Application Specific Architectures

Introduction and Motivation

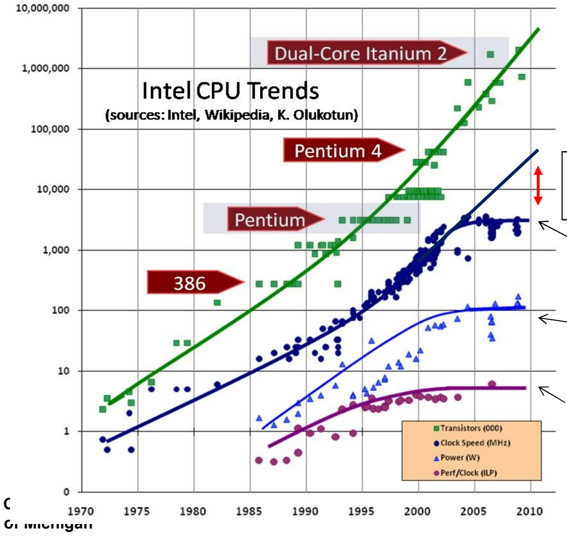
Todd Austin

EECS 573
Fall 2016
University of Michigan

Architecture’s Diminishing Return

- Staples of value we strive for…
  - High Speed
  - Low Power
  - Low Cost
- Tricks of the trade
  - Faster clock rates, via pipelining
  - Higher instruction throughput, via ILP extraction
  - Homogeneous parallel systems
- Strong evidence of diminishing return, PIII vs. P4
  - PIII vs. P4: 22% less P4 throughput (0.35 vs. 0.45 SPECInt/MHz)
  - Parallel resources not fully harnessed by today’s software
- Less return ⇒ less value ⇒ ☹

Moore's Law Performance Gap



Today, gap is cresting 10x

Lack of perceived value

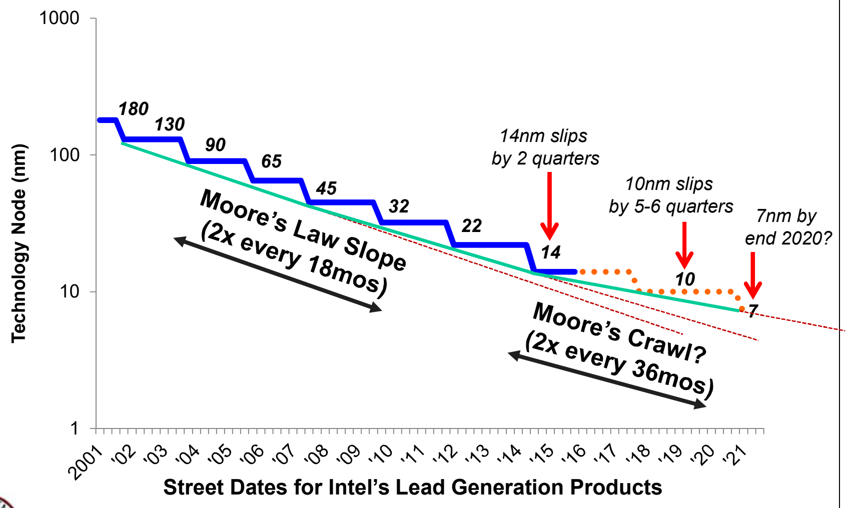
Dark silicon

Diminished ILP

Advanced C
University of Michigan

ecific Architectures
Todd Austin

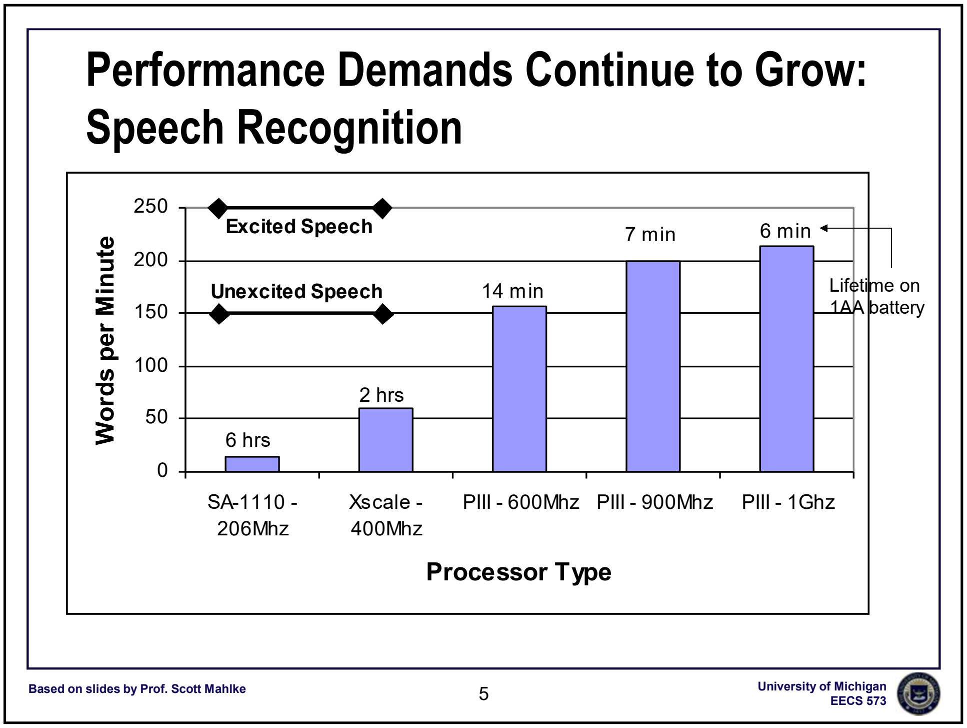
Is Density Still Scaling?



Advanced Computer Architecture Laboratory
University of Michigan

Application Specific Architectures
Todd Austin

Compiled with David Brooks @ Harvard

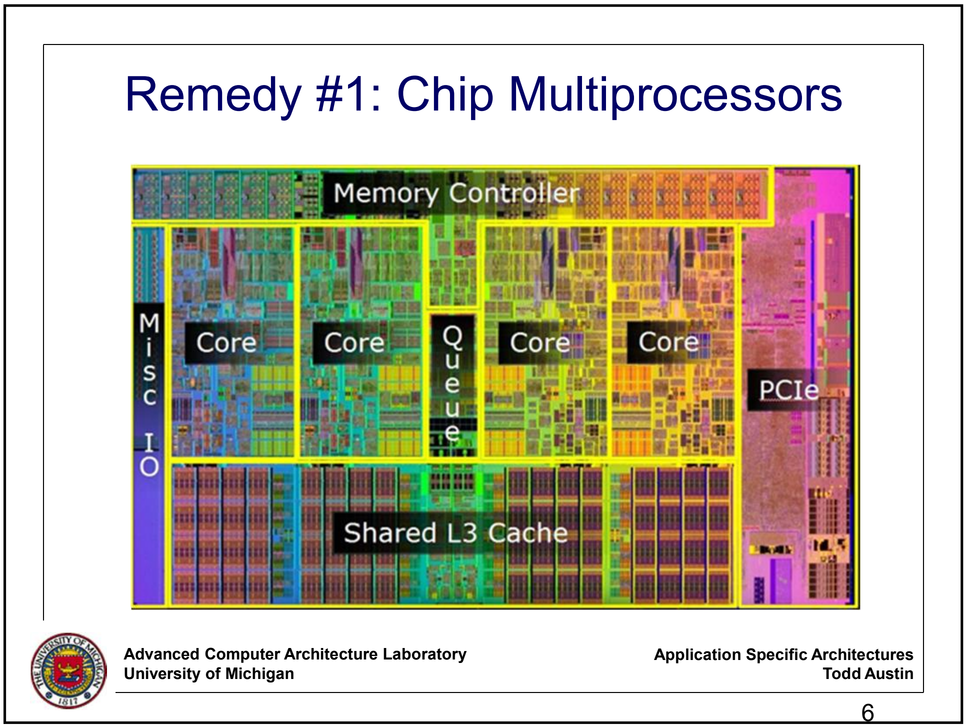


Performance Demands Continue to Grow: Speech Recognition

Based on slides by Prof. Scott Mahlke

5

University of Michigan
EECS 573

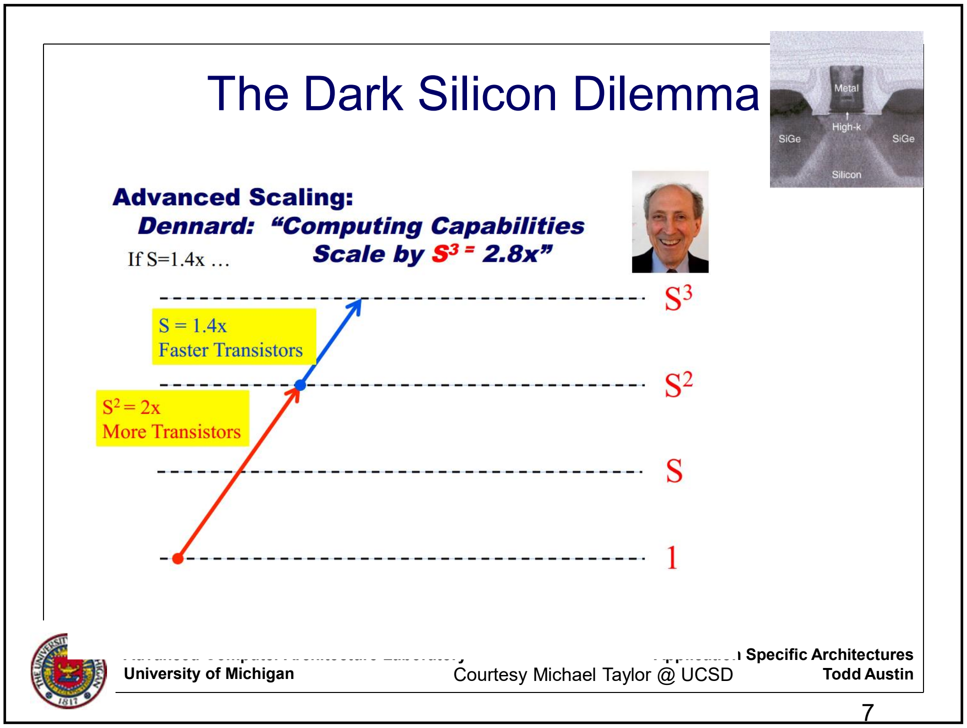


Remedy #1: Chip Multiprocessors

Advanced Computer Architecture Laboratory
University of Michigan

Application Specific Architectures
Todd Austin

6



The Dark Silicon Dilemma

Advanced Scaling:
Dennard: "Computing Capabilities Scale by S^3 = 2.8x"

If S=1.4x …

S = 1.4x
Faster Transistors

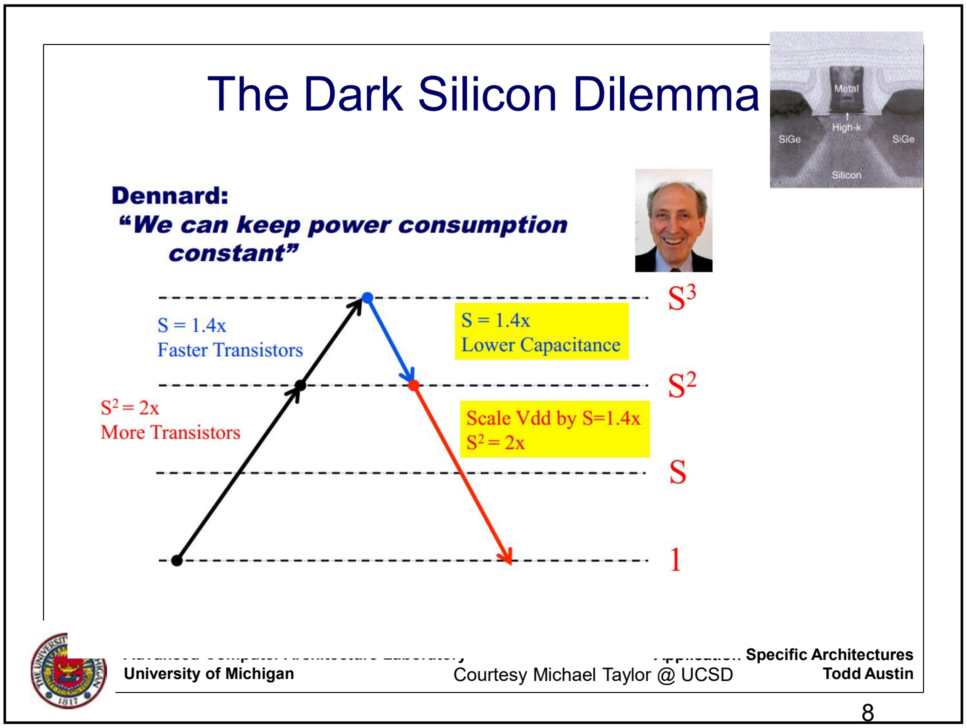
S^2 = 2x
More Transistors

University of Michigan

Courtesy Michael Taylor @ UCSD

Specific Architectures
Todd Austin

7



The Dark Silicon Dilemma

Dennard:
"We can keep power consumption constant"

S = 1.4x
Faster Transistors

S^2 = 2x
More Transistors

S = 1.4x
Lower Capacitance

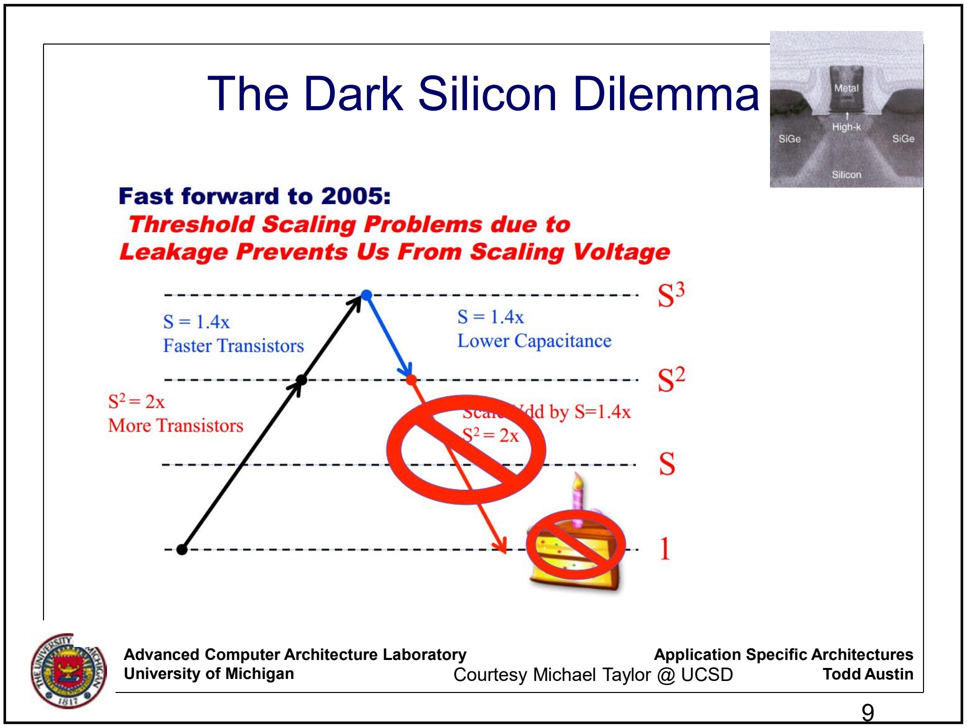
Scale Vdd by S=1.4x
S^2 = 2x

University of Michigan

Courtesy Michael Taylor @ UCSD

Specific Architectures
Todd Austin

8



The Dark Silicon Dilemma

Fast forward to 2005:
Threshold Scaling Problems due to Leakage Prevents Us From Scaling Voltage

