On the Rules of Low-Power Design

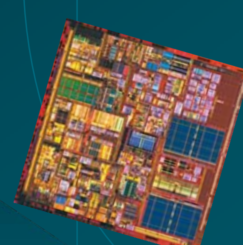
(and How to Break Them)

Prof. Todd Austin

Advanced Computer Architecture Lab
University of Michigan

austin@umich.edu

Once upon a time…



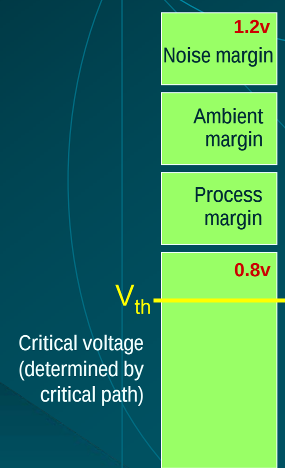
Rules of Low-Power Design

$P = aCV^2f + VI_{leak}$

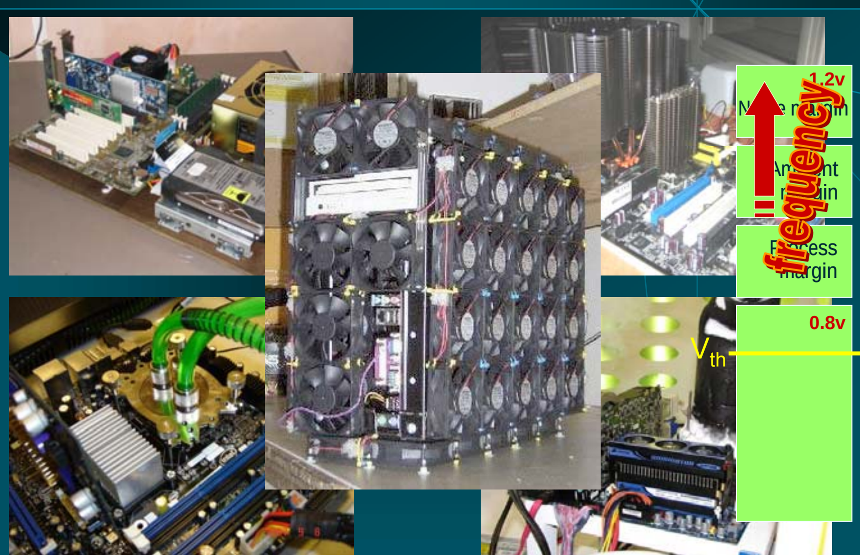
1. Minimize switching activity
2. Design for lower load capacitance
3. Reduce frequency
4. Reduce leakage

and the most important of all:

