Blurring the Layers of Abstractions: Time to take a step back?

Krisztián Flautner

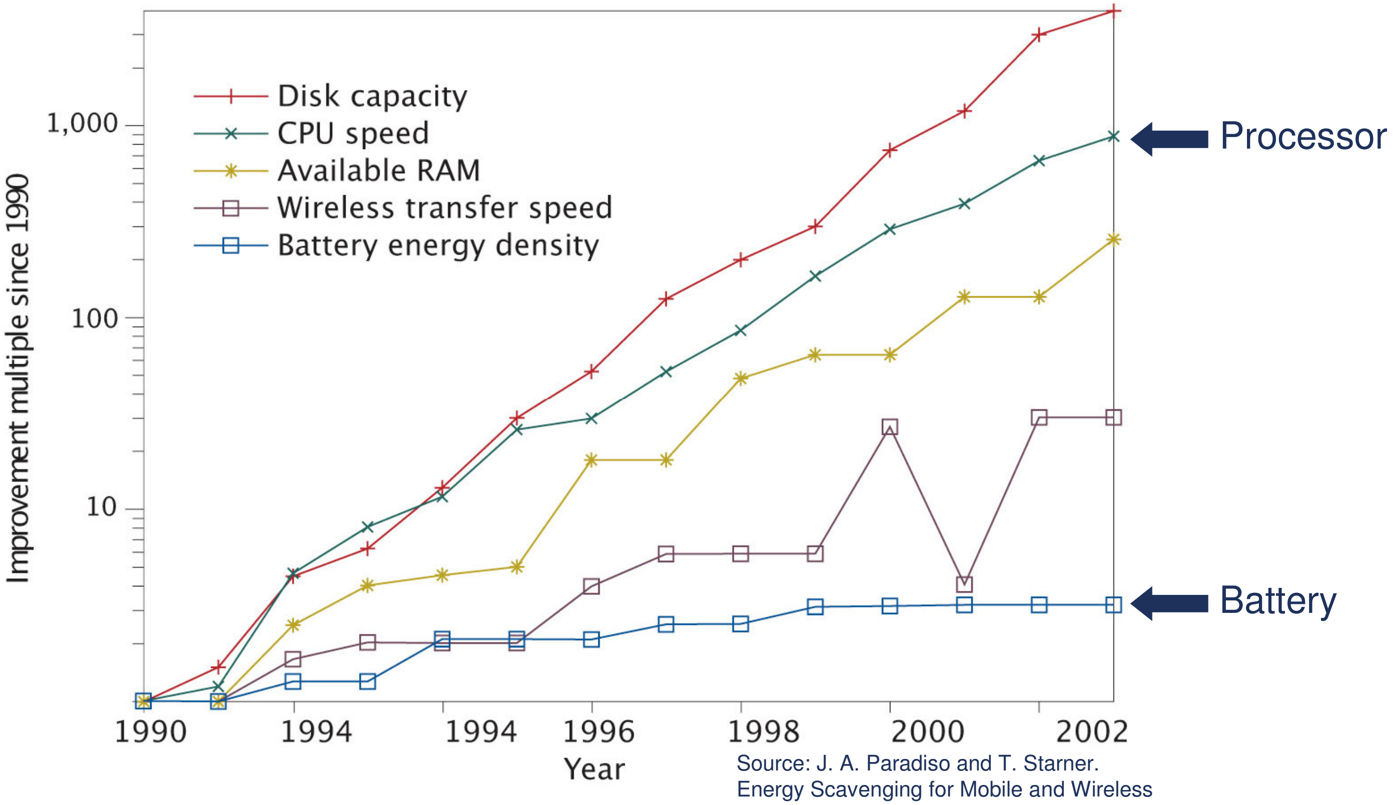
Director of Research

ARM Limited

krisztian.flautner@arm.com

April 12, 2007

Historical trends set high expectations



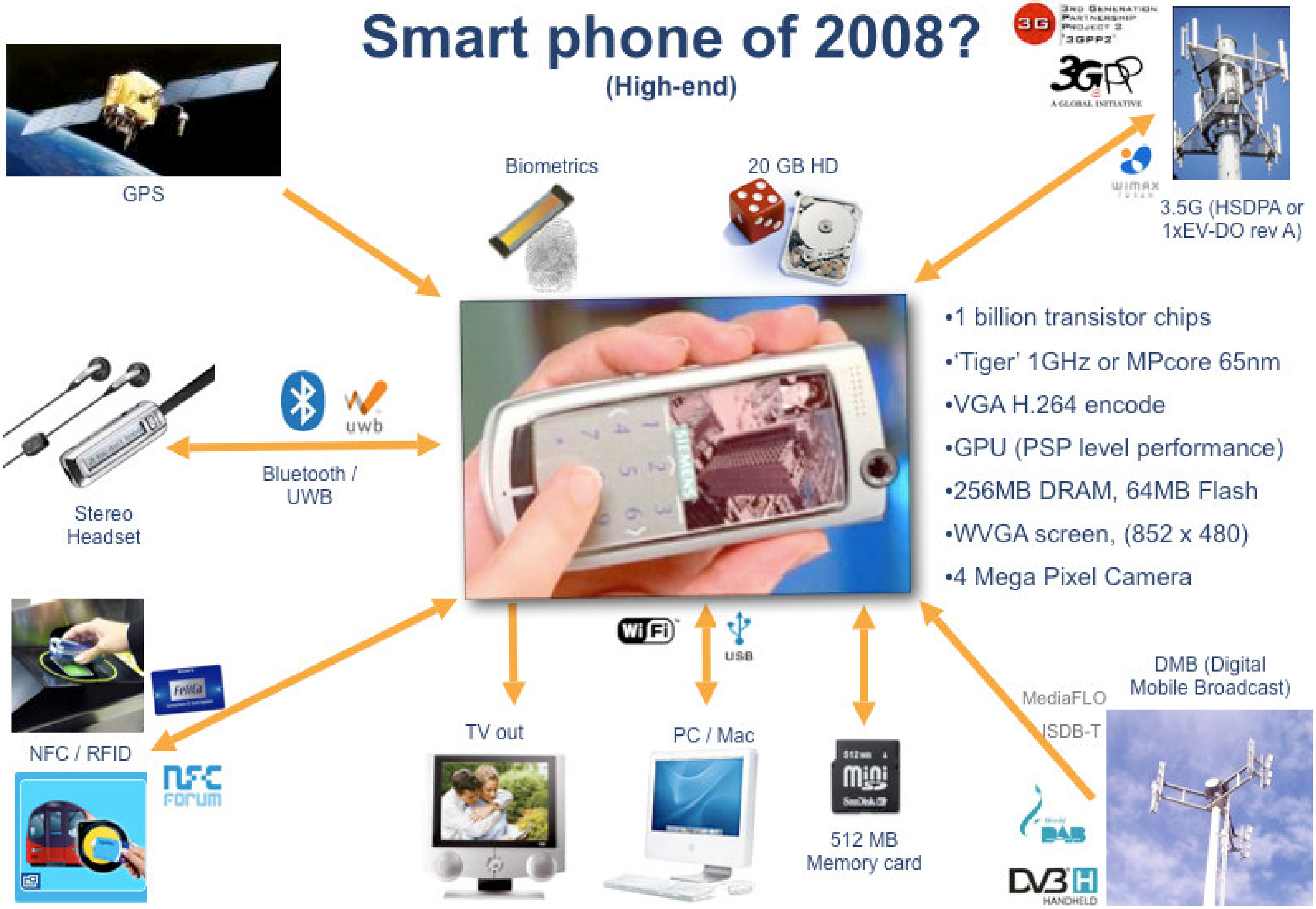
Source: J. A. Paradiso and T. Starner. Energy Scavenging for Mobile and Wireless Electronics IEEE Pervasive Computing, 2005.

