OPB HWICAP

DS 280 (v1.3) March 15, 2004 **Product Specification**

Introduction

This product specification describes the functionality of the HWICAP core for the On-Chip Peripheral Bus (OPB). The HWICAP module enables an embedded microprocessor (either 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 user's circuit structure and functionality during the circuit's operation.

Sample applications include:

- 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 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 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.

On-chip reconfiguration is accomplished using a read-modify-write mechanism. To modify a FPGA circuit, the peripheral determines the configuration frames that must be modified, and then reads each frame one at a time into a Block RAM. The contents of each frame is modified before being written back to the ICAP. The read-modify-write cycle is performed a single frame at a time. The HWICAP also provides for readback of configuration resource states. In this case, the frames are read back into the Block RAMs and the corresponding bits are extracted from the Block RAMs.

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex-II Pro™, Virtex-II™	
Version	opb_hwicap	v1.00a
Resources Used		
	Min	Max
Slices	120	128
LUTs	213	224
FFs	152	155
Block RAMs	1	1
Provided with Core		
Documentation	Product Specification	
Design File Formats	Verilog	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx® Implementation Tools	6.2i or later	
Verification	N/A	
Simulation	N/Ar	
Synthesis	XST	
Support		
Provided by Xilinx, Inc.		

Features

The initial version of the HWICAP includes support for resource reading and modification of the CLB LUTs and Flip-Flops. In addition, it has the capability to download partial bitstreams generated by the ISE tools—the partial bitstreams are transferred from local memory to the ICAP.

- OPB v2.1 bus interface
- Enables Read/Write of CLB LUTs
- Enables Read/Write of CLB Flip-Flop properties
- Enables downloading partial bitstream generated by the ISE tools
- ICAP interface operates at clock rate of OPB
- Support for MicroBlaze™ and PowerPC embedded processors
- Support for Virtex-II Pro™ and Virtex-II™ FPGA families

