



**HUNT ENGINEERING**  
Chestnut Court, Burton Row,  
Brent Knoll, Somerset, TA9 4BP, UK  
Tel: (+44) (0)1278 760188,  
Fax: (+44) (0)1278 760199,  
Email: sales@hunteng.co.uk  
www.hunteng.co.uk  
www.hunt-dsp.com



## The HERON FPGA9 Example3

Rev 1.1 R. Williams 29-04-05

The HERON-FPGA9 is a module that has a Virtex-II Pro Xilinx FPGA and 256Mbytes of DDR SDRAM memory. The Virtex-II Pro FPGA includes a Power PC processor core.

Users can use the HERON-FPGA9 to provide a custom storage capability using the DDR SDRAM to store large amounts of data. The module may also use the FPGA and Power PC processor core to process the data stored in the DDR SDRAM.

With 256Mbytes of DDR memory available the HERON-FPGA9 is ideally suited for creating a large FIFO, with the memory interface of the DDR SDRAM providing the storage for the FIFO.

The DDR memory of the HERON-FPGA9 is split into two parts, each 128Mbytes in size. Each part of the memory is used to create an individual FIFO. The result is two 128Mbyte FIFOs interfaced to two separate pairs of HERON input and output FIFOs.

As such Example3 is highly suitable for applications that require buffering between high speed data sources like A/D converters of IO modules, and slower non-real-time host PC interfaces such as PCI.

### History

Example revision 1.0	26-08-04	Developed from Example3 for the HERON-FPGA7
Example revision 1.1	29-04-05	Removed reference to specific ISE versions

## **What the Bit-stream Does**

Example3 (SDRAM FIFO) for the HERON-FPGA9 is supplied on the HUNT ENGINEERING CD, and web site. The FPGA source code is supplied along with a bit-stream that can be loaded directly onto the HERON-FPGA9.

If you make changes to the project and re-build it you can change the functionality to be whatever you want, but if you use the supplied bit-stream you need to know what it is doing. This document describes that for you.

