A Device-Level Architecture for FPGA Co-Processors in Embedded Computing Platforms

Chris Conger, Andrew White, and Dr. David Bueno

Honeywell Inc., Space Electronic Systems

Honeywell

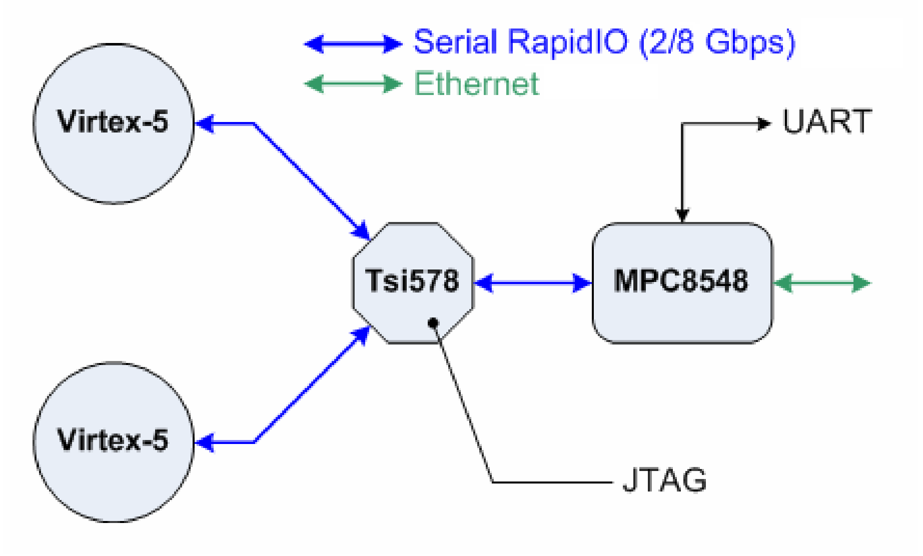
Introduction

- **FPGAs widely used in payload processing platforms**
  - Hardware accelerators to software processors
  - Pre-processor devices for sensors or other external inputs
- **Inter-device interconnect and device-level architecture critical for performance of overall system**
  - Bus-based interconnects restrict achievable speedup via sub-optimal data throughput efficiency
  - FPGA’s must provide data movement between multiple targets
    - **On-chip internal processing memory**
    - **External storage memory**
    - **Off-chip interconnect interfaces**
  - Control mechanism for FPGAs from software also performance-critical
- **Presenting architecture of experimental testbed, featuring:**
  - Xilinx Virtex-5 FPGAs, custom-designed architecture for efficient data movement and control
  - Freescale MPC8548 PowerPC software processors
  - Serial RapidIO system-level interconnect

Prototype Overview (I)

Honeywell

- **Heterogeneous system of processing elements**
  - One SW processor, two FPGAs
  - Connected via Serial RapidIO
  - SBC serves as master and user interface, runs Linux
  - FPGAs are slaves to SBC (generic hardware co-procs)
- **Powered by single ATX**
- **Impressive capabilities**
  - Each RapidIO link operates at 8 Gbps in each direction
  - 25.6 Gbps external memory throughput for FPGAs
  - Transparent remote memory access



- **Tundra Tsi578 Serial RapidIO 8/16-port switch and development board**
- **Freescale PowerQUICC-III 8548 Advanced Mezzanine Card (AMC)**
- **Xilinx ML523 boards (x2), XC5VLX110T-FF1136-1**
- **Xilinx RapidIO IP core, Serial PHY, I/O-Logical LOG**