- Most things have been improving at an exponential rate (except batteries)
- Past trends induce an expectation of perpetual growth
- How does one deliver the expected system performance in a fixed energy budget?!

… encourage bold hw requirements

Smart phone of 2008?

(High-end)



GPS

Biometrics

20 GB HD

3.5G (HSDPA or 1xEV-DO rev A)

Stereo Headset

Bluetooth / UWB

NFC / RFID

TV out

PC / Mac

512 MB Memory card

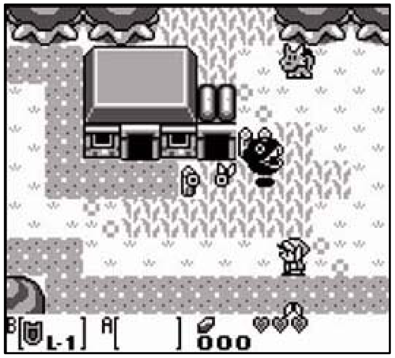
MediaFLO

ISDB-T

DMB (Digital Mobile Broadcast)

- •1 billion transistor chips
- •'Tiger' 1GHz or MPcore 65nm
- •VGA H.264 encode
- •GPU (PSP level performance)
- •256MB DRAM, 64MB Flash
- •WVGA screen, (852 x 480)
- •4 Mega Pixel Camera

15 Years of Change - our *entitlement*



Nintendo GameBoy (1989)

CPU: 8-bit Z-80 processor, 1.05 MHz
Screen: 2.6" 160 x 140 LCD
Connectivity: 4 players by serial cable

Price: introduction price - $169

>1000x performance
for the same price

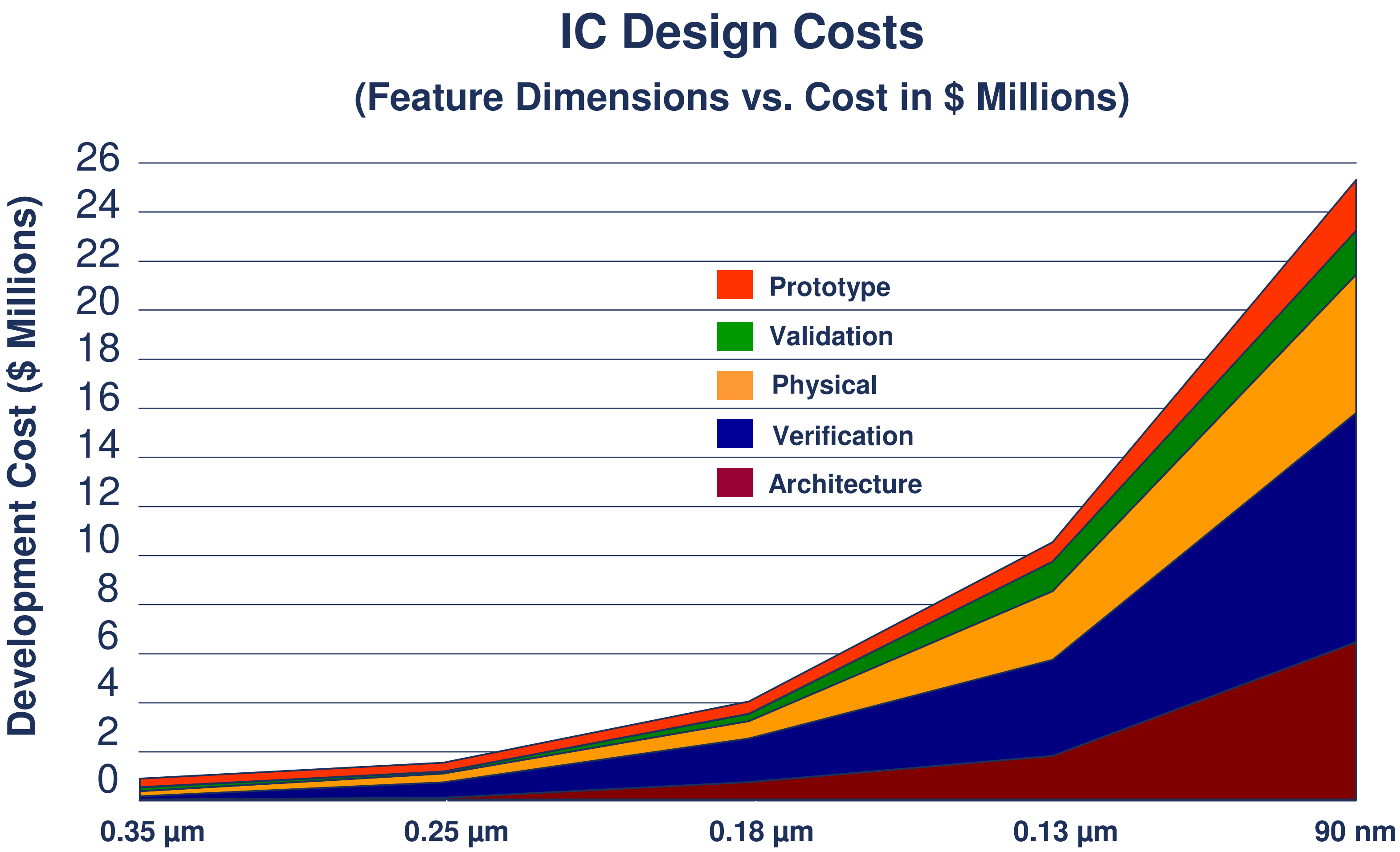
© Nintendo

Nintendo DS (2004)

