

Low Power Systems on a Chip – Today's Challenge

Trevor Mudge

Bredt Professor of Engineering

The University of Michigan, Ann Arbor

July 2004



Outline

- Why low power? – Trends
- Dynamic power management
- Static power management
- Power and design uncertainty
- Conclusions



Why does power matter?



1983

Motorola DynaTAC 8000X

1995

Nokia 232

2003

Nokia 6600

Battery life	0.5h - 1h talk, 8h standby	1h talk, 13h standby	4h talk, 240h (>1 week) standby
Battery	Lead Acid, 500g	NiMh, 100g	Li-Ion, 21g
Weight	800g	205g	125g
Features	Talk for brokers	Talk for the masses	Talk, play, web, snap, video, organize
Price	\$3995	\$500	\$500

- The disappearing battery – despite only incremental capacity improvements: the rest of the system has become more power efficient
- Power has major impact on form factor, features, and cost



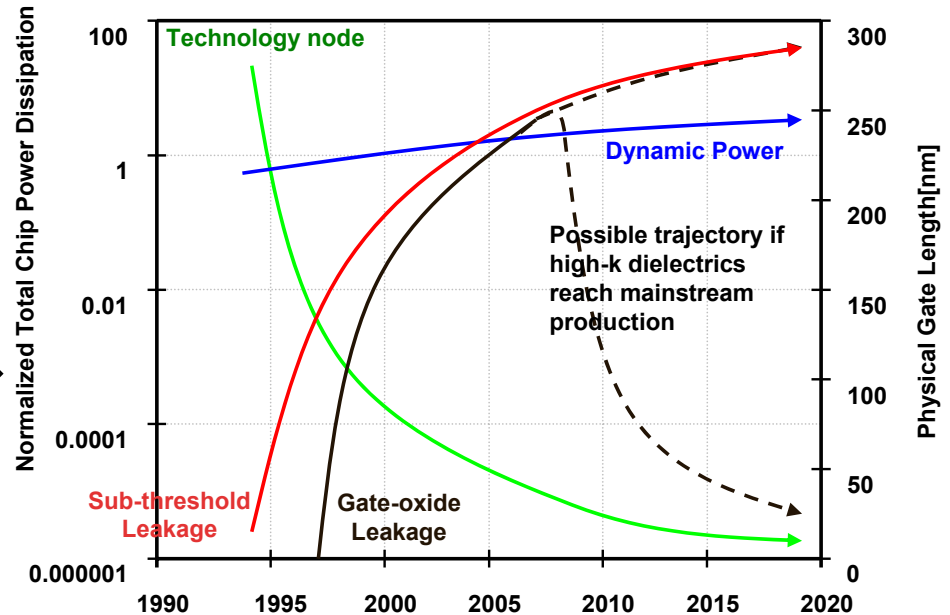
First-class design issue: Power

- What the end-users really want: supercomputer performance in their pockets...
 - Untethered operation, always-on communications
 - Driven by applications (games, positioning, advanced signal processing, etc.)
- Technology scaling trends are not in our favour:
 - Need creative ways of dealing with increasing leakage power
 - New processes are expensive
 - Diminishing performance gains from process scaling
 - Dynamic power remains high
- Solutions need to cut across traditional boundaries (SW / architecture / microarch / circuits)



Technology Trends

- Silicon is likely to be the technology of choice for the next 4-5 generations
 - 130nm → 90nm → 60nm → 45nm → 30nm
 - 2 or 3 years between generations
 - ~10 ± 2 Years
 - after 2015 there may be a paradigm shift to non-Si technology
- Consequences for power →



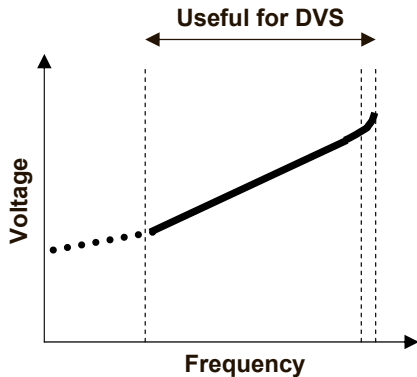
Dynamic Power

- Voltage scaling – by now a familiar story
 - relies on the quadratic law (also helps leakage)
 - familiar but not widely implemented
 - LongRun 1 & 2, SpeedStep, DPM (dynamic power mgmt/IBM)
- The quadratic law implies that parallelism is good
 - only true if leakage is not considered



