Cisco Green Research Symposium
5 March 2008

FPGA-based ASIC
Design and Verification

Dejan Markovic
Electrical Engineering Department
University of California, Los Angeles
The Issues I am Going to Address

- **Power efficiency** = energy efficiency \( \sim C \cdot V^2 \)

- **Design complexity**

- **Design re-entry**
  - Algorithm (Matlab or C)
  - Fixed point description
  - RTL (behavioral, structural)
  - Test vectors for logic analysis

- **In this talk, I will demonstrate**
  - Power efficiency of 2.1GOPS/mW (90nm CMOS)
  - 70GOPS in 3.5mm\(^2\)
  - FPGA-based design and verification
Optimization Approach

- Power efficiency: circuit-level (C,V)
- Performance and area: architectural techniques
- Unified Simulink description

Automated environment for optimal hardware design and verification
Circuit-Level Optimization Framework

- **Sensitivity based optimization**
  - Balance sensitivity to all variables
  - Variables: gate size, $V_{dd}$, $V_{th}$

  $$\text{minimize } Energy(V_{dd}, V_{th}, W)$$

  $$\text{subject to } Delay(V_{dd}, V_{th}, W) \leq D_{con}$$

- **Constraints**
  $$V_{dd}^{\text{min}} < V_{dd} < V_{dd}^{\text{max}}$$
  $$V_{th}^{\text{min}} < V_{th} < V_{th}^{\text{max}}$$
  $$W^{\text{min}} < W$$

- **Reference design**
  - $D_{min}$ sizing @ $V_{dd}^{\text{max}}$, $V_{th}^{\text{ref}}$

**Goal:** find optimal $E$-$D$ tradeoff for a datapath
E-D space is the key for architecture optimization

[D. Markovic, V. Stojanovic, B. Nikolic, M.A. Horowitz, R.W. Brodersen, JSSC Aug’04]
Scaling Impacts Architecture

Process:
- L-Vt
- S-Vt
- H-Vt

Architecture:
- P2: parallel 2
- T2: time-mux 2

$E_{op} / E_{ref} @ V_{dd}^{max}$ vs. $T_{op} / T_{ref} @ V_{dd}^{max}$

Scaling
MDL to RTL conversion, automated P&R flow

Simulink to Silicon Mapping

Simulink
  Fix-pt lib
  MDL
  Custom tool 1
  ASIC backend
  Speed Power Area

[R. Davis et al., JSSC Mar’02]
Including FPGA Emulation

**XSG hardware library, RTL translation scripts**

```
Simulink
  Hw lib

RTL

Custom tool 2

FPGA backend

ASIC backend

Speed Power Area

ASIC and FPGA are I/O equivalent
```

[K. Kuusilinna et al., book chap. in SoC Revolution, KAP 2003]
Closing the Loop: I/O Verification

I/O hardware library, automated FPGA flow

[Simulink]

[Hw lib]

[Custom tool 2]

[Custom tool 3]

[I/O lib]

[RTL]

[FPGA backend]

[ASIC backend]

[Speed Power Area]

FPGA implements ASIC logic analysis

[D. Markovic, C. Chang, B. Richards, H. So, B. Nikolic, R.W. Brodersen, CICC’07]
Design Approach

- Unified Simulink design environment
  - Enter design once!
  - Algorithm verification
  - Macro-architecture
  - FPGA based ASIC debug

- Hardware-equivalent Simulink blocks
  - Add, mult, shift, mux…
    - Word-size, latency
Block Characterization

Library blocks / macros synthesized @ $V_{DD}^{ref}$

Latency

Cycle Time

Energy

Pipeline logic scaling
FO4 inv simulation

Speed
Power
Area

$T_{Clk} @ V_{DD}^{ref}$

mult

$V_{DD}$ scaling

gate sizing

$T_{Clk} @ V_{DD}^{opt}$

Goal: balanced logic depth and E/D sensitivity
Methodology for Architecture Selection

- Energy-Area-Delay space for architecture comparison
  - Time-mux, parallelism, pipelining, $V_{DD}$ scaling, sizing…

![Diagram showing energy-area-delay trade-off with blocks for Block-level and Datapath, highlighting initial and optimal designs with different techniques like time-mux, parallel, pipeline, gate sizing, int, fold, and optimal design with $V_{DD}$ scaling.]
Example: 4x4 SVD Algorithm

