EE216B: VLSI Signal Processing

Introduction

Prof. Dejan Marković
ee216b@gmail.com

EE216B Elevator Pitch

Area/energy-efficient mapping
of advanced DSP algorithms
to hardware
### Background?

**Familiarity with**

- Digital ICs
- VLSI design
- Signal processing

### What is This Course About?

<table>
<thead>
<tr>
<th>Algorithm Modeling</th>
<th>Simulink/XSG Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- bit-true cycle-accurate</td>
</tr>
<tr>
<td></td>
<td>- hw-equivalent blocks</td>
</tr>
<tr>
<td></td>
<td>- target: FPGA or ASIC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Signal Proc. Architectures</th>
<th>Min Energy &amp; Area</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- interleaving, folding</td>
</tr>
<tr>
<td></td>
<td>- iterative sqrt/div</td>
</tr>
<tr>
<td></td>
<td>- loop retiming</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Circuit Optimization</th>
<th>Opt Energy-Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- parallelism, time-mux</td>
</tr>
<tr>
<td></td>
<td>- circuit topology</td>
</tr>
<tr>
<td></td>
<td>- $V_{dd}$, $V_{th}$, gate size</td>
</tr>
</tbody>
</table>

*Complex DSP*
Course Objectives

- The implementation of signal processing systems in CMOS technology
- To understand the issues involved in the design of signal processing systems

DSP Chip Design Challenges

- Power-limited performance
- More flexibility (multi-mode, multi-standard)
- Algorithm and hardware design are separate
- Increasing computational complexity
**Course Outcomes**

Systematic methodology for:
algorithm specification,
architecture mapping, and
hardware optimizations

- **Outcome 1:** hardware-friendly algorithm development
- **Outcome 2:** optimized hardware implementation

---

**Course Highlights**

- A design methodology starting from a high-level description to an implementation optimized for performance, power and area

- Unified description of algorithm and hardware parameters
  - Methodology for automated wordlength reduction
  - Automated exploration of many architectural solutions
  - Design flow for FPGA and custom hardware including chip verification

- Examples to show wide throughput range (kS/s to GS/s)
  - Outcomes: energy/area optimal design, technology portability

- Online resources: examples, references, tutorials etc.
Create a wiki account using your UCLA username
Course Material

- Lecture notes
- CAD tutorials
- Class project
- Selected papers from IEEExplorenow available online at [http://ieeexplore.ieee.org](http://ieeexplore.ieee.org)

Books

- **Textbook:** DSP Architecture Design Essentials
  - A free draft available online

- **Supplemental books (not required)**
  - And a few other books (see course wiki)
Material Based on a Book

- To be published 2012
  - Hard copy
  - eBook formats
  - Supplemental online material

**Course/Book Development**

- **Over 15 years of effort and revisions...**
  - Course material from UC Berkeley (Communication Signal Processing, EE225C), ~1995-2003
    - Prof. Robert W. Brodersen, Jan M. Rabaey, Borivoje Nikolić
  - The concepts were applied and expanded by researchers from the Berkeley Wireless Research Center (BWRC), 2000-2006
    - W. Rhett Davis, Chen Chang, Changchun Shi, Hayden So, Brian Richards, Dejan Marković
  - UCLA course (VLSI Signal Processing, EE216B), 2006-2008
    - Prof. Dejan Marković
  - The concepts expanded by researchers from UCLA, 2006-2010
    - Sarah Gibson, Vaibhav Karkare, Rashmi Nanda, Cheng C. Wang, Chia-Hsiang Yang

- **All of this is integrated into the course/book**
  - Lots of practical ideas and working examples
Chip Examples: Energy-Efficient DSP Kernels

DSP architecture optimization methodology

![Diagram showing energy-efficiency comparison between different chip examples.]

