In many applications it is important to process data as quickly as possible. Faster algorithm speed can push into new applications spaces, offering customers capabilities not previously possible. Faster processing can also be traded off for lower cost and lower power if more can be done per unit of time, thus reducing clock rates or bus width. When algorithm acceleration can be done generically, and not tied to a single specific algorithm, systems can adapt to changing requirements, offering even more advantages. These generic application accelerators can be done efficiently with FPGAs if the data can be routed into and out of the FPGA efficiently and if FPGA configuration can be handled quickly and easily.

In this application note we will look at what it takes to create an algorithm acceleration subsystem using a Gennum PCI Express Local Bus Bridge and an IDT PCI Express switch in conjunction with a Xilinx FPGA. Also included in the design will be high-performance DDR3 memories from Micron. The memory components will provide the data storage needed to support the FPGA-based algorithm while the PCI Express interface will provide the data transport required for high speed algorithm data as well as configuration information used by the FPGA to ‘define’ the algorithm. The PCI Express Local Bus Bridge is used so that the FPGA resources can be dedicated to algorithm support as well as simplifying the configuration, power on reset and other initialization requirements that are best done without algorithm user intervention.

Topics in this application note are organized by design task (Device Selection, FPGA Design, and Board Design) and each topic is a stand-alone section, with a short introduction or overview, followed by the step-by-step design guidelines. The reader may skip over sections and go directly to the topic of interest, or start at the beginning and follow each topic in order to quickly go through the entire design process, step-by-step. The topics covered in this application note include the following:

1) Device Selection
   a. Gennum PCI Express Local Bus Bridge
      i. Functional Overview
      ii. Device Selection
   b. Micron DDR3 Memory
      i. Functional Overview
      ii. Device Selection
   c. IDT PCI Express Switch
      i. Functional Overview
      ii. Device Selection
   d. Xilinx Virtex-5 FPGAs
      i. Virtex-5 Functional Overview
      ii. Device Selection

2) FPGA Design
   a. Using the IP Cores
      i. IP Core Overview
      ii. Functional Description
      iii. Pin Descriptions
      iv. Example Design- Step-by-Step
   b. Integration of the IP Core
      i. Integration Overview

Device Selection

The first phase of the design project is device selection, where the designer decides on the key components to be used in the design. In our algorithm acceleration application we will assume we have already selected Gennum, Xilinx, Micron and IDT as the suppliers for the various components.

A block diagram of our target application is given in Figure 1 below. The Host Processor communicates with the Algorithm Acceleration hardware via the PCI Express interface. Multiple target FPGA-based algorithm acceleration targets can be supported by the use of the PCI Express switch. The switch and acceleration target can be considered a building block, which can be replicated to the right of the diagram as many times as required. A complete acceleration ‘farm’ with multiple copies of the acceleration hardware could be implemented. For very compute intensive applications like dynamic fluid flow, electromagnetic modeling and chemical-molecular simulations it would be possible to have several ranks of algorithm accelerators all working in parallel to achieve truly massive increases in throughput.

Figure 1: Block Diagram of Algorithm Acceleration Application
In the rest of this section of the application note we will review the Gennum PCI Express Bridge offering. Micron DDR3 device Family, the IDT PCI Express Switch and the Xilinx FPGA offerings. This combination of devices is excellent for implementing any high-performance compute intensive design. The following sections provide a quick overview of each device family, identifies key features that make them good selections for our target application and determines the best device within the family for our specific application.

Gennum PCI Express Local Bus Bridge
Gennum offers two PCI Express Local Bus bridge devices. The GN4124 is a four lane PCI Express-to-Local bus endpoint bridge and the GN4121 is a Single Lane version. Since we want to support the highest bandwidth as possible it will be important to use the four lane version.

Gennum GN4124 Product Description
The GN4124 is a four lane PCI Express to local bus bridge that is designed to work as a companion for low-cost FPGA devices to provide a complete bridging solution for general applications. In addition to a 4-lane PCI Express compliant PHY interface, the GN4124 also contains the data link and transaction layers, and an applications interface that is ideally suited to FPGA interfacing using a small number of pins.

Since the PCI Express transaction/link IP is hard-wired into the GN4124, there is no need to license PCIe IP. The level of integration and very low power operation of the GN4124 make it an ideal alternative to using a PIPE PHY, where IP licensing and the cost of FPGA resources and power consumption is unattractive by comparison. Using the GN4124 allows FPGA resources to be spent on what differentiates the product rather than on implementing the PCI Express protocol.

Since the GN4124 contains a complete type 0 PCI configuration space, it is live on power-up so that a plug-and-play BIOS can auto-detect it and enumerate it without an attached FPGA having to be configured.

An FPGA configuration bitstream may be downloaded from the host system over PCIe to the attached FPGA using the on-chip FPGA Configuration Loader (FCL). This eliminates the expense of a dedicated FPGA ROM and makes on-the-fly reconfiguration and firmware upgrades simple. Any application requiring dynamic reconfiguration or firmware upgrades over PCIe can benefit from the use of the GN4124 as a companion device irrespective of the size or type of FPGA device used. In addition, the use of on-the-fly configuration is a significant time-saver during the design development cycle. Because the development system does not need to be rebooted when the FPGA is updated there is a significant reduction in system downtime. The wait between each iteration is significantly reduced and productivity is thus very high.

Local Bus Interface
The local bus interface uses a combination of single and dual data rate SSTL I/O to accomplish very high data rates using the fewest possible pins. A single data rate clock is used for SSTL control signals and separate dual data rate source synchronous clocking is used for the DDR SSTL data. The SDR control signals operate at up to 200MHz and the DDR I/O operate at up to 400MT/s across 16 bits using a 200MHz DDR clock. This provides 800MB/s in each direction. Optional on-chip termination may be enabled to enhance signal integrity and simplify board design.

The local bus may operate asynchronously from the PCI Express rate. In order to save power, the local bus clock can operate at the lowest possible rate required by an application. The local bus protocol facilitates 4 types of transactions:

- PCIe-to-Local Target Writes: a PCIe agent (such as the host processor/root complex) writes data to the local bus.
- PCIe-to-Local Target Reads: a PCIe agent reads data from the local bus. Reads are split into a request phase (address phase) and a completion phase (data phase)
- Local-to-PCIe Master Writes: the attached FPGA writes data to a PCIe device (such as host memory).
- Local-to-PCIe Master Reads: the attached FPGA reads data from a PCIe device.

The PCIe-to-Local transactions would typical involve a target controller implemented in the FPGA. Local-to-PCIe Master transactions allow a DMA controller in the FPGA to access PCI Express devices.

PCI Express Application Layer
The on-chip applications layer transfers data between the PCI Express port and an attached FPGA using the local bus interface. It also provides a mechanism to access internal registers through configuration space access and through one of the Base Address Registers (BAR4). The applications layer also supports the transmission of message signaled interrupts.

Interrupt Controller
A flexible interrupt controller automatically generates PCIe message signaled interrupts from either external pins (GPIO pins) or internally generated interrupt sources. In addition, the interrupt controller can route any interrupt source to up to 4 GPIO pins.

2-Wire Serial Controller
An on-chip I2C compatible “2-wire” controller provides both a master and target mode. After device reset “boot master mode” is optionally initiated so that default configuration register values, such as Subsystem Vendor ID and BAR sizes, can be automatically loaded from a small serial EEPROM. After initialization, target mode allows an external 2-wire master to access on-chip registers. The 2-wire master mode controller in the GN4124 can be accessed over PCI Express to access external 2-wire target devices.

General Purpose IO
Sixteen General Purpose IO pins are available for user applications. They may be used as inputs or outputs. A subset may be used for interrupt inputs.

In order to simplify design, Gennum provides FPGA IP for interfacing to the Local Bus of the GN4124. This IP will be described in more detail in the FPGA Design section of this application note.

The local bus FPGA IP is licensed, and royalty free for use with the GN4124. It uses a layered approach so that part or all of the IP may be used. The layers are:

- Physical Attachment Layer (PAL): provides the interface to the GN4124 taking the high-speed, narrow external buses and widening them out for use inside the FPGA
- Application Attachment Layer (AAL): provides a target interface
and the FlexDMA master mode DMA block.