S^3

S = 1.4x
Faster Transistors

S = 1.4x
Lower Capacitance

S^2

S^2 = 2x
More Transistors

Scale Vdd by S=1.4x
S^2 = 2x

S

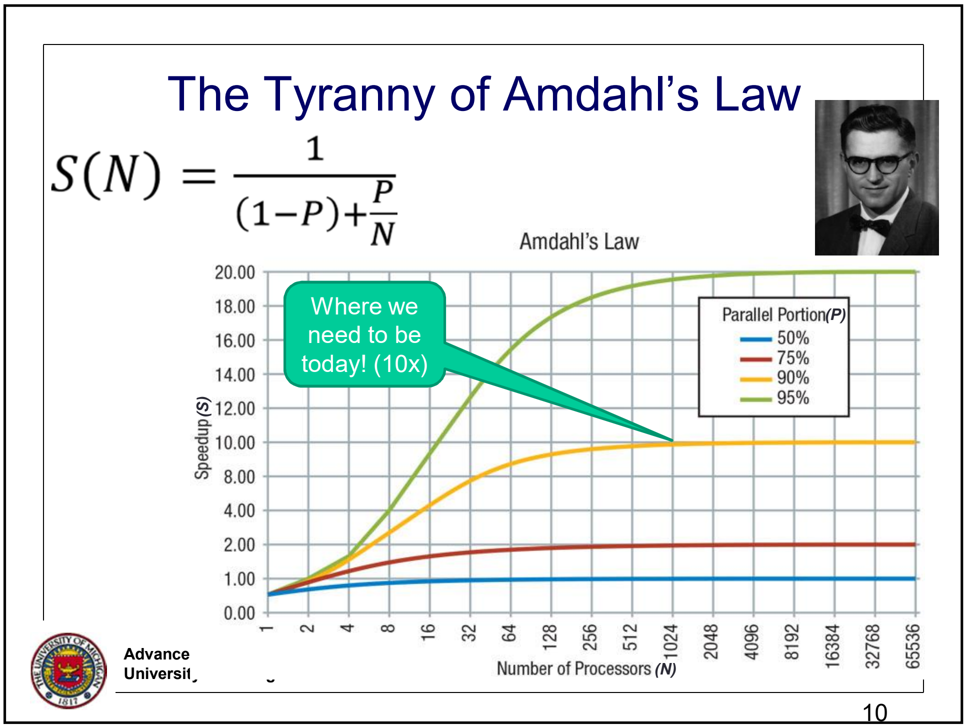
1

Advanced Computer Architecture Laboratory
University of Michigan

Courtesy Michael Taylor @ UCSD

Application Specific Architectures
Todd Austin

9



The Tyranny of Amdahl's Law

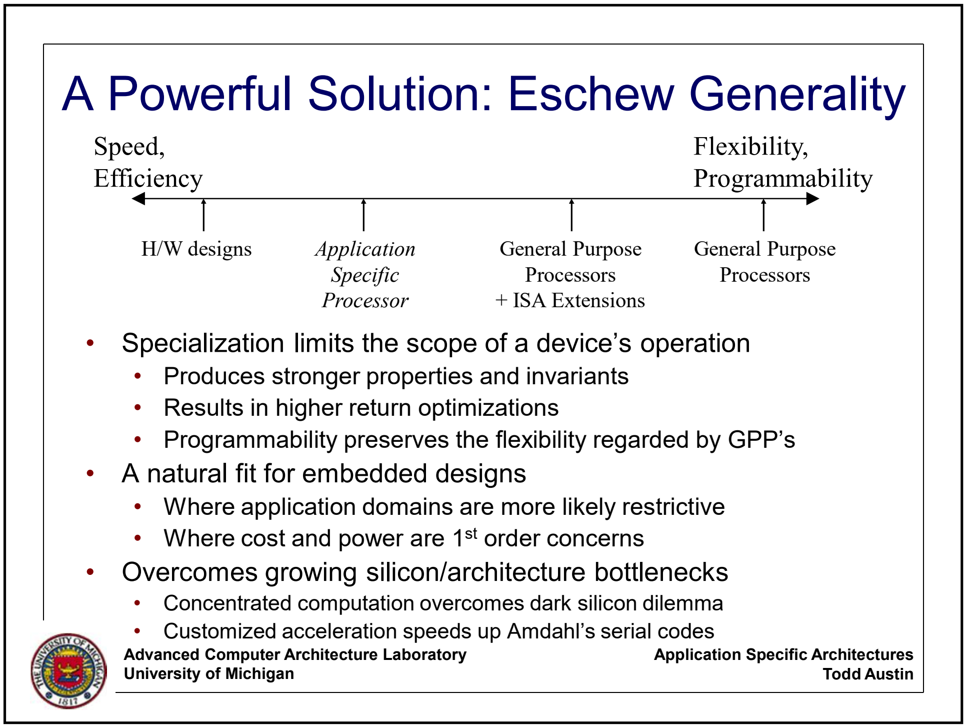
$$S(N) = \frac{1}{(1-P)+\frac{P}{N}}$$

Amdahl's Law

Where we need to be today! (10x)

Advance
Universit

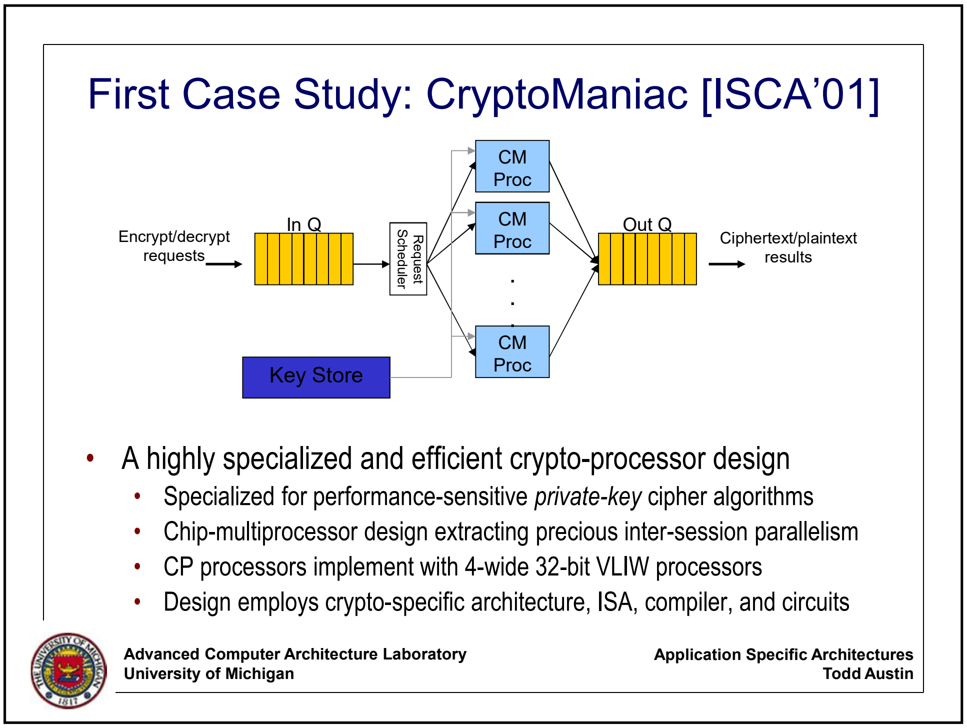
10



A Powerful Solution: Eschew Generality

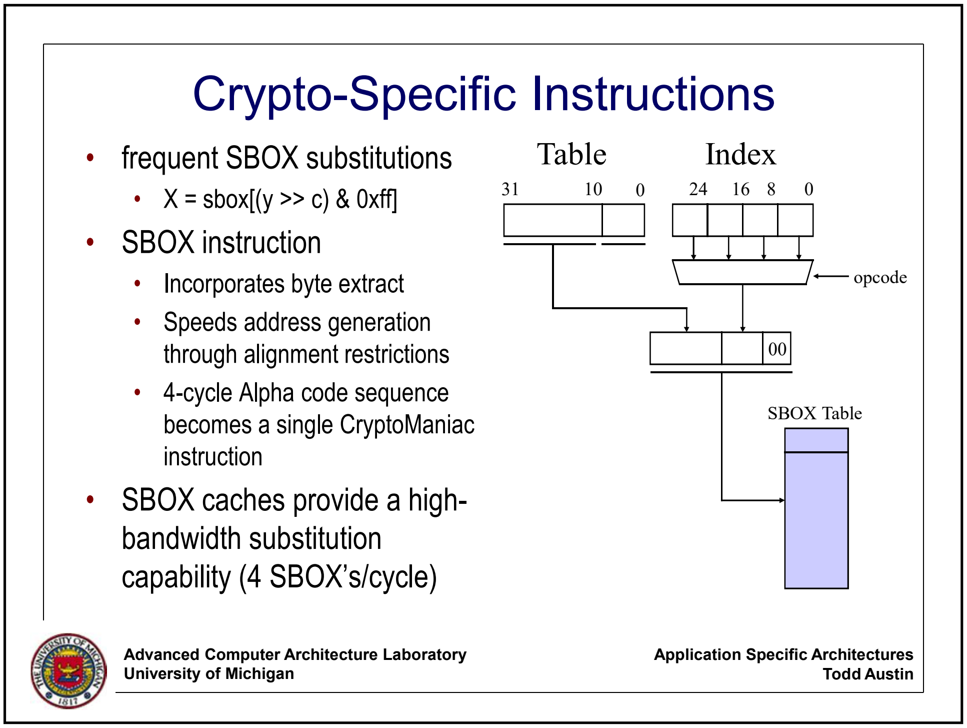
Speed,
Efficiency
Flexibility,
Programmability
H/W designs
Application Specific Processor
General Purpose Processors + ISA Extensions
General Purpose Processors

- Specialization limits the scope of a device's operation
  - Produces stronger properties and invariants
  - Results in higher return optimizations
  - Programmability preserves the flexibility regarded by GPP's
- A natural fit for embedded designs
  - Where application domains are more likely restrictive
  - Where cost and power are 1st order concerns
- Overcomes growing silicon/architecture bottlenecks
  - Concentrated computation overcomes dark silicon dilemma
  - Customized acceleration speeds up Amdahl's serial codes



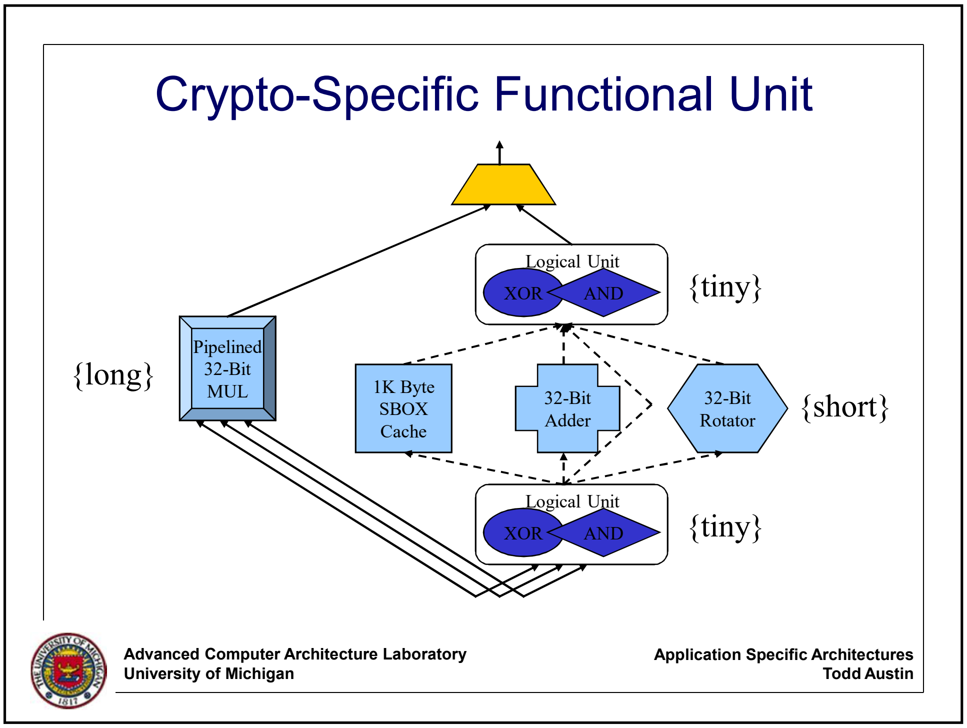
First Case Study: CryptoManiac [ISCA'01]

- A highly specialized and efficient crypto-processor design
  - Specialized for performance-sensitive *private-key* cipher algorithms
  - Chip-multiprocessor design extracting precious inter-session parallelism
  - CP processors implement with 4-wide 32-bit VLIW processors
  - Design employs crypto-specific architecture, ISA, compiler, and circuits

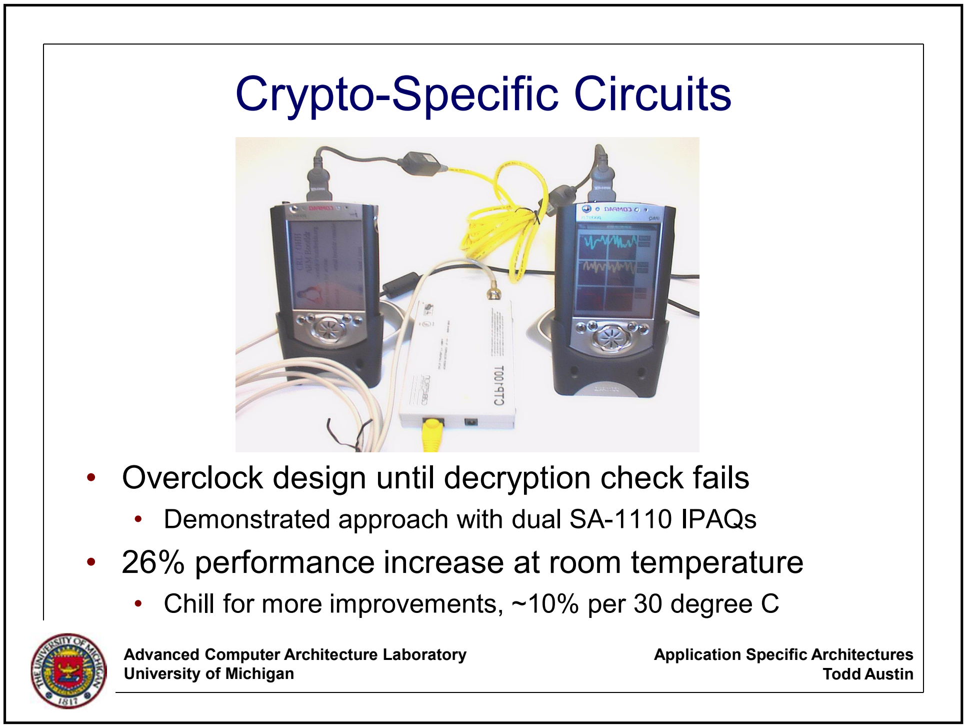


Crypto-Specific Instructions

- frequent SBOX substitutions
  - X = sbox[(y >> c) & 0xff]
- SBOX instruction
  - Incorporates byte extract
  - Speeds address generation through alignment restrictions
  - 4-cycle Alpha code sequence becomes a single CryptoManiac instruction
- SBOX caches provide a high-bandwidth substitution capability (4 SBOX's/cycle)