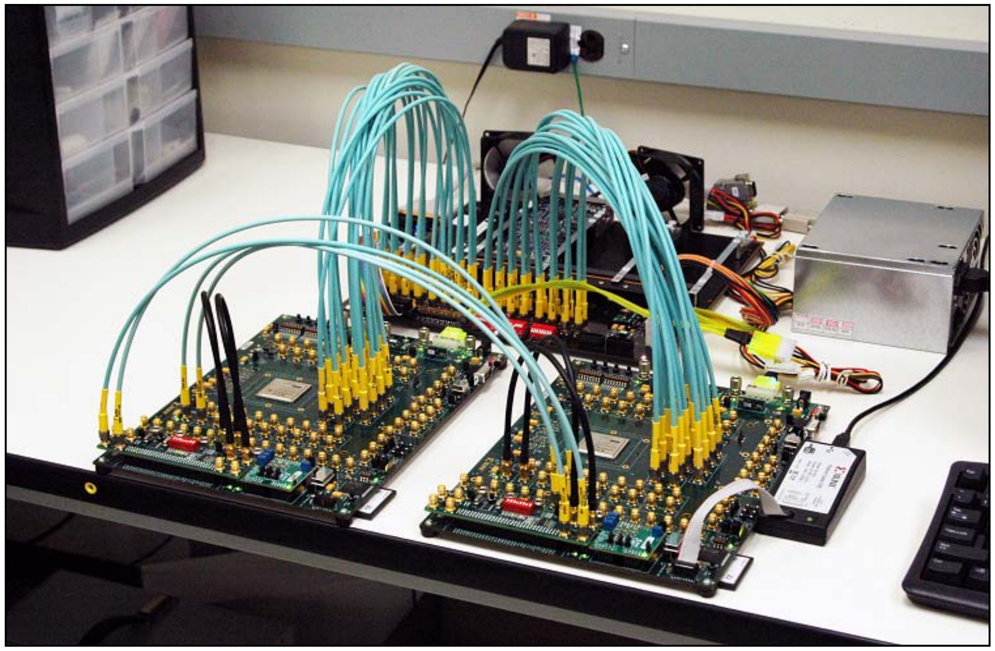
Prototype Overview (II)

Honeywell

- **Testbed architecture and environment:**
  - PowerQUICC SBC serves as testbed “master” node (runs Linux), FPGAs act as slaves w/ remotely-accessible memory
  - Applications built in C with PowerPC cross-compiler, extremely simple API to use RapidIO and FPGAs
  - Majority of FPGA design (VHDL) can remain fixed, only *computational modules* need to be changed for different apps