CMOS Power and Energy

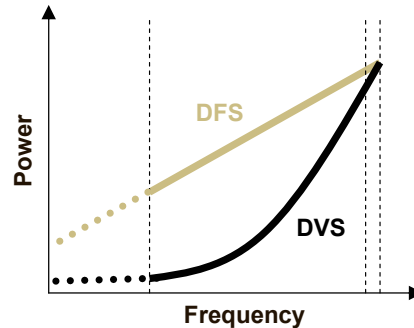
- Power and Energy consumption trends of a workload running at different frequency and voltage levels.
- DFS: frequency scaling only, DVS: frequency & voltage scaling



$$f \sim (v_{dd} - v_t)^\alpha / v_{dd}$$

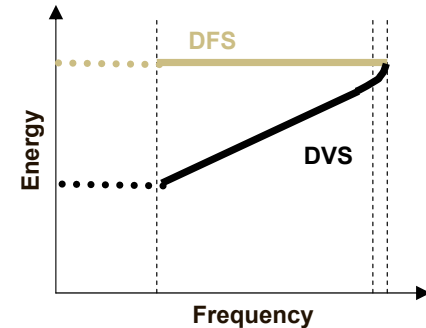
$$\alpha \approx 1.3$$

$$v_t / v_{max} \approx 0.3$$



$$P = C v_{dd}^2 f + v_{dd} I_{leak}$$

Avg. power \sim heat



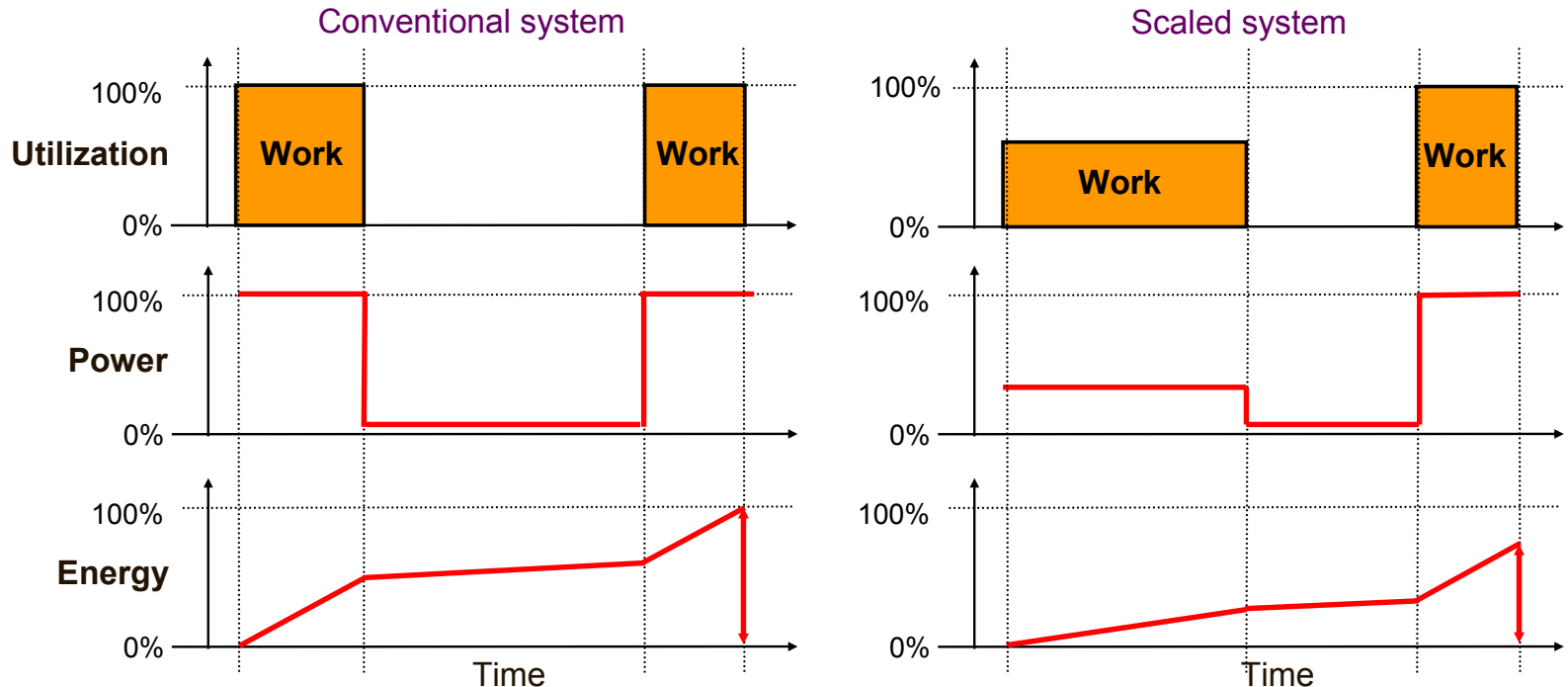
$$E = \int P_{dt}$$

Need DVS to save energy

Must reduce voltage to save energy and extend battery life



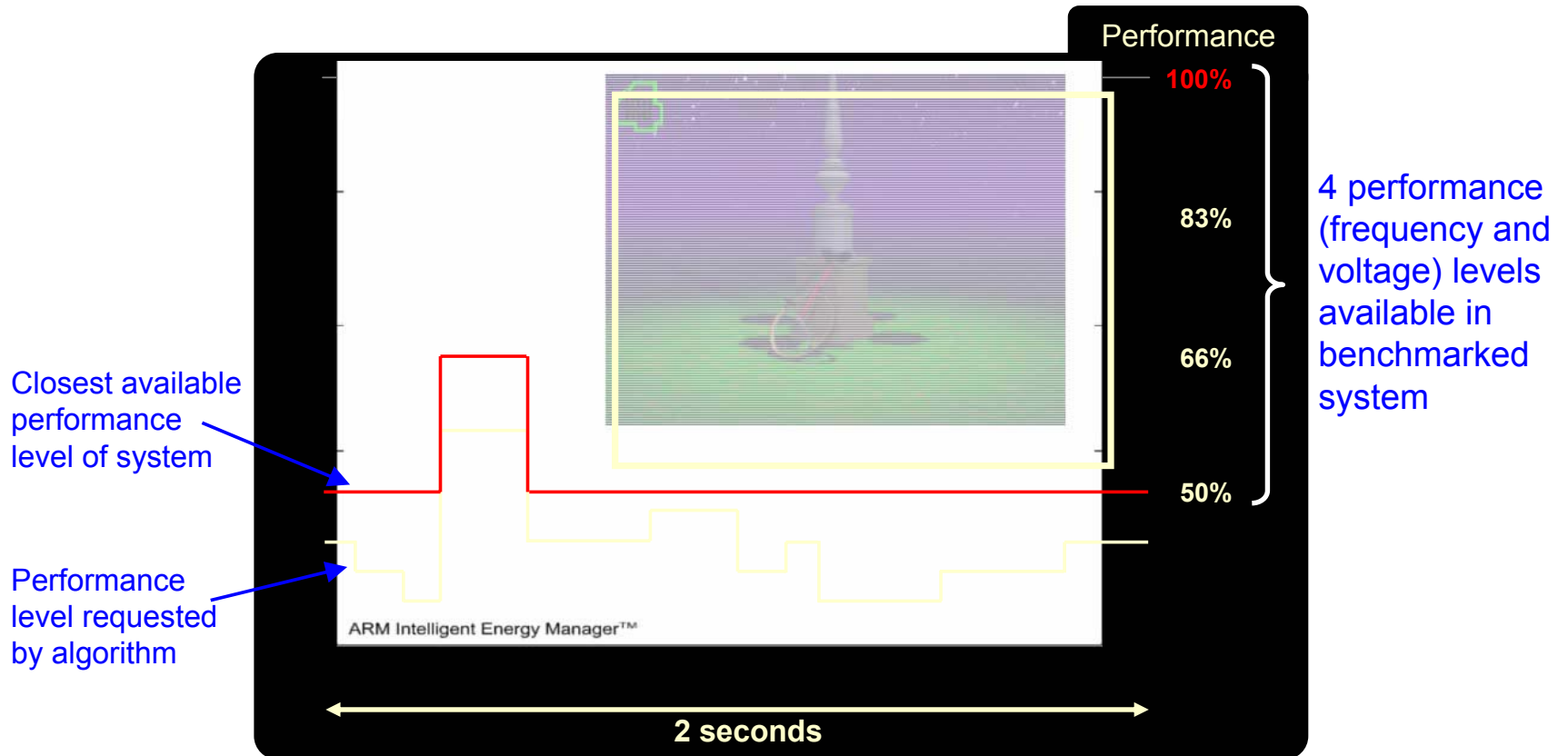
Performance scaling for energy efficiency