- User Application Layer (UAL): the local bus core ships with a benchmarking application layer. However, this is easily replaced by the users application.

We will use the Gemmm GN4124 as the connection from the PCI Express switch to our Xilinx FPGA. It provides the number of ports required, the number of lanes and simplifies connection to the FPGA by using a standard high-speed interface and a supplied FPGA IP Core.

**Micron DDR3 Memory Solutions**

Micron offers an exceptionally wide range of DDR3 memory options that include memory modules and components. Memory modules offer a significant increase in memory capacity- both width and depth- since typically 8 or more devices are mounted on a single module. For high capacity requirements a module usually is a significant savings in terms of board-space as well. Because module form factor and functionality are standard it is easy to replace a module with one of higher or lower capacity as well. This makes it easy to create a common platform for a design that can be used in both high-end and low-end applications by increasing memory size and speed.

Micro Technology has DDR3 Memory Modules from 512MB to 4GB and from 800MT/s to 1333MT/s. Table 1 below shows a summary of some of the Micron Technology products.

For our target system we need multiple memory interfaces in order to support the high data traffic requirement of the acceleration FPGA. Typically we will have a data source memory, data destination memory and a memory to support data traffic to and from the PCI Express interface. These memories should be identical in speed and size so that they can be swapped back and forth depending on the application requirements.

If each memory block has a capacity requirement of 4Gb, we could use 2 of the 2Gb x8 devices in each sub-system to achieve our capacity requirement. If we put the devices in parallel we can get significant data throughput by operating them at the 1066MT/s rate. Out target device is then the MT41J256M4HX-187E.

**Micron MT41J Functional Description**

The DDR3 SDRAM uses a double data rate architecture to achieve high-speed operation. The double data rate architecture is an 8n-pre-fetch architecture with an interface designed to transfer two data words per clock cycle at the I/O pins. A single read or write access for the DDR3 SDRAM consists of a single 8n-bit-wide, one-clock-cycle data transfer at the internal DRAM core and eight corresponding n-bit-wide, one-half-clock cycle data transfers at the I/O pins.

The differential data strobe (DQS, DQS#) is transmitted externally, along with data, for use in data capture at the DDR3 SDRAM input receiver. DQS is center-aligned with data for WRITEs. The read data is transmitted by the DDR3 SDRAM and edge-aligned to the data strobos.

The DDR3 SDRAM operates from a differential clock (CK and CK#). The crossing of CK going HIGH and CK# going LOW is referred to as the positive edge of CK. Control, command, and address signals

<table>
<thead>
<tr>
<th>Part #</th>
<th>Size GB</th>
<th>Width</th>
<th>Pwr</th>
<th>Rate (MT/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT16JSF25664HY-1G1</td>
<td>2</td>
<td>x8</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT16JSF25664HY-1G4</td>
<td>2</td>
<td>x16</td>
<td>1.5V</td>
<td>1333</td>
</tr>
<tr>
<td>MT16JSS51264HY-1G1</td>
<td>4</td>
<td>x8</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT16JSS51264HY-80B</td>
<td>4</td>
<td>x16</td>
<td>1.5V</td>
<td>800</td>
</tr>
<tr>
<td>MT4JSC6464HY-1G1</td>
<td>.5</td>
<td>x4</td>
<td>1.5V</td>
<td>1067</td>
</tr>
<tr>
<td>MT4JSC6464HY-1G4</td>
<td>.5</td>
<td>x4</td>
<td>1.5V</td>
<td>1333</td>
</tr>
<tr>
<td>MT8JSF12864HY-1G1</td>
<td>1</td>
<td>x8</td>
<td>1.5V</td>
<td>1067</td>
</tr>
<tr>
<td>MT8JSF12864HY-1G4</td>
<td>1</td>
<td>x8</td>
<td>1.5V</td>
<td>1333</td>
</tr>
</tbody>
</table>

Table 1: Micron DDR3 Memory Modules

Individual components are a good choice when the depth and width of the memory are constrained, board space is at a premium, low power is important and FPGA pins are limited. As long as the performance and memory capacity are satisfied by an individual component it is usually a good choice. Memory components are pin compatible over capacity and performance ranges, so a single design can target multiple applications by creating manufacturing options. Because components must be soldered to the board, unlike modules which are easily plugged in or removed from a board, build options based on components is a bit more complex and inventory intensive, but is still a benefit in many cases.

Micron Technology offers many DDR3 devices with capacities from 1Gb to 2 Gb with x4, x8 and x16 configurations. The MT41J devices operate at 1.5V and the MT41K devices operate at 1.35V for lower power applications. Data rates are 1066MT/s and 1333MT/s for the MT41J devices. For the MT41K devices data rates are 800MT/s and 1066MT/s. Table 2 below summarizes this data.

<table>
<thead>
<tr>
<th>Part #</th>
<th>Size GB</th>
<th>Width</th>
<th>Pwr</th>
<th>Rate (MT/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT41J128MB8Y-15E</td>
<td>1Gb</td>
<td>x8</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT41J128MB8Y-187E</td>
<td>1Gb</td>
<td>x8</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT41J128MB9H-X-15E</td>
<td>1Gb</td>
<td>x8</td>
<td>1.5V</td>
<td>1333</td>
</tr>
<tr>
<td>MT41J128MB9H-X-187E</td>
<td>1Gb</td>
<td>x8</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT41J256M4HX-15E</td>
<td>1Gb</td>
<td>x4</td>
<td>1.5V</td>
<td>1333</td>
</tr>
<tr>
<td>MT41J256M4HX-187E</td>
<td>1Gb</td>
<td>x4</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT41J64M16LA-15E</td>
<td>1Gb</td>
<td>x16</td>
<td>1.5V</td>
<td>1333</td>
</tr>
<tr>
<td>MT41J64M16LA-187E</td>
<td>1Gb</td>
<td>x16</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT41J256M8UE-187E</td>
<td>2Gb</td>
<td>x8</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT41J512M8UE-187E</td>
<td>2Gb</td>
<td>x8</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT41K128MB8H-X-187E</td>
<td>1Gb</td>
<td>x8</td>
<td>1.35V</td>
<td>1066</td>
</tr>
<tr>
<td>MT41K128MB8H-X-25</td>
<td>1Gb</td>
<td>x8</td>
<td>1.35V</td>
<td>800</td>
</tr>
<tr>
<td>MT41K256M4HX-187E</td>
<td>1Gb</td>
<td>x4</td>
<td>1.35V</td>
<td>1066</td>
</tr>
<tr>
<td>MT41K256M4HX-25</td>
<td>1Gb</td>
<td>x4</td>
<td>1.35V</td>
<td>800</td>
</tr>
</tbody>
</table>

Table 2: Micron DDR3 Memory Devices

(Note: the BY devices use an 86pin FBGA package and the HX devices use a 76pin FBGA package).

Micron Technology has DDR3 Memory Modules from 512MB to 4GB and from 800MT/s to 1333MT/s. Table 1 below shows a summary of some of the Micron Technology products.

<table>
<thead>
<tr>
<th>Part #</th>
<th>Size GB</th>
<th>Width</th>
<th>Pwr</th>
<th>Rate (MT/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT16JSF25664HY-1G1</td>
<td>2</td>
<td>x8</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT16JSF25664HY-1G4</td>
<td>2</td>
<td>x16</td>
<td>1.5V</td>
<td>1333</td>
</tr>
<tr>
<td>MT16JSS51264HY-1G1</td>
<td>4</td>
<td>x8</td>
<td>1.5V</td>
<td>1066</td>
</tr>
<tr>
<td>MT16JSS51264HY-80B</td>
<td>4</td>
<td>x16</td>
<td>1.5V</td>
<td>800</td>
</tr>
<tr>
<td>MT4JSC6464HY-1G1</td>
<td>.5</td>
<td>x4</td>
<td>1.5V</td>
<td>1067</td>
</tr>
<tr>
<td>MT4JSC6464HY-1G4</td>
<td>.5</td>
<td>x4</td>
<td>1.5V</td>
<td>1333</td>
</tr>
<tr>
<td>MT8JSF12864HY-1G1</td>
<td>1</td>
<td>x8</td>
<td>1.5V</td>
<td>1067</td>
</tr>
<tr>
<td>MT8JSF12864HY-1G4</td>
<td>1</td>
<td>x8</td>
<td>1.5V</td>
<td>1333</td>
</tr>
</tbody>
</table>

Table 1: Micron DDR3 Memory Modules
are registered at every possible edge of CK. Input data is registered on the first rising edge of DQS after the WRITE preamble, and output data is referenced on the first rising edge of DQS after the READ preamble.

