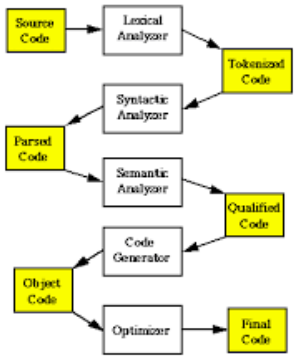
Computer Engineering Lab Faculty Research Pitches

Computer Science and Engineering
University of Michigan – Ann Arbor

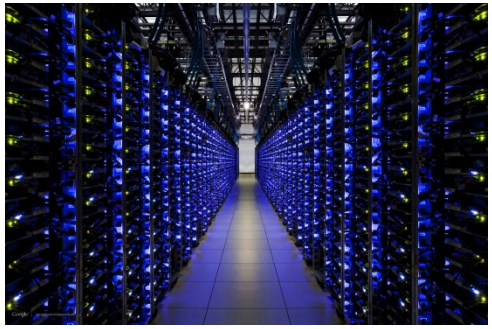
March 10, 2017

Broad Research Spectrum



Compilers

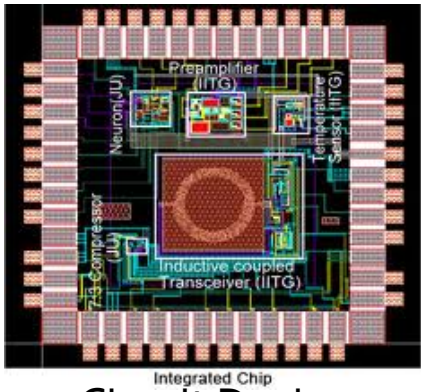
Design Viability



Data-centers

Hardware security

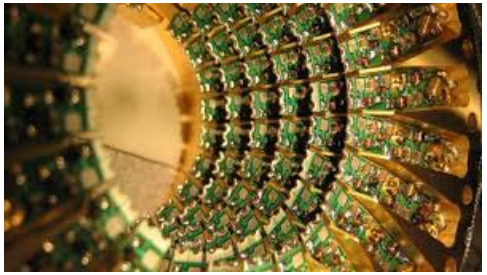
HW Accelerators
& GP-GPUs



Circuit Design

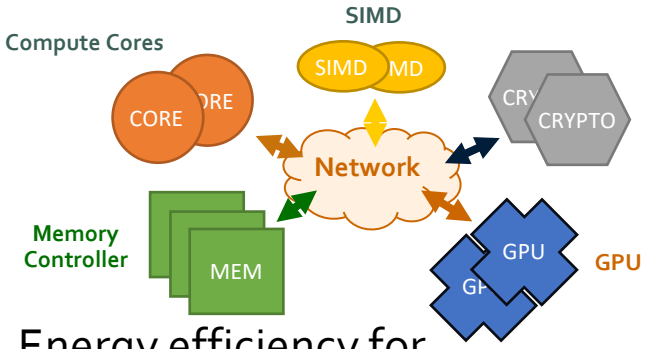
Reliable Design

Embedded & wearable systems

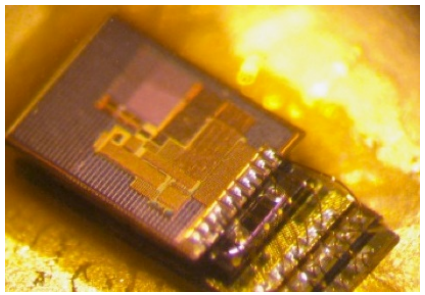


Quantum Computing

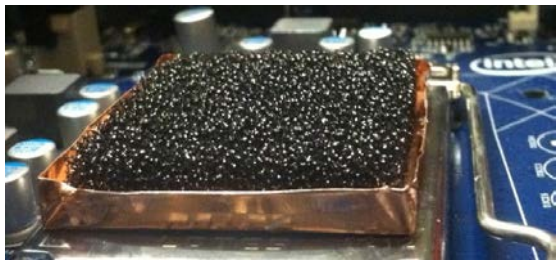
Some of Our Recent Ideas



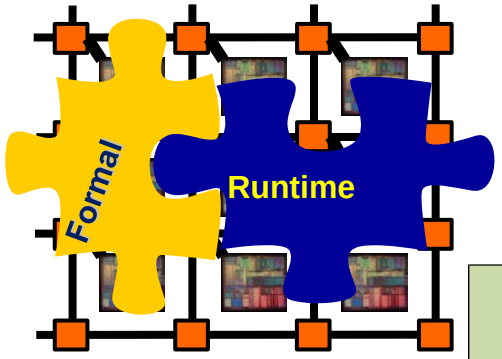
Energy efficiency for heterogeneous systems



Micromote for Smartdust

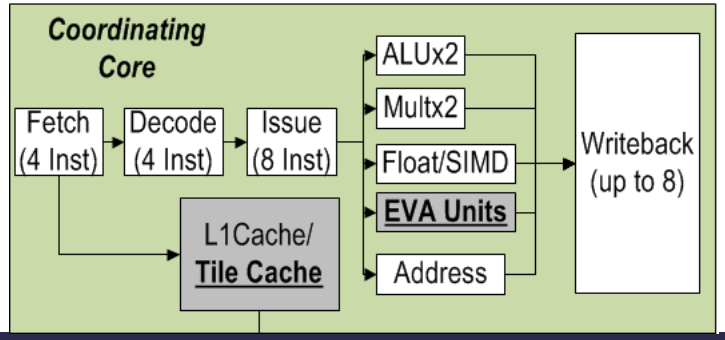


Computational sprinting

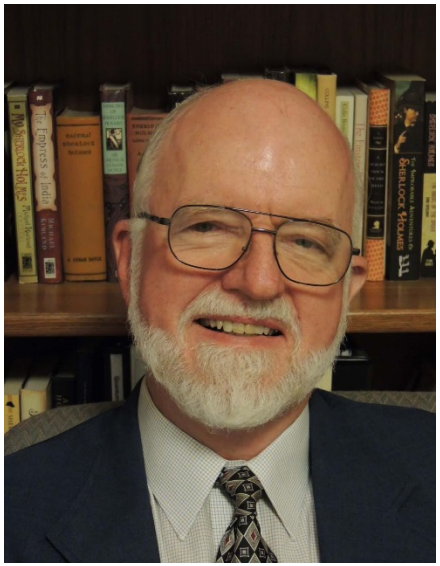


ForEVeR runtime

EVA – heterogeneous design



Embedded system security



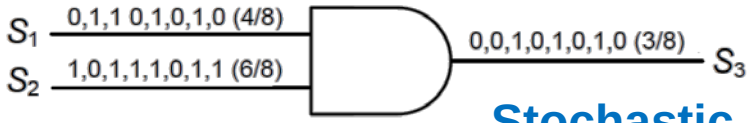
PROF. JOHN HAYES

John P. Hayes

- **Research Interests**
- CAD: Design and testing of VLSI circuits. Reliable computer architectures.
- Unconventional computing techniques, such as
  - Quantum computing
  - Stochastic computing = **Today's topic**
  - Deep learning networks

