UCLA Electrical Engineering

Spring 2024: ECE216B

VLSI Signal Processing

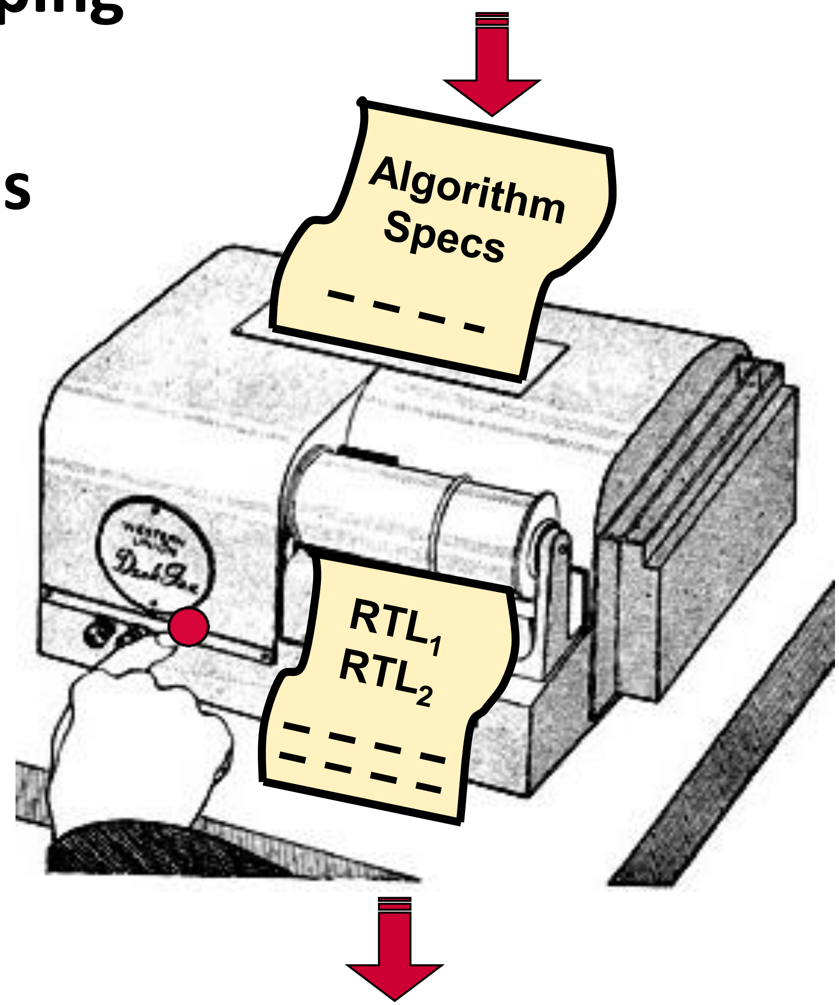
Prof. Dejan Marković
ee216b@gmail.com

Elevator Pitch

Area/energy-efficient mapping

of advanced DSP algorithms

to hardware



Background?

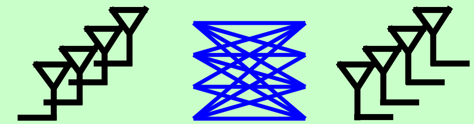
Familiarity with

- **Digital ICs**
- **VLSI design**
- **Signal processing**

What is This Course About?

Algorithm Modeling

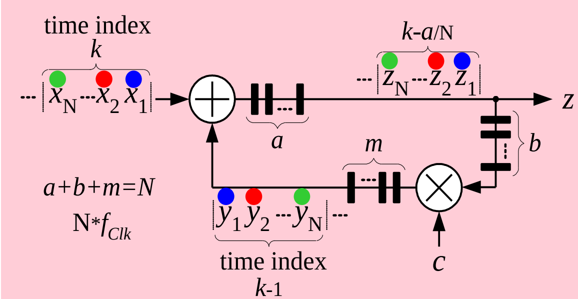
High-level Model
- bit-true cycle-accurate
- hw-equivalent blocks
- target: FPGA or ASIC



Complex DSP

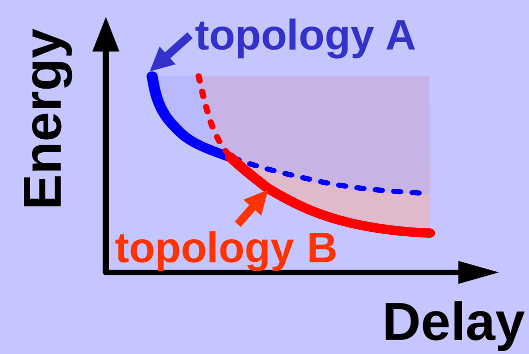
Signal Proc. Architectures

Min Energy & Area
- interleaving, folding
- iterative sqrt/div
- loop retiming



Circuit Optimization

Opt Energy-Delay
- parallelism, time-mux
- circuit topology
- Vdd, Vth, gate size



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**
- **Flexibility (multi-mode, multi-standard)**
- **Separate algorithm & hardware design**
- **Increasing computational complexity**

Course Outcomes

Systematic methodology for:
algorithm modeling,
architecture exploration,
and hardware optimizations

1 • **Hardware-friendly algorithm development**

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**
  - 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.**

Course Material

- **Lecture notes**
- **Homework**
- **CAD tutorials**
- **Class project**
- **Selected papers from IEEExplore (http://ieeexplore.ieee.org)**

Books

- **Textbook: DSP Architecture Design Essentials**
  - Not required

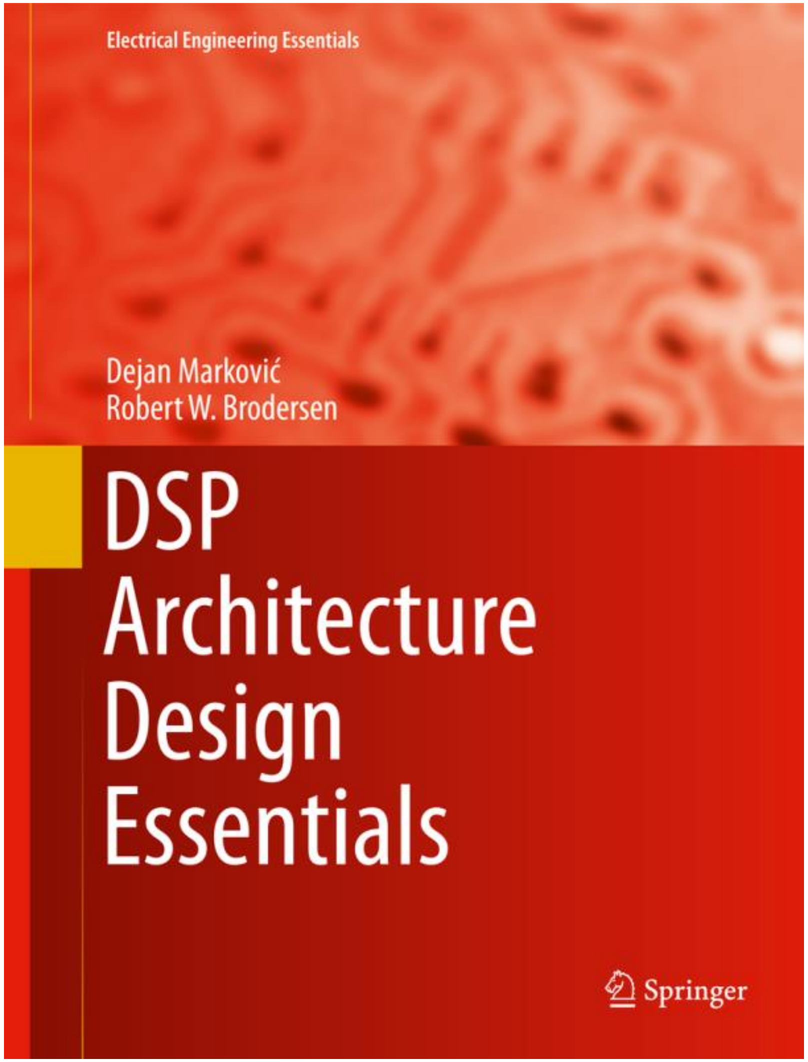
- **Supplemental books**
  - Oppenheim, Schafer, "Discrete-Time Signal Processing," Prentice Hall (1999)
  - K. Parhi, "VLSI Digital Signal Processing Systems: Design and Implementation," Wiley (1999)
  - Rabaey, Nikolic, Chandrakasan, "Digital Integrated Circuits: A Design Perspective," Prentice Hall (2003)

Textbook

Springer

July 2012

extra online materials

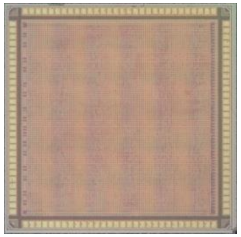


Course/Book Development

- **Over 15 years of effort and revisions…**
  - Course material from UC Berkeley (Communication Signal Processing, EE225C), ~1995-2003
    - Profs. 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), 2007-2012
    - Prof. Dejan Marković
  - The concepts expanded by researchers from UCLA, 2006-2013
    - Sarah Gibson, Vaibhav Karkare, Rashmi Nanda, Cheng C. Wang, Chia-Hsiang Yang, Tsung-Han Yu, Fang-Li Yuan
- **The course/book is based on the above material**
  - Lots of practical ideas and working examples

Energy-Efficient DSP Chips

4x4 SVD

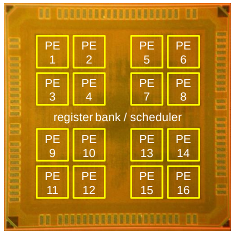


[VLSI'06]

2 GOPS/mW

100 MS/s

16x16 SD

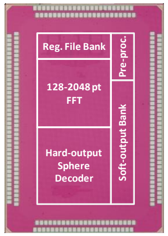


[ESSCIRC'09]

17 GOPS/mW

256 MS/s

8x8 SD



[VLSI'10]

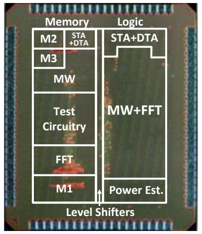
10 GOPS/mW

160 MS/s

DSP architecture optimization

examples

Cogno

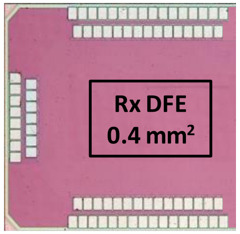


[VLSI'11]

5 GOPS/mW

200 MS/s

RxDFE

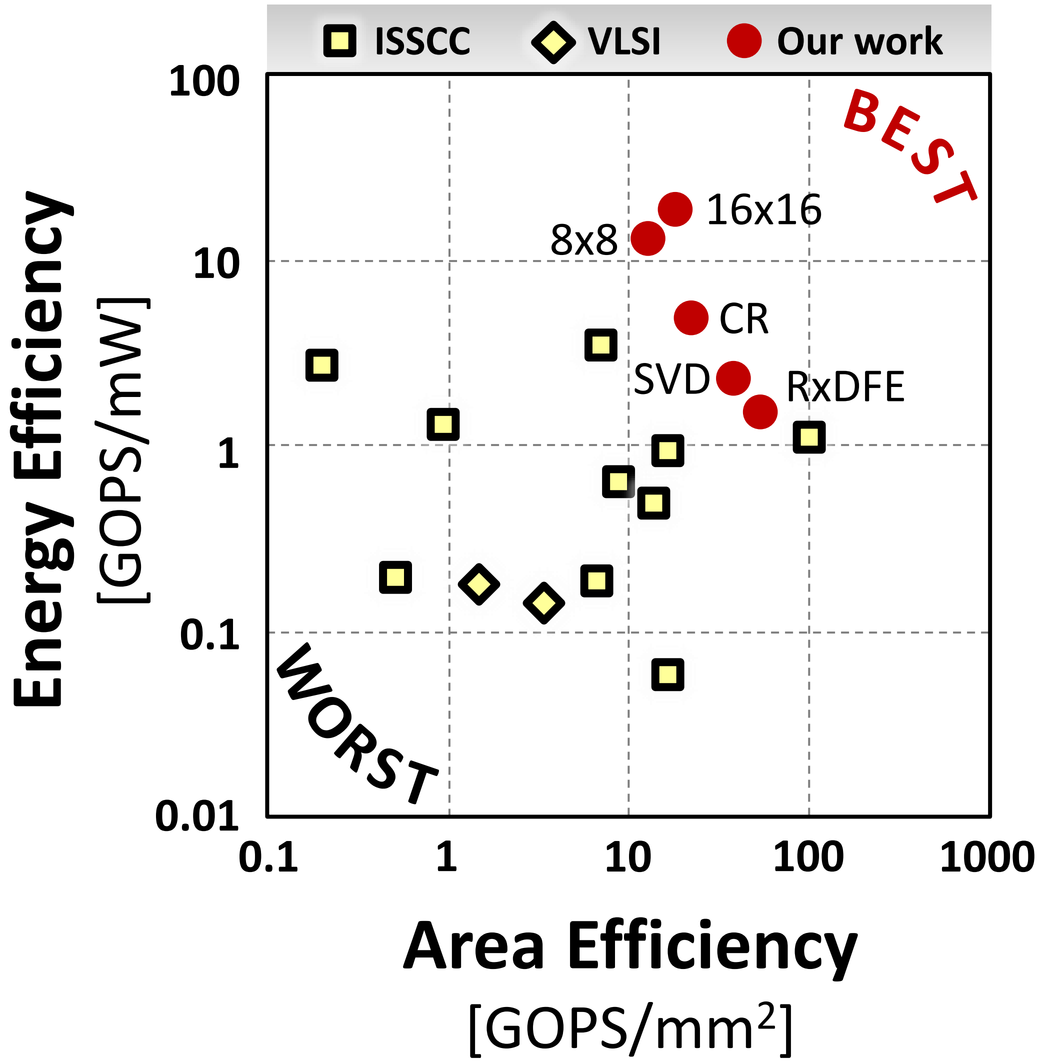


[A-SSCC'11]

