PLBV46 to DCR Bridge (v1.01a)

DS578 December 17, 2009 **Product Specification**

Introduction

The PLBV46 to DCR Bridge translates transactions received on its PLB interface into DCR master operations. Its design utilizes an PLB interface module to abstract PLB transactions into a simple SRAM style protocol that is easier to design with.

The Device Control Register (DCR) bus is used primarily for accessing the status and control registers within the various PLB masters and slaves. The main advantage of using the bridge, instead of the CPU to control the DCR bus, is that it provides a memory mapped interface that may be preferable to the use of special move to/move from DCR instructions.

Because the bridge typically runs at a slower clock frequency than the CPU, its timing requirements are also less stringent. The PLBV46 to DCR Bridge implements a simple and flexible method for communicating with DCR devices.

Features

- Connects as a 32-bit slave on PLB V4.6 buses of 32-bit, 64-bit, or 128-bit
- 32-bit DCR master data width with a 10-bit DCR address bus
- Memory-mapped interface from PLB to DCR, no special instructions required
- Increased timing flexibility in typical systems where the PLB clock is slower than the CPU clock
- Allows master devices other than the CPU to access the DCR bus
- Provides a mechanism where CoreConnect™ systems without a CPU can support DCR devices

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex®-4, Virtex-5, Virtex-5FX	
Version of Core	plbv46_dcr _bridge	v1.01a
Resources Used		
	Min	Max
Slices	See Table 4 and Table 5	
LUTs		
FFs		
Block RAMs	N/A	
Special Features	N/A	
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	N/A	
Additional Items	N/A	
Design Tool Requirements		
Xilinx Implementation Tools	ISE® 11.4	
Verification	Mentor Graphics ModelSim 6.4b 0r later	
Simulation	Mentor Graphics ModelSim 6.4b or later	
Synthesis	XST	
Support		
Provided by Xilinx, Inc.		

Functional Description

Figure 1 shows internal diagram of the PLBV46 to DCR Bridge core.