Read and write accesses to the DDR3 SDRAM are burst-oriented. Accesses start at a selected location and continue for a programmed number of locations in a programmed sequence. Accesses begin with the registration of an ACTIVATE command, which is then followed by a READ or WRITE command. The address bits registered coincident with the ACTIVATE command are used to select the bank and row to be accessed. The address bits registered coincident with the READ or WRITE commands are used to select the bank and the starting column location for the burst access.

As with standard DDR SDRAM, the pipelined, multibank architecture of DDR3 SDRAM allows for concurrent operation, thereby providing high bandwidth by hiding row pre-charge and activation time. A self refresh mode is provided, along with a power-saving, power-down mode.

Input and Output Descriptions

Descriptions of the device input and output signals for the MT41J/K Family are listed in Table 3 along with a detailed description of the operation of each signal. This information is provided for review and for reference when determining the specifics of the FPGA memory interface during the FPGA Design step. A more complete description of the operation of the MT41J256M8E-187E is provided in the datasheet referenced at the end of this application note.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODT0</td>
<td>Input</td>
<td>On-die termination: ODT (registered HIGH) enables termination resistance internal to the DDR2 SDRAM. When enabled, ODT is only applied to each of the following balls: DQ0–DQ15, LDQS, UDQS, LDQS#, UDQS#, and UDQS#. The ODT input will be ignored if disabled via the LOAD MODE command.</td>
</tr>
<tr>
<td>CK, CK#</td>
<td>Input</td>
<td>Clock: CK and CK# are differential clock inputs. All address and control input signals are sampled on the crossing of the positive edge of CK and negative edge of CK#. Output data (DQs and DQS/ DQS#) is referenced to the crossings of CK and CK#.</td>
</tr>
<tr>
<td>CKE</td>
<td>Input</td>
<td>Clock enable: CKE (registered HIGH) activates and CKE (registered LOW) deactivates clocking circuitry on the DDR2 SDRAM. The specific circuitry that is enabled/disabled is dependent on the DDR2 SDRAM configuration and operating mode. CKE LOW provides pre-charge power-down mode and SELF REFRESH operation (all banks idle), or active power-down (row active in any bank). CKE is synchronous for power-down entry, power-down exit, output disable, and for self refresh entry. CKE is asynchronous for SELF REFRESH exit. Input buffers (excluding CK, CK#, CKE, and ODT) are disabled during power-down. Input buffers (excluding CKE) are disabled during refresh. CKE is an SSTL_18 input but will detect a LVCMOS LOW level once VDD is applied during first power-up. After VREF has become stable during the power on and initialization sequence, it must be maintained for proper operation of the CKE receiver. For proper SELF REFRESH operation, VREF must be maintained.</td>
</tr>
<tr>
<td>CS#</td>
<td>Input</td>
<td>Chip select: CS# enables (registered LOW) and disables (registered HIGH) the command decoder. All commands are masked when CS# is registered HIGH. CS# provides for external bank selection on systems with multiple ranks. CS# is considered part of the command code.</td>
</tr>
<tr>
<td>RAS#, CAS#, WE#</td>
<td>Input</td>
<td>Command inputs: RAS#, CAS#, and WE# (along with CS#) define the command being entered.</td>
</tr>
<tr>
<td>LDM, UDM</td>
<td>Input</td>
<td>Lower and Upper Input data mask: L/UDM is an input mask signal for write data. Input data is masked when L/UDM is concurrently sampled HIGH during a WRITE access. L/UDM is sampled on both edges of DQS. Although L/UDM is input-only, the L/UDM loading is designed to match that of DQ and DQS balls.</td>
</tr>
<tr>
<td>BA0–BA2</td>
<td>Input</td>
<td>Bank address inputs: BA0–BA2 indicate to which bank an ACTIVE, READ, WRITE, or PRECHARGE command is being applied. BA0–BA2 define which mode register including MR, EMR, EMR(2), and EMR(3) is loaded during the LOAD MODE command.</td>
</tr>
<tr>
<td>AO–A13</td>
<td>Input</td>
<td>Address inputs: Provide the row address for ACTIVE commands, and the column address and auto pre-charge bit (A10) for READ/WRITE commands, to select one location out of the memory array in the respective bank. A10 sampled during a PRECHARGE command determines whether the PRECHARGE applies to one bank (A10 LOW, bank selected by BA2–BA0) or all banks (A10 HIGH). The address inputs also provide the op-code during a LOAD MODE command.</td>
</tr>
<tr>
<td>DQ0–DQ15</td>
<td>I/O</td>
<td>Data input/output: Bidirectional data bus.</td>
</tr>
<tr>
<td>LDQS, UDQS#</td>
<td>I/O</td>
<td>Lower Data strobe, Upper data strobe: Output with read data, edge aligned with read data. Input with write data. Center aligned with write data.</td>
</tr>
<tr>
<td>RESET#</td>
<td>Input</td>
<td>Reset: RESET# is an active LOW CMOS input referenced to VSS. The RESET# input receiver is a CMOS input defined as a rail-to-rail signal with DC HIGH ≥ 0.8 × VDDQ and DC LOW ≤ 0.2 × VDDQ. RESET# assertion and desertion are asynchronous.</td>
</tr>
<tr>
<td>ZQ</td>
<td>Reference</td>
<td>External reference ball for output drive calibration: This ball is tied to an external 240Ω resistor (RZQ), which is tied to VSSQ.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>UDM</td>
<td>Input</td>
<td>Input data mask: UDM is an upper-byte, input mask signal for write data. Upper-byte input data is masked when UDM is sampled HIGH along with that input data during a WRITE access. Although the UDM ball is input-only, the UDM loading is designed to match that of the DQ and DQS balls. UDM is referenced to VREFDQ.</td>
</tr>
<tr>
<td>VDD</td>
<td>Supply</td>
<td>Power supply: 1.8V ±0.1V.</td>
</tr>
<tr>
<td>VDDSPD</td>
<td>Supply</td>
<td>SPD EEPROM positive power supply: +1.7V to +3.6V.</td>
</tr>
<tr>
<td>VREF</td>
<td>Supply</td>
<td>SSTL_18 reference voltage.</td>
</tr>
<tr>
<td>VSS</td>
<td>Supply</td>
<td>Ground.</td>
</tr>
<tr>
<td>NC</td>
<td>-</td>
<td>No Connect: These pins are not connected on the module.</td>
</tr>
</tbody>
</table>

Table 3: MT41J/K Family Pin Descriptions

IDT PCI Express Switch
IDT has a wide range of PCI Express switches for Gen2 and Gen1 applications. In our application high-throughput is important so we will focus on Gen2 (5Gb/s) applications only. IDT has two main classes of PCI Express switch - System Interconnect and I/O Expansion.

The IDT PCI Express Gen1 and Gen2 system interconnect solutions consist of a broad family of configurations for control, data and services plane traffic, the solutions provide deterministic, non-blocking, line rate performance and advanced support for inter-processor communications and peripheral sharing in blade and large-scale embedded applications. Multi-root support is provided via a partition-based switch architecture.

The IDT family of PCIe switches for I/O expansion is the broadest, most targeted set of solutions aimed at providing high-performance “aggregation” switching to fill the connectivity gap created by systems with limited high performance I/O expansion. The family offers the most comprehensive set of solutions with configurations ranging from 3 to 48 lanes and 3 to 12 ports. Table 4 shows a list of the device combinations available by number of lanes and ports.