5. Decrease supply voltage!



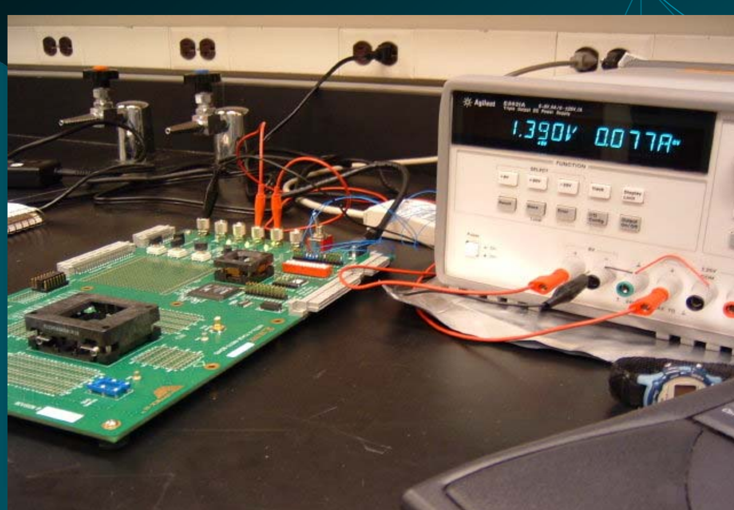
Overclockers Break the Rules

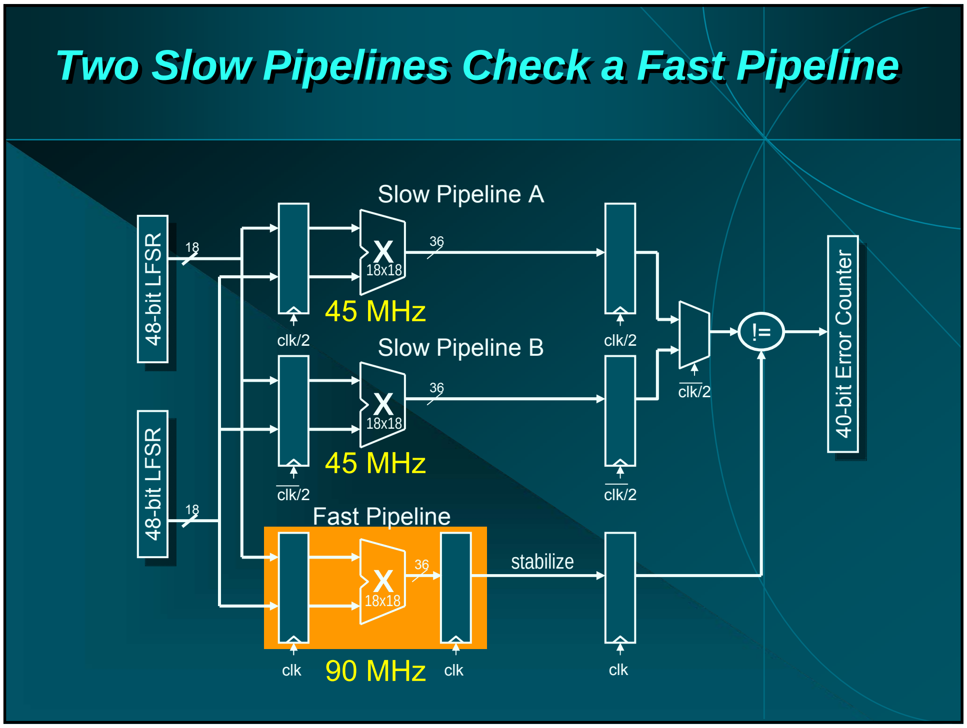


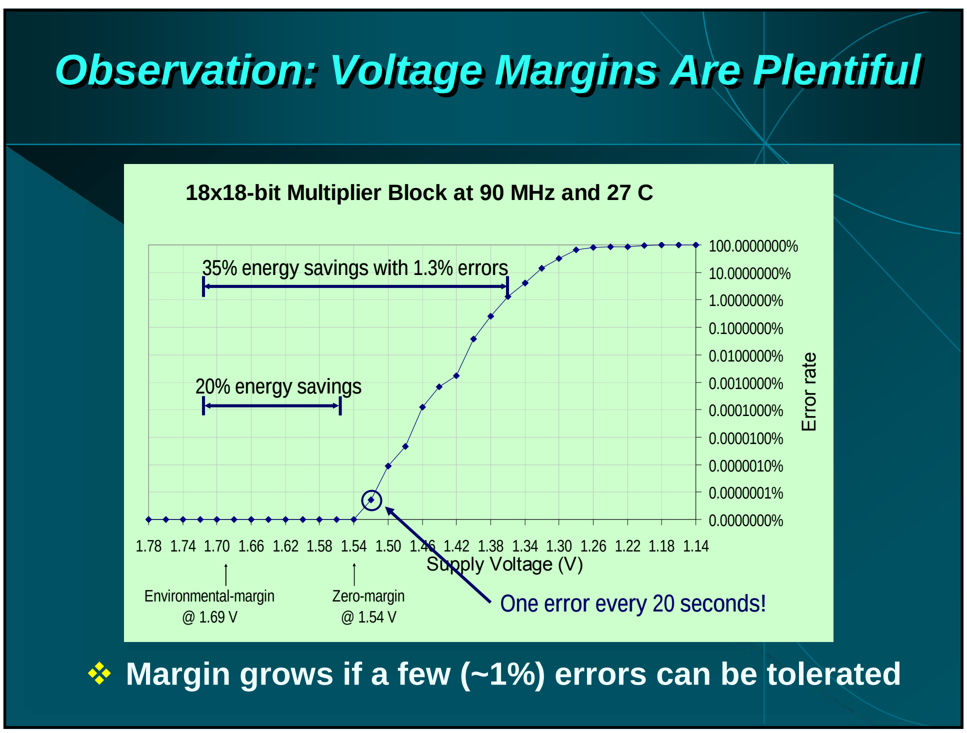
Goals of This Presentation

- Review some of the rules of low-power design ✔
- Show how clever designs can break these rules
  - Razor resilient circuits
  - Subliminal subthreshold voltage processor
- Highlight the benefits of taking a rule-breaking approach to technical research

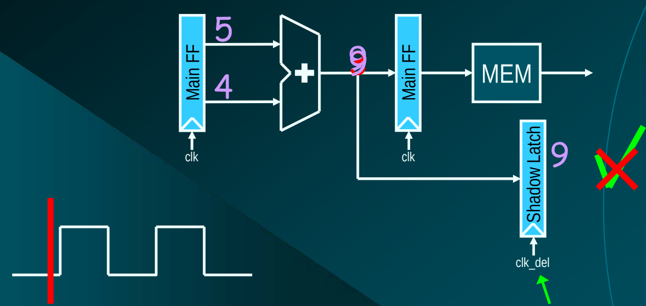
Investigating Overclocking




Two Slow Pipelines Check a Fast Pipeline
Slow Pipeline A
48-bit LFSR
18
X
18x18
36
45 MHz
clk/2
Slow Pipeline B
45 MHz
clk/2
!=
40-bit Error Counter
48-bit LFSR
Fast Pipeline
stabilize
clk
90 MHz