12 GOPS/mW

3.6 GS/s

Efficiency Comparison



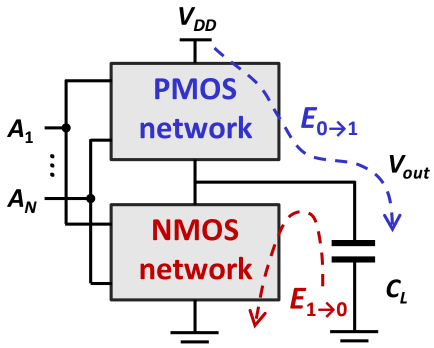
Organization

- **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

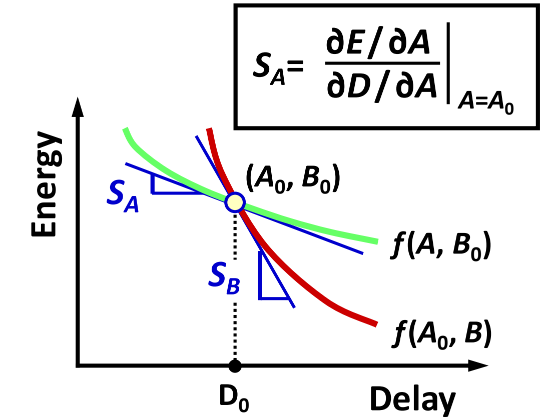
Ch 1: Energy and Delay Models



Energy and delay models of logic gates as a function of gate size and voltage…

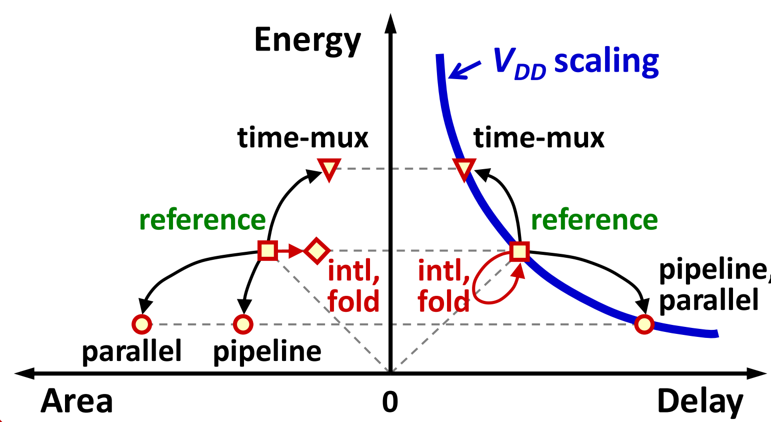
are used to formulate sensitivity optimization, result: energy-delay plots

Ch 2: Circuit Optimization

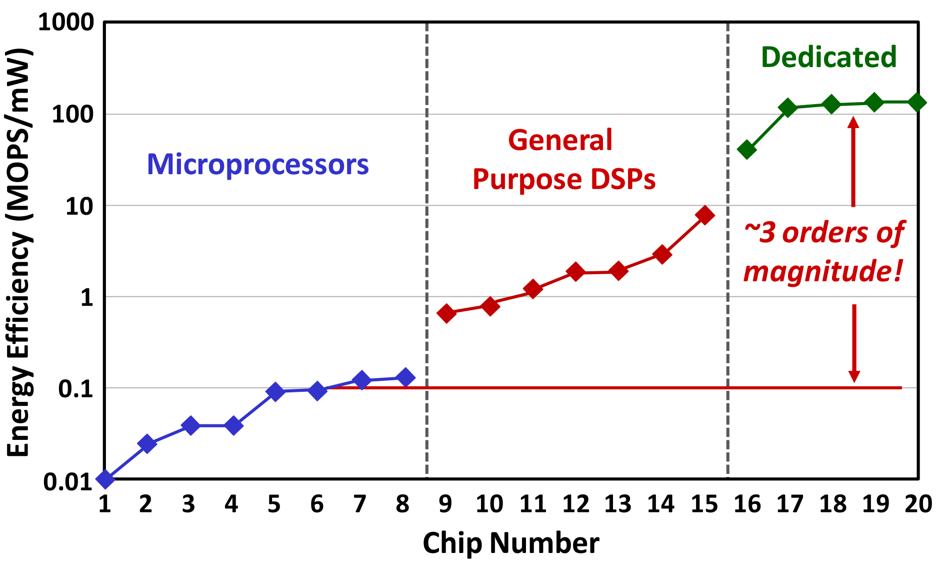


Extension to architecture tradeoff analysis…

Ch 3: Architecture Techniques

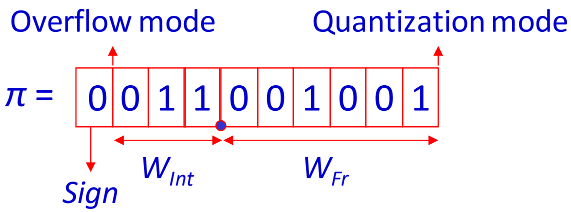


Ch 4: Architecture Flexibility



Part 2: DSP Operations and Their Architecture

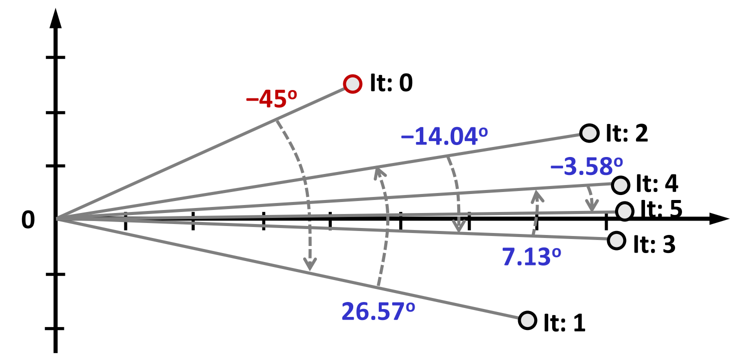
Ch 5: Arithmetic for DSP



Number representation, quantization modes, fixed-point arithmetic

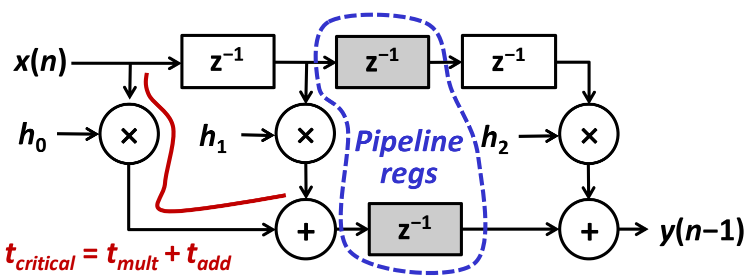
Iterative DSP algorithms for standard ops, convergence analysis, the choice of initial condition

Ch 6: CORDIC, Divider, Square Root



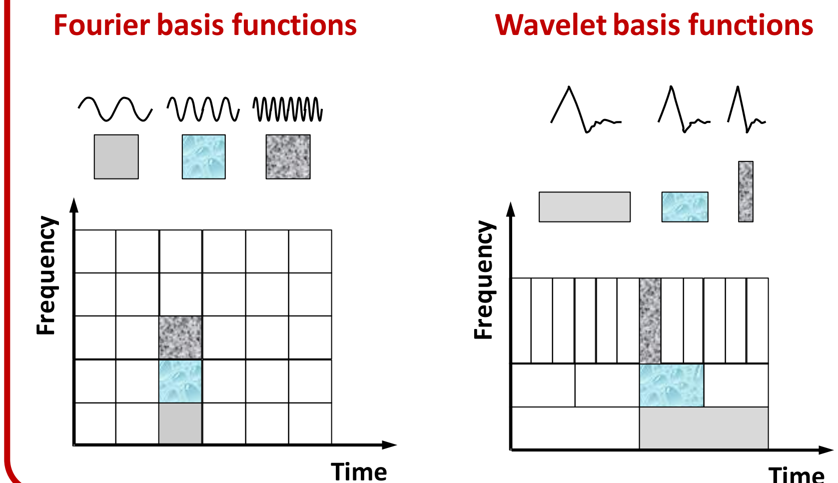
Direct and recursive digital filters, direct and transposed, pipelined…

Ch 7: Digital Filters



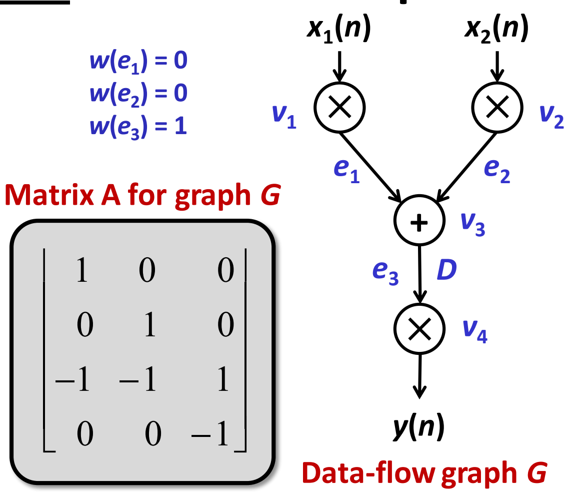
FFT and wavelets (multi-rate filters)

Ch 8: Time-Frequency Analysis



Part 3: Architecture Model & Optimization

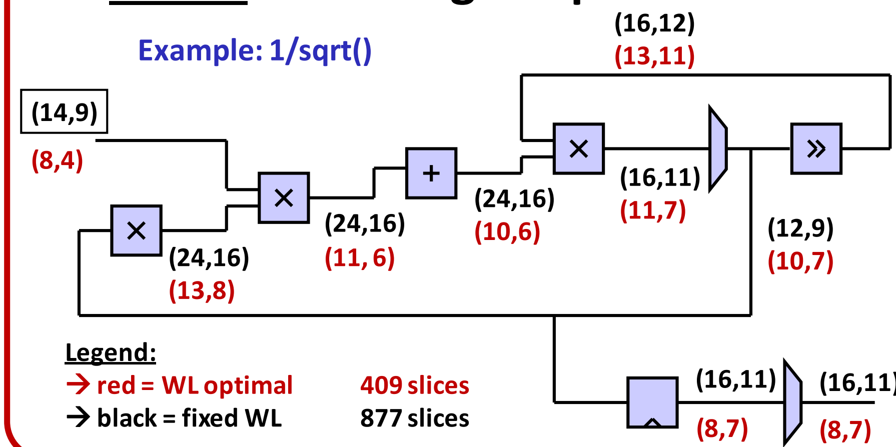
Ch 9: Data-Flow Graph Model



$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

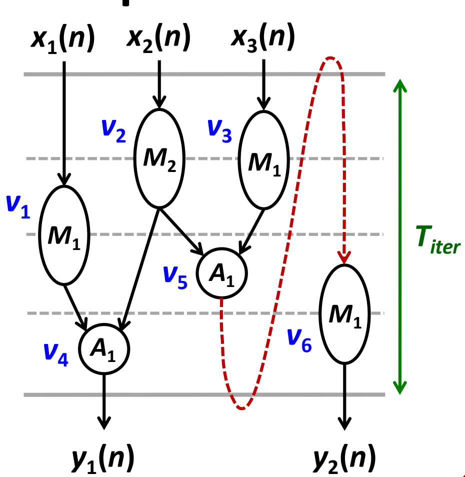
Data-flow graph G

Ch 10: Wordlength Optimization



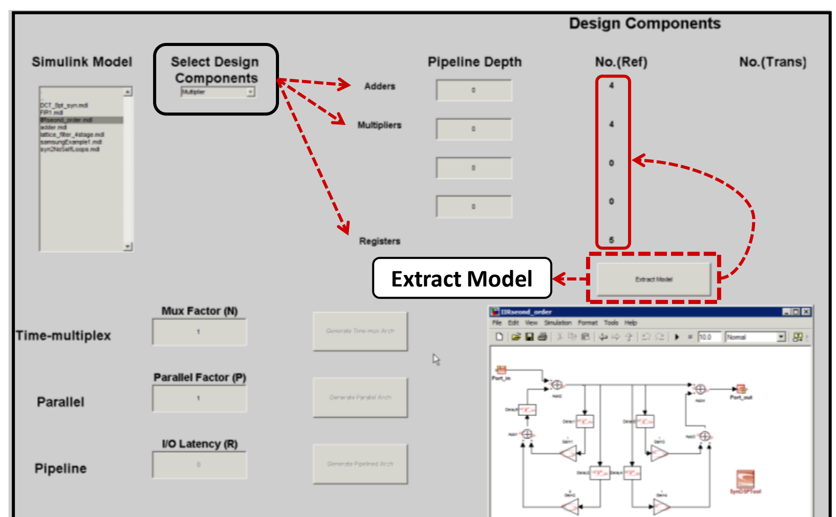
Automated wordlength selection

Ch 11: Architectural Optimization



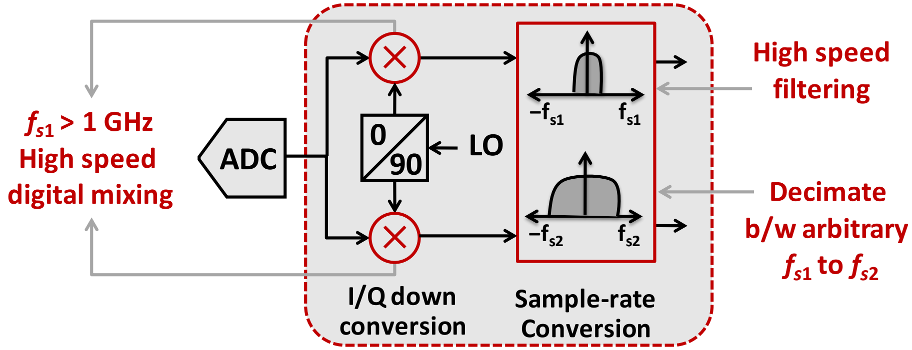
DFG model is used for architecture transformations based on high-level scheduling and retiming, an automated GUI tool is built…