CPU: ARM9™ (66 MHz) and ARM7™ (33MHz)
Screen: Two 3" 256 X 192 colour LCDs
Connectivity: 16 players wirelessly
and embedded WLAN

Price: exp. introduction price - $150

The cost of fulfilling our expectations



Source: International Business Strategies

Problem definition

Market expects exponential improvements

- in silicon integration densities and per-transistor costs

Increasingly difficult to deliver

- Complexity implies high design and manufacturing costs
- Pressure on cutting time-to-market
- High risk: breakeven at increasingly higher volumes

Much of the problem lies with physics

- And the way we think about getting “around” it
- Trends are on a vector in the wrong direction
- Designing for the worst-case is becoming unsustainable
  - Design cost is on a fast ramp
  - Increased Si variation: typical is much better than worst

The shape of the solution

Spend transistors on easing hard design problems

Break some abstractions

- Computation may not always be correct
- Miss timing some of the time, compensate at run-time

Speculate on correctness

- Assume that circuits work as expected, recover if not
- Speculation is key to minimizing run-time overhead

Need fault tolerance features in mainstream systems

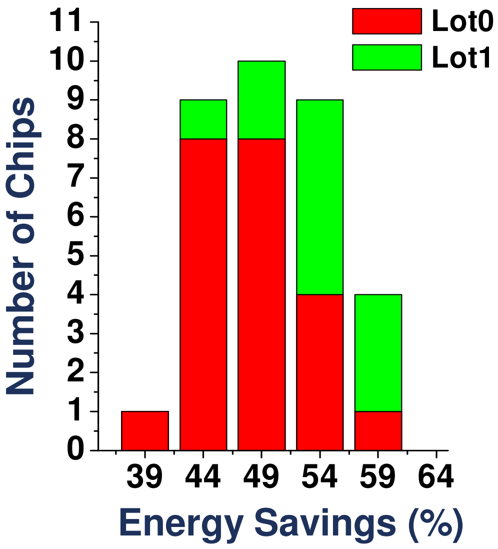
- ... at commodity prices
- ... full redundancy is not an option

Razor: a microarchitecture for the nanoscale era

- **The wall ahead of Moore’s Law is made of rubber**
  - How far it can be stretched is an open (and expensive) question
  - Good control of emerging Si process technologies is increasingly difficult
  - Engineering complexity (cost) is on a steep incline
- **Unfortunately we cannot change the laws of physics**
  - But can make it easier for engineers to deal with them

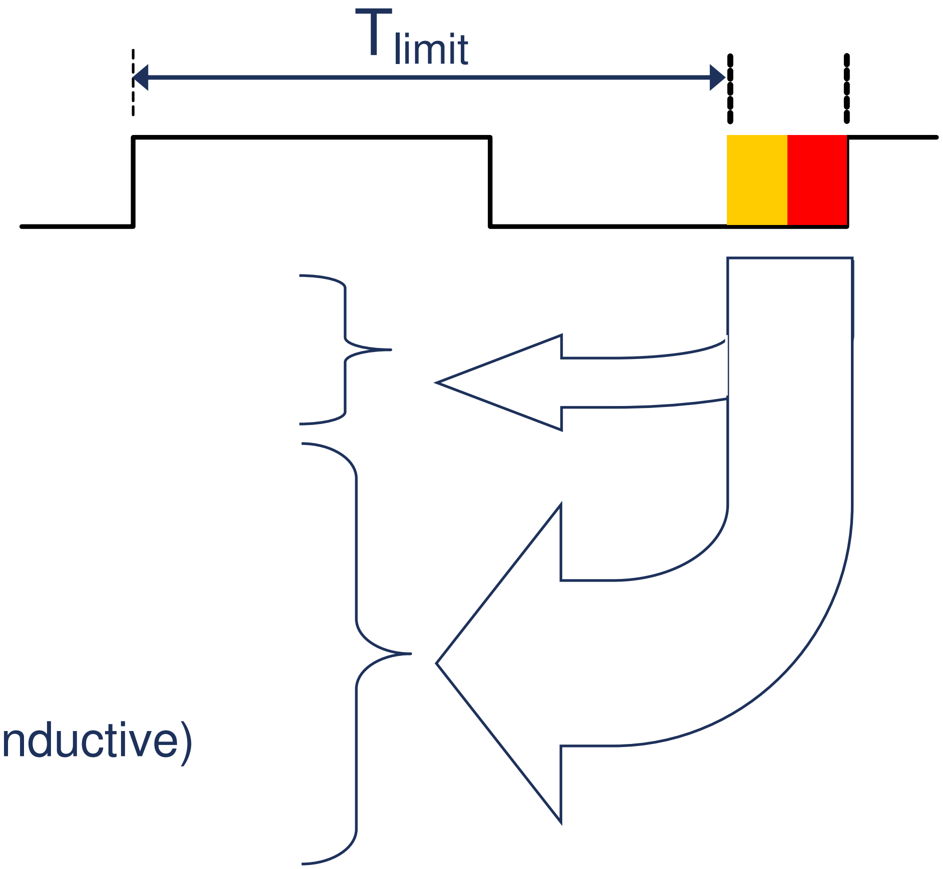
- **Today: most chips designed for & operated at worst-case parameters**
  - Achievable but means that all chips run either too slow or using too much power almost all the time
  - Example on right: all chips run using 40%-60% less energy when adapted to the specifics of their dies
- **Tomorrow: rarely occurring corner cases will severely limit advantages from process scaling**
  - Need to optimize for the typical case while correcting for worst case conditions