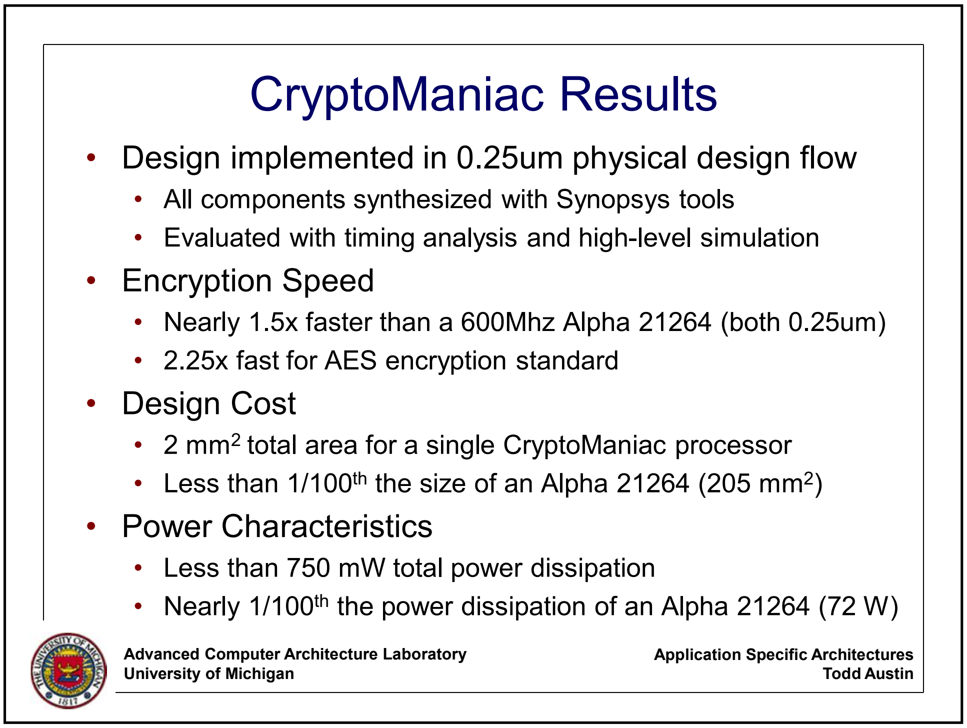


Crypto-Specific Functional Unit



Crypto-Specific Circuits

- Overclock design until decryption check fails
  - Demonstrated approach with dual SA-1110 IPAQs
- 26% performance increase at room temperature
  - Chill for more improvements, ~10% per 30 degree C

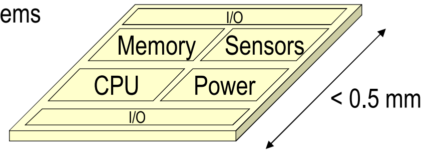


CryptoManiac Results

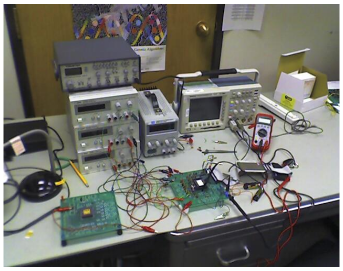
- Design implemented in 0.25um physical design flow
  - All components synthesized with Synopsys tools
  - Evaluated with timing analysis and high-level simulation
- Encryption Speed
  - Nearly 1.5x faster than a 600Mhz Alpha 21264 (both 0.25um)
  - 2.25x fast for AES encryption standard
- Design Cost
  - 2 mm^2 total area for a single CryptoManiac processor
  - Less than 1/100th the size of an Alpha 21264 (205 mm^2)
- Power Characteristics
  - Less than 750 mW total power dissipation
  - Nearly 1/100th the power dissipation of an Alpha 21264 (72 W)

Second Case Study: Subliminal Systems [ISCA'05]

- Project goals
  - Explore area-constrained low-energy systems
  - Develop 100% silicon platforms
  - Target form factors below 1 mm^3

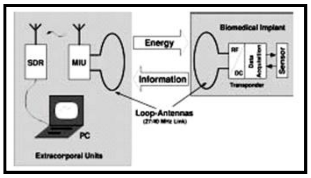


- Technology Developments
  - Subthreshold-voltage processors and memories
  - Robust subthreshold circuit/cell designs
  - Compact integrated wireless interfaces
  - Energy scavenging technologies
  - Sensor designs

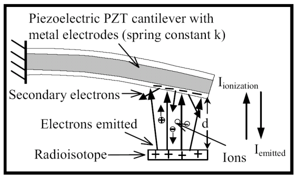


Energy Efficiency: A Key Requirement

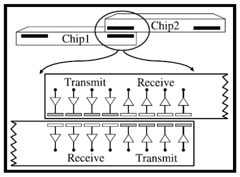
- They live on a limited amount of energy generated from a small battery or scavenged from the environment.
- Traditionally the communication component is the most power-hungry element of the system. However, new trends are emerging:



Passive telemetry

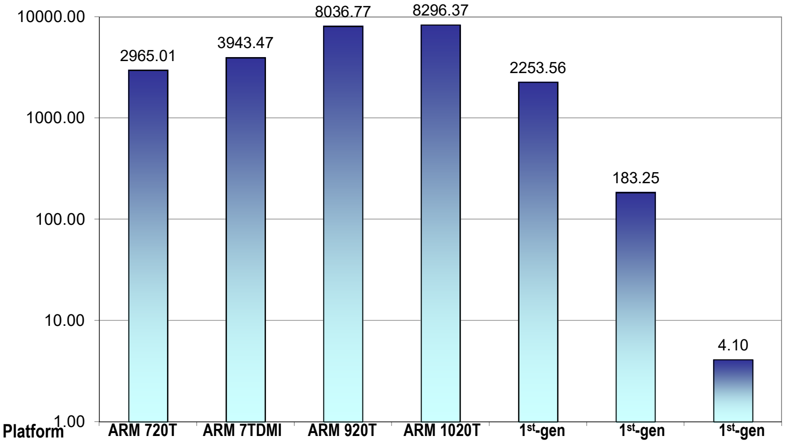


Self-powered RF



Proximity comm.

Performance of Various Platforms



Platform	ARM 720T	ARM 7TDMI	ARM 920T	ARM 1020T	1^{st}-gen	1^{st}-gen	1^{st}-gen
Voltage (V)	**1.2**	**1.2**	**1.2**	**1.2**	**1.2**	**0.5**	**0.232**
Speed (Hz)	**100M**	**133M**	**250M**	**325M**	**114M**	**9M**	**168k**

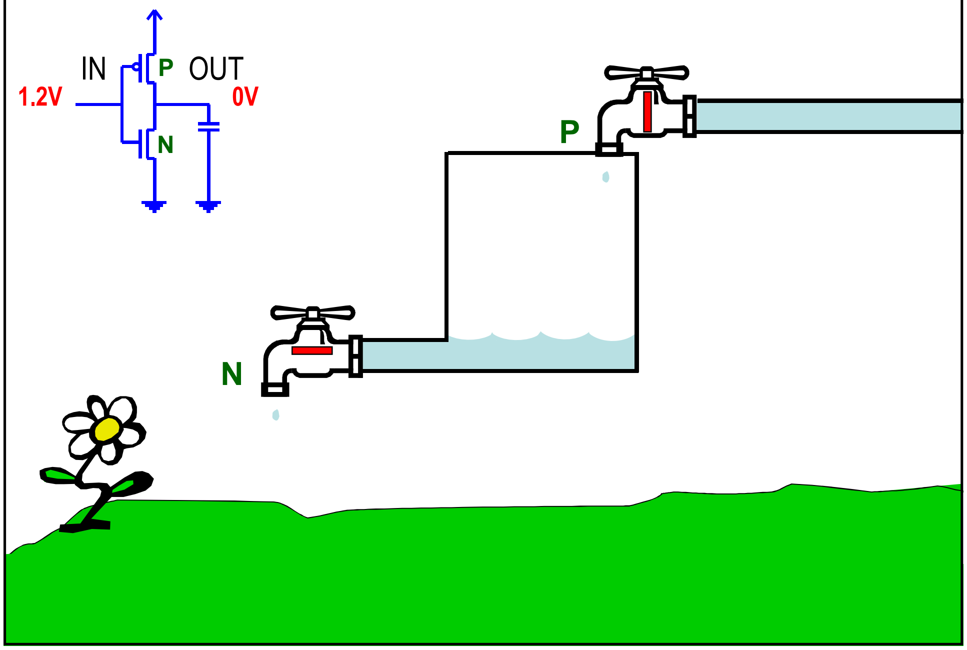
xRT rating: how many times faster than real-time the processor can handle the worst-case data stream rate on the most computationally intensive sensor benchmark