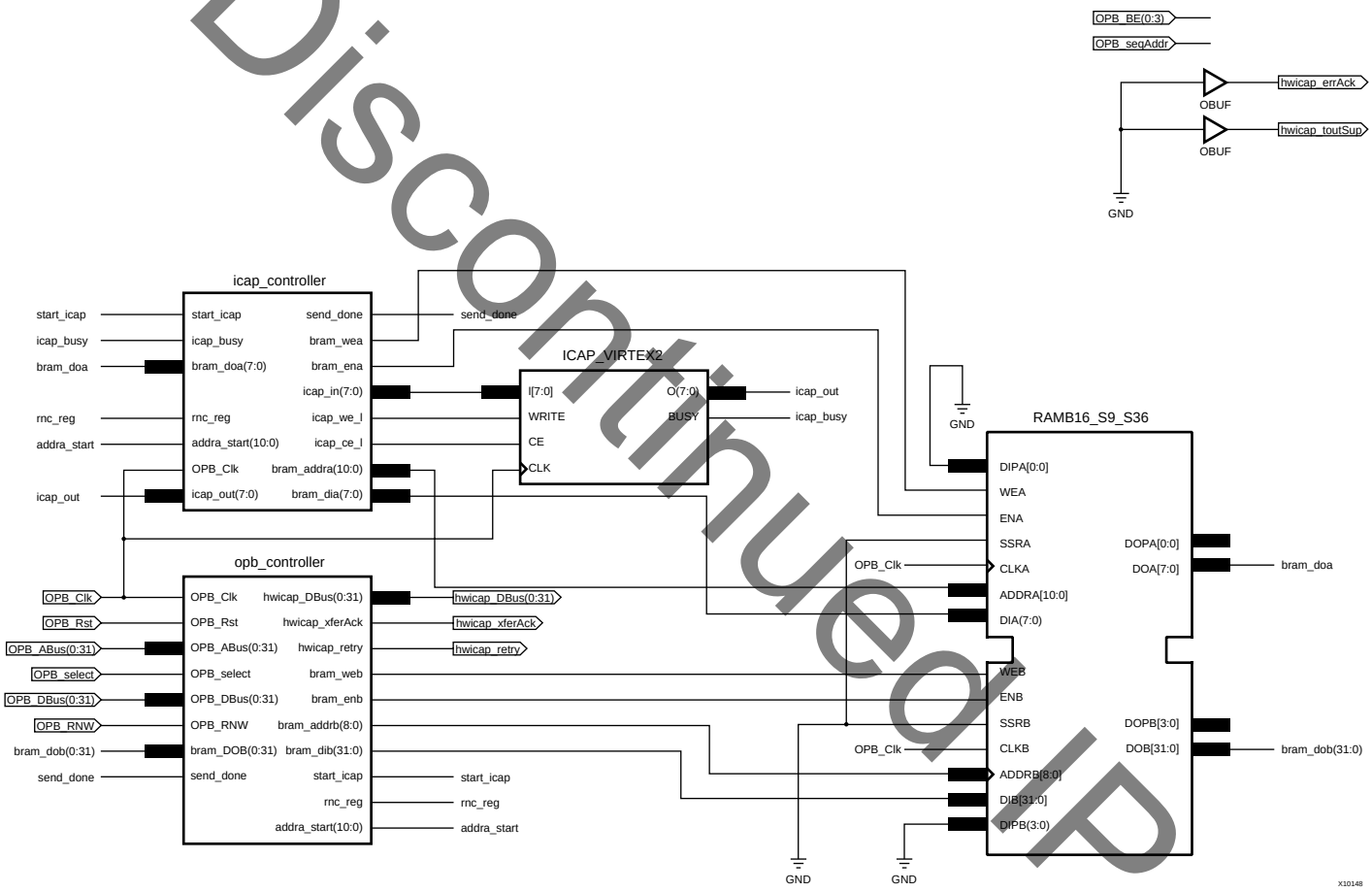

Figure 1: **Functional Block Diagram**

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 glitchless 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 readback or modification, then they will maintain their state.

If HWICAP is used to download partial bitstreams from the mainstream tools then there are issues with LUT RAM or LUT shift register placement or operation since the partial bitstreams configure full columns of the FPGA.

Important Note: The HWICAP core uses the ICAP found inside Virtex II™ and Virtex-II Pro™ 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, select another mode pin setting to avoid disabling the ICAP interface. JTAG configuration will still be available because it overrides other means of configuration, and the HWICAP core will function as intended.

Besides being disabled with the Boundary Scan mode pin setting, the ICAP will be disabled if the persist bit in the device configuration logic’s control register is set. When using bitgen one must make sure that the Persist option is 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.

HWICAP Parameters

The HWICAP is a fixed IP block. It does not have parameters to modify its implementation. The only modifiable parameters are the address range for the OPB bus. To specify the address range use the C_BASEADDR and C_HIGHADDR parameters. Note that the lower 12 bits of C_BASEADDR must be 0 and that the valid range is 0x1000. For example, C_BASEADDR = 0xFFFE2000, C_HIGHADDR = 0xFFFE2FFF.

HWICAP I/O Signals

The I/O signals for the HWICAP are listed in Table 1.

HWICAP Address Map and Register Descriptions

Table 1: **HWICAP I/O Signals**

Signal Name	Interface	I/O	Description	Status
OPB_Clk	OPB	I	OPB Clock	Used
OPB_Rst	OPB	I	OPB Reset	Used
OPB_ABus[0:31]	OPB	I	OPB Address Bus	Used
OPB_BE[0:3]	OPB	I	OPB Byte Enables	Not Used
OPB_DBus[0:31]	OPB	I	OPB Data Bus	Used
OPB_RNW	OPB	I	OPB Read, Not Write	Used
OPB_select	OPB	I	OPB Select	Used
OPB_seqAddr	OPB	I	OPB Sequential Address	Not Used
hwicap_DBus[0:31]	OPB	O	HWICAP OPB Data Bus	Used
hwicap_xferAck	OPB	O	HWICAP OPB Transfer Acknowledge	Used
hwicap_errAck	OPB	O	HWICAP OPB Error Acknowledge	Not Used
hwicap_retry	OPB	O	HWICAP OPB retry	Used
hwicap_toutSup	OPB	O	HWICAP Timeout Suppress	Not Used

Register Data Types and Organization

Registers in the HWICAP are accessed as a word (4 bytes). All register accesses conform to the OPB-IPIF register location convention. The addresses of the HWICAP registers are provided in the Address Map section.

The HWICAP registers are organized as big-endian data. The bit and byte labeling for the big-endian data types is shown in Figure 2.