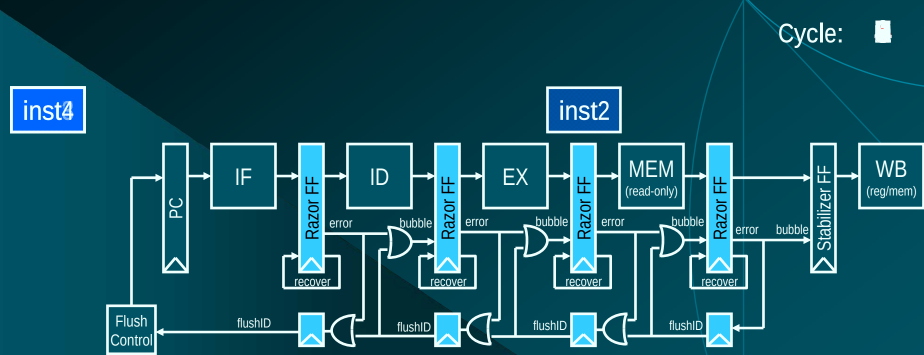

Observation: Voltage Margins Are Plentiful
18x18-bit Multiplier Block at 90 MHz and 27 C
35% energy savings with 1.3% errors
20% energy savings
100.0000000%
10.0000000%
1.0000000%
0.1000000%
0.0100000%
0.0010000%
0.0001000%
0.0000100%
0.0000010%
0.0000001%
0.0000000%
Error rate
1.78 1.74 1.70 1.66 1.62 1.58 1.54 1.50 1.46 1.42 1.38 1.34 1.30 1.26 1.22 1.18 1.14
Supply Voltage (V)
Environmental-margin @ 1.69 V
Zero-margin @ 1.54 V
One error every 20 seconds!
Margin grows if a few (~1%) errors can be tolerated

Razor Resilient Circuits



- Double-sampling metastability tolerant latches detect timing errors
  - Second sample is correct-by-design
- Microarchitectural support restores program state
  - Timing errors treated like branch mispredictions

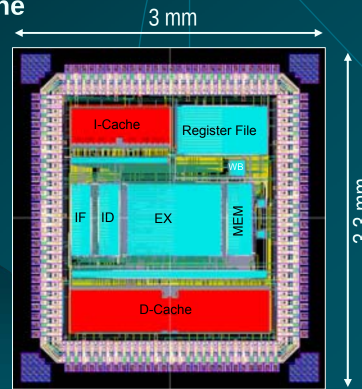
Distributed Pipeline Recovery



- Builds on existing branch prediction framework
- Multiple cycle penalty for timing failure
- Scalable design as all communication is local

Razor Prototype Design

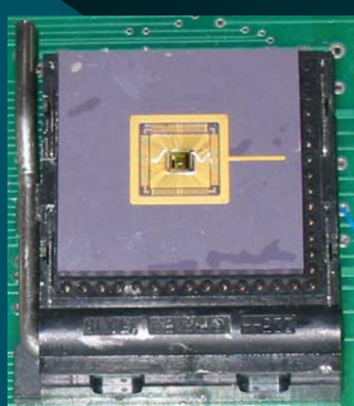
- Six stage 64-bit Alpha pipeline
  - 200MHz in 0.18mm @ 1.8V
  - tunable via sw from 200-50MHz, 1.8-1.1V
- 32-entry, 3-port RF, 8K I-Cache/8K D-Cache
- Branch-not-taken branch predictor
- Full scan capability



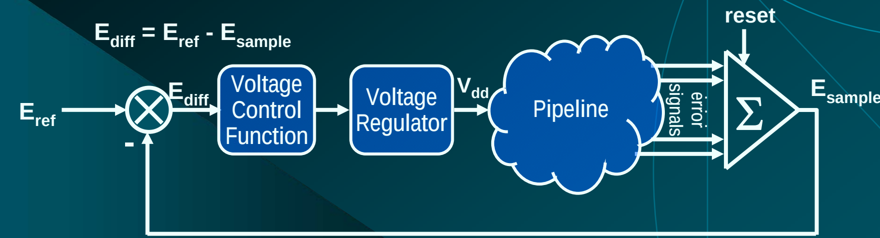
Razor overhead:

- 192 Razor FF out of 2408 (9%)
- Error-free power overhead:
  - Razor flip-flops: < 1%
  - Short path buffer: 2.1%
- Recovery power overhead:
  - 18x an inst, for pipeline recovery

