Building Efficient, Reconfigurable Hardware using Hierarchical Interconnects

November 16, 2011

Cheng C. Wang

Advisor: Dejan Marković
Why Efficiency?

- **Represents design tradeoffs**
  - Balancing performance, power, and area

- **Energy Efficiency**
  - Quantifies *work* per unit of energy
  - Generally measured in billions of operations per milliwatt (GOPS/mW)
  - Translates to battery life, thermal limit, and reliability

- **Area Efficiency**
  - Quantifies *work* per unit of area
  - Generally measured in billions of operations per mm$^2$ (GOPS/mm$^2$)
  - Translates to size, cost, and yield
How Efficient are Today’s Chips?

ISSCC & VLSI 1999-2011, averaged

Average Energy Efficiency (GOPS/mW)

Average Area Efficiency (GOPS/mm²)

- \(100\)
- \(10\)
- \(1\)
- \(0.1\)
- \(0.01\)
- \(0.001\)
- \(0.0001\)

- Dedicated
- Prog. DSP
- FPGA
- \(\mu\)Proc
Efficiency vs. Flexibility

ISSCC & VLSI 1999-2011, averaged

Average Energy Efficiency (GOPS/mW)

Average Area Efficiency (GOPS/mm²)

Efficiency

Flexibility

Dedicated

Prog. DSP

FPGA

μProc

UCLA
Efficiency vs. Flexibility

ISSCC & VLSI 1999-2011, averaged

Can we have both?
Current Solution – Integrate Both!

- Use processor to control dedicated accelerators
  - Large portions of “dark silicon” – low area efficiency
  - Accelerators are designed for **fixed** standards

![NVIDIA Tegra 2 Diagram]

- **Dedicated DFE**
Keeping Up with the Standards

- Frequent change in standards = frequent re-design
  - New dedicated chips required at least once a year
  - Is this a scalable solution?

### Media Standards

<table>
<thead>
<tr>
<th>Year</th>
<th>Media Type</th>
<th>Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>Video</td>
<td>JPEG</td>
</tr>
<tr>
<td>1990</td>
<td>Video</td>
<td>H.261</td>
</tr>
<tr>
<td>1990</td>
<td>Video</td>
<td>MPEG-1</td>
</tr>
<tr>
<td>2000</td>
<td>Video</td>
<td>H.263</td>
</tr>
<tr>
<td>2000</td>
<td>Audio</td>
<td>WMV/VC1</td>
</tr>
<tr>
<td>2000</td>
<td>Audio</td>
<td>MPEG-4</td>
</tr>
<tr>
<td>2000</td>
<td>Audio</td>
<td>H.264</td>
</tr>
<tr>
<td>2010</td>
<td>Audio</td>
<td>H.264 MVC</td>
</tr>
<tr>
<td>2010</td>
<td>Audio</td>
<td>JPEG XR</td>
</tr>
<tr>
<td>2010</td>
<td>Audio</td>
<td>H.265</td>
</tr>
</tbody>
</table>

### Radio Standards

<table>
<thead>
<tr>
<th>Year</th>
<th>Radio Tech.</th>
<th>Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>Analog</td>
<td>NMT, AMPS, ETACS</td>
</tr>
<tr>
<td>1990</td>
<td>CDMA</td>
<td>IS-95, IS-2000</td>
</tr>
<tr>
<td>2000</td>
<td>TDMA</td>
<td>GSM, EDGE</td>
</tr>
<tr>
<td>2010</td>
<td>LTE</td>
<td>LTE, HSPA, EV-DO, HSPA+</td>
</tr>
<tr>
<td>2010</td>
<td>OFDMA</td>
<td>WLAN, 802.11a, 802.11g, 802.11n, 802.16e, 802.16m</td>
</tr>
<tr>
<td>2010</td>
<td>OFDMA</td>
<td>LTE-Adv.</td>
</tr>
<tr>
<td>2010</td>
<td>Wired</td>
<td>3G, 4G</td>
</tr>
</tbody>
</table>

UCLA
Cost of Chip Design

- Increasing cost with every generation
  - Increasing design complexity
  - Increasing man-hours per designs
  - Expensive CAD licenses

Total Development Cost ($M)

- Increasing cost with every generation
  - Increasing design complexity
  - Increasing man-hours per designs
  - Expensive CAD licenses