RapidIO®

TUNDRA



XILINX®

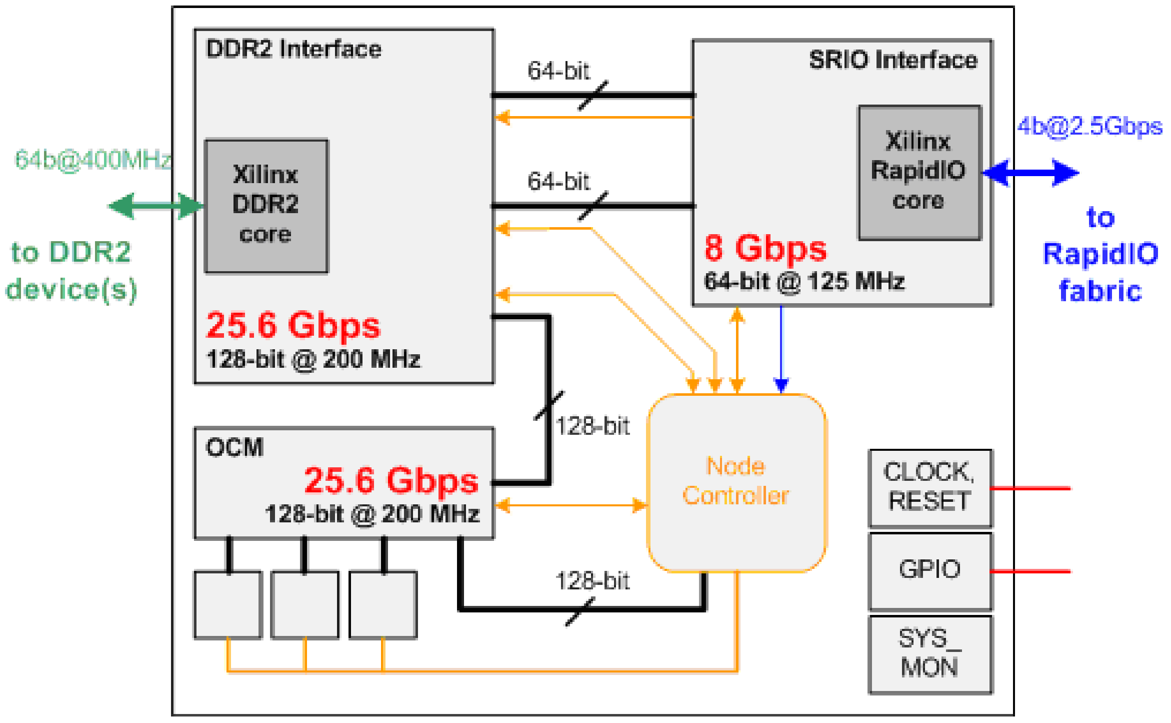
freescale™ semiconductor

FPGA Device Architecture

Honeywell

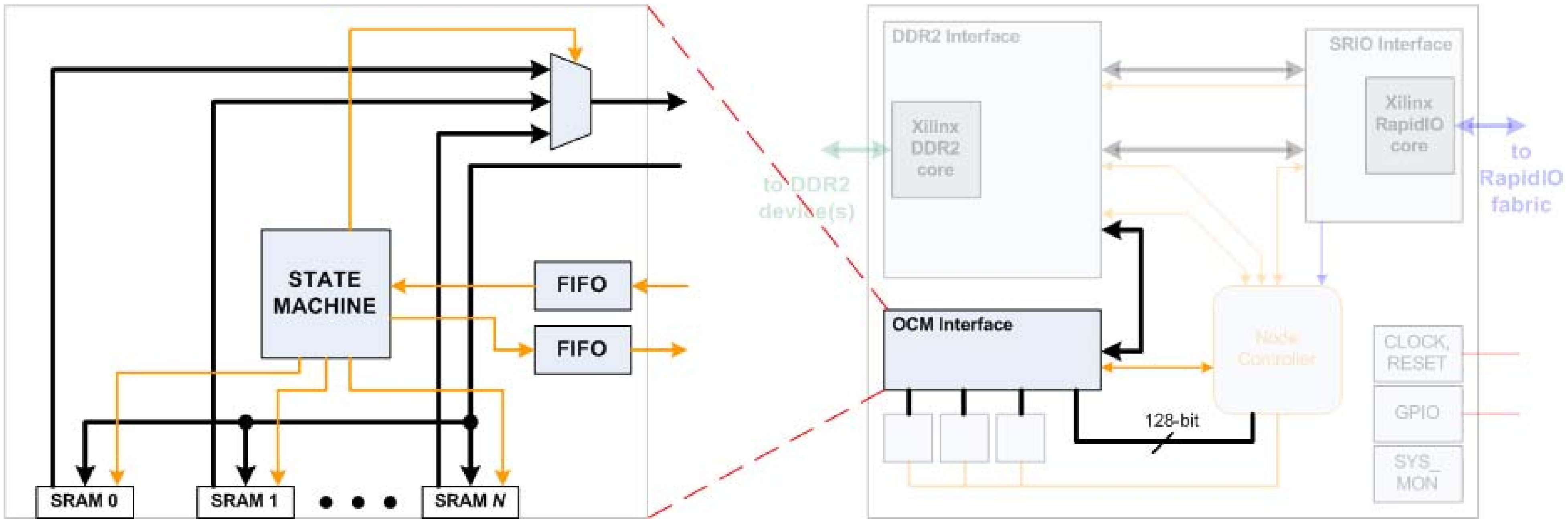
- **High-performance computational resource**
- **Majority of design provides *framework within which one can develop applications***
  - FPGA architecture does not perform computation
  - Example co-processors
- **FPGA HDL represents majority of design effort**
  - Evolving node design
  - Leveraged previous work
- **Highly-efficient design**