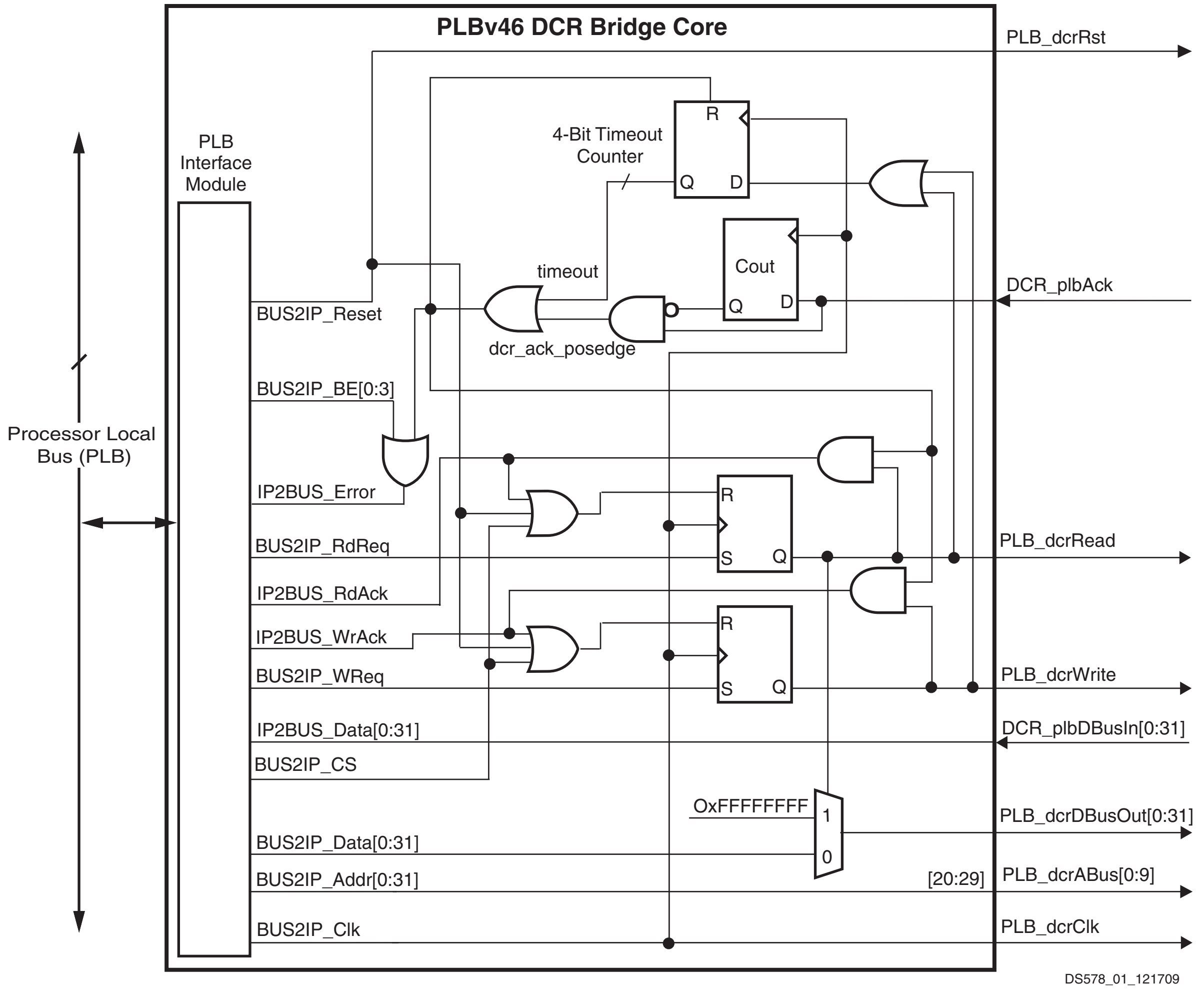

Figure 1: **PLBV46 to DCR Bridge Internal Diagram**

The PLB Interface Module provides necessary address decoding logic and interface between PLB and DCR Bridge core signals.

The read/write requests from the PLB Interface Module are latched by set-reset flip flops, which then drive the PLB_dcrRead/PLB_dcrWrite signals.

Simultaneously, the corresponding DCR address (PLB_dcrABus) and write data bus out (PLB_dcrDBusOut) signals are driven. A time-out counter is also enabled and DCR_plbDBusIn is sampled.

When a DCR slave returns an acknowledge (rising edge of DCR_plbAck detected) or, a DCR timeout occurs after 16 clock cycles, the corresponding IP2Bus_WrAck or IP2Bus_RdAck signal is asserted back to the PLB slave interface module to end the transaction.

For debugging purposes, the data bus out is set to 0xFFFFFFFF during DCR reads so that a read timeout would return 0xFFFFFFFF by default. PLB_dcrClk and PLB_dcrRst are directly connected to

Bus2IP_Clk/Bus2IP_Rst for convenience, even though these signals can be taken directly from the PLB clock and reset lines.

The DCR is organized as a 32-bit bus data width with 10-bit of address to reference a particular 32-bit DCR register. This 1K x 32 bit DCR address space maps into a 4KB address space on PLB. Because DCR does not support byte enables, the byte enables are ignored. Consequently, you must be careful when accessing DCR registers with 1-byte to 3-byte instructions, because the inactive byte lanes carry undefined data. The errors will be reported back to PLB, if the transactions are less than 32-bit. Xilinx recommends that you use only full word transactions.

PLBV46 to DCR Bridge I/O Signals

The PLBV46 to DCR Bridge I/O signals are listed and described in Table 1.