- **Razor puts the lid on implementation complexity**



Key concepts: Design Margins

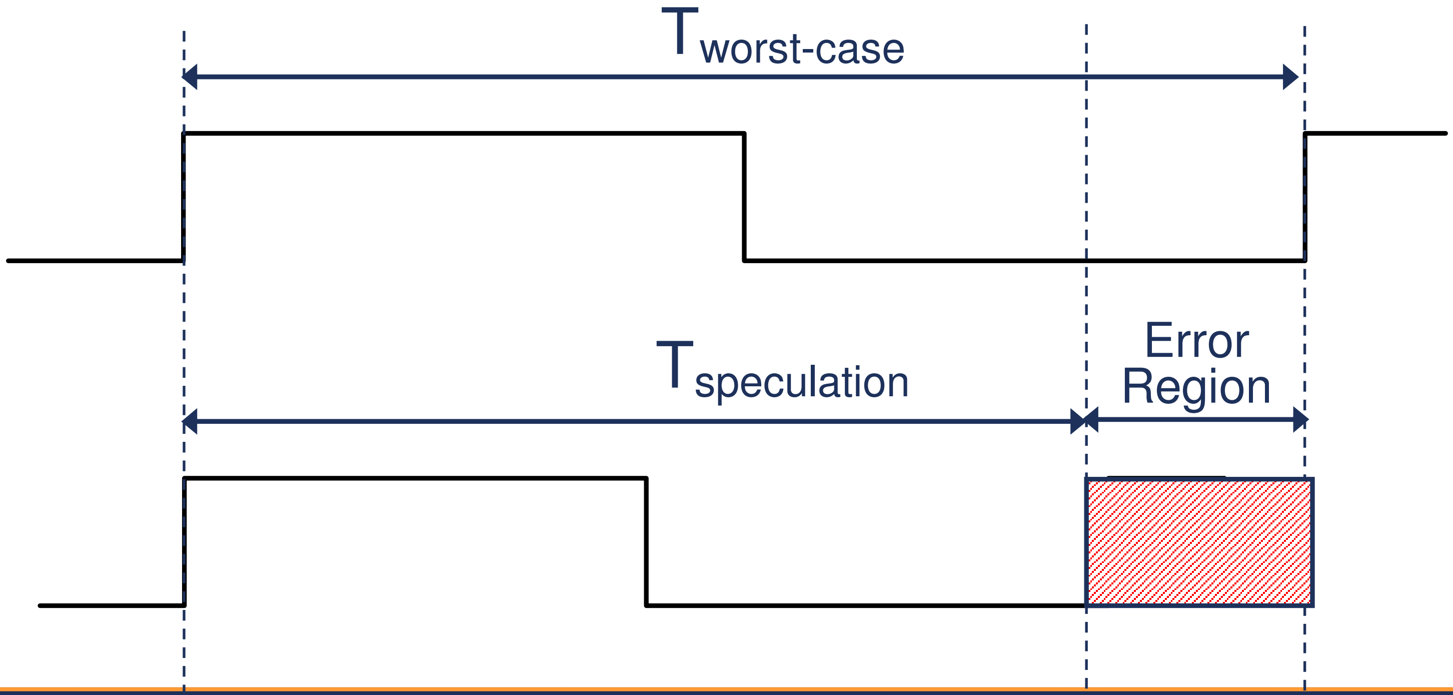
- Traditional design margining
  - Design for the worst-case conditions
    - Static
      - Inter-die (SS, FF, FS, SF)
      - Intra-die variations
    - Dynamic
      - Power supply variations
      - IR drop
      - Temperature fluctuations
      - Coupling noise (capacitive, inductive)
      - Clock jitter



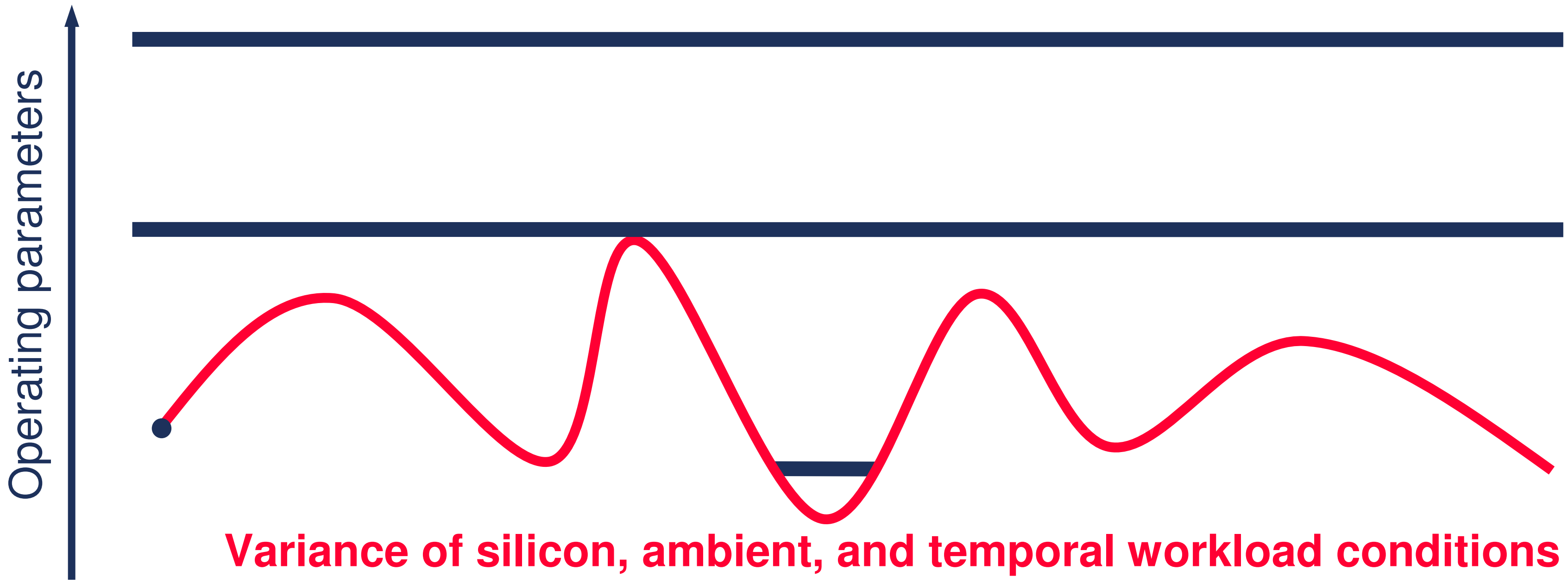
- ***BUT worst-case conditions do not happen all of the time***

Why assume worst-case all the time?

- Worst-case could happen
- Timing Speculation
  - Predict “typical” circuit delay, but check the result using worst-case assumptions
  - Clock frequency can be increased, but with additional cycles required for error recovery



