the Xilinx IPIF are known to follow this assumption. Otherwise a PLBv46 burst is used and the OPB_be() signal is ignored.

Address Decode Cycle

OPB transactions begin with an address decode cycle. A design parameter allows the user to specify the number of address ranges the bridge will respond to. Each range has two parameters, C_RNGn_BASEADDR and C_RNGn_HIGHADDR, that specify the 32-bit lower and upper boundary for that range. These parameters define that portion of the total system address space to which the OPB slave will respond.

Write Transactions

When an OPB Master requests write access to the bridge, the slave immediately accepts from 1 to 16 full words of data, then buffers the data. The bridge will not accept write data that is addressed beyond the end of an address range, even if the ranges are back-to-back. For a single word write, indicated by the deassertion by the master of OPB_seqAddr, the slave captures the byte enable pattern, OPB_ABus(0:29), and Sl_xferAcks the word. If OPB_seqAddr=1, then the slave counts the number of words written to the buffer to provide the plbv46_master_burst with the fixed length for a write burst on the PLB bus. Byte enables are ignored for PLB burst operations, therefore the OPB master should ensure that all byte enables are asserted high when OPB_seqAddr is asserted high.

After signaling a write request to the plbv46_master_burst (of either a single or burst), the OPB slave blocks (by issuing retries) until it receives confirmation of the PLB operation complete status.

The nature of posted writes prohibits the return of an PLB transaction failure to the OPB master that originated the request. Any error status is therefore lost.

Read Transactions

On receipt of the OPB master read transaction indication, the OPB slave captures the address and OPB seqAddr qualifier, then asserts Sl_retry to force the master off the bus. It then blocks any further requests using Sl_retry. Simultaneously it makes a fixed size read request to the plbv46_master_burst of 1 or 16 words [1]. The OPB only has an indeterminate burst operation, therefore the bridge must *over read* to fulfill the future transaction retry that will claim the data.

When the original master [2] (that initiated the prefetch) returns to bus, it will gain access to the slave by presenting a matching address to the original prefetch request. The slave delivers data to the master until it deasserts the OPB_select signal (ending the transaction normally) or the prefetch buffer empties. The buffer might empty early (and contain < 16 words) if a 16-word prefetch would have accessed data beyond the end of the address range. The result is that all data up to the end of the address range would be transfer-acknowledged to the OPB Master, but nothing beyond that.

If the plbv46_master_burst signals an error at the completion of the read prefetch attempt, the slave will capture this status and *clear the prefetch buffer*. When the original master returns to claim the prefetch data, it will receive an OPB_errAck assertion. Per OPB protocol, the OPB_errAck asserts concurrently with the OPB_xferAck. Upon the selection of the slave, the master will see a continuous stream of OPB_xferAcks until deselection.

Multiple read prefetches may be required for long read bursts. OPB masters must be careful when performing read bursts at address locations which have read side effects because of the prefetching feature (coherency or destructive read problems). Single beat reads should be used when accessing any special memory locations, such as peripherals that destroy the contents of a register when it is read.

The prefetch timeout counter (of width determined by C_PREFETCH_TIMEOUT) starts counting down as soon as the plbv46_master_burst has returned valid data (or an error) to the bridge. The OPB master has until the timer expires to retrieve all of the data. If the timer expires in the middle of a transaction, it will clear the prefetch buffer without returning an error.

PLBV46 Interface

The plbv46_master_burst_v1_00_a pcore services bridge requests for access to the PLB.

The bridge does not utilize the plbv46_master_burst bus lock feature.

Reset

The user must ensure that both sides of the bridge are reset simultaneously with overlapping reset signals. The bridge is not designed to recover from independently applied resets.

[1] When OPB_seqAddr=0 the bridge knows the explicit size of the read request is 1. In all other cases, the read length is unknown and the bridge resorts to reading in16, 32-bit word chunks.

[2] It is important to recognize that the original master may *not* be the one that gets the repeated transaction. If two masters want to read from the same address, then the first may kick off the read prefetch, but the second may actually receive the data upon retry. The bridge has no way to qualify the address with the master that initiated the request. This condition is extremely unlikely. However, no harm should result because further read attempts by the first master would simply result in a brand new prefetch.