Vision chip for retinal implant

- **Stochastic computing** = Computation using random bit-streams



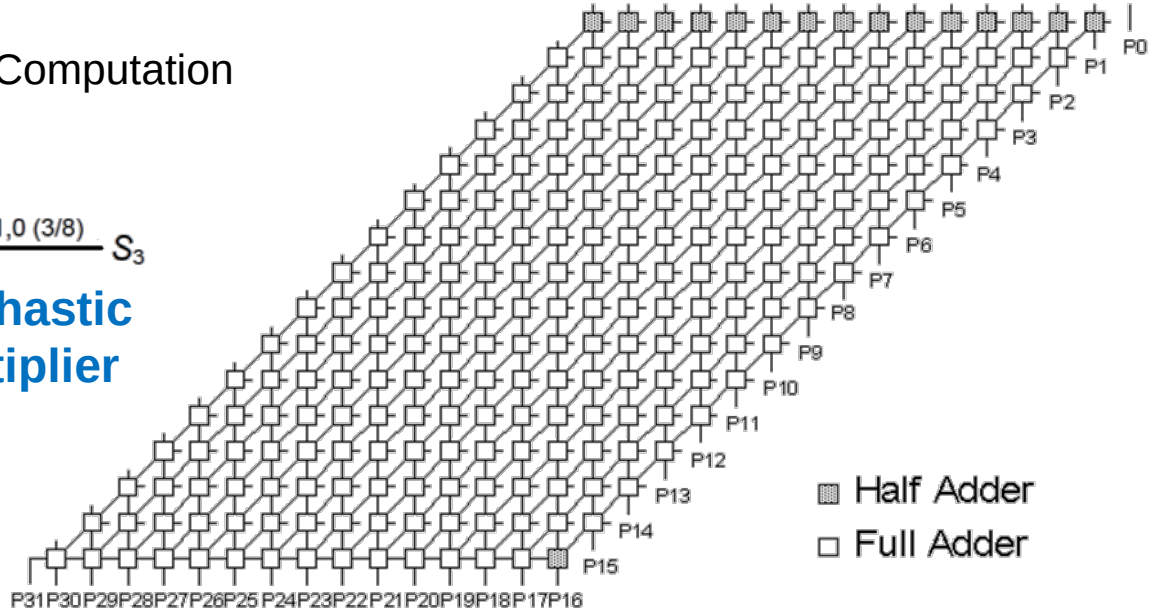
Stochastic multiplier

Advantages

- Tiny circuits
- Error tolerance

Applications

- Image processing
- Decoding complex ECCs
- Building artificial neural nets



Conventional multiplier

What is Stochastic Computing (SC)?

- *Def. 1*: A form of computing akin to neural computing

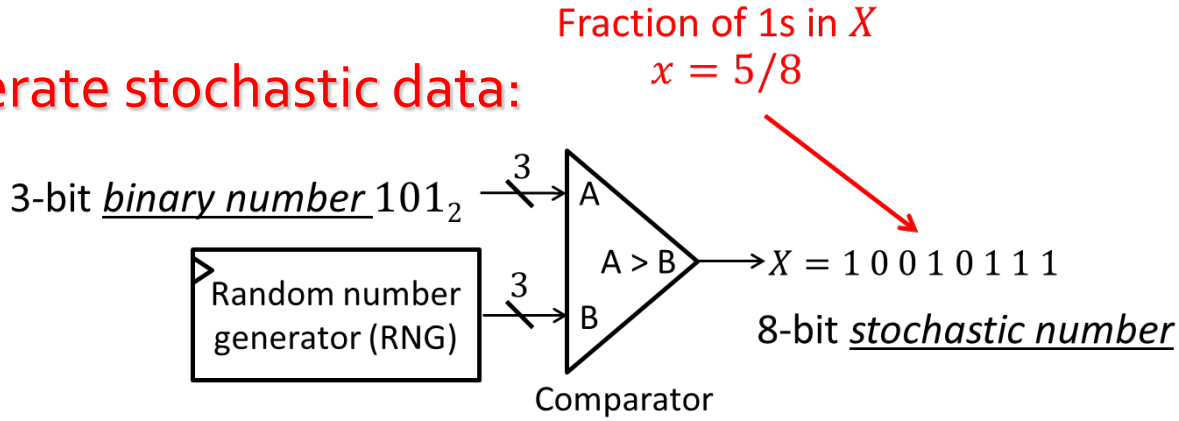
Biological neural signals

Stochastic computing signals

01000101001010101110000000010000000

- *Def. 2*: Computing with random bit-steams denoting probabilities

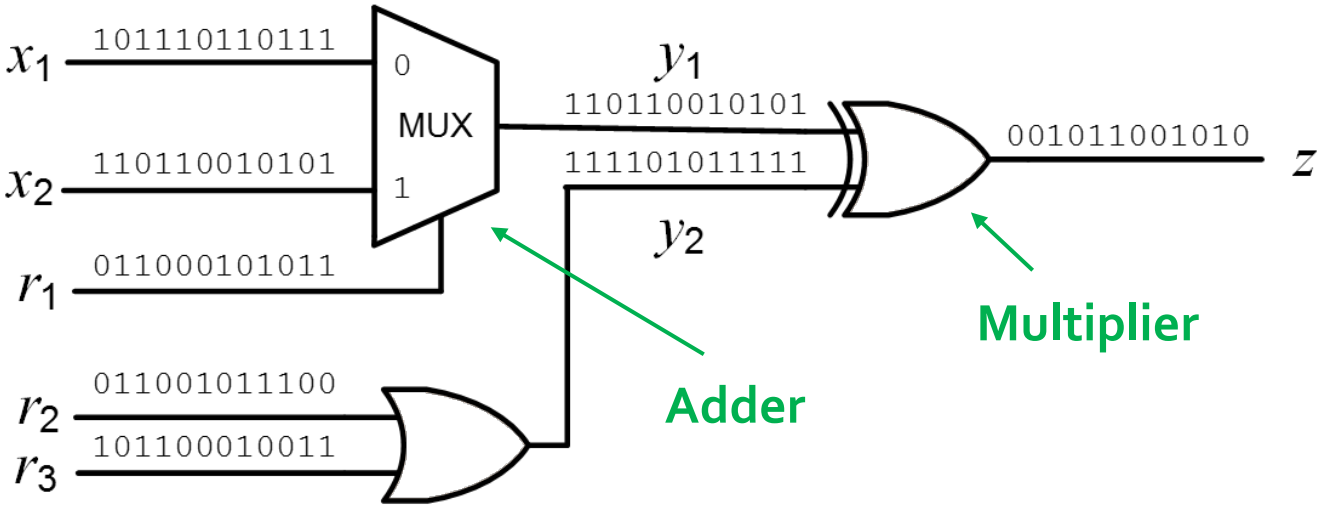
How to generate stochastic data:



Stochastic Numbers (SNs)

- An N-bit bit-stream X with N_1 1's is a ***stochastic number*** (SN) of value $p_X = N_1/N$. Usually p_X is the *probability* of a 1 appearing in X. This is ***unipolar*** representation.
- For example, 1000 and 0100010011000000 both have $p_X = 0.25$
- SN's lie in the unit interval [0,1], but arbitrary numbers can be approximated and scaled to lie in [0,1].
- If X's value is interpreted as $2p_X - 1$, then X denotes SN's over the interval [–1,1]; this is ***bipolar*** SC for signed arithmetic.