Table 1: **PLBV46 to DCR Bridge I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
System Signals					
P1	SPLB_Clk	PLB	I	-	PLB clock
P2	SPLB_Rst	PLB	I	-	PLB reset, active high
PLB Slave Interface Input Signals					
P3	PLB_ABus[0 : C_SPLB_AWIDTH - 1]	PLB	I	-	PLB address bus
P4	PLB_PAValid	PLB	I	-	PLB primary address valid
P5	PLB_masterID[0 : C_SPLB_MID_WIDTH - 1]	PLB	I	-	PLB current master identifier
P6	PLB_RNW	PLB	I	-	PLB read not write
P7	PLB_BE[0 : (C_SPLB_DWIDTH/8) - 1]	PLB	I	-	PLB byte enables
P8	PLB_size[0 : 3]	PLB	I	-	PLB size of requested transfer
P9	PLB_type[0 : 2]	PLB	I	-	PLB transfer type
P10	PLB_wrDBus[0 : C_SPLB_DWIDTH - 1]	PLB	I	-	PLB write data bus
Unused PLB Slave Interface Input Signals					
P11	PLB_UABus[0 : C_SPLB_AWIDTH - 1]	PLB	I	-	PLB upper address bits
P12	PLB_SAValid	PLB	I	-	PLB secondary address valid
P13	PLB_rdPrim	PLB	I	-	PLB secondary to primary read request indicator
P14	PLB_wrPrim	PLB	I	-	PLB secondary to primary write request indicator
P15	PLB_abort	PLB	I	-	PLB abort bus request
P16	PLB_busLock	PLB	I	-	PLB bus lock
P17	PLB_MSize	PLB	I	-	PLB data bus width indicator

Table 1: **PLBV46 to DCR Bridge I/O Signal Description *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P18	PLB_lockErr	PLB	I	-	PLB lock error
P19	PLB_wrBurst	PLB	I	-	PLB burst write transfer
P20	PLB_rdBurst	PLB	I	-	PLB burst read transfer
P21	PLB_wrPendReq	PLB	I	-	PLB pending bus write request
P22	PLB_rdPendReq	PLB	I	-	PLB pending bus read request
P23	PLB_wrPendPri[0 : 1]	PLB	I	-	PLB pending write request priority
P24	PLB_rdPendPri[0 : 1]	PLB	I	-	PLB pending read request priority
P25	PLB_reqPri[0 : 1]	PLB	I	-	PLB current request priority
P26	PLB_TAttribute	PLB	I	-	PLB transfer attribute
PLB Slave Interface Output Signals					
P27	Sl_addrAck	PLB	O	0	Slave address acknowledge
P28	Sl_SSize[0 : 1]	PLB	O	0	Slave data bus size
P29	Sl_wait	PLB	O	0	Slave wait
P30	Sl_rearbitrate	PLB	O	0	Slave bus rearbitrate
P31	Sl_wrDAck	PLB	O	0	Slave write data acknowledge
P32	Sl_wrComp	PLB	O	0	Slave write transfer complete
P33	Sl_rdDBus[0 : C_SPLB_DWIDTH - 1]	PLB	O	0	Slave read data bus
P34	Sl_rdDAck	PLB	O	0	Slave read data acknowledge
P35	Sl_rdComp	PLB	O	0	Slave read transfer complete
P36	Sl_MBusy[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave busy
P37	Sl_MWrErr[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave write error
P38	Sl_MRdErr[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave read error
Unused PLB Slave Interface Output Signals					
P39	Sl_wrBTerm	PLB	O	0	Slave terminate write burst transfer
P40	Sl_rdWdAddr[0 : 3]	PLB	O	0	Slave read word address
P41	Sl_rdBTerm	PLB	O	0	Slave terminate read burst transfer
P42	Sl_MIRQ[0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Master interrupt request