Ch 12: Simulink-Hardware Flow



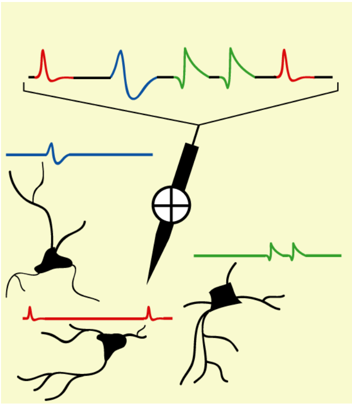
Part 4: Design Examples: GHz to kHz

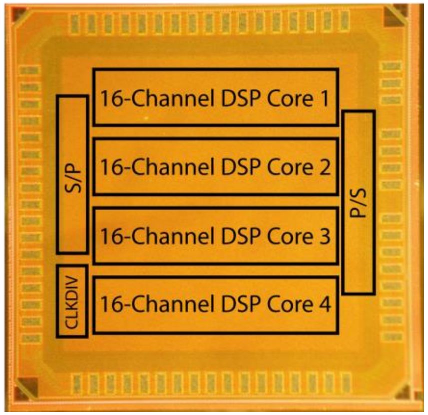
Ch 13: Multi-GHz Radio DSP



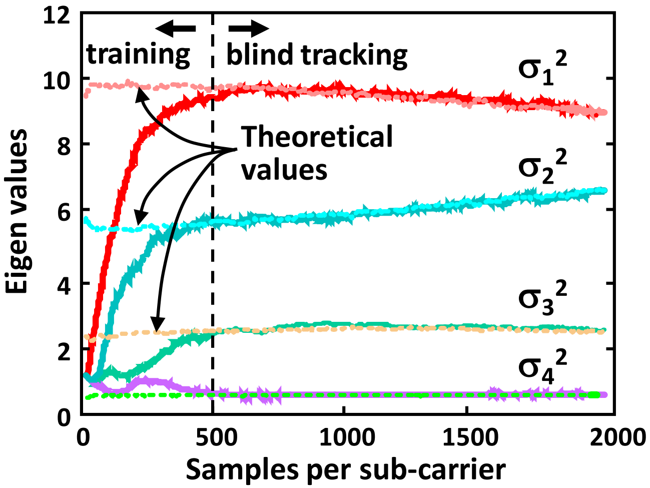
High-speed (GHz+) digital filtering

Ch 16: kHz-rate Neural Processors





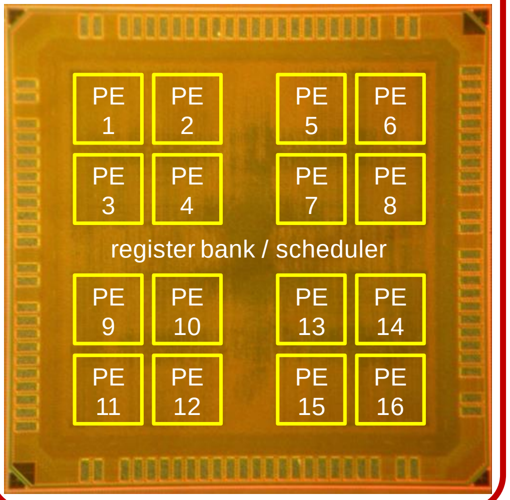
Ch 14: Dedicated MHz-rate Decoders



Adaptive channel gain tracking, parallel data processing (SVD)

Increased number of antennas, added flexibility for multi-mode operation

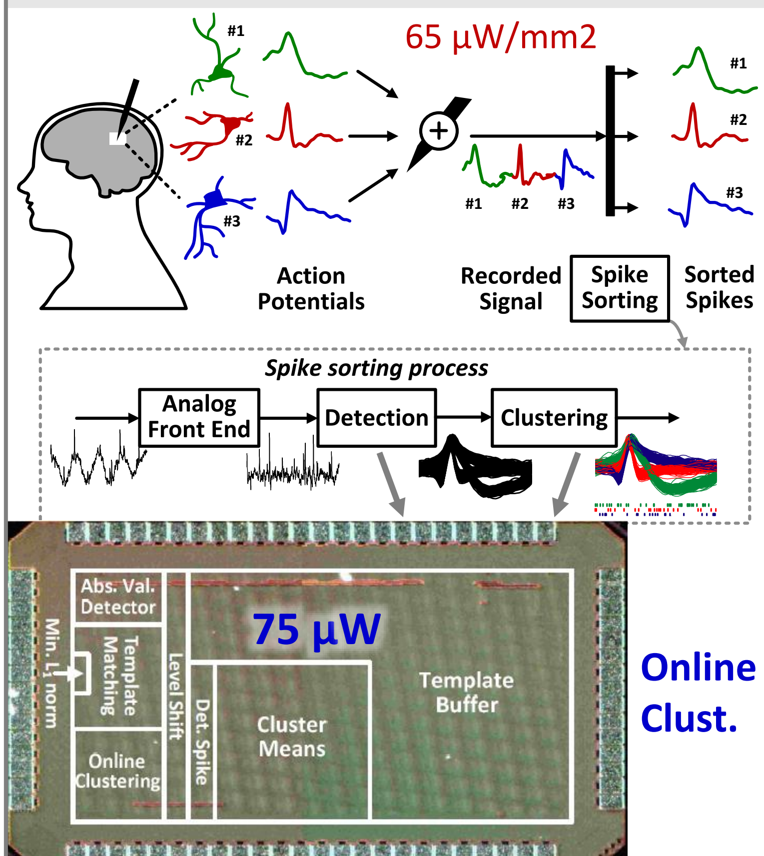
Ch 15: Flexible MHz-rate Decoders



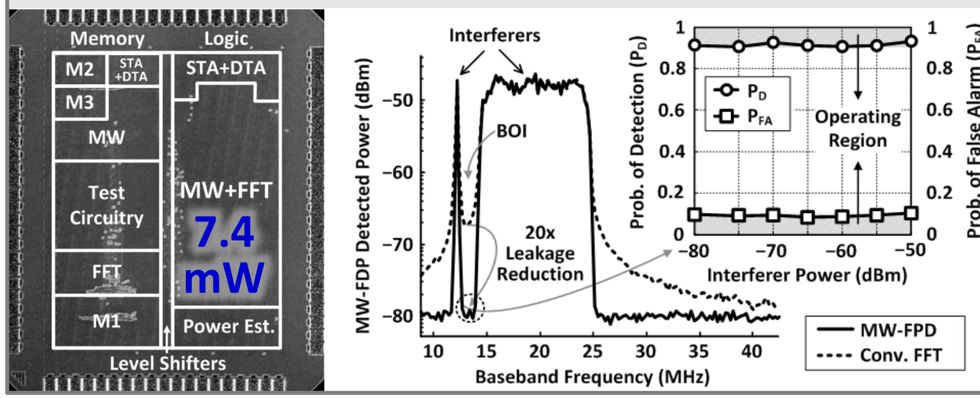
Wide Range of Examples

- **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/mm^2 to mW/mm^2

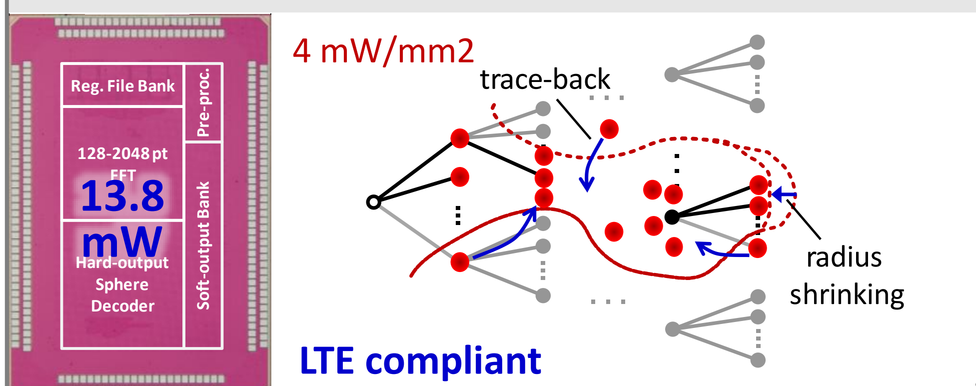
16-ch Neural-spike Clustering



200MHz Cognitive Radio Spectrum Sensing



Multi-core 8x8 MIMO Sphere Decoder



Class Topics

- **Circuit and DSP basics**
  - Circuit and architecture techniques
  - Scheduling and retiming
- **Arithmetic for DSP**
- **Evolving tools landscape**
  - Matlab/Simulink, Synphony HLS, Stratus HLS, PyGears
  - CADA (Configurable Architecture Design Automation)
- **Building blocks**
  - Filters, time-frequency analysis, DSP kernels
- **Systems**
  - Communications, Biomedical, Adaptive learning

Design Trajectory: From DSP Theory…

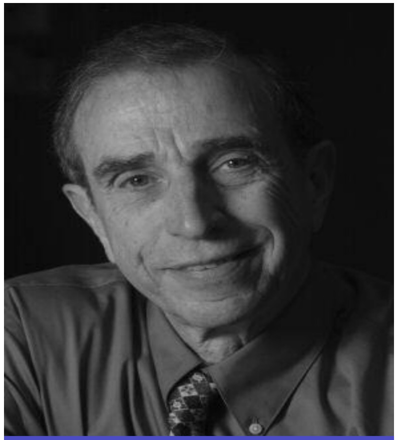
Sample &
Quantize

Audio
Video
Radar

Add
Multiply
Memory

Digital Signal Processing

Harry Nyquist

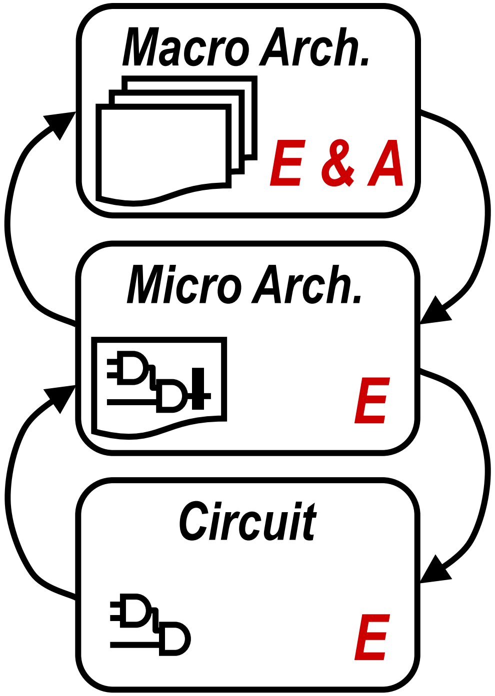


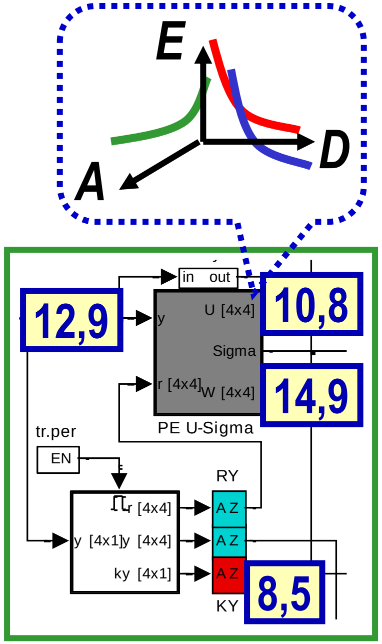
Alan Oppenheim

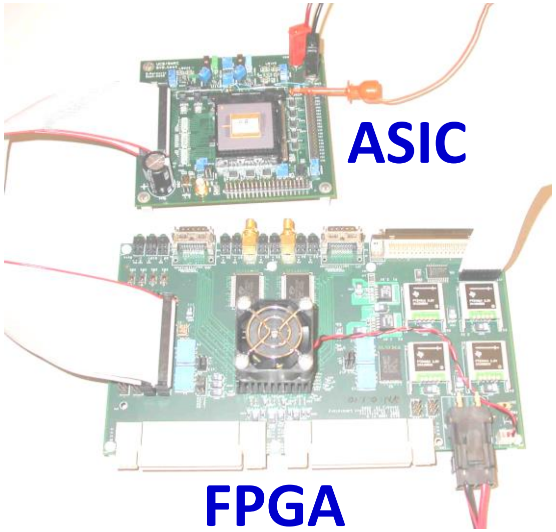
Jean Baptiste Fourier

…to Optimized Hardware Realization

Automated design + verification







optimization ***hardware design*** ***I/O verification***

Class Organization

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

24S ECE216B: Schedule and Syllabus

Weeks 1-5: Methods	
1	(4/2) Introduction
	(4/4) Energy, Delay Models
2	(4/9) Circuit Optimization
	(4/11) Arch. Techniques
3	(4/16) Architecture Flexibility
	(4/18) Arithmetic for DSP
4	(4/23) CORDIC, Div, Sqrt
	(4/25) Digital Filters
5	(4/30) CGRA and UDSP
	(5/5) HLS, CADA Intro

