LogiCORE IP XPS HWICAP (v5.00a)

DS586 July 23, 2010 **Product Specification**

Introduction

This product specification describes the functionality of the HWICAP core for the Processor Local Bus (PLB). The XPS HWICAP (Hardware ICAP) IP enables an embedded microprocessor, such as the MicroBlaze™ or PowerPC® to read and write the FPGA configuration memory through the Internal Configuration Access Port (ICAP) at run time, which enables a user to write software programs for an embedded processor that modifies the circuit structure and functionality during the circuit's operation.

Features

The XPS HWICAP includes support for resource reading and modification of the CLB LUTs and Flip-Flops.

- PLB v4.6 based PLB interface
- Partial bitstream loading is possible
- Enables Read/Write of CLB LUTs
- Enables Read/Write of CLB Flip-Flop properties
- ICAP interface operates at a maximum clock rate of 100MHzs
- Support for MicroBlaze and PowerPC embedded processors
- Support for Virtex-4, Virtex-5, Virtex-6 and Spartan-6 FPGA families

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	Spartan®-6/XA, Virtex®-4, Virtex-5, QVirtex-4, QrVirtex-4, Virtex-6
Supported User Interfaces	PLB Interface
Resources	
See Table 15, Table 16, Table 17, and Table 18.	
Provided with Core	
Documentation	Product Specification
Design Files	VHDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	UCF (user constraints file)
Simulation Model	Not Providedl
Tested Design Tools	
Design Entry Tools	Xilinx Platform Studio (XPS)
Simulation	Mentor Graphic ModelSim v6.5c and above
Synthesis Tools	Xilinx Synthesis Tool (XST)12
Support	
Provided by Xilinx, Inc.	

Notes:

1. For a complete listing of supported devices, see the release notes for this core.

Functional Description

The XPS HWICAP controller provides the interface necessary to transfer bitstreams to and from the ICAP. The CPU bursts the required bitstream data directly from main memory. Incoming data is stored within a Write FIFO, from where it can be fed to the ICAP. All the bitstreams must be stored in main memory before they can be used to reconfigure the FPGA. Any required bitstreams are preloaded into main memory when the system boots. The XPS HWICAP also provides for read back of configuration resource states. In this case, the frames are read back into the Read FIFO one at a time and the CPU will then be able to read the frame data directly from the Read FIFO.

Sample applications

- A DSP system, like software defined radio, where the filters and algorithms are modified at run time to receive and transmit at variable frequencies, or adapt to variable protocols.
- A debug system where trigger conditions are implemented as comparator circuits and modified at run time to enable variable trigger conditions. The system can also have counters to measure the amount of data sampled. The final counts of the counters can be modified to vary the amounts of data sampled.
- A crossbar switch where the fundamental switches are implemented using multiplexers in the routing fabric. The crossbar connections are reconfigured at run time by reconfiguring the routing multiplexers

The XPS HWICAP top-level block diagram is shown in Figure 1.

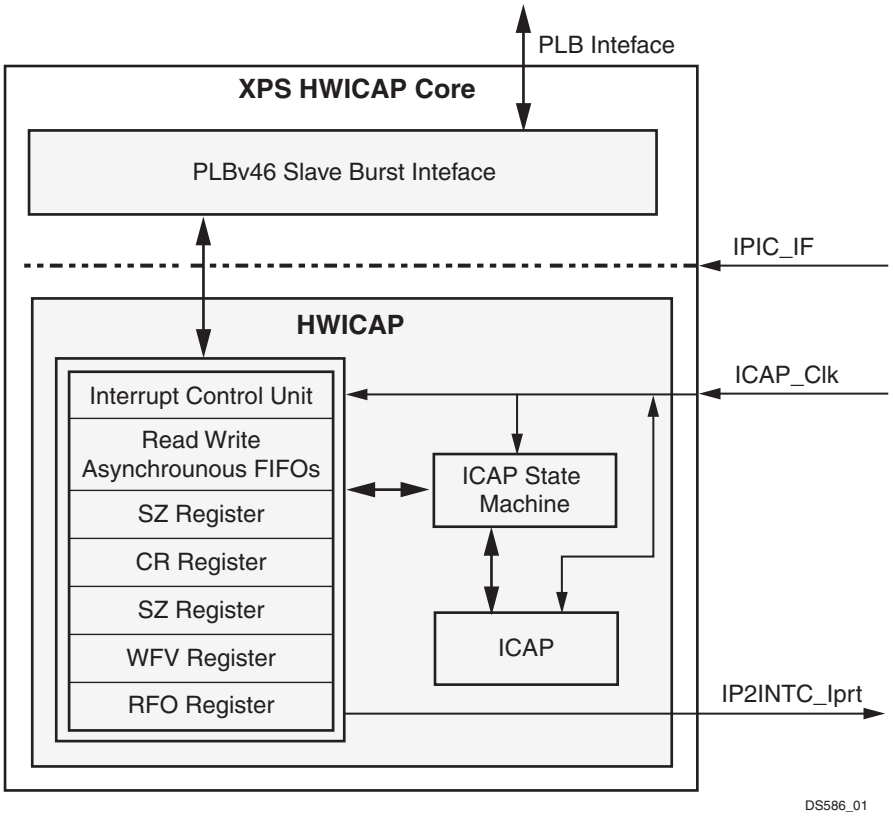

Figure 1: **Top Level Block Diagram for the XPS HWICAP Core**

PLBV46 Slave Burst Module

PLBV46 Salve Burst Module provides the bidirectional interface between HWICAP core and the PLB. The base element of the PLB Interface Module is slave attachment with burst support, which provides the basic functionality of PLB slave operation.

IPIC_IF Module

IPIC_IF module incorporates logic to acknowledge the write and read transactions initiated by the plbv46 slave burst module to write/read the HWICAP module registers and FIFOs.

HWICAP Module

The HWICAP module provides the interface to the Internal Configuration Access Port (ICAP). It has a write FIFO, which will store the configuration locally. The Processor writes the configuration in to the write FIFO. Simultaneously the data stored in the write FIFO transferred to the ICAP. Processor can read the configuration from the ICAP, which will be stored in side the read FIFO. FIFOs are required as the rate of data flow from the processor interface is different from ICAP interface. FIFO depth can be parameterizable using the generics C_WRITE_FIFO_DEPTH, C_READ_FIFO_DEPTH.

Note: In case of Virtex-4, Virtex-5 and Virtex-6: If the SPLB_Clk is greater than 100 MHz, the ICAP_Clk should be connected to 100 MHz. If the SPLB_Clk is less than or equal to 100 MHz, the ICAP_Clk should be connected to the frequency equivalent to SPLB_clk frequency.

Note: In case of Spartan-6 the maximum frequency of operation of ICAP is 20 MHz. The ICAP_Clk must be connected to a frequency less than or equal to 20 MHz.

