Bridging the GPGPU-FPGA Efficiency Gap

Christopher W. Fletcher\textsuperscript{1,2}, Ilia Lebedev\textsuperscript{2}, Narges B. Asadi\textsuperscript{3}, Daniel R. Burke\textsuperscript{2}, John Wawrzynek\textsuperscript{2}

\textsuperscript{1}: M.I.T., \textsuperscript{2}: UC Berkeley, \textsuperscript{3}: Stanford
Design by µArch Template

**Idea:** FPGAs designs from architectural templates
Design by μArch Template

Idea: FPGAs designs from architectural templates
Idea: FPGAs designs from architectural templates

Design by μArch Template
Idea: FPGAs designs from architectural templates

Q: How do template-based designs on an FPGA compare to general purpose processors?
Cell Signaling Networks

Goal: Given flow cytometry data, learn the structure of cell signaling networks

- Flow Cytometry
  - “Raw” count of how many proteins, of each type, there are in a cell

- Cell Signaling Networks
  - Structures that model protein signaling pathways
FPGA & GPGPU (Top level)

- Once the application becomes compute-bound, FPGA \approx an application-specific GPGPU
FPGA & GPGPU (Top level)

- Once the application becomes compute-bound, FPGA ≈ an application-specific GPGPU
FPGA & GPGPU (Top level)

- Once the application becomes compute-bound, FPGA ≈ an application-specific GPGPU

Picture based on the Nvidia Fermi.
FPGA & GPGPU (Top level)

- Once the application becomes compute-bound, FPGA ≈ an application-specific GPGPU

Picture based on the Nvidia Fermi.
FPGA & GPGPU (Top level)

- Once the application becomes compute-bound,
  FPGA $\approx$ an application-specific GPGPU
FPGA & GPGPU (Top level)

- Once the application becomes compute-bound, FPGA ≈ an application-specific GPGPU
FPGA & GPGPU (Top level)

- Once the application becomes compute-bound, FPGA ≈ an application-specific GPGPU

Picture based on the Nvidia Fermi.
FPGA & GPGPU (per Node)

**Context**

- FPGA & GPGPU

**Architecture**

- FPGA
  - Control (local orders)
  - Interconnect Control (data)
  - INIT
  - Serialization
  - Threads
  - Data
  - Post processing
- GPGPU
  - Instruction Cache
  - Thread Block Scheduler
  - Dispatcher
  - Register File
  - Core
  - LD/ST
  - SFU
  - Interconnect Network
  - Shared L1 Cache

**Closing**

- February 28th
- FPGA 2011
- 14
FPGA & GPGPU (per Node)

FPGA

GPGPU

Control (local orders)

Interconnect Control (data)

INIT

Serialization

Threads

Data

Post processing

GPGPU Core

Dispatch Port

Operand Collector

Result Queue

Floating Point

Integer

LOG Table

En

Order Sampler

Graph Sampler

Data

From Previous Core

To Next Core GS

Local orders

FPGA Core

LD/ST

SFU

Register File

Dispatcher

Thread Block Scheduler

Instruction Cache

Register File

LD/ST

SFU

Instruction Cache

Thread Block Scheduler

Interconnect Network

Shared L1 Cache
Summary of Results

Our study compares the throughput of an FPGA core relative to a GPGPU core.

Example: A Virtex-5 core achieves 4.26─4.30x throughput, relative to a GT 330m core.

<table>
<thead>
<tr>
<th></th>
<th>Nvidia GT 330m (mobile)</th>
<th>Nvidia GTX 480</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtex 5, 155t</td>
<td>4.26 – 4.30x</td>
<td>3.00 – 3.30x</td>
</tr>
<tr>
<td>Virtex 6, 240t</td>
<td>4.00 – 4.36x</td>
<td>3.00 – 3.11x</td>
</tr>
</tbody>
</table>

Example: A Virtex-5 core achieves 4.26–4.30x throughput, relative to a GT 330m core.

Despite the GPGPU’s higher core clock frequency, the FPGA core maintains competitive throughput.
Acknowledgements

For making this work possible, we thank:

– NIH / Cancer Research Institute Grant #130826-02
– Members of the ...
  • Stanford Nolan Lab
  • Berkeley Reconfigurable Computing Group
  • M.I.T. Hornet Group
– UC Berkeley Wireless Research Center (BWRC)
– Contributors to the GateLib research library