Razor Prototype Testbed

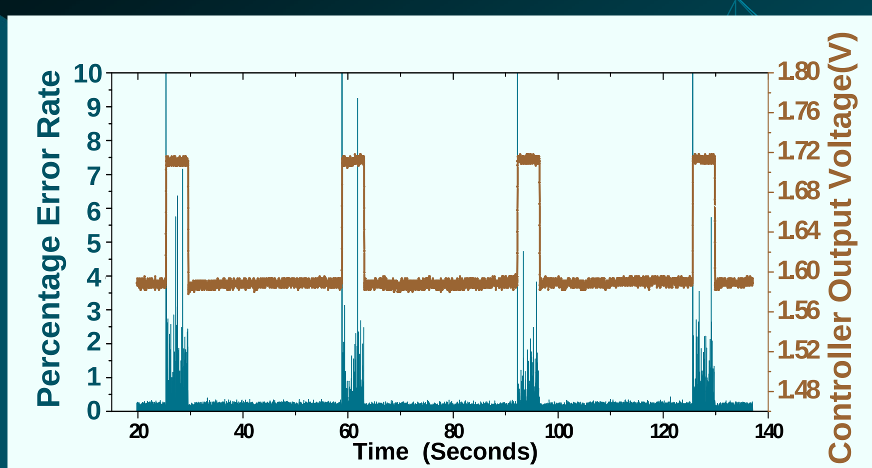


Razor-Based Dynamic Voltage Scaling

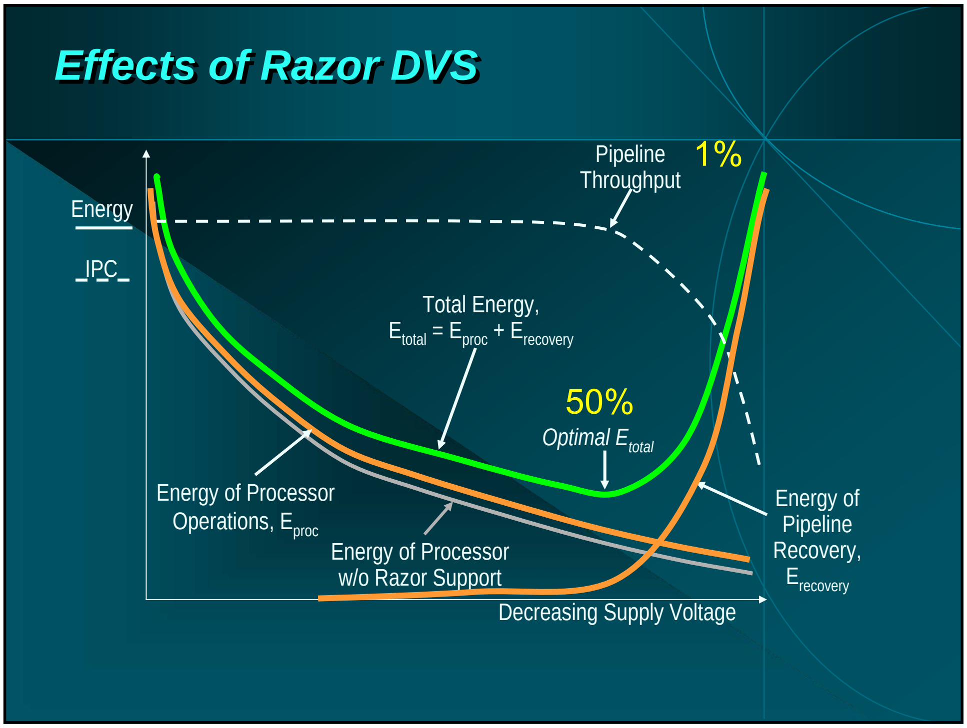


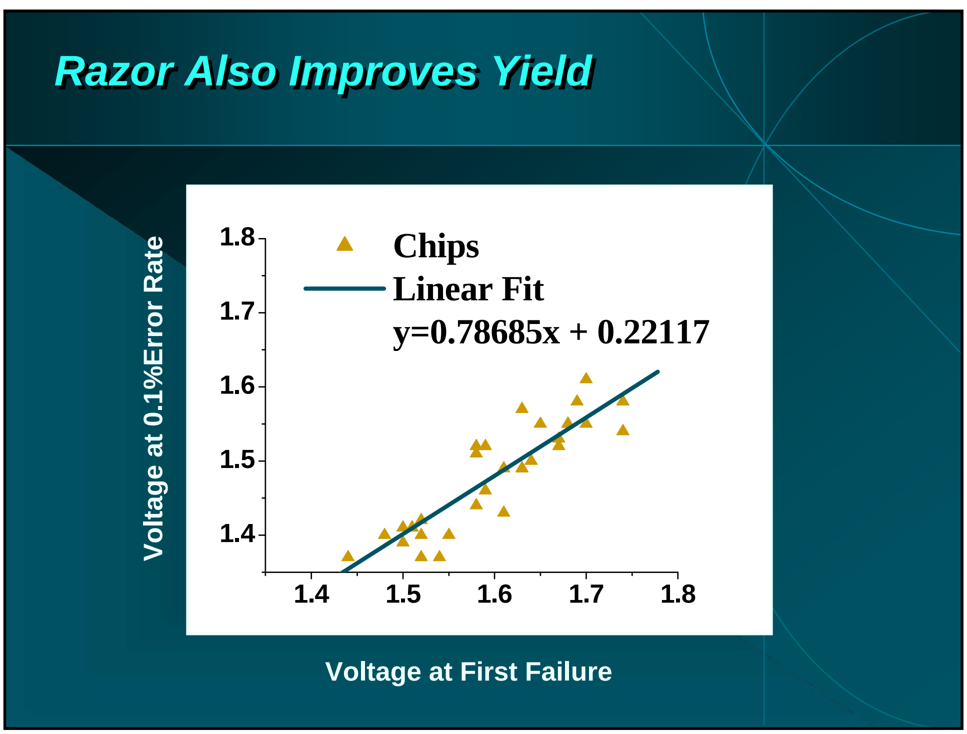
- Current design utilizes a very simple *proportional* control function
  - Control algorithm implemented in software

