OPB to DCR Bridge (v1.00b)

Introduction

The OPB to DCR Bridge translates transactions received on its OPB slave interface into DCR master operations. Its design utilizes an Intellectual Property Interface (IPIF) module to abstract OPB 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 OPB 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 OPB to DCR Bridge implements a simple and flexible method for communicating with DCR devices.

Features

- 32-bit DCR master with a 10-bit DCR address bus
- Memory-mapped interface from OPB to DCR, no special instructions required
- Increased timing flexibility in typical systems where the OPB 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™ IP Facts		
Core Specifics		
See EDK Supported Device Families.		
Version of Core	opb to dcr_bridge	v1.00b
Resources Used		
	Min	Max
Slices	See Table 4 and Table 5	
LUTs		
FFs		
Block RAMs	N/A	N/A
Provided with Core		
Documentation	Design 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.		

Functional Description

Figure 1 shows a high-level diagram of the design of the IPIF-based OPB to DCR Bridge.

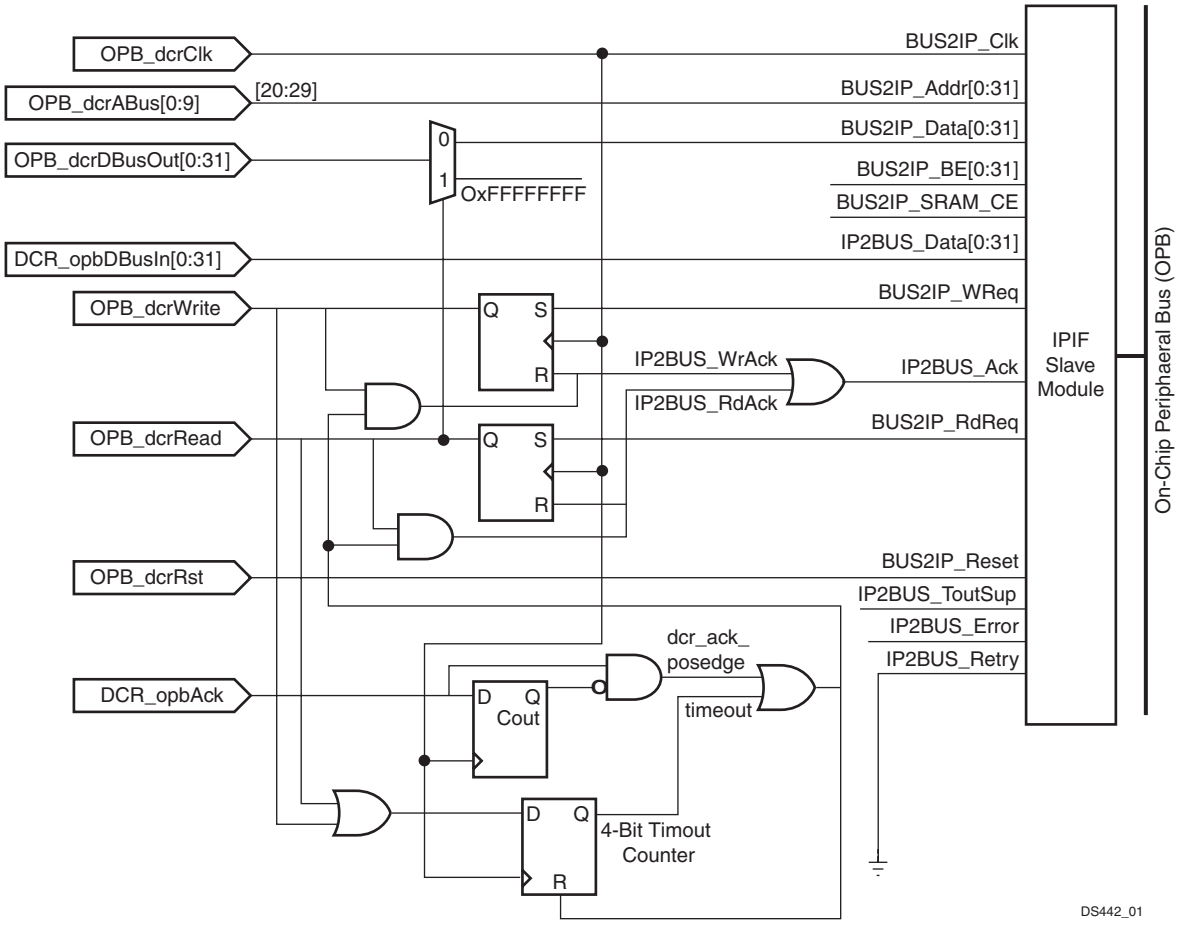

Figure 1: **OPB to DCR Bridge Internal Diagram**

The read/write requests from the IPIF are latched by set-reset flip flops, which then drive the OPB_dcrRead/OPB_dcrWrite signals.

