LogiCORE IP AXI HWICAP (v2.03.a)

DS817 July 25, 2012 **Product Specification**

Introduction

The Xilinx® LogiCORE™ IP Advanced eXtensible Interface (AXI) HWICAP (Hardware Internal Configuration Access Port) core for the AXI Interface enables an embedded microprocessor, such as the MicroBlaze™ processor, to read and write the FPGA configuration memory through the Internal Configuration Access Port (ICAP/ICAPE2). This enables you to write software programs that modify the circuit structure and functionality during the operation of the circuit.

Features

- Supports resource reading
- Supports long frame reads
- AXI Interface is based on the AXI4-Lite specification
- Partial bitstream loading
- Enables Read/Write of Configurable Logic Block (CLB) Lookup Tables (LUTs)
- Enables Read/Write of CLB Flip-Flop properties
- Supports the MicroBlaze embedded processor

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	Zynq™-7000[2], Virtex®-7[3], Kintex™-7[3], Artix™-7, Virtex-6[4], Spartan®-6[5]
Supported User Interfaces	AXI4-Lite
Resources	See Table 16, Table 17 and Table 18.
Provided with Core	
Design Files	ISE®: VHDL Vivado™: VHDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	Not Provided
Supported S/W Drivers[6]	Standalone and Linux
Tested Design Flows[7]	
Design Entry	Xilinx Platform Studio (XPS) Vivado Design Suite[8]
Simulation	Mentor Graphic ModelSim ISE Simulator (ISim)
Synthesis	Xilinx Synthesis Technology (XST) Vivado Synthesis
Support	
Provided by Xilinx@ www.xilinx.com/support	

Notes:

1. For a complete list of supported derivative devices, see Embedded Edition Derivative Device Support.
2. Supported in ISE Design Suite implementations only.
3. For more information, see DS180 *7 Series FPGAs Overview Advanced Product Specification*.
4. For more information, see the DS150 *Virtex-6 Family Overview Product Specification*.
5. For more information, see DS160 *Spartan-6 Family Overview Product Specification*.
6. Standalone driver information can be found in the EDK or SDK installation directory.
   See xilinx_drivers.htm in *<install_directory>*/doc/usenglish.
   Linux OS and driver support information is available from wiki.xilinx.com. (Not updated for Kintex-7, Virtex-6, and Spartan-6 FPGAs in the Linux driver)
7. For a listing of the supported tool versions, see the Xilinx Design Tools: Release Note Guide.
8. Supports only 7 series devices.

Functional Description

The AXI HWICAP controller provides the interface necessary to transfer bitstreams to and from the ICAP. For writes to the ICAP, the required bitstream data from main memory is stored within a Write First In First Out (FIFO), from where it can be sent to the ICAP. The AXI 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 Central Processing Unit (CPU) is then able to read the frame data directly from the Read FIFO.

Sample Applications

- A Digital Signal Processing (DSP) system, such as a 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 logic. The crossbar connections are reconfigured at run time by reconfiguring the routing multiplexers.

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

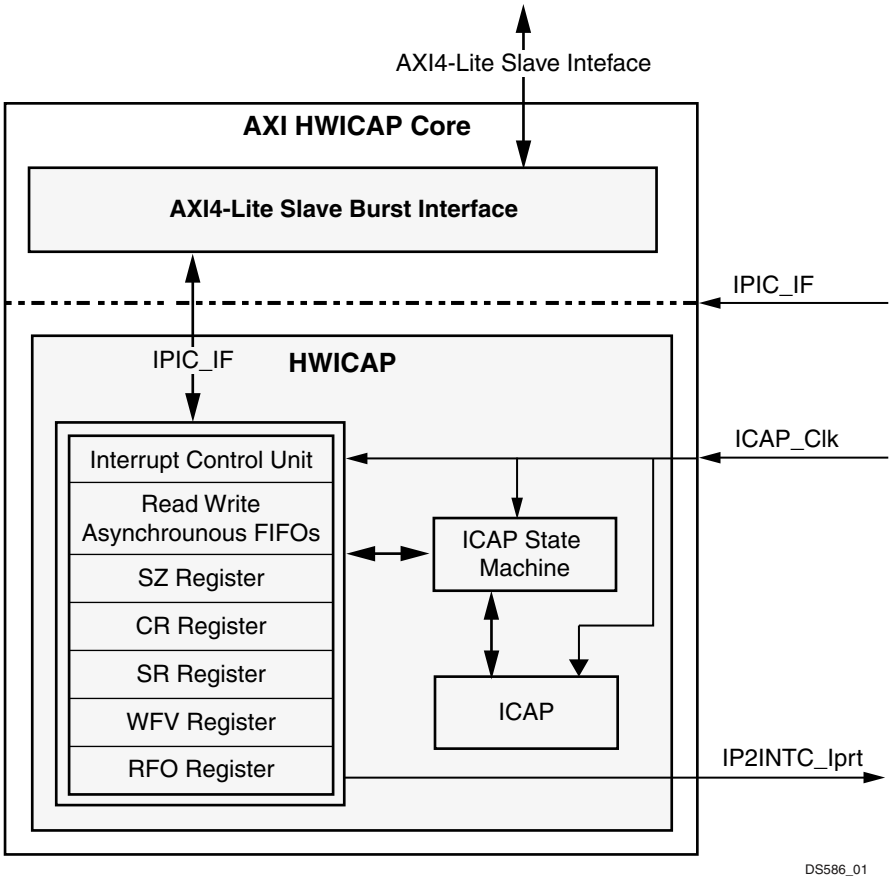

Figure 1: **Top Level Block Diagram for the AXI HWICAP Core**

AXI4-Lite Interface Module

The AXI4-Lite Interface Module provides the bidirectional interface between the HWICAP core and the AXI. The base element of the AXI Interface Module is the slave attachment, which provides the basic functionality of AXI slave operation.

IPIC_IF Module

The IPIC_IF module incorporates logic to acknowledge the write and read transactions initiated by the AXI slave module to write or 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 stores the configuration locally. The processor writes the configuration into the write FIFO. Simultaneously, the data stored in the write FIFO is transferred to the ICAP. The processor reads the configuration from the ICAP, which is stored inside the read FIFO. FIFOs are used because the rate of data flow from the processor interface is different from the ICAP interface. FIFO depth can be parameterizable using the generics C_WRITE_FIFO_DEPTH and C_READ_FIFO_DEPTH.

The operation of ICAPE2 in Virtex-7 and Kintex-7 devices is very similar to that of ICAP in Virtex-6 devices. As such all the Virtex-6 device related items in this document are also applicable to Virtex-7 and Kintex-7 devices.

The FIFO data width is dependent on the device family. For the Virtex-7, Kintex-7, and Virtex-6 Field Programmable Gate Array (FPGAs), the FIFO data width is 32 bits; for the Spartan-6 FPGA, the FIFO data width is 16 bits.

Note: With the Virtex-7, Kintex-7, and Virtex-6 devices, if `S_AXI_AClk` is greater than 100 MHz, connect the `ICAP_Clk` to 100 MHz. If the `S_AXI_AClk` is less than, or equal to, 100 MHz, connect the `ICAP_Clk` to the frequency equivalent to the `S_AXI_AClk` frequency. The Virtex-6 devices have a limiting frequency of maximum 100 MHz on `ICAP_Clk`.