Example: Bipolar Circuit



Random numbers.
$p_{ri} = 0.5$

- Circuit's ***logic function***

$$z(x_1, x_2, r_1, r_2, r_3) = (x_1 \wedge \bar{r}_1 \;\; x_2 \wedge r_1) \oplus (r_2 r_3)$$

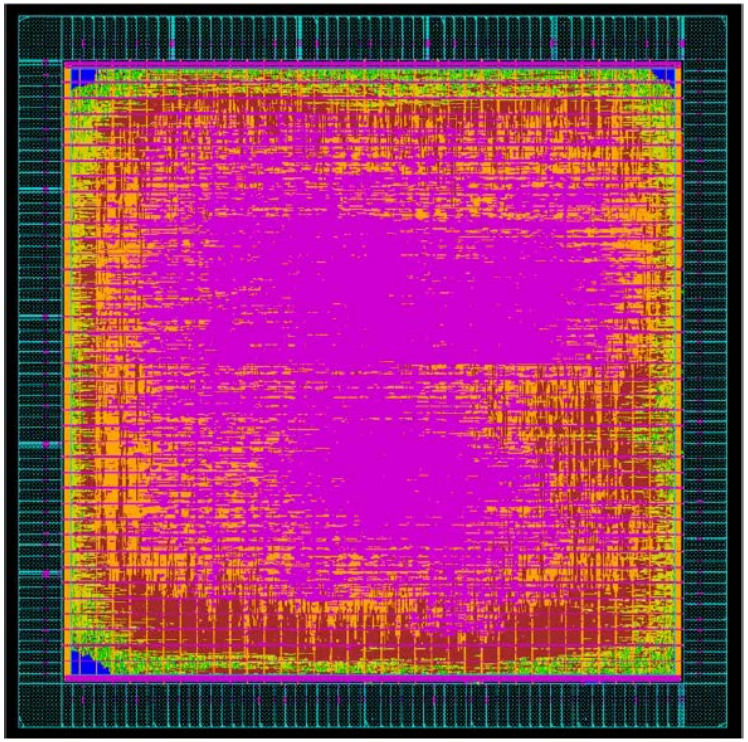
- Circuit's ***stochastic (arithmetic) function***

$$Z(X_1, X_2) = -0.25(X_1 + X_2)$$

Pros and Cons of SC

- ***Advantages***
  - Complex arithmetic operations can be realized by tiny logic circuits, suggesting low-cost massive parallelism
  - High tolerance of bit-flips (soft errors)
  - Variable precision
- ***Disadvantages***
  - Long bit-streams needed
  - Correlation among bit-streams reduces accuracy
  - Stochastic-binary number conversion is costly
  - SC design methodology is poorly understood

Application 1: LDPC Decoding



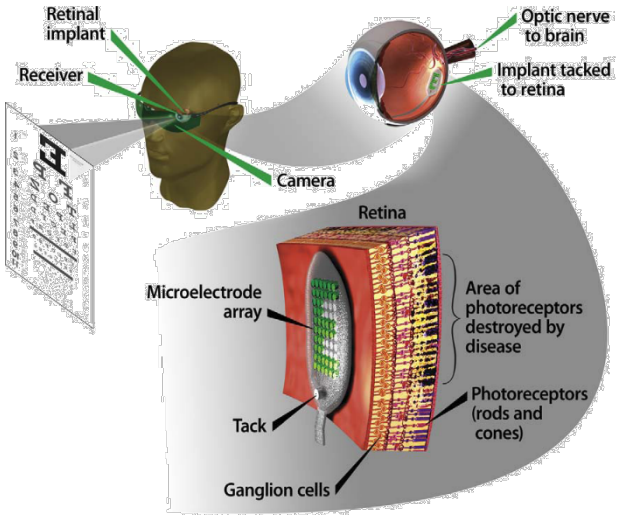
Layout of decoder chip for a (2048, 1723) LDPC code [Naderi et al. 2011]

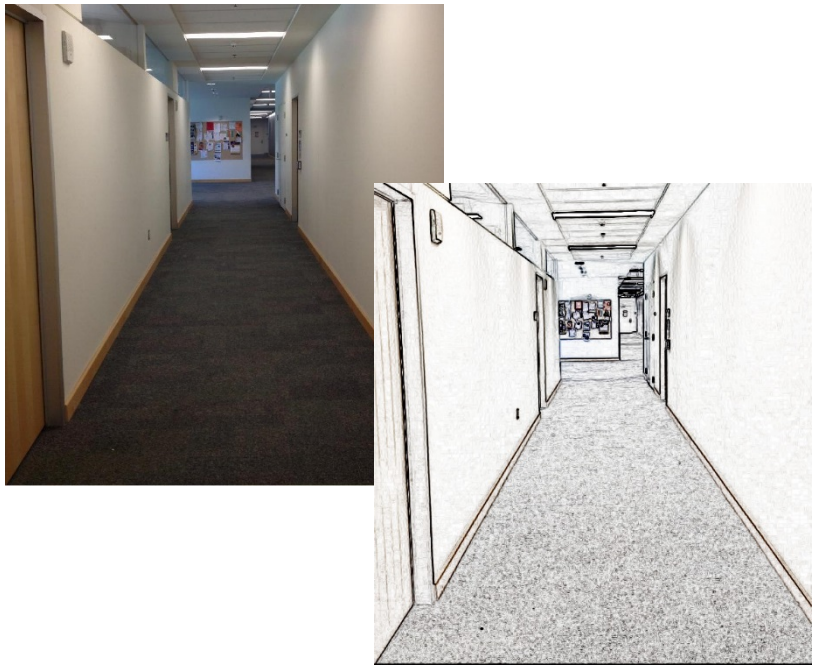
Size: 3.93 mm^2 using 90nm CMOS technology

Max. throughput: 172.4 Gb/s

Application 2: Biomedical Implants

- Retinal implant for the blind to provide edge detection

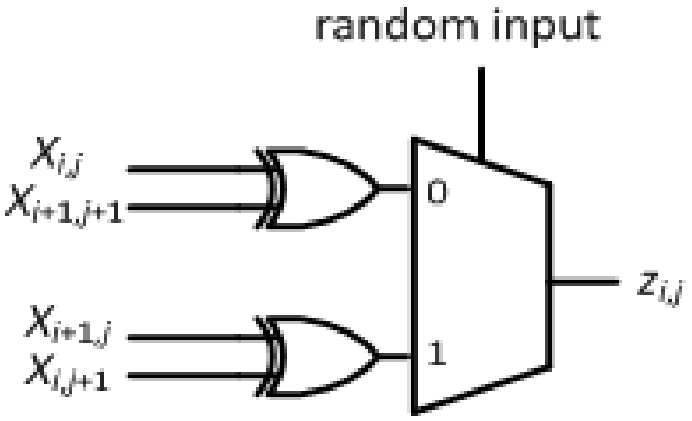




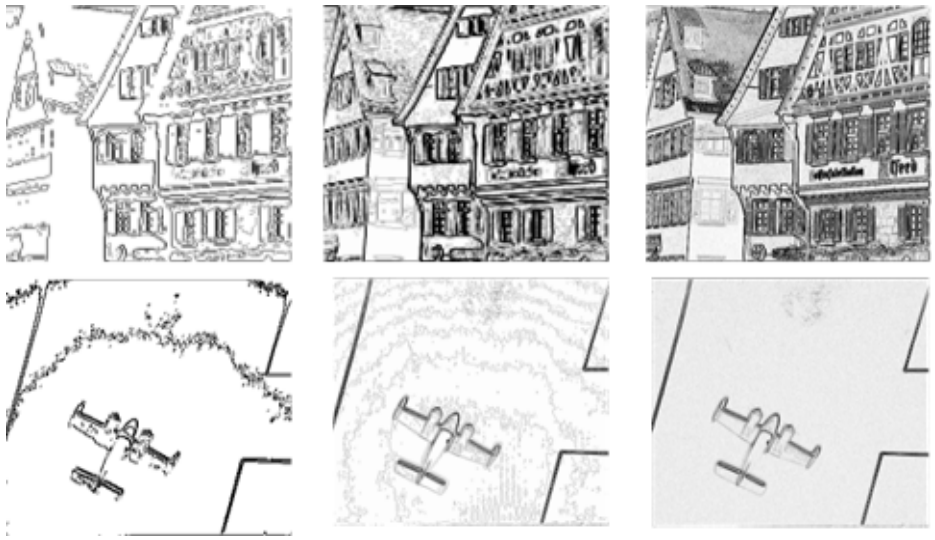
Edge Detection (contd)