- Xilinx Virtex-5 LX110T FPGA
- 128 MB external DDR2 SDRAM
- 128 MB CompactFlash card
- 1x/4x Serial RapidIO link, 2.5 GHz
- Internal clock freq.'s from 100-200 MHz
- All-hardware control fabric



Focus on Components: On-Chip Memory Controller

Honeywell

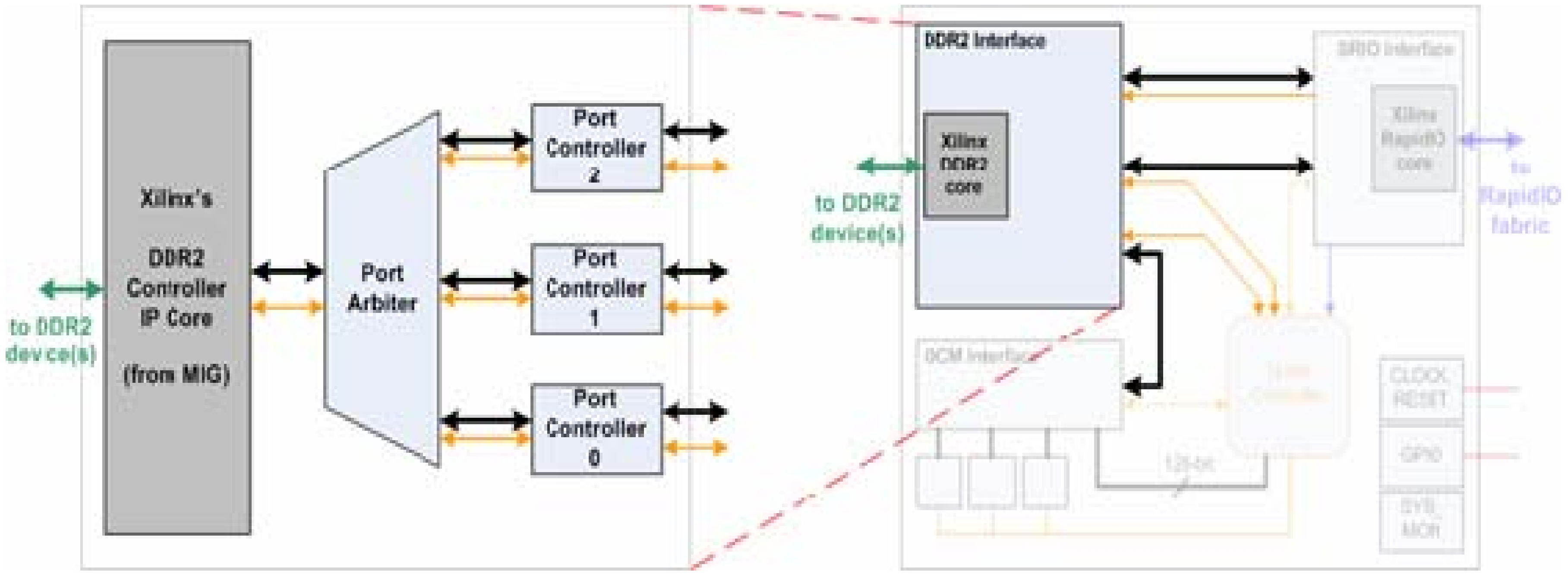


- Up to 128 KB/co-proc
- 128-bit, 200 MHz internal (can be faster)

- **On-Chip Memory (OCM) Controller**
  - Muxes a single pair of FIFOs with numerous SRAM interfaces
  - Commands received through command FIFO (read or write)
  - Command responses sent through another FIFO after all data has been transferred
- **Up to four (4) SRAM interfaces**
  - One reserved for Node Controller, so up to three (3) co-processors
  - Responsible for moving data in/out of co-processors