I/O Signals

The I/O signals are listed and described in Table 1.

Table 1: I/O Signals

Port	Signal Name	Interface	I/O	Initial State	Description
ICAP Interface Signals					
P1	ICAP_Clk (1)	ICAP	I	-	ICAP clock
PLB Bus Request and Qualifier Signals					
P2	SPLB_Clk	PLB	I	-	PLB main bus clock
P3	SPLB_Rst	PLB	I	-	PLB main bus reset
P4	PLB_ABus(0 : C_SPLB_AWIDTH-1)	PLB	I	-	PLB address bus
P5	PLB_PAValid	PLB	I	-	PLB primary address valid indicator
P6	PLB_masterID(0 : C_SPLB_MID_WIDTH - 1)	PLB	I	-	PLB current master identifier
P7	PLB_RNW	PLB	I	-	PLB read not write
P8	PLB_BE[0 : (C_SPLB_DWIDTH/8 - 1)	PLB	I	-	PLB byte enables
P9	PLB_wrBurst	PLB	I	-	PLB write burst
P10	PLB_rdBurst	PLB	I	-	PLB read burst
P11	PLB_size(0 : 3)	PLB	I	-	PLB transfer size

Table 1: I/O Signals *(Cont'd)*

Port	Signal Name	Interface	I/O	Initial State	Description
P12	PLB_type(0 : 2)	PLB	I	-	PLB transfer type
P13	PLB_wrDBus(0 : C_SPLB_DWIDTH - 1)	PLB	I	-	PLB write data bus
P14	PLB_MSize(0:1)	PLB	I	-	PLB data bus width indicator
P15	Sl_addrAck	PLB	O	0	Slave address acknowledge
P16	Sl_SSize(0:1)	PLB	O	0	Slave data bus size
P17	Sl_wait	PLB	O	0	Slave wait indicator
P18	Sl_rearbitrate	PLB	O	0	Slave rearbitrate bus indicator
P19	Sl_wrDack	PLB	O	0	Slave write data acknowledge
P20	Sl_wrComp	PLB	O	0	Slave write transfer complete indicator
P21	Sl_wrBTerm	PLB	O	0	Slave terminate read burst transfer
P22	Sl_rdBus(0:C_SPLB_DWIDTH - 1)	PLB	O	0	Slave read data bus
P23	Sl_rdDAck	PLB	O	0	Slave read data acknowledge
P24	Sl_rdComp	PLB	O	0	Slave read transfer complete indicator
P25	Sl_rdBTerm	PLB	O	0	Slave terminate read burst transfer
P26	Sl_rdWdAddr(0:3)	PLB	O	0	Slave read word address
P27	Sl_MBusy(0:C_SPLB_NUM_ MASTERS-1)	PLB	O	0	Slave busy indicator
P28	Sl_MWrErr(0:C_SPLB_NUM_ MASTERS-1)	PLB	O	0	Slave write error indicator
P29	Sl_MRdErr(0:C_SPLB_NUM_ MASTERS-1)	PLB	O	0	Slave read error indicator
Unused PLB signals					
P30	PLB_UABus(0:C_SPLB_AWIDTH-1)	PLB	I	-	PLB upper address bits
P31	PLB_SAValid	PLB	I	-	PLB secondary address valid
P32	PLB_rdPrim	PLB	I	-	PLB secondary to primary read request indicator
P33	PLB_wrPrim	PLB	I	-	PLB secondary to primary write request indicator
P34	PLB_abort	PLB	I	-	PLB abort bus request
P35	PLB_busLock	PLB	I	-	PLB bus lock
P36	PLB_TAttribute(0:15)	PLB	I	-	PLB transfer attribute
P37	PLB_lockerr	PLB	I	-	PLB lock error
P38	PLB_wrPendReq	PLB	I	-	PLB pending bus write request
P39	PLB_rdPendReq	PLB	I	-	PLB pending bus read request
P40	PLB_rdPendPri(0:1)	PLB	I	-	PLB pending read request priority
P41	PLB_wrPendPri(0,1)	PLB	I	-	PLB pending write request priority
P42	PLB_reqPri(0:1)	PLB	I	-	PLB current request priority

Table 1: I/O Signals *(Cont'd)*

Port	Signal Name	Interface	I/O	Initial State	Description
P43	SI_MIRQ(0:C_SPLB_NUM_MASTERS-1)	PLB	O	0	Master interrupt request
System Signals					
p44	IP2INTC_Irpt	PLB	O	0	XPS HWICAP Interrupt

Notes:

1. ICAP_Clk signal is added to the design to support the limitation on the ICAP i.e.., ICAP works upto a maximum frequency of 100 MHz. in case of Virtex-4, Virtex-5 and Virtex-6. In case of Spartan-6 the limitiation on ICAP operating frequency is 20 MHz.

Design Parameters

To allow the user to create a core that is uniquely tailored for the user's system, certain features are parameterizable in the 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 features that are parameterizable in the core are as shown in Table 2.

Table 2: **Design Parameters**

Generic	Parameter Description	Parameter Name	Allowable Values	Default Value	VHDL Type
PLB Parameters					
G1	XPS HWICAP Base Address	C_BASEADDR	Valid Word Aligned Address(1)	None(2)	std_logic_vector
G2	XPS HWICAP High Address	C_HIGHADDR	C_HIGHADDR -C_BASEADDR must be a power of 2 >= to C_BASEADDR+1FF(1)	None(2)	std_logic_vector
G3	PLB Data Bus Width	C_SPLB_DWIDTH	32, 64, 128	32	integer
G4	PLB Address Bus Width	C_SPLB_AWIDTH	32	32	integer
G5	PLB Point-to-Point or shared bus topology	C_SPLB_P2P	0 : Shared bus topology 1 : Reserved	0	integer
G6	PLB master ID bus width	C_SPLB_MID_WIDTH	$\log_2$(C_SPLB_NUM_MASTERS) with a minimum value of 1	3	integer
G7	Number of PLB masters	C_SPLB_NUM_MASTERS	1 - 16	8	integer
G8	Width of slave data bus	C_SPLB_NATIVE_DWIDTH	32	32	integer
G9	Width of the smallest master that will be interacting with this slave	C_SPLB_SMALLEST_MASTER	32, 64, 128	32	integer
G10	Write FIFO depth	C_WRITE_FIFO_DEPTH(3)(6)	64, 128, 256, 512,1024	64	integer
G11	Read FIFO depth	C_READ_FIFO_DEPTH(3)(6)	128, 256	128	integer
G12	Select FIFO type(4)	C_BRAM_SRL_FIFO_TYPE	0,1	1	integer
System Parameters					