Design Parameters

Table 2: **Bridge Design Parameters**

Feature/Description	Parameter Name	Allowable Values	Default Values	VHDL Type
Decoder Address Range Definition				
Number of Address Ranges	C_ NUM_ADDR_RNG	1-4	1	integer
Address range definition base address	C_RNGn_BASEADDR (0 <= n <= 3)	0x00000000 to 0xFFFFFFFF	X"FFFFFFFF"	std_logic_ vector
Address range definition high address	C_RNGn_HIGHADDR (0<=n<=3)	0x00000000 to 0xFFFFFFFF	X"00000000"	std_logic_ vector
Bridge Configuration				
Establishes the ratio of PLB to OPB bus clock periods. The clocks must be synchronous with minimal phase difference.	C_BUS_CLOCK_ PERIOD_RATIO	1=1:1, 2=1:2	1	integer
Specifies the width of the timeout counter that determines the amount of time (in PLBV46 clocks) the bridge waits for a master to retrieve all the read prefetch data before the prefetch buffer is flushed and new transactions are accepted again.	C_PREFETCH _TIMEOUT	5-32	10	integer
PLB I/O Specification				
Specifies the Number of Used Address bits out of the available 64 bits of PLBV46 addressing	C_MPLB_AWIDTH	32	32	integer
Width of the PLB Data Bus to which the Master is attached	C_MPLB_DWIDTH	32, 64, 128	32	integer
Specifies the internal native data width of the Master	C_MPLB_NATIVE _DWIDTH	32	32	integer
FPGA Family Type				
Xilinx FPGA Family	C_FAMILY	spartan3, virtex4, virtex5	"virtex4"	string

Allowable Parameter Combinations

The current implementation of the PLBV46 Master Burst has the following restrictions which apply to parameter value settings:

- The assigned value for C_MPLB_AWIDTH is currently restricted to 32.
- The assigned value for C_MPLB_NATIVE_DWIDTH is currently restricted to 32.

Parameter-Port Dependencies

N/A

Device Utilization and Performance Benchmarks

Core Performance

Because the opb_plbv46_bridge is a module that will be used with other design pieces in the FPGA, the resource utilization and timing numbers reported in this section are estimates only. When the opb_plbv46_bridge is combined with other pieces of the FPGA design, the utilization of FPGA resources and timing of the design will vary from the results reported here.

For Spartan®-3E FPGA systems the performance of the PLBv46 interface in 1:2 clock ratio mode should meet or exceed 90 MHz. Similarly, for Virtex®-5 FPGA systems the performance should meet or exceed 120 MHz. In some system configurations (in either 1:1 or 1:2 clock ratio mode) the OPB bus could be the limiting factor thus preventing the PLBv46 interface from reaching full speed. Use of the core in 1:1 clock ratio mode is offered only as an option and no clock frequency performance numbers are provided for it.

The plbv46_opb_bridge resource utilization benchmarks for an xc5vlx50-1-ff676 FPGA for a variety of generic parameter combinations applied on top of a base parameter set are shown in Table 3.

Table 3: **FPGA Resource Utilization Benchmarks**

Parameter Values (For Example)							Device Resources		
C_NUM_ADDR_RNG	RNG0 size	RNG1 size	RNG2 size	RNG3 size	C_BUS_CLOCK_PERIOD_RATIO	C_PREFETCH_TIMEOUT	Slice Registers	Slice LUTs	Occupied Slices
1	2^{32}	X	X	X	1	10	414	571	346
2	0x20000000	0x20000000	X	X	1	10	408	567	314
3	0x20000000	0x20000000	0x20000000	X	1	10	408	559	298
4	0x20000000	0x20000000	0x20000000	0x20000000	1	10	408	564	367
4	0x200	0x200	0x200	0x200	1	10	368	496	320
4	0x200	0x200	0x200	0x200	2	20	380	517	305

Notes: Generic parameters used:

1. C_MPLB_AWIDTH=32
2. C_MPLB_DWIDTH=32
3. C_MPLB_NATIVE_DWIDTH=32
4. C_FAMILY="virtex5"

System Performance

To measure the system performance (F_{MAX}) of this core, this core was added as the Device Under Test (DUT) to a Virtex-4 FPGA system as shown in Figure 2, to a Virtex-5 FPGA system as shown in Figure 3, and to a Spartan-3A FPGA system as shown in Figure 4. The DUT in this core is the PLBv46 OPB bridge and the OPB PLBv46 bridge with the OPB buses connected.

Because the OPB to PLBv46 Bridge core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When this core is combined with other designs in the system, the utilization of FPGA resources and timing of the core design will vary from the results reported here.