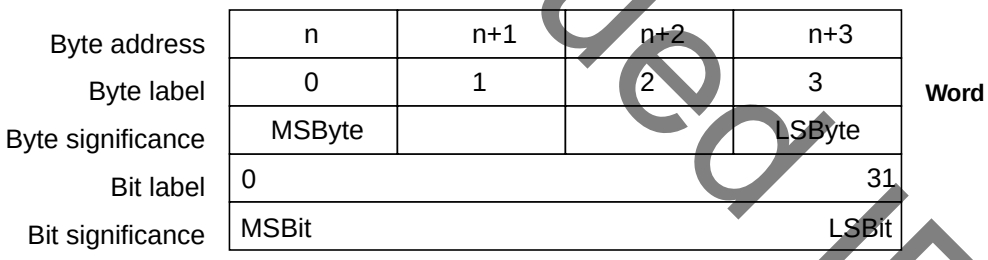

Figure 2: **Big-Endian Data Types**

Registers of the HWICAP

Information on the following registers and memories used in assembly language programming are described in this section.

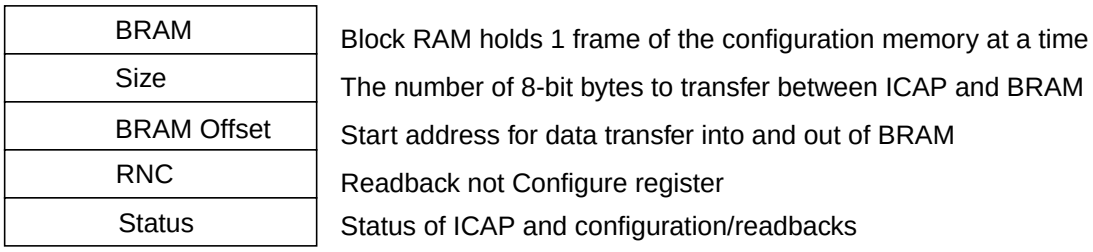

Figure 3: **HWICAP Register Set**

The HWICAP internal register set is described in Table 2.

Table 2: **HWICAP Registers**

Bit Size	Name	Description
16K	BRAM	Block RAM Used as a configuration memory cache. One frame of data is read back from the ICAP port and stored here where the processor modifies the data. After modification, the HWICAP downloads the data back to the configuration memory through the ICAP port.
32	Size Register	Indicates the number of 8-bit bytes to transfer between BRAM and ICAP (or ICAP to BRAM).
32	BRAM Offset Register	Indicates BRAM starting address location to transfer data between BRAM and ICAP (or ICAP to BRAM).
1	RNC Register	Readback not Configure register. The RNC register determines the direction of the data transfer. It controls whether a configuration or readback takes place. Writing to this register initiates the transfer. A value of 1 initiates a readback while writing a value of 0 initiates a configuration.
1	Status Register	The status register contains the ICAP status and the done bit. The done bit is set to zero during configuration or readback. When the current configuration or readback completes, the done bit is set to one.

As shown in Table 3, the status register contains the ICAP status bits and the done bit.

Table 3: **Status Register**

Bit Location	Name	Access	Reset Value	Description
31-9	Reserved	Read	0	Reserved bits
8	cfgerr_n	Read	1	Configuration error
7	dalign	Read	0	Data alignment, found syncword
6	rip	Read	0	Readback in progress
5	in_abort_n	Read	1	Super8 (SelectMAP) abort
4	Always 1	Read	1	Always 1
3	Always 1	Read	1	Always 1
2	Always 1	Read	1	Always 1
1	Always 1	Read	1	Always 1
0	Done	Read	1	HWICAP done with configuration or readback

Address Map

HWICAP BASE ADDRESS + 0h: BRAM Space. Access: Read and Write.

HWICPA BASE ADDRESS + 800h: Size Register. Access: Read and Write.

HWICAP BASE ADDRESS + 804h: Offset Register. Access: Read and Write.

HWICAP BASE ADDRESS + 808h: Read not Configure (RNC) register. Access: Write only.

HWICAP BASE ADDRESS + 80Ch: Status Register. Access: Read only.

Interrupts

HWICAP does not generate any interrupts.

Software Support

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

Core History

Table 4 shows the revisions of this core. The 6.2.1i drivers have been updated to treat done as bit 0 rather than the entire status register.

Table 4: **Core Revisions**

EDK Version	Description
6.2i	Initial release
6.2.1i	Fixed 00-bug. Changed done to status register.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
1/13/04	1.0	Initial release.
1/23/04	1.1	Added block diagram and LogiCORE™ logo.
1/30/04	1.2	Added warning about JTAG mode. Minor corrections.
3/15/04	1.3	Updated for EDK6.2.1. Done changed to Status register.