Table 2: **Design Parameters *(Cont'd)***

Generic	Parameter Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G12	XILINX FPGA Family	C_FAMILY	spartan3, aspartan3, spartan3an, spartan3a, spartan3e, spartan3adsp, aspartan3e, aspartan3a, aspartan3adsp, virtex4, virtex5,virtex5fx, qvirtex4, qrvirtex4, spartan6, aspartan6, virtex6.virtex6cx	spartan3	string
G13	Simulation(5)	C_SIMULATION	1: FIFO Model 2: UNISIM Model	2	integer

Notes:

1. Address range specified by C_BASEADDR and C_HIGHADDR. C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR +1 and must be a power of 2. For example, C_BASEADDR = 0x10000000, C_HIGHADDR = 0x100001FF.
2. No default value will be specified to insure that the actual value is set, i.e. if the value is not set, a compiler error will be generated.
3. This parameter must be set to 256 if the C_FAMILY = virtex6/spartan6, as the frame size in virtex6/spartan6 family is 162 bytes
4. The parameter C_BRAM_SRL_FIFO_TYPE selects the FIFO type to be BRAM (1) or Distributed RAM (0)
5. The parameter C_SIMULATION must be set to 2 for simulations using the ICAP unisim model.
6. The Actual depth of the Write/Read FIFO is one less than the parameter defining the depth of the FIFOs (i.e.., either C_WRITE_FIFO_DEPTH or C_READ_FIFO_DEPTH)

Parameter - Port Dependencies

The dependencies between the XPS HWICAP 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: **Parameter-Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G3	C_SPLB_DWIDTH	P8, P13, P22	-	Affects the size of the PLB data bus
G4	C_SPLB_AWIDTH	P4, P30	-	Affects the size of the PLB address bus
G6	C_SPLB_MID_WIDTH	P6	G9	Affects the width of the PLB master ID
G7	C_SPLB_NUM_MASTERS	P27, P28, P29, P43	-	Identify the specific master on the PLB
I/O Signals				
P4	PLB_ABus[0:C_SPLB_ AWIDTH - 1]	-	G4	Width varies with the size of the PLB address bus
P6	PLB_masterID[0: C_SPLB_MID_WIDTH - 1]	-	G6	Width varies with the size of the number of masters on the PLB
P8	PLB_BE[0:[C_SPLB_ DWIDTH/8] - 1]	-	G3	Width varies with the size of the PLB data bus
P13	PLB_wrDBus[0:C_SPLB_ DWIDTH - 1]	-	G3	Width varies with the size of the PLB data bus
P22	Sl_rdBus[0:C_SPLB_ DWIDTH - 1]	-	G3	Width varies with the size of the PLB data bus

Table 3: **Parameter-Port Dependencies *(Cont'd)***

Generic or Port	Name	Affects	Depends	Relationship Description
P27	SI_MBusy[0:C_SPLB_NUM_MASTERS - 1]	-	G7	Width varies with the number of masters on the PLB
P28	SI_MWrErr[0:C_SPLB_NUM_MASTERS - 1]	-	G7	Width varies with the number of masters on the PLB
P29	SI_MRdErr[0:C_SPLB_NUM_MASTERS - 1]	-	G7	Width varies with the number of masters on the PLB
P43	SI_MIRQ(0:C_SPLB_NUM_MASTERS-1)	-	G7	Width varies with the number of masters on the PLB

XPS HWICAP Register Definition

The internal registers of the XPS HWICAP are offset from the base address C_BASEADDR. The XPS HWICAP internal register set is described in Table 4.

Table 4: **XPS HWICAP Registers**

Register Name	C_BASEADDR + Address	Access
Global Interrupt Enable Register (GIE)	C_BASEADDR + 0x01C	Read/Write
IP Interrupt Enable Register (IPIER)	C_BASEADDR + 0x020	Read/Write
IP Interrupt Enable Register (IPIER)	C_BASEADDR + 0x028	Read/Write
Write FIFO Register (WF)	C_BASEADDR + 0x100	Write
Read FIFO Register (RF)	C_BASEADDR + 0x104	Read
Size Register (SZ)	C_BASEADDR + 0x108	Write
Control Register (CR)	C_BASEADDR + 0x10C	Read/Write
Status Register (SR)	C_BASEADDR + 0x110	Read
Write FIFO Vacancy Register (WFV)	C_BASEADDR + 0x114	Read
Read FIFO Occupancy Register (RFO)	C_BASEADDR + 0x118	Read

Write FIFO Register (WF)

This is a 32-bit Write FIFO as shown in Figure 2. The bit definitions for the Write FIFO are shown in Table 5. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

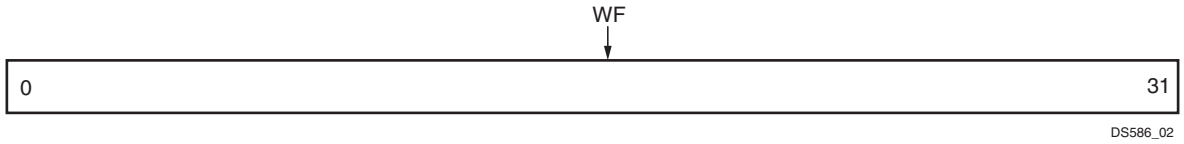

Figure 2: **Write FIFO (WF)**

Table 5: **Write FIFO Bit Definitions**

Bit	Name	Access	Reset Value	Description
0 - 31	WF	Write	0	Data written into the FIFO

Read FIFO Register (RF)

This is a 32-bit Read FIFO as shown in Figure 3. The bit definitions for the Read FIFO are shown in Table 6. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

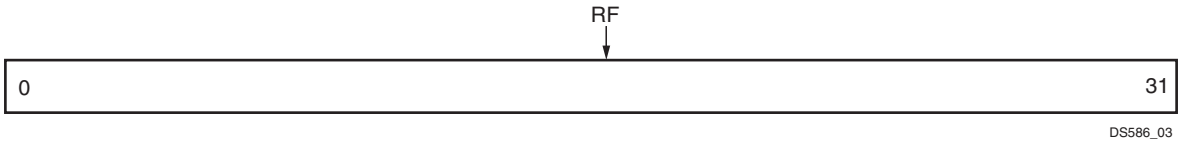

Figure 3: **Read FIFO (RF)**

Table 6: **Read FIFO Bit Definitions**

Bit	Name	Access	Reset Value	Description
0 - 31	RF	Read	0	Data read from the FIFO

Size Register (SZ)

This is a 12-bit write register as shown in Figure 4. The SZ register determines the number of words to be transferred from the ICAP to the read FIFO. The bit definitions for the register are shown in Table 7. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

Figure 4: **Size Register (SZ)**

Table 7: **Size Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
0 - 19	Reserved	N/A	0	Reserved bits
20 - 31	Size	Write	0	Number of words to be transferred from the ICAP to the FIFO

Control Register (CR)

This is a 4-bit write register as shown in Figure 5. The CR register determines the direction of the data transfer. It controls whether a configuration or read back takes place. Writing to this register initiates the transfer. The bit definitions for the register are shown in Table 8. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