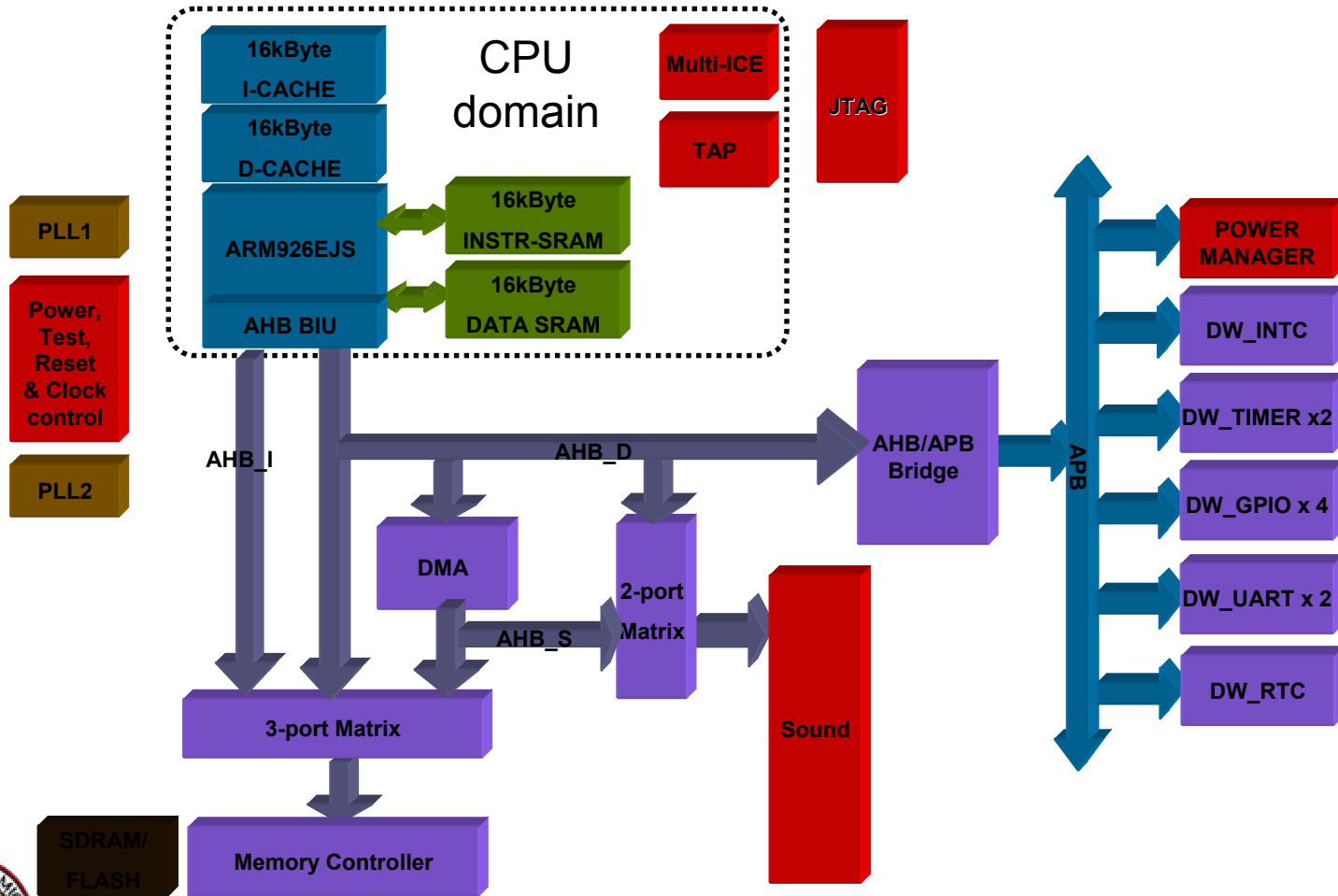
- Reduced processing rate enables more efficient operation
- Use dynamic voltage scaling (DVS) and threshold scaling (ABB)



