



**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 1.2 R. Williams 02-08-05

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.

The DDR memory interface of the HERON-FPGA9 enables high speed data storage, with access speeds of 1.6Gbytes/sec possible per bank between the DDR memory and FPGA. This data rate is demonstrated by the example.

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
Example revision 1.2	02-08-05	Increased the memory access rate of the example

## **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.

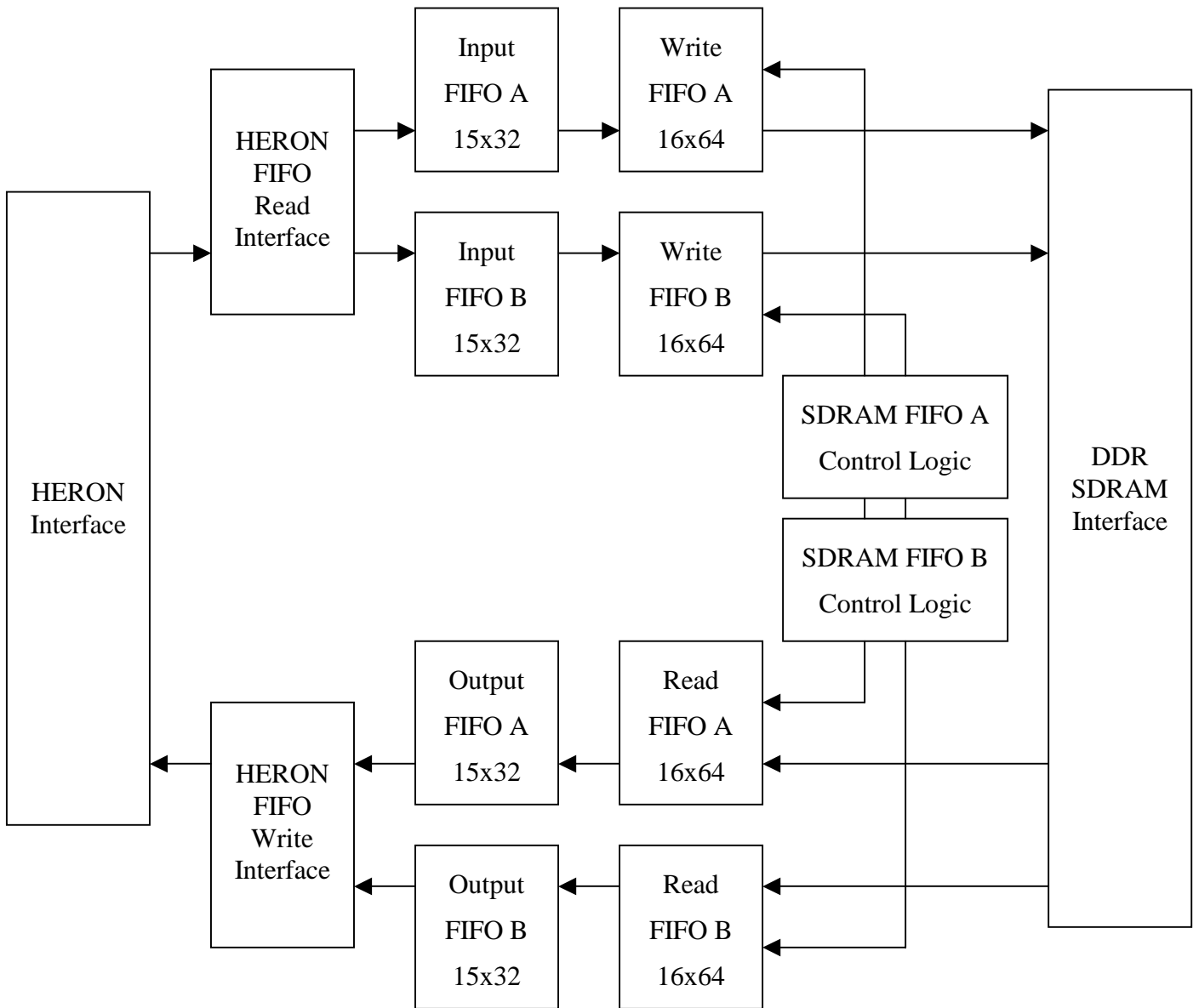
The HERON-FPGA9 is fitted with a 50MHz oscillator which is used to generate a 200MHz clock source for the FPGA. The 200MHz clock source is used to generate all DDR SDRAM clock signals. This is automatically handled by the Hardware Interface Layer.