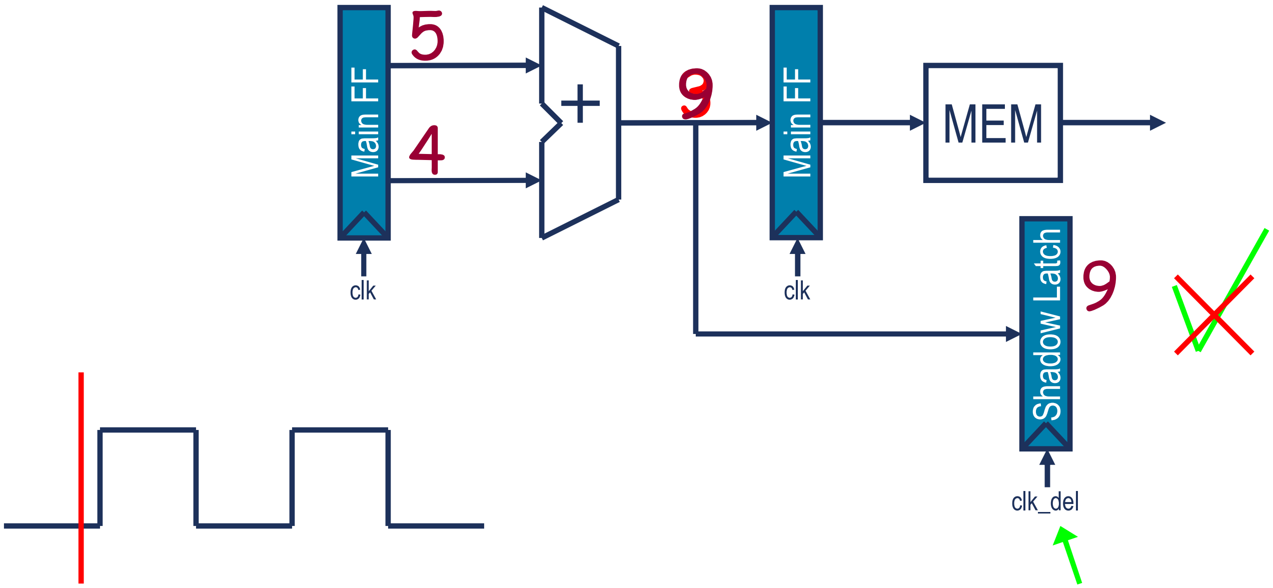
One size does not fit all



Variance of silicon, ambient, and temporal workload conditions

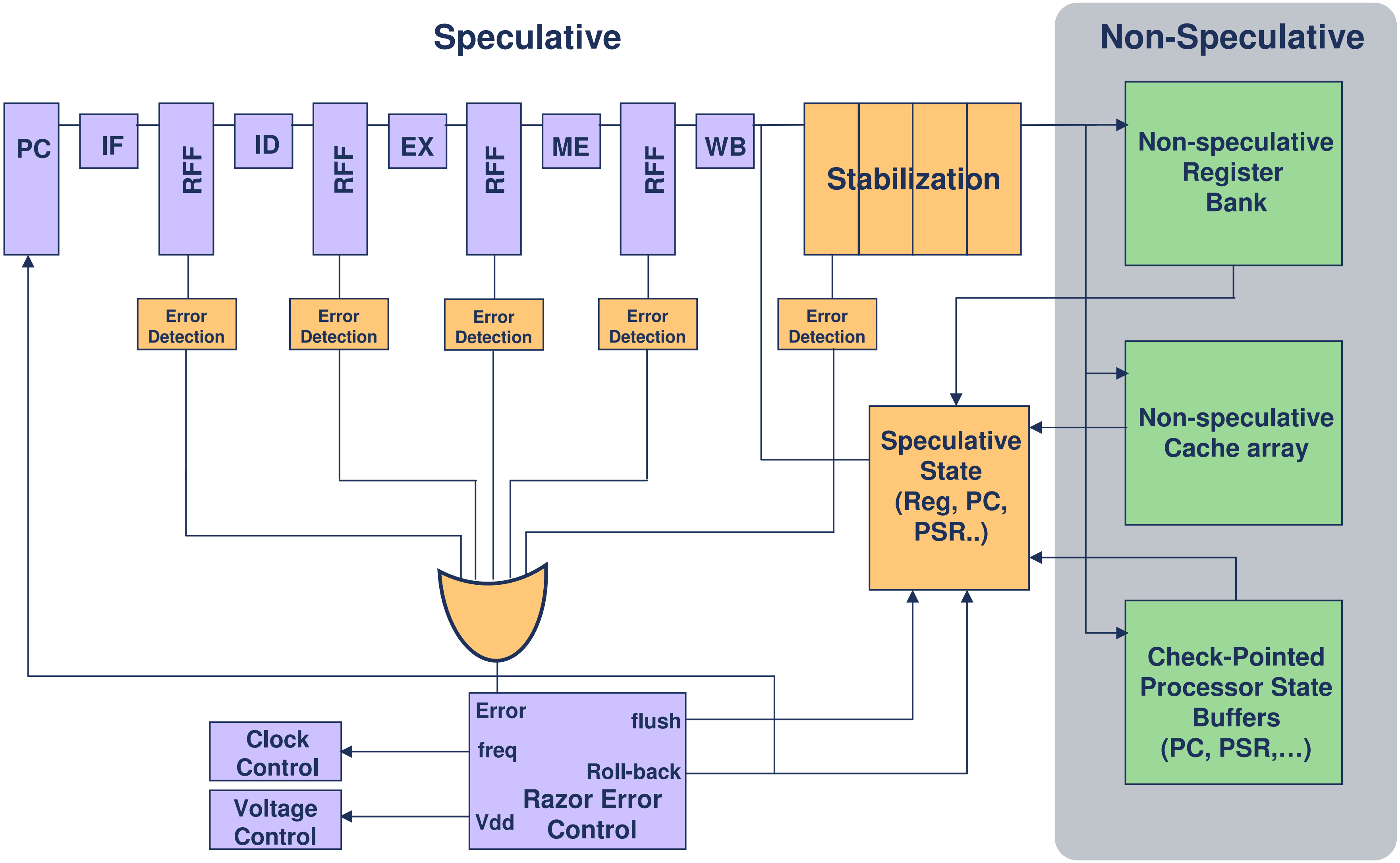
- Conventional design is too conservative with operating margins
- Tune optimal operating points over life-time of each device
- Need fault tolerance to be able to find best operating point
  - Design must be pushed over the edge to learn where the edge is