Commercial Example: IEM

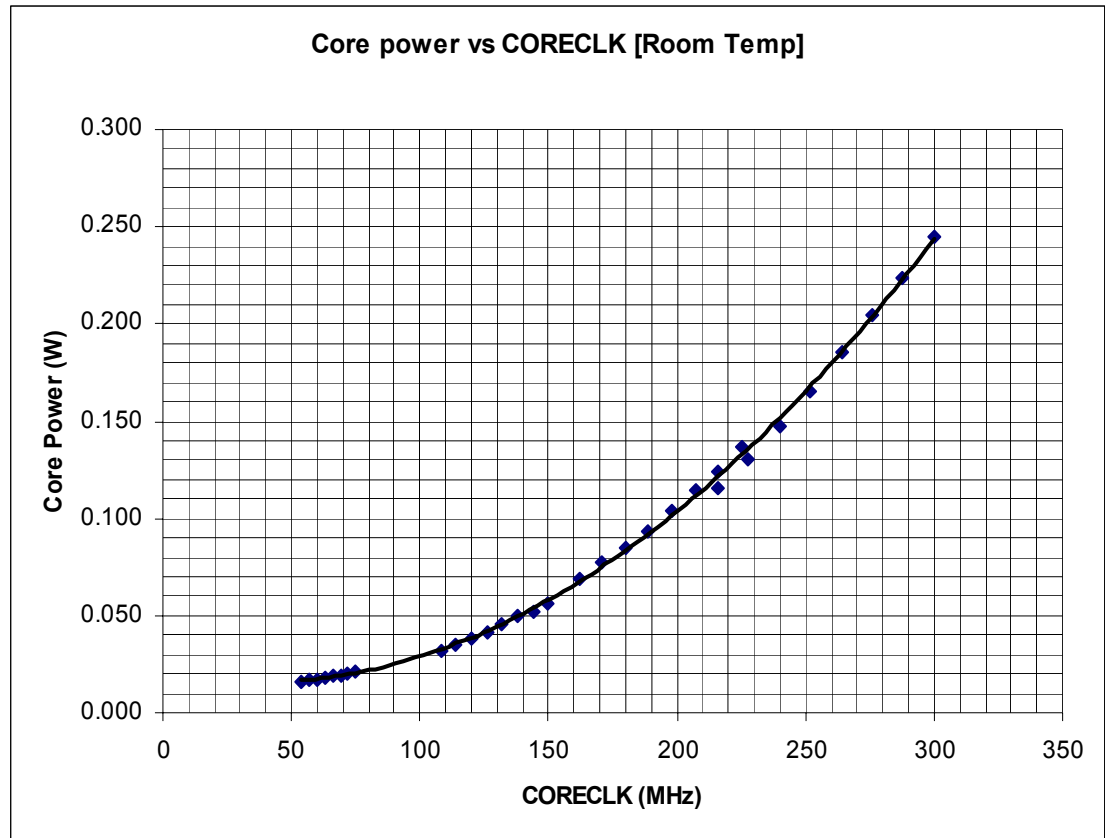


IEM test chip: AM926EJ-S core



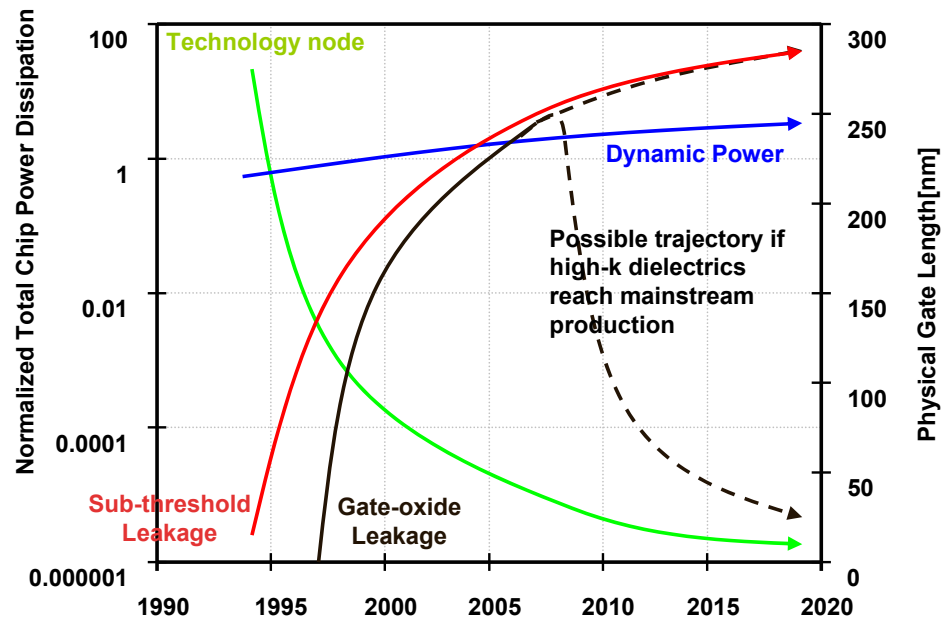
DVS926: Power Analysis

- Room Temp
- Cached workload (Dhrystone)
- CPU core V/I measurements
- Four run-time frequency divisions: 100%, 75%, 50%, 25%
- Boot time PLL settings:
 - 216MHz
 - 228MHz
 - 240MHz
 - 252MHz
 - 264MHz
 - 276MHz
 - 288MHz
 - 300MHz



Static Power

- Growing importance
- Reducing activity no longer works
- Voltage scaling still works
- Powering off works if state is not an issue