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 three small FIFOs. This first small FIFO is a 15x32 asynchronous FIFO generated using the Core Generator.

The next two FIFOs are synchronous FIFOs that directly interface to the DDR interface of the USER\_AP entity.

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

With six small FIFOs associated with each SDRAM FIFO there is a total FIFO depth from HERON input FIFO to HERON output FIFO of 33554526 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.

## **Accessing Memory at High Speed**

The HERON-FPGA9 is fitted with 256Mbytes of DDR SDRAM memory. This memory is based on SDRAM technology, and as such, memory is read or written by first opening a row which contains the addressed cells. When the row has been opened the data is then read or written. At the end of the data transfer for that row, the row must then be closed.

It is important when accessing SDRAM memory that many words are transferred while a row is open. This is to ensure that the overhead of the row open and row close operations does not out-weigh the time taken in transferring data.

Conventional SDRAM memory transfers data in bursts of many words where one word of data is transferred on every clock cycle of the memory. 'Double Data Rate' DDR memory doubles this data rate by being able to transfer one word on the rising edge and one word on the falling edge of the memory clock signal.

On the HERON-FPGA9 the user logic inside the FPGA has access to DDR memory via the HE\_DDR component provided in the Hardware Interface Layer. The HE\_DDR component is designed to access the external DDR memory as efficiently as possible. As such it presents FIFO interfaces for read data and write data that allow bursts to be performed to memory. The FIFO interfaces are organised as rising data FIFOs and falling data FIFOs which makes it possible to burst at high data rates to and from memory.

For memory write operations, the HE\_DDR component presents three separate FIFO interfaces. One FIFO for storing the write addresses, one FIFO for storing write rising data and one FIFO for storing write falling data. Similarly for memory read operations the HE\_DDR component presents three more FIFO interfaces. The first is used to store the read addresses, the second for storing read rising data and the third for storing read falling data.

The FIFO components implemented inside the HE\_DDR component are asynchronous. On the memory side of the HE\_DDR FIFOs all accesses are performed from a 200MHz clock signal. On the user side of the HE\_DDR FIFOs all accesses are performed at a separate clock rate created by the user.

When accessing memory, the HE\_DDR component always bursts at 1.6Gbytes/sec. The HE\_DDR component transfers in bursts of four words, two words of rising data and two words of falling data.

The data burst is performed at 200MHz with data transferred on both rising edge and falling edge of the clock. This means that 8 bytes are transferred on every clock cycle, which is a data rate of 1.6Gbytes/sec.

The data rate on the user logic side of the interface FIFOs is set by two factors. The first factor is the clock rate at which the interface FIFOs are accessed and the second factor is whether rising data and falling data FIFOs are accessed sequentially or concurrently.

To match the performance of the user logic to the DDR memory, the user clock rate should be set to 200MHz, and the rising data and falling data FIFOs should be accessed concurrently. In transferring to or from the rising data FIFOs and falling data FIFOs in the same clock cycle it becomes possible to transfer 8 bytes every cycle. If this process is performed at a clock rate of 200MHz, the user logic will achieve a data rate of 1.6Gbytes/sec.