<table>
<thead>
<tr>
<th>Device</th>
<th>Lanes</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>89HPES4T4G2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>89HPES6T6G2</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>89HPES12T3G2</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>89HPES16T4AG2</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>89HPES16T4G2</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>89HPES24T3G2</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>89HPES24T6G2</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>89HPES32T8G2</td>
<td>32</td>
<td>8</td>
</tr>
<tr>
<td>89HPES48T12G2</td>
<td>48</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 4: IDT PCI Express I/O Switches Gen2

In our application we will not need multi-root support so the I/O Expansion family is our best choice. Within this family there are a variety of Lane and Port combinations. Most of the devices have 4 times the number of lanes as ports. In our system we want to match up with the Gennum 4-lane architecture so 4 lanes is what we need for the switch. We need 3 ports, so we can have an upstream port, a downstream port and a target port. The 89HPES12T3G2 device is the best combination for us with 3 ports, each with 4 lanes.

89HPES12T3G2 Description
Utilizing standard PCI Express interconnect, the PES12T3G2 provides the most efficient fan-out solution for applications requiring high throughput, low latency, and simple board layout with a minimum number of board layers. It provides 12 Gbps (96 Gbps) of aggregated, full-duplex switching capacity through 12 integrated serial lanes, using proven and robust IDT technology. Each lane provides 5 Gbps of bandwidth in both directions and is fully compliant with PCI Express Base Specification, Revision 2.0.

The PES12T3G2 is based on a flexible and efficient layered architecture. The PCI Express layer consists of SerDes, Physical, Data Link and Transaction layers in compliance with PCI Express Base specification Revision 2.0. The PES12T3G2 can operate either as a store and forward or cut-through switch and is designed to switch memory and I/O transactions. It supports eight Traffic Classes (TCs) and one Virtual Channel (VC) with sophisticated resource management to enable efficient switching and I/O connectivity for servers, storage, and embedded processors with limited connectivity.

The PES12T3G2 contains two SMBus interfaces. The slave interface provides full access to the configuration registers in the PES12T3G2, allowing every configuration register in the device to be read or written by an external agent. The master interface allows the default configuration register values of the PES12T3G2 to be overridden following a reset with values programmed in an external serial EEPROM. The master interface is also used by an external Hot-Plug I/O expander.

The PES12T3G2 supports PCI Express Hot-Plug on each downstream port. To reduce the number of pins required on the device, the PES12T3G2 utilizes an external I/O expander, such as that used on PC motherboards, connected to the SMBus master interface. Following reset and configuration, whenever the state of a Hot-Plug output needs to be modified, the PES12T3G2 generates an SMBus transaction to the I/O expander with the new value of all of the outputs. Whenever a Hot-Plug input changes, the I/O expander generates an interrupt which is received on the IOEXPINTN input pin (alternate function of GPIO) of the PES12T3G2.
In response to an I/O expander interrupt, the PES12T3G2 generates an SMBus transaction to read the state of all of the Hot-Plug inputs from the I/O expander.

The PES12T3G2 provides 9 General Purpose Input/Output (GPIO) pins that may be used by the system designer as bit I/O ports. Each GPIO pin may be configured independently as an input or output through software control. Some GPIO pins are shared with other on-chip functions. These alternate functions may be enabled via software, SMBus slave interface, or serial configuration EEPROM.

For our application we will use a Serial EEPROM to configure the PES12T3G2 so it will be operational as soon as possible. Once the system is up and running, software may be used to create alternate configurations as required.

The 89HPES12T3G2 is an excellent companion to the Gennum Local Bus Bridge and will easily support additional acceleration targets in larger compute ‘farms’ as required by our target application.

### Xilinx Virtex-5 Generation of FPGAs

The Virtex™-5 Family uses the second generation ASMBL™ (Advanced Silicon Modular Block) column-based architecture and contains four distinct platforms (sub-families). Each platform (LX, LXT and SXT) contains a different ratio of features to address the needs of a wide variety of advanced logic designs. In addition to the most advanced, high-performance logic fabric, Virtex-5 FPGAs contain many hard-IP system level blocks, including powerful 36-Kbit block RAM/FIFOs, second generation 25 x 18 DSP slices, SelectIO™ technology with built-in digitally-controlled impedance, ChipSync™ source-synchronous interface blocks, system monitor functionality, enhanced clock management tiles with integrated DCM (Digital Clock Managers) and phase-locked-loop (PLL) clock generators, and advanced configuration options. The LXT and SXT devices also contain power-optimized high-speed serial transceiver blocks for enhanced serial connectivity, a PCI Express™ compliant integrated Endpoint block, and tri-mode Ethernet MACs (Media Access Controllers). The Virtex-5 LX, SXT, and FXT platforms include advanced high-speed serial connectivity and link/transaction layer capability.

### Xilinx Virtex-5 LX Family Overview

Xilinx Virtex-5 family devices support very high performance applications and memory interfaces. The selection of the target Virtex-5 device depends on the on-chip IP requirements, the IO requirements and the logic requirements. For our target application we will need the as large a device as possible to support the 3 memory interfaces, the local bus interface and the large amount of logic resources used in target applications.

Table 5 shows the various device options in the Virtex-5 LX family. We will select the XC5VLX330 because it is the largest device.

<table>
<thead>
<tr>
<th>Device</th>
<th>Virtex-5 Slices</th>
<th>IOs</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC5VLX30</td>
<td>4,800</td>
<td>360</td>
</tr>
<tr>
<td>XC5VLX50</td>
<td>7,200</td>
<td>480</td>
</tr>
<tr>
<td>XC5VLX85</td>
<td>12,960</td>
<td>480</td>
</tr>
<tr>
<td>XC5VLX110</td>
<td>17,280</td>
<td>680</td>
</tr>
<tr>
<td>XC5VLX155</td>
<td>24,320</td>
<td>680</td>
</tr>
<tr>
<td>XC5VLX220</td>
<td>34,560</td>
<td>680</td>
</tr>
<tr>
<td>XC5VLX330</td>
<td>51,840</td>
<td>960</td>
</tr>
</tbody>
</table>

### FPGA Design

Once the key devices have been selected the FPGA stage of the design can be started. The FPGA Design will need to integrate the Gennum Interface for the GN4124 and the DDR3 memory interface. Because the rest of the FPGA will be used for algorithm logic we will not include an embedded system in the FPGA design.

#### Gennum GN4124 FPGA Interface

In order to simplify the FPGA design, Gennum provides FPGA IP for interfacing to the Local Bus of the GN4124. The local bus FPGA IP is licensed, and royalty free for use with the GN4124.

The top level block diagram for the IP Core is shown in Figure 2, below. The Interface IP uses a layered approach so that part or all of the IP may be used. The layers are:

- **Physical Attachment Layer (PAL):** Shown on the right side of Figure 2, it provides the interface to the GN4124 taking the high-speed, narrow external buses and widening them out for use inside the FPGA
- **Application Attachment Layer (AAL):** Shown in the middle of Figure 2, it provides a target interface and the FlexDMA master mode DMA block.
- **User Application Layer (UAL):** Shown on the left side of Figure 2, the local bus core ships with a benchmarking application layer. However, this is easily replaced by the users application.
The Gennum supplied IP Core is easily integrated into the target design. The FIFO-based interfaces simplify sending and receiving application data while the Target Bus Interface supports functions like configuration, running and stopping the User Application, control and status information.

The interface between the FPGA and the GN4124 is SSTL, using DDR for the Data Bus to increase bandwidth and minimize pin use. Details on layout considerations will be given in the next chapter of this application note.

The Interface IP Core along with extensive User Guides and Data Sheets are available, after registration and verification by a Gennum sales representative, for download from the Gennum web site. They are provided free of charge for users of the GN4124.

**DDR3 Interface for the XilinxVirtex-5**

The designer will next need to create a memory interface design that can be targeted to the Xilinx Virtex-5 FPGA. The Xilinx application note, High Performance DDR3 SDRAM Interfaces in Virtex-5 Devices, guides us thru the design process and we can use the published reference design as a starting point for our design. The application note and reference design use the Xilinx ML561 Memory Evaluation Board as the target hardware platform. This board is a valuable memory design platform and can be used to test and evaluate your own memory design in a proven environment. An address on the Nu Horizons development tool web page for this valuable tool is given in the reference section of this application note.