Roberts cross formula for edge detection:

$$Z_{i,j} = 0.5 \times (|X_{i,j} - X_{i+1,j+1}| + |X_{i,j+1} - X_{i+1,j}|)$$



4 pixel inputs **SC edge detector** output



Edge detector output after: 4 32 256 Clock cycles

Some Research Challenges

- **Theory:** Better understand key costs and tradeoffs:
  - Accuracy / precision / bit-stream length
  - Power / energy
  - Error tolerance
- **Practice:** Exploit SC's potential applications
  - Develop efficient synthesis algorithms for sequential SC circuits
  - Reduce the high cost of binary-stochastic conversion
  - Exploit nanotechnologies with inherently stochastic behavior
  - Explore relations between SC and neural systems (both biological and artificial)
  - Explore relations between SC and other probabilistic computing techniques such as quantum computing

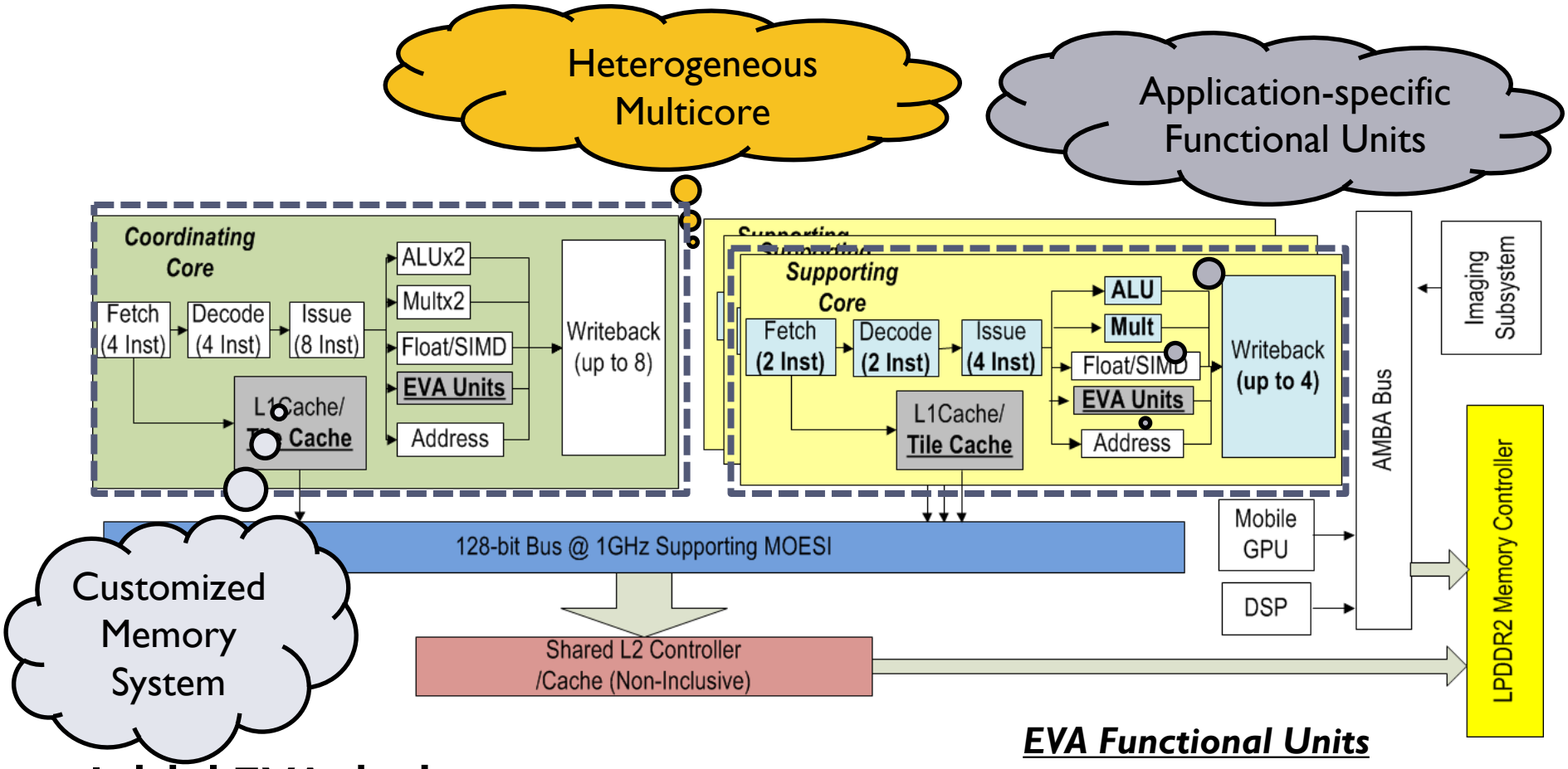
PROF. TODD AUSTIN

Introductions…

- Background
  - Computer Architect == H/W + S/W
  - PhD in CS from UW-Madison in '96
  - An architect at Intel until '99, then Michigan
- Teaching
  - Architecture, compilers, programming, security
  - Special focus on CSE development in Ethiopia
  - Co-author of an undergraduate computer architecture textbook with Andy Tanenbaum

- Research
  - Computing in a Post Moore's Law World
    - Watch a talk on the challenges: http://bit.ly/PostMoore
  - Meeting the Challenges of Hardware Security
    - Watch my H/W security tutorial: http://bit.ly/HWSecurity
  - Director of C-FAR : Center for Future Architectures Research
    - Focused on scaling in 2020-2030 silicon
    - Performance, power and cost
    - 27 faculty at 14 universities, 92 students

Architecture: Embedded Computer Vision



Initial EVA design:

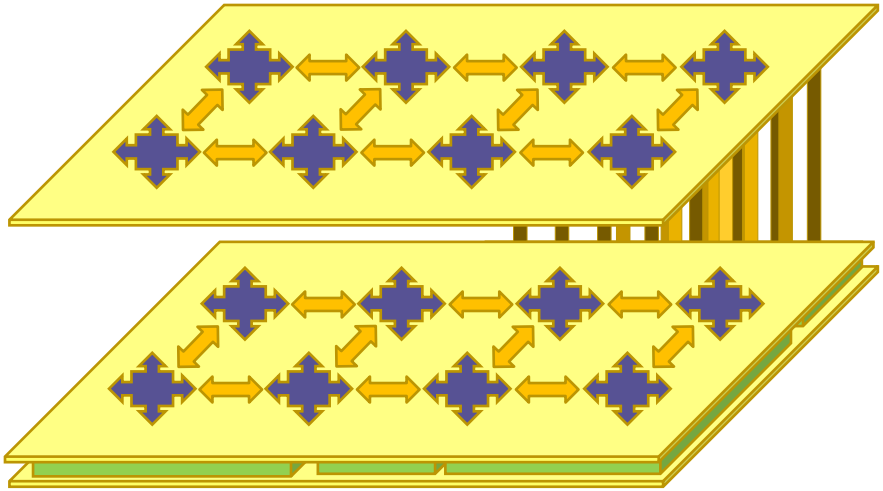
90x greater efficiency for computer vision algorithms