Note: `ICAP_Clk` must be less than or equal to 20 MHz, 12 MHz, or 4 MHz in Spartan-6 FPGAs. The Spartan-6 LX4 to LX75/T devices have a limiting frequency of 20 MHz. The Spartan-6 LX100/T to LX150/T devices have a limiting frequency of 12 MHz. The Spartan-6 Low Power devices have a limiting frequency of 4 MHz.

Input/Output Signals

The AXI4 HWICAP core Input/Output (I/O) signals are 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
P2	EOS_IN	NA	I	-	End of Startup. This pins has to be connected to a valid EOS signal if Startup primitive is not included in the HWICAP IP
AXI Bus Request and Qualifier Signals					
P3	S_AXI_ACLK	AXI	I	-	AXI Clock
P4	S_AXI_ARESETN	AXI	I	-	AXI Reset; Active-Low
P5	S_AXI_AWADDR(C_S_AXI_ADDR_WIDTH-1:0)	AXI	I	-	AXI write address: The write address bus gives the address of the write location.

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

Port	Signal Name	Interface	I/O	Initial State	Description
P6	S_AXI_AWVALID	AXI	I	-	AXI Write address valid: This signal indicates that a valid write address and control information are available.
P7	S_AXI_AWREADY	AXI	O	-	Write address ready: This signal indicates that the slave is ready to accept an address and associated control signal.
P8	S_AXI_WDATA[(C_S_AXI_DATA_WIDTH-1): 0]	AXI	I	-	Write data
P9	S_AXI_WSTRB[(C_S_AXI_DATA_WIDTH/8)-1):0]	AXI	I	-	Write strobes: This signal indicates which byte lanes to update in memory.
P10	S_AXI_WVALID	AXI	I	-	Write valid: This signal indicates that valid write data and strobes are available.
P11	S_AXI_WREADY	AXI	O	-	Write ready: This signal indicates that the slave can accept the write data.
P12	S_AXI_BRESP[1:0]	AXI	O	0	Write response: This signal indicates the status of the write transaction. • "00" - OKAY (normal response) • "10" - SLVERR (error response) • "11" - DECERR (not issued by core)
P13	S_AXI_BVALID	AXI	O	0	Write response valid: This signal indicates that a valid write response is available.
P14	S_AXI_BREADY	AXI	I	-	Response ready: This signal indicates that the master can accept the response information.
P15	S_AXI_ARADDR[(C_S_AXI_ADDR_WIDTH-1): 0]	AXI	I	-	Read Address: The read address bus gives the address of a read transaction.
P16	S_AXI_ARVALID	AXI	I		Read address valid: This signal indicates, when HIGH, that the read address and control information is valid and remains stable until the address acknowledgement signal, S_AXI_ARREADY, is high.
P17	S_AXI_ARREADY	AXI	O	1	Read address ready: This signal indicates that the slave is ready to accept and address and associated control signals.
P18	S_AXI_RDATA[(C_S_AXI_DATA_WIDTH-1): 0]	AXI	O	0	Read data
P19	S_AXI_RRESP[1:0]	AXI	O	0	Read response: This signal indicates the status of the read transfer. • "00" - OKAY (normal response) • "10" - SLVERR (error condition) • "11" - DECERR (not issued by core)
P20	S_AXI_RVALID	AXI	O	0	Read valid: This signal indicates that the master can accept the read data and response information.
P21	S_AXI_RREADY	AXI	I	-	Read ready: This signal indicates that the master can accept the read data and response information.

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

Port	Signal Name	Interface	I/O	Initial State	Description
System Signals					
P22	IP2INTC_Irpt	AXI	O	0	AXI HWICAP Interrupt

Notes:

1. `ICAP_Clk` must be less than or equal to 20 MHz, 12 MHz, or 4 MHz in Spartan-6 FPGAs. The 7-series and Virtex-6 devices have a limiting frequency of maximum 100 MHz on `ICAP_Clk`. See the DC and Switching Characteristics document for more details.

Design Parameters

To allow you to create a core that is uniquely tailored for the system, certain features are parameterizable in the core. This allows you 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 shown in Table 2.

In addition to the parameters listed in Table 2, there are also parameters that are inferred for each AXI interface in the Embedded Development Kit (EDK) tools. Through the design, these EDK-inferred parameters control the behavior of the AXI Interconnect. For a complete list of the interconnect settings related to the AXI interface, see the DS768 AXI Interconnect IP data sheet.

Table 2: **Design Parameters**

Generic	Parameter Description	Parameter Name	Allowable Values	Default Value	VHDL Type
AXI Parameters					
G1	AXI data bus width	C_S_AXI_DATA_WIDTH	32	32	integer
G2	AXI address Bus Width	C_S_AXI_ADDR_WIDTH	9	9	integer
System Parameters					
G3	Write FIFO depth	C_WRITE_FIFO_DEPTH(1)(4)	64, 128, 256, 512,1024	64	integer
G4	Read FIFO depth	C_READ_FIFO_DEPTH(1)(4)	128, 256	128	integer
G5	Select FIFO type(2)	C_BRAM_SRL_FIFO_TYPE	0,1	1	integer
G6	ICAP Data Width	C_ICAP_DWIDTH_S	X8, X16, X32	X32	String
G7	Enable Lite Mode(5)	C_MODE	0, 1	0	integer
G8	Disable Read	C_NOREAD	0, 1	0	integer
G9	Xilinx FPGA Family	C_FAMILY	spartan6, virtex6, virtex7, kintex7, zynq	virtex6	string
G10	Simulation(3)	C_SIMULATION	1, 2	2	integer
G11	Include Startup in HWICAP IP	C_INCLUDE_STARTUP	1, 0	0	integer
G12	ID of the device(3)	C_DEVICE_ID	Any 32-bit hex number	4224093	std_logic_vector
G13	Async Operation	C_ENABLE_ASYNC	0, 1	0	Integer

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

Generic	Parameter Description	Parameter Name	Allowable Values	Default Value	VHDL Type

Notes:

1. This parameter must be set to 256 if the C_FAMILY = virtex6/spartan6, as the frame size in the Virtex-6 and Spartan-6 families is 162 bytes.
2. The parameter C_BRAM_SRL_FIFO_TYPE selects the FIFO type to be block RAM (1) or Distributed RAM (0).
3. The parameter C_SIMULATION must be set to 2 for implementation and simulations using the ICAP UNISIM model. The parameter C_DEVICE_ID can be used to pass a different value of device ID for purpose of simulation.
4. The actual depth of the Write/Read FIFO is one less than the parameter that defines the depth of the FIFOs (for example, either C_WRITE_FIFO_DEPTH or C_READ_FIFO_DEPTH).
5. Lite Mode, when enabled, does not instantiate the WrFIFO. This is to reduce the device utilization. Further, if need be, the Reads can be totally disabled in this mode.
6. The parameter C_ENABLE_ASYNC is applicable when the IP is invoked in Vivado.