The Basics of Subthreshold Circuit Operation

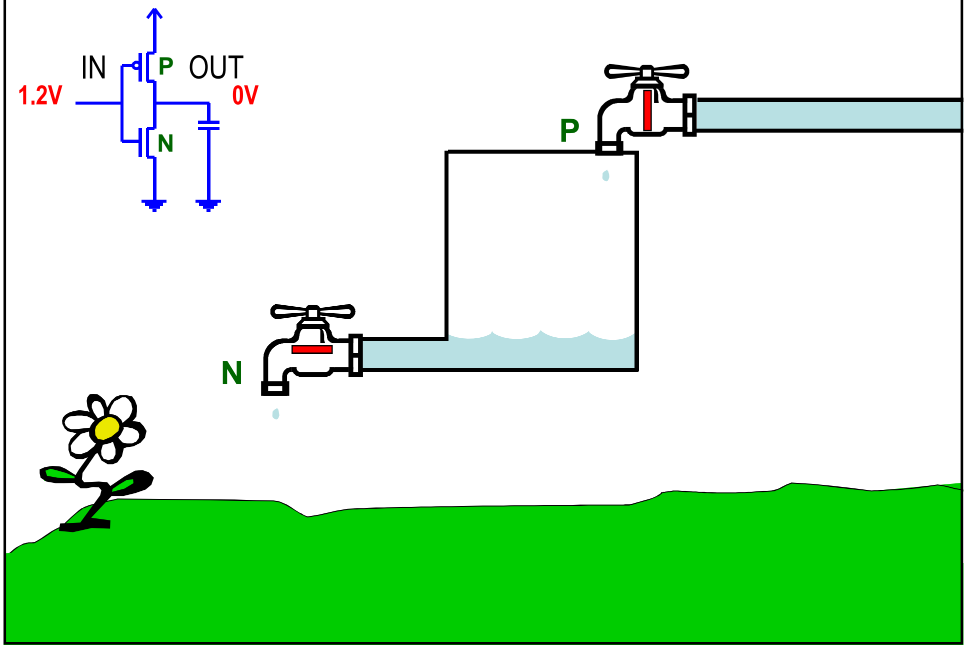
A Short Animation by Leyla Nazhandali

Episode 1: Inverter operation in superthreshold domain

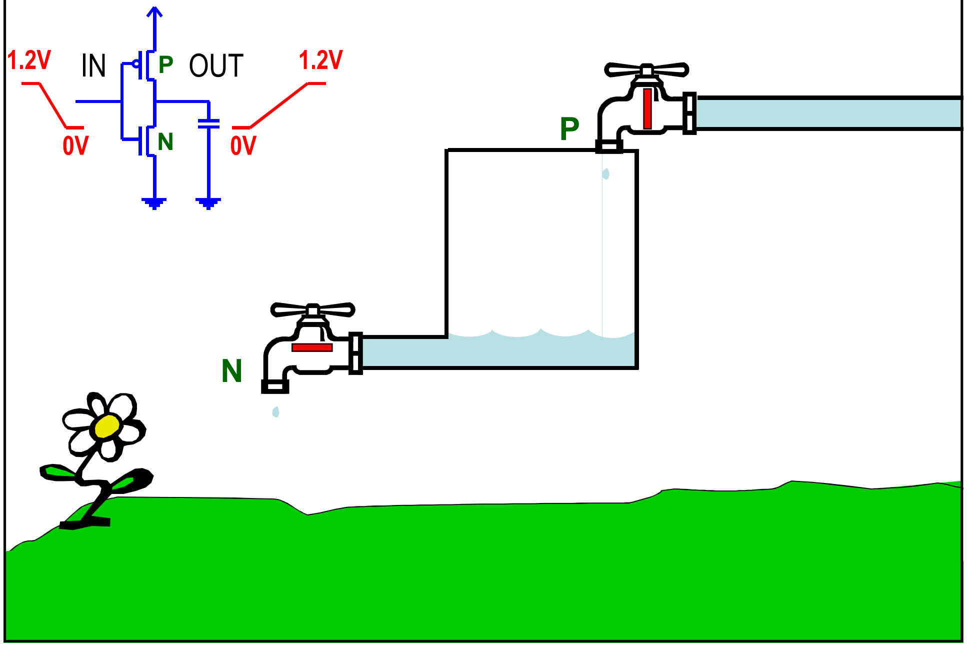
Superthreshold



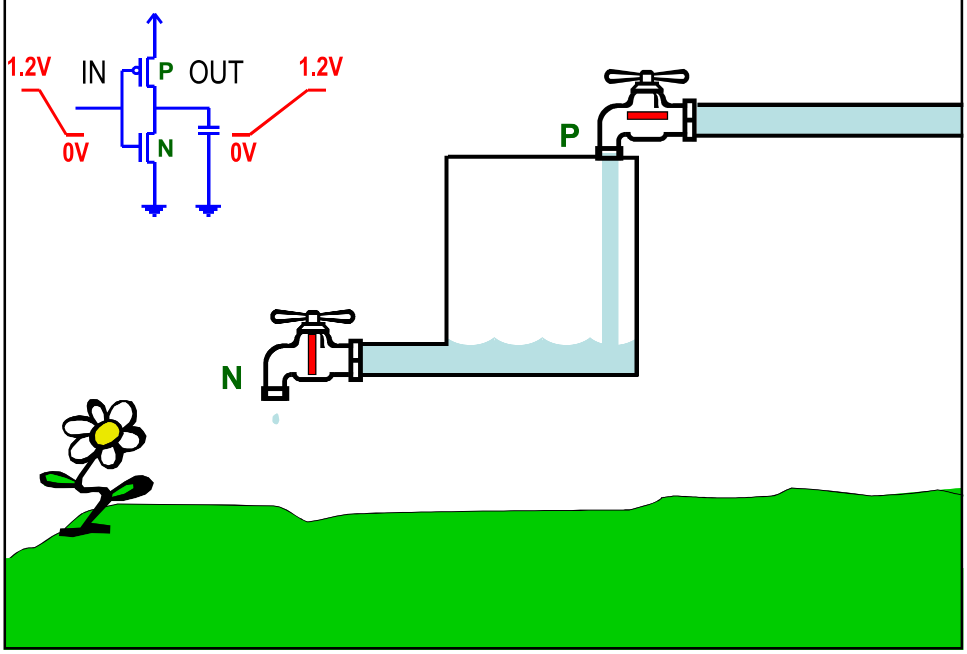
Superthreshold



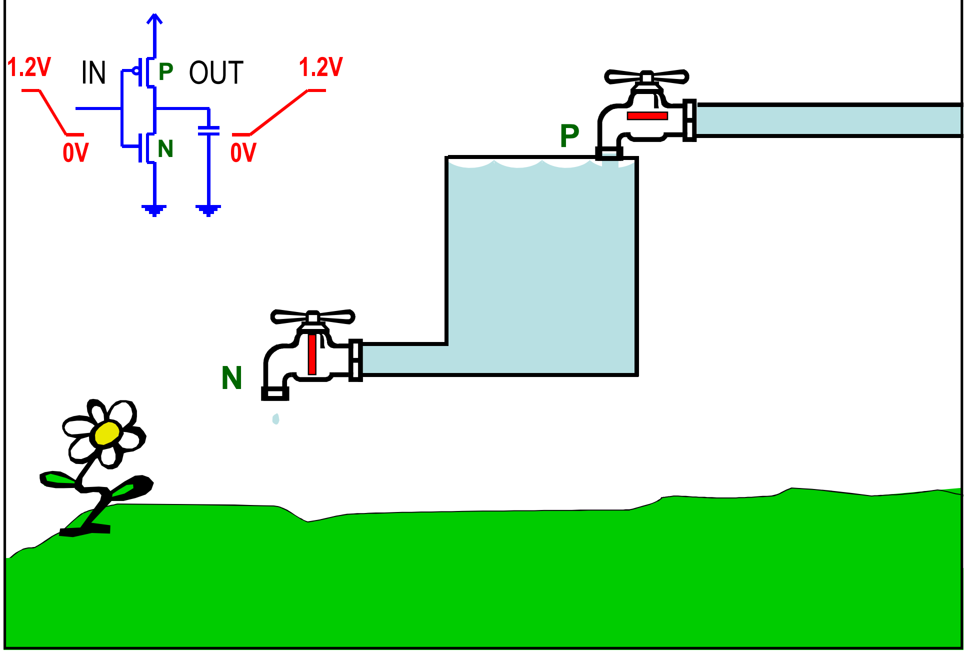
Superthreshold



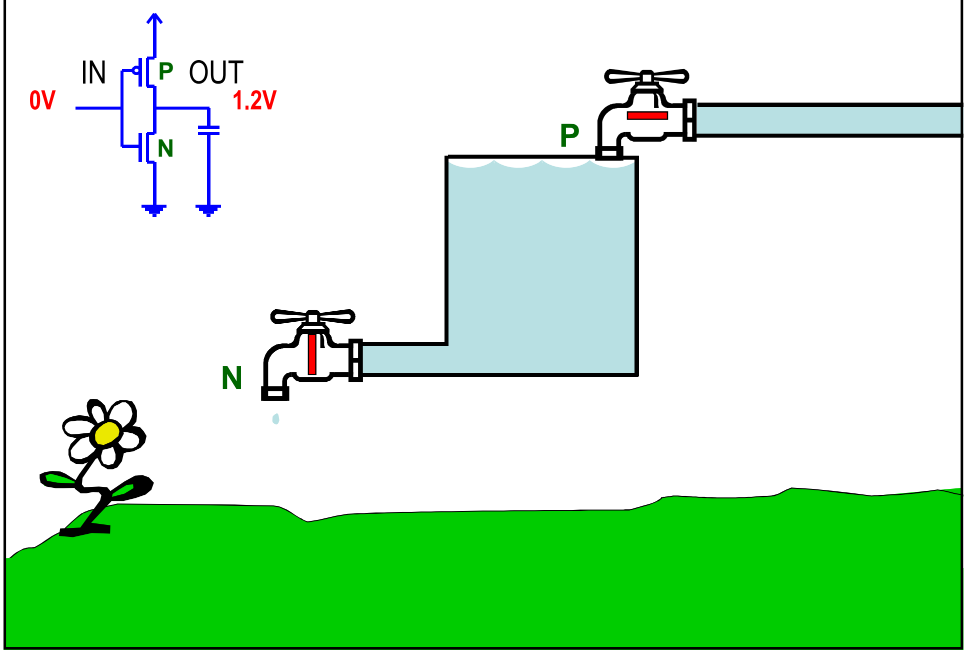
Superthreshold



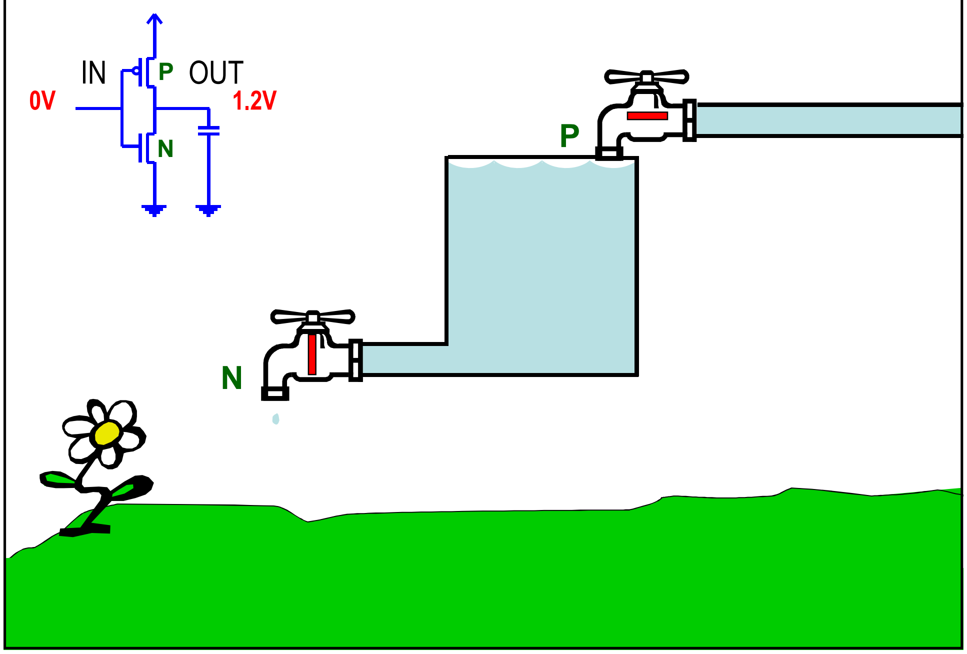
Superthreshold



Superthreshold

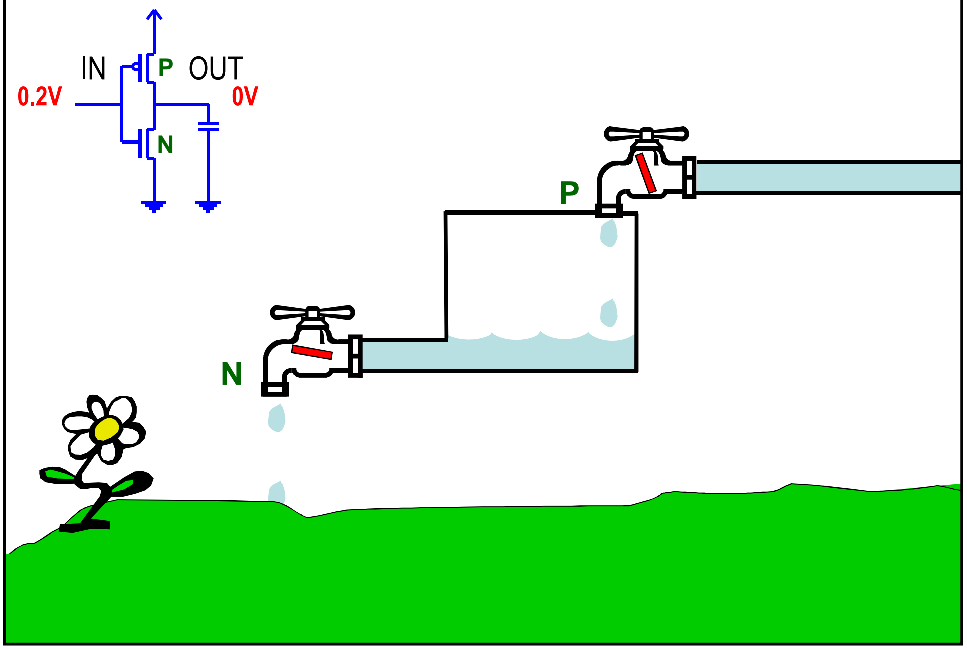


Superthreshold

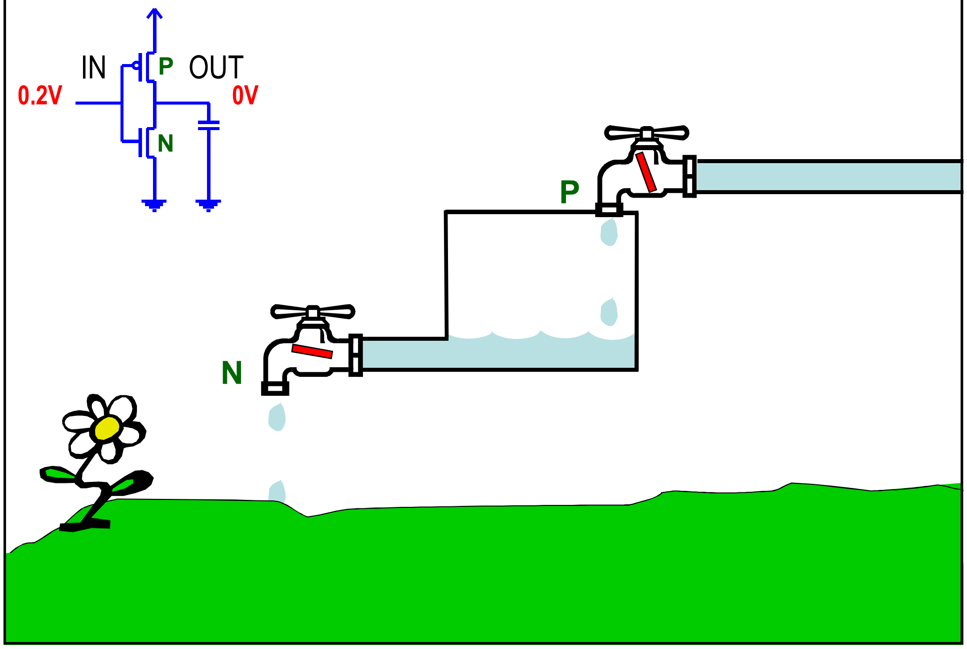


Episode 2: Inverter operation in subthreshold domain

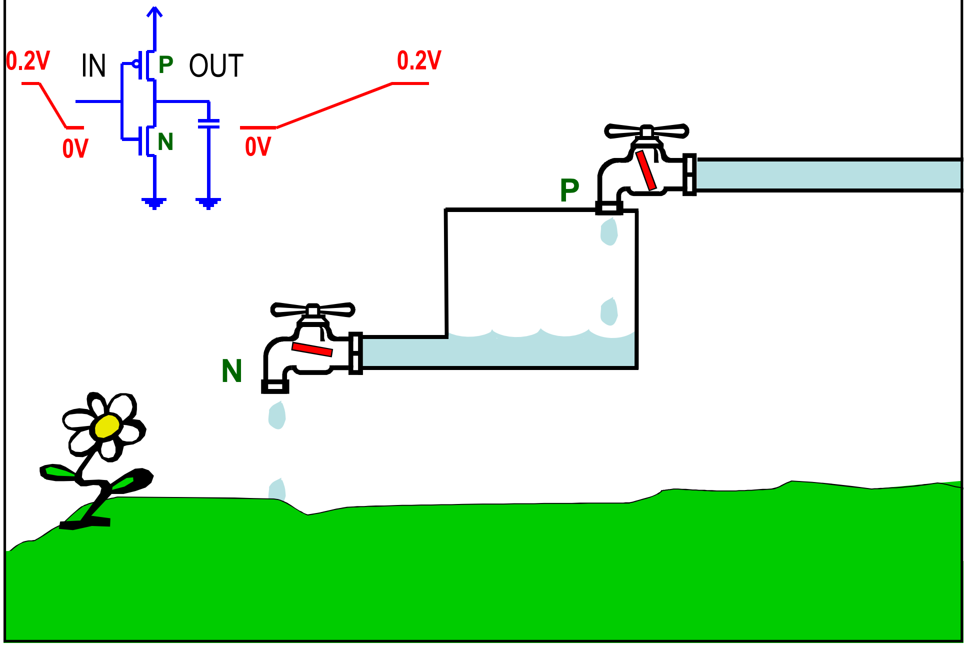
Subthreshold



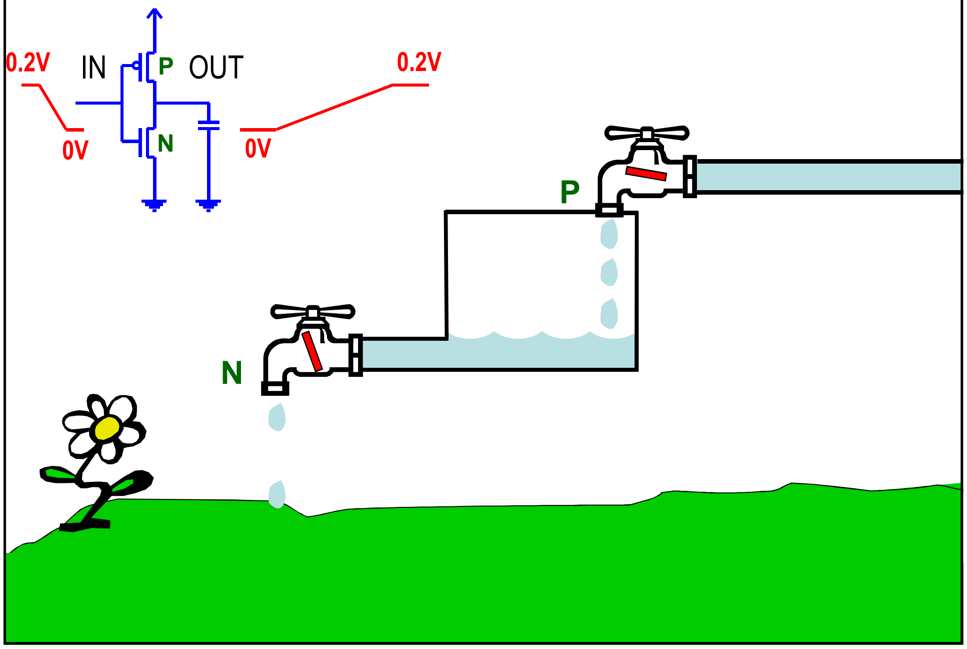