Weeks 6-10: Flows	
6	(5/7) Midterm exam
	(5/9) Data-flow Graphs
7	(5/14) SDR TxRx Design, Opt.
	(5/16) Intro to AI/ML Hw
8	(5/21) Architecture Studies
	(5/23) FFTs & Wavelets
9	(5/28) FFT Architecture Opt.
	(5/30) FPGA Architecture
10	(6/4) Project Presentations
	(6/6) Project Presentations

State-of-the-Art: *discussion of class research explorations on state-of-the-art in productivity tools, hardware, applications*

Architecture Studies (5/21)

- **Focus on AI/ML scheduling**
- **A shift from hardware to software**
- **Example AI/ML Hw/Sw co-design**
  - AHA, MoCA, Efficient Compute
  - List of papers / topics coming by 5/9

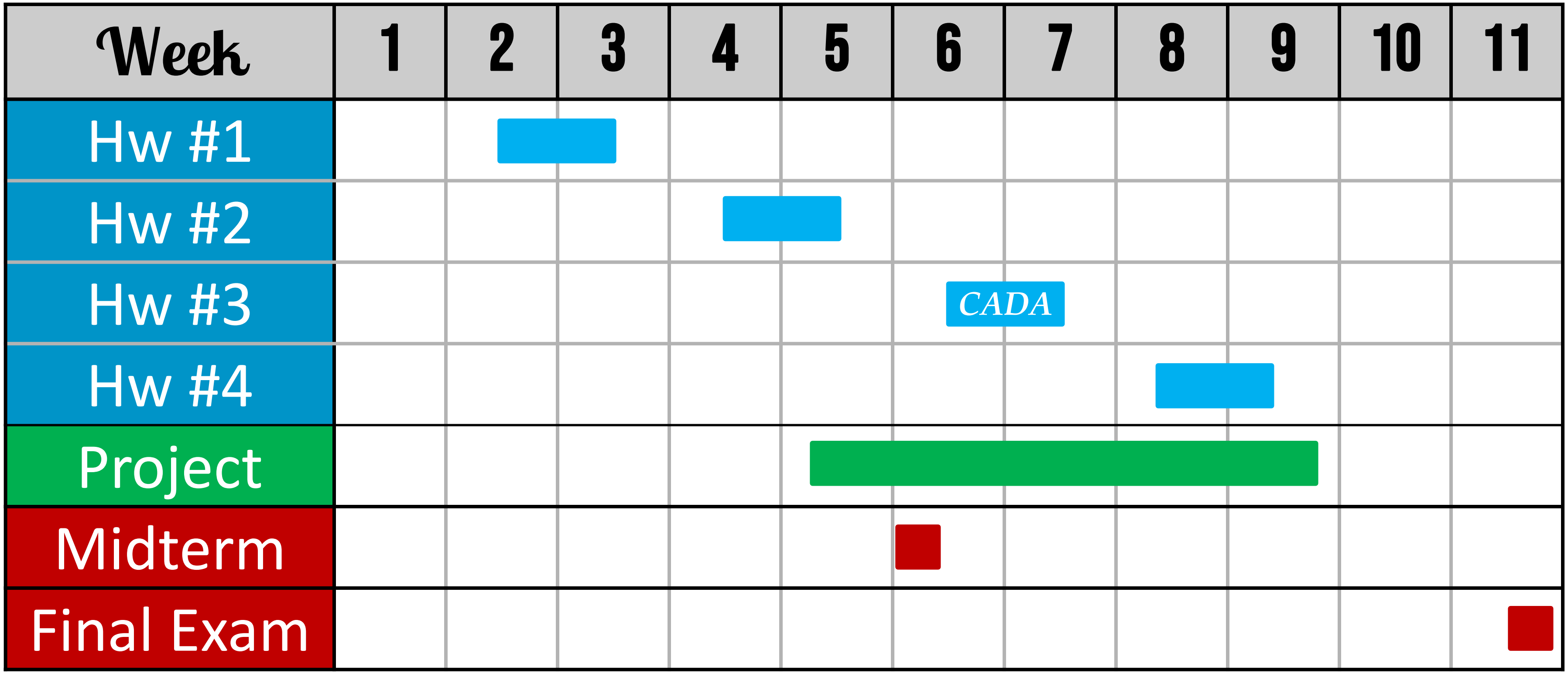
Grading Policy & Organization

20%	• 4 homework sets
30%	• Project
25%	• Midterm
25%	• Final exam

Homework and Project

- **Bi-weekly homework (4 assignments)**
  - Fine-grain DSP blocks
- **Final project: an AI/ML multi-function accelerator**
  - Work in teams of three (~15 projects total)

Gantt Chart



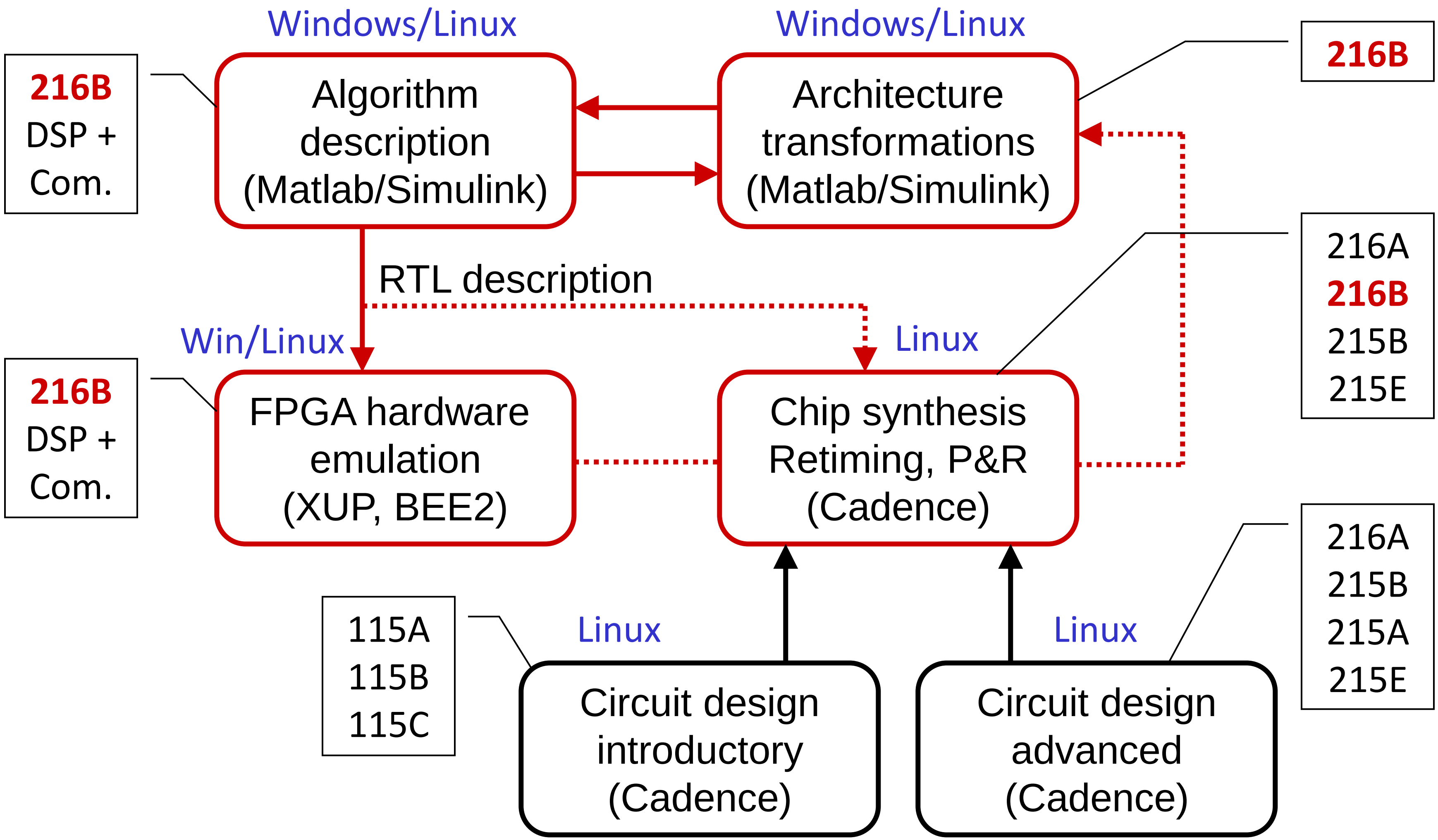
- **In-class presentations in weeks 8 (5/21)**

Synopsys 32/28nm GTech

32/28nm EDK + libraries

- **EDK + libs:** **Synopsys kit and libs**
- **Std cell**
- **I/O**
- **Mem**
- **PLL**
- **Ref. designs**