Example Voltage Controller Response



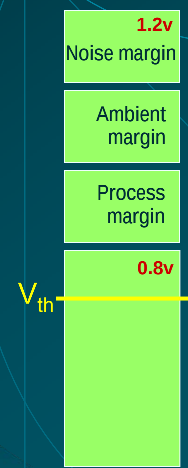
- Two minute snapshot of a 15 minute run


Effects of Razor DVS
Pipeline Throughput
1%
Energy
IPC
Total Energy,
E_total = E_proc + E_recovery
50%
Optimal E_total
Energy of Processor Operations, E_proc
Energy of Processor w/o Razor Support
Energy of Pipeline Recovery, E_recovery
Decreasing Supply Voltage


Razor Also Improves Yield
Chips
Linear Fit
y=0.78685x + 0.22117
1.8
1.7
1.6
1.5
1.4
1.4
1.5
1.6
1.7
1.8
Voltage at 0.1%Error Rate
Voltage at First Failure

How Razor Breaks the Rules

- Traditional worst-case design techniques must observe margin rules for reliable operation
- Incorporating timing-error correction mechanisms allow margins to be erased
- Infrequent use of critical paths allow for even deeper cuts in V_{dd}



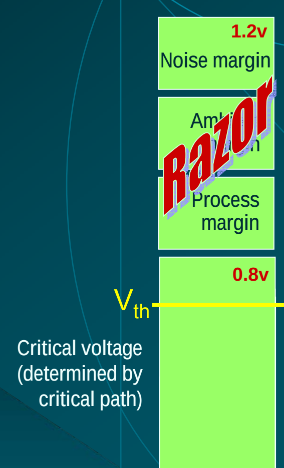
Back to the Rules

$$P = aCV^2f + VI_{leak}$$

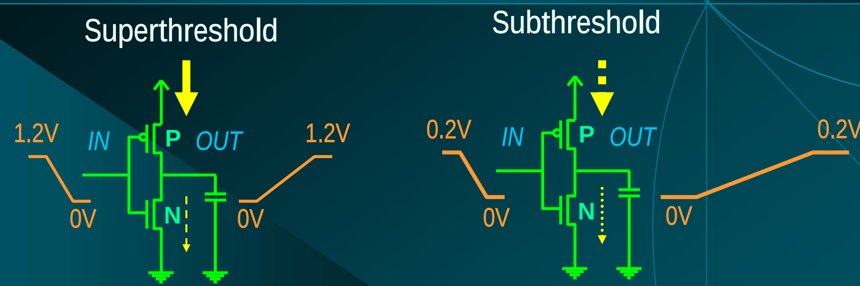
1. Minimize switching activity
2. Design for lower load capacitance
3. Reduce frequency
4. Reduce leakage

and the most important of all:

5. Decrease supply voltage!

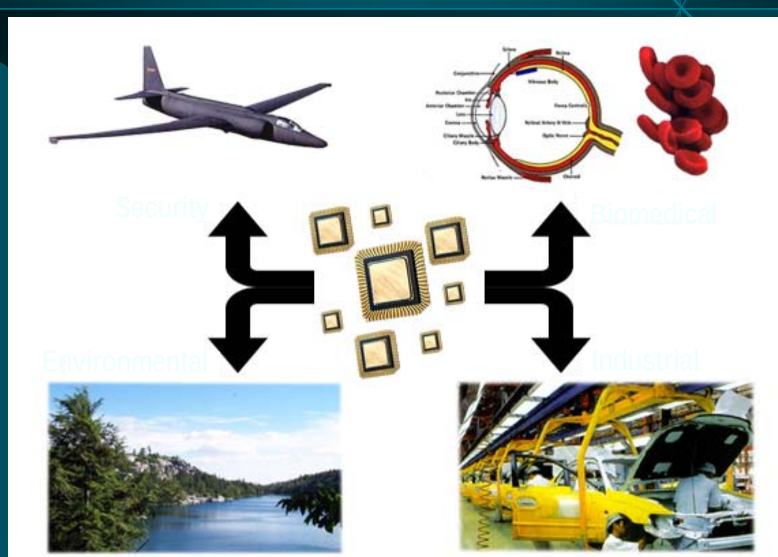


Subthreshold Circuits Break The Rules



- **Static logic still works below V_{th}**
  - **Differences in I_{leak} continue to (dis)charge outputs**
  - **But diminished I_{on}/I_{off} results in big delays**
- **Approach works if the apps are not too demanding**