EVA Functional Units

Monopoly Compare,
Dot Product Unit,
Vector Max,
Decision Tree Compare

Post-Moore: Assembly-Time Customization

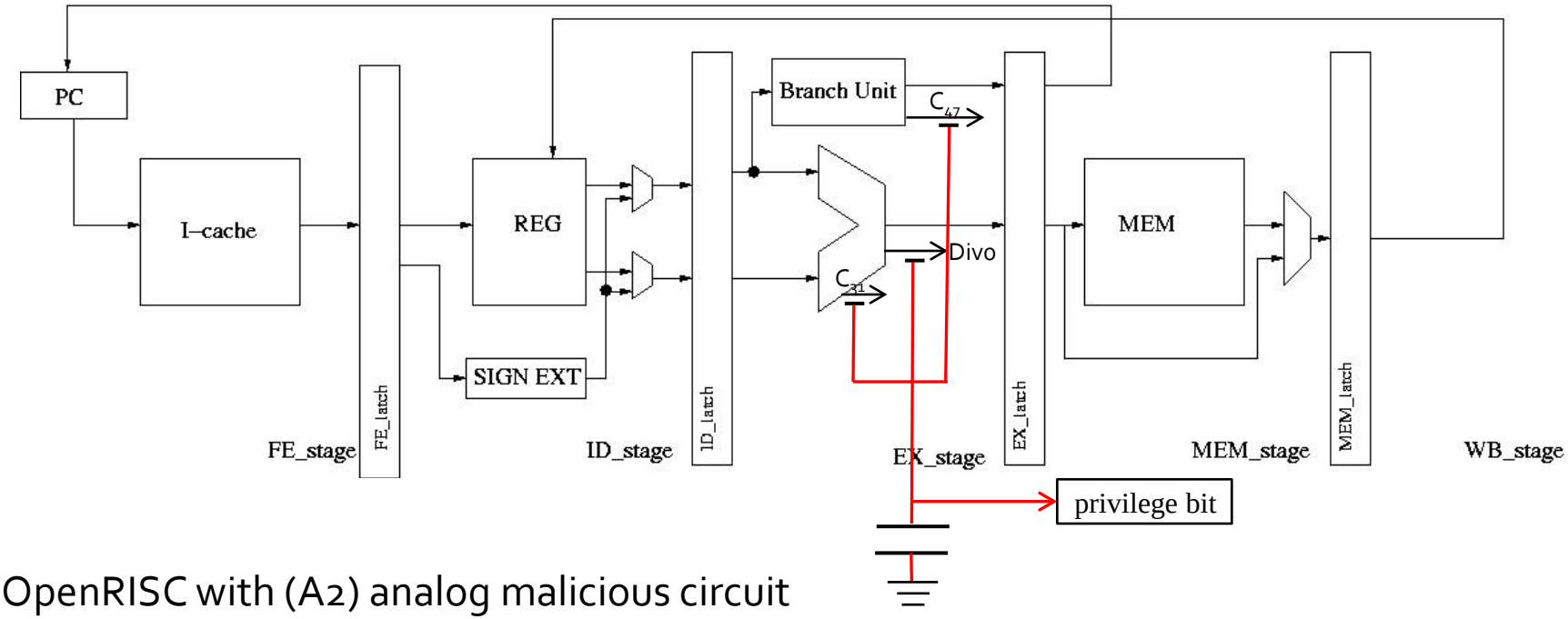
- ***Brick-and-mortar silicon*** explores ***assembly-time customization***, i.e., MCMs + 3D + FPGA interconnect



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

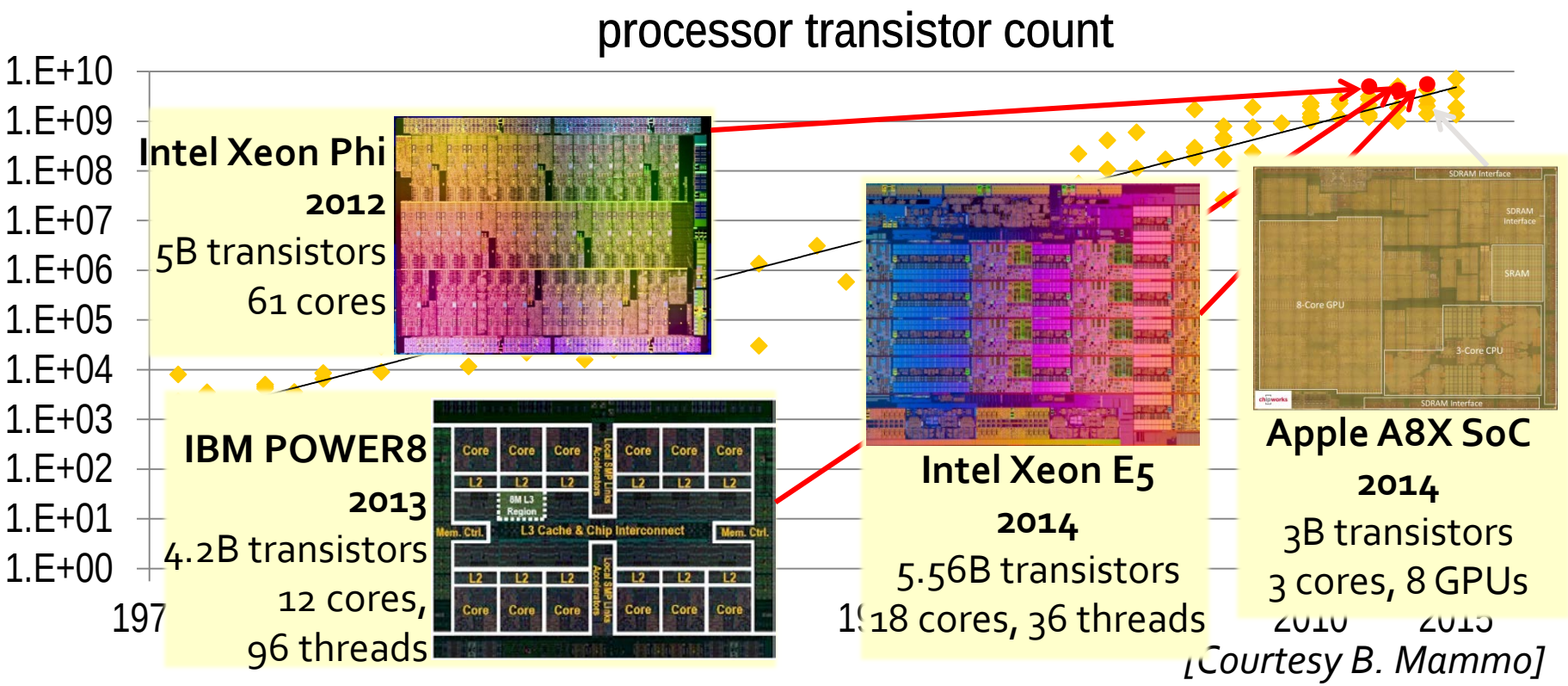
HW Security: A2 Analog Malicious Circuit



- OpenRISC with (A2) analog malicious circuit
  - Charge share with infrequent signals (e.g., Divo, C_{31}) to charge up leaky passive cap
  - If cap charges up fully, CPU privilege bit is set
- Attack: 1) frequently execute unlikely trigger code sequence, 2) own machine (as you now have privilege mode access)
- Taped out chip, attack sequence working in the lab, no false positives detected
  - Malicious circuit is not detectable by current protections (i.e., lacks power/timing signature and it has no digital representation)

PROF. VALERIA BERTACCO

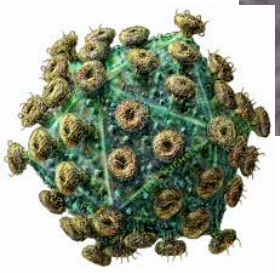
Computing requires silicon chips…



[Courtesy B. Mammo]

…big, fat, silicon chips!