The high level idea



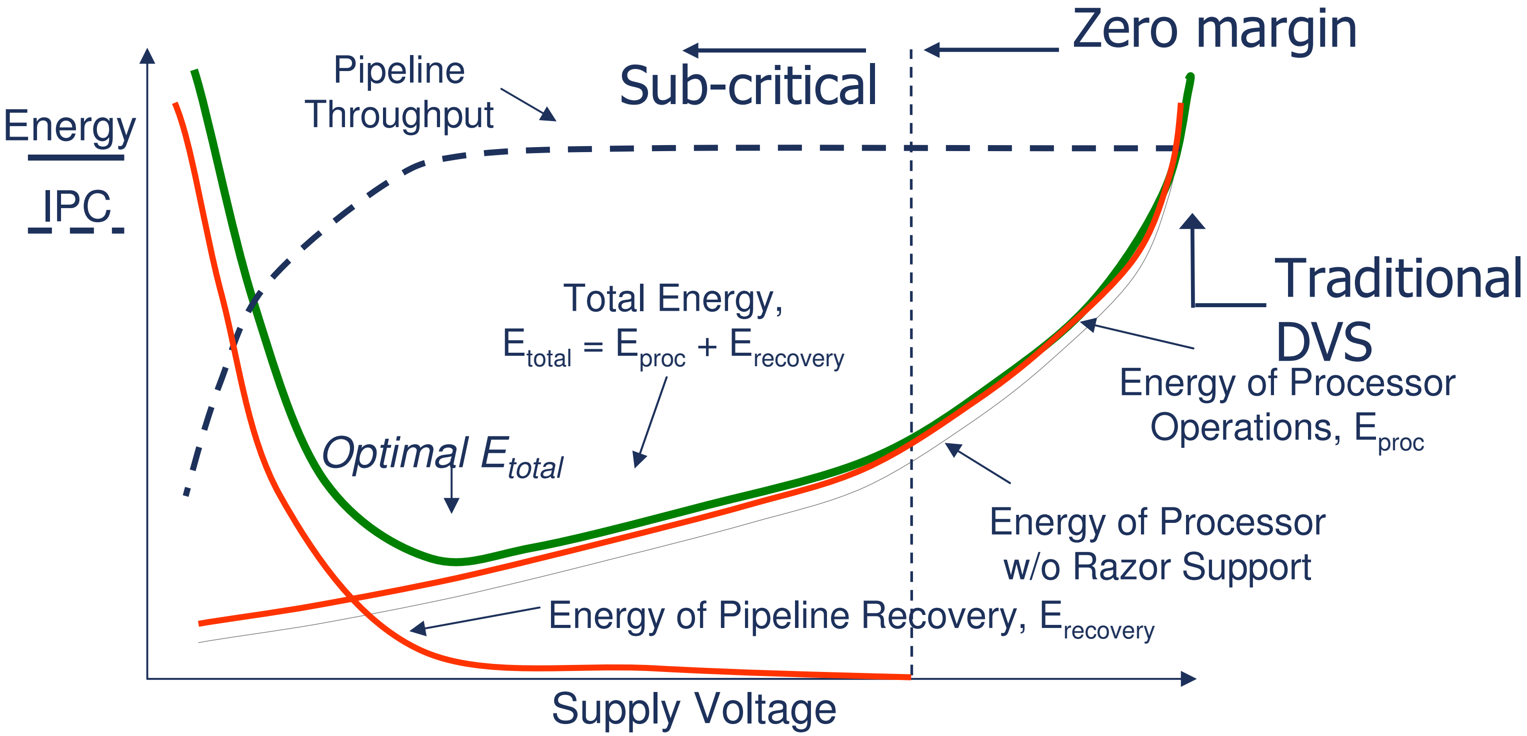
- Double-sampling metastability tolerant latches detect timing errors
- Microarchitectural support restores state
  - Timing errors are similar to branch mispredictions

Razor added to a standard pipeline

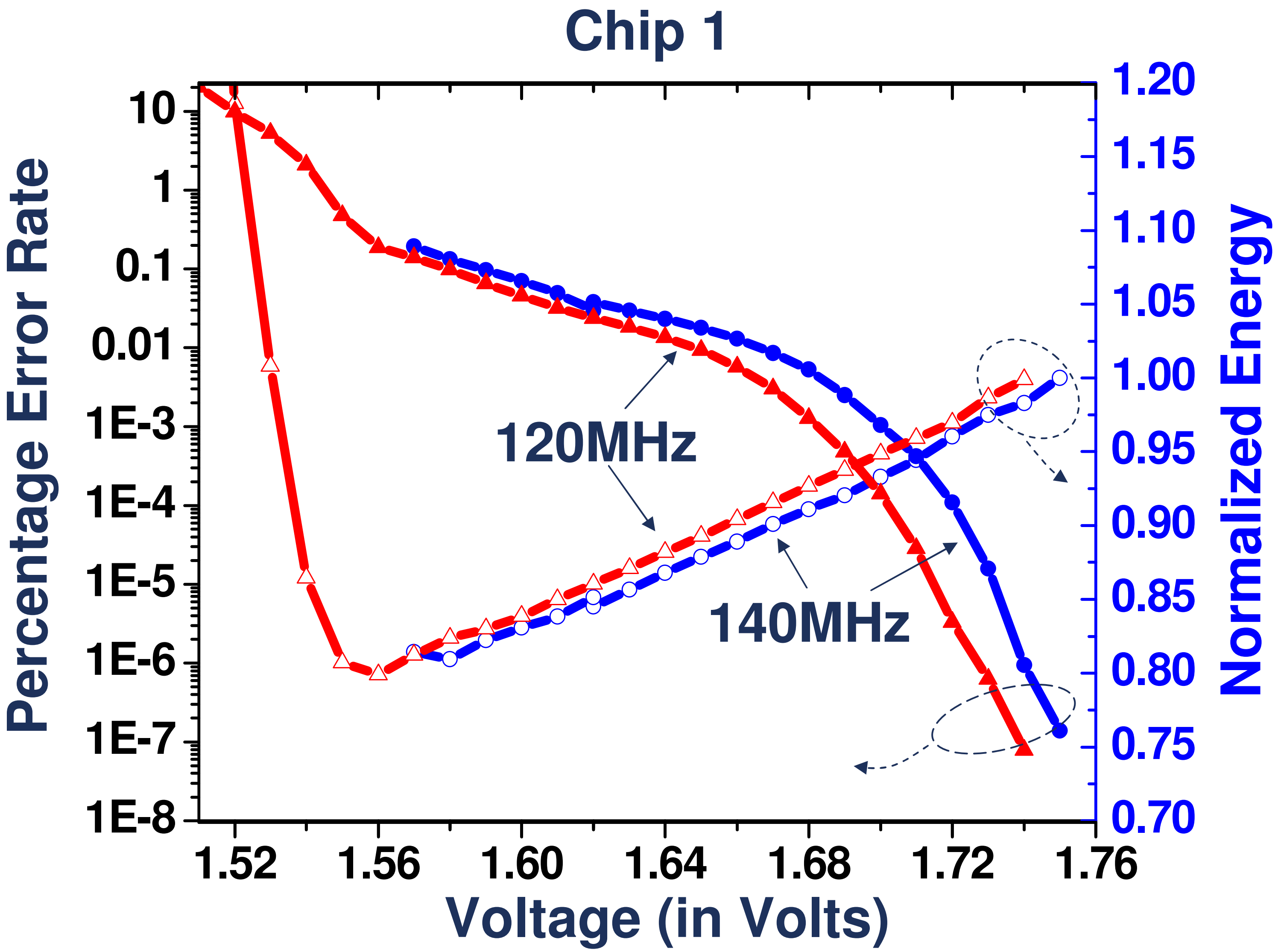


DVS using Razor

- Dynamic Voltage Scaling
  - Scale processor voltage and frequency based on run-time requirements
  - How low can the supply voltage be scaled?
  - Traditional DVS uses delay chain padded with worst-case safety margins
- Safety margins reduce possible energy savings
- Worst-case conditions are extremely rare
- Razor DVS uses in-situ error detection and correction to eliminate safety margins