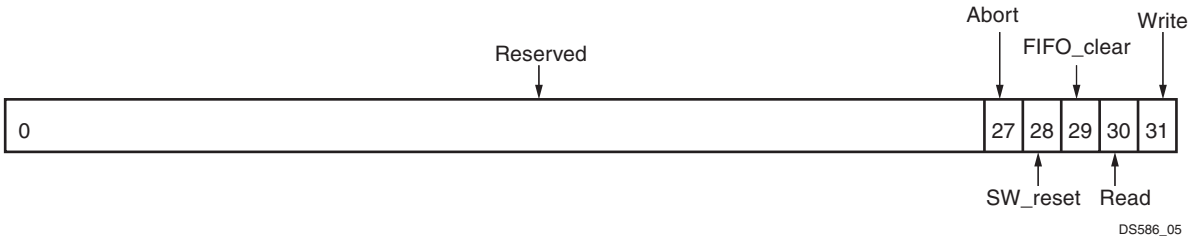

Figure 5: **Control Register (CR)**

Table 8: **Control Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
0 - 26	Reserved	N/A	'0'	Reserved bits
27	Abort	Read/Write	'0'	'1' = Aborts the read or write of the ICAP and clears the FIFOs
28	SW_reset	Read/Write	'0'	'1' = Resets all the registers
29	FIFO_clear	Read/Write	'0'	'1' = Clears the FIFOs
30	Read	Read/Write	'0'	'1' = Initiates readback of bitstream in to the Read FIFO
31	Write	Read/Write	'0'	'1' = Initiates writing of bitstream in to the ICAP

Status Register (SR)

This is a 9-bit read register as shown in Figure 6. The Status Register contains the ICAP status bits. The bit definitions for the register are shown in Table 9. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

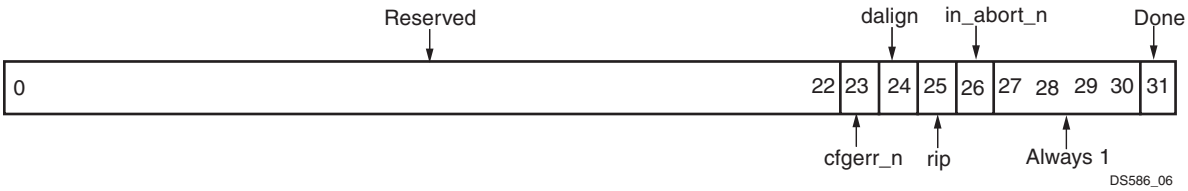

Figure 6: **Status Register (SR)**

Table 9: **Status Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
0 - 22	Reserved	N/A	0	Reserved bits
23	cfgerr_n	Read	1	Configuration error
24	dalign	Read	0	Data alignment, found syncword
25	rip	Read	0	Read back in progress
26	in_abort_n	Read	1	Super8 (Select MAP) abort
27- 30	Always 1	Read	1	Always 1
31	Done	Read	1	XPS HWICAP done with configuration or read

Write FIFO Vacancy Register (WFV)

This is an 11-bit read register as shown in Figure 7. The write FIFO vacancy register indicates vacancy of the write FIFO. The actual depth of the Write FIFO is one less than the C_WRITE_FIFO_DEPTH. The bit definitions for the register are shown in Table 10. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

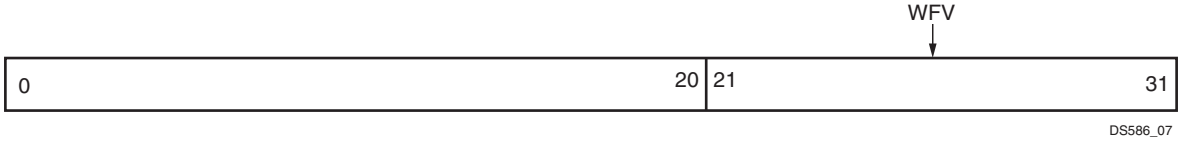

Figure 7: **Write FIFO Vacancy Register (WFV**

Table 10: Write FIFO Vacancy Register Bit Definitions

Bit	Name	Access	Reset Value	Description
0 - 21	Reserved	NA	-	Reserved
22 - 31	WFV	Read	0	Write FIFO Vacancy

Read FIFO Occupancy Register (RFO)

This is an 8-bit read register as shown in Figure 8. The read FIFO occupancy register indicates occupancy of the read FIFO. The actual depth of the Write FIFO is one less than the C_WRITE_FIFO_DEPTH. The bit definitions for the register are shown in Table 11. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

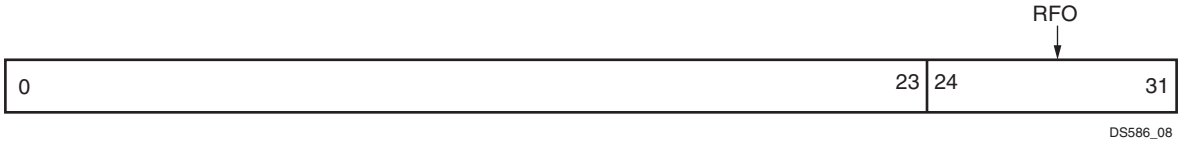

Figure 8: **Read FIFO Occupancy Register (RFO)**

Table 11: **Read FIFO Occupancy Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
0 - 24	Reserved	N/A	-	Reserved
25 - 31	RFO	Read	0	Read FIFO Occupancy

XPS HWICAP Interrupt Descriptions

The interrupt signals generated by the XPS HWICAP are managed by the Interrupt Service Controller (ISC). This unit provides many of the features commonly provided for interrupt handling. Please refer to the *Processor IP Reference Guide* under Part 1 for a complete description of the GIE, IPISR and IPIER. The XPS XPSHWICAP has four unique interrupts that are sent to the CPU.

Global Interrupt Enable Register (GIE)

The Global Interrupt Enable Register (GIE) is used to globally enable the final interrupt output from the Interrupt Controller as shown in Figure 9 and described in Table 12. This bit is a read/write bit and is cleared upon reset.

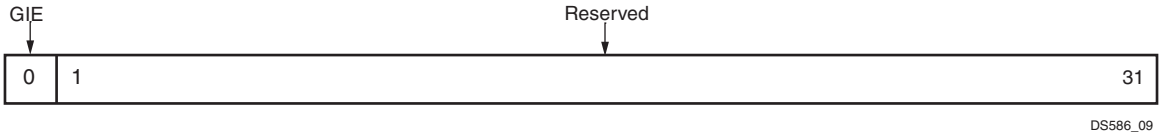

Figure 9: **Global Interrupt Enable Register (GIER)**

Table 12: **Global Interrupt Enable Register Bit Definitions**

Bit(s)	Name	Access	Reset Value	Description
0	GIE	R/W	'0'	'0' = Disabled '1' = Enabled
1 - 31	Reserved	N/A	N/A	Reserved

IP Interrupt Status Register (IPISR)

Four unique interrupt conditions are possible in HWICAP core. The Interrupt Controller has a register that can enable each interrupt independently. Bit assignment in the Interrupt register for a 32-bit data bus is shown in Figure 10 and described in Table 13. The interrupt register is a read/toggle on write register and by writing a '1' to a bit position within the register causes the corresponding bit position in the register to 'toggle'. All register bits are cleared upon reset.