Source: Int'l Business Strategies & Altera

Need reconfigurable hardware!
Candidates for Reconfiguration

- **Average Area Efficiency (GOPS/mm²)**
- **Average Energy Efficiency (GOPS/mW)**

- **µProc**
- **FPGA**
- **Prog. DSP**
- **Dedicated**

- **Parallel HW**
- **High Throughput**

- **SW Controlled**
- **Low Throughput**
## Design Tradeoffs

<table>
<thead>
<tr>
<th>Efficiency (ASIC)</th>
<th>vs.</th>
<th>Flexibility (FPGA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Design, Physical Design</td>
<td>Logic Design only</td>
<td></td>
</tr>
<tr>
<td>Licenses: Synthesis, P&amp;R, etc.</td>
<td>Fewer Licenses</td>
<td></td>
</tr>
<tr>
<td>90nm or older</td>
<td>32nm or newer</td>
<td></td>
</tr>
<tr>
<td>2 – 4 months fabrication time</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Expensive to Design</td>
<td>Inexpensive to Design</td>
<td></td>
</tr>
<tr>
<td>Efficient Operation</td>
<td>Inefficient Operation</td>
<td></td>
</tr>
</tbody>
</table>
An Efficient Reconfigurable HW

Goal: narrow this gap

- µProc
- FPGA
- Programmed DSP
- Dedicated

Average Energy Efficiency (GOPS/mW)

Average Area Efficiency (GOPS/mm²)
Outline

- Understand FPGA inefficiency
- Current Work and Results
- Proposal: An Efficient Interconnect Architecture
- Next Steps
Why are FPGAs Inefficient?

- Compared to ASICs, FPGAs incur penalties in:
  - Area (17 – 54 x)
  - Speed (3 – 7 x)
  - Power (6 – 62 x)

- Main culprit: interconnect!

Current Architecture is not Scalable

2D-mesh interconnects
Current Architecture is not Scalable

Full connectivity impractical

Heuristically reduce switch-box array connectivity

$O(N^2)$ worst-case complexity
A Scalable Architecture

Beneš Network

Switch Matrix

uni-cast

multi-cast
A Scalable Architecture
Efficient for Large Scale

\(O(N \cdot \log N)\) complexity instead of \(O(N^2)\)
Efficient for Large Scale

But inefficient for local connections!
Hierarchical Interconnect Architecture

Folded Beneš Network

Unidirectional routing using Switch Matrix (SM)

Full connectivity

$O(N \cdot \log N)$ complexity

Hierarchical Interconnect Architecture

Unidirectional routing using Switch Matrix (SM)

Full connectivity

$O(N \cdot \log N)$ complexity

Routing congestion doubles for every hierarchy — $O(N)$

Mirrored Physical Architecture

Mirrored Physical Architecture

Required # of tracks
(\(\leftrightarrow\) = 2 unidirectional wires)
Mirrored Physical Architecture

Required # of tracks
(\longleftrightarrow = 2 \text{ unidirectional wires})

8 8
Mirrored Physical Architecture

Required # of tracks
(\[\leftrightarrow\] = 2 unidirectional wires)
Required # of tracks
( ↔ = 2 unidirectional wires)

8 8 16
Mirrored Physical Architecture

Required # of tracks
( $\rightarrow \rightarrow = 2$ unidirectional wires)

8  8  16
Mirrored Physical Architecture

Required # of tracks
(\(\leftarrow\rightarrow\) = 2 unidirectional wires)

Routing congestion doubles for every two hierarchies — \(O(\sqrt{N})\)
Most Recent Chip: A 2048-LUT FPGA

- SM Stage
- Full SM
- Half SM
- LUT 128
- N6:1
- N6:2
- N8:2
Most Recent Chip: A 2048-LUT FPGA

- Full SM
- Half SM

SM Stage

1 2 3 4 5 6 7 8 9 10

128 LUT

N6:1 N6:2

N8:2
Most Recent Chip: A 2048-LUT FPGA

SM Stage 1 2 3 4 5 6 7 8 9 10

☐ : Full SM
☐ : Half SM

N8:2
N6:1
N6:2

1 6 7 8 9 10

UCLA
Most Recent Chip: A 2048-LUT FPGA

- 4-LUT Logic CLBs: 256
- 4-LUT Logic + DSP CLBs: 224
- 8-LUT Block RAM (1kb) CLBs: 16
Logic CLB Design