Parameter - Port Dependencies

The dependencies between the AXI HWICAP core design parameters and the I/O signals are described in Table 3. When certain features are parameterized out of the design, the related logic is no longer 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				
G2	C_S_AXI_ADDR_WIDTH	P4, P14	-	Affects the number of bits in address bus
G1	C_S_AXI_DATA_WIDTH	P7, P8,P17	-	Affects the number of bits in data bus
I/O Signals				
P4	S_AXI_AWADDR[(C_S_AXI_ ADDR_WIDTH-1): 0]	-	G4	Width of the S_AXI_AWADDR varies with C_S_AXI_ADDR_WIDTH
P7	S_AXI_WDATA[(C_S_AXI_ DATA_WIDTH - 1): 0]	-	G3	Width of the S_AXI_WDATA varies according to C_S_AXI_DATA_WIDTH
P8	S_AXI_WSTRB[((C_S_AXI_ DATA_WIDTH/8)-1): 0]	-	G3	Width of the S_AXI_WSTRB varies according to C_S_AXI_DATA_WIDTH
P14	S_AXI_ARADDR[(C_S_AXI_ ADDR_WIDTH -1): 0]	-	G4	Width of the S_AXI_ARADDR varies with C_S_AXI_ADDR_WIDTH
P17	S_AXI_RDATA[(C_S_AXI_ DATA_WIDTH-1): 0]	-	G3	Width of the S_AXI_RDATA varies according to C_S_AXI_DATA_WIDTH

Register Definition

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

Table 4: **Internal Registers**

C_BASEADDR + Address	Register Name	Access	Default Value	Description
C_BASEADDR + 0x01C	Global Interrupt Enable Register	Read/Write	0x0	Global interrupt enable register
C_BASEADDR + 0x020	IP Interrupt Status Register	Read/Write	0x0	IP interrupt status register
C_BASEADDR + 0x028	IP Interrupt Enable Register	Read/Write	0x0	IP interrupt enable register
C_BASEADDR + 0x100	Write FIFO Register	Write	0x0	Data write register
C_BASEADDR + 0x104	Read FIFO Register	Read	0x0	Data read register
C_BASEADDR + 0x108	Size Register	Write	0x0	Size register
C_BASEADDR + 0x10C	Control Register	Read/Write	0x0	IP control register
C_BASEADDR + 0x110	Status Register	Read	0x0	Status register
C_BASEADDR + 0x114	Write FIFO Vacancy Register	Read	(1)	Write FIFO vacancy register
C_BASEADDR + 0x118	Read FIFO Occupancy Register	Read	0x0	Read FIFO occupancy register
C_BASEADDR + 0x11C	Abort Status Register	Read	0x0	Abort Status Register

Notes:

1. This value is based on the Write FIFO size. For example, if the Write FIFO is of 1024 depth, this value would be 3FF.

Write FIFO Register

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

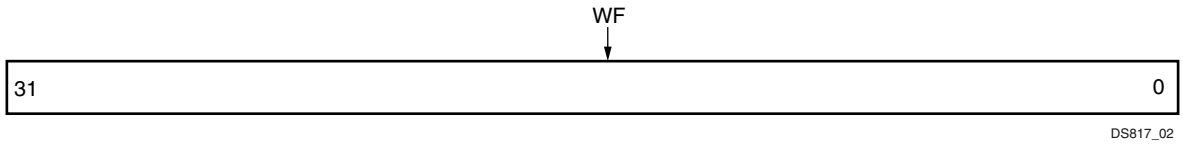

Figure 2: **Write FIFO (WF)**

Table 5: **Write FIFO Bit Definitions (C_BASEADDR + 0x100)**

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

Read FIFO Register

The Read FIFO (RF) register shown in Figure 3 is a 32-bit Read FIFO. 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.

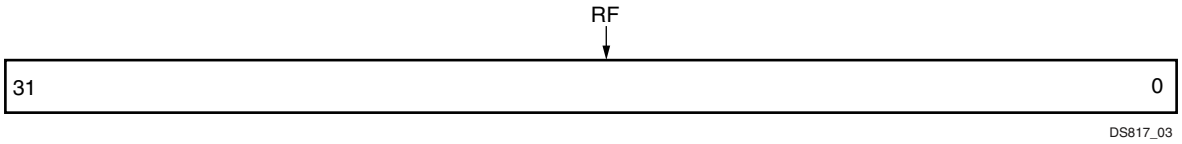

Figure 3: **Read FIFO (RF)**

Table 6: **Read FIFO Bit Definitions (C_BASEADDR + 0x104)**

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

Size Register

The Size (SZ) register shown in Figure 4 is a 12-bit write register that 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 the C_BASEADDR value is as shown in Table 4.

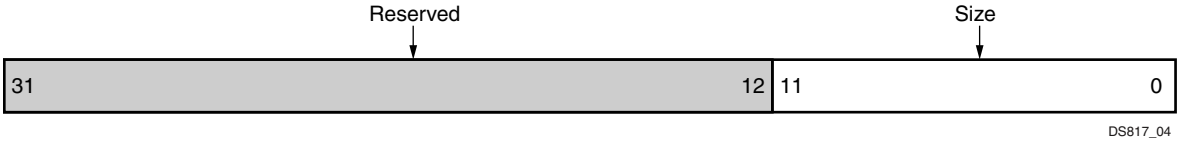

Figure 4: **Size Register (SZ)**

Table 7: **Size Register Bit Definitions (C_BASEADDR + 0x108)**

Bit	Name	Access	Reset Value	Description
31 - 12	Reserved	N/A	0	Reserved bits
11 - 0	Size	Write	0	Number of words to be transferred from the ICAP to the FIFO

Control Register

The Control Register (CR) shown in Figure 5 is a 32-bit write register that 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 the C_BASEADDR value are shown in Table 4.

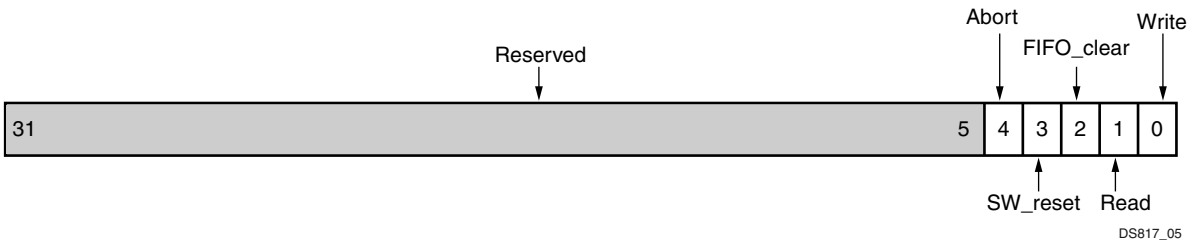

Figure 5: **Control Register (CR)**

Table 8: **Control Register Bit Definitions (C_BASEADDR + 0x10C)**

Bit	Name	Access	Reset Value	Description
31 - 5	Reserved	N/A	'0'	Reserved bits
4	Abort	Read/Write	'0'	'1' = Aborts the read or write of the ICAP and clears the FIFOs
3	SW_reset	Read/Write	'0'	'1' = Resets all the registers
2	FIFO_clear	Read/Write	'0'	'1' = Clears the FIFOs
1	Read	Read/Write	'0'	'1' = Initiates readback of bitstream in to the Read FIFO
0	Write	Read/Write	'0'	'1' = Initiates writing of bitstream in to the ICAP

Status Register

The Status Register (SR) shown in Figure 6 is a 32-bit read register that contains the ICAP status bits. The bit definitions for the register are shown in Table 9. The offset and accessibility of this register from the C_BASEADDR value are shown in Table 4.