Transistors keep getting smaller…



100nm - virus

A silicon atom is .25nm in diameter

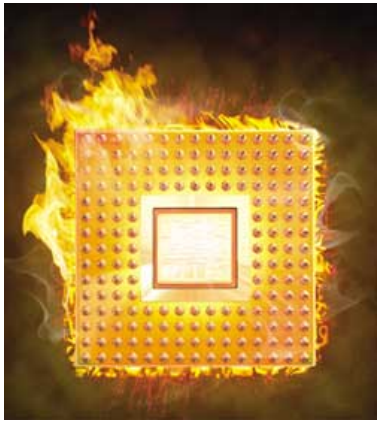
…and also fragile & highly volatile

The BLAB (Bertacco Lab) to the rescue!

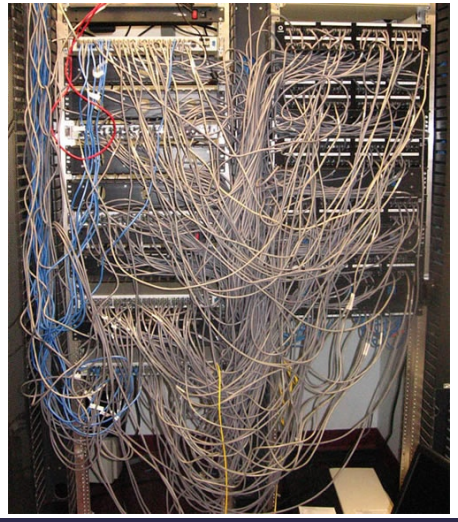
System viability, in the face of faults, errors, attacks and unbearable design complexity

Protect from:

- ***Transistor failures***



- ***Designer mistakes***

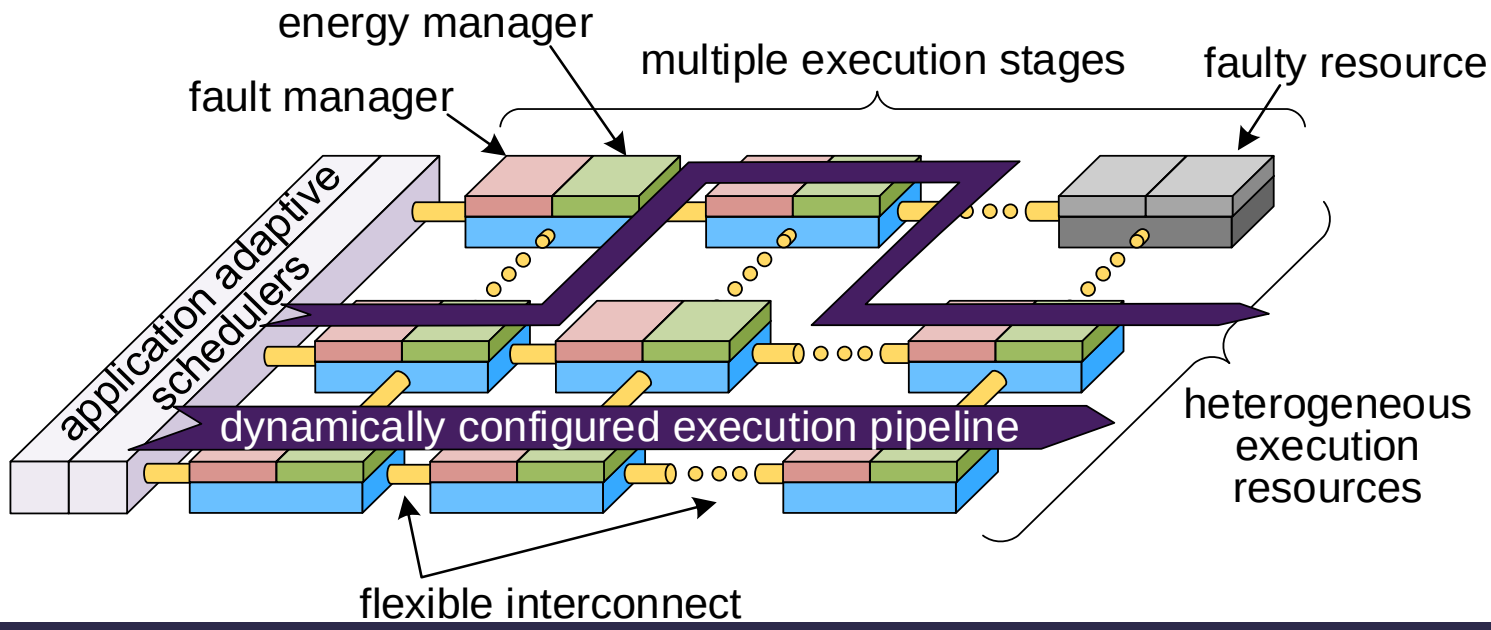


- ***Unbearable complexity***

ReDEEM: **Reliability** from the ground up

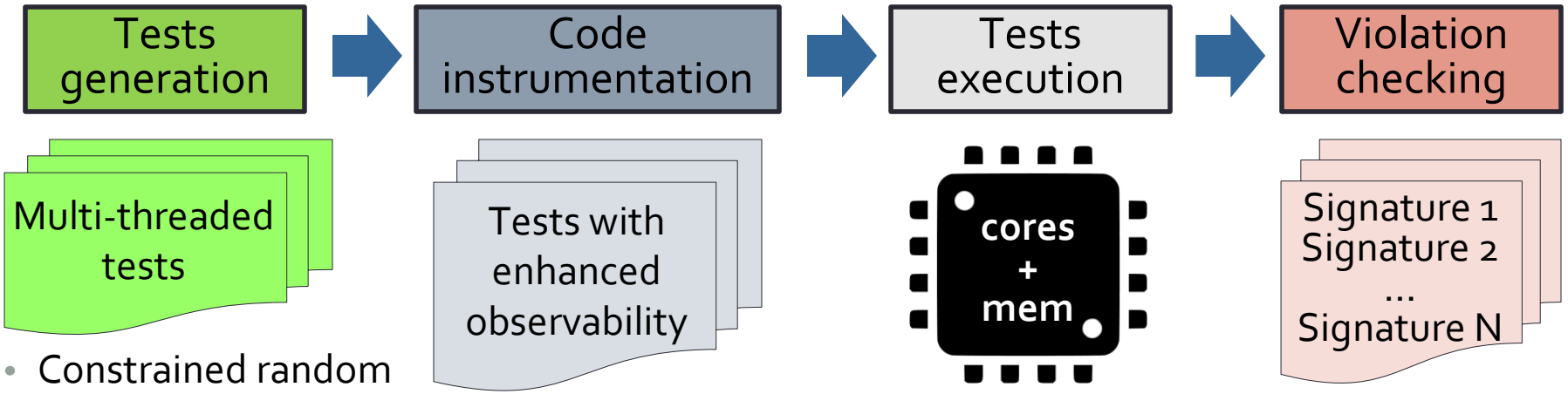
An energy efficient, reliable microarchitecture designed from the ground up

- **Distributed and decoupled execution resources** for high reliability
  - Execution pipelines constructed dynamically
- **Heterogeneous execution resources** for performance-power diversity
- **Application-adaptive schedulers** for creating energy-efficient pipelines



Bug, where art you?