CAD Environment



Lecture

1

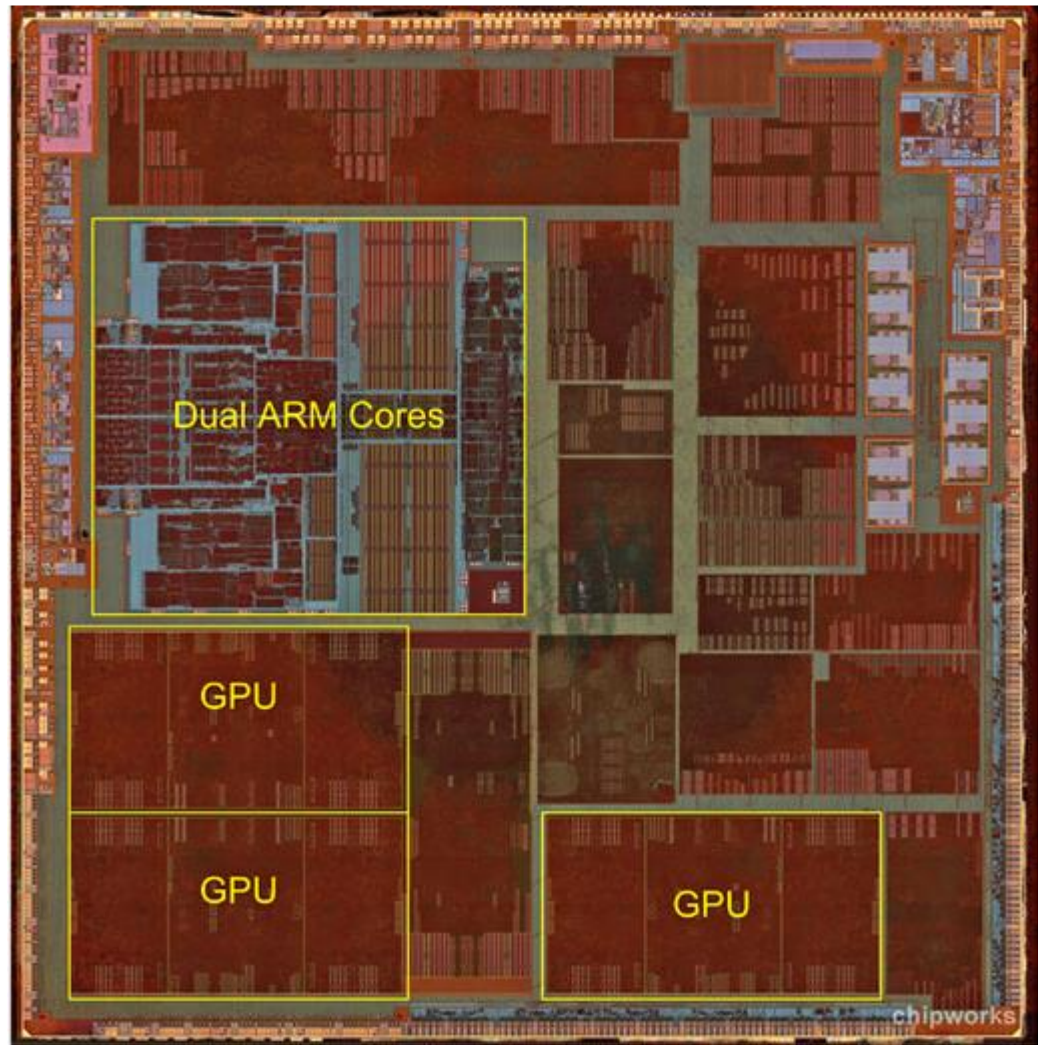
ECE216B

Introduction

Prof. Dejan Marković

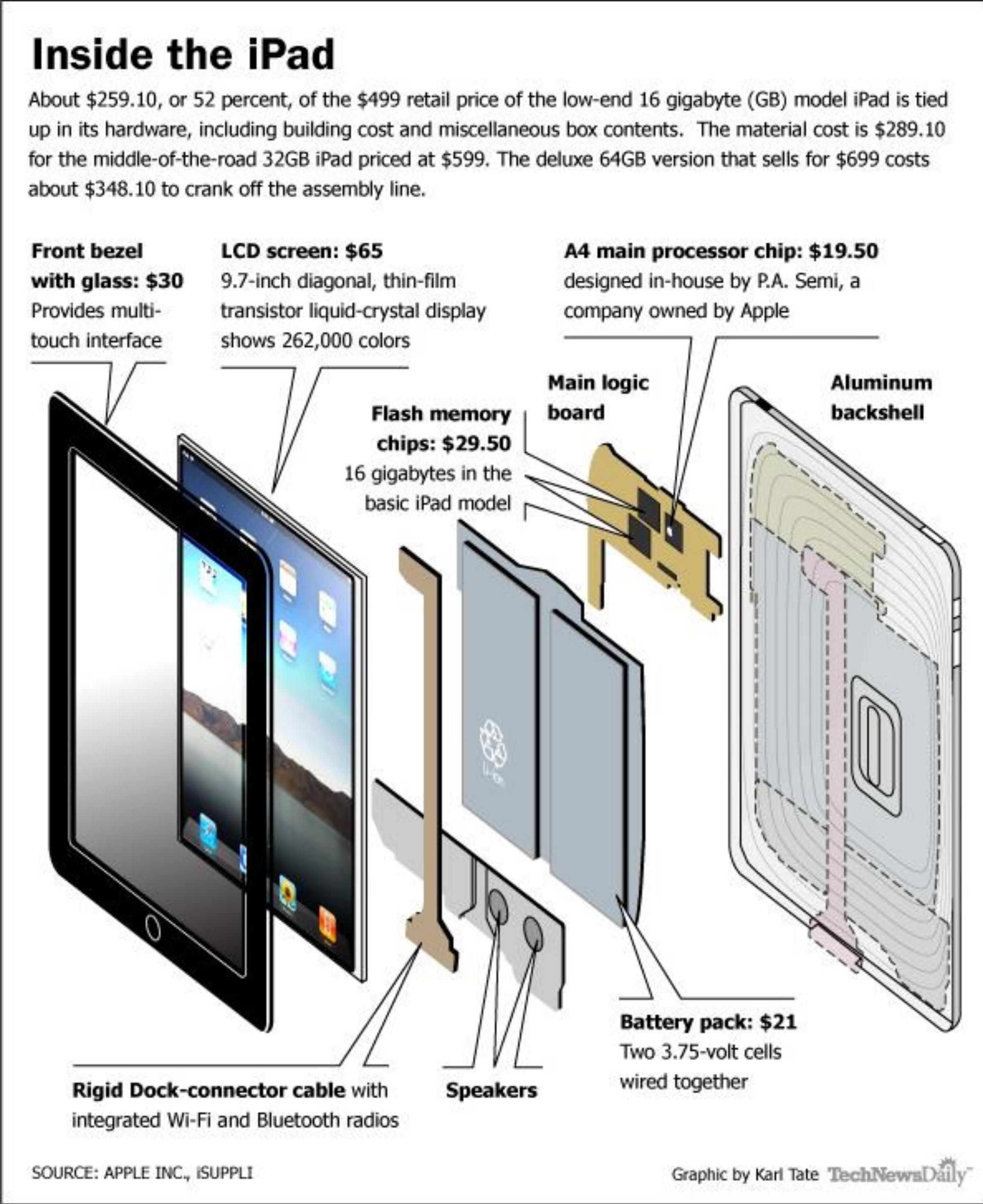
ee216b@gmail.com

iPhone5 A6 Processor



From: Google Images

A6X | iPad4



Dual-core CPU

More capable GPUs

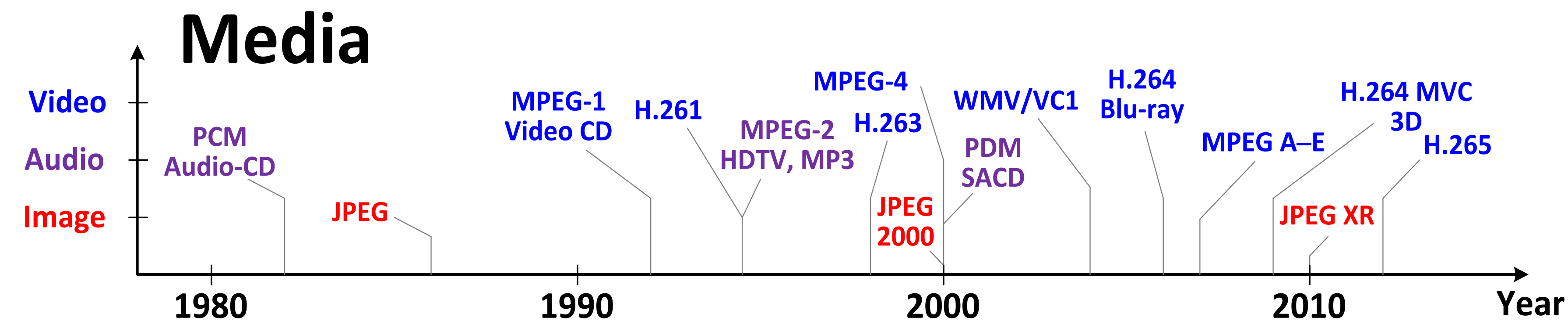
http://photos.appleinsider.com

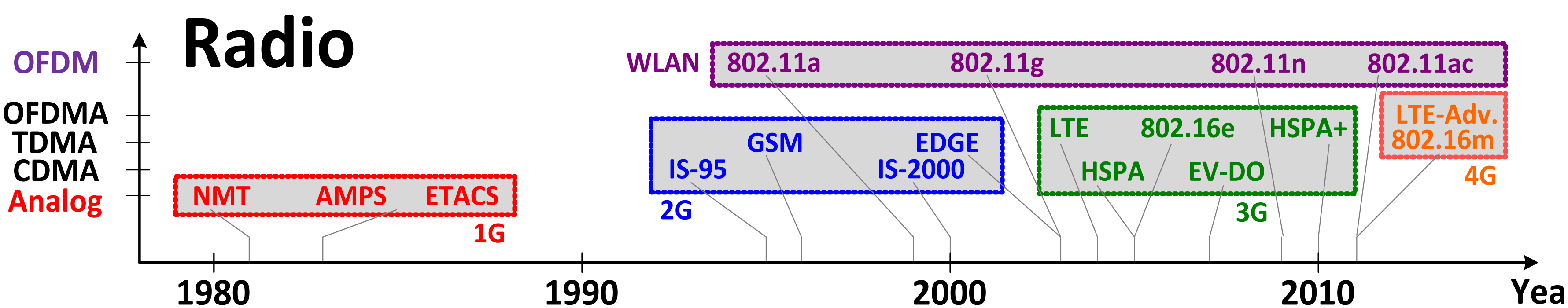
Signal processing content **expanding**

Specialized hardware for energy efficiency

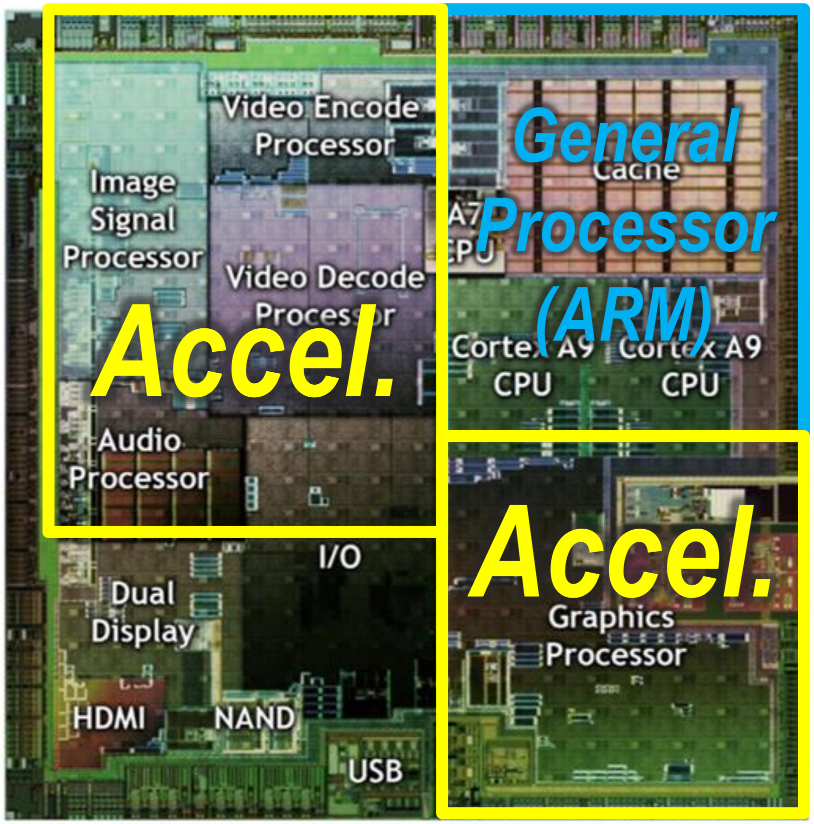
Keeping up with Standards

New standard = New chip?





Today: CPUs + Accelerators



NVIDIA Tegra 2

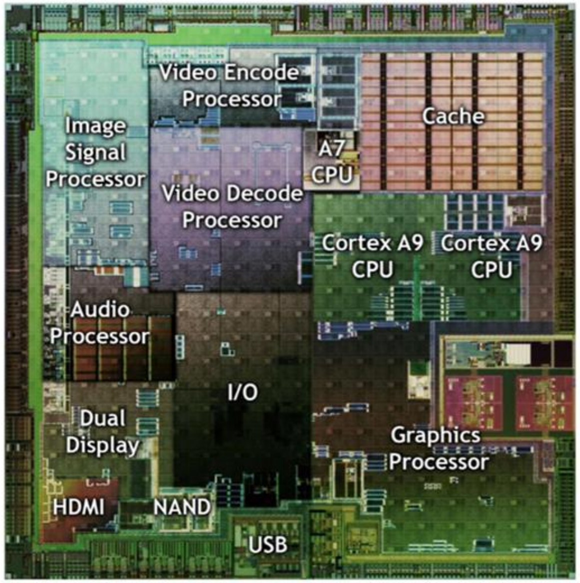
Accelerators:

- Increasingly larger fraction of chip area
- **Low area utilization** a.k.a. **DARK** silicon
- Accelerators for **fixed standards**

Architecture Insights | Recent Tegra Chips

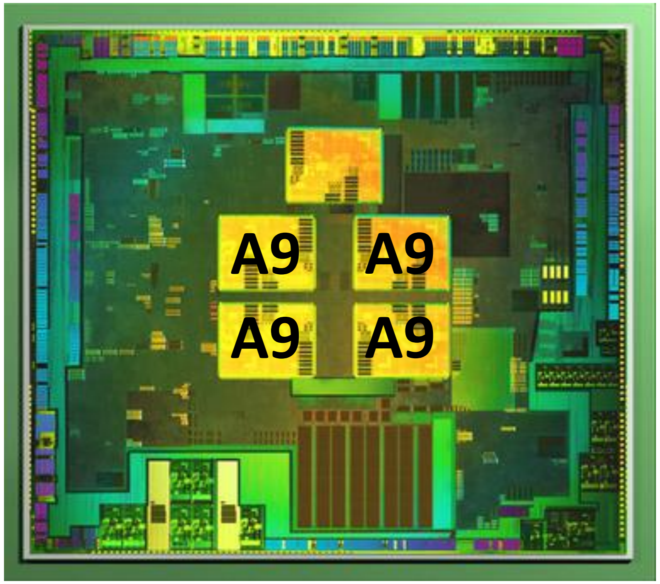
Customization, increasing number of cores…

Tegra 2 (2011)



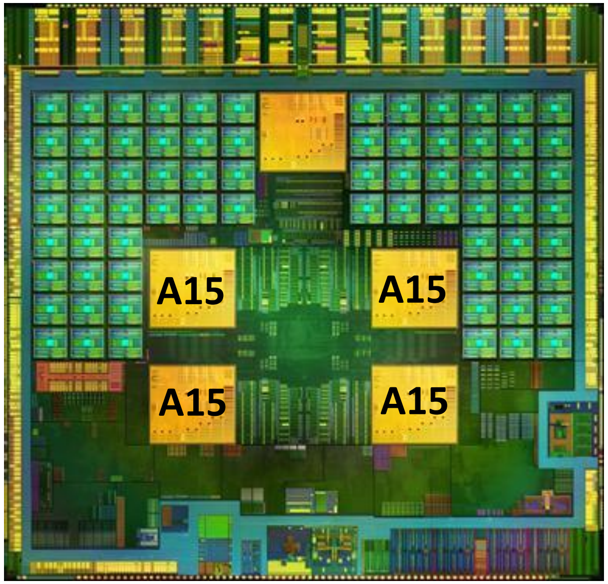
- Dual-A9

Tegra 3 (2012)



- Quad-A9
- Power-saver core

Tegra 4 (2013)