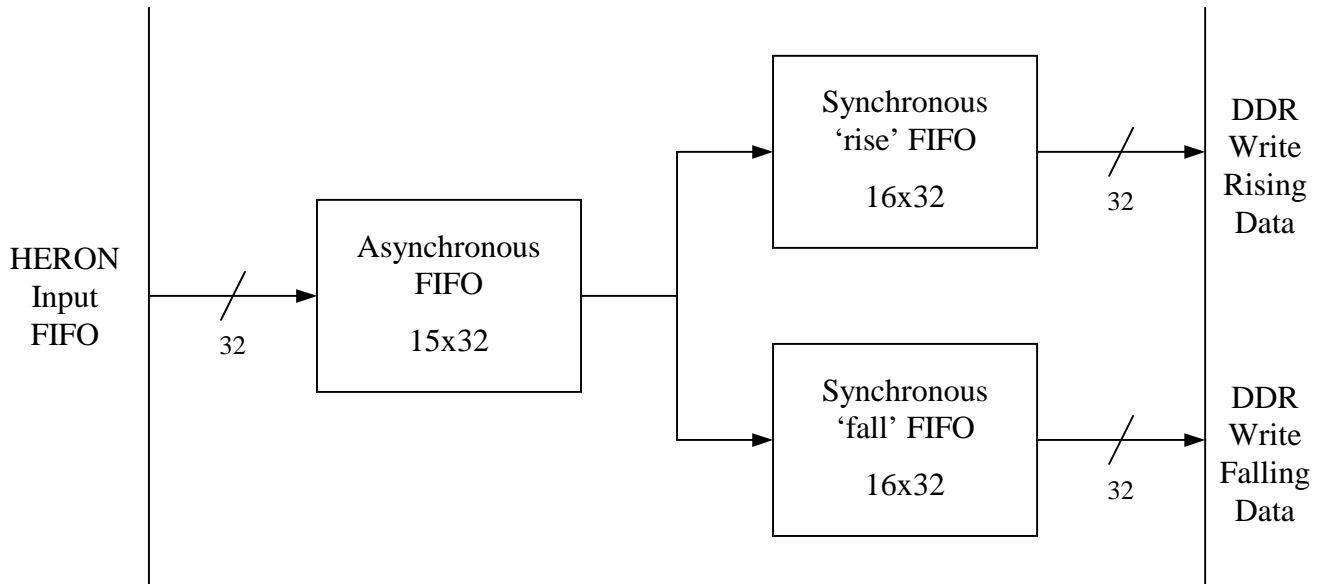
In Example3 this is how the HE\_DDR interface FIFOs are accessed. In addition to providing two large 128Mbyte FIFOs Example3 therefore also provides as an example of how to access the DDR memory at the full rate of 1.6Gbytes/sec.

## Writing to Memory at 1.6Gbytes/sec

In Example3, two independent 128Mbyte FIFOs are created by using the two banks of DDR memory fitted to the HERON-FPGA9. The write process to each DDR FIFO is identical for each of the two banks.

For 'FIFO A', data arrives via HERON Input FIFO 0 and for 'FIFO B' data arrives via HERON Input FIFO 1.

The write process is shown in the diagram below.



Data is first transferred from the HERON Input FIFO to the asynchronous FIFO. This transfer is done in the FCLK\_G domain at 100MHz. One word of data can be transferred on every cycle and therefore the data rate at this stage is 400Mbytes/sec.

Data arriving in the asynchronous FIFO is then transferred alternately to first the 'rise' FIFO and then the 'fall' FIFO. The read side of the asynchronous FIFO and write side of the synchronous FIFOs are all clocked at 200MHz. One word can be read from the asynchronous FIFO on every cycle and therefore the data rate at this stage is 800Mbytes/sec.

Finally, data in the synchronous FIFOs is transferred to the write FIFO interfaces of the HE\_DDR component. The logic at this stage is written such that two words can be transferred on every cycle at 200MHz. The data rate at this stage is 1.6Gbytes/sec. In this stage, counters are used to generate the write address values. Logic is used in combination with the read process to keep track of the amount of data stored in the DDR FIFO.