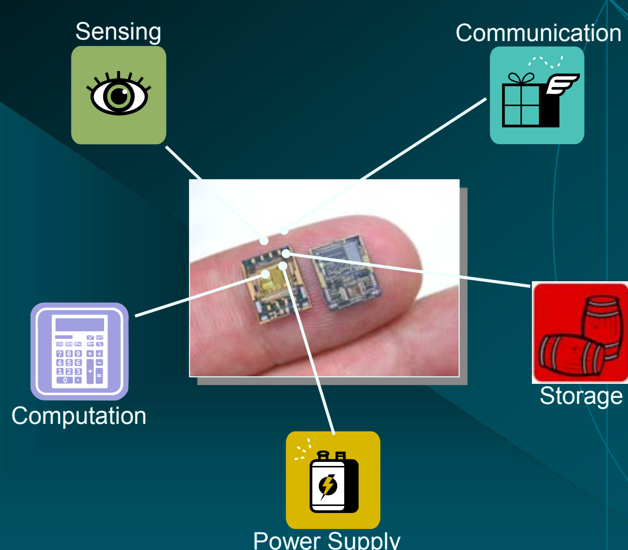
Sensing Applications



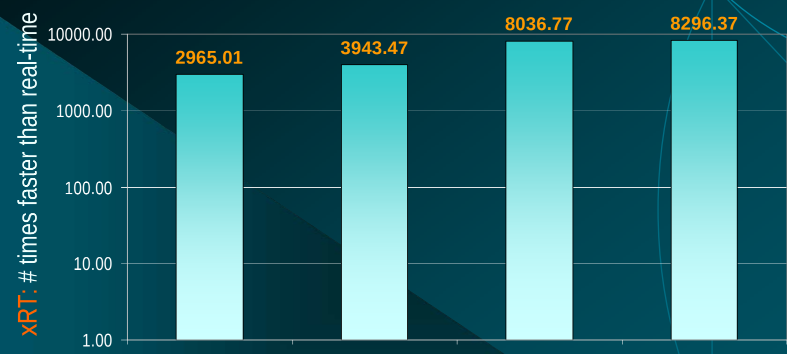
Sensor Processing Data Rates

Phenomena	Sample Rate	Sample Precision
Low Frequency Band (< 100 Hz)		
Ambient light level	0.017 - 1 Hz	16 bits
Atmospheric temperature	0.017 - 1 Hz	16 bits
Body temperature	0.1 - 1 Hz	8 bits
Natural seismic vibration	0.2 - 100 Hz	8 bits
Heart rate	0.8 - 3.2 Hz	1 bit
Wind speed	1 - 10 Hz	16 bits
Oral-nasal airflow	16 - 25 Hz	8 bits
Blood pressure	50 - 100 Hz	8 bits
Mid Frequency Band (100 Hz - 1 kHz)		
Engine temperature and pressure	100 - 150 Hz	16 bits
EOG (eyeball electrical activity)	100 - 200 Hz	16 bits
ECG (heart electrical activity)	100 - 250 Hz	8 bits
High Frequency Band (> 1 kHz)		
EMG (skeletal muscle activity)	100 - 5 kHz	8 bits
Audio (human hearing range)	15 Hz - 44 kHz	16 bits

Sensor Processor



Sensing Performance Demands are Low



Platform	ARM 720T	ARM 7TDMI	ARM 920T	ARM 1020T
Voltage (V)	1.2	1.2	1.2	1.2
Speed (Hz)	100M	133M	250M	325M

Fast Growing Leakage Complicates Design

$$E_{inst} = E_{cycle}\ CPI$$

- CPI → Cycles per Instruction
- E_{inst} → Energy per Instruction
- E_{cycle} → Energy per Cycle

$$E_{cycle} = N(½\alpha C_s V_{dd}^2 + V_{dd} I_{leak} t_{clk}$$

- α → Activity factor - average number of transistor switches per transistor per cycle
- C_s → Total circuit capacitance
- V_{dd} → Supply Voltage
- I_{leak} → Leakage current
- t_{clk} → Clock period

Fast Growing Leakage Complicates Design

$$E_{cycle} = N(\tfrac{1}{2}\alpha C_s V_{dd}^2 + V_{dd} I_{leak} t_{clk})$$

- Activity factor - average number of transistor switches per transistor per cycle
- Total circuit capacitance
- Supply Voltage
- Leakage current
- Clock period

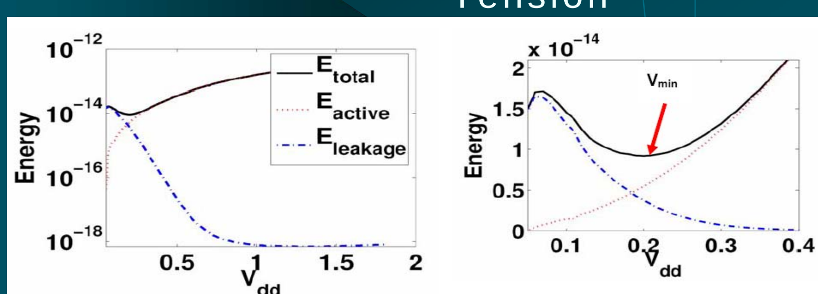
Impact of voltage reduction					
	I_{leak}	t_{clk}	E_{leak}	E_{dyn}	E_{cycle}
Superthreshold	⇓ linear	⇑ linear	~const.	⇓ quad.	⇓ quad.
Subthreshold	⇓ linear	⇑ exp.	⇑ ~exp.	⇓ quad.	???

Tension

Fast Growing Leakage Complicates Design

Impact of voltage reduction					
	I_{leak}	t_{clk}	E_{leak}	E_{dyn}	E_{cycle}
Superthreshold	⇓ linear	⇑ linear	~const.	⇓ quad.	⇓ quad.
Subthreshold	⇓ linear	⇑ exp.	⇑ ~exp.	⇓ quad.	???