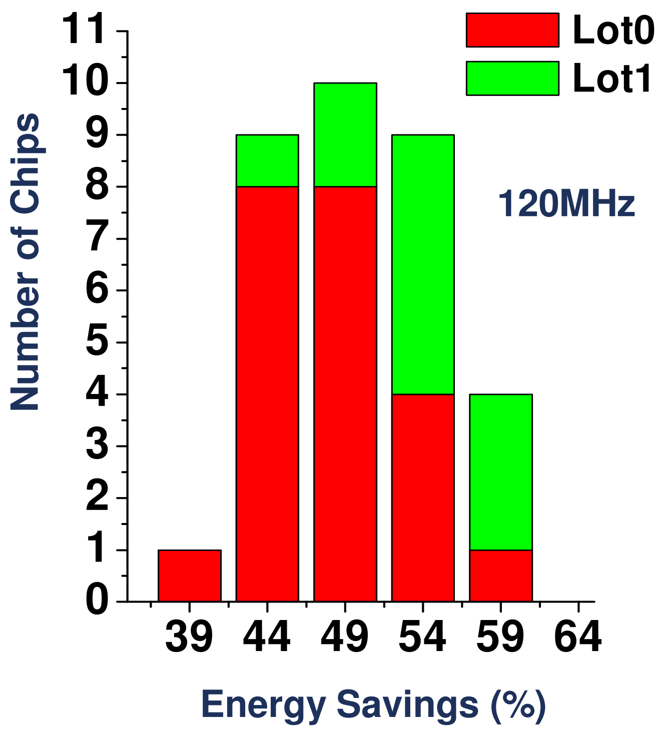


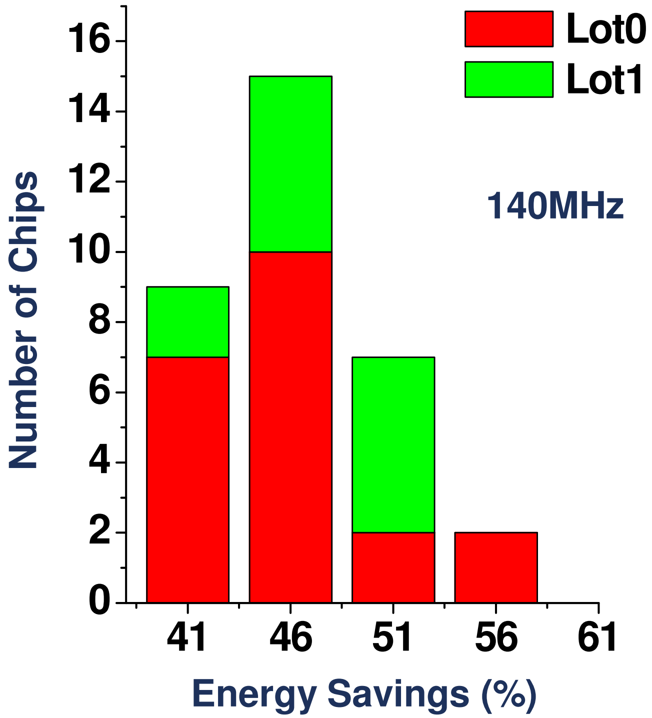
Chip 1 Measurement Results



Energy use is normalized to the amount at the point of first failure

Can one afford not to be energy efficient?!





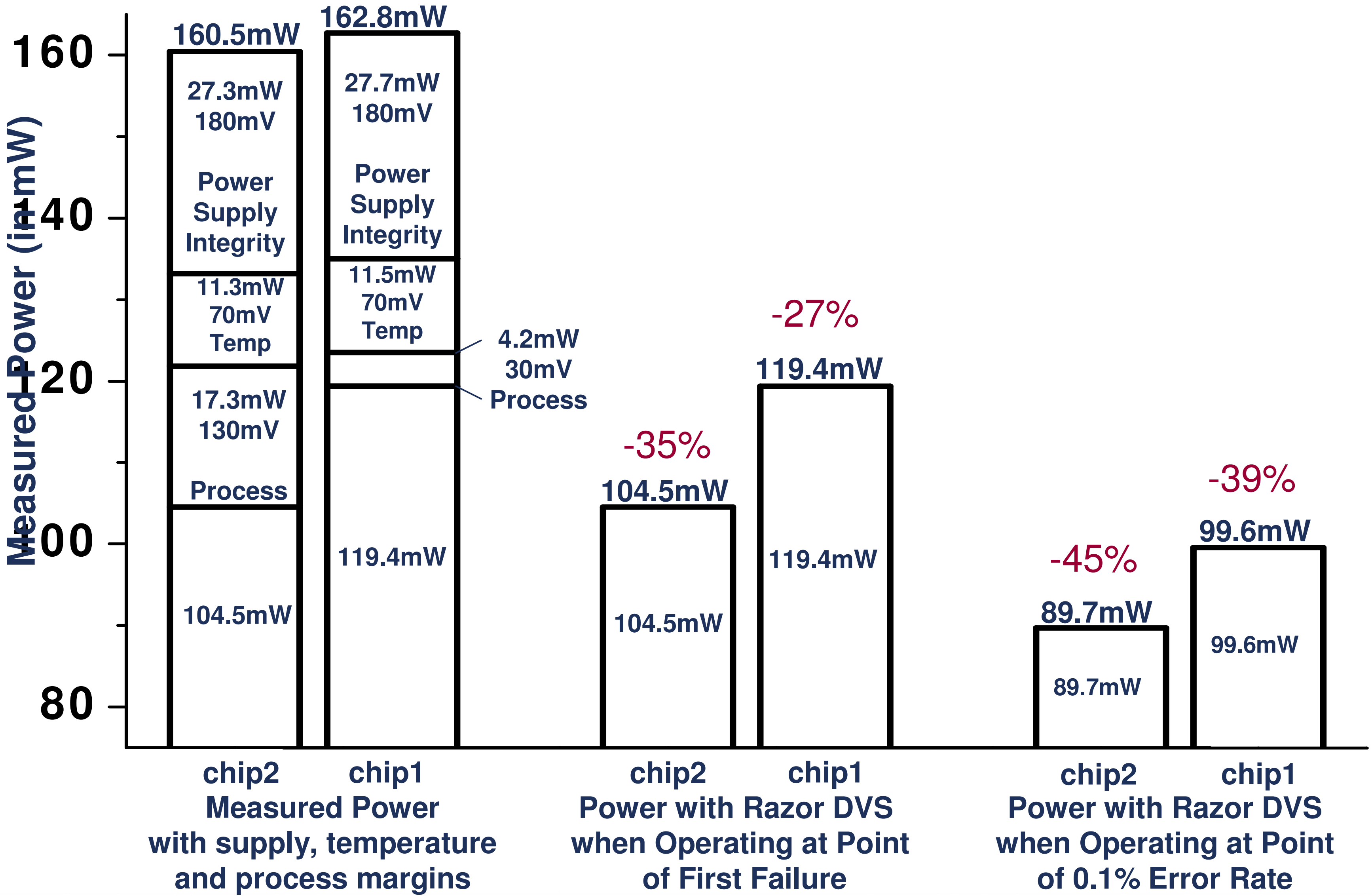
Under typical case conditions all chips are at least 39% more energy efficient