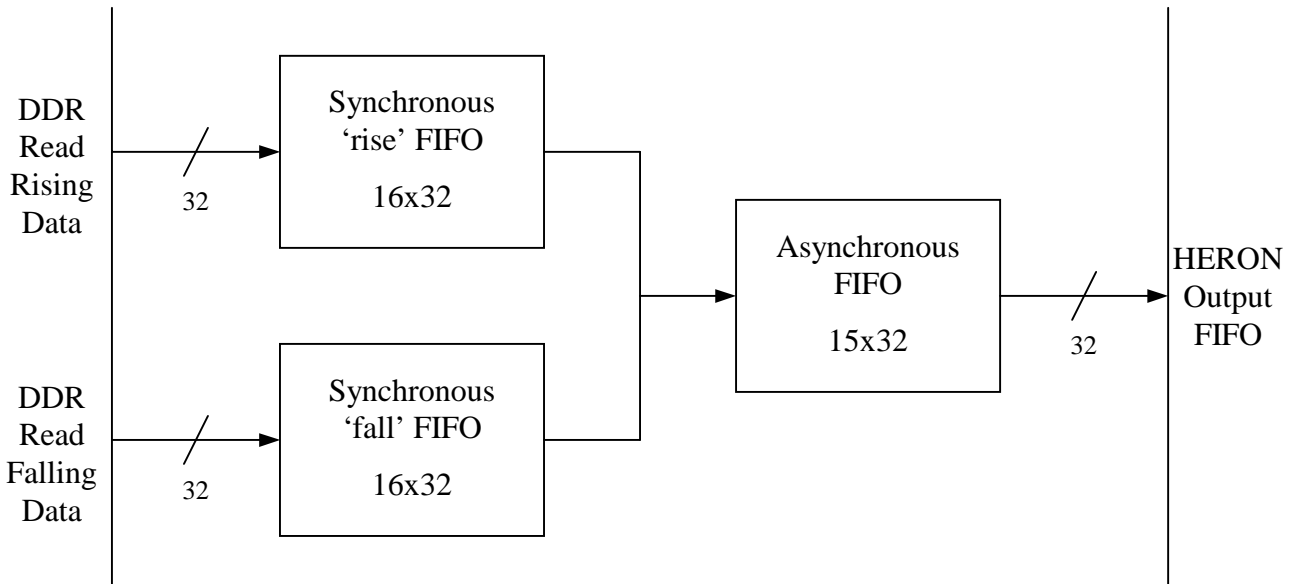


## Reading from Memory at 1.6Gbytes/sec

In Example3, two independent 128Mbyte FIFOs are created by using the two banks of DDR memory fitted to the HERON-FPGA9. The read process from each DDR FIFO is identical for each of the two banks.

For 'FIFO A', data is output via HERON Output FIFO 0 and for 'FIFO B' data is output on HERON Output FIFO 1.

The read process is shown in the diagram below.



Data is first transferred from the HE\_DDR component to the synchronous FIFOs. The logic in this stage is written such that two words can be transferred on every cycle at 200MHz. The data rate at this stage is 1.6Gbytes/sec. In this stage, counters are used to generate the read address values. Logic is used in combination with the write process to keep track of the amount of data stored in the DDR FIFO.

Data arriving in the synchronous FIFOs is then transferred in alternate cycles from first the 'rise' FIFO and then the 'fall' FIFO to the asynchronous FIFO. The read side of the synchronous FIFOs and write side of the asynchronous FIFO are all clocked at 200MHz. One word can be written to the asynchronous FIFO on every cycle and therefore the data rate at this stage is 800Mbytes/sec.

Finally data is transferred from the asynchronous FIFO to the HERON Output FIFO. This transfer is done in the FCLK\_G domain at 100MHz. One word of data can be transferred on every cycle and therefore the data rate at this stage is 400Mbytes/sec.

## **Example3 Performance**

The HERON-FPGA9 is fitted with two identical independent banks of DDR memory, each 128Mbytes in size. For one bank of DDR memory, data can be transferred at the rate of 1.6Gbytes/sec. Each bank can transfer data concurrently with data transfer on the other bank.

For one bank of memory data can either be read at 1.6Gbytes/sec or data can be written at 1.6Gbytes/sec. It is not possible to both read and write at the same time.

The HERON Input FIFO interface of the HERON-FPGA9 can transfer data at 400Mbytes/sec. This bandwidth must be shared among the six input FIFOs. In Example3 two HERON Input FIFOs are used to transfer data to two DDR FIFOs. If data is arriving concurrently on both FIFOs then the 400Mbytes/sec data rate must drop as it becomes shared between FIFOs.

Similarly, the HERON Output FIFO interface can transfer data at 400Mbytes/sec, but this must also be shared among the six output FIFOs. For Example3 this again means that the 400Mbytes/sec bandwidth must be shared between two output FIFO processes.

As the DDR memory is directly interfaced to HERON FIFOs the sustained data rate through one DDR FIFO can not be greater than 400Mbytes/sec. The HE\_DDR interface however is capable of 1.6Gbytes/sec.

In Example3 logic is provided that demonstrates a 1.6Gbyte/sec connection to memory within the user logic section of the project, but the full bandwidth is only used between the Hardware Interface Layer (HE\_DDR component) and the synchronous 16x32 FIFO components.

Of course, if the HERON-FIFO connection was not directly used, internal FPGA logic could be developed that made greater use of the available bandwidth.