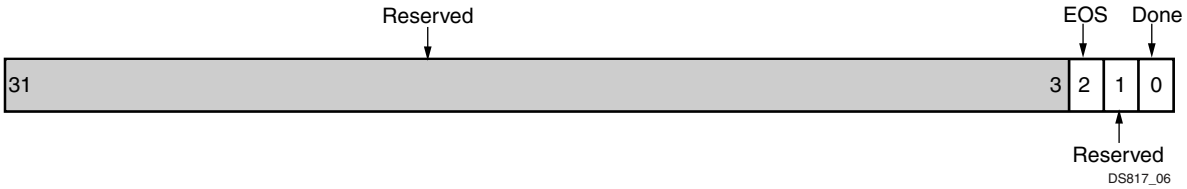

Figure 6: **Status Register (SR)**

Table 9: **Status Register Bit Definitions (C_BASEADDR + 0x110)**

Bit	Name	Access	Reset Value	Description
31 - 3	Reserved	N/A	0	Reserved bits
2	EOS Bit	Read	1	Indicates that the EOS is complete
1	Reserved	N/A	0	Reserved
0	Done	Read	1	AXI HWICAP done with configuration or read

Write FIFO Vacancy Register

The Write FIFO Vacancy (WFV) register shown in Figure 7 is a 32-bit read register that indicates the 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 are shown in Table 4.

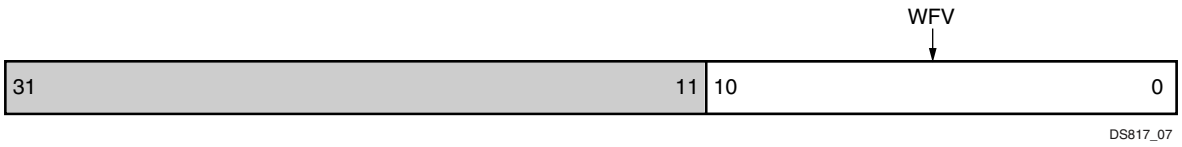

Figure 7: **Write FIFO Vacancy Register (WFV**

Table 10: **Write FIFO Vacancy Register Bit Definitions (C_BASEADDR + 0x114)**

Bit	Name	Access	Reset Value	Description
31 - 11	Reserved	NA	-	Reserved
10 - 0	WFV	Read	(1)	Write FIFO Vacancy

Notes:
1. This value is based on the Write FIFO size. For example, if the Write FIFO is of 1024 depth, this value would be 3FF.
2. This register is of no importance when C_MODE is set to 1.

Read FIFO Occupancy Register

The Read FIFO Occupancy (RFO) register shown in Figure 8 is a 32-bit read register that indicates occupancy of the read FIFO. The actual depth of the Read FIFO is one less than the C_READ_FIFO_DEPTH. The bit definitions for the register are shown in Table 11. The offset and accessibility of this register from C_BASEADDR value are shown in Table 4. This register is of no importance when C_NOREAD is set to 1.

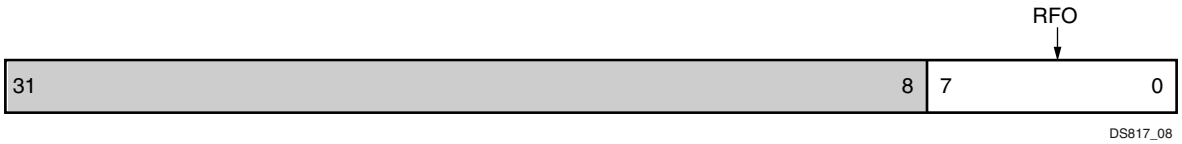

Figure 8: **Read FIFO Occupancy Register (RFO)**

Table 11: **Read FIFO Occupancy Register Bit Definitions (C_BASEADDR + 0x118)**

Bit	Name	Access	Reset Value	Description
31 - 8	Reserved	N/A	-	Reserved
7 - 0	RFO	Read	0	Read FIFO Occupancy

Global Interrupt Enable Register

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

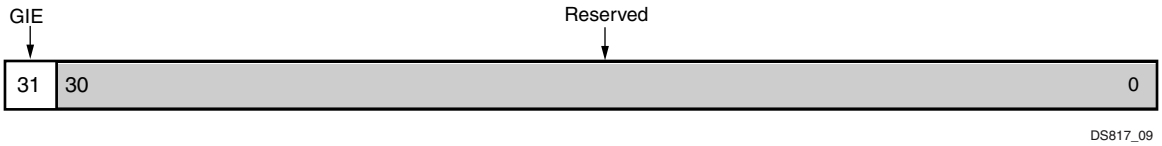

Figure 9: **Global Interrupt Enable Register (GIER)**

Table 12: **Global Interrupt Enable Register Bit Definitions (C_BASEADDR + 0x01C)**

Bit(s)	Name	Access	Reset Value	Description
31	GIE	R/W	'0'	'0' = Disabled '1' = Enabled
30 - 0	Reserved	N/A	N/A	Reserved

IP Interrupt Status Register

Four unique interrupt conditions are possible in the 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.

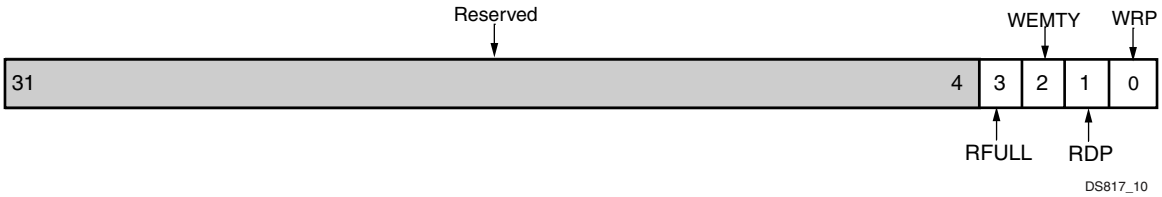

Figure 10: **IP Interrupt Status Register (IPISR)**

Table 13: **IP Interrupt Status Register Bit Definitions (C_BASEADDR + 0x020)**

Bit(s)	Name	Access	Reset Value	Description
31 - 4	Reserved	N/A	N/A	Reserved
3	RFULL	R/TOW(1)	'0'	Read FIFO full
2	WEMTY	R/TOW(1)	'0'	Write FIFO empty
1	RDP	R/TOW(1)	'0'	Interrupt set and remains set if the read FIFO occupancy is greater than half of the read FIFO size.
0	WRP	R/TOW(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.
2. WRP and WEMTY bits are of no importance when C_MODE is 1. It is recommended that these interrupts be disabled when HWICAP is configured in 'Lite Mode'

IP Interrupt Enable Register

The IP Interrupt Enable Register (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 (C_BASEADDR + 0x028)**

Bit(s)	Name	Access	Reset Value	Description
31 - 4	Reserved	N/A	N/A	Reserved
3	RFULLE	R/W(1)	'0'	Read FIFO full interrupt enable
2	WEMTYE	R/W(1)	'0'	Write FIFO empty interrupt enable
1	RDPE	R/W(1)	'0'	Read FIFO occupancy greater than half of its size interrupt enable
0	WRPE	R/W(1)	'0'	Write FIFO occupancy less than half of its size interrupt enable

Notes:

1. Writing a 1 to this bit enables the particular interrupt. Writing 0 to this bit disables the particular interrupt.
2. WRP and WEMTY bits are of no importance when C_MODE is 1. It is recommended that these interrupts be disabled when HWICAP is configured in 'Lite Mode'.