Focus on Components: DDR2 Memory Controller

Honeywell



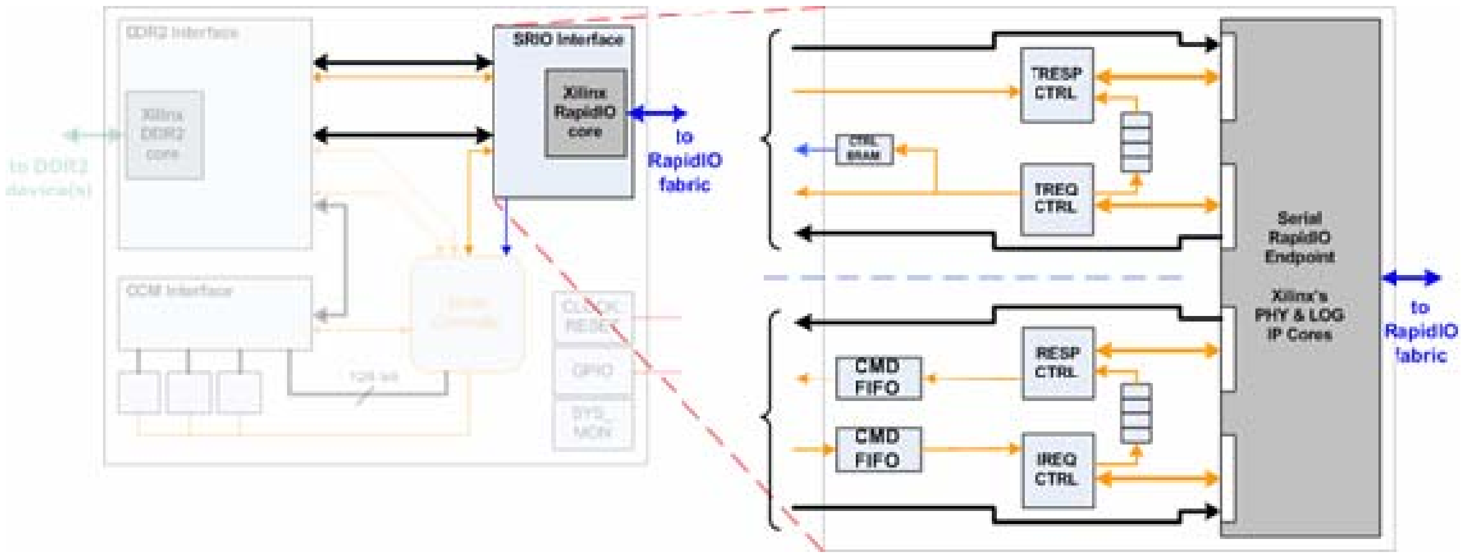
- 128 MB DDR2 memory
- 64-bit @ 400 MHz DDR (up to 666 MHz DDR)
- 128-bit, 200 MHz internal

- **External DDR2 memory controller**
  - Central component in FPGA design, all transfers to/from DDR2
  - Xilinx-provided DDR2 memory controller at heart, wrapped with custom multi-interface wrapper
- **Provides dedicated control and data ports for each internal component** (2 data FIFOs, 2 control FIFOs)
  - OCM controller, RapidIO interface (2 ports)
  - Improves concurrency, although still ultimately serialized
- **Port arbiter prevents starvation… simple round-robin**