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

When a DCR slave returns an acknowledge (rising edge of DCR_opbAck detected) or, a DCR timeout occurs after 16 clock cycles, the corresponding IP2Bus_Ack (IP2Bus_RdAck OR IP2Bus_WrAck) signal is asserted back to the IPIF to end the transaction.

The DCR protocol specifies that timeouts are not treated as errors, therefore a DCR timeout terminates the transaction without any errors being reported back to OPB.

For debugging purposes, the data bus out is set to 0xFFFFFFFF during DCR reads so that a read timeout would return 0xFFFFFFFF by default. OPB_dcrClk and OPB_dcrRst are directly connected to Bus2IP_Clk/Bus2IP_Rst for convenience, even though these signals can be taken directly from the OPB clock and reset lines.

The DCR is organized as a 32-bit bus with 10 bits 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 OPB. Because DCR does not support byte enables, the byte enables are ignored. Consequently, you must be careful when accessing DCR registers with 1- to 3-byte instructions, because the inactive byte lanes carry undefined data. Xilinx recommends that you use only full word transactions.

OPB to DCR Bridge Design Parameters

To allow the user to obtain a OPB to DCR Bridge that is uniquely tailored to the user's system, certain features can be parameterized in the Xilinx OPB to DCR Bridge design. 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 Table 1 lists design parameters for the OPB to DCR Bridge.

Table 1: **OPB to DCR Bridge Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
	Design Parameters				
G1	OPB to DCR Bridge Base Address	C_BASEADDR	Valid Word Aligned Address	None [1]	std_logic_vector
G2	OPB to DCR Bridge Data Bus Width	C_OPB_DWIDTH	32	32	integer
G3	OPB to DCR Bridge Address Bus Width	C_OPB_AWIDTH	32	32	integer
G4	OPB to DCR Bridge High Address	C_HIGHADDR	C_HIGHADDR -C_BASEADDR must be a power of 2 >= to C_BASEADDR+ 0xFFF	None [1]	std_logic_vector

Notes:

1. No default values will be specified for C_BASEADDR to ensure that the actual value is set. That is, if the value is not set, a compiler error will be set. C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR +1.

OPB to DCR Bridge I/O Signals

The I/O signals for the OPB to DCR Bridge are shown in Figure 1 and described in Table 2.

Table 2: **OPB to DCR Bridge I/O Signals**

Port	Signal Name	Interface	I/O	Initial Diagram State	Description
	OPB Slave Signals				
P1	OPB_ABus(0:C_OPB_ AWIDTH-1)	OPB	I		OPB Address Bus
P2	OPB_BE(0:(C_OPB_ AWIDTH/8)-1)	OPB	I		OPB Byte Enable