Tension



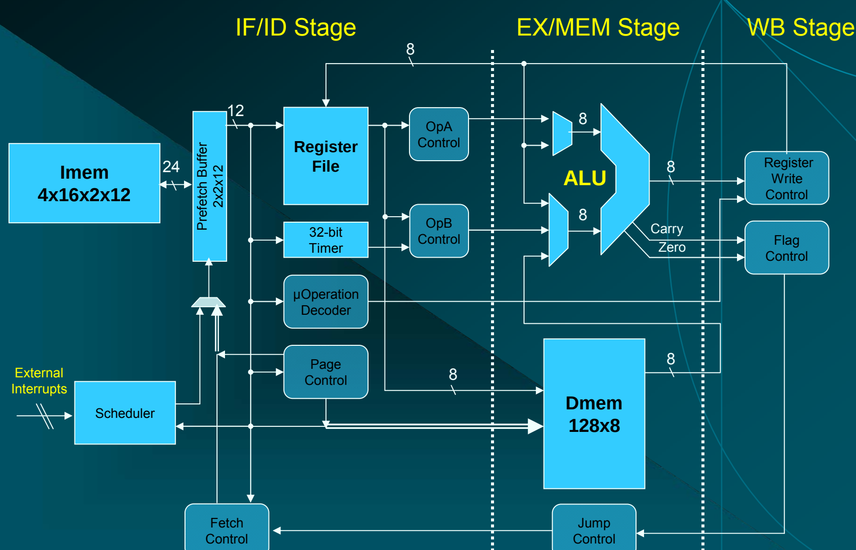
Lessons from Architectural Studies

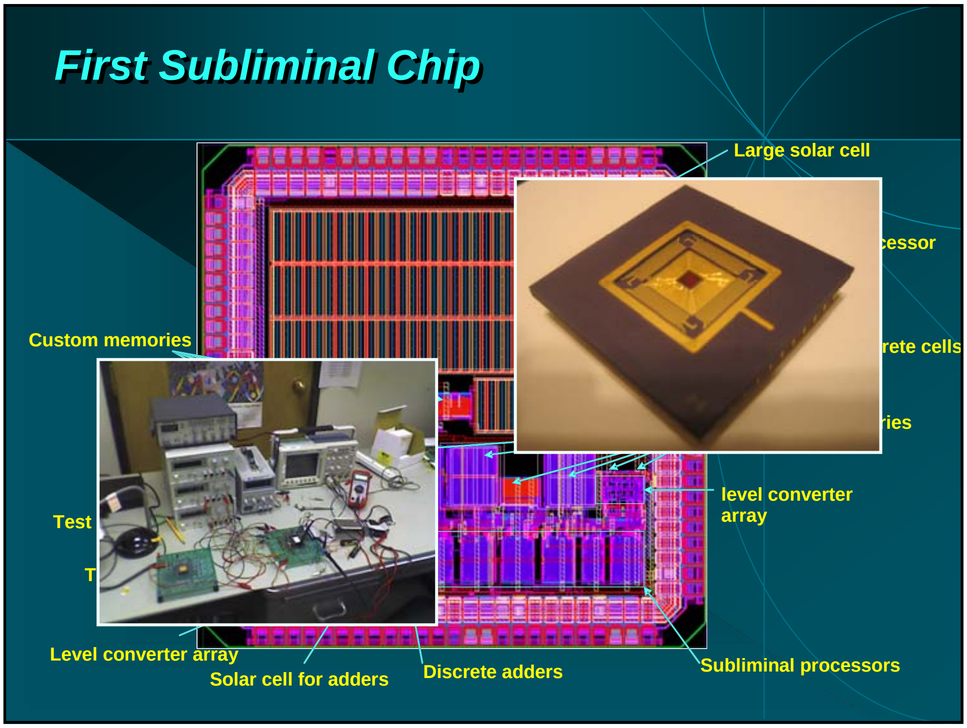
- To minimize energy at subthreshold voltages, architects must:

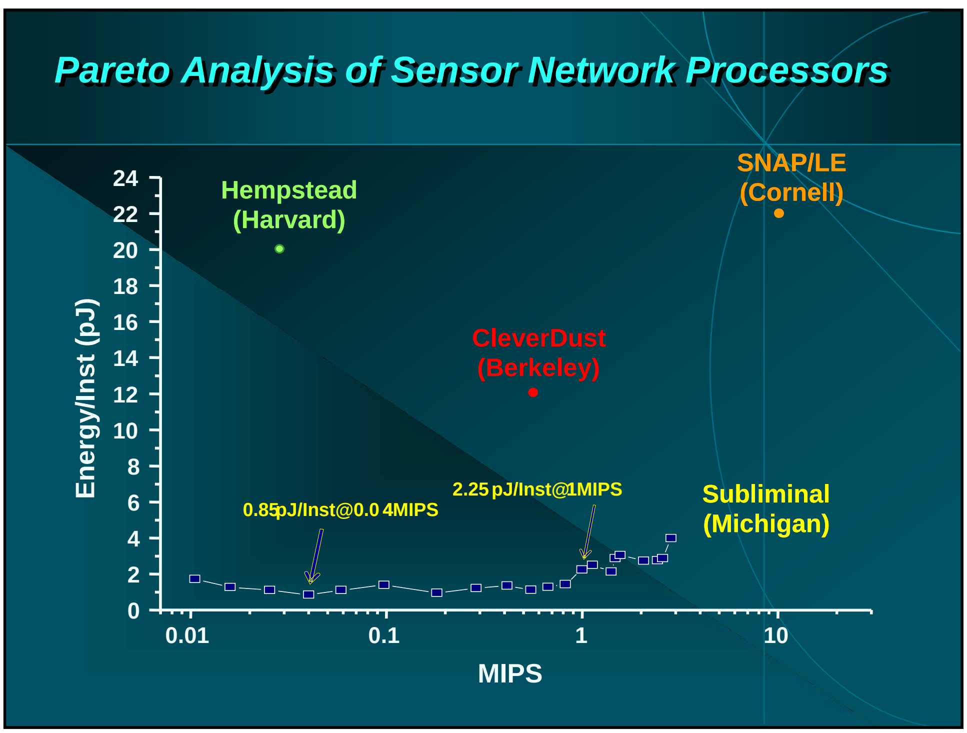
Minimize area	⇨	To reduce leakage energy per cycle
Maximize Transistor utility	⇨	To reduce V_{min} and energy per cycle
Minimize CPI	⇨	To reduce Energy per instruction

- Winning designs tend to be compromising designs that balance area, transistor utility and CPI
- Memory comprises the single largest factor of leakage energy, therefore, efficient designs must reduce memory storage requirements

Subliminal Architectural Overview




First Subliminal Chip
Large solar cell
Custom memories
Test
level converter
array
Level converter array
Solar cell for adders
Discrete adders
Subliminal processors

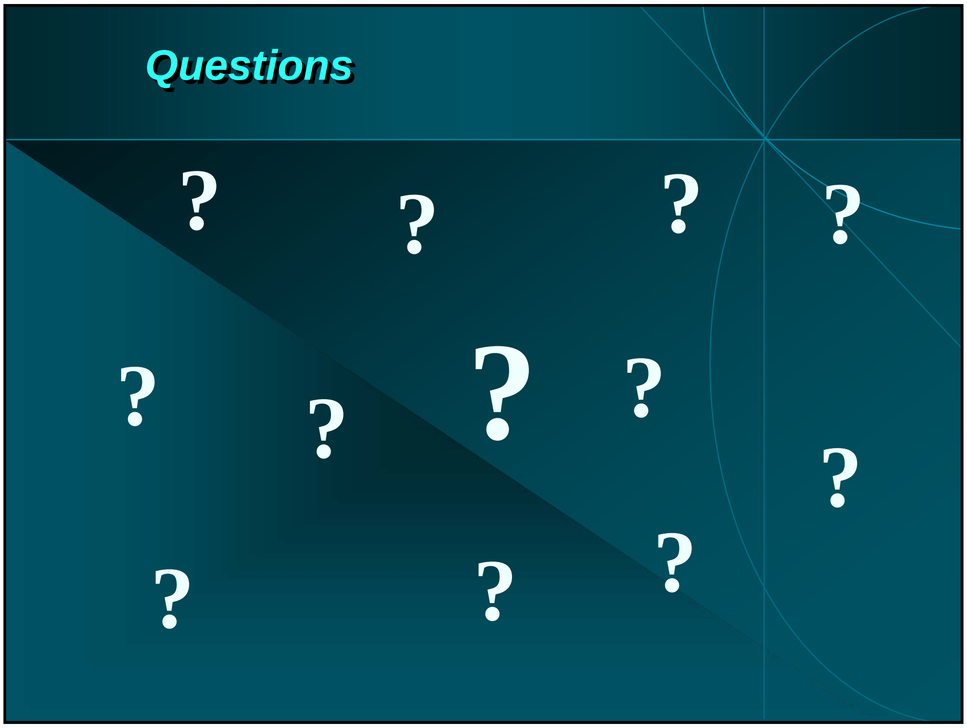

Pareto Analysis of Sensor Network Processors
Hempstead
(Harvard)
SNAP/LE
(Cornell)
CleverDust
(Berkeley)
Subliminal
(Michigan)
0.85pJ/Inst@0.04MIPS
2.25 pJ/Inst@1MIPS
Energy/Inst (pJ)
MIPS
0
2
4
6
8
10
12
14
16
18
20
22
24
0.01
0.1
1
10

How Subliminal Breaks the Rules

- Traditional circuit design relies an transistor switching to perform computation
- Static logic circuits continue to operate below V_{th} by modulating leakage currents
- Approach lends itself to low-demand sensor apps, as long as care is taken to build an efficient processor

What I Really Learned…

- A rule-breaking approach to technical research is effective and engaging
- You will find yourself on very fertile ground
  - "It is that which everyone knows is certainly true, that is indeed false."
  - "The early bird gets the worm."
  - "If you are not failing some of the time, you are not trying hard enough."
- You will more fully engage your colleagues
  - One half will think crazy idea will never work
  - One half will be intrigued (with your crazy idea)


Questions