Focus on Components: SRIO Endpoint

Honeywell

- I/O-Logical interface
- 1-/4-bit @ 2.5 Gbps link
- 64-bit @ 125 MHz internal

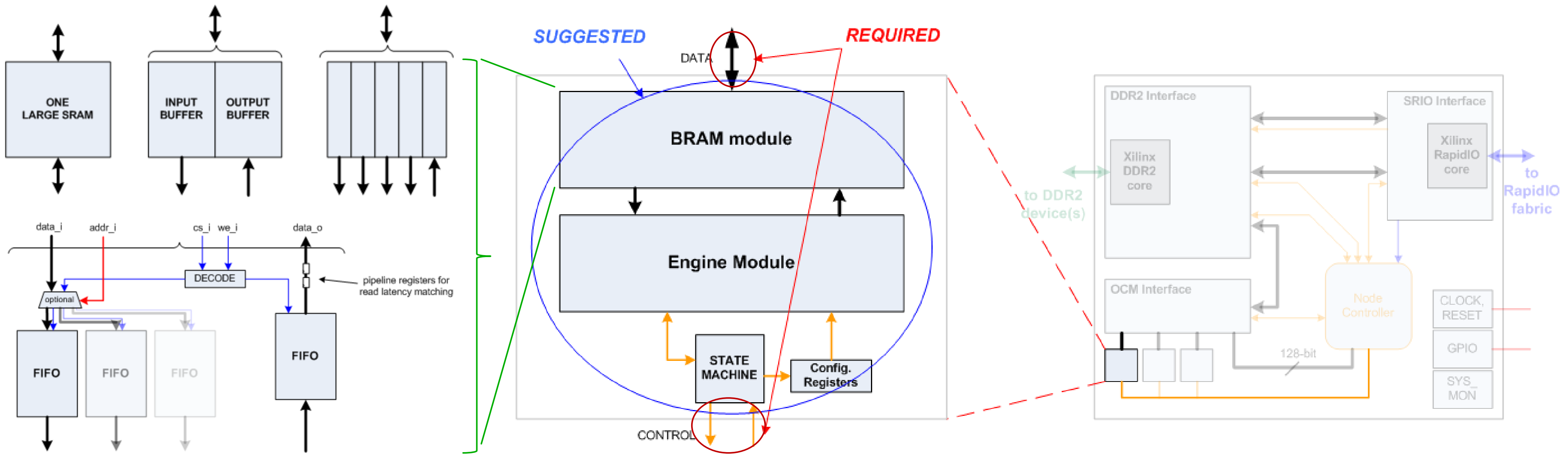


- **RapidIO endpoint design (from Xilinx) wrapped to provide simplified command interface**
  - RapidIO core provides four (4) independent ports to user
  - Two pairs of two ports… *initiator* port and *target* port
- **Target port (upper-half of figure) is independent of rest of local control logic (provides transparency)**
- **Interface logic must be extremely efficient to not slow down RapidIO link**

Focus on Components: App-Specific Modules

Honeywell

- **Application development for FPGA done in HDL**
  - Xilinx ISE software used for design synthesis and generation
  - *Majority of FPGA design should remain fixed*
- **Ideally, user only needs to design co-processors**
  - See figure below, co-processors only small part of design
  - Developer can focus on application-related computation
  - Node controller allows control fabric and data path to remain fixed
- **Proposed co-processor module "wrapper" standard**



FPGA Architecture Control Fabric

Honeywell

- **“Node Controller” is critical part of FPGA architecture**
  - Provides control over data movement throughout device, as well as activity of computational modules, and other basic operations
  - Behaves very similar to software processor
    - **Much higher performance than PowerPC/Microblaze alternative**
    - **However, much simpler functionality = fewer features and capabilities, but also less “overhead”**
- **Software processor sends instruction “streams” to FPGA via RapidIO, FPGA executes instructions**
  - Examples of instructions include “move data from A to B,” “process X amount of data,” “take timestamp,” etc…
  - By sending “bundles” of instructions, individual network transactions are not required for software processor to verify each instruction
  - Loops, etc… permit continuous, self-contained operation of FPGA *without having to hard-code that particular behavior*