Abort Status Register

An abort is an interruption which occurs in the configuration or read-back sequence when the state of `icap_we` changes while `icap_ce` is asserted. During a configuration Abort, internal status is driven onto the `icap_dout[7:0]` pins over the next four clock cycles. These four 8-bit words are stored in the abort status register with the first status word available in the 31-24 bits while the last status word is in the 7-0 bits. After the abort sequence finishes, you can re-synchronize the configuration logic and resume configuration. Abort enables you to know the status of the ICAP during the configuration or reading the configuration.

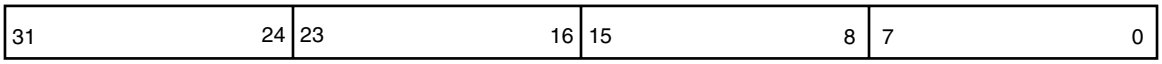

Figure 12: **Abort Status Register**

Table 15: **Abort Status Register (C_BASEADDR + 11C)**

Bit	Name	Access	Reset Value	Description
31 - 24	Status 0	R	0	First Abort Status word
23 - 16	Status 1	R	0	Second Abort Status word
15 - 8	Status 2	R	0	Third Abort Status word
7-0	Status 3	R	0	Fourth Abort Status Word

Customizing and Generating the Core

The AXI HWICAP can be found in **Embedded_Processing\AXI_Peripheral\Low_Speed_Peripheral** in the Vivado IP Catalog.

To access the AXI HWICAP, do the following:

1. Open an existing project or create a new project in the Vivado IP Catalog.
2. Open the IP catalog and navigate to any of the above taxonomies.
3. Double-click **AXI HWICAP** to bring up the AXI HWICAP GUI.

Vivado IP Catalog System Parameter Screen

The AXI HWICAP GUI contains one screen (Figure 13) that displays information about the core, allows configuration of the core, and provides the ability to generate the cores.