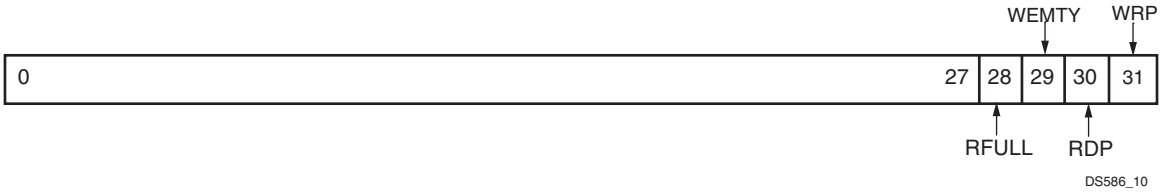

Figure 10: **IP Interrupt Status Register (IPISR)**

Table 13: **IP Interrupt Status Register Bit Definitions**

Bit(s)	Name	Access	Reset Value	Description
0 - 27	Undefined	N/A	N/A	Undefined
28	RFULL	R/TOW<RD Red><SP Superscript>(1)	'0'	Read FIFO full
29	WEMTY	R/TOW<RD Red><SP Superscript>(1)	'0'	Write FIFO empty
30	RDP	R/TOW<RD Red><SP Superscript>(1)	'0'	Interrupt set and remains set if the read FIFO occupancy is greater than half of the read FIFO size
31	WRP	R/TOW<RD Red><SP Superscript>(1)	'0'	Interrupt set and remains set if the write FIFO occupancy is less than half of the write FIFO size

Notes:

1. TOW = Toggle On Write. Writing a '1' to a bit position within the register causes the corresponding bit position in the register to toggle.

IP Interrupt Enable Register (IPIER)

The IPIER has an enable bit for each defined bit of the IPISR as shown in Figure 11 and described in Table 14. All bits are cleared upon reset.

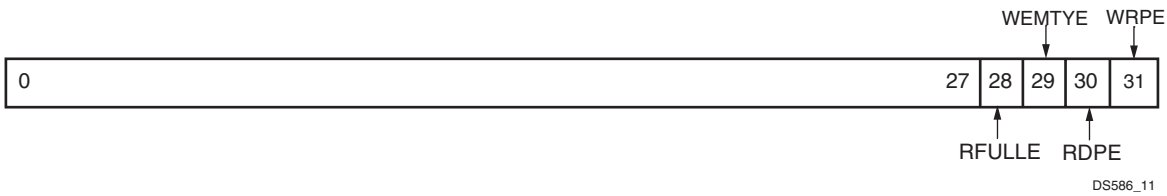

Figure 11: **IP Interrupt Enable Register (IPIER)**

Table 14: **IP Interrupt Enable Register Bit Definitions**

Bit(s)	Name	Access	Reset Value	Description
0 - 27	Undefined	N/A	N/A	Undefined.
28	RFULLE	R/W<RD Red>(1)	'0'	Read FIFO full interrupt enable
29	WEMTYE	R/W<RD Red>(1)	'0'	Write FIFO empty interrupt enable
30	RDPE	R/W<RD Red>(1)	'0'	Read FIFO occupancy greater than half of its size interrupt enable
31	WRPE	R/W<RD Red>(1)	'0'	Write FIFO occupancy less than half of its size interrupt enable

Notes:

1. Writing '1' to this bit will enable the particular interrupt. Writing '0' to this bit will disable the particular interrupt.

Abort

An Abort is an interruption in the configuration or read-back sequence occurring when the state of icap_we changes while icap_ce is asserted. During a configuration Abort, internal status is driven onto the icap_dout[7:4] pins over the next four clock cycles. The other icap_dout pins are always high. After the Abort sequence finishes, the user can re-synchronize the configuration logic and resume configuration. Abort enable the user to know the status of the ICAP during the configuration or reading the configuration.

Software Support

Documentation for the associated software drivers for this hardware module are also available in EDK.

User Application Hints

The use of the XPS HWICAP is outlined in the steps below.

- Read or Configuration (write)
  - Write in to the Control Register (CR) to initiates the read or write of bit-stream. The CR register determines the direction of the data transfer. Writing "0x00000001" in to the Control Register (CR) initiates the configuration. Writing "0x00000002" in to the Control Register (CR) initiates the read.
  - Write the bit-stream in to the Write FIFO Register (WF) to configure. Get the bit-stream from the Read FIFO Register (RF) to read.
  - Done bit in the Status Register (SR) indicates whether the ICAP interface is busy with writing/reading data from/to the ICAP bus. It doesn't not indicate that the read/configuration with ICAP is completed successfully
  - Hardware clears the Control Register (CR) bits after the successful completion of the read or configuration
  - Software should not initiate another read or configuration to ICAP until the read or configuration bit in the Control Register (CR) is cleared

- Abort
  - Write in to the Control Register (CR) to initiates the read or write of bit-stream. The CR register determines the direction of the data transfer. Writing "0x00000001" in to the Control Register (CR) initiates the configuration. Writing "0x00000002" in to the Control Register (CR) initiates the read.
  - Write the bit-stream in to the Write FIFO Register (WF) to configure. Get the bit-stream from the Read FIFO Register (RF) to read.
  - Write '1' in to the 27th bit of the Control Register (CR) to initiates abort
  - Done bit in the Status Register (SR) indicates whether the ICAP interface is busy with writing/reading data from/to the ICAP bus. It doesn't not indicate that the read/configuration with ICAP is completed successfully.
  - Hardware clears the Control Register (CR) bits after the successful completion of the abort-on read or abort-on configuration or normal abort
  - Software should not initiate another read or configuration to ICAP until the read or configuration bit in the Control Register (CR) is cleared

Timing Diagrams

The following timing diagram Figure 12, Figure 13 and Figure 14 shows read and write cycles of XPS HWICAP core for Virtex-4, Virtex5, Virtex-6 and Spartan-6 family of FPGA devices.