Performance Results – Serial RapidIO

Honeywell

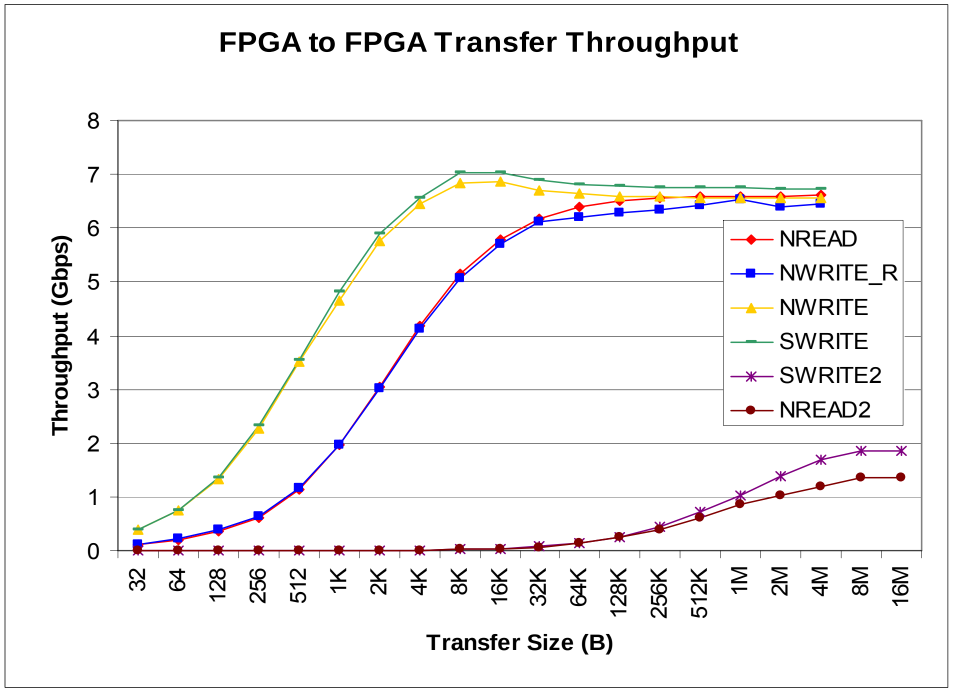
- **Basic throughput & latency results**
  - All basic I/O-Logical **transaction types**
  - All possible **source/destination combos**
  - Varied transfer size from **32 B to 16 MB**
- **Latency defined from perspective of initiating element (all data transferred at initiator)**
- **Throughput defined by dividing latency by time**
- **Assumes sequential memory access**
- **FPGA-to-FPGA**
  - Best-case throughput on SWRITE (7 Gbps, 93% of 7.52 Gbps max)
  - "Hump" seen due to DDR2 memory (hump occurs as xfer sizes cross rows)
- **AMC-to-FPGA**
  - Disappointingly low throughput for AMC-initiated RapidIO transactions
  - Only SWRITE and NREAD shown, to highlight the extremes

FPGA TO FPGA

• NREAD	- read operation
• NWRITE_R	- write with response
• NWRITE	- response-less write
• SWRITE	- response-less write

AMC TO FPGA

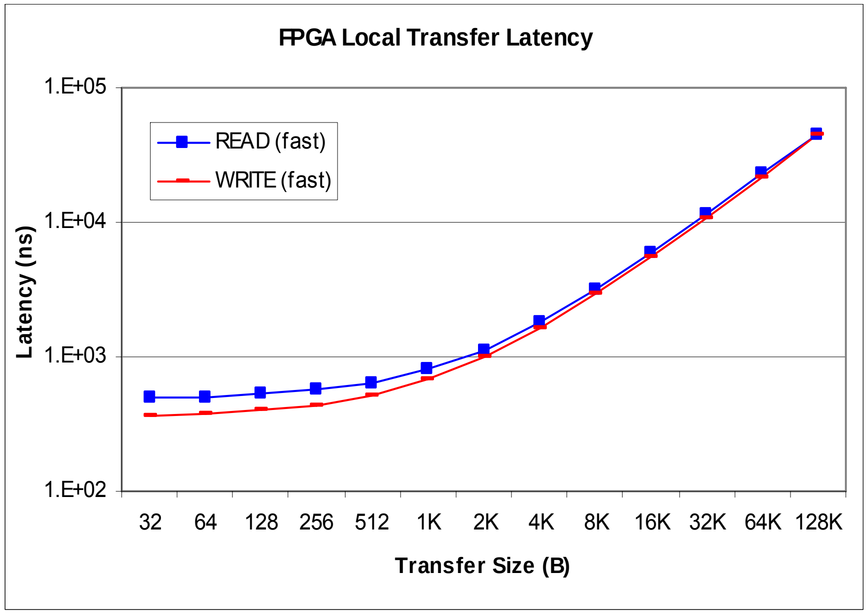
• SWRITE2	- response-less write
• NREAD2	- read operation