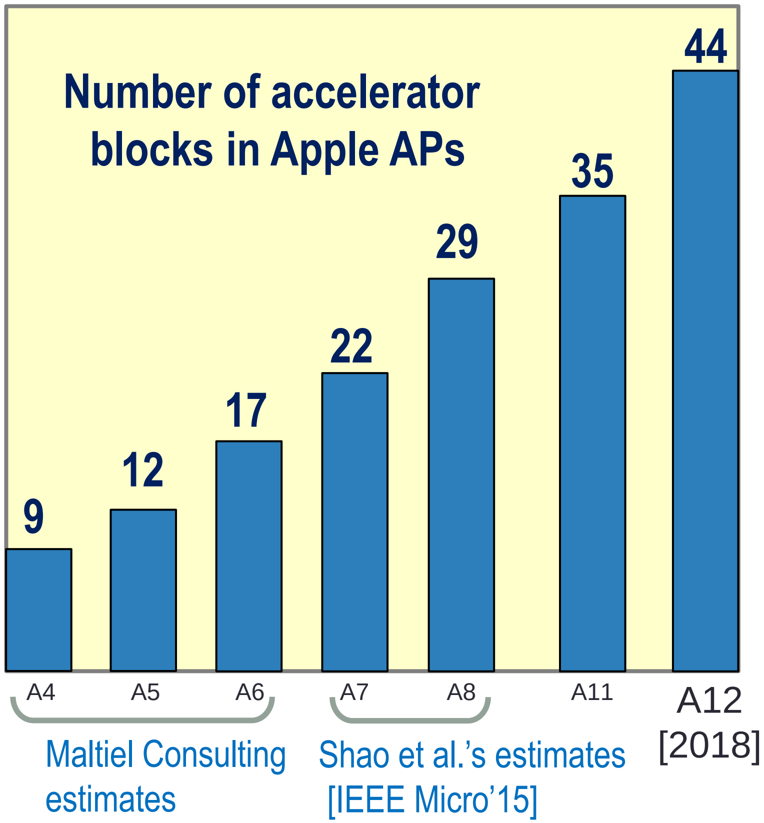


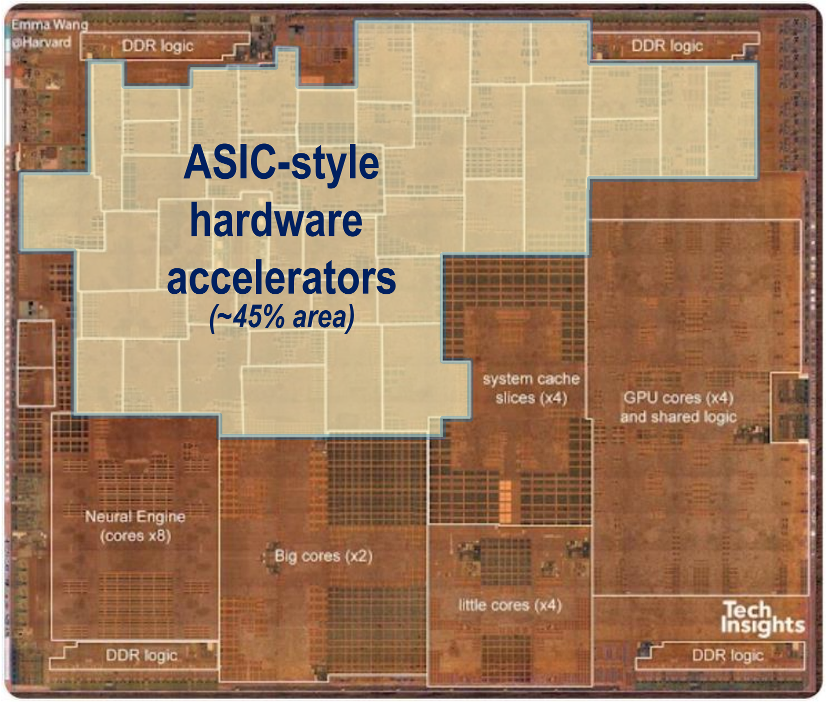
- 72 GPU cores
- LTE modem
- Computational camera

From: Google Images

Heterogeneous Computing in Mobile SoCs

Increasing "dark silicon" area (A12: ~45%, A15: ~55%), <10% chip is active





Apple A12 die photo

ASSPs: Rapid Growth in Development Cost

Towards economic end of scaling…

The Basic Problem

Algorithm designers | **Chip designers**

Shannon limit, Raleigh fading, cyclostationary process ⇨ **?** ^$*#^$E(W^$^&$

^$*#^$E(W^$^&$ **?** ⇦ Gate delay, leakage power number of bits, latency

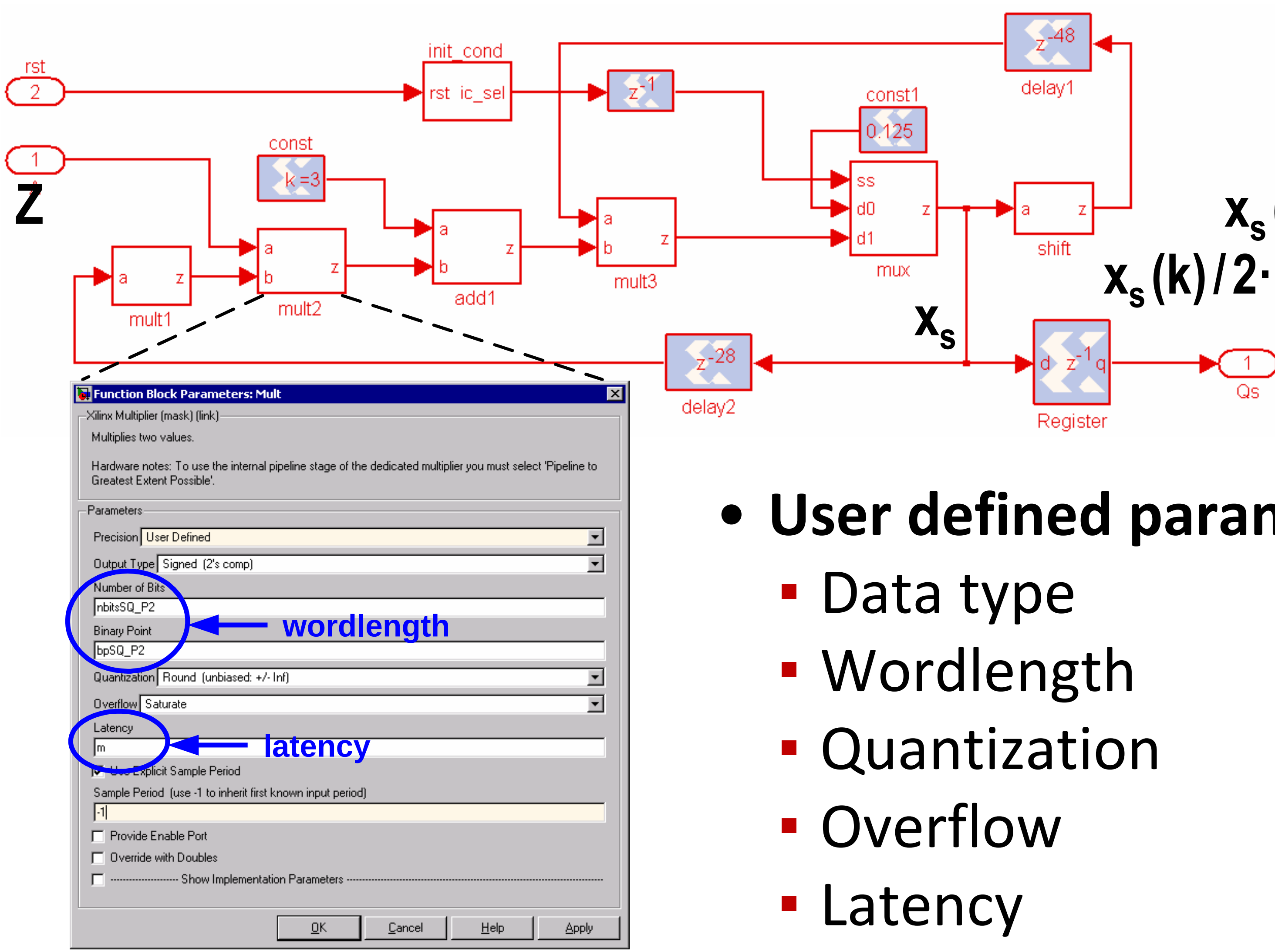
- **Very constrained implementation choices**
- **Design reentry (Matlab/C, HDL)**

Algorithm-Hardware Co-Design Approach

- **Unified HLS (e.g. Simulink) environment**
  - Enter design only once!
  - Algorithm verification / emulation
  - Abstract view of architecture
  - FPGA based ASIC debug

- **Hardware-equivalent blocks**
  - Basic ops: add, multiply, shift, mux…
  - Implementation constraints
    - Word-size, latency

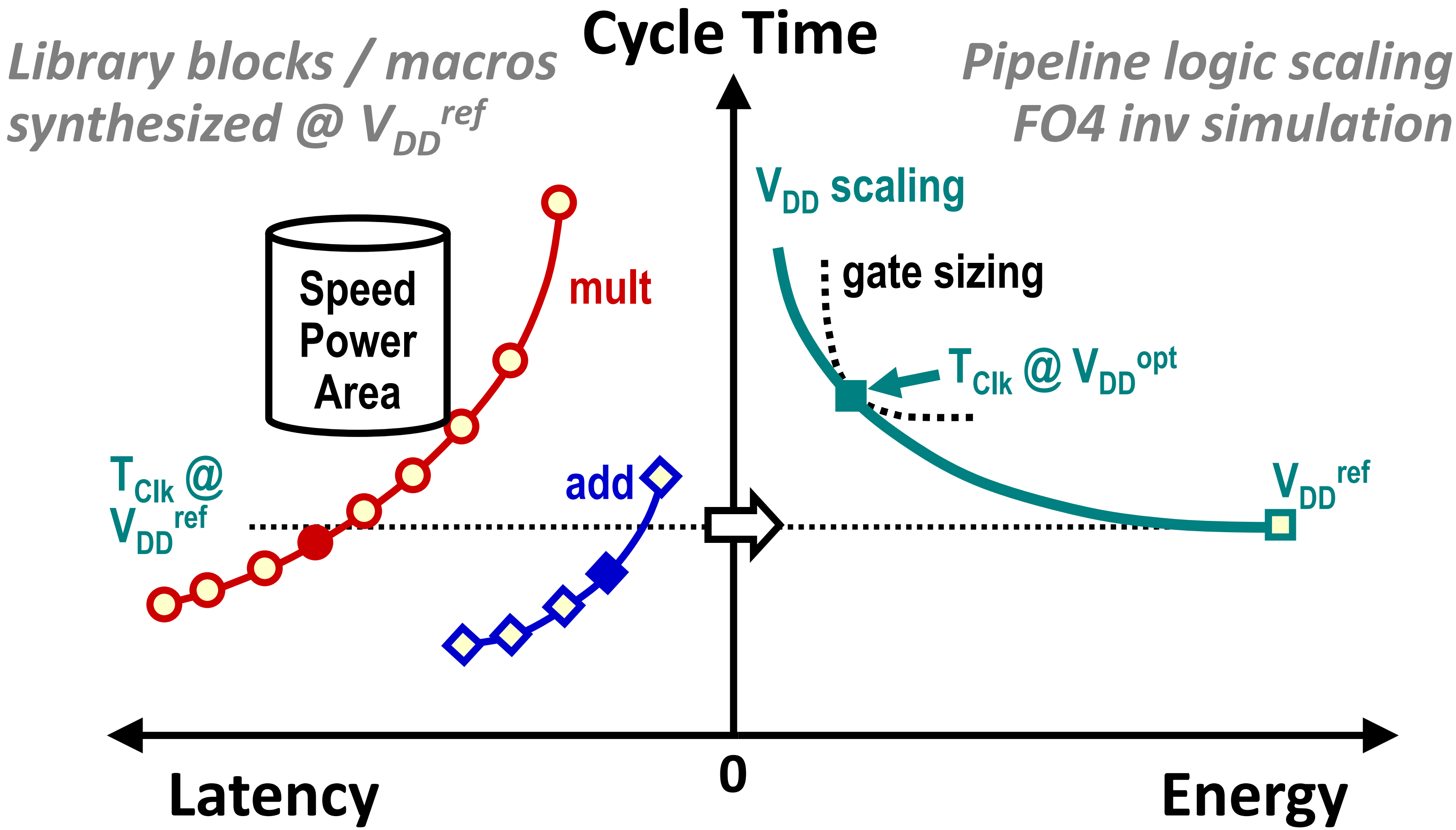
XSG Model Example: Iterative 1/sqrt()



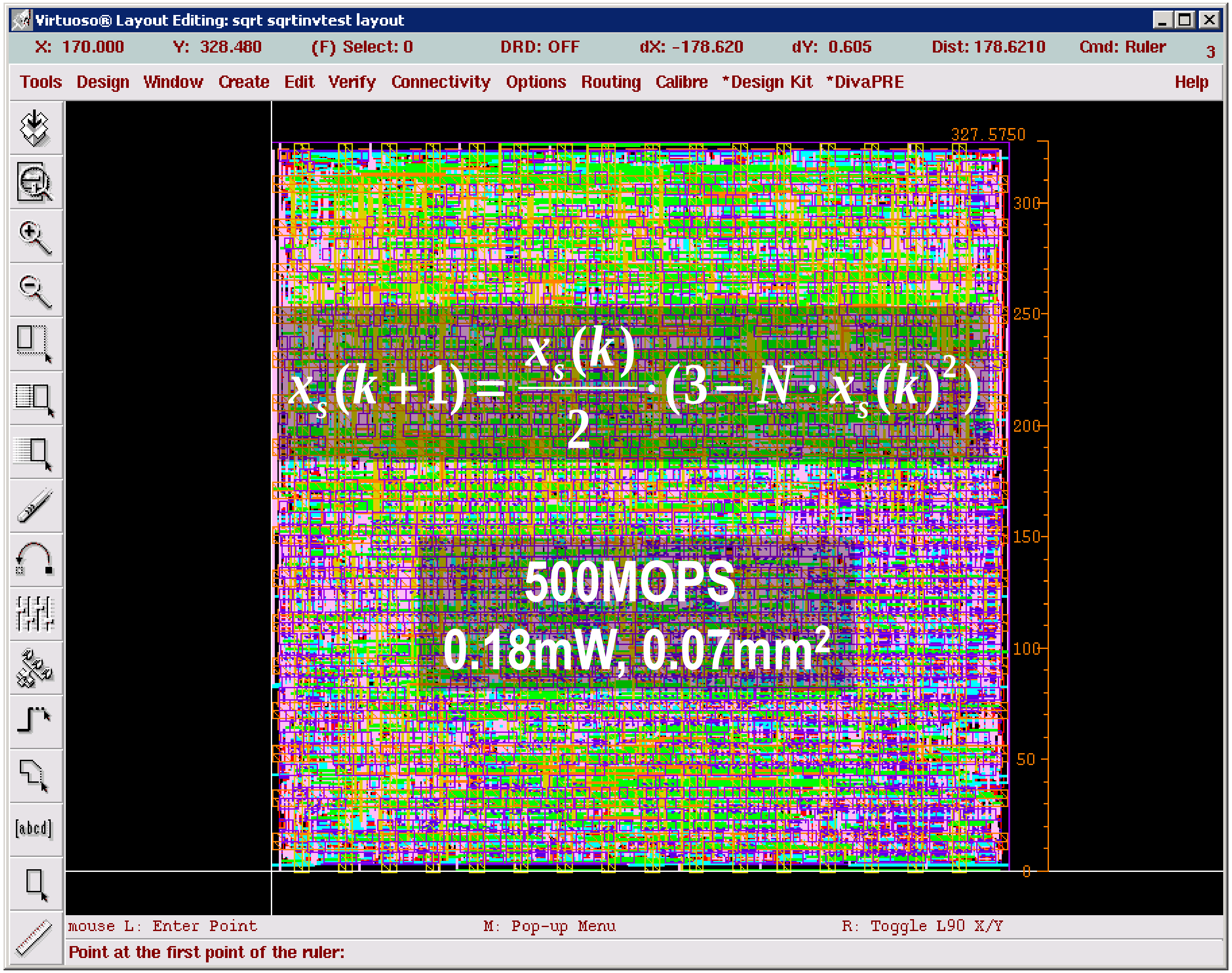
$$x_s(k+1) = x_s(k)/2 \cdot (3 - Z \cdot x_s^2(k))$$