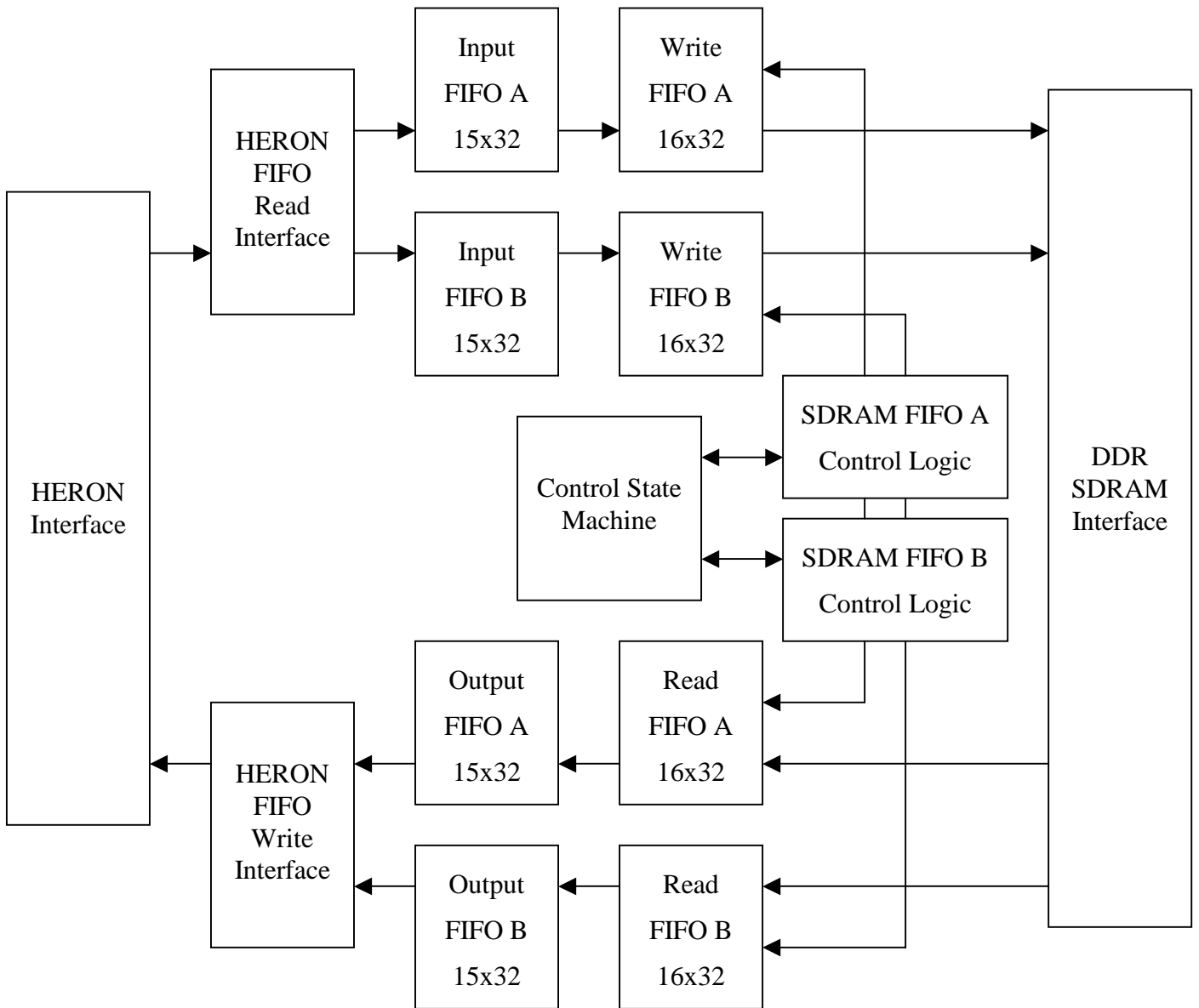
The HERON-FPGA9 is fitted with a 50MHz oscillator which is used to generate a 200MHz clock source for the FPGA. The 200MHz clock source is used to generate all DDR SDRAM clock signals. This is automatically handled by the Hardware Interface Layer.

The HERON-FPGA9 is also fitted with a 100MHz oscillator on OSC3. This clock is used to drive the FIFO clocks directly, or to drive the FIFO clocks after being divided by 2.

For a HEPC8 the HERON FIFO clocks must be driven at less than 60Mhz, whereas for the HEPC9 they must be driven at between 60Mhz and 100Mhz. For this reason there are two different bit-streams supplied for the HEPC8 and HEPC9 as follows.

2vp7ff672.hcb	HERON-FPGA9 fitted to HEART based carrier e.g HEPC9 (100Mhz FIFO clocks)
2vp7ff672_pc8.hcb	HERON-FPGA9 fitted to HEPC8 (50Mhz FIFO clocks)

**FUNCTIONAL BLOCK DIAGRAM**



## **SDRAM FIFO Operation**

The HERON-FPGA9 provides two 128Mbyte banks of DDR SDRAM memory. This memory is organised as two 32-bit wide memory interfaces, each with 32Mlocations. The SDRAM FIFO example uses both banks to create two independent FIFOs of 32Mlocations by 32-bits. The first FIFO is SDRAM FIFO A, and the second FIFO is SDRAM FIFO B.

All data from HERON Input FIFO 0 is passed to SDRAM FIFO A. Data output from SDRAM FIFO A is passed to HERON Output FIFO 0. Similarly, all data from HERON Input FIFO 1 is passed to FIFO B, and data output from that FIFO is passed to HERON Output FIFO 1.

Between each HERON input FIFO and SDRAM FIFO are two small FIFOs. This first small FIFO is a 15x32 asynchronous FIFO generated using the Core Generator.

The second small FIFO is a synchronous FIFO that directly interfaces to the DDR interface of the USER\_AP entity.