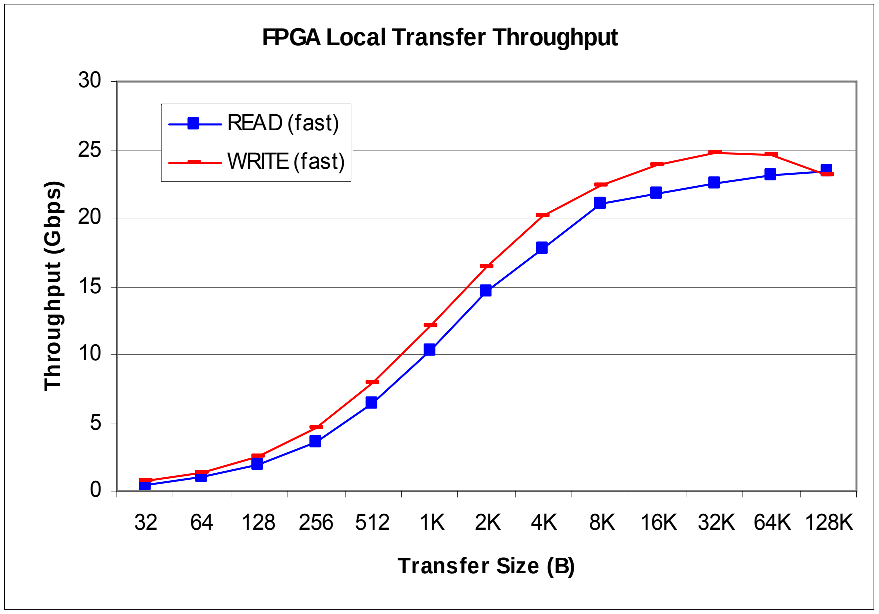


Performance Results – FPGA Local Memory

Honeywell

- **Local data transfers**
  - Movement of data between DDR2 and on-chip BlockRAM
  - Transfer sizes capped at 128 KB
  - Two types: read and write
- **Transfer latencies defined as time to move all data**
  - Best-case latency = ~500 ns
  - Worst-case latency = ~44 μs
  - Performance difference for smallest sizes due to read latency of DDR2 controller
- **Great throughput, up to nearly 25 Gbps!!**





Performance Results – FPGA Computation

Honeywell

- **Beyond data movement, what can the FPGA do?**
  - **“Instruction rate”**
    - Defined as maximum rate at which instructions can be executed by Node Controller
      - Good estimation of average time in between instructions
      - Measured by executing consecutive timestamp instructions
    - Max instruction rate = 7 clock cycles, or 56 ns (125 MHz clock)
  - **Illustration of sustained computational performance:**
    - Processing 600×800 video frames, in 20 chunks
    - Single frame: 480,000 B, 480 μs = 8 Gbps
    - 100 frames: 48,000,000 B, 1.431 s = 268 Mbps

Coarse-grained control critical for performance!!

Conclusions

Honeywell

- **Presented architecture and performance of experimental testbed**
  - FPGA architecture designed to maximize concurrency and efficiency
  - Serial RapidIO provides much-improved inter-device data throughput
- **Impressive sustained performance demonstrated**
  - ~6.5Gbps data throughput between FPGA devices
  - Up to 25Gbps data throughput between internal and external memories
    - **Can be sustained even with concurrent transmission of data over RapidIO link!**
- **Room for improvement and extension of control mechanism**
  - Support wider-range of instructions for FPGA
  - Prototype designed with radar/image processing in mind, consider other user applications (i.e. styles of computation modules)
- **For more information contact us at:**

- **Chris Conger - chrisley.conger@honeywell.com**
- **Andrew White - andrew.t.white@honeywell.com**
- **David Bueno - david.bueno@honeywell.com**