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

Block Characterization



Automated Chip Synthesis



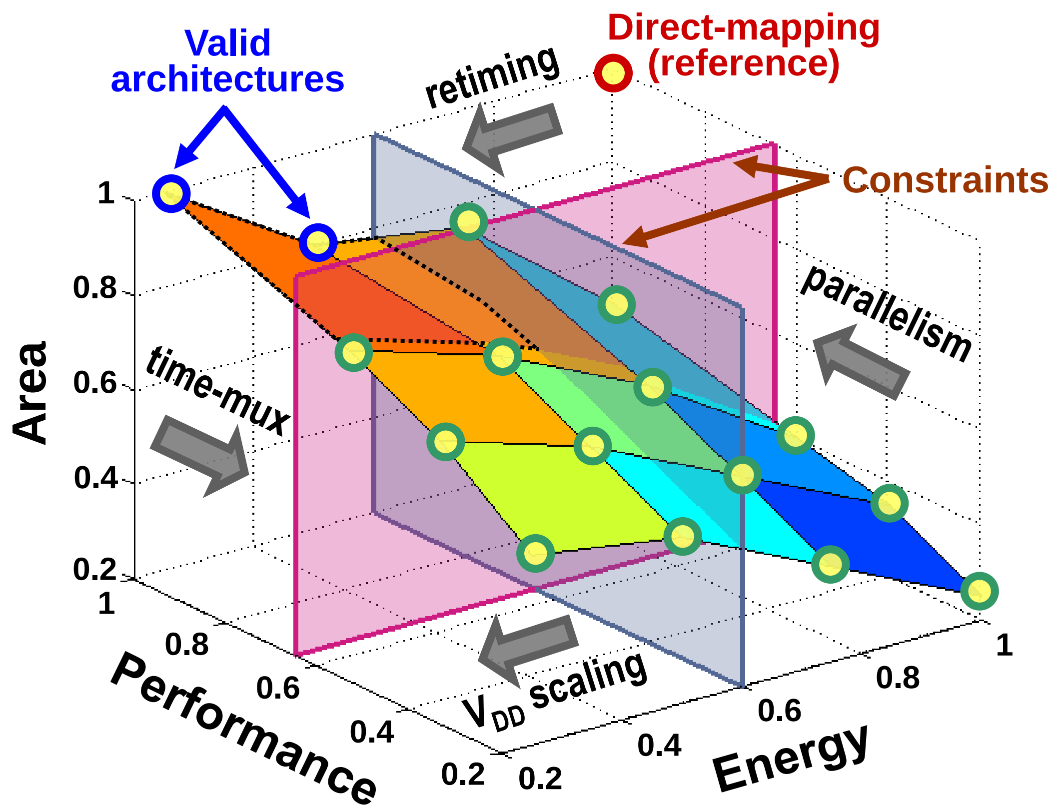
10,000 FPGA slices

⇔

$1mm^2$ (90nm CMOS)

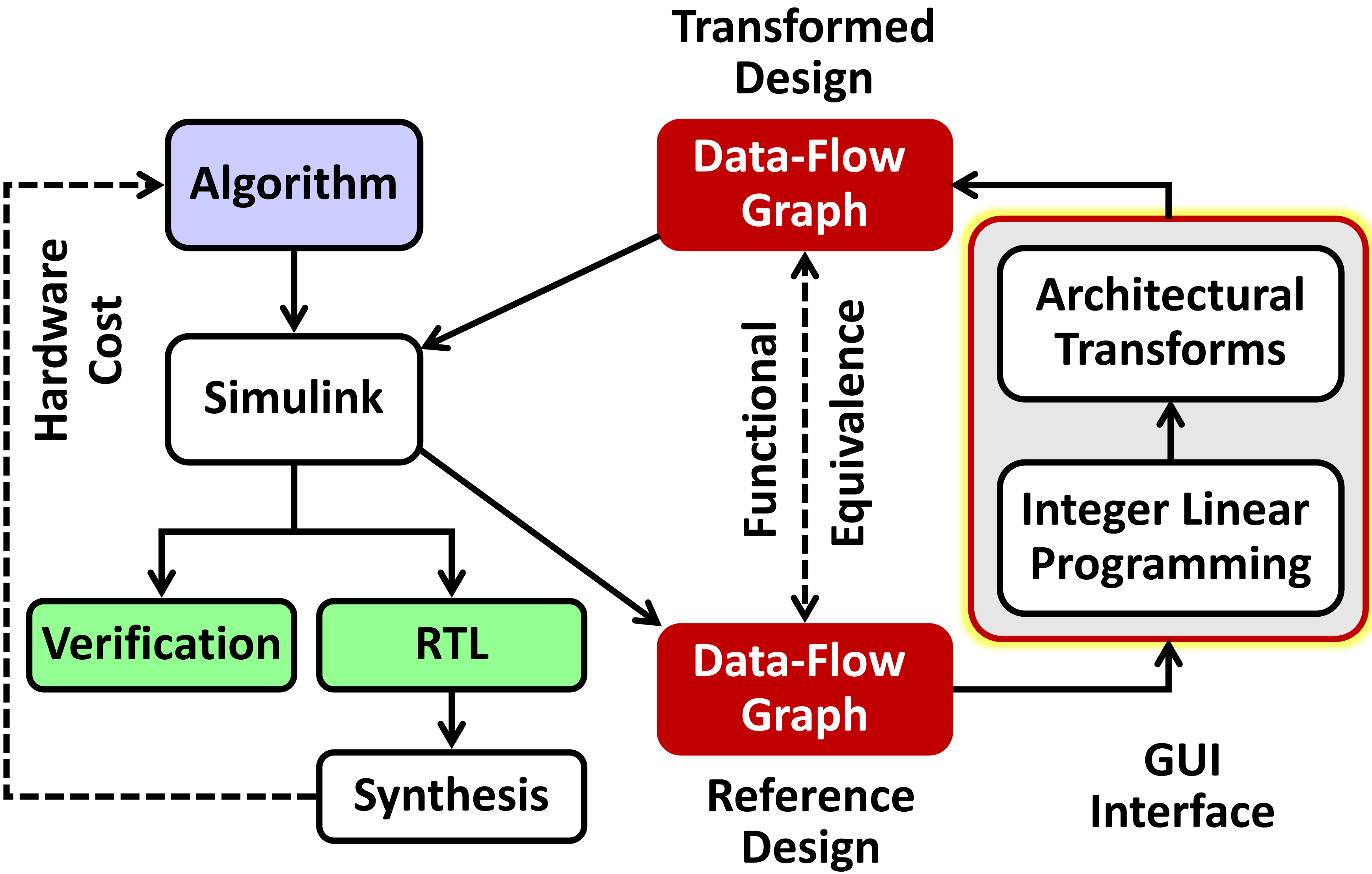
Energy-Area-Performance Space

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



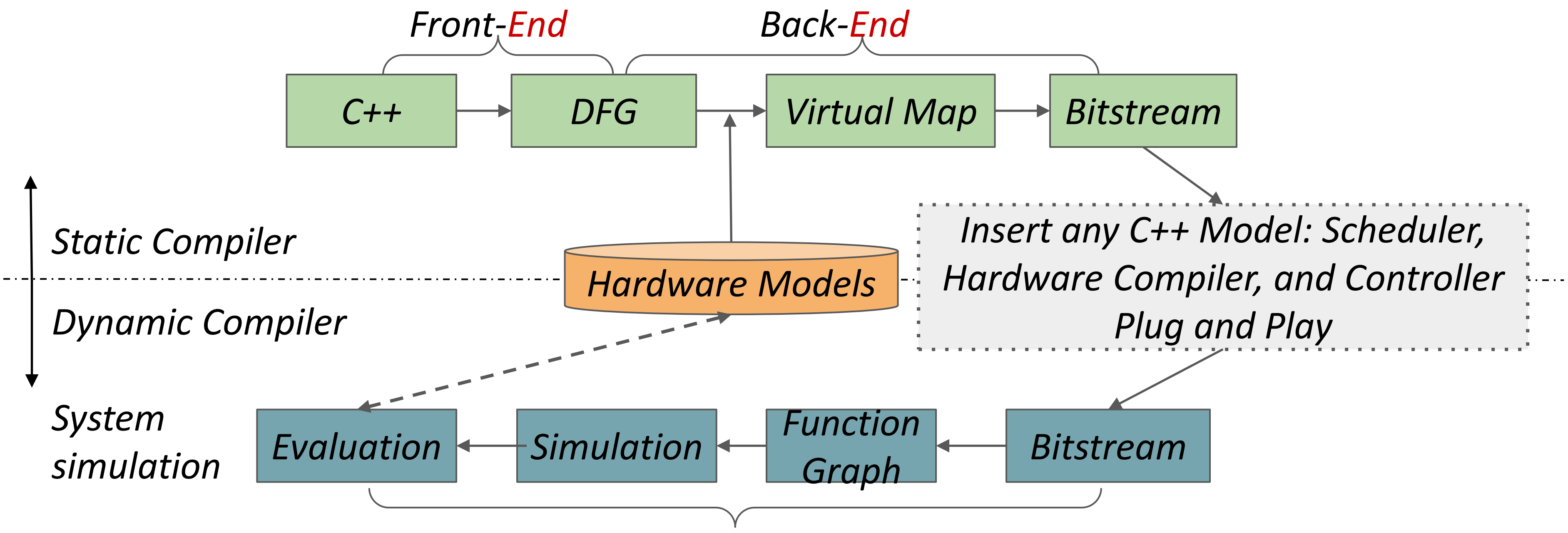
[Rashmi Nanda]

Simulink & Data-flow Graphs

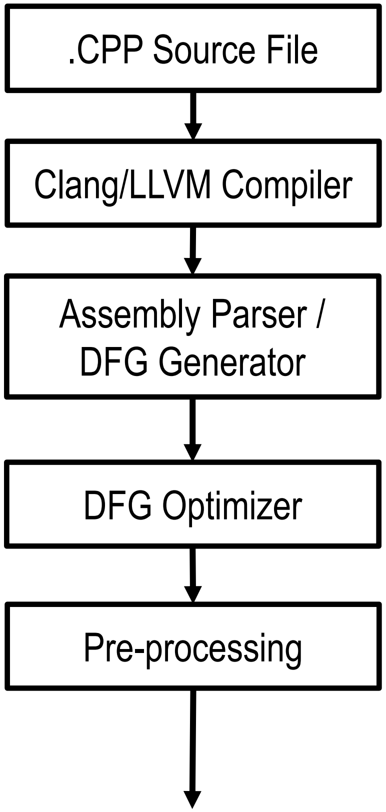


An End-to-End Software Stack

- **Hardware-aware C++ to binary compiler**
- **Integrated validation**



Software: Compiler Front-End



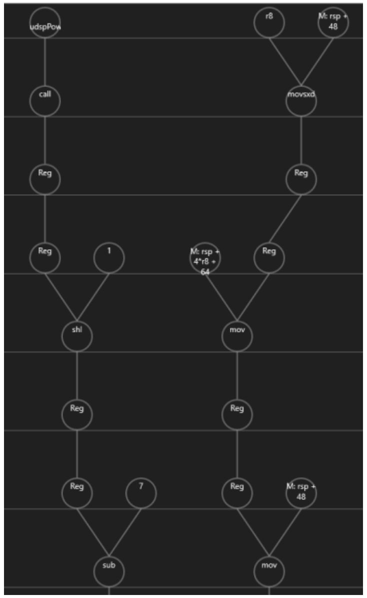
.CPP Source File

```cpp
int main()
{
    int c[10];
    int a = 3;
    int b = 5;
    int i = 0;
    int d = 6;
    while (i <= 10)
    {
        b += 9 + d;
        a +=  4 * udspOperator(b, 3);
        c[i] = 3 * udspOperator( a , 3) - 7;
        i++;
    }
}
```