- Worst-case design uses margins for corners that are very infrequent, or even impossible

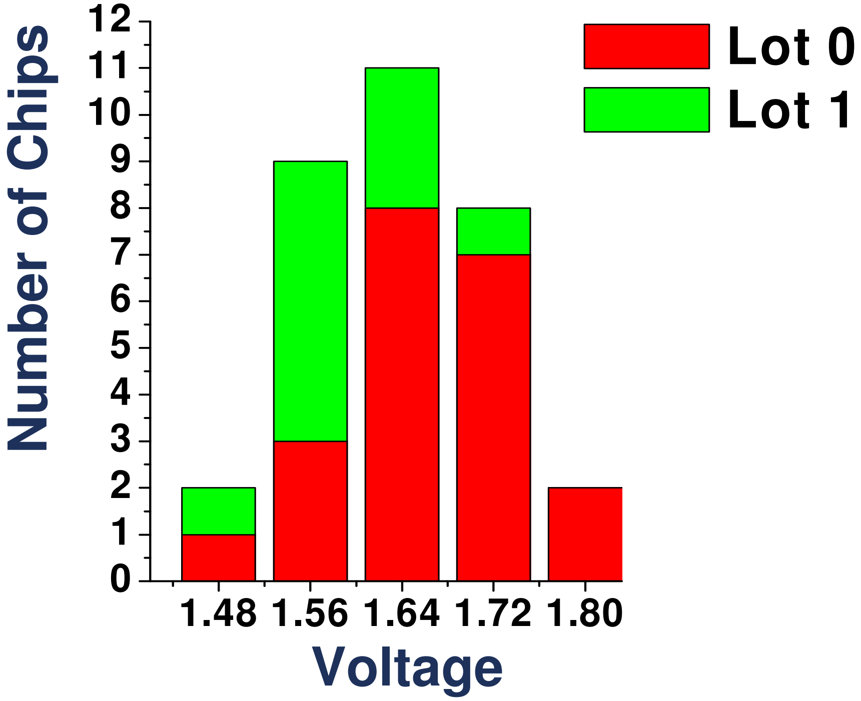
Typical-case operation requires an understanding of when and how systems break

- Razor specifies the microarchitectural requirements

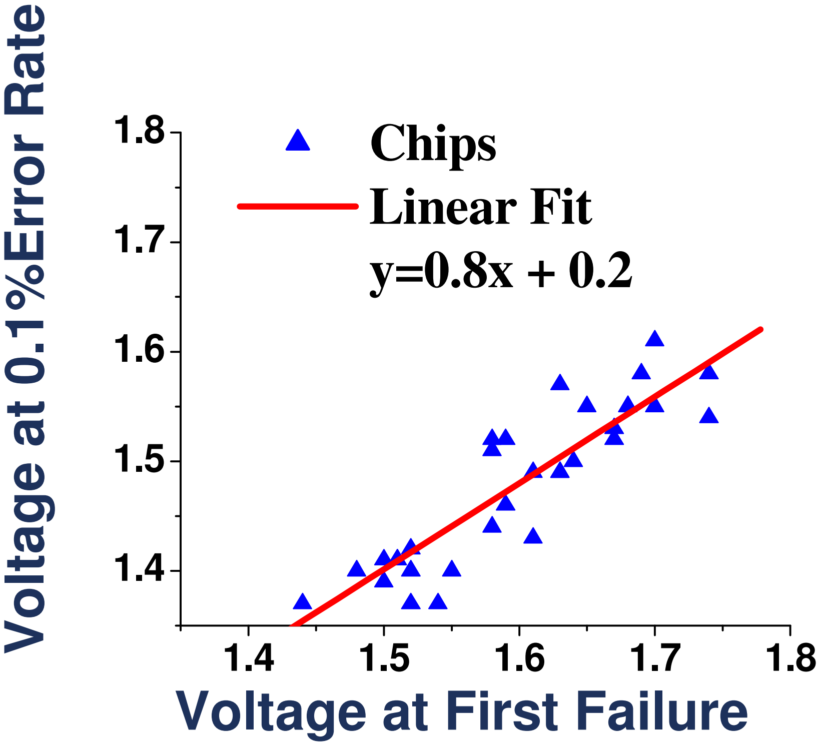
Erring occasionally is most energy efficient



Impact of Process Variation



Distribution of
Point of First Failure

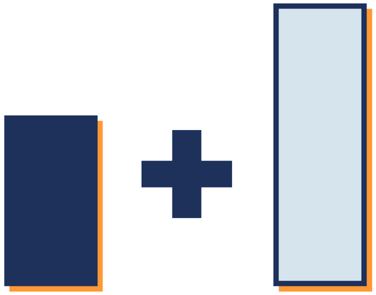


Point of 0.1% Error Rate
vs
Point of First Failure

No snake oil

- **Deep submicron processes are problematic**
  - Process variation, IR-drop, temperature fluctuation, model uncertainty
- **Design-time solution: larger built-in safety margins**
  - Yields lower performance, higher cost, higher power devices
- **Most severe problems are also the least frequent**
  - Soft errors, capacitive, inductive noise, charge sharing, floating body effect…
  - Only pay a penalty when the problem actually occurs!

- **Razor removes design-time safety margins at run-time**
  - Worst-case margins are always preserved (but moved off the critical path)
  - Design-time certainty about full range of operation
  - Improvements in worst-case characterization improve the operating range and may reduce the deployment overhead of Razor



Conclusions

Need to find creative solutions for design problems

- To be able to live up to technology expectations

Focus on how to enable typical-case operation

- Worst-case may be much, much worse than typical

Speculation on correctness

- Timing, SEU, wearout (?)

Trade-off margining and fault tolerance

Fin