Subthreshold



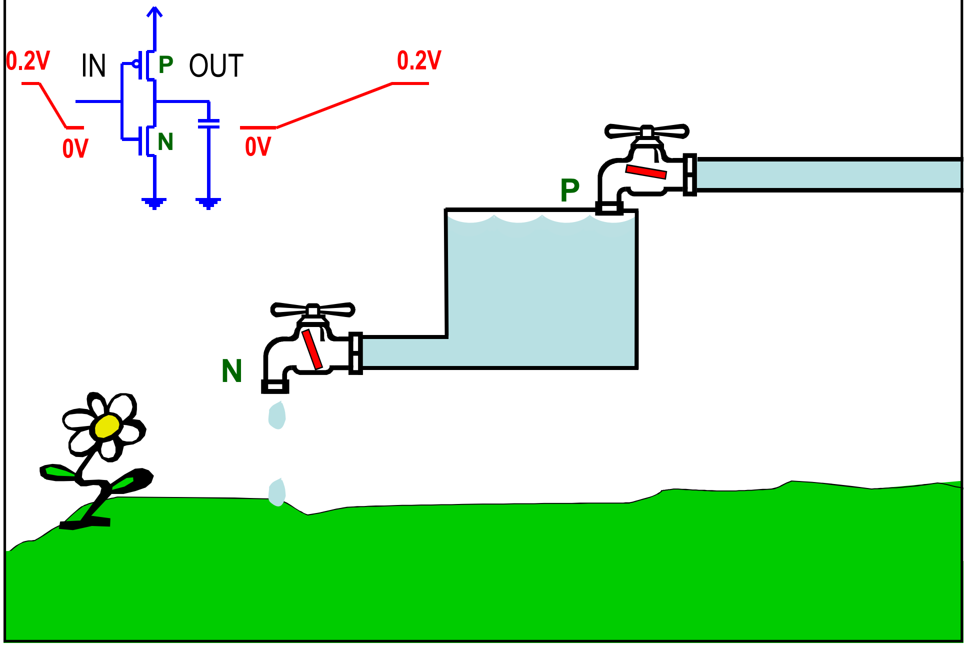
Subthreshold



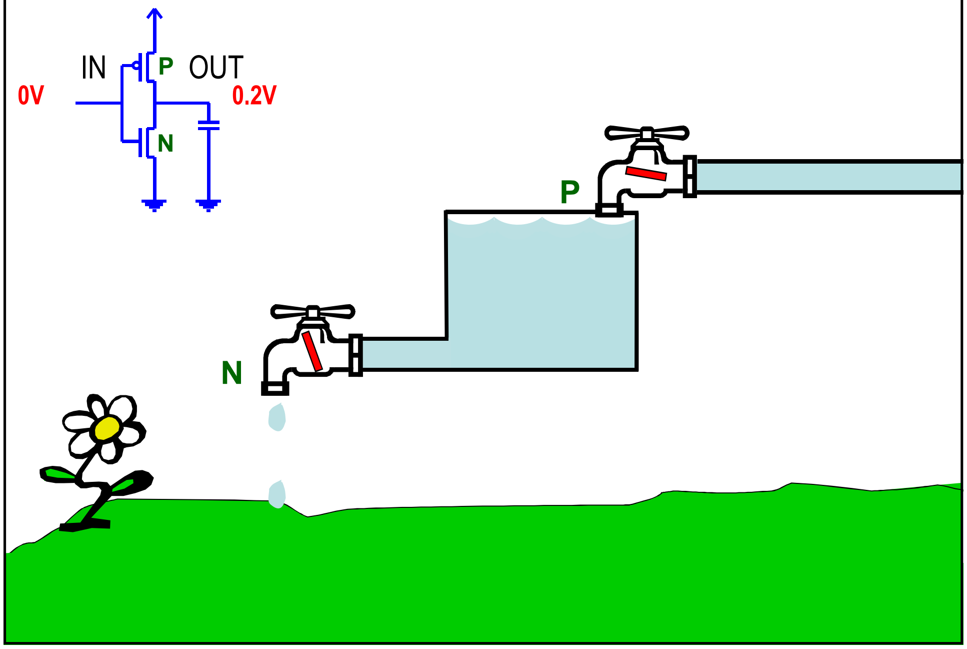
Subthreshold



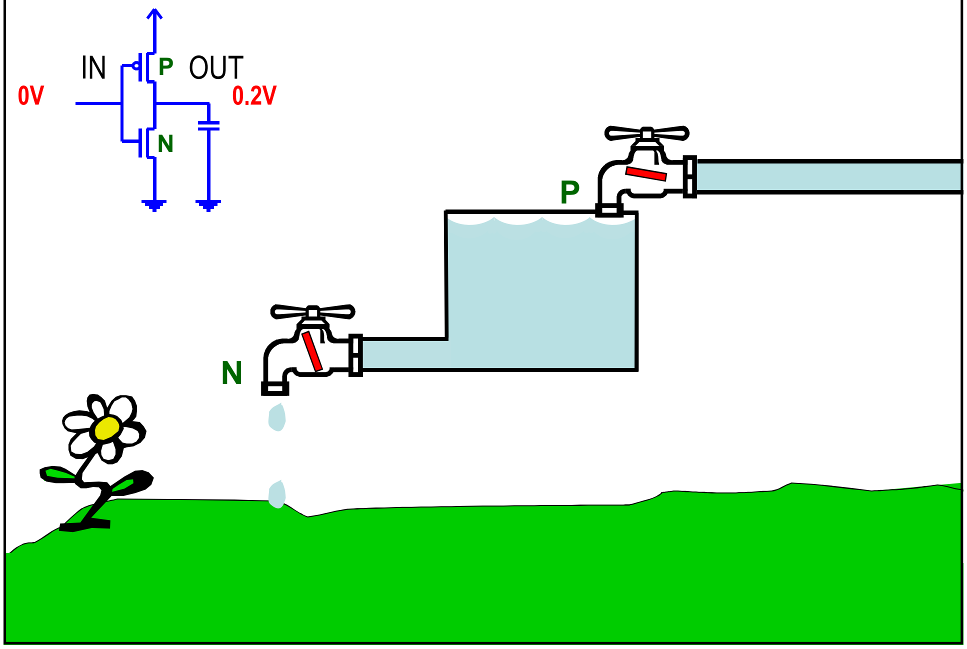
Subthreshold



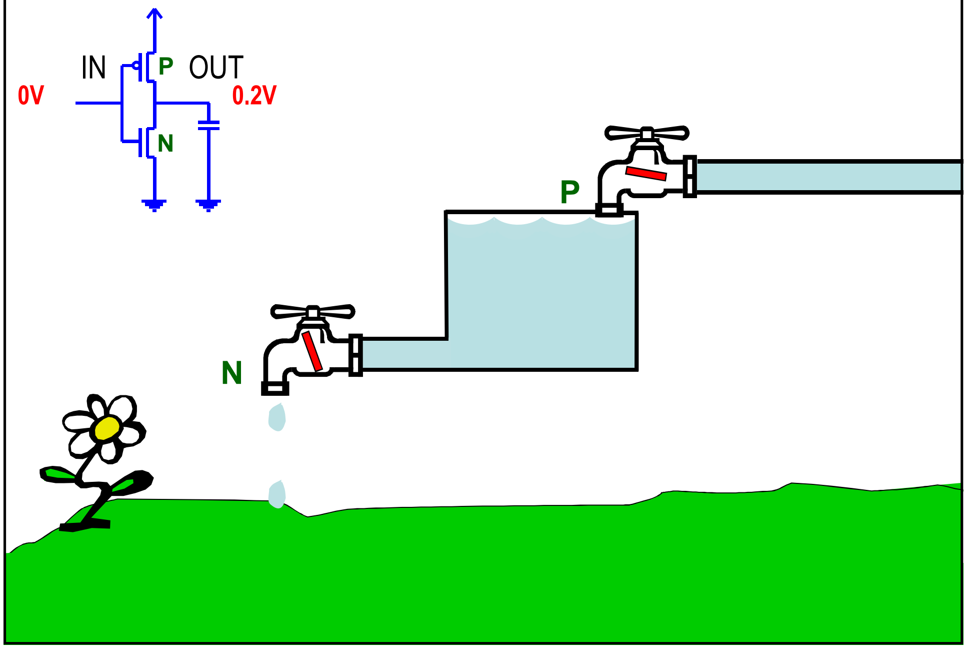
Subthreshold



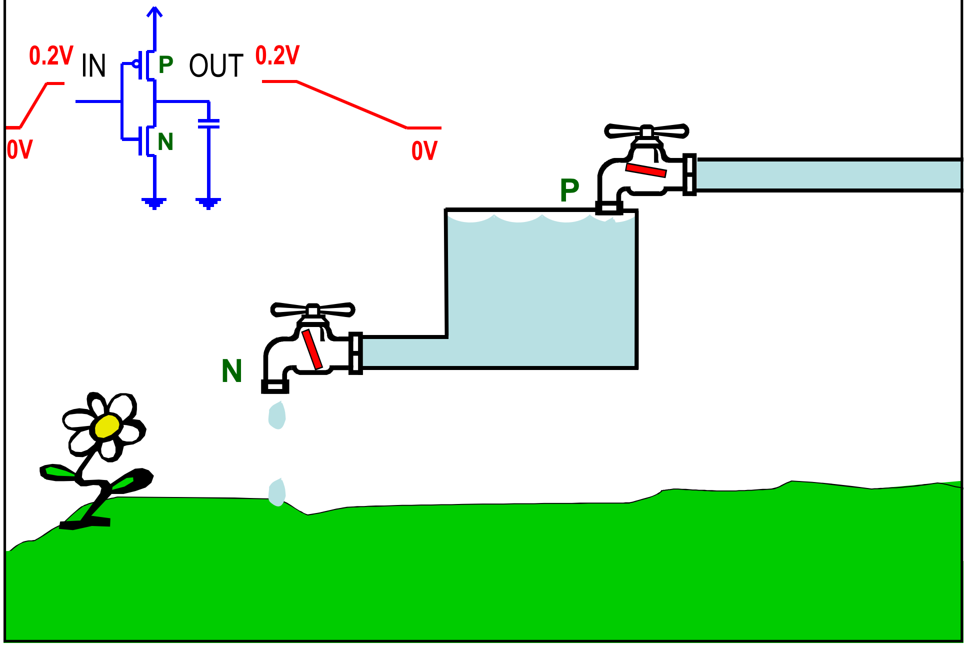
Subthreshold



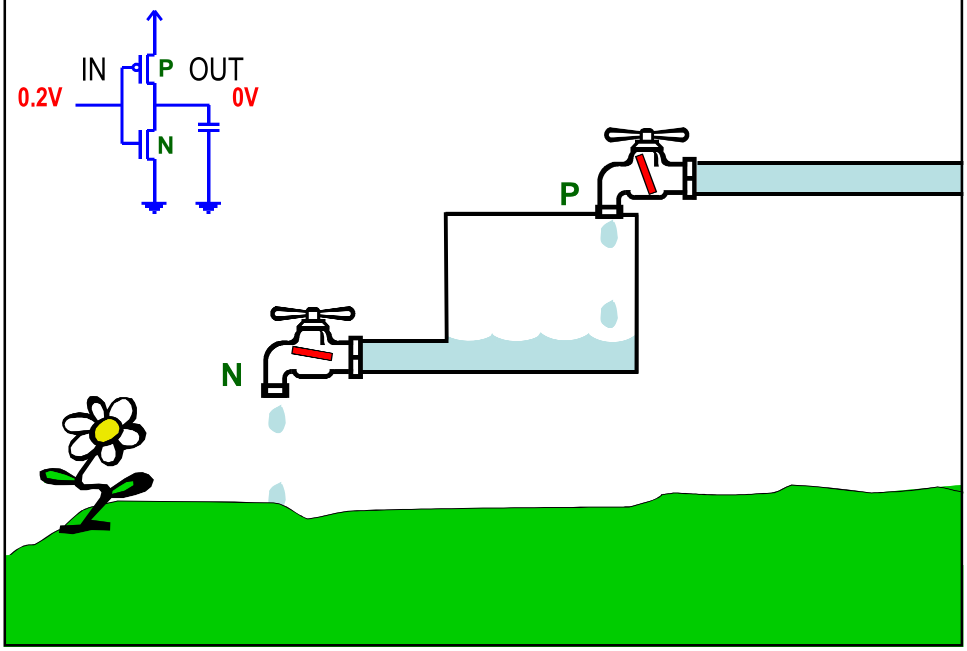
Subthreshold



Subthreshold



Subthreshold



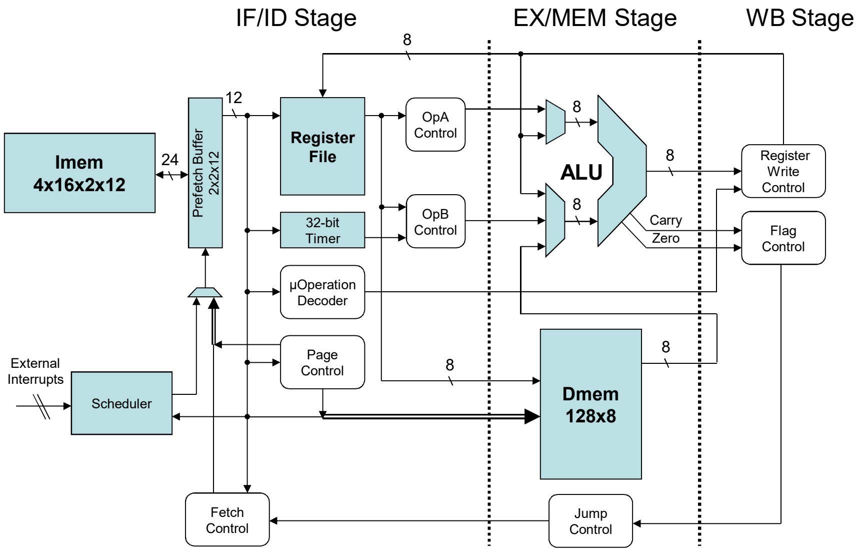
Summary from Architecture Study

- We studied 21 different processors experimenting with following options:
  - Number of stages
  - w/ vs. w/o instruction prefetch buffer
  - w/ vs. w/o explicit register file
  - Harvard vs. Von-Neumann architecture
- To minimize energy at subthreshold voltages, architects must:

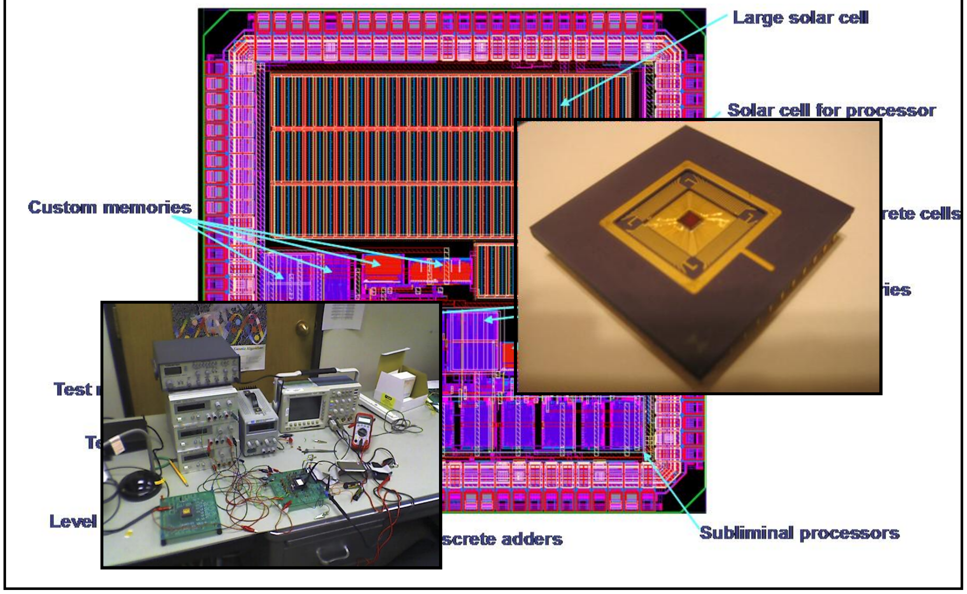
Minimize area	⇨	**To reduce leakage energy per cycle**
Maximize Transistor utility	⇨	**To reduce V_{min} and energy per cycle**
Minimize CPI	⇨	**To reduce Energy per instruction**

- The memory comprises the single largest factor of leakage energy, as such, efficient designs must reduce memory storage requirements.

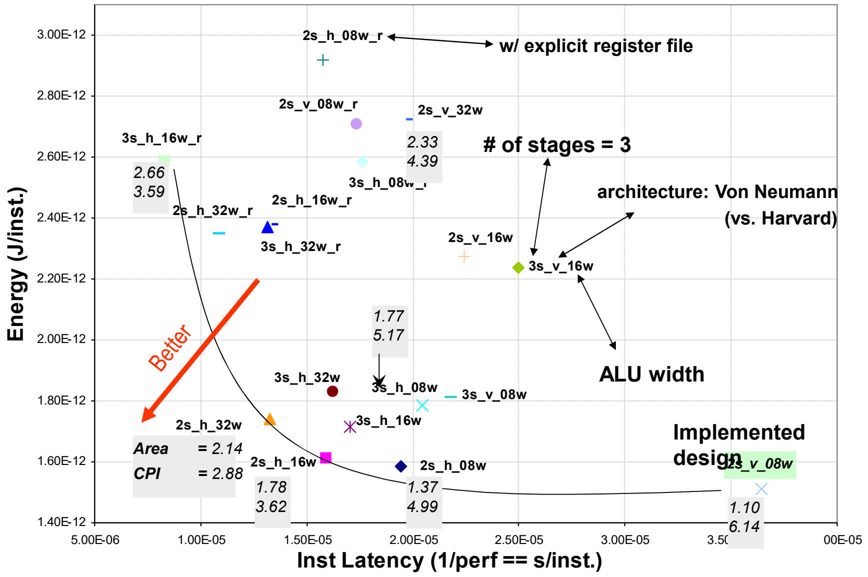
Microarchitecture Overview



First Subliminal Chip

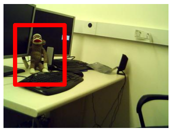


Pareto Analysis for Several Processors

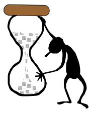


Third Case Study: Taking Computer Vision Mobile

- Embedded mobile computation on the rise
  - Smart Phones, Tablets
  - Improved sensors
    - High megapixel cameras, HD video
  - New capabilities from new sensors



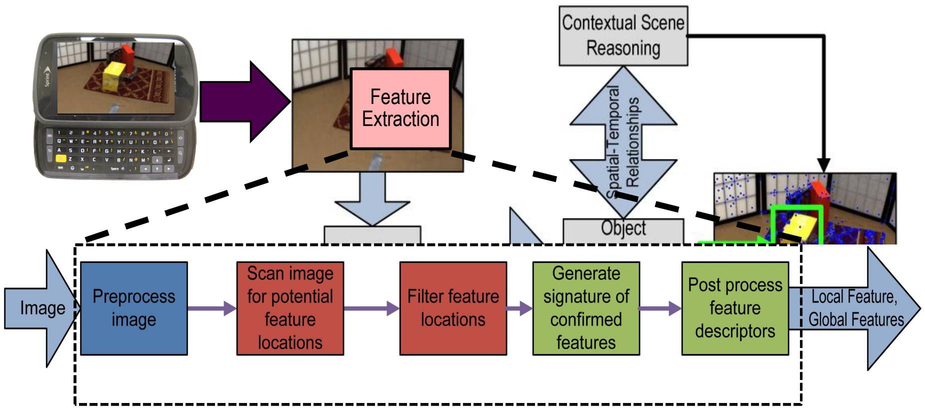
  - There is a need for near real time computation
    - Users don’t want to wait



- Why not use the cloud?
  - High latency
  - Bandwidth Limits
  - Reliability

Computer Vision

Typical computer vision pipeline



Feature Extraction

Contextual Scene Reasoning

Spatial-Temporal Relationships

Object

Image

Preprocess image

Scan image for potential feature locations

Filter feature locations

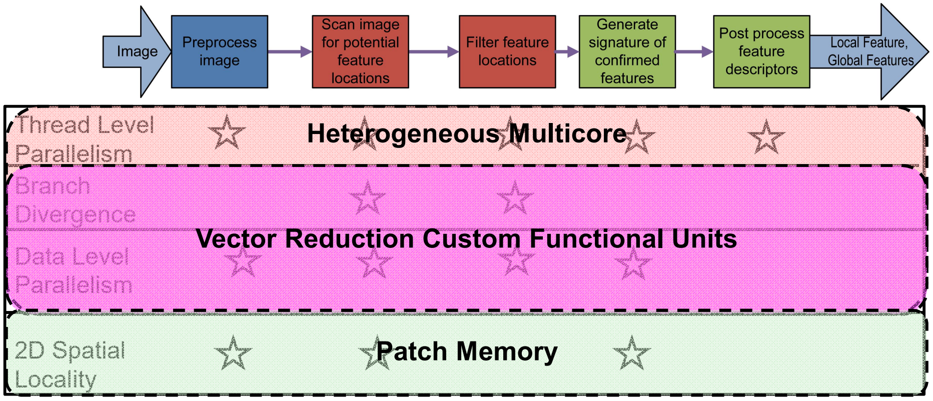
Generate signature of confirmed features

Post process feature descriptors

Local Feature, Global Features

Feature Extraction Characteristics

- 3 Algorithms
  - FAST – corner detection
  - HoG – general object shape detector
  - SIFT – specific object/blob detector