Adrian Coscroaba from Xilinx has authored a magazine article which provides additional details on the reference design, in particular on the methods used for read and write leveling and delay calibration. The article is titled Designing DDR3 SDRAM Controllers with Today's FPGAs, listed in the reference section of this application note. Some of the descriptions in this document use content from that article as well.

**DDR3 Memory Controller Design**

The reference design uses an output from the Xilinx Memory Interface Generator (MIG) as a starting point. In the newest release of the Xilinx ISE Toolset, MIG has been updated to the 3.0 revision. This application note will use MIG 3.0 even though the design referenced in the Xilinx application note "High Performance DDR3 SDRAM Interfaces in Virtex-5 Devices" uses the 2.0 version. The resulting design will be similar, but will benefit from the new capabilities included in MIG 3.0 (read the release notes associated with MIG 3.0 on the Xilinx website for more details).

The next section will provide a quick overview of MIG, but because we will only be using the output of MIG and will focus on the changes required for implementing a DDR3 interface, only an overview of MIG is needed for the purposes of this design. If you are interested in more details on the operation of MIG refer to the Step-by-Step Application notes on DDR2 designs with Spartan-3 and Virtex-5 from Nu Horizons. These are available on the Nu Horizons web site and are listed in the reference section of this document.
MIG Overview

MIG 3.0 is included as a function within the stand alone CORE Generator provided with ESE 1.1 and can be run as an accessory. It is used to generate memory interfaces for Xilinx FPGAs. MIG generates Verilog or VHDL RTL design files, UCF constraints, and script files. The script files are used to run synthesis, map, and place and route (par) for the selected configuration. Creating memory interfaces is very easy when using the MIG tool. The designer can specify all the important parameters for a specific memory interface (address range, etc) and the MIG tool automatically creates the required design information used in conjunction with a Xilinx DDR2 Memory Controller IP Core.

Once MIG has been run it creates a complete RTL memory controller and interface. Shown below is an example of the various parameters selected by the user in MIG and produced as an output report by MIG. Many of the parameters will be familiar to the designer and are self explanatory.

Datasheet

Generated by MIG Version 3.0 on Sun May 17 13:23:30 2009

FPGA:
Target Device : xc5vlx330-ff1760
Speed Grade : -2
Selected Compatible Devices : xc5vlx110-ff1760, xc5vlx220-ff1760, xc5vlx155-ff1760

Options:
HDL : VHDL
Synthesis Tool : XST
Module Name : DDR35VLX330
No of Controllers : 1

DCI for DQ/DQS : enabled
DCI for Address/Control : disabled
Class for Address and Control : Class II
Debug Signals : Disable
Two Bytes per Bank : enabled
System Clock : Differential
IODELAY High Performance Mode : HIGH

FLICT-----------------------------------------------*/
/* Controller 0 */
FLICT-----------------------------------------------*/

Interface Parameters:
Frequency : 150
Data Width : 8
Depth : 1
Row Address : 15
Column Address : 10
Bank Address : 3
Data Mask : 0
ECC : ECC Disabled

Memory Configuration : DDR2_SDRAM :Components
Part Number : MT47H256M8XX-3
Supported Part Numbers : MT47H256M8HG-3

Other Options:
PLL : enabled
Add Test Bench : disabled

Selected Banks and Pins usage:
Data :bank 11(38) -> Number of pins used : 0
      bank 13(38) -> Number of pins used : 0
      bank 15(38) -> Number of pins used : 0
      bank 17(38) -> Number of pins used : 0
      bank 19(38) -> Number of pins used : 0
      bank 21(38) -> Number of pins used : 0
      bank 23(38) -> Number of pins used : 10
      bank 25(38) -> Number of pins used : 0
Using a PCI-Express Bridge and Switch with a Xilinx FPGA in Algorithm Acceleration Applications: A Step-By-Step Guide

Address/Control: bank 11(38) -> Number of pins used : 0
      bank 13(38) -> Number of pins used : 0
      bank 15(38) -> Number of pins used : 0
      bank 17(38) -> Number of pins used : 0
      bank 19(38) -> Number of pins used : 0
      bank 21(38) -> Number of pins used : 0
      bank 23(38) -> Number of pins used : 26
      bank 25(38) -> Number of pins used : 0

System Control: bank 11(38) -> Number of pins used : 0
      bank 13(38) -> Number of pins used : 0
      bank 15(38) -> Number of pins used : 0
      bank 17(38) -> Number of pins used : 0
      bank 19(38) -> Number of pins used : 3
      bank 21(38) -> Number of pins used : 0
      bank 23(38) -> Number of pins used : 0
      bank 25(38) -> Number of pins used : 0

System Clock: bank 3(19) -> Number of pins used : 4
      bank 4(19) -> Number of pins used : 0
      bank 6(38) -> Number of pins used : 0
      bank 11(38) -> Number of pins used : 0
      bank 12(38) -> Number of pins used : 0
      bank 14(38) -> Number of pins used : 0
      bank 18(38) -> Number of pins used : 0
      bank 20(38) -> Number of pins used : 0
      bank 22(38) -> Number of pins used : 0
      bank 24(38) -> Number of pins used : 0

Total IOs used : 43

Design Parameters:
  Mode Register:
    Burst Length : 4(010)
    Burst Type : sequential(0)
    CAS Latency : 5(101)
    Mode : normal(0)
    DLL Reset : no(0)
    PD Mode : fast exit(0)
    Write Recovery : 3(010)
  Extended Mode Register:
    DLL Enable : Enable-Normal(0)
    Output Drive Strength : Fullstrength(0)
    RTT (nominal) - ODT : 75ohms(01)
    Additive Latency (AL) : 0(000)
    OCD Operation : OCD Exit(000)
    DQS# Enable : Enable(0)
    RDQS Enable : Disable(0)
    Outputs : Enable(0)

For the reader interested in how to use the MIG for a variety of memory architectures (DDR SDRAM, DDR2 SRAM, DDR2 SDRAM, QDRII SRAM, and RLDRAM I) and for different Xilinx FPGA families (Spartan-3/3E/3A/3AN/3A DSP FPGA Families and Virtex-4/5 FPGA Families) a detailed description is given in the MIG Users Guide referenced at the end of this document.

**Xilinx DDR3 Interface Reference Design**

There are two key functions that must be provided by the DDR3 memory interface: data capture and controller implementation.

**Data Capture**

The data capture technique must use the source synchronous read data and read strobes to create the proper data capture window (the precise timing for when the valid data is to be captured by the FPGA). This is done by delaying the data strobe using a programmable chain of 75ps tap delay elements called IODELAY. The number of delay elements is determined during the read calibration stage to ensure that the clocking signal to the IDDR register, a delayed DQS, is properly centered to the data (DQ) valid window. This calibration is performed during the system initialization phase after power up to ensure that setup and hold times are met at the IDDR register.

The 75 ps delay elements in the IODELAY are also automatically calibrated for any temperature and voltage variations that may occur during system operation. In the second stage of data capture, the outputs of the IDDR register are routed to the fabric, where they are recaptured and synchronized to the FPGA system clock that runs the rest of the DDR3 controller logic. The read data is stored and provided for the back-end user interface to the rest of the FPGA design.

The write data to the memory device is generated by the output DDR registers (ODDR) available in Virtex-5 I/O blocks. The ODDR transmit data (DQ) and strobe (DQS) signals center-aligned using the internal CLK0 and CLK90 signals. These are generated by the outputs of the digital clock manager (DCM) that drives the rest of the FPGA logic and the DDR3 memory devices.

Because this reference design is for a point-to-point, 32-bit-wide DDR3 component implementation, “write leveling” was not required. Therefore, the DQS and DQ signals need not be further skewed to compensate for different flight times to the inputs of the DDR3 memory components. For DDR3 DIMM implementations that require write leveling, designers can employ additional programmable circuits in the Virtex-5 FPGA to implement skew for DQS and DQ signals.

The Xilinx Virtex-5 design does not use the DDR3 multi-purpose register (MPR) that provides read data for calibration before normal operation. The previous-generation DDR2 reference design using a write followed by a read calibration pattern during system initialization was proven to be effective for read timing adjustments. Thus, a similar calibration pattern was used in this DDR3 reference design.