Table 1: **PLBV46 to DCR Bridge I/O Signal Description *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
DCR Interface Signals					
P43	DCR_plbAck	PLB	I	-	DCR to PLB acknowledge
P44	DCR_plbDBusIn[0 : C_SPLB_NATIVE_DWIDTH - 1]	PLB	I	-	DCR to PLB data bus in
P45	PLB_dcrRead	PLB	O	0	PLB to DCR read
P46	PLB_dcrWrite	PLB	O	0	PLB to DCR write
P47	PLB_dcrABus[0 : 9]	PLB	O	0	PLB to DCR address bus
P48	PLB_dcrDBusOut[0 : C_SPLB_NATIVE_DWIDTH - 1]	PLB	O	0	PLB to DCR data bus out
P49	PLB_dcrClk	PLB	O	0	PLB to DCR clock
P50	PLB_dcrRst	PLB	O	0	PLB to DCR reset

PLBV46 to DCR Bridge Design Parameters

To allow the user to create a PLBV46 to DCR Bridge that is uniquely tailored for the user's system, certain features can be parameterized. This allows the user to have a design that utilizes only the resources required by the system and runs at the best possible performance. The features that can be parameterized in the PLBV46 to DCR Bridge core are as shown in Table 2.

Table 2: **PLBV46 to DCR Bridge Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameter					
G1	Target FPGA family	C_FAMILY	virtex4, virtex5, virtex5fx	virtex4	string
PLB Parameters					
G2	PLBV46 to DCR Bridge Base Address	C_BASEADDR	Valid Address[1]	None	std_logic_vector
G3	PLBV46 to DCR Bridge High Address	C_HIGHADDR	Valid Address[2]	None	std_logic_vector
G4	PLB address width	C_SPLB_AWIDTH	32	32	integer
G5	PLB data width	C_SPLB_DWIDTH	32, 64, 128	32	integer
G6	Selects point-to-point or shared bus topology	C_SPLB_P2P	0 = Shared Bus Topology 1 = Reserved	0	integer
G7	PLB Master ID Bus Width	C_SPLB_MID_WIDTH	$\log_2$(C_SPLB_NUM_MASTERS) with a minimum value of 1	1	integer
G8	Number of PLB Masters	C_SPLB_NUM_MASTERS	1 - 16	1	integer

Table 2: **PLBV46 to DCR Bridge Design Parameters *(Contd)***

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G9	Width of the Slave Data Bus	C_SPLB_NATIVE _DWIDTH	32	32	integer
G10	The burst support from PLB masters	C_SPLB_ SUPPORT_ BURSTS	0[3]	0	integer

Notes:

1. No default values will be specified for C_BASEADDR to ensure that the actual value is set, i.e. if the value is not set, a compiler error will be set. The C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR + 1. It should be a valid word aligned address.
2. C_HIGHADDR - C_BASEADDR must be a power of 2 ≥ to C_BASEADDR + 0xFFF.
3. Burst is not supported. Fixed to default value.

PLBV46 to DCR Bridge Parameter Port Dependencies

The dependencies between the PLBV46 to DCR Bridge core design parameters and I/O signals are described in Table 3. In addition, when certain features are parameterized out of the design, the related logic will no longer be a part of the design. The unused input signals and related output signals are set to a specified value.

Table 3: **PLBV46 to DCR Bridge Parameter-Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G4	C_SPLB_AWIDTH	P3, P11	-	Affects the number of bits in address bus
G5	C_SPLB_DWIDTH	P7, P10, P33	-	Affects the number of bits in data bus
G7	C_SPLB_MID_WIDTH	P5	G8	Affects the number of bits required for the PLB_masterID input bus for slave devices. This value is equal to log$_2$(C_SPLB_NUM_MASTERS)
G8	C_SPLB_NUM_ MASTERS	P36, P37, P38, P42	-	Affects the number of PLB masters
G9	C_SPLB_NATIVE_DWIDTH	P44, P48	-	Affects the data bus width of slaves
I/O Signals				
P3	PLB_ABus[0 : C_SPLB_AWIDTH - 1]	-	G4	Width of PLB_ABus varies with the size of the PLB Address bus
P5	PLB_masterID[0 : C_SPLB_MID_WIDTH - 1]	-	G7	Width of PLB_masterID varies with the size of the PLB master ID width
P7	PLB_BE[0 : (C_SPLB_DWIDTH/8) -1]	-	G5	Width of PLB_BE varies with the size of the PLB Data bus
P10	PLB_wrDBus[0 : C_SPLB_DWIDTH - 1]	-	G5	Width of PLB_wrDBus varies with the size of the PLB Data bus