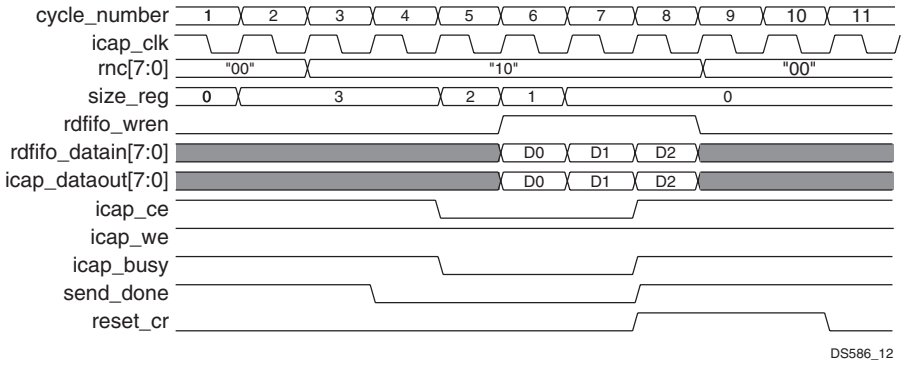

Figure 12: **Read Cycle - Virtex-4, Virtex5 and Virtex-6 Devices**

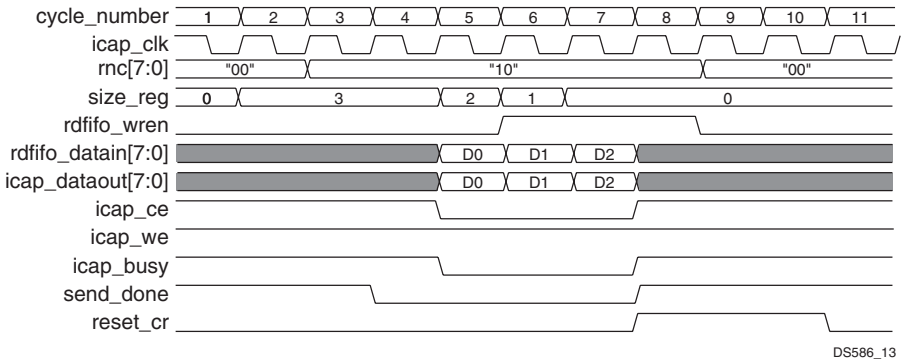

Figure 13: **Read Cycle - Spartan -6 Devices**

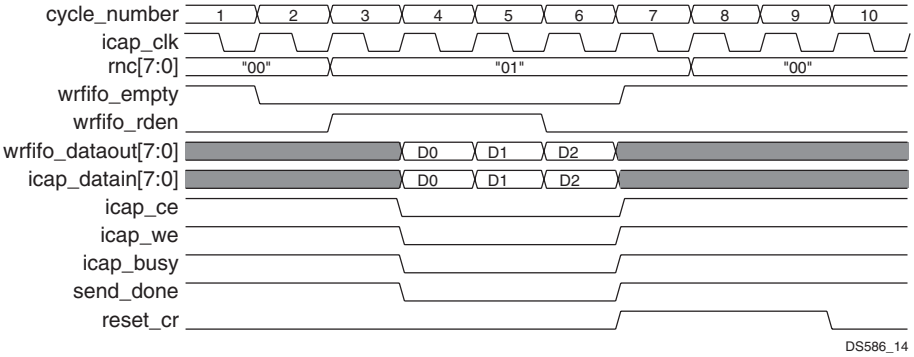

Figure 14: **Write Cycle**

Limitations

A frame is the smallest granularity in which the FPGA allows configuration data to be read and written. A configuration frame is a collection of bits that is 1 bit wide and spans the full column of the FPGA. Configuration frames in the CLB space also contain IOB configuration data at the top and bottom, which configure the IOBs at the top and bottom of the FPGA. A single column of CLBs contains multiple configuration frames.

Although a single CLB LUT or flip-flop can be modified, the underlying mechanism requires that the full column be read into Block RAM. This implies that other logic in the same column can be modified. In most cases, this effect can be ignored. When the frame is written back to the configuration memory the sections of the column that were not modified are written with the same data. Because the FPGA memory cells have glitch less transitions, when rewritten, the unmodified logic will continue to operate unaffected.

Two exceptions to this rule exist: when LUTs are configured in Shift Register Mode or as a RAM. If a LUT is modified or just read back in a column that also has a LUT RAM or LUT shift register, then the LUT or shift register will be interrupted and it will lose its state. To resolve the problem, the LUT shift registers and LUT RAMs should be placed in columns that are not read back or modified. If the LUT RAMs or shift register in a column do not change state during the read back or modification, then they will maintain their state.

Important Notes:

1. The HWICAP core uses the ICAP found inside Virtex-4, Virtex-5, Virtex-6 and Spartan-6 devices. The ICAP port interface is similar to the SelectMAP interface, but is accessible from general interconnects rather than the device pins. The JTAG or "Boundary Scan" configuration mode pin setting (M2:M0 = 101) will disable the ICAP interface. Therefore, when using the HWICAP core, another mode pin setting must be used. If JTAG will be used as the primary configuration method, another mode pin setting must be selected to avoid disabling the ICAP interface. JTAG configuration will remain available because it overrides other means of configuration, and the HWICAP core will function as intended. Besides being disabled by the Boundary Scan mode pin setting, the ICAP will also be disabled if the persist bit in the device configuration logic's control register is set. When using bitgen, the Persist option must be set to No, which is the default. This option is generally specified in the bitgen.ut file in the etc. subdirectory of the EDK project. The maximum frequency of operation for ICAP on Virtex-4, Virtex-5 and Virtex-6 is 100 MHz. In case of Spartan-6 the maximum frequency of operation for ICAP is 20 MHz.
2. In case of If Virtex-4, Virtex-5 and Virtex-6 the PLB operates at less than 100MHz then the ICAP clock must be given frequency equivalent to PLB clock frequency. I.e If the PLB frequency is 90 MHz, then ICAP clock also should be 90 MHz. Suggested to derive two independent clocks from clock generator even the frequencies are same. If the PLB operates greater than 100 MHz, then the ICAP clock must be fixed to 100 MHz.
3. In case of Spartan-6 the ICAP clock must be connected to 20 MHz.

Design Constraints

Timing Constraints on the clocks:

The core has two different clock domains: SPLB_Clk and ICAP_Clk. A timing ignore constraint should be added to isolate these two clock domains. The constraints given below can be used with the XPS HWICAP core.

Period Constraints for Clock Nets

If the clock generator instantiated as below.

BEGIN clock_generator

PORT CLKOUT1 = PLB_Clk_125_0000MHz

PORT CLKOUT2 = ICAP_CLK_100MHz

END

Then the following constraints are required in the UCF.

NET "PLB_Clk_125_0000MHz" TNM = "PLBCLK_GRP";

NET "ICAP_CLK_100MHz" TNM = "ICAPCLK_GRP";

TIMESPEC TS_TIG0 = FROM "PLBCLK_GRP" TO "ICAPCLK_GRP" TIG;

TIMESPEC TS_TIG1 = FROM "ICAPCLK_GRP" TO "PLBCLK_GRP" TIG;

Timing Constraints on the ICAP interface:

If the Internal Configuration Access Port (ICAP) is used as the configuration port for partially reconfiguring the FPGA, timing constraints can be very useful to understand the potential performance of this interface. It is important to understand that the paths to the ICAP and from the ICAP are not covered by PERIOD constraints. The ICAP inputs and outputs are not considered synchronous by TRCE. This means that the inputs to and the outputs from the ICAP must be constrained using the exception constraint: NET MAXDELAY.