- Memory is important

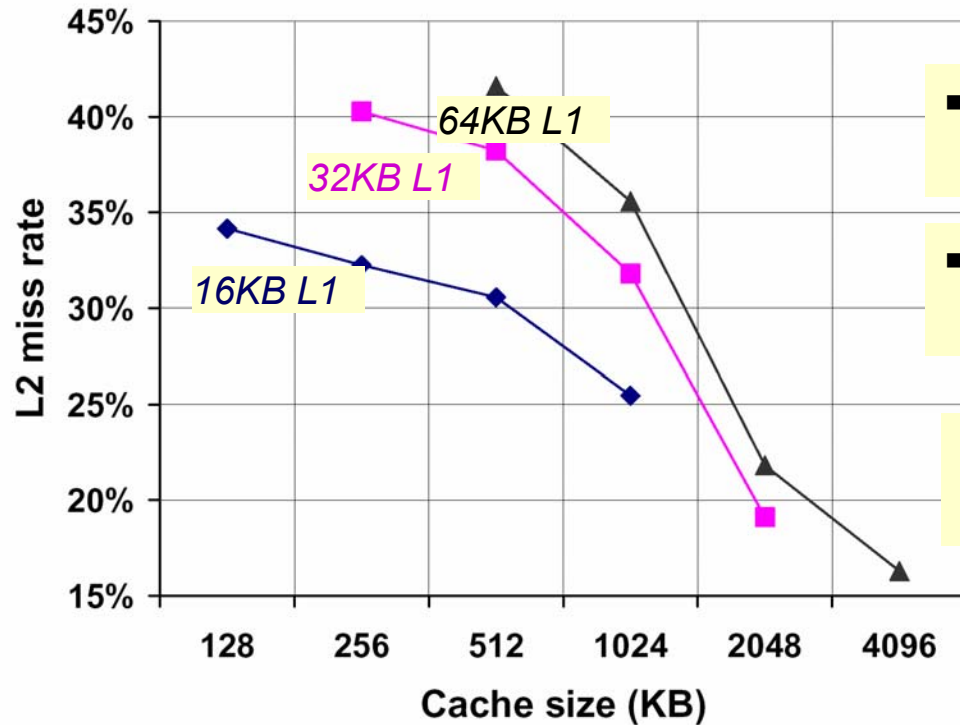


Cache Leakage Power

- On-chip caches are becoming bigger
 - 2x64KB L1 / 1.5MB L2 for Alpha 21464
 - 256KB L2 / 3MB(6MB) L3 for Itanium 2
- Increasing on-chip cache leakage power
 - feature size shrinking / V_{TH} decreasing
 - increasing fraction of leakage power by L2 & L3 caches
 - consuming constant leakage power
 - less frequent access (less dynamic power)
- We can maintain cache performance by trading cache size for power
 - counter intuitive: larger caches consuming less power