**Controller Implementation**

The controller is implemented using FPGA fabric and internal memory resources and leverages most of the MIG 3.0 generated DDR2 design. The controller has the ability to keep four banks open at a time. The banks are opened in the order of the commands that are presented to the controller. In the event that four banks are already opened and an access arrives to the fifth bank, the least recently used bank will be closed and the new bank will be opened. All the banks are closed during auto refresh and will be opened as commands are presented to the controller.

The controller state machine manages issuing the commands in the correct sequencing order while determining the timing requirements of the memory. Before the controller issues the commands to the memory:

1. The controller decodes the address located in the FIFO.
2. The controller opens a row in a bank if that bank and row are not already opened. In the case of an access to a different row in an already opened bank, the controller closes the row in that bank and opens the new row. The controller moves to the Read/Write states after opening the banks if the banks are already opened.
3. After arriving in the Write state, if the controller gets a Read command, the controller waits for the write_to_read time before issuing the Read command. Similarly, in the Read state, when
the controller sees a Write command from the command logic block, the controller waits for the read_to_write time before issuing the Write command. In the Read or Write state, the controller also asserts the read enable to the address FIFO to get the next address.

4. The commands are pipelined to synchronize with the Address signals before being issued to the DDR3 memory.

Memory Controller Interface

The memory controller must be integrated into the rest of the design. Typically this involves connecting the memory controllers user interface to the portion of the design that ‘drives’ the memory controller. This could be a simple as a processor bus or more complicated and involve significant optimization of the data access sequence to memory. Any user logic will communicate with the memory controller over the same set of signals however. The balance of this section provides a quick overview of these interface signals and a short functional description of the operation of the controller. More details of the controller are available in the Xilinx xapp867 application note listed in the reference section of this application note.

Signal Descriptions

The interface signals used to interact with the memory controller are given in Table 6 below. The signal name is listed along with the type of signal (input or output) and a short description of each signal. A short description of the operation of the memory controller follows the table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS_CLK, SYS_CLKb</td>
<td>Input</td>
<td>Differential system clock inputs.</td>
</tr>
<tr>
<td>reset_in_n</td>
<td>Input</td>
<td>Active low system reset signal</td>
</tr>
<tr>
<td>Burst_done</td>
<td>Input</td>
<td>Indicates that the current burst transaction is done</td>
</tr>
<tr>
<td>User_command_register</td>
<td>Input</td>
<td>Memory Command</td>
</tr>
<tr>
<td>User_data_mask</td>
<td>Input</td>
<td>Data mask for partial write operations</td>
</tr>
<tr>
<td>User_input_data</td>
<td>Input</td>
<td>Data to write to memory</td>
</tr>
<tr>
<td>User_input_address</td>
<td>Input</td>
<td>Address of memory location</td>
</tr>
<tr>
<td>Init_done</td>
<td>Output</td>
<td>Indicates memory initialization is complete</td>
</tr>
<tr>
<td>Ar_done</td>
<td>Output</td>
<td>Indicates memory auto refresh is complete</td>
</tr>
<tr>
<td>Auto_ref_req</td>
<td>Output</td>
<td>Indicates memory refresh is required</td>
</tr>
<tr>
<td>User_cmd_ack</td>
<td>Output</td>
<td>Command acknowledge for the user read or write command</td>
</tr>
<tr>
<td>Clk_tb</td>
<td>Output</td>
<td>Provided to the user for synchronization</td>
</tr>
<tr>
<td>Clk90_tb</td>
<td>Output</td>
<td>Provided to the user for synchronization</td>
</tr>
<tr>
<td>Sys_rst_tb</td>
<td>Output</td>
<td>Provided to the user for synchronization</td>
</tr>
<tr>
<td>Sys_rst90_tb</td>
<td>Output</td>
<td>Provided to the user for synchronization</td>
</tr>
<tr>
<td>Sys_rst180_tb</td>
<td>Output</td>
<td>Provided to the user for synchronization</td>
</tr>
<tr>
<td>User_data_valid</td>
<td>Output</td>
<td>Indicates data read from memory is available</td>
</tr>
<tr>
<td>User_output_data</td>
<td>Output</td>
<td>Data read from memory</td>
</tr>
</tbody>
</table>

Table 6: Memory Controller User Interface Signals

Functional Description

The memory controller user interface simplifies the process of interacting with the off-chip DDR3 memory device. Low level tasks like initialization, refresh timing, read and write data capture and other details are simplified by the memory controller and the user can focus on higher level actions like reading and writing.

There are commands supported by the controller and these are specified on the user command signals. The commands are Nop, memory initialization, auto-refresh, write, read and load mode.

- Nop: No operation.
- Auto Refresh: Used to execute a refresh command.
- Write: Executes a sequential block write to memory.
- Read: Executes a sequential read from memory.
- Load Mode: Loads user data to the memory mode register.
• Pre-charge: Deactivate the open row in a particular bank.

• Active: Activate a row in preparation of a Read or Write command.

The user indicates the operation desired on the user_command_register signals. When clocked by the SYS_CLK (rising edge) the command is executed by the controller. The Auto Refresh command is a multi-cycle command and is complete when the Ar_done signal is asserted.

The Write command begins with the user indicating a write on the user_command_register. The controller acknowledges receipt of the command when the user_cmd_ack signal is asserted. After the acknowledge is received the user can place the address and data on the controller inputs. The user then asserts burst-done to indicate the write cycle is complete. (The exact timing of each of these signals is given in detail in xapp454. For the purposes of this overview these details will be skipped).

Memory Initialization is done automatically in the physical layer portion of the design and need not be initiated from the user interface.

Integration of the Reference Design

The reference design can be downloaded and integrated into an existing design in several ways. Source code is provided in Verilog and VHDL and has simulation directories for ModelSim. Additionally, the design has been run through both XST and Synplify Pro. The existing design meets internal fabric timing at 400MHz in a -3 speed grade of an LX500T Virtex-5 FPGA (using the included UCF file and ISE 9.2 SP3).

In order to incorporate the DDR3 design into an ISE project simply follow the steps listed below for a Verilog based design (VHDL and VHDL and has simulation directories for ModelSim). Additionally, the design has been run through both XST and Synplify Pro. The existing design meets internal fabric timing at 400MHz in a -3 speed grade of an LX500T Virtex-5 FPGA (using the included UCF file and ISE 9.2 SP3).

1) Unzip the XAPP_867.zip file
2) Unzip the src.zip which contains the both the verilog and the vhdl source code
3) Unzip the sim.zip which contains sample .do file along with a .bat file for windows
   a) Edit the .do file to point to the appropriate gbl.v directory
   b) Other modifications are possible based on the simulation environment
4) Unzip the implement_xst.zip or implement_synplify.zip to open an ISE project based on the particular synthesis tool (the design has been verified with the settings in the project).

After the user connects the memory and the Local Bus Interface into their design and simulates the application successfully the FPGA design portion of the project is completed. The next step is the Board Design (in many cases some of the FPGA and Board Design activities can be done in parallel and this is usually advantageous. For the purposes of this application note will assume that these phases are done sequentially however).

During the board design portion of the design process there are several key design considerations that need to be addressed. These considerations can be grouped as follows: Pin Assignment considerations, Component Placement considerations, Simultaneous Switching Output considerations and Power Supply design considerations. Each of these considerations will be addressed in detail in the following sections of this application note. Skip over the sections that you are not interested in since there is a significant amount of information in each section. You can return to these topics later when they are of more interest.

The main Pin Assignment and Layout considerations in our design will be for either the Gennum Interface or the DDR3 Memory Interface. We will consider the DDR3 Interface first.

DDR3 Interface Pin Assignment and Layout Considerations

MIG 3.0 generates pin assignments for a memory interface based on certain rules depending on the design technique, but does not provide the best possible pin assignment for every board implementation. During layout it might be necessary to swap pin locations depending on the number of layers available and the interface topology. The best way to change the pin assignment is to first apply changes on a byte basis then swap bits within a byte. Calculate the PCB loopback length, if required, after pin swapping and trace matching. The following rules of thumb are provided to help designers determine how pins can be swapped.

Any changes to the pin assignments require modifications to the UCF provided by MIG and might require changes to the source code depending on the changes made.

