A Scalable Processor with Embedded Software for Large-Scale Scientific Applications

Daniel Alex Finkelstein and Haldun Hadimioglu
Polytechnic University, Brooklyn, NY, USA
Outline

1. Motivation/Goals and Related Work
2. Peripheral Context
3. Experimental Platform
4. Associative Streaming Memory Processor
5. Conclusions
Motivation & Goals

1. Address the memory wall problem
   • More intelligence in memory and storage usage
   • Target the CPU, main memory, and peripherals
   • Exploit regular scientific large-scale applications
     • For high-speed processing, reconfigurable fabric is needed
   • Use embedded RISC cores for ‘slow’ tasks and algorithms too difficult to directly map onto FPGA fabric.
Motivation & Goals

2. RISC + FPGA fabric combination
   - Schedules large & fine grained data movements
   - Associates data streams, data within streams, and mixes data streams on-chip or using latency-controlled peripherals (cache)

3. Scalability to handle large problems: Efficient Multi-Chip Configurations

4. Different mixture of processor-memory-fabric-peripheral composition
Related Work

1. Molen & Garp
   • Uses both FPGA reconfigurable logic and processor cores, but for application acceleration. Garp also allows FPGA direct access to main memory.

2. RAW
   • Programmable insofar as the instructions and data can be rerouted through the tiles via switching.

3. RAMP
   • Though a simulation environment, some of the (many) proposed features include dataflow architectures for programming languages, high-bandwidth peripherals, and reusable logic cores for the FPGA fabric.

4. RSVP
   • Decoupled operand prefetch, vector stream units, and detailed vector stream descriptors for media-rich applications, but can be generalized.
Supercomputing Applications

Our criteria:

• Large data sets
• Floating point operations
• Regular data structures
• SPEC CPU 2000 FP suite
  - Stable, predictable memory profiles
<table>
<thead>
<tr>
<th></th>
<th>Intel 975X Express Chipset</th>
<th>Intel IXP2855 Network Processor</th>
<th>Intel IXP2855 Network Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR DRAM</td>
<td>85.6 Gbps</td>
<td>57.6 Gbps</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>DDR2 DRAM</td>
<td>3 Gbps</td>
<td>RDRAM</td>
<td>Ethernet</td>
</tr>
<tr>
<td>SATA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Xilinx DDR DRAM</th>
<th>Xilinx DDR2 DRAM</th>
<th>Xilinx Rocket I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR DRAM</td>
<td>400 Mbps</td>
<td>667 Mbps</td>
<td>75 Gbps</td>
</tr>
<tr>
<td>DDR2 DRAM</td>
<td></td>
<td></td>
<td>Virtex-ll Pro</td>
</tr>
<tr>
<td>RDRAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet</td>
<td></td>
<td></td>
<td>Virtex-II Pro X</td>
</tr>
</tbody>
</table>

Peripherals are getting faster, but are usually separated from the processors by several levels of the memory hierarchy.
ML310 Development Board
ML310 Development Board

XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
ML310
Development Board

- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
- 256 MB DDR DRAM
ML310 Development Board

- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
- 256 MB DDR DRAM
- SanDisk
ML310 Development Board

- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
- 256 MB DDR DRAM
- SanDisk
- IDE interfaces
ML310
Development Board

- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
- 256 MB DDR DRAM
- SanDisk
- IDE interfaces
- PCI
ML310 Development Board

- XC2VP30 FPGA with 2 PowerPC 405 RISC Cores
- 256 MB DDR DRAM
- SanDisk
- IDE interfaces
- PCI
- Ethernet
PowerPC - Custom IP Interface
PowerPC - Custom IP Interface
PowerPC - Custom IP Interface
PowerPC - Custom IP Interface Memory Options
PowerPC - Custom IP Interface Memory Options

PowerPC Core

OCM (Block RAM)

PLB

PLB2OPB

OPB

custom IP

64 I

32 D

D

I

64

32
PowerPC - Custom IP Interface
Memory Options

- OCM (Block RAM)
  - 64 I
  - 32 D

- DDR DRAM
  - 32 64
  - 16 32 64

- PLB
  - D 64
  - I 64

- PLB2OPB
- OPB
- custom IP
- PowerPC Core
PowerPC: A High Level Controller
PowerPC: A High Level Controller

- PowerPC (PPC) core (with some hardware additions) can execute user programs
PowerPC: A High Level Controller

- PowerPC (PPC) core (with some hardware additions) can execute user programs
- PPC core may also perform control and monitoring functions for the processor system
High Level Controller Features

- Peripheral latency monitoring
- Peripheral data prefetch control
- Peripheral resource scheduling
- Coarse-grained data association, mixing, and buffering

- Store delays between data request and data arrival for each peripheral
- Begin retrieving data streams and process immediately or buffer locally, based on latency values
- Control access to peripherals
- Vectorized data streams allow the controller to associate streams with each other (dependencies), re-order data into memory buffers (useful in matrix-matrix multiplication and transposes), etc.
Low Level Controller Features

- The LLC is the Custom IP (shown earlier) connected to the OPB
- The LLC resides within the FPGA’s configurable logic blocks (CLBs)
- The LLC performs both computations on the data streams and fine-grained data mixing

Low level controller is unique for each application.

Reusable functional logic can be dynamically mapped onto CLBs at compile-time or run-time.

Control signals exchanged with HLC to indicate status of buffers, dout_rdy, etc.

Data element arithmetic operations (int/FP) performed at this level. Elements in streams may be remixed to satisfy output constraints, dependent instructions, etc.
HLC - LLC Interaction
Conclusions

• Fast peripherals need not be limited by intermediate controllers, buses, and operating systems.

• Controller logic can be integrated alongside traditional processor components in a single package.

• Intelligent use of memory peripherals on local buses reduces processing latencies.

• Embedded software saves resources better devoted to time-sensitive computation.
Acknowledgments

We wish to acknowledge the support of Xilinx in providing us with ML310 development boards, design tools, and technical support.

This work was supported in part by a research fellowship from the U.S. Department of Education GAANN.
Thank you.

contact: dfinke01@cis.poly.edu