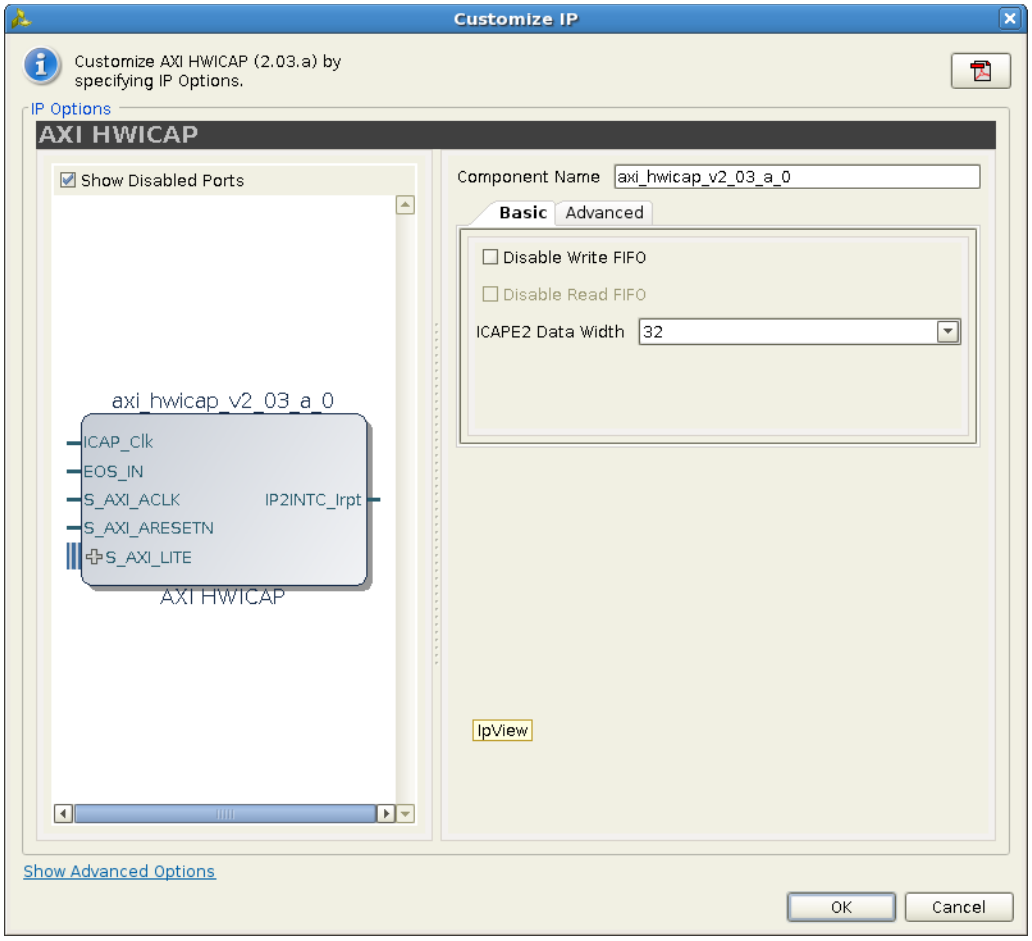

Figure 13: **Vivado IP Catalog Parameter Screen (Basic)**

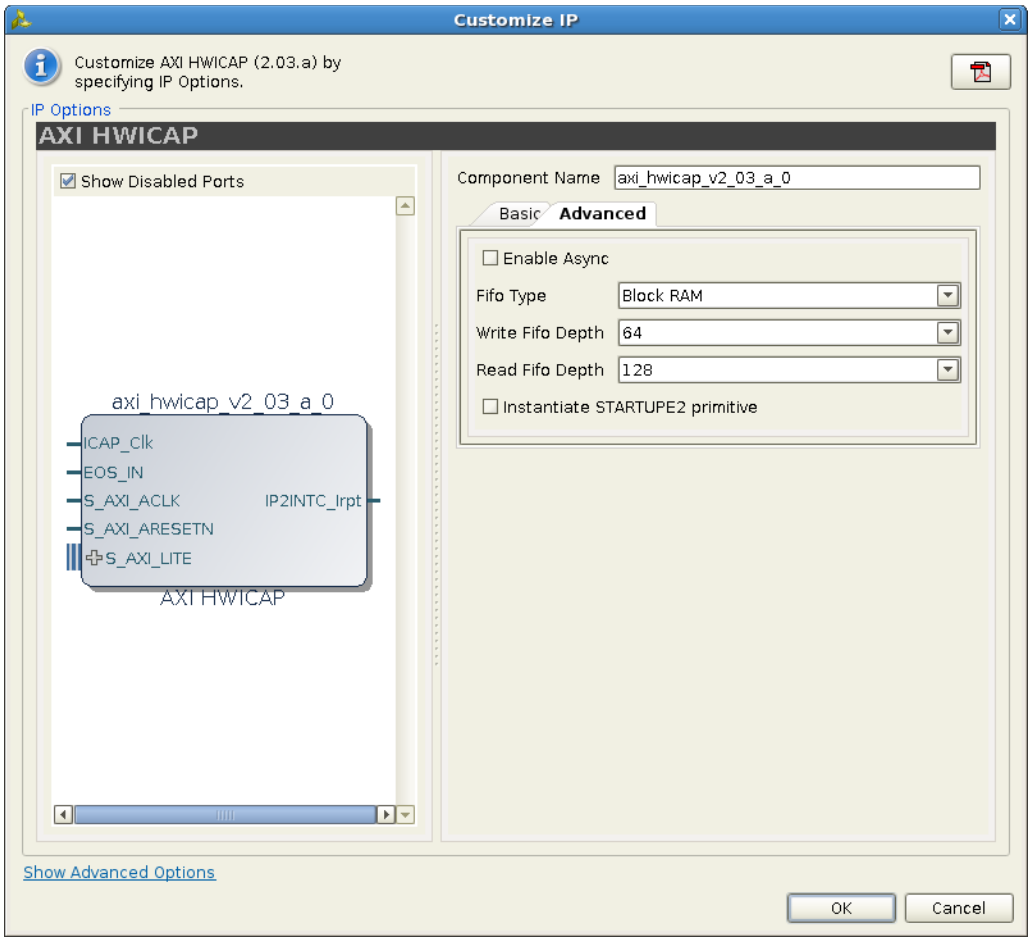

Figure 14: **Vivado IP Catalog Parameter Screen (Advanced)**

In most cases the HWICAP can be configured with the option specified on the **Basic** tab. All the options available in this GUI are essentially similar to that in the EDK XPS GUI. The GUI has been split into two tabs for sake of simplicity and ease-of-use.

Component Name

The base name of the output files generated for the core. Names must begin with a letter and can be composed of any of the following characters: a to z, 0 to 9, and "_".

Basic Options

The following subsections describe the fundamental options that affect the AXI HWICAP core.

Disable Write FIFO

Checking this option disables the instantiation of Write FIFO thereby reducing the utilization.

Disable Read FIFO

Checking this option disables the Read FIFOs. This option is available only when instantiation of Write FIFO is disabled. Reading back the data from ICAP is not possible when this is selected.

ICAPE2 Data Width

Select the width of the ICAPE2 primitive. Available options are 8, 16, 32. By default this is set to 32.

Advance Options

The following subsections describe the fundamental options that affect the AXI HWICAP core.

Enable Aysnc

Select this box if the ICAP_Clk and AXI Clocks are asynchronous.

FIFO Type

Choose the type of FIFO to be instantiated. You can choose between Block RAM or Distributed RAM type of FIFO.

Instantiate STARTUPE2 Primitive

Select this box to instantiate the STARTUPE2 primitive. The EOS output from STARTUPE2 is used to control the user access to ICAPE2. EOS output signifies the end of device configuration. ICAPE2 is allowed to access only after EOS goes high.

Output Generation (Vivado)

The output hierarchy when the core is generated from Vivado tool is shown in Figure 15.

The component name of the IP generated is axi_hwicap_v2_03_a_0.

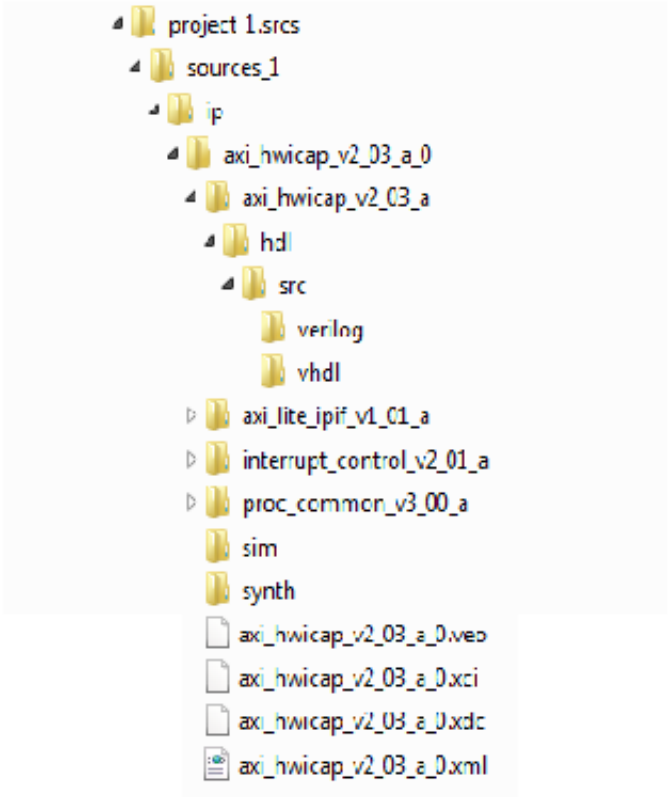

Figure 15: **Output Hierarchy**

Vivado IP Catalog delivers the RTL files of the axi_hwicap core as well as all its helper cores. The axi_lite_ipif_v1_01_a, interrupt_control_v2_01_a, proc_common_v3_00_a are the helper cores used by the axi_hwicap. The RTL files of axi_hwicap are delivered under /ip/axi_hwicap_v1_00_a/hdl/src/vhdl. The sim and synth folders contain the wrappers for simulation and synthesis respectively. The tool also delivers the instantiation template file .veo/.vho and the xdc constraints file.

Software Support

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

User Application Hints

The use of the AXI HWICAP is outlined in the following steps.

Read or Configuration (write)

- Write the bitstream in to the Write FIFO Register to configure. Get the bitstream from the Read FIFO Register to read.
- Write in to the Control Register to initiate the read or write of a bitstream. The CR register determines the direction of the data transfer. Writing "0x00000001" into the Control Register initiates the write configuration. Writing "0x00000002" into the Control Register initiates the read. This is shown as `rnc[7:0]` in Figure 16, Figure 17 and Figure 18.
- Done bit in the Status Register indicates whether the ICAP interface is busy with writing or reading data from or to the ICAP bus. It does not indicate that the read or configuration with ICAP is completed successfully.
- Hardware clears the Control Register 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 is cleared.

Write Sequence for 'Lite Mode'

- Write the instruction into the Write FIFO Register to configure.
- Write into the Control Register to initiate the write of the instruction. The CR register determines the direction of the data transfer. Writing "0x00000001" into the Control Register initiates the write configuration.
- Hardware clears the Control Register bits after the successful completion of the read or configuration.
- Write the second instruction to Write FIFO Register. Write to Control Register to initiate the write to the ICAP.
- Continue the process until all the instructions have been written to ICAP.

Abort

- Write into the Control Register to initiate the read or write of bitstream. The CR register determines the direction of the data transfer. Writing "0x00000001" into the Control Register initiates the configuration. Writing "0x00000002" into the Control Register initiates the read. This is shown as `rnc[7:0]` in Figure 16, Figure 17 and Figure 18.
- Write the bitstream into the Write FIFO Register to configure. Get the bitstream from the Read FIFO Register to read.
- Write a 1 into the fifth bit of the Control Register to initiates abort.
- Done bit in the Status Register indicates whether the ICAP interface is busy with writing/reading data from/to the ICAP bus. It does not indicate that the read/configuration with ICAP is completed successfully.
- Hardware clears the Control Register 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 is cleared.

Timing Diagrams

The timing diagrams shown in Figure 16, Figure 17, and Figure 18 show the read and write cycles of AXI HWICAP core for the Virtex-6 and Spartan-6 FPGAs.