- Four 3-/4-input LUTs
- Local carry chain
- Selectable FF/non-FF outputs
DSP CLB Design

- Support for 5-/6- input LUTs
- Configurable adder/subtractor (8b or 4b)
- Wallace tree multiplier (4b × 4b)
DSP CLB Modes

3–6 input LUT:
DSP CLB Modes

8b or two 4b adders:

- Out Stage A: C0, S0, C1, S1, C4, S4, C3, S3
- Out Stage B: C1, S1, C5, S5, C2, S2, C6, S6
- Out Stage C: C3, S3, C7, S7, C2, S2
- Out Stage D: CO0, CO2
Wallace tree multiplier:
configuration circuitry

- custom SRAM-based bit cell
  - BL and WL driven from scan chains

area: 1.4μm × 1.8μm
5× reduction over DFF
Logic : Interconnect = 1 : 1

- Logic Area
- Interconnect Area

2D-Mesh

This Work

3-4× interconnect area reduction

M. Lin, et al., FPGA 2006
Automated Mapper Flow

- Standard Cell Library
- DesignWare Library
- Logic Synthesis
- RTL
- Netlist
- Logic Optimization
- Place & Route
- Simulink Testbench
- Bitstream Creation
- Bitstream
- IBOB / MATLAB
- IBOB Board
- Hier. FPGA
- ZDOK+
- ZDOK+
- Commercial
- Custom

Hier.
FPGA
### Measurement from Mapped Designs

<table>
<thead>
<tr>
<th>Design</th>
<th>Resource Utilization</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Logic (256)</td>
<td>DSP (224)</td>
</tr>
<tr>
<td>175 L/DSP 16b Acc.</td>
<td>256</td>
<td>224</td>
</tr>
<tr>
<td>112 DSP 16b Acc.</td>
<td>4</td>
<td>224</td>
</tr>
<tr>
<td>32-tap 16b FIR</td>
<td>132</td>
<td>209</td>
</tr>
<tr>
<td>2x2 MIMO 64-pt FFT</td>
<td>196</td>
<td>93</td>
</tr>
</tbody>
</table>
Energy/Op vs. Delay

Power Ratio @ $F_{\text{max}}$
- Clock: 28%
- Active: 46%
- Leak: 26%

Power Ratio @ $E_{\text{min}}$
- Clock: 18%
- Leak: 44%
- Active: 38%

- 370MHz @ 1.0V
- 300MHz @ 0.88V
- 200MHz @ 0.74V
- 100MHz @ 0.59V
- 55MHz @ 0.5V
Comparison Against Prior Work

- **Flexible DSP**
  - 32nm custom
  - 50MHz @ 0.34V
  - [A. Agarwal]
  - 0.65

- **This Work**
  - 65nm synthesized
  - 55MHz @ 0.5V
  - 1.13

- **Commercial FPGA**
  - 40nm custom
  - 0.9V
  - [A. George]
  - 0.05

**Energy (pJ/Op)**

- **Flexible DSP**
  - 0.2
  - 0.4
  - 0.6
  - 0.8
  - 1.0
  - 1.2
  - 1.4
  - 1.6
  - 1.8
  - 2.0
  - 2.2
  - 2.4
  - 2.6
  - 2.8
  - 3.0

**Delay (ns)**

- 0
  - 5
  - 10
  - 15

** GOPS/mW**

- (16b op)

A. Agarwal, *et al.*, ISSCC 2010
A. George, *et al.*, IEEE Comp. in Sci. and Eng., 2011
Die Micrograph + Chip Summary

Technology: 65nm 1P9M CMOS
Core $V_{DD}$: 0.34 to 1.0V
Frequency: 40 to 400MHz
I/Os: 75 bidirectional
Core Size: 2.52mm × 1.56mm
Gate Count: 2.73M
CLB Count: 256 Logic, 224 DSP
Block RAMs: 16 128×8b
Config. Bits: 297,472b

[C. Wang, et al., VLSI 2011]
Where Are We Now?

- Average Area Efficiency (GOPS/mm^2)
- Average Energy Efficiency (GOPS/mW)

What’s Next?