Between each SDRAM FIFO and HERON output FIFO are again, two more small FIFOs. The first is a synchronous FIFO to directly interface to the DDR memory, and the second is the CoreGen 15x32 FIFO.

With four small FIFOs associated with each SDRAM FIFO there is a total FIFO depth from HERON input FIFO to HERON output FIFO of 33554494 words.

Four LEDs are used to represent the state of both SDRAM FIFOs, as follows:

- LED 0 is illuminated when SDRAM FIFO A is empty
- LED 1 is illuminated when SDRAM FIFO A is full
- LED 2 is illuminated when SDRAM FIFO B is empty
- LED 3 is illuminated when SDRAM FIFO B is full

For each LED there is a small counter that ensures the LED is illuminated long enough to be seen, even if the full or empty condition lasts for only a few clock cycles.

LED 4 will always flash to indicate that the system FIFO clock FCLK\_G is running.

## **Typical Use of the Example Bit-stream**

The typical use of the example bit-stream is to provide two independent FIFOs, each over 32Mlocations in depth, and 32-bits wide. The first SDRAM FIFO reads and writes from HERON FIFO 0, and the second SDRAM FIFO reads and writes from HERON FIFO 1.

Therefore, the HERON-FPGA9 can be placed in a system with other modules and by correctly connecting FIFOs from surrounding modules to FIFO 0 and FIFO 1 of the HERON-FPGA9, two large data buffers can be inserted in the flow of data through the system.

## **Where are the Bit-streams and Example Program?**

The bit streams for this example can be found on the HUNT ENGINEERING CD in the directory `\fpga\fpga9v1\Sdram_Fifo(ex3)`. The name of the bitstream reflects the Xilinx FPGA part number and the Carrier board type, as explained earlier in this document.

An easier way to navigate to the correct directory is to select the “Files” link next to the “SDRAM FIFO” link under the IP sections of the CD browser.

The source files for the FPGA example can be found in the ‘`src`’ sub-directory. The sources in the ‘`\fpga9v1\common`’ directory are also required. A project for the Xilinx ISE development tools is provided for you in the ‘`ISE`’ sub-directory.

## FPGA Example Code

You should understand the HUNT ENGINEERING VHDL support for HERON modules before looking at this section. If you do not then please review Example1 again (the 'Getting Started' example for FPGA modules).

As you are expected to be already familiar with Example1, this section only discusses the points that are unique to example3.

In the USER-AP entity for Example3, contained in the file 'user\_ap3.vhd', there are options that allow you to select the FIFO clock frequency. Just as with Example1, you can select if the 100Mhz input clock is divided by 2 to provide the FIFO clock frequency. This allows example2 to be able to clock the FIFOs at 50Mhz (suitable for HEPC8) or 100Mhz (suitable for HEPC9).

You must also remember to set the HIGH\_FCLK\_G option to show if this clock is higher or lower than 60Mhz.

For example2 the correct options for an HEPC8 are:

DIV2_FCLK	FCLK_G_DOMAIN	HIGH_FCLK_G	HIGH_FCLK_RD	HIGH_FCLK_WR
True	True	False	n/a	n/a

This will set both input and output FIFOs to be clocked at 50Mhz.

For example2 the correct options for an HEPC9 are :-

DIV2_FCLK	FCLK_G_DOMAIN	HIGH_FCLK_G	HIGH_FCLK_RD	HIGH_FCLK_WR
False	True	True	n/a	n/a

This will set both input and output FIFOs to be clocked at 100Mhz.

You need to consider the timing constraints that are defined in the '.ucf' file for your design. Actually if you use a time specification that is more strict than needed there is no problem, so the standard '.ucf' file have the FIFO clocks specified as 100Mhz, along with the SDRAM clocks defined as 200MHz. If the project builds (as Example3 does) with this specification it is still guaranteed to work at lower clock speeds. If you add new clock nets into your design then you need to add new timing constraints into your design.