Table 2: **OPB to DCR Bridge I/O Signals *(Contd)***

Port	Signal Name	Interface	I/O	Initial Diagram State	Description
P3	OPB_DBus(0:C_OPB_ DWIDTH-1)	OPB	I		OPB Data Bus
P4	OPB_RNW	OPB	I		Read Not Write
P5	OPB_Select	OPB	I		OPB select
P6	OPB_seqAddr	OPB	I		OPB sequential address (unused)
P7	OPB_Timeout	OPB	I		OPB Time Out
P8	OPB_Retry	OPB	I		OPB Retry
P9	OPB_ErrAck	OPB	I		OPB Error Acknowledge
P10	OPB_XferAck	OPB	I		OPB Transfer Acknowledge
P11	Sln_DBus(0:C_OPB_ DWIDTH-1)	OPB	O	0	Output Data Bus
P12	Sln_ErrAck	OPB	O	0	Slave Error Acknowledge
P13	Sln_Retry	OPB	O	0	Slave Bus Cycle Retry (always inactive)
P14	Sln_ToutSup	OPB	O	0	Slave Time Out Suppress
P15	Sln_XferAck	OPB	O	0	Slave Transfer Acknowledge
	DCR Master Signals				
P16	DCR_opbAck	OPB	I	0	DCR to OPB acknowledge
P17	DCR_opbDBusIn[0:31]	OPB	I		DCR to OPB data bus in
P18	OPB_dcrABus[0:9]	OPB	O	0	OPB to DCR address bus
P19	OPB_dcrClk	OPB	O		OPB to DCR clock
P20	OPB_dcrDBusOut[0:31]	OPB	O	0	OPB to DCR data bus out
P21	OPB_dcrRead	OPB	O	0	OPB to DCR read
P22	OPB_dcrRst	OPB	O	1	OPB to DCR reset
P23	OPB_dcrWrite	OPB	O	0	OPB to DCR write
	System				
P24	OPB_Clk	System	I		System clock
P25	OPB_Rst	System	I	1	System Reset (active high)

Parameter Port Dependencies

The width of many of the OPB to DCR Bridge signals depends on the parameter. In addition, when certain features are parameterized away, the related input signals are unconnected. The dependencies between the OPB to DCR Bridge design parameters and I/O signals are described in Table 3.

Table 3: **Parameter-Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G1	C_BASEADDR	None	G3	Bus width affects maximum allowable address.
G2	C_OPB_DWIDTH	P3, P4	None	Affects number of bits in bus.
G3	C_OPB_AWIDTH	P1, P2	None	Affects number of bits in bus.
G4	C_HIGHADDR	None	G1,G3	
	I/O Signals			
P1	OPB_ABus(0:C_OPB_AWIDTH-1)	None	G3	Width varies with the size of the OPB Address bus.
P2	OPB_BE(0:(C_OPB_AWIDTH/8)-1)	None	G3	Width varies with the size of the OPB Address bus.
P3	OPB_DBus(0:C_OPB_DWIDTH-1)	None	G2	Width varies with the size of the OPB Data bus.
P4	Sln_DBus(0:C_OPB_DWIDTH-1)	None	G2	Width varies with the size of the OPB Data bus.