Organization

- The material is organized into four parts

1. Technology Metrics
   - Performance, area, energy tradeoffs and their implication on architecture design

2. DSP Operations & Their Architecture
   - Number representation, fixed-point, basic operations (direct, iterative) & their architecture

3. Architecture Modeling & Optimized Implementation
   - Data-flow graph model, high-level scheduling and retiming, quantization, design flow

4. Design Examples: GHz to kHz
   - Radio baseband DSP, parallel data processing (MIMO, neural spikes), architecture flexibility
Part 1: Technology Metrics

**Ch 1: Energy and Delay Models**
Energy and delay models of logic gates as a function of gate size and voltage... are used to formulate sensitivity optimization, result: energy-delay plots

**Ch 2: Circuit Optimization**

\[
S_A = \frac{\partial E}{\partial A}, \quad S_B = \frac{\partial D}{\partial A}
\]

\[A = A_0, \quad S_A = f(A_0, B), \quad f(A, B_0)\]

\[D_0 = f(A_0, B_0)\]

Part 2: DSP Operations and Their Architecture

**Ch 5: Arithmetic for DSP**
Number representation, quantization modes, fixed-point arithmetic

**Ch 6: CORDIC, Divider, Square Root**
Iterative DSP algorithms for standard ops, convergence analysis, the choice of initial condition

**Ch 7: Digital Filters**
Direct and recursive digital filters, direct and transposed, pipelined...

**Ch 8: Time-Frequency (multi-rate filters)**
FFT and wavelets (multi-rate filters)

Ch 9: Data-Flow Graph Model

DFG model is used for architecture transformations based on high-level scheduling and retiming, an automated GUI tool is built...

Ch 10: Wordlength Optimization

Example: 1/sqrt()

Automated wordlength selection

Ch 11: Architectural Optimization

Data-flow graph G

Ch 12: Simulink-Hardware Flow

Matrix A for graph G

Part 4: Design Examples: GHz to kHz

Ch 13: Multi-GHz Radio DSP

High-speed (GHz+) digital filtering

Ch 14: Dedicated MHz-rate Decoders

Adaptive channel gain tracking, parallel data processing (SVD)

Ch 15: Flexible MHz-rate Decoders

Increased number of antennas, added flexibility for multi-mode operation
**Additional Design Examples**

- Integrated circuits for future radio and healthcare devices
  - 4 orders of magnitude in speed: kHz (neural) to GHz (radio)
  - 3 orders of magnitude in power: μW/mm² to mW/mm²

**16-ch Neural-spike Clustering**

- Action Potentials
- Recorded Signal
- Spike Sorting
- Sorted Spikes

**200MHz Cognitive Radio Spectrum Sensing**

- Interference
- Spectrum Sensing
- LTE compliant

**Multi-core 8x8 MIMO Sphere Decoder**

- 4 mW/mm²
- LTE compliant

**Class Topics**

- **Circuit and DSP basics**
  - Circuit and architecture techniques
  - Scheduling and retiming
- **Arithmetic for DSP**
- **Tools: Matlab/Simulink, Synphony HLS**
- **Building blocks**
  - Filters, time-frequency analysis, DSP kernels
- **Systems**
  - Communications baseband
  - Biomedical sensors
  - Multimedia
Design Trajectory: From DSP Theory...

Digital Signal Processing

Harry Nyquist
Alan Oppenheim
Jean Baptiste Fourier

Sample & Quantize
Audio Video Radar
Add Multiply Memory

...to Optimized Hardware Realization

* Design, Optimization, Verification in Matlab/Simulink

Automated environment for hardware design and verification

Macro Arch.  E & A
Micro Arch.  E
Circuit  E

optimization  hardware design  I/O verification

ASIC
FPGA
Class Organization

- 4 homework assignments
- 1 term-long design project
- Midterm
- Final

EE216B Weekly Schedule

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>OH 56-147E Eng-4</td>
<td>Lecture 8500 BH</td>
<td>Lecture 8500 BH</td>
<td>OH 56-147E Eng-4</td>
<td></td>
</tr>
</tbody>
</table>

Instructor Info:
Dejan Marković / ee216b@gmail.com
56-147E Eng-IV / Tel: 310-825-8656
Grading Policy and Timeline

- Homeworks: 20%
- Midterm: 25%
- Project: 30%
- Final: 25%

Homeworks and Project

- **Bi-weekly homeworks (4 assignments)**
  - Implement individual DSP blocks

- **Final project: a DSP system**
  - Work in teams of two (if > 2, we need to talk)
  - Phase 1: proposal
  - Phase 2: mid-term report
  - Presentation + 4-page report
**EE216B Design Flow**

- **DSP algorithm**
- **Timed dataflow**
  - SysGen
  - Synplify
  - B-box
  - HDL
- **Architectural Transformations**
- **FPGA backend**
- **ASIC backend**
- **Hardware co-simulation**