- This complexity is hard to optimize in RTL
  - 270 adders, 370 multipliers, 8 sqrt, 8 div
  - Recursive LMS-based algorithm (nested feedback loops)

\[
\begin{align*}
\mathbf{w}_i(k) &= \mathbf{w}_i(k-1) + \mu_i \cdot [\mathbf{y}_i(k) \cdot \mathbf{y}_i^\dagger(k) \cdot \mathbf{w}_i(k-1) - \sigma_i^2(k-1) \cdot \mathbf{w}_i(k-1)] \\
\sigma_i^2(k) &= \mathbf{w}_i^\dagger(k) \cdot \mathbf{w}_i(k) \\
u_i(k) &= \mathbf{w}_i(k) / \sqrt{\sigma_i^2(k)}
\end{align*}
\]

(i = 1, 2, 3, 4)

\[
\begin{align*}
\mathbf{y}_{i+1}(k) &= \mathbf{y}_i(k) - [\mathbf{w}_i^\dagger(k) \cdot \mathbf{y}_i(k) \cdot \mathbf{w}_i(k)] / \sigma_i^2(k)
\end{align*}
\]
**Energy/Area Optimization**

- **Starting point:** fixed architecture

![Diagram with energy, area, and delay axes, showing improvements through interlacing and folding with factors 13.8x and 2.6x, leading to a 16-bit design.](image)
Step 1: Word-length optimization

Energy/Area Optimization

Interl. 13.8x  
Fold 2.6x

30%  

16b design

word-size 30%

Area

Delay
**Step 2:** Gate size & $V_{DD}$ optimization

- **Interl.** 13.8x
- **Fold** 2.6x
- **Initial synthesis**
  - word-size 30%
  - sizing 40%
- **16b design**
  - energy savings 30%
Step 2: Gate size & $V_{DD}$ optimization

- Interl. 13.8x
- Fold 2.6x
- 30%
- 20%
- Final design

- 16b design
- word-size 30%
- Initial synthesis sizing 40%
- Optim. $V_{DD}$, W

$V_{DD}$ scaling
Hardware Results

- Result of Energy-Area-Performance Optimization

- **2.1 GOPS/mW**
  - 70 GOPS @ 100MHz
  - Power = 34mW

- **20 GOPS/mm²**
  - 70 GOPS in 3.5mm²

- Functional test was performed with FPGA

**Comparison with ISSCC chips**

[D. Markovic, B. Nikolic, R.W. Brodersen, JSSC Apr’07]
Goal: use Simulink testbench (TB) for ASIC verification

- Develop custom interface blocks (I/O)
- Place I/O and ASIC RTL into TB model

*Simulink implicitly provides the testbench*
Simulink I/O Test Model for the SVD

Emulation-based ASIC I/O test
Experimental Setup

- FPGA board
- ASIC board
- GPIO

Real-time at-speed ASIC verification
4x4 MIMO channel tracking

Theoretical and hardware performance comparison for Eigenvalues

- $\sigma_1^2$
- $\sigma_2^2$
- $\sigma_3^2$
- $\sigma_4^2$

Number of Symbols [k]

Up to 10 b/s/Hz with adaptive PSK
From Simulink to Optimized Hardware

Direct mapped DFG → Scheduler → Architecture Solutions → Hardware
(Simulink) (C++ / MOSEK) (Simulink/SynDSP) (FPGA/ASIC)

Automated Architecture Generation Flow

Reference Direct-mapping

ILP Scheduling & Bellman-Ford Retiming: optimal + reduced CPU time
Each point on the surface is an optimal architecture automatically generated in Simulink after modified ILP scheduling and retiming.

System designer can choose from feasible optimal solutions.

It is not just about functionality, but how good a solution is, and how many alternatives exist.
Conclusions

- Simulink provides level of abstraction needed for complete ASIC development
  - Hardware emulation of algorithms
  - Technology-driven architecture selection
  - FPGA-based ASIC verification
    - Logic analysis can be fully ported onto FPGA

- Energy-area-delay space is a compact way for comparing multiple architectural realizations
  - ILP-based formulation automates architecture design

- Complex algorithms in 90nm can achieve
  - 2.1 GOPS/mW, 20 GOPS/mm²
References

**ASIC design and verification**


**More publications available online**

- www.ee.ucla.edu/~dejan
Acknowledgments

- **Funding support**
  - C2S2 Focus Center Research Program, contract 2003-CT-888

- **Infrastructure support**
  - ST Microelectronics, Xilinx (hardware)
  - Synplicity, Synopsys, Cadence (software)