The following MAXDELAY constraints are required in the UCF:

NET "xps_hwicap_0/xps_hwicap_0/HWICAP_CTRL_I/icap_statemachine_I1/Icap_datain<*>" MAXDELAY = 2 ns;

NET "xps_hwicap_0/xps_hwicap_0/HWICAP_CTRL_I/icap_statemachine_I1/Icap_ce" MAXDELAY = 2 ns;

NET "xps_hwicap_0/xps_hwicap_0/HWICAP_CTRL_I/icap_statemachine_I1/Icap_we" MAXDELAY = 2 ns;

The asterisk represents the entire bus (that is, 0, 1, 2, …). The NET MAXDELAY constraint constrains only the net delay. It does not take the setup time or clock-to-out time into consideration. The ICAP component cannot be added to time groups because it is not considered a synchronous element. Therefore, the ICAP cannot be made a synchronous component by use of a TPSYNC constraint. The ICAP component is a special type of component

Design Implementation

Target Technology

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

Device Utilization and Performance Benchmarks

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

The XPS HWICAP resource utilization for various parameter combinations measured with the Virtex-4 F{GA as the target device are detailed in Table 15.

Table 15: **Performance and Resource Utilization Benchmarks on the Virtex-4 FPGA (xc4vlx40-ff1148-10)**

Parameter Values						Device Resources			Performance	
C_BRAM_SRL_FIFO_TYPE	C_SPLB_NUM_MASTERS	C_SPLB_SMALLEST_MASTER	C_SPLB_DWIDTH	C_WRITE_FIFO_DEPTH	C_READ_FIFO_DEPTH	Slices	Slice Flip-Flops	LUTs	PLB f_{MAX} (MHz)	ICAP f_{MAX} (MHz)
0	1	32	32	128	128	1,355	728	2,115	125	100
0	4	64	64	256	128	1,862	878	2,795	125	100
0	8	64	64	512	128	2,597	1,046	4,113	125	100
1	8	64	128	512	128	773	700	849	125	100
1	8	128	128	1024	128	814	716	856	125	100

XPS HWICAP resource utilization for various parameter combinations measured with Virtex-5 as the target device are detailed in Table 16.

Table 16: **Performance and Resource Utilization Benchmarks on the Virtex-5 FPGA (xc5vfx70t-ffg1136-1)**

Parameter Values						Device Resources			Performance	
C_BRAM_SRL_FIFO_TYPE	C_SPLB_NUM_MASTERS	C_SPLB_SMALLEST_MASTER	C_SPLB_DWIDTH	C_WRITE_FIFO_DEPTH	C_READ_FIFO_DEPTH	Slices	Slice Flip-Flops	LUTs	PLB f_{MAX} (MHz)	ICAP f_{MAX} (MHz)
0	1	32	32	128	128	616	728	1,405	150	100
0	4	64	64	256	128	856	878	1,963	150	100
0	8	64	64	512	128	1,191	1,046	3,117	150	100
1	8	64	128	512	128	408	700	707	150	100
1	8	128	128	1024	128	413	716	714	150	100

XPS HWICAP resource utilization for various parameter combinations measured with the Virtex-6 FPGA as the target device are detailed in Table 17.

Table 17: **Performance and Resource Utilization Benchmarks on the Virtex-6 FPGA (xc6vlx195t-ff1156-1)**

Parameter Values						Device Resources			Performance	
C_BRAM_SRL_FIFO_TYPE	C_SPLB_NUM_MASTERS	C_SPLB_SMALLEST_MASTER	C_SPLB_DWIDTH	C_WRITE_FIFO_DEPTH	C_READ_FIFO_DEPTH	Slices	Slice Flip-Flops	LUTs	PLB f_{MAX} (MHz)	ICAP f_{MAX} (MHz)
0	1	32	32	128	128	340	714	900	160	100
0	4	64	64	256	128	378	739	1,035	160	100
0	8	64	64	512	128	460	755	1,263	160	100
1	8	64	128	512	128	321	694	752	160	100
1	8	128	128	1024	128	326	710	776	160	100

XPS HWICAP resource utilization for various parameter combinations measured with the Spartan-6 FPGA as the target device are detailed in Table 18.

Table 18: **Performance and Resource Utilization Benchmarks on the Spartan-6 FPGA (xc6slx45t-fgg484-1)**

Parameter Values						Device Resources			Performance	
C_BRAM_SRL_FIFO_TYPE	C_SPLB_NUM_MASTERS	C_SPLB_SMALLEST_MASTER	C_SPLB_DWIDTH	C_WRITE_FIFO_DEPTH	C_READ_FIFO_DEPTH	Slices	Slice Flip-Flops	LUTs	PLB f_{MAX} (MHz)	ICAP f_{MAX} (MHz)
0	1	32	32	128	128	277	622	720	100	20
0	4	64	64	256	128	326	646	803	100	20
0	8	64	64	512	128	367	662	930	100	20
1	8	64	128	512	128	279	632	684	100	20
1	8	128	128	1024	128	283	648	701	100	20

System Performance

To measure the system performance (Fmax) of the XPS HWICAP core, it was added to a Virtex-4 FPGA system, a Virtex-5 FPGA system, Virtex-6 FPGA system, and a Spartan-6 FPGA system as shown in Figure 15, Figure 16, Figure 17 and Figure 18.

Because the XPS HWICAP 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.