Table 3: **PLBV46 to DCR Bridge Parameter-Port Dependencies *(Contd)***

Generic or Port	Name	Affects	Depends	Relationship Description
P11	PLB_UABus[0 : C_SPLB_AWIDTH - 1]	-	G4	Width of PLB_UABus varies with the size of the PLB upper Address bus
P33	Sl_rdDBus[0 : C_SPLB_DWIDTH - 1]	-	G5	Width of Sl_rdDBus varies with the size of the PLB Data bus
P36	Sl_MBusy[0 : C_SPLB_NUM_MASTERS - 1]	-	G8	Width of Sl_MBusy varies with the no. of the PLB masters
P37	Sl_MWrErr[0 : C_SPLB_NUM_MASTERS - 1]	-	G8	Width of Sl_MWrErr varies with the no. of the PLB masters
P38	Sl_MRdErr[0 : C_SPLB_NUM_MASTERS - 1]	-	G8	Width of Sl_MRdErr varies with the no. of the PLB masters
P42	Sl_MIRQ[0 : C_SPLB_NUM_MASTERS - 1]	-	G8	Width of Sl_MIRQ varies with the no. of the PLB masters
P44	DCR_plbDBusIn[0 : C_SPLB_NATIVE_DWIDTH - 1]	-	G9	Width of DCR slave input data bus varies with the size of C_SPLB_NATIVE_DWIDTH
P48	PLB_dcrDBusOut[0 : C_SPLB_NATIVE_DWIDTH - 1]	-	G9	Width of DCR slave output data bus varies with the size of C_SPLB_NATIVE_DWIDTH

Timing Diagrams of PLBV46 to DCR Bridge

Figure 2 and Figure 3 show examples of write and read transactions of PLBV46 to DCR Bridge.