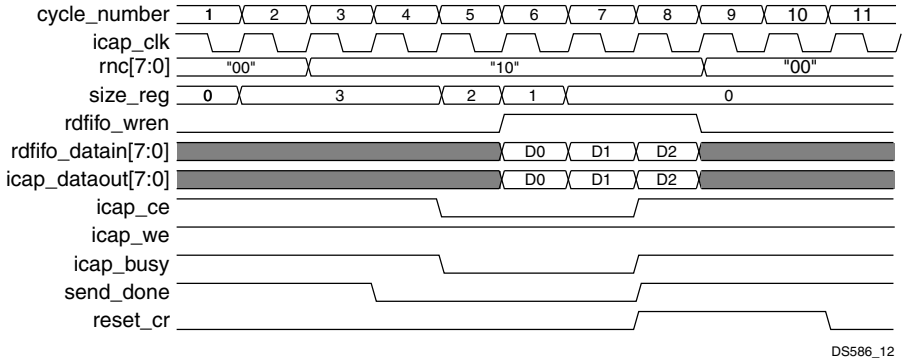

Figure 16: **Read Cycle for Virtex-6 Devices**

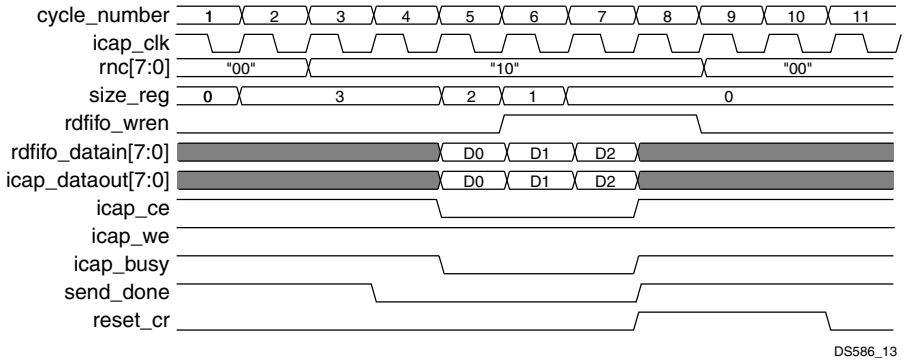

Figure 17: **Read Cycle for Spartan-6 Devices**

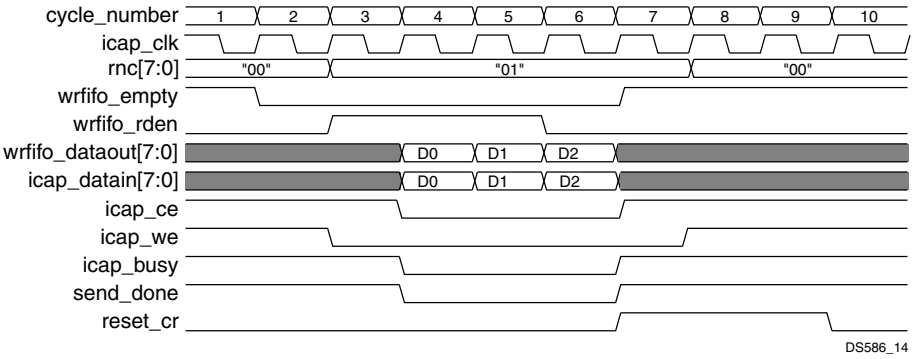

Figure 18: **Write Cycle**

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 data that is 1-bit wide and spans the full column of the FPGA. Configuration frames in the CLB space also contain Input Output Block (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 Random Access Memory (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 continues to operate unaffected.

There are two exceptions to this rule: 1) when the LUTs are configured in the Shift Register mode and 2) when the LUTs are configured 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 is interrupted and it loses its state. To resolve the issue, 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 maintain their state.

Important Notes:

- The HWICAP core uses the ICAP found inside 7 series, 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 Joint Test Action Group (JTAG) or “Boundary Scan” configuration mode pin setting (M2:M0 = 101) disables the ICAP interface. If JTAG is used as the primary configuration method, another mode pin setting must be selected to avoid disabling the ICAP interface. JTAG configuration remains 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 is also 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.
- For Virtex-6 devices, the ICAP clock must be given a frequency equivalent to the AXI clock frequency if the AXI operates at less than 100 MHz. For example, if the AXI frequency is 90 MHz, the ICAP clock should also be 90 MHz. Xilinx recommends deriving two independent clocks from the clock generator even if the frequencies are same. If the AXI operates greater than 100 MHz, the ICAP clock must be fixed at 100 MHz.
- There can be cases when you access the ICAP as soon as JTAG programming is over. In such cases, the ICAP is not accessible as JTAG might be performing some other tasks. In such cases it is highly recommended that you allow sufficient time before initiating any transaction to ICAP.

Design Constraints

Timing Constraints on the Clocks

The core has two different clock domains: S_AXI_ACLK and ICAP_Clk. A timing ignore constraint is generated at run-time when the core is being used in the EDK system. For Vivado design tools, the equivalent XDC constraints are delivered when the core is generated.

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

Table 16 lists the AXI HWICAP resource utilization data for various parameter combinations measured with Zynq and 7 series devices.

Note: Resources numbers for Zynq-7000 (Artix-7 and Kintex-7 FPGA based) devices are expected to be similar to 7 series device numbers.

Table 16: **Performance and Resource Utilization Benchmarks on 7 Series FPGAs and Zynq-7000 Devices**

Parameter Values					Device Resources			Performance	
C_MODE	FIFO_TYPE	C_S_AXI_WIDTH	C_WRITE_FIFO_DEPTH	C_READ_FIFO_DEPTH	Slices	Slice Flip-Flops	LUTs	AXI f_{MAX} (MHz)	ICAP f_{MAX} (MHz)
0	0	32	128	128	336	704	706	160	80
0	0	32	256	128	365	720	840	160	80
0	0	32	512	128	458	737	1048	160	80
0	1	32	512	128	277	672	544	160	80
0	1	32	1024	128	291	688	538	160	80
1	1	32	256	128	276	587	544	160	80

Table 17 lists the AXI HWICAP resource utilization data for various parameter combinations measured with the Virtex-6 FPGA as the target device.

Table 17: **Performance and Resource Utilization Benchmarks on the Virtex-6 FPGA (xc6vlx240t)**

	Parameter Values				Device Resources			Performance	
C_MODE	FIFO_TYPE	C_S_AXI_WIDTH	C_WRITE_FIFO_DEPTH	C_READ_FIFO_DEPTH	Slices	Slice Flip-Flops	LUTs	AXI f_{MAX} (MHz)	ICAP f_{MAX} (MHz)
0	0	32	128	128	302	680	720	160	80
0	0	32	256	128	351	696	818	160	80
0	0	32	512	128	436	715	1037	160	80
0	1	32	512	128	254	651	545	160	80
0	1	32	1024	128	253	667	560	160	80
1	1	32	256	128	218	546	425	160	80

Table 18 lists the AXI HWICAP resource utilization data for various parameter combinations measured with the Spartan-6 FPGA as the target device.

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

	Parameter Values				Device Resources			Performance	
C_MODE	FIFO_TYPE	C_S_AXI_WIDTH	C_WRITE_FIFO_DEPTH	C_READ_FIFO_DEPTH	Slices	Slice Flip-Flops	LUTs	AXI f_{MAX} (MHz)	ICAP f_{MAX} (MHz)
0	0	32	128	128	217	597	569	100	20
0	0	32	256	128	233	618	633	100	20
0	0	32	512	128	272	632	769	100	20
0	1	32	512	128	212	594	495	100	20
0	1	32	1024	128	237	612	489	100	20
1	1	32	256	128	175	468	373	100	20

System Performance

To measure the system performance (F_{MAX}) of the AXI HWICAP core, it was added as the Device Under Test (DUT) to an FPGA system as shown in Figure 19.