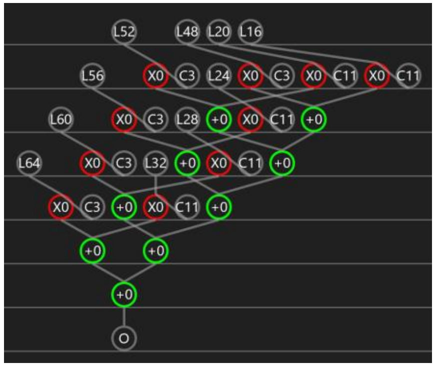
Clang/LLVM Compiler

```asm
.LBB0_1:                                # =>This Inner Loop Header: Depth=1
    .cv_loc 0 1 20 0                    # LLVM_Test.cpp:20:0
    cmp dword ptr [rsp + 48], 10
    jg  .LBB0_3
# %bb.2:                                #   in Loop: Header=BB0_1 Depth=1
.Ltmp1:
    .cv_loc 0 1 22 0                    # LLVM_Test.cpp:22:0
    mov eax, dword ptr [rsp + 44]
    add eax, 9
    add eax, dword ptr [rsp + 52]
    mov dword ptr [rsp + 52], eax
    .cv_loc 0 1 23 0                    # LLVM_Test.cpp:23:0
    mov ecx, dword ptr [rsp + 52]
    mov edx, 3
    call    "?udspOperator@@YAHHH@Z"
    shl eax, 2
    add eax, dword ptr [rsp + 56]
    mov dword ptr [rsp + 56], eax
    .cv_loc 0 1 24 0                    # LLVM_Test.cpp:24:0
    mov ecx, dword ptr [rsp + 56]
    mov edx, 3
    call    "?udspOperator@@YAHHH@Z"
    imul    eax, eax, 3
```

DFG Generator

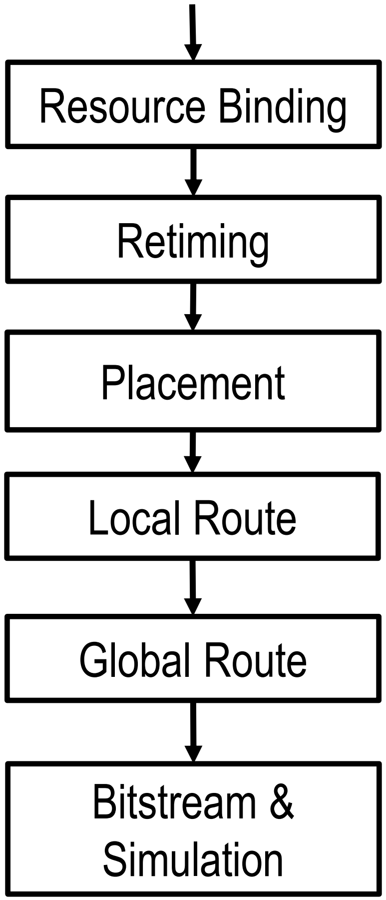


DFG Optimizer + Pre-processing

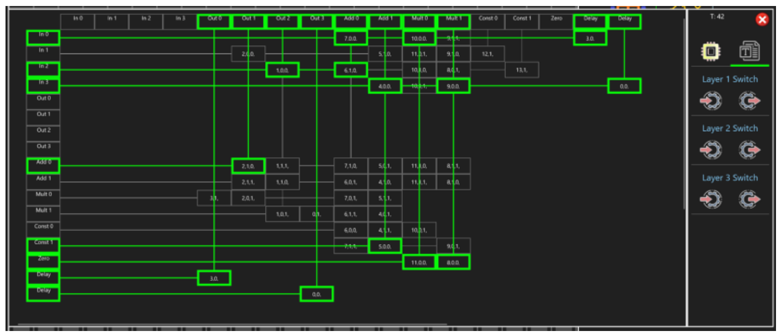


Python source: future work

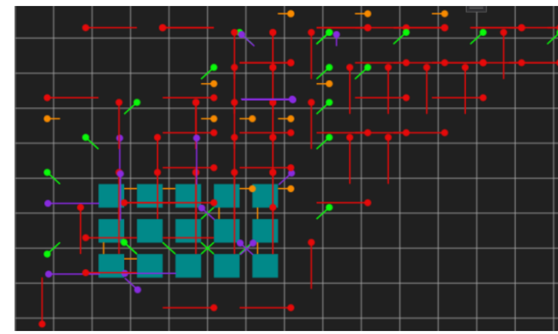
Software: Compiler Back-End



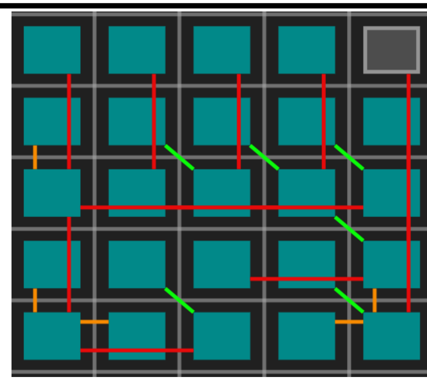
Resource Binding



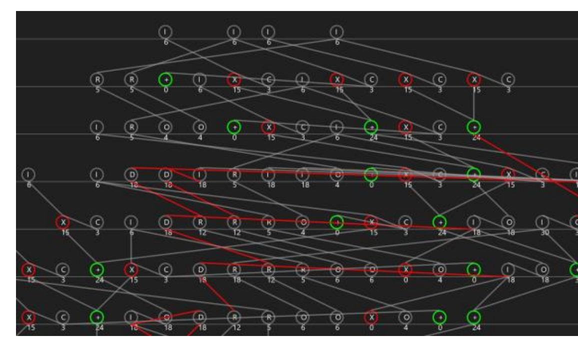
Global Route



Placement & Local Route



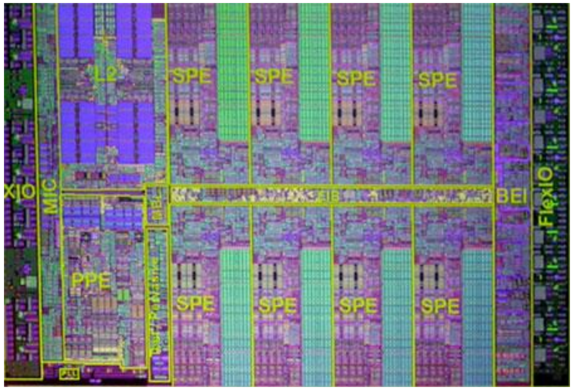
Simulation



Parallel Data Processing

Single dimensional → Multidimensional data

Multi-core Processors

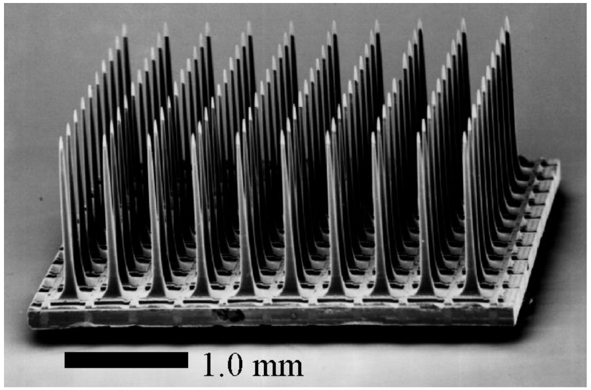


IBM / Sony / Toshiba

MIMO Communications

Belkin

Neuroscience

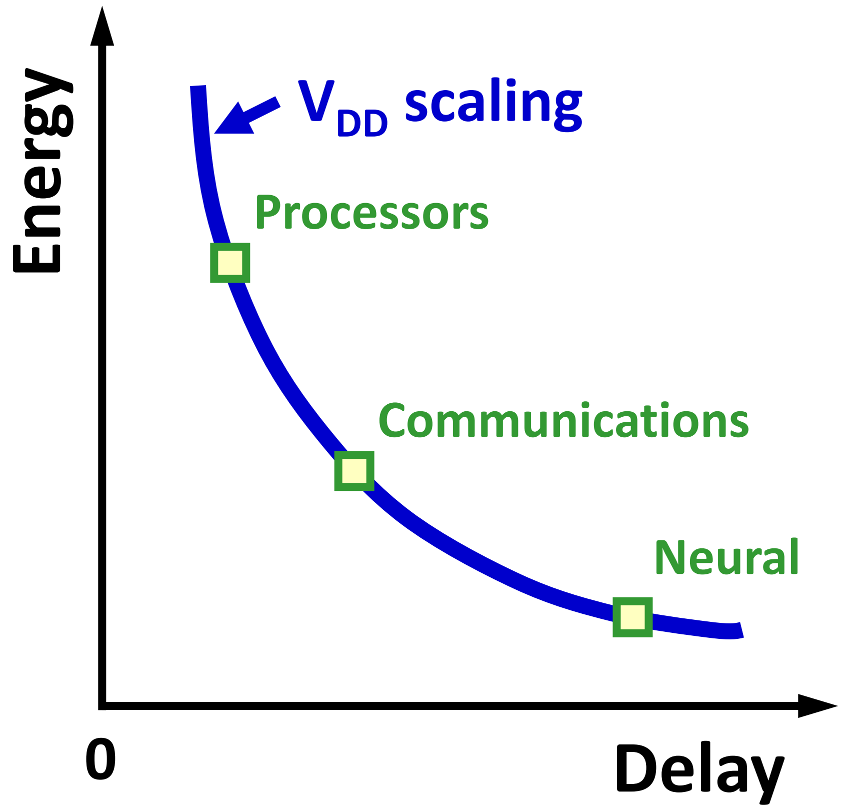


www.sci.utah.edu

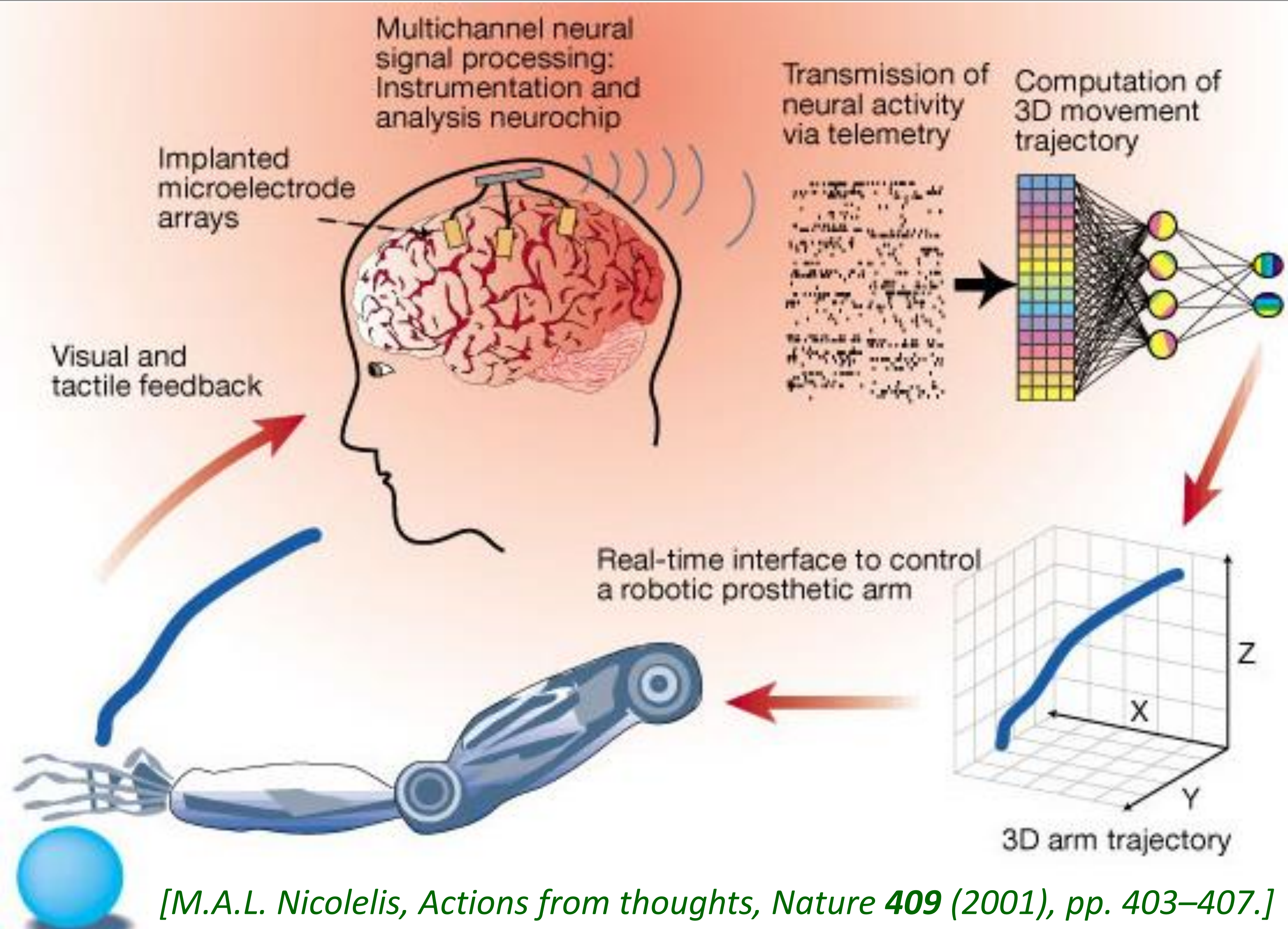
Algorithm-hardware co-design?

Energy-Delay Tradeoff

- **Processors**
  - Maximize performance
  - Highest V_{DD} required
- **Communications**
  - Minimize energy & area
  - Typically, sensitivity ~1
- **Neuroscience**
  - Power density << 0.8mW/mm^2
  - Aggressive V_{DD} scaling



Parallel Data in Neuroscience



*[M.A.L. Nicolelis, Actions from thoughts, Nature **409** (2001), pp. 403–407.]*

Summary: Focus of This Course

3 components of the design problem

- **Algorithm specification**
  - Floating-point, implementation independent, system simulation
- **Architecture mapping**
  - High-level synthesis based approach
  - Rapid architecture tradeoffs
- **Hardware optimizations**
  - Real-time emulation
  - FPGA/ASIC implementation