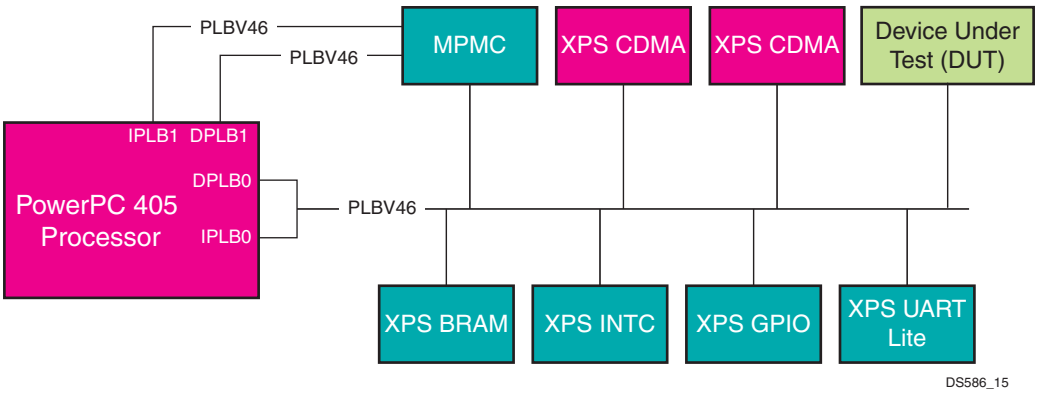

Figure 15: **Virtex-4 FX FPGA System with the XPS HWICAP as the DUT**

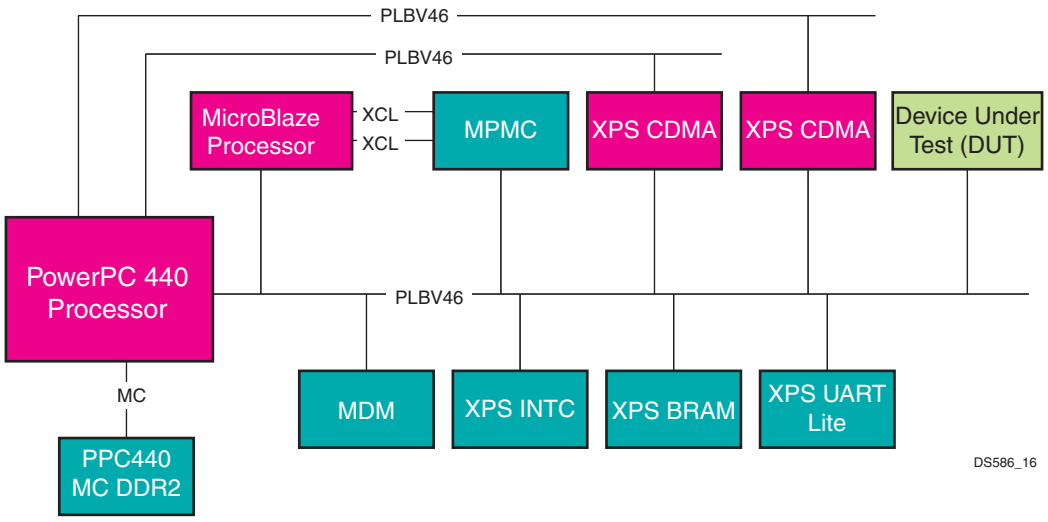

Figure 16: **Virtex-5 FXT FPGA System with the XPS HWICAP as the DUT**

Virtex-5 FXT FPGA System with the XPS HWICAP as the DUT

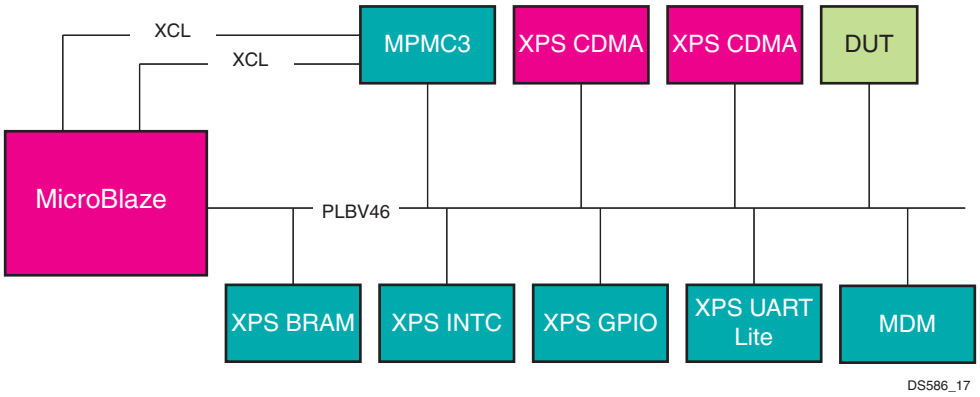

Figure 17: **Virtex-6 FPGA System with the XPS HWICAP as the DUT**

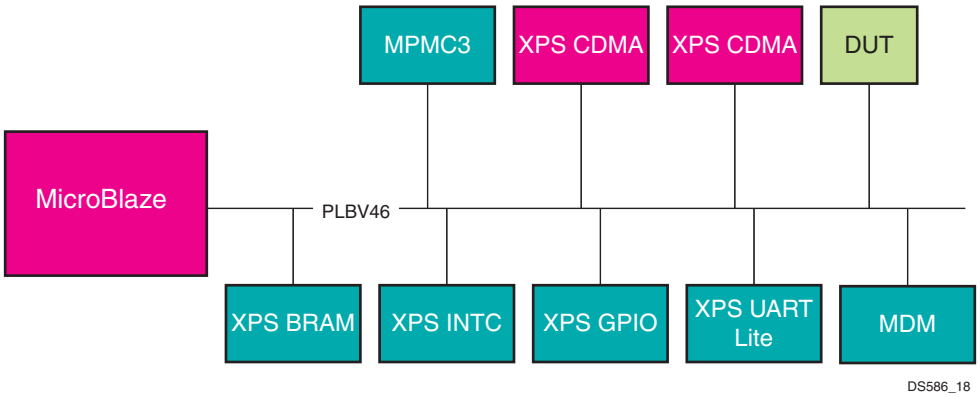

Figure 18: **Spartan-6 FPGA System with the XPS HWICAP as the DUT**

The target FPGA was then filled with logic to drive the LUT and BRAM 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 19.

Table 19: **XPS HWICAP Core System Performance**

Target FPGA	Target F_{MAX} (MHz)
V4FX60 -10	100
V5FX70T -1	125
V6LX130t - 1	150
S6LX45t - 2	100

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

Reference Documents

The following document contains reference information important to understanding the XPS HWICAP core :

1. IBM CoreConnect 128-Bit Processor Local Bus, Architectural Specification (v4.6)
2. PLBv46_Slave_Burst_v1_00_a (DS562) Design Specification

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*.

Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx ISE Design Suite Embedded Edition software under the terms of the Xilinx End User License. The core is generated using the Xilinx ISE Embedded Edition software (EDK).

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE modules and software, please contact your local Xilinx sales representative.

Revision History

Date	Version	Revision
08/23/07	1.0	Initial release
10/18/07	1.1	Added Table 16 Performance and Resource Utilization Benchmarks on Virtex-5.
04/24/09	1.2	Replaced references to supported device families and tool name(s) with hyperlink to PDF file.
06/03/09	1.3	Added ICAP_Clk to the design to provide separate clock for ICAP and updated top-level block diagram and I\O Signals table; replaced SRL FIFO's with FIFO Generator based Asynchronous FIFOs; added parameter C_BRAM_SRL_FIFO_TYPE to select the FIFO type shown in the Design Parameters table; updated the Device Utilization and Performance Benchmarks section with the latest values.
11/19/09	1.4	Virtex-6 family support added; asdded Table notes 4 and 5 to the Design Parameters table. added.
05/02/10	1.5	Spartan-6 family support added.
7/23/10	1.6	Updated for 12.2 release.

Notice of Disclaimer