MTraceCheck: Efficient post-silicon multi-core validation of non-deterministic behavior in memory consistency models



- Constrained random
- Designed for memory consistency validation

- Static code analysis
- Compact signature

- Minimal test perturbation
- Signature collection

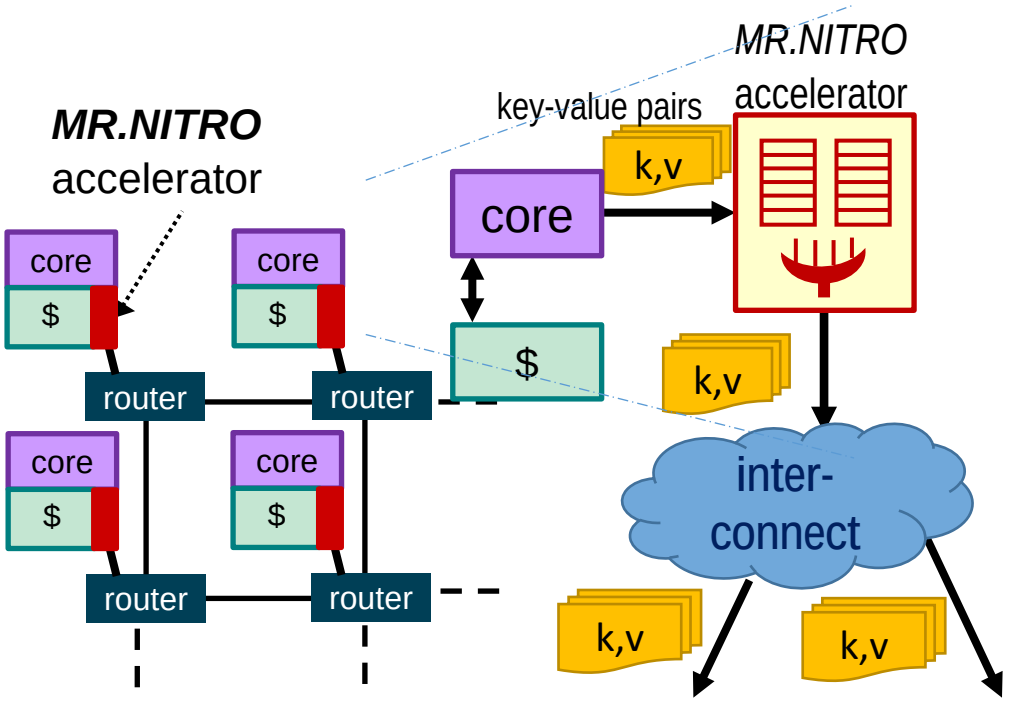
- Collective checking for signatures

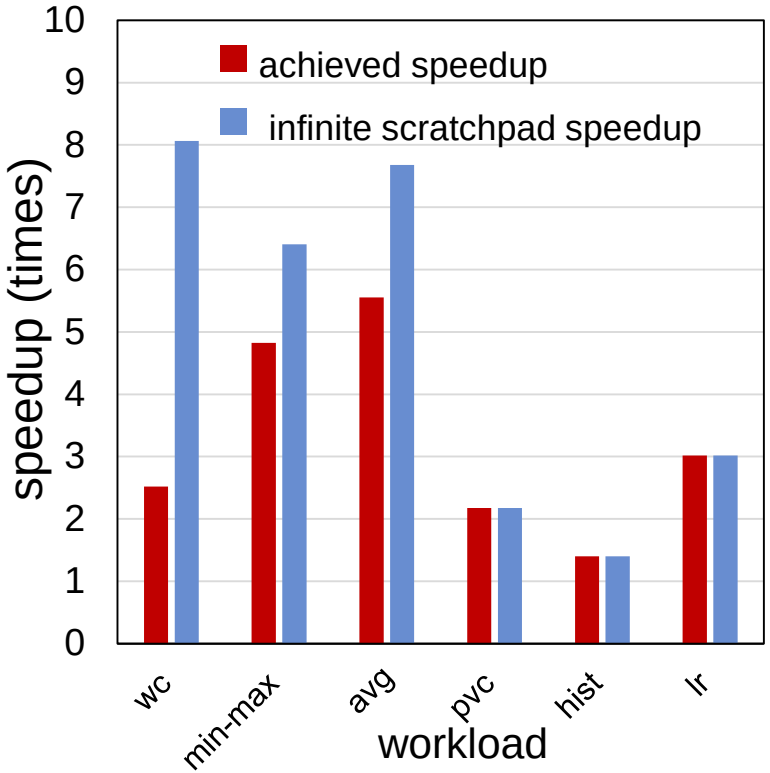
- Experimental results (bare-metal platforms & full-system simulation)
  - Minimal perturbation for observability: memory accesses reduced by 93%
  - Checking computation requirements reduced by 81%
  - To appear at ISCA'17

Divide and conquer big-data apps

MR.NITRO: accelerate map-reduce applications via distributed accelerator architecture

- Accelerators receives key-value pairs from both local core and interconnect
- They aggregate kv-pairs at source and destination





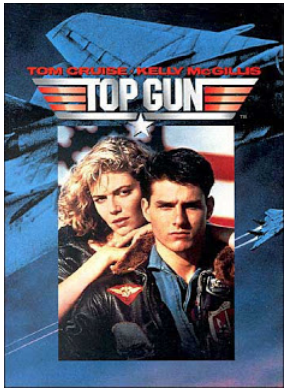
>3X speedup over CMP-based distributed MapReduce.

Area o/h <6%

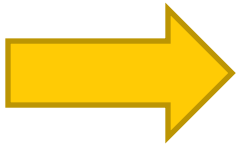
PROF. SCOTT MAHLKE

CCCP Group

- Compilers Creating Custom Processors
- "I feel the need,
  the need for speed"
  - Maverick and Goose