**Software Environment: Big Picture**

- **Windows/Linux**
  - Algorithm description (Matlab/Simulink)
  - Architecture transformations (Simulink/C++)
  - RTL description
  - FPGA hardware emulation (XUP, BEE2)
  - Chip synthesis Retimeing, P&R (Cadence)
- **Linux**
  - Circuit design introductory (Cadence)
  - Circuit design advanced (Cadence)

**Tools and Environments**

- 216B
  - DSP + Comm.
  - Windows/Linux
- 215A
- 215B
- 215C
- 216A
- 216B
- 216E
XUP Virtex-II Pro Based FPGA Board

- You can borrow this board if you’d like (first-come first-serve)

### Resources

- 14k slices (~0.5M gates)
- 136 mults
- 2448Kb BRAM

---

**The Basic Problem**

<table>
<thead>
<tr>
<th>Algorithm designers</th>
<th>Chip designers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shannon limit, Raleigh fading, cyclostationary process</td>
<td>? $^<em>$#$^</em>$E($W^<em>$$^</em>$) $^<em>$#$^</em>$E($W^<em>$$^</em>$) ?</td>
</tr>
<tr>
<td>$^<em>$#$^</em>$E($W^<em>$$^</em>$) ?</td>
<td>Gate delay, leakage power number of bits, latency</td>
</tr>
</tbody>
</table>

- Very constrained implementation choices
- Design reentry (Matlab/C, HDL)
Proposed Approach

- **Unified Simulink environment**
  - Enter design only once!
  - Algorithm verification / emulation
  - Abstract view of architecture
  - FPGA based ASIC debug

- **Hardware-equivalent blocks**
  - Basic operators
    - Add, multiply, shift, mux...
  - Implementation constraints
    - Word-size, latency

Hardware Libraries

- **Xilinx System Generator**
- **Synphony HLS**
XSG Model Example: Iterative 1/sqrt()

\[ x_s(k+1) = \frac{x_s(k)}{2 \cdot (3 - Z \cdot x_s^2(k))} \]

- User defined parameters
  - Data type
  - Wordlength
  - Quantization
  - Overflow
  - Latency
  - Sample period

Block Characterization

Library blocks / macros synthesized @ \( V_{DD}^{ref} \)

Cycle Time

Pipeline logic scaling

FO4 inv simulation

Speed

Power

Area

Latency

Energy

\( V_{DD} \) scaling

gate sizing

\( T_{Cik} @ V_{DD}^{opt} \)
ASIC Synthesis

10,000 FPGA slices \( \Leftrightarrow \) 1mm\(^2\) (90nm CMOS)

Energy-Area-Performance Mapping

- Each point is an architecture automatically generated in Simulink using scheduling and retiming

[Rashmi Nanda]
New Trend: Parallel Data Processing

- Power limited technology scaling
  - Increased impact of process variations
  - More leakage power, multiple threshold devices

- Single dimensional $\rightarrow$ Multidimensional data

Multi-core Processors | MIMO Communications | Neuroscience

IBM / Sony / Toshiba | Belkin | www.sci.utah.edu

Energy-Delay Tradeoff

- Processors
  - Maximize performance
  - Highest $V_{DD}$ required

- Communications
  - Minimize energy & area
  - Typically, sensitivity $\sim 1$

- Neuroscience
  - Power density: $0.8$ mW/mm$^2$
  - Aggressive $V_{DD}$ scaling
Parallel Data in Neuroscience


Animal Models

Observation of brain injured vs. naïve rat pups

Main probe locations
Microelectrodes

64 site nanoprobe  Tungsten electrode

Micromachined probes

*Courtesy: S. Masmanidis*

---

Monitoring of Freely-behaving Animals

- Exploration of enriched environment:
  Brain injured vs. naïve pups

**Inhabitants**

naïve  rich environment

injured  social

**WiFi Cage**

level 1  level 2  level 3

4 ft (1.22 m)  4 ft  4 ft
Summary: Focus of This Course

3 components of the design problem

- **Algorithm specification** – Matlab (or C)
  - Floating point, implementation independent, system simulation

- **Architecture mapping**
  - Simulink for data flow
  - Stateflow for control

- **Hardware optimizations**
  - Real-time emulation
  - FPGA/ASIC implementation