Timing Diagram of OPB to DCR Bridge

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

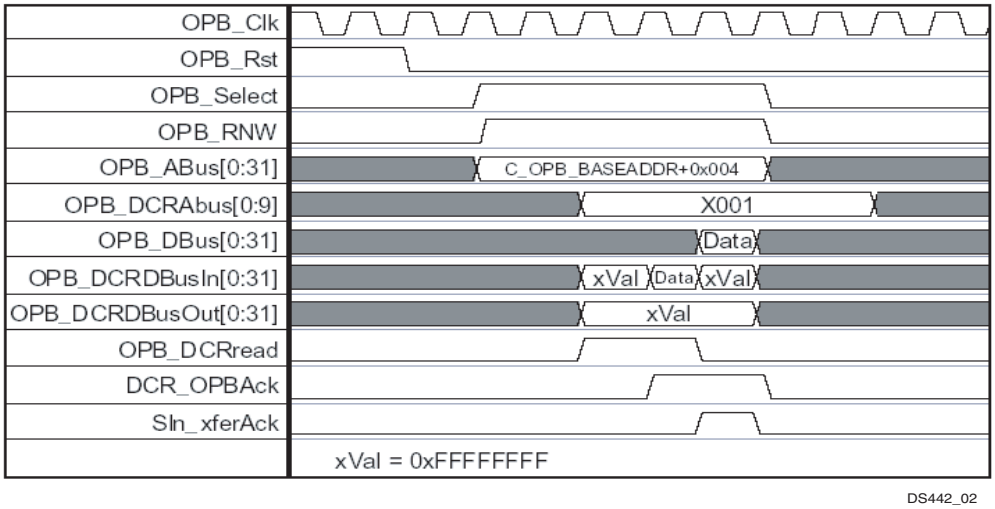

Figure 2: **OPB to DCR Bridge Read Cycle Timing Diagram**

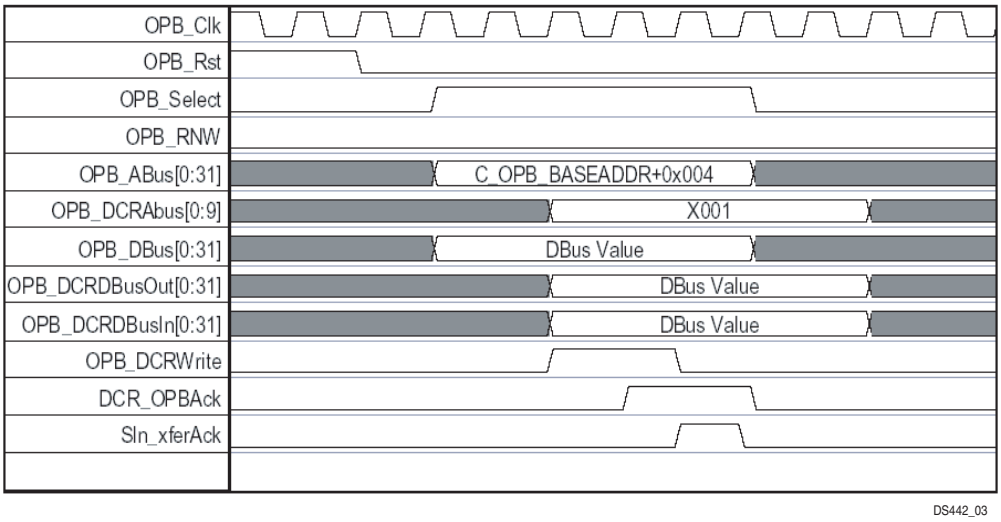

Figure 3: **OPB to DCR Bridge Write Cycle Timing Diagram**

Design Implementation

Target Technology

The target technology is an FPGA listed in EDK Supported Device Families.

Device Utilization and Performance Benchmarks

Table 4: **Performance and Resource Utilization Benchmarks on Virtex®-4 (Device: xc4vlx200ff1513-10)**

Parameter Values		Device Resources			f_{MAX} (MHz)
C_OPB_BASEADDR	C_OPB_DWIDTH	Slices	Slice Flip-Flops	LUTs	f_{MAX}
0x10000000	32	73	19	117	126

Table 5: **Performance and Resource Utilization Benchmarks on Virtex-5 (Device: xc5vlx220ff1760-2)**

Parameter Values		Device Resources			f_{MAX} (MHz)
C_OPB_BASEADDR	C_OPB_DWIDTH	Slices	Slice Flip-Flops	LUTs	f_{MAX}
0x10000000	32	0	17	107	199

Specification Exceptions

No known exceptions.

Related Documents

- IBM CoreConnect 64-Bit On-Chip Peripheral Bus: Architecture Specifications
- IBM CoreConnect 32-Bit Device Control Register Bus: Architecture Specifications

Notice of Disclaimer

Revision History

Date	Version	Revision
06/25/01	1.0	Initial Xilinx release
7/30/02	1.1	Add utilization table
7/31/02	1.2	Add parameters and I/O signals section
01/07/03	1.3	Update copyright
07/07/03	1.4	Update to new template
09/10/03	1.4.1	Correct Base Address definition
09/18/03	1.4.2	Correct graphics to GSC standard and update trademarks
01/26/04	1.4.3	Updates to TM and Copyright
03/24/04	1.4.4	Fixed version number (from 1.01a to 1.00a) updated eps image.
06/24/04	1.5	Removed references to device families that are not V2p, per CR 190407
8/11/04	1.6	Updated for Gmm; updated trademarks and supported device family listing.
9/8/04	1.6.1	Incorporated CR 193089: In Fig 1, signal name DCR_opbDBusIn(0:31) was OPB_dcrDBusIn(0:31).
09/22/04	1.6.2	Fixed version number from 1.00a to 1.00.b. Updated document to reflect the changes in design as per new OPB_IPIF(v3_01_a).Added IP2BUS_ACK signal in OPB to DCR Internal Diagram.
4/14/05	1.7	Updated for EDK 7.1.1 SP1; updated trademarks and supported device listing; made content edits; reformatted tables.
10/04/05	1.8	Converted to new DS template; updated figures to graphic standards.
05/16/06	1.9	Updated to add support for Virtex-5 devices.
04/24/09	2.0	Replaced references to supported device families and tool name(s) with hyperlinks to PDF files; Updated trademark information.