Cache Miss Statistics



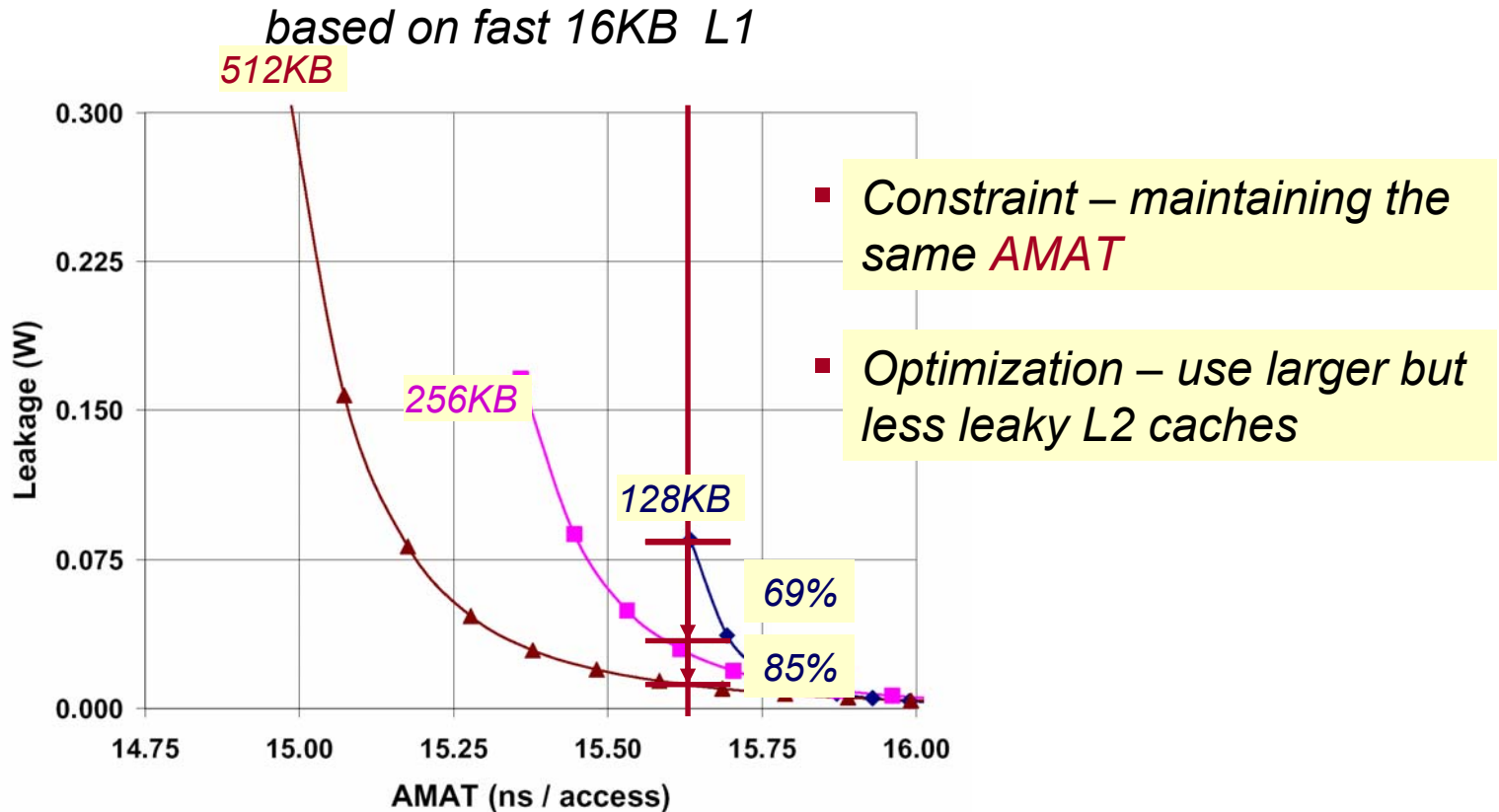
- SPEC 2000 INT/FP – average all 25 benchmarks