For all MIG 3.0 Virtex-5 designs, the address and control pins can be swapped with each other as needed to avoid crossing of the nets on the printed circuit board.

All strobe signals must be placed on clock-capable inputs (such as DQS, CQ, and QK) with P connected to the P side and N connected to the N side of the pair. If only single-ended strobes are provided, the signal is placed on the P input of the clock-capable I/O pair.

Data lines used to read data from a memory are placed in the same bank as their associated strobe. Data is captured in the ISERDES block using the strobe signal. The strobe is passed through the BUFI to delay it with respect to the data input. Address and control signals are to be placed together in the same bank or placed in banks near each other to minimize the route delays for these signals inside the FPGA.

The strobe must be placed on clock-capable I/O with the data for the strobe placed in the same bank. A byte can be swapped with another byte as long as all the necessary signals associated with
that byte (strobe, data, and data mask) are located in the same bank. Within a bank, strobes can be swapped with other strobes while the rest of the pins in a bank can be swapped as needed.

**Signal Termination Considerations**

For the proper operation of the memory interface specific signal termination guidelines need to be followed. The following guidelines apply to termination for Virtex-5 FPGAs when interfacing to DDR2 or DDR3 SDRAM devices:

1. Single-ended signals are to be terminated with a pull-up of 50Ω to VTT at the load. A split 100Ω termination to VCCO and 100Ω termination to GND can be used, but takes more power. For bidirectional signals, the termination is needed at both ends of the signal (DCI/ODT or external termination).
2. Differential signals are to be terminated with a 100Ω differential termination at the load. For bidirectional signals, termination is needed at both ends of the signal (DCI/ODT or external termination).
3. All termination must be placed as close to the load as possible. The termination can be placed before or after the load provided that the termination is placed within one inch of the load pin.
4. DCI can be used at the FPGA as long as the DCI rules are followed (such as VRN/VRP).

A summary of Xilinx termination recommendations for the DDR2/3 devices used in UG199 given in the reference section of this document are given in Table 7 below.

### Table 7: DDR2/3 SDRAM Device Terminations

<table>
<thead>
<tr>
<th>Signal</th>
<th>FPGA Driver</th>
<th>Termination at FPGA</th>
<th>Termination at Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data (DQ)</td>
<td>SSTL18-II</td>
<td>No termination</td>
<td>No termination (use 75Ω ODT)</td>
</tr>
<tr>
<td>Data Strobe (DQS, DQS#)</td>
<td>DIFF_SSTL18-II_DCI</td>
<td>No termination</td>
<td>No termination (use 75Ω ODT)</td>
</tr>
<tr>
<td>Data Mask (DM)</td>
<td>SSTL18-II</td>
<td>No termination</td>
<td>No termination (use 75Ω ODT)</td>
</tr>
<tr>
<td>Clock (CK, CK̅)</td>
<td>SSTL18-II</td>
<td>No termination</td>
<td>100Ω differential termination between pair</td>
</tr>
<tr>
<td>Address (A, BA)</td>
<td>SSTL18-II</td>
<td>No termination</td>
<td>50Ω pull-up to 0.9V after the last component</td>
</tr>
<tr>
<td>Control (RAS, CAS, WE, CE, and CKE)</td>
<td>SSTL18-II</td>
<td>No termination</td>
<td>50Ω pull-up to 0.9V after the last component</td>
</tr>
</tbody>
</table>

**Timing Related Board Layout Considerations**

Once signals have been assigned and trace geometry has been optimized the trace lengths of critical signals need to be examined to make sure timing constraints are met. For DDR2 memory interface designs with Virtex-5 FPGAs running at 400MHz the following maximum electrical delays need to be met:

1) ± 25 ps maximum electrical delay between any DQ and its associated DQS/DQS#
2) ± 50 ps maximum electrical delay between any address and control signals and the corresponding CK/CK#.
3) ± 100 ps maximum electrical delay between any DQS/DQS# and CK/CK#.

**DDR3 Interface Simultaneous Switching Output Considerations**

Ground bounce must be controlled to ensure proper operation of high-performance FPGA devices. When multiple output drivers change state at the same time, power supply disturbance occurs. These disturbances can cause undesired transient behavior in output drivers, input receivers, or in internal logic. These disturbances are often referred to as Simultaneous Switching Output (SSO) noise. The SSO limits govern the number and type of I/O output drivers that can be switched simultaneously while maintaining a safe level of SSO noise. SSO of an individual FPGA I/O bank is calculated by summing the SSO contributions of the individual I/O standards in the bank. The SSO contribution is the percentage of full utilization of any one I/O standard in any one bank. The Weighted Average SSO (WASSO) calculation is the done by combining the SSO contributions of all I/O in a bank into a single figure.

When using MIG the bank assignment step helps restrict memory interface SSOs to within the guidelines required for controlling ground bounce. After the designer includes all signals in the design the WASSO calculation should be done to insure the final WASSO calculation is within the device requirements. When your signal assignment is complete follow the steps in the balance of this section to check your design meets the device WASSO constraints.

A Microsoft Excel-based spreadsheet entitled “WASSO Calculator” has been created by Xilinx to automate these calculations. The WASSO calculator uses PCB geometry, such as board thickness, via diameter, and breakout trace width and length, to determine board inductance. It determines the smallest undershoot and logic-Low threshold voltage among all input devices, calculates the average output capacitance, and determines the SSO allowance by taking into account all of the board-level design parameters mentioned in this document. In addition, the
WASSO calculator performs checks to ensure the overall design does not exceed the SSO allowance.

The WASSO Calculator with support for Virtex-5 is available as part of xapp689 (rev 1.2). Simply download the calculator, refer to xapp689 (rev 1.2) for a description of each entry and enter the specifications for your PCB Design Parameters, the devices being driven by the FPGA, the parameters for the FPGA output loading and the FPGA bank allocation of resources. The calculator will determine what percentage of the WASSO limit is being utilized for each bank. If your design meets the WASSO requirements of the target device the green OK cells will be highlighted in the spreadsheet.

**DDR3 Interface Example Board Layout**

An example board layout for the Virtex-5 XC5VLXT50T and Micron MT41J64M16LA-187E (running at 800MT/s) is available as part of the Xilinx ML561 ML561 Memory Interface Development Board. These are not our specific target devices, but the considerations are similar enough that it is a useful starting point. This kit contains a variety of memory interface example designs for the Virtex-5 Family. Complete schematics, layout files, reference designs and more are all available with this design. Component placement considerations will be reviewed followed by signal integrity design concerns and measurements.

---

![Figure 3: FPGA and DDR3 Memory Device Component Placement](image)

**Example Component Placement**

Placement of memory device in relation to the Virtex-5 FPGA is shown in Figure 1. The pinout for the memory interface in the Virtex-5 FPGA is located in I/O Bank 11, which is on the south side of the XC5VLXT50T. The relative placement of the key components makes it easy to route the printed circuit board, given the pinout generated by MIG. Notice that it helps to understand the orientation of the devices when assigning I/O banks to the FPGA in MIG. If the orientation is changed later, MIG can easily be re-run with new I/O bank assignments.

**Signal Integrity Considerations**

It is an important design step to simulate signal integrity prior to fabricating a PCB when using high performance memory components. If this step is skipped you may find that a small error in signal placement or additional vias can reduce the overall memory performance due to signal integrity issues. Xilinx has provided detailed signal integrity simulations with correlation results to an actual hardware design for the ML561 Memory Interface Development Board. A reference for the user guide for this board can be found in the reference section of this document. (Even though the measurements on the ML561 board are for DDR2 devices, the same steps, techniques and methodology applies for DDR3 memories and a review of this process is an excellent step prior to completing a DDR3 design.)

In the Signal Integrity Correlation section of UG199 the simulation model of a specific memory interface signal is shown along with the simulated results and the actual hardware measurement on the fabricated PC board. The simulation includes models for the FPGA output, board signal traces, vias, and memory component. Actual hardware measurements were done at a probe point on a via location on the memory device. The simulation and board measurements correlate well with one another and demonstrate the value in doing the simulation prior to board fabrication.