- Our Work
- Dedicated
- Prog. DSP
- FPGA
- μProc
Many applications require >1 GOPS/mW efficiency
- Communication DSP in cell phones
- Neural DSP in implanted neural recordings
- Multimedia DSP in many portable devices

Standards/algorithms are evolving, but core blocks are always required!
- Comm. DSP: FFT, Complex Multiplier
- Neural DSP: Wavelet Transforms, FIR/IIR filters
- Multim. DSP: DCT, Wavelet Transform
Coarse-Grain Accelerators

- Many core blocks are control-heavy: leads to inefficiency

<table>
<thead>
<tr>
<th>Design</th>
<th>Result</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$V_{DD}$ (V)</td>
</tr>
<tr>
<td>175 L/DSP 16b Acc.</td>
<td>1.0</td>
<td>179</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>8.6</td>
</tr>
<tr>
<td>2x2 MIMO 64-pt FFT</td>
<td>1.0</td>
<td>82.7</td>
</tr>
<tr>
<td></td>
<td>0.78</td>
<td>26.5</td>
</tr>
</tbody>
</table>

- Proposed coarse-grain accelerators for communication
  - 8-2048 point reconfigurable FFT
  - 48x32-bit reconfigurable multiplier
  - Programmable wordlength and pipeline stages
Coarser Granularity in DSP CLB

- Additionally, merge Logic and DSP CLB

### LUT
- 4-input LUTs
- \( \text{inA} \rightarrow \text{lutA} \)
- \( \text{lutB} \rightarrow \text{lutD} \)
- \( \text{lutC} \rightarrow \text{lutD} \)

### DSP
- Input Swap Stage
- \( \text{inA[0]} \rightarrow \text{lutA[0]} \)
- \( \text{inA[1]} \rightarrow \text{lutA[1]} \)
- \( \text{inA[2]} \rightarrow \text{lutA[2]} \)
- \( \text{inA[3]} \rightarrow \text{lutA[3]} \)

### Output
- \( \text{lutD} \rightarrow \text{outD} \)
- \( \text{lutC} \rightarrow \text{outC} \)
- \( \text{lutB} \rightarrow \text{outB} \)
- \( \text{lutA} \rightarrow \text{outA} \)

- S/U Adder
- Subtractor
- \( \text{SUM[8:6]} \rightarrow \text{FF} \)
- \( \text{MULT[15:12]} \rightarrow \text{FF} \)

- S/U * S/U Multiplier
- \( \text{SUM[3:2]} \rightarrow \text{FF} \)
- \( \text{MULT[7:4]} \rightarrow \text{FF} \)

- \( \text{SUM[1:0]} \rightarrow \text{FF} \)
- \( \text{MULT[3:0]} \rightarrow \text{FF} \)
Current interconnect architectures
- Manually optimized
- 2 types of Switch Matrices
Goal: optimize interconnect through Place & Route
- Verify performance of various architectures
- Many types of Switch Matrices
Designing 5T-Bitcell

- Originally proposed for read stability
  - We want it for area savings

6T

5T

1.4µm

1.8µm

1.0µm
Dual Voltage Domains

- Keep SRAM at a higher VDD
  - Use HVT to minimize leakage
- Use scaled VDD for datapath of static mux
  - SRAM overdrives pass-transistor

VDD_H: 1.2V
VDD_L: 0.3 – 1V

Bit Cell (HVT)

Interconnect (LVT)
Looking into the Future: MEMS switch

- Build interconnect switch with MEMS relays
  - Low on resistance (< 1kΩ)
  - Slow, but irrelevant for static mux

- Build a Bit Cell in 4 relays!
Interconnect

CLB Logic

Integrating Relays

- Relays are fabricated with 2 metal layers
  - No Si substrate required
  - Can be “stacked” on top of CMOS logic

Potentially another 2x area reduction

Our Target

Average Energy Efficiency (GOPS/mW) vs. Average Area Efficiency (GOPS/mm²)

- Projected Work
- Dedicated
- Our Prior Work

- FPGA
- Prog. DSP
- μProc
Conclusion

- ASIC is not scalable
  - Highly efficient, but not reconfigurable
  - Increasing cost makes constant re-design infeasible

- Needed: An efficient, reconfigurable HW
  - Interconnect architecture: key to efficiency and flexibility
  - Demonstrated a 2048-LUT FPGA:
    - $3-4x$ lower interconnect area
    - $22x$ higher energy efficiency

- Next step: optimize DSP and interconnect designs
  - Circuit-level techniques and potential MEMS devices