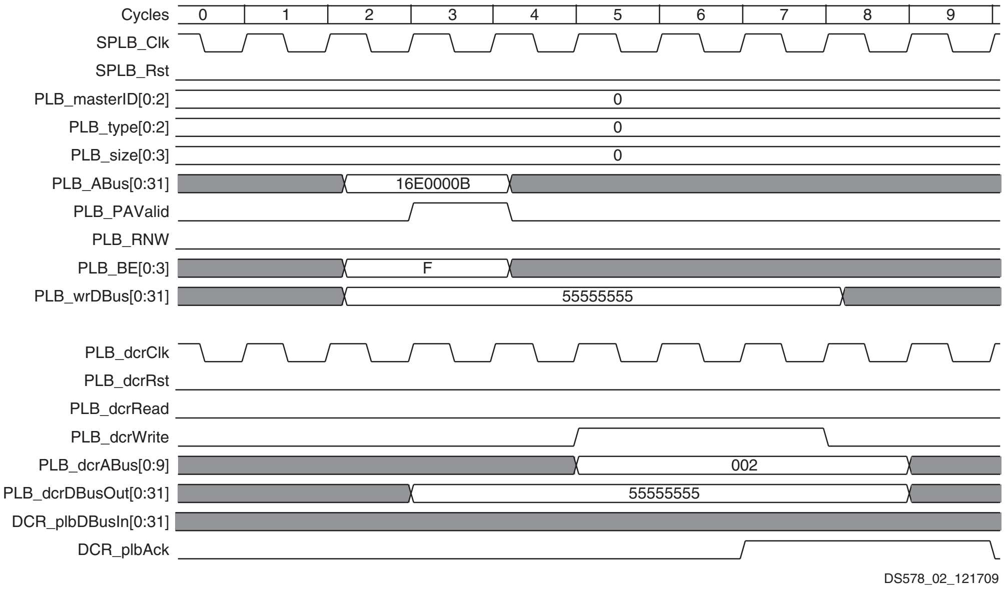

Figure 2: **PLBV46 to DCR Bridge Write Cycle Timing Diagram**

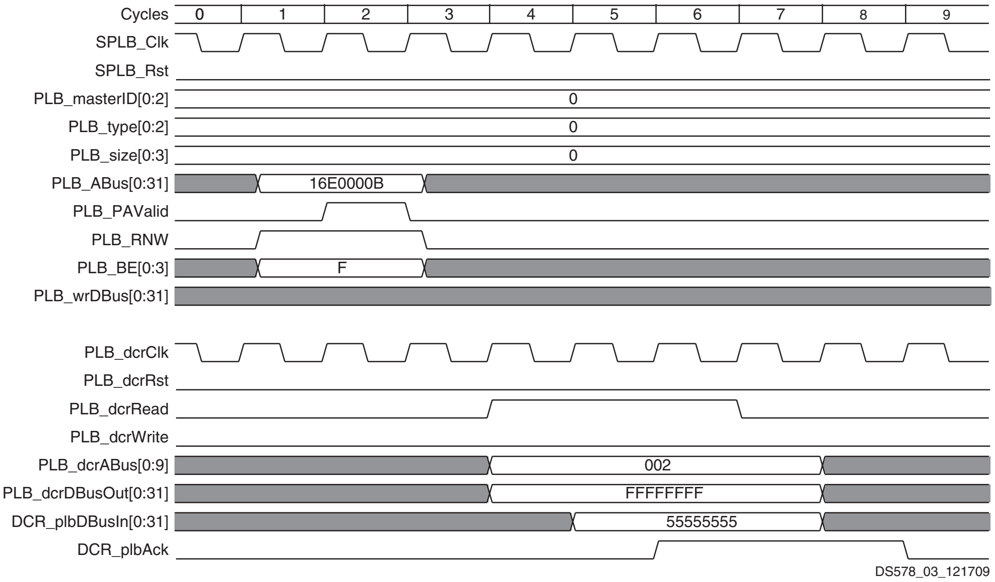

Figure 3: **PLBV46 to DCR Bridge Read Cycle Timing Diagram**

Design Implementation

Target Technology

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

Device Utilization and Performance Benchmarks

Since the PLBV46 to DCR Bridge core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only, and will vary from the results reported here.

The PLBV46 to DCR Bridge resource utilization for various parameter combinations measured with Virtex®-4 as the target device is detailed in Table 4.

Table 4: **Performance and Resource Utilization Benchmarks on Virtex-4 (xc4vlx40-ff1148-11)**

Parameter Values		Device Resources			Performance
C_BASEADDR	C_SPLB_DWIDTH	Slices	Slice Flip-Flops	LUTs	F_{Max} (MHz)
0x10000000	32	211	183	140	166

The PLBV46 to DCR Bridge resource utilization for various parameter combinations measured with Virtex-5 as the target device is detailed in Table 5.

Table 5: **Performance and Resource Utilization Benchmarks on Virtex-5 (xc5vlx50-ff1153-2)**

Parameter Values		Device Resources		Performance
C_BASEADDR	C_SPLB_DWIDTH	Slice Flip-Flops	LUTs	F_{Max} (MHz)
0x10000000	32	183	131	227

Specification Exceptions

N/A

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.*

Reference Documents

1. IBM CoreConnect 128-Bit Processor Local Bus: Architecture Specifications version 4.6
2. IBM CoreConnect 32-Bit Device Control Register Bus: Architecture Specifications Version 3.5

Revision History

Date	Version	Revision
2/23/09	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.
12/17/09	1.2	Incorporated CR518057 to correct PLB_dcrABus[0] signal value to 002 (was 16E) in Figures 2 and 3; listed supported device families and tools in LogiCORE Table; updated images and legal matter.

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.