Image

Preprocess image

Scan image for potential feature locations

Filter feature locations

Generate signature of confirmed features

Post process feature descriptors

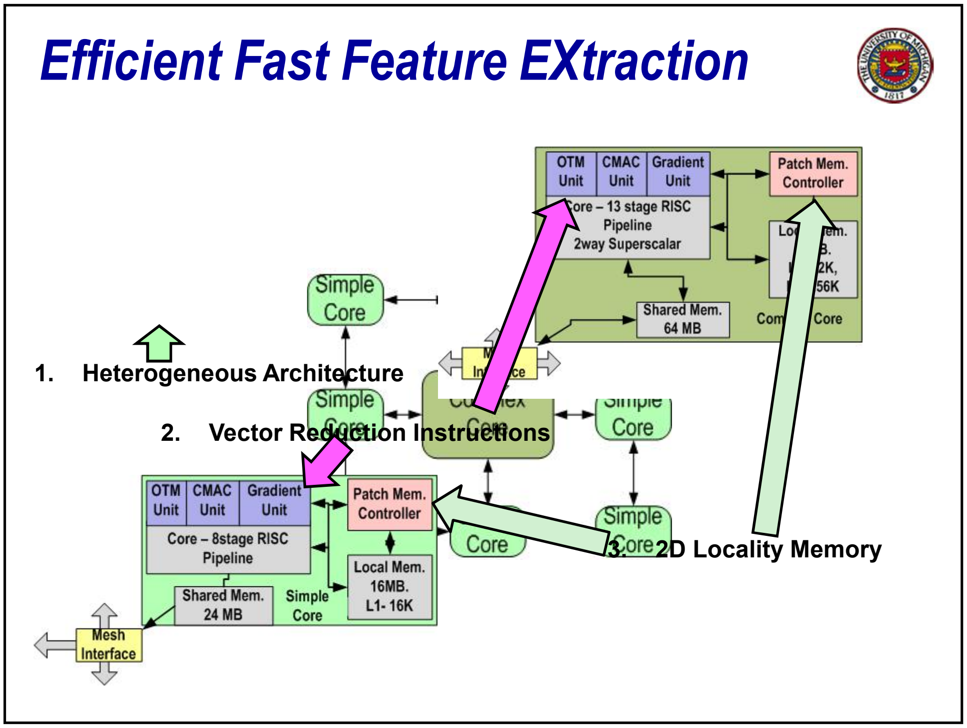
Local Feature, Global Features

Thread Level Parallelism ☆ ☆ ☆ ☆ ☆ — Heterogeneous Multicore

Branch Divergence ☆ ☆

Data Level Parallelism ☆ ☆ ☆ ☆ — Vector Reduction Custom Functional Units

2D Spatial Locality ☆ ☆ ☆ — Patch Memory

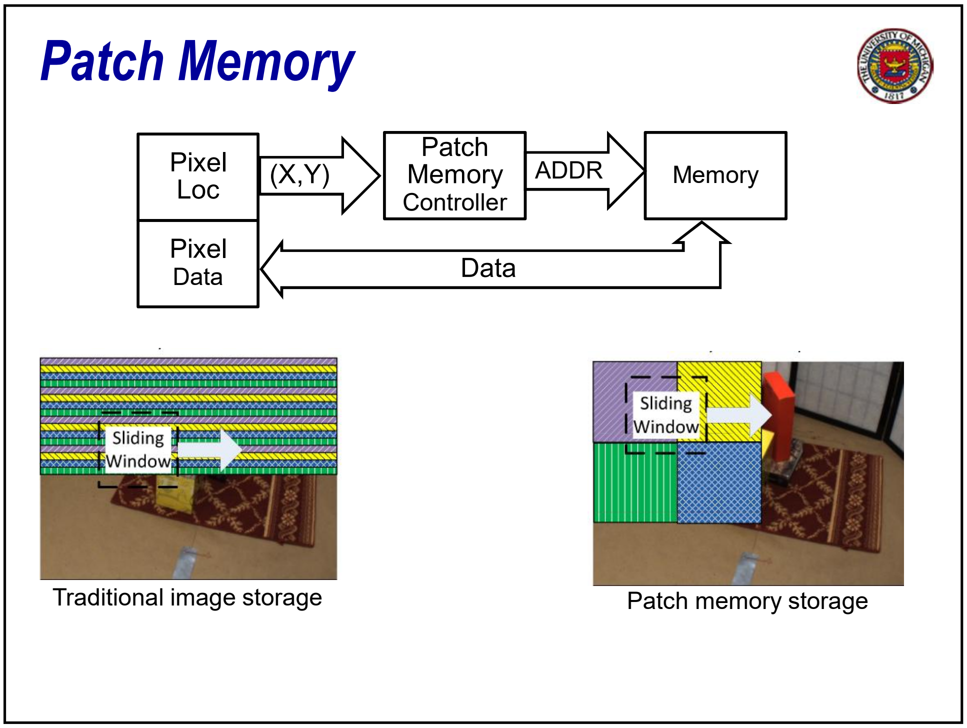


Efficient Fast Feature EXtraction

1. Heterogeneous Architecture

2. Vector Reduction Instructions

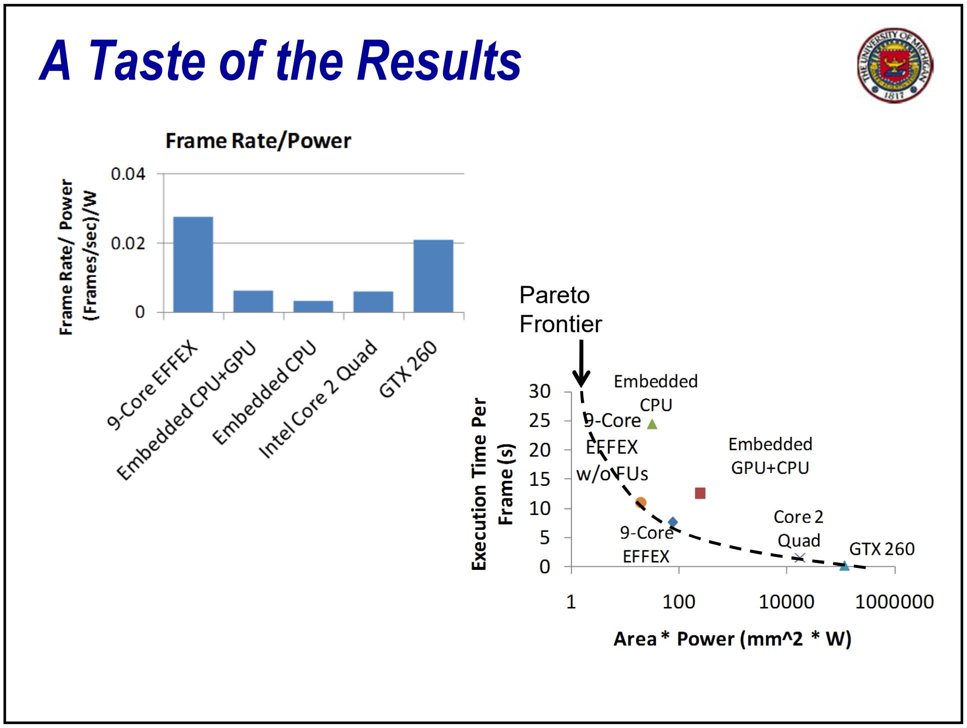
3. 2D Locality Memory



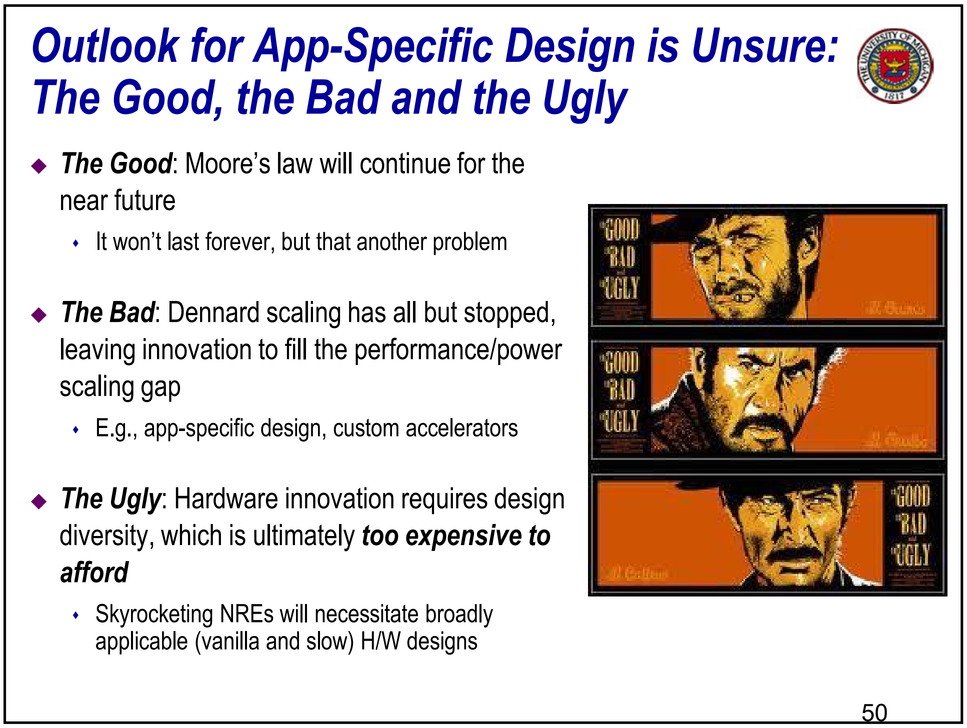
Patch Memory

Traditional image storage

Patch memory storage



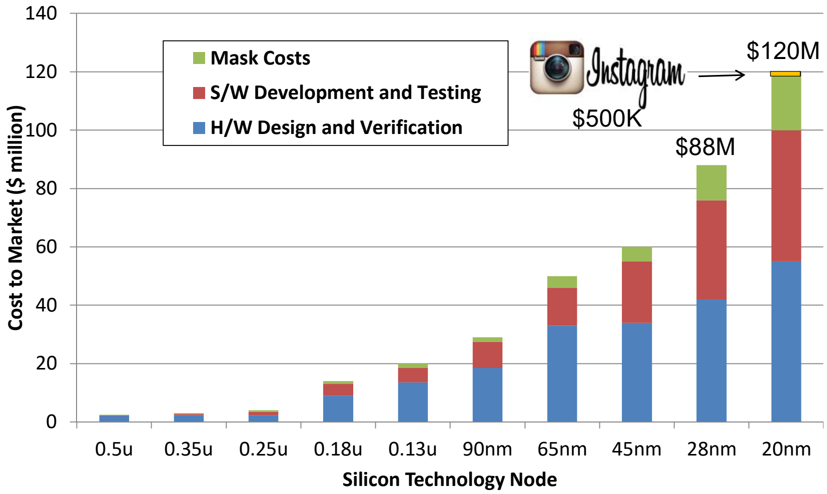
A Taste of the Results



Outlook for App-Specific Design is Unsure: The Good, the Bad and the Ugly

- ***The Good***: Moore’s law will continue for the near future
  - It won’t last forever, but that another problem
- ***The Bad***: Dennard scaling has all but stopped, leaving innovation to fill the performance/power scaling gap
  - E.g., app-specific design, custom accelerators
- ***The Ugly***: Hardware innovation requires design diversity, which is ultimately ***too expensive to afford***
  - Skyrocketing NREs will necessitate broadly applicable (vanilla and slow) H/W designs

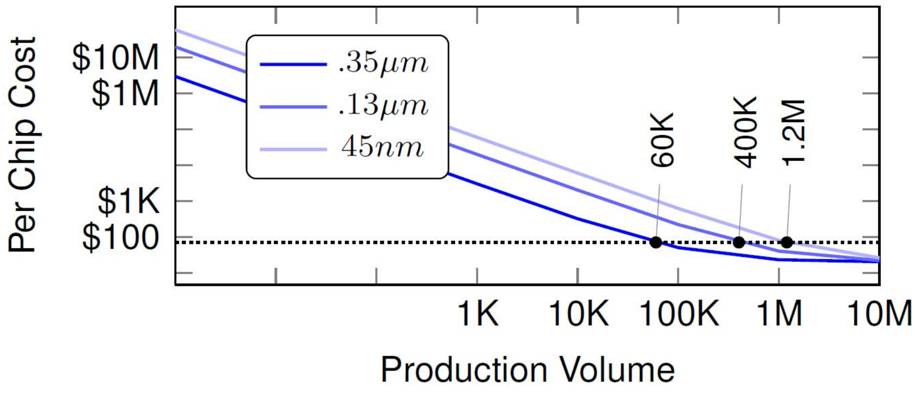
Design Costs Are Skyrocketing



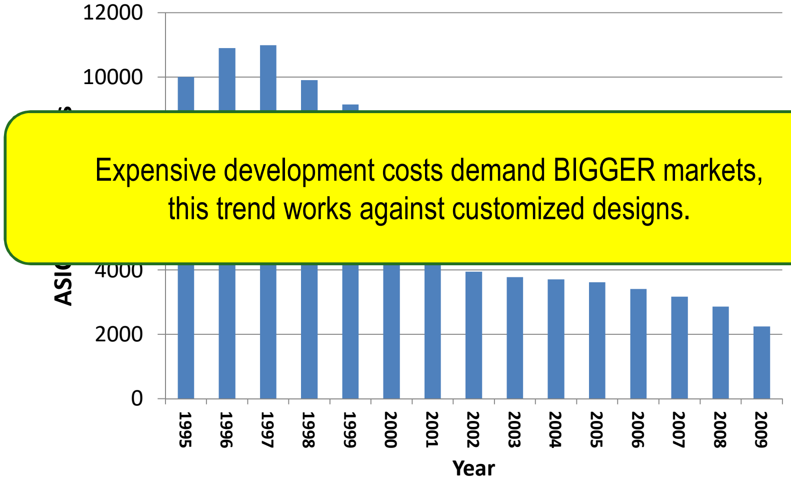
Source: International Business Strategies

High Costs Will be a Showstopper

- Heterogeneous designs often serve smaller markets



Outcome: "Nanodiversity" is Dwindling

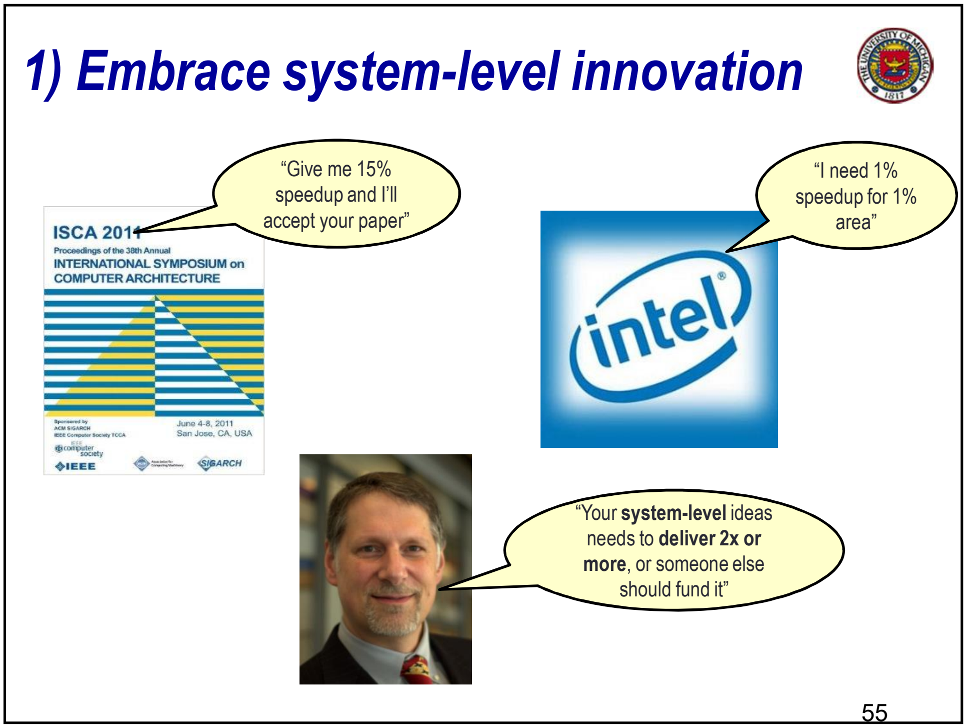


Expensive development costs demand BIGGER markets,
this trend works against customized designs.

Source: Gartner Group

The Remedy: Scale Innovation

- Ultimate goal: ***accelerate system architecture innovation*** and make it efficient and inexpensive enough that ***anyone can do it anywhere***
  - Approach #1: Embrace system-level innovation
  - Approach #2: Leverage technology advances on CMOS silicon
  - Approach #3: Reduce the cost to design custom hardware
  - Approach #4: Widen the applicability of custom hardware
  - Approach #5: Reduce the cost of manufacturing custom H/W



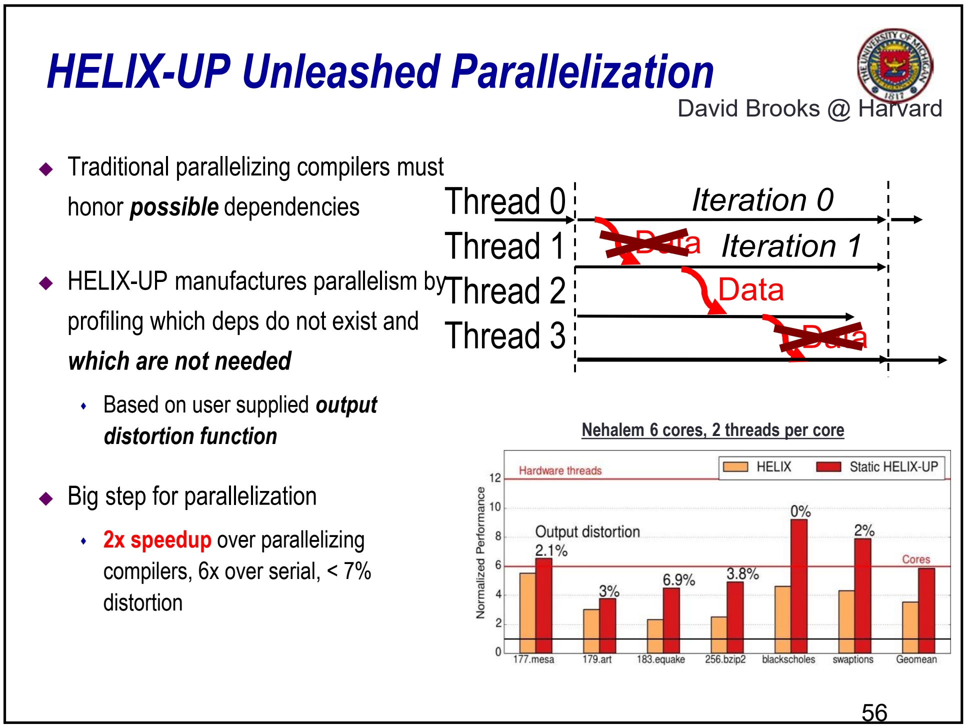
1) Embrace system-level innovation

"Give me 15% speedup and I'll accept your paper"

"I need 1% speedup for 1% area"

"Your **system-level** ideas needs to **deliver 2x or more**, or someone else should fund it"

ISCA 2011
Proceedings of the 38th Annual
INTERNATIONAL SYMPOSIUM on COMPUTER ARCHITECTURE
June 4-8, 2011
San Jose, CA, USA
Sponsored by
ACM SIGARCH
IEEE Computer Society TCCA
IEEE computer society
IEEE
SIGARCH
intel



HELIX-UP Unleashed Parallelization

David Brooks @ Harvard

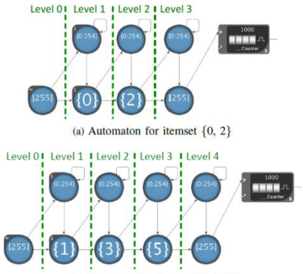
- Traditional parallelizing compilers must honor ***possible*** dependencies
- HELIX-UP manufactures parallelism by profiling which deps do not exist and ***which are not needed***
  - Based on user supplied ***output distortion function***
- Big step for parallelization
  - **2x speedup** over parallelizing compilers, 6x over serial, < 7% distortion

Nehalem 6 cores, 2 threads per core

Association Rule Mining with the Automata Processor

Kevin Skadron @ UVA

- Micron’s Automata processor
  - Implements FSMs at memory
  - Massively parallel with accelerators
- Mapped data-mining ARM rules to memory-based FSMs
  - ARM algorithms identify relationships between data elements
  - Implementations are often memory bottlenecked
- Big-data sets had big speedups
  - 90x+ over single CPU performance
  - **2-9x+ speedups** over CMPs and GPUs
- Joint effort with UVA and Micron

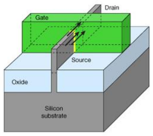


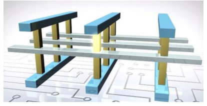
(a) Automaton for itemset {0, 2}

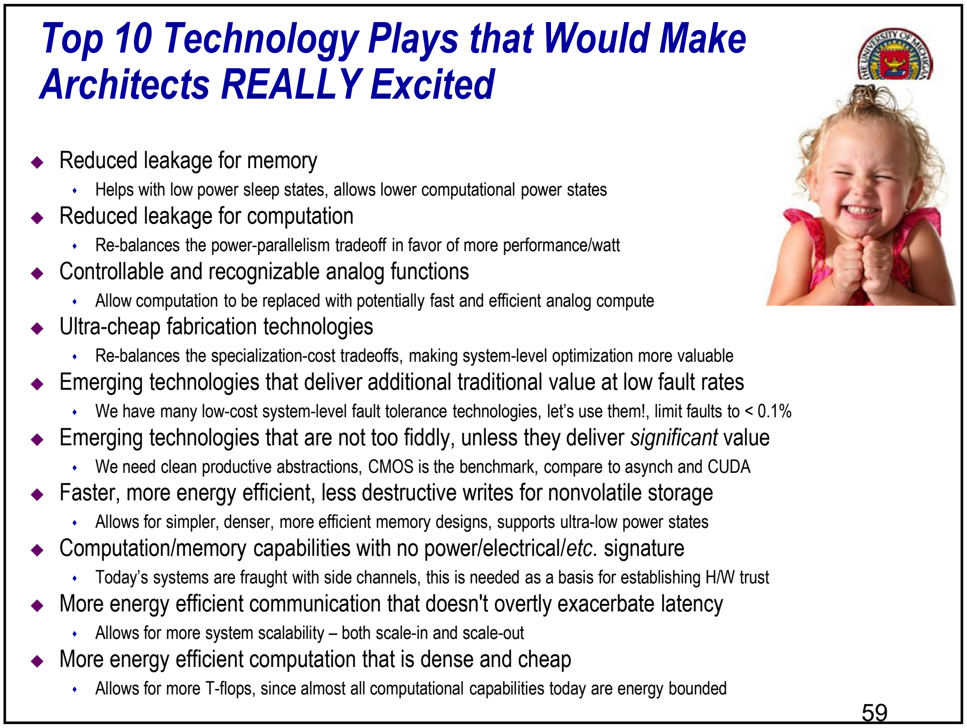
(b) Automaton for itemset {1, 3, 5}

2) Leverage technology advances on CMOS silicon

- Recent success: the reduced leakage and transient fault protection of FinFETs
- Upcoming: the density and durability of Intel/Micron’s XPoint memory technology
- Many additional opportunities possible: TFETs, CNTs, spin-tronics, novel materials, analog accelerators, etc…
- Key challenge: integration of non-silicon technologies
- Advice: to maximize benefits of these devices, architects need to work with device and materials researchers