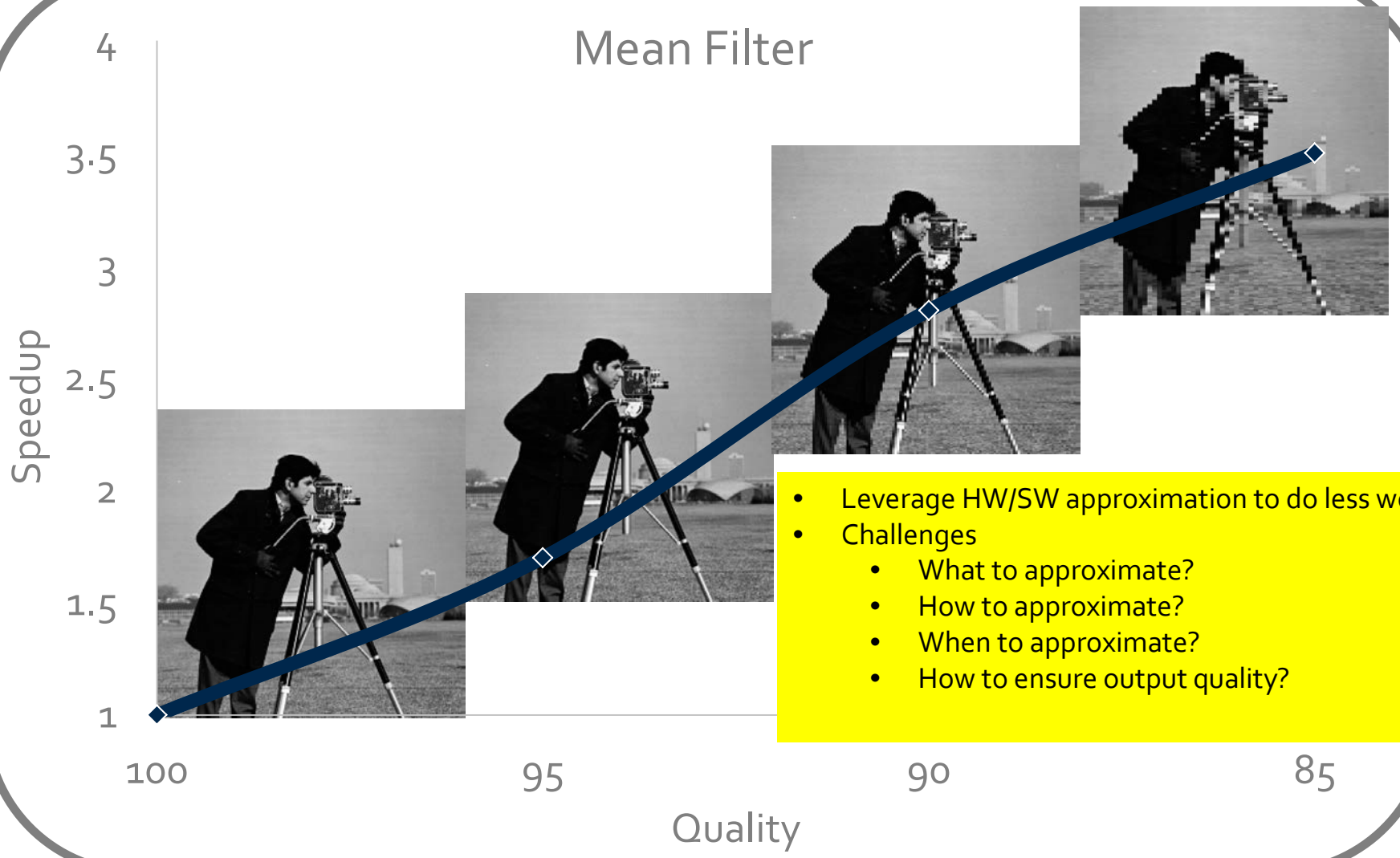


- Processors



- Customize design to the application
- Customize application to design
- Compute, memory, interconnect, …
- Increase efficiency
- Ensure programmability
- Processors vs accelerators
- Fine-grain heterogeneity

Approximate Computing



- Leverage HW/SW approximation to do less work
- Challenges
  - What to approximate?
  - How to approximate?
  - When to approximate?
  - How to ensure output quality?

Neural Network Compilation

Target Hardware

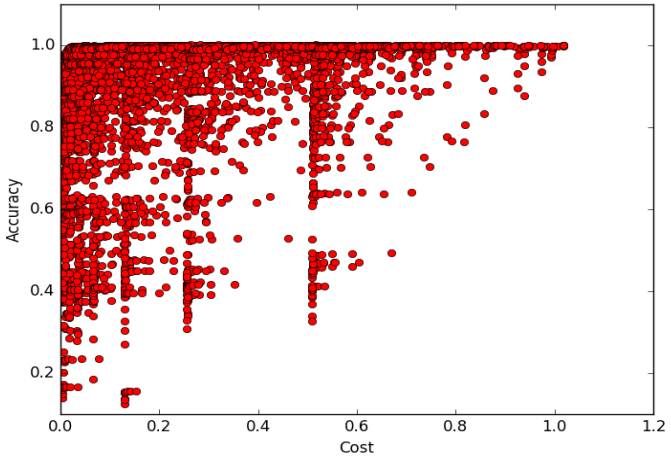
Compiler

DNN

Optimize network
- Eliminate redundancy – Prune the heck out of it
- Bitwidth analysis
- Float/fixed pt

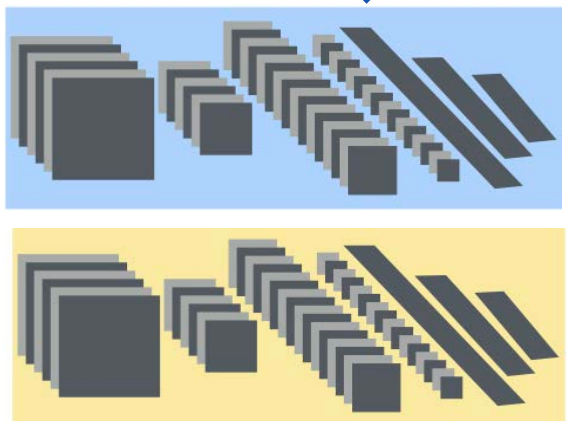
Customize for target hardware
- Create ensemble of NNs
- Splitter
- Data heterogeneous NNs

MNIST



Input

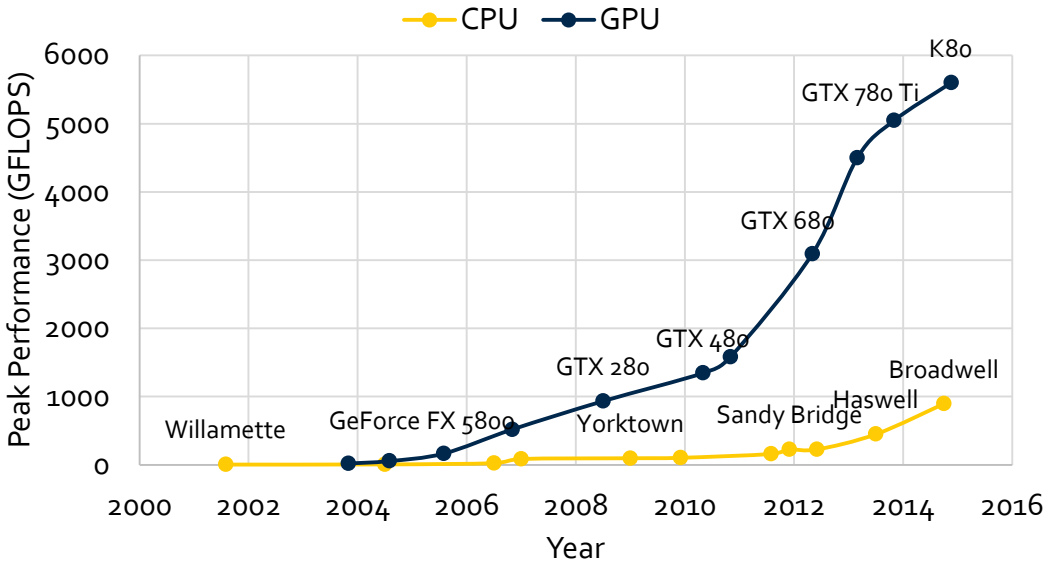
Network Selector

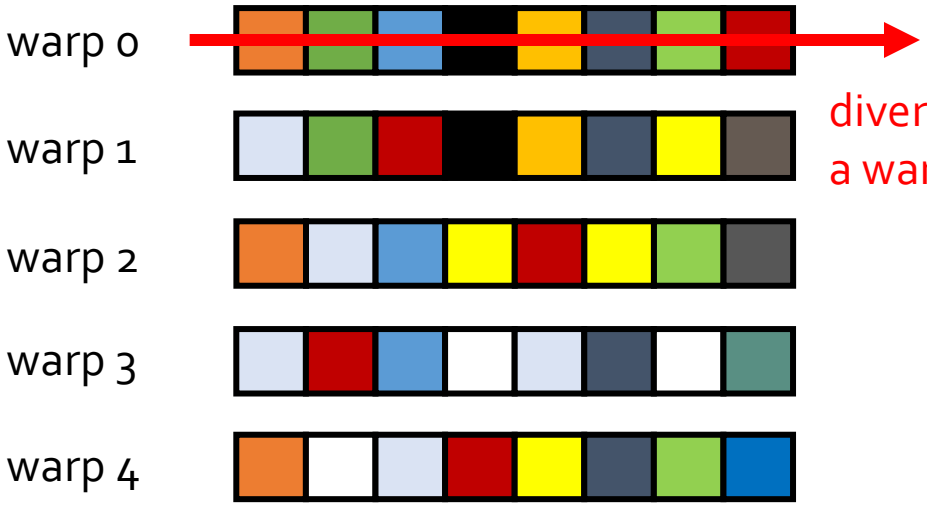


Input Partitioning

Customized CNNs

Energy-first Throughput Processors





divergent inside a warp

Rethink GPU design from the ground up for non-graphics workloads

- Memory system
  - Reducing latency
  - Caching
- Data path
  - Customizable compute
  - Register file design
- Shared GPUs
  - Pre-emption
  - Multi-kernel support

QUESTIONS?