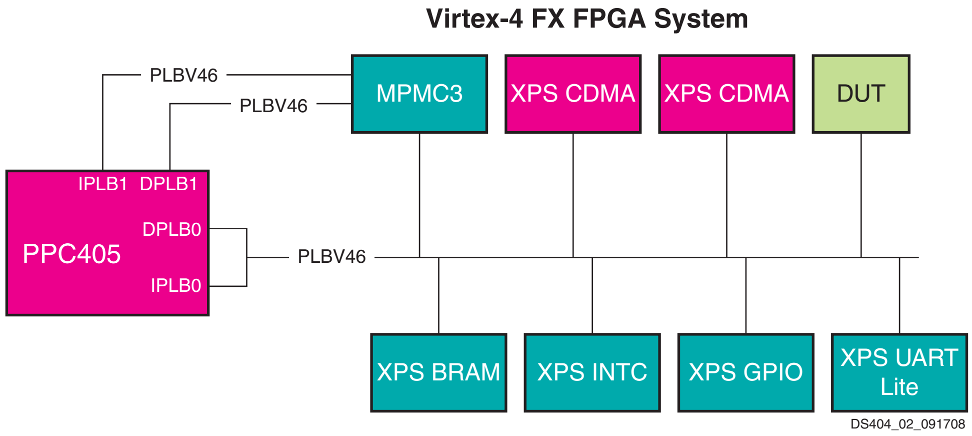

Figure 2: **Virtex-4 FX FPGA System**

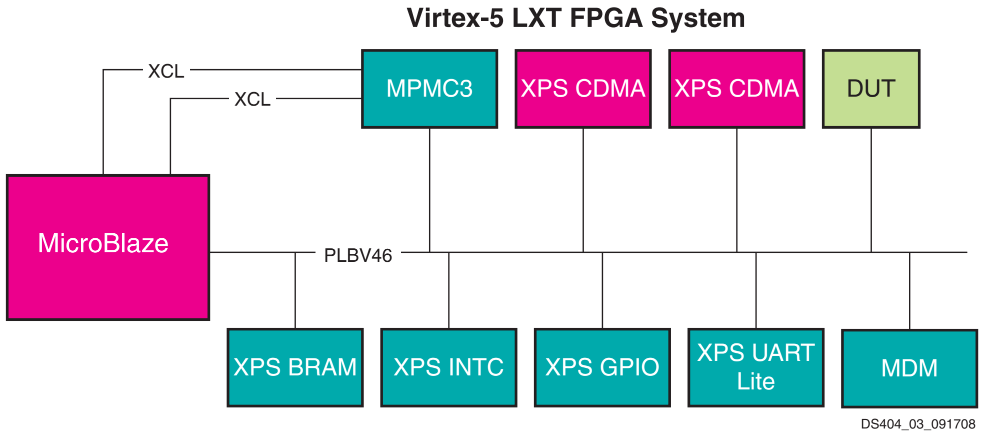

Figure 3: **Virtex-5 LX FPGA System**

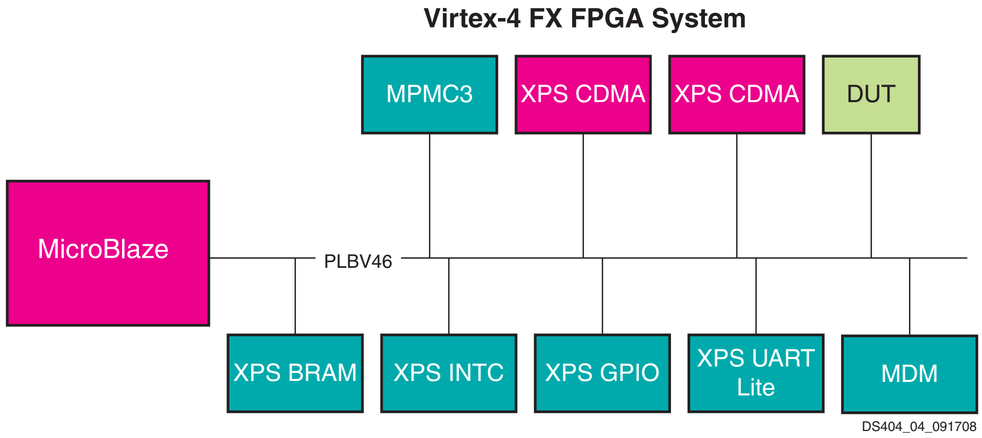

Figure 4: **Spartan-3A FPGA System**

The target FPGA was then filled with logic to drive the LUT and block RAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target F_{MAX} numbers are shown in Table 4.

Table 4: **OPB to PLBv46 Bridge Core System Performance**

Target FPGA	Target F_{MAX} (MHz)
S3A700 -4	90
V4FX60 -10	100
V5LXT50 -1	120

The target F_{MAX} is influenced by the exact system and is provided solely for guidance. It is not a guaranteed value across all systems.

Reference Documents

The following documents contain reference information important to understanding the OPB to PLBV46 Bridge design:

1. *IBM CoreConnect 128-Bit Processor Local Bus: Architecture Specification*
2. *IBM CoreConnect 64-Bit On-Chip Peripheral Bus: Architecture Specifications*
3. *Xilinx PLBv46 Interconnect and Interfaces Simplifications and Feature Subset Specification*

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
6/11/07	1.0	Initial Xilinx release.
10/3/07	1.1	Added FMax Margin System Performance section.
12/13/07	1.2	Added Virtex-II Pro FPGA support.
9/17/08	1.3	Updated for EDK11.1 release; removed Virtex-II support.
04/24/09	1.4	Replaced references to supported device families and tool name(s) with hyperlinks to PDF files; Updated trademark information. Assigned a new Doc ID - DS726 to replace old Doc ID - DS404. Another data sheet already had the DS404 number.

Notice of Disclaimer

.Xilinx is providing this product documentation, hereinafter “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.