A selection of the critical diagrams and measurements are given in appendix 1 of this document. Please refer to UG199 for a more complete description and additional examples of simulation results and hardware measurements.

**Example Signal Traces**

The Xilinx ML561 evaluation platform contains a Virtex-5 to DDR3 device interface that shows some of the important techniques used to route key signals between the FPGA and the DDR3 component. In general the trace length for common signal busses (like data or address signals) need to all be the same length to keep skew between signals within the recommended tolerance. Additionally, signals like DQS, with a positive and negative signal need to be routed close to each other to make sure the signal has the best possible skew and noise characteristics.

An example layout of signals between the FPGA and the DDR3 devices is shown in Figure 4 below. The yellow lines are the signal traces and some of the important signals are identified. Dual traces, for both a positive and negative signal, can be seen as routing as close together as possible for the entire length of the trace. For example, DDR3 Clk0, Clk1 and DRQ are good examples of this. Notice that other signals like DQ and Address have closely matched trace lengths but need not travel next to each other the whole length. In fact, looking at a signal like DQ, it bends and weaves to make sure the trace length is similar to the rest of the bus, but adjacency isn’t an important criteria.
The example layout above comes from the Xilinx Memory Interface Development Board listed in the reference section of this application note. Complete layout files, schematics and bill of materials information are included with a board purchase. This Development Board is an excellent way to get a head start on any memory interface design.

Gennum Interface Pin Assignment and Layout Considerations
The pinout and layout considerations for the Gennum Interface are fairly simple in comparison to the DDR3 interface. The use of SSTL signals and standard DDR clocking for the data signals makes it easy to interconnect the FPGA and the GN4124. Signals within the TXData and RxData busses should be routed together, with little difference in routing length, to minimize skew. Because these are point to point signals, if they can be routed with a minimum (hopefully no) vias the resulting signals will be clean and should result in a very robust design. If pin locations can be made reasonably close together, within the guidelines of the SSO calculations given above, this will make it easier to keep signal length variations small. Notice that each bus has its own differential clock signal. These signals should also be length matches with their respective data busses to minimize skew between clock and data. Because the clock signals are differential, each trace should run near each other to minimize differences in common mode noise and routing pattern. This will keep the clock signals clean and deliver robust data.

The control signals should also be grouped and routing length variations minimized. Since these signals are single data rate this requirement is less critical, but should still be observed as much as possible.

Careful placement of the FPGA next to the GN4124 and allocation of FPGA signals to specific banks pins with knowledge of the GN4124 pinout makes a big difference in the resulting routing.

Power Supply Design Considerations
Power supply design and the selection of decoupling components can be of critical importance to any subsystem with a combination of analog and digital functions like Ethernet.

Perhaps the most critical consideration is the sequence of applying power to the system. It is usually the best practice to ramp the core first (the 1.2V supply) followed by the 1.8V and then the 3.3V for the IOs to reduce stress on the device. This can be accomplished in a variety of ways, and is usually achieved with the selection of the proper value of power supply module external components.

The other key considerations can be grouped into these categories: Power Supply, PCB Decoupling Capacitors, PCB Bypass Capacitors and PCB Bulk Capacitors.

Power Supply
- Ensure adequate power supply ratings. Verify that all power supplies and voltage regulators can supply the amount of current required.
- Power supply output ripple should be limited to less than 50 mV.
- Noise levels on all power planes and ground planes should be limited to less than 50 mV.
- Ferrite beads should be rated for 4 – 6 times the amount of current they are expected to supply. Any de-rating over temperature should also be accounted for.
PCB Decoupling Capacitors
- Every high-speed semiconductor device on the PCB assembly requires decoupling capacitors. One decoupling cap for every power pin is necessary.
- Decoupling capacitor value is application dependent. Typical decoupling capacitor values may range from 0.1 μF to 0.001 μF.
- The total decoupling capacitance should be greater than the load capacitance presented to the digital output buffers.
- Typically, Class II dielectric capacitors are chosen for decoupling purposes. The first choice would be an X7R dielectric ceramic capacitor for its excellent stability and good package size vs. capacitance characteristics. Low inductance is of the utmost importance when considering decoupling capacitor characteristics.
- Each decoupling capacitor should be located as close as possible to the power pin that it is decoupling.
- All decoupling capacitor leads should be as short as possible. The best solutions are plane connection vias inside the surface mount pads. When using vias outside the surface mount pads, pad-to-via connections should be less than 5 – 10 mils in length.
- Trace connections should be as wide as possible to lower inductance.

PCB Bypass Capacitors
- Bypass capacitors should be placed near all power entry points on the PCB. These caps will allow unwanted high-frequency noise from entering the design; the noise will simply be shunted to ground.
- Bypass capacitors should be utilized on all power supply connections and all voltage regulators in the design.
- Bypass capacitor values are application dependent and will be dictated by the frequencies present in the power supplies.
- All bypass capacitor leads should be as short as possible. The best solutions are plane connection vias inside the surface mount pads. When using vias outside the surface mount pads, pad-to-via connections should be less than 5 – 10 mils in length. Trace connections should be as wide as possible to lower inductance.
- Whenever a ferrite bead is implemented, bulk capacitance must be used on each side of the ferrite bead.

PCB Bulk Capacitors
- Bulk capacitors must be properly utilized in order to minimize switching noise. Bulk capacitance helps maintain constant DC voltage and current levels.
- Bulk capacitors should be utilized on all power planes and all voltage regulators in the design.
- All bulk capacitor leads should be as short as possible. The best solutions are plane connection vias inside the surface mount pads. When using vias outside the surface mount pads, pad-to-via connections should be less than 5 – 10 mils in length. Trace connections should be as wide as possible to lower inductance.

Conclusion

This application note has provided a step-by-step guide to designing an Application Accelerator using a Gennum PCI Express Local Bus Bridge, an IDT PCI Express switch, Micron DDR3 Memory devices, and a Xilinx Virtex-5 FPGA. The reader should be able to more easily implement PCI Express interfaces and DDR3 interfaces with a better understanding of the major design considerations and concerns present in these types of designs. It is hoped that the reader can also extend the concepts presented in this app note to other Xilinx FPGAs, Gennum PCI Express Local Bus Bridge, IDT PCI Express Switches and Micron DDR3 designs to simplify those projects as well.

For the reader who wants additional levels of detail on the Xilinx, IDT, Gennum and Micron products used in this app note a list of valuable documents is given in the Reference section at the end of this document. Additionally, Nu Horizons sponsored a PCI Express Summit with multiple product presentations covering some of these devices. Point your browser to http://www.nuhorizons.com/xpresstrack/pcisummit to learn more about the information generated during the summit. Note: you will need to obtain a login ID and Password from your Nu Horizons FAE.

References
1) Micron Technology MT41J Family Data Sheet (2Gb_DDR3_TOC.fm - Rev C 1/31/08 EN)  
2) Gennum PCI Express Local Bridge Web Page (GN4124) http://www.gennum.com/video/products/gn4124
3) IDT PCI Express Switch for Gen2 Web Page http://www.idt.com/?partid=89HPE12T3G2ZABC
7) WASSO Calculator for Virtex-5 devices (Rev ) http://www.xilinx.com/bvdocs/userguides/ug190_SSO_Calculator.zip
8) High-Performance DDR2 SDRAM Interface In Virtex-5 Devices (XAPP858, Rev ) http://www.xilinx.com/support/documentation/application_notes/xapp858.pdf
10) DDR3 Reference Design (xapp867.zip) http://www.xilinx.com/bvdocs/appnotes/xapp867.zip
11) Designing DDR3 SDRAM Controllers with Today's FPGAs, Adrian Cosoroba- Xilinx (12/12/2007)  
    http://www.pldesignline.com/showArticle.jhtml?articleID=204801895

Notice Of Disclaimer

Nu Horizons is disclosing this Application Note to you "AS-IS" with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Nu Horizons. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. Nu Horizons makes no representations or warranties, whether express or implied, statutory or otherwise, including, without limitation, implied warranties of merchantability, noninfringement, or fitness for a particular purpose. In no event will Nu Horizons be liable for any loss of data, lost profits, or for any special, incidental, consequential, or indirect damages arising from your use of this application note.