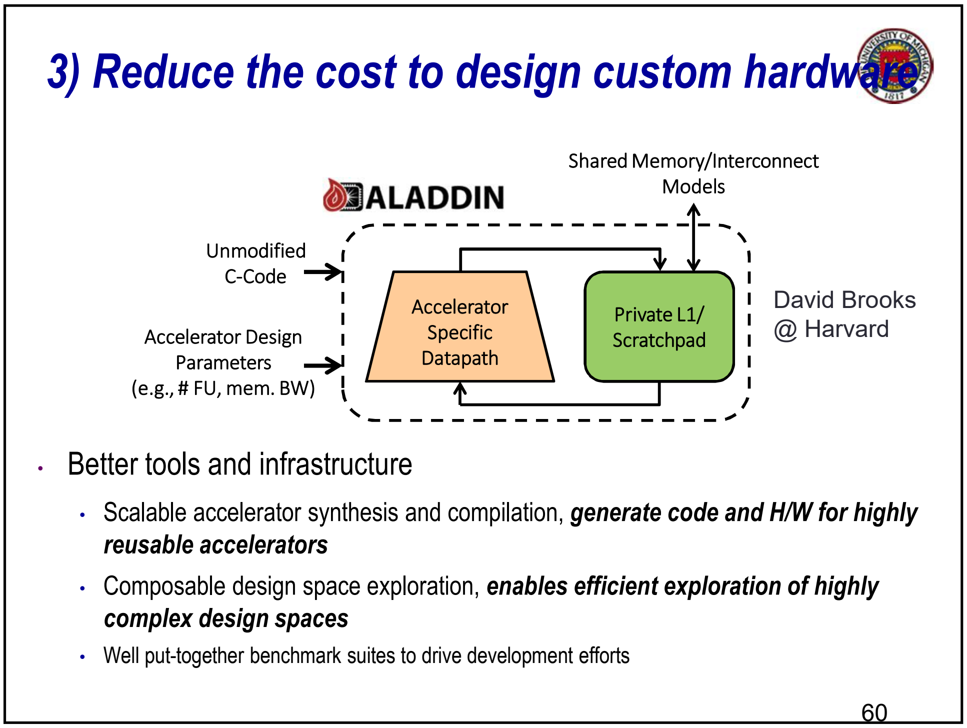






Top 10 Technology Plays that Would Make Architects REALLY Excited

- Reduced leakage for memory
  - Helps with low power sleep states, allows lower computational power states
- Reduced leakage for computation
  - Re-balances the power-parallelism tradeoff in favor of more performance/watt
- Controllable and recognizable analog functions
  - Allow computation to be replaced with potentially fast and efficient analog compute
- Ultra-cheap fabrication technologies
  - Re-balances the specialization-cost tradeoffs, making system-level optimization more valuable
- Emerging technologies that deliver additional traditional value at low fault rates
  - We have many low-cost system-level fault tolerance technologies, let's use them!, limit faults to < 0.1%
- Emerging technologies that are not too fiddly, unless they deliver *significant* value
  - We need clean productive abstractions, CMOS is the benchmark, compare to asynch and CUDA
- Faster, more energy efficient, less destructive writes for nonvolatile storage
  - Allows for simpler, denser, more efficient memory designs, supports ultra-low power states
- Computation/memory capabilities with no power/electrical/*etc*. signature
  - Today's systems are fraught with side channels, this is needed as a basis for establishing H/W trust
- More energy efficient communication that doesn't overtly exacerbate latency
  - Allows for more system scalability – both scale-in and scale-out
- More energy efficient computation that is dense and cheap
  - Allows for more T-flops, since almost all computational capabilities today are energy bounded

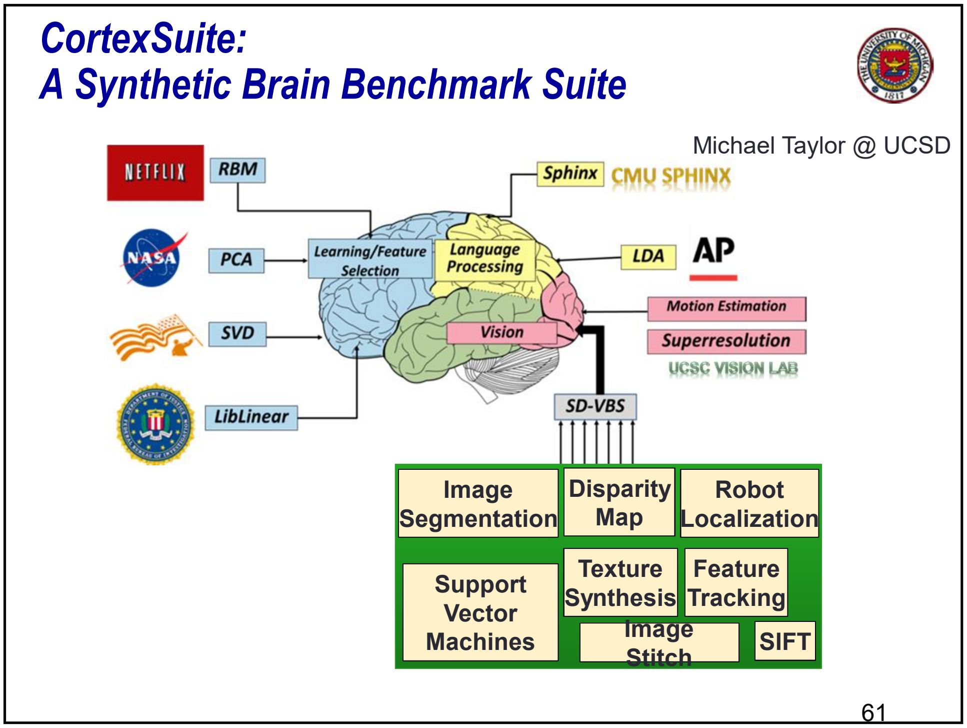


3) Reduce the cost to design custom hardware

ALADDIN
Shared Memory/Interconnect Models
Unmodified C-Code
Accelerator Design Parameters (e.g., # FU, mem. BW)
Accelerator Specific Datapath
Private L1/ Scratchpad

David Brooks @ Harvard

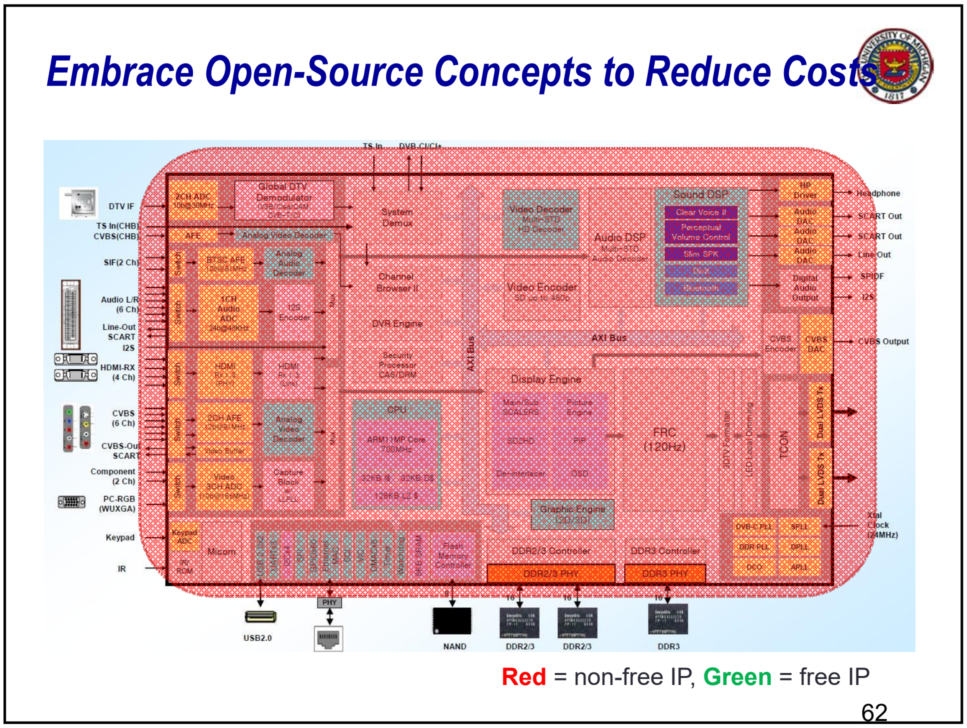
- Better tools and infrastructure
  - Scalable accelerator synthesis and compilation, ***generate code and H/W for highly reusable accelerators***
  - Composable design space exploration, ***enables efficient exploration of highly complex design spaces***
  - Well put-together benchmark suites to drive development efforts



CortexSuite: A Synthetic Brain Benchmark Suite

Michael Taylor @ UCSD

61

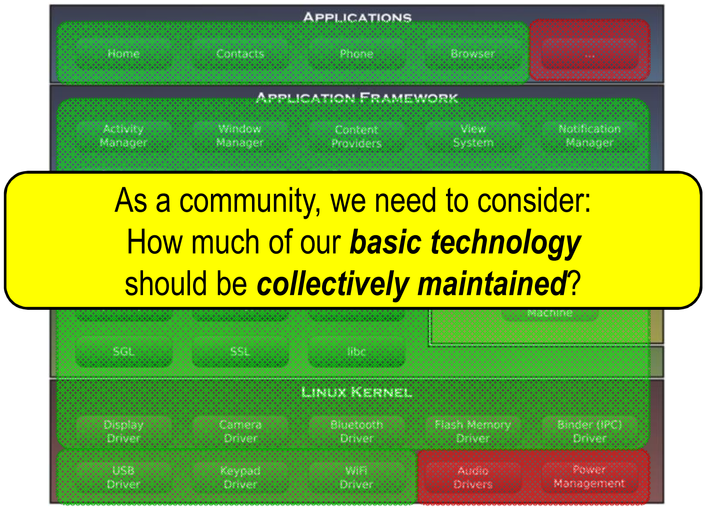


Embrace Open-Source Concepts to Reduce Costs

Red = non-free IP, **Green** = free IP

62

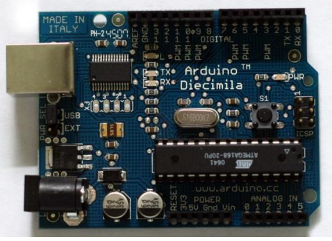
Embrace Open-Source Concepts to Reduce Costs



As a community, we need to consider:
How much of our ***basic technology***
should be ***collectively maintained***?

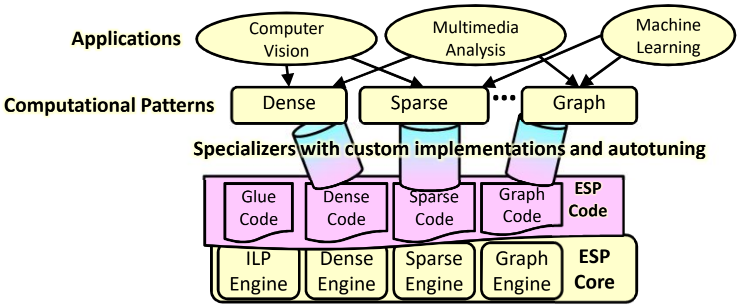
Red = non-free IP, **Green** = free IP

Open-Source H/W is Growing



4) Widen the Applicability of Customized H/W

Krste Asanovic @ UC-Berkeley

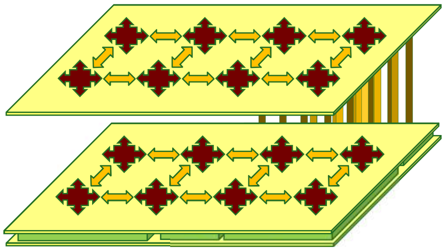


- ESP: Ensembles of Specialized Processors
  - Ensembles are algorithmic-specific processors optimized for code "patterns"
  - Approach uses ***composable customization*** to deliver speed and efficiency that is widely applicable to general purpose programs
  - Grand challenges remain: ***what are the components*** and ***how are they connected***?

5) Reduce the cost of manufacturing customized H/W

Martha Kim @ Columbia

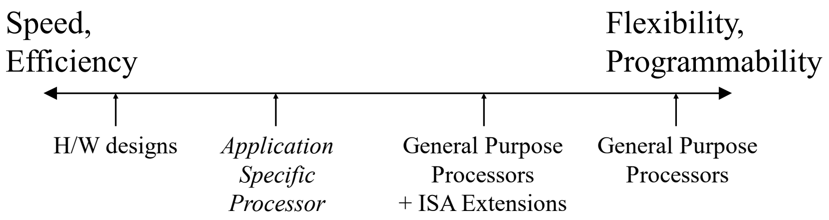
- Another thought experiment: what if building a house were like fabricating a chip?
- ***Brick-and-mortar silicon: assembly-time customization***, i.e., MCMs + 3D + FPGA interconnect



Brick-and-mortar silicon design flow:
1) Assemble brick layer
2) Connect with mortar layer
3) Package assembly
4) Deploy software

- Diversity via brick ecosystem & interconnect flexibility
- Brick design costs amortized across all designs
- Robust interconnect and custom bricks rival ASIC speeds
- Facilitates non-silicon integration and mature design strategies

Summary: Benefits of App-Specific Design



Speed,
Efficiency

Flexibility,
Programmability

H/W designs

Application Specific Processor

General Purpose Processors + ISA Extensions

General Purpose Processors

- Specialization limits the scope of a device's operation
  - Produces stronger properties and invariants
  - Results in higher return optimizations
  - Programmability preserves the flexibility regarded by GPP's
- A natural fit for embedded designs
  - Where application domains are more likely restrictive
  - Where cost and power are 1st order concerns
- Overcomes growing silicon/architecture bottlenecks
  - Concentrated computation overcomes dark silicon dilemma
  - Customized acceleration speeds up Amdahl's serial codes