- Using “sim-cache” in SimpleScalar suite

- L2 miss rate is local miss rate



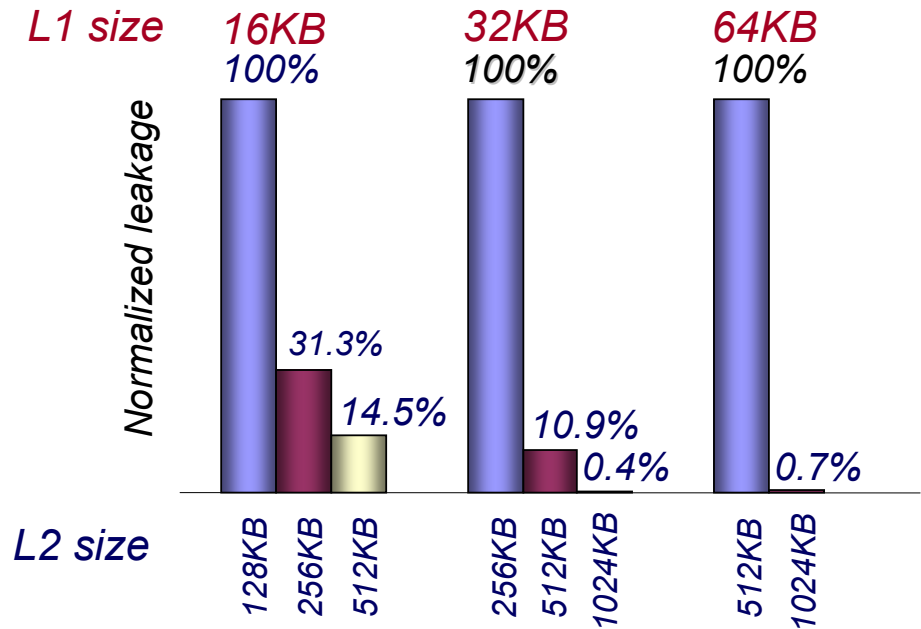
Optimizing L2 leakage at fixed L1 size



L2 Leakage Saving at Fixed L1 Size

Conclusion

- Cost- effective # of VTH for cache leakage reduction
 - depending on the target access time, but **1** or **2** high VTH's is enough for leakage reduction
- Cache leakage
 - another **design constraint** in processor design
 - trade-off among delay / area / **leakage**



L1 Leakage Reduction: Drowsy Caches

Instead of being sophisticated about predicting the working set, **reduce the penalty for being wrong**

Algorithm:

- Periodically put all lines in cache into drowsy mode.
- When accessed, wake up the line.

- Optimize across circuit-microarchitecture boundary:
 - Use of the appropriate circuit technique enables simplified microarchitectural control.
- Requirement: ***state preservation in low leakage mode.***

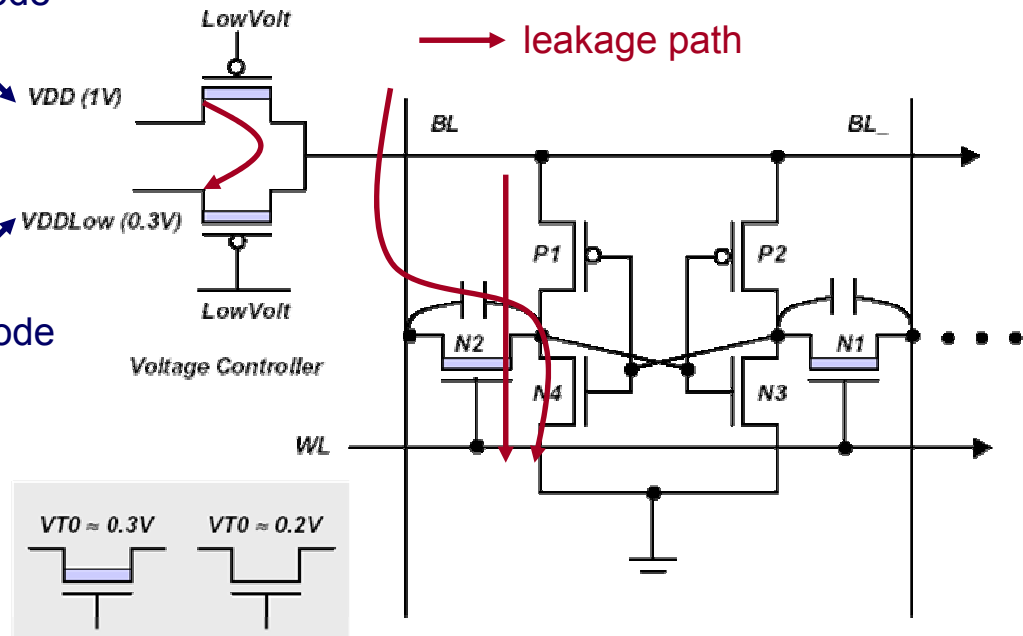


Drowsy Memory Using DVS