Because the core is 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 design vary from the results reported here.

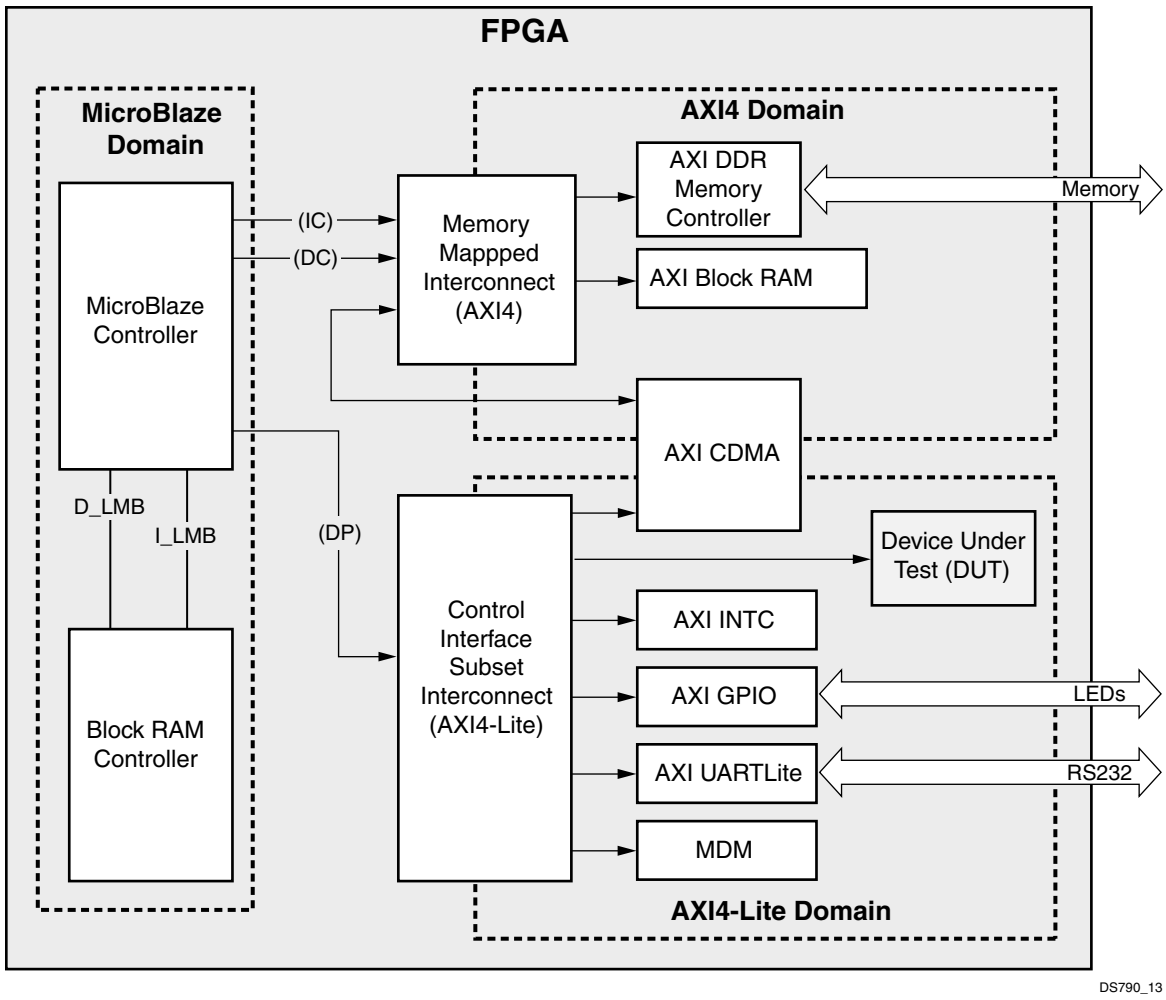

Figure 19: **FPGA System with the AXI HWICAP Core as the Device Under Test (DUT)**

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 19.

Table 19: **System Performance**

Target FPGA	Target F_{MAX} (MHz)		
	AXI4	AXI4-Lite	MicroBlaze
xc6slx45t (1)	90 MHz	120 MHz	80
xc6vlx240t (2)	135 MHz	180 MHz	135

Notes:

1. Spartan-6 FPGA LUT utilization: 60%; Block RAM utilization: 70%; I/O utilization: 80%; MicroBlaze™ Controller not AXI4 interconnect; AXI4 interconnect configured with a single clock of 120 MHz.
2. Virtex-6 FPGA LUT utilization: 70%; Block RAM utilization: 70%; I/O utilization: 80%.

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

To search for Xilinx documentation, go to www.xilinx.com/support.

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

1. AXI4 Advanced Microcontroller Bus Architecture (AMBA®) AXI Protocol Version: 2.0 Specification Advanced RISC Machine (ARM® IHI 0022C)
2. *LogiCORE IP AXI4-Lite IPIF (v1.01a) Data Sheet* (DS765)
3. *Spartan-6 Family Overview* (DS160)
4. *Virtex-6 Family Overview* (DS150)
5. *7 Series FPGAs Overview Advanced Product Specification* (DS180)
6. *Virtex-6 FPGA Configuration User Guide* (UG360)
7. *Spartan-6 FPGA Configuration User Guide* (UG380)
8. *7 Series FPGA Configuration User Guide* (UG470)

Support

Xilinx provides technical support for this LogiCORE IP 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*.

Licensing and Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx® Vivado Design Suite and Integrated Software Environment (ISE) Design Suite Embedded Edition software under the terms of the Xilinx End User License.

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, contact your local Xilinx sales representative.

Revision History

Date	Version	Revision
12/14/10	1.0	Initial release.
3/1/2011	1.1	Updated to v2.00a for the 13.1 release.
6/22/2011	1.2	Updated for the 13.2 release.
10/19/11	1.3	**Summary of 2.01.a Core Version Changes** • Updated for the 13.3 release Xilinx tools • Added Virtex-7 and Kintex-7 as supported devices **Summary of Major Documentation Changes** • Added Virtex-7 and Kintex-7 devices to IP Facts table • Added references to DS180 *7 Series FPGAs Overview Advanced Product Specification* in footnote 2 of IP Facts Table and Reference Documents section • Added List of Acronyms • Updated Notice of Disclaimer and copyright text • Removed Interrupt Descriptions section • Removed Timing Constraints on the ICAP Interface section • Updated the System Performance section.
04/24/12	1.4	**Summary of 2.02.a Major Changes** • Updated for the 14.1 release Xilinx tools • Added Zynq-7000 and 7 series support • Removed Spartan-6 FPGA Note on page 6 • Added two new parameters to the System Parameters section in Table 2, Design Parameters • Added one new register at the bottom of Table 4, Internal Registers • Updated Figure 6, Status Registers and corresponding bits in Table 9 • Renamed Abort section as Abort Status Register. Added new figure and table • Added Write Sequence for Lite Mode section • Added EOS_IN signal to Table 1, I/O Signals
07/25/12	1.5	• Updated for the 14.2/2012.2 release tools • Added Vivado design tools information. Added Artix-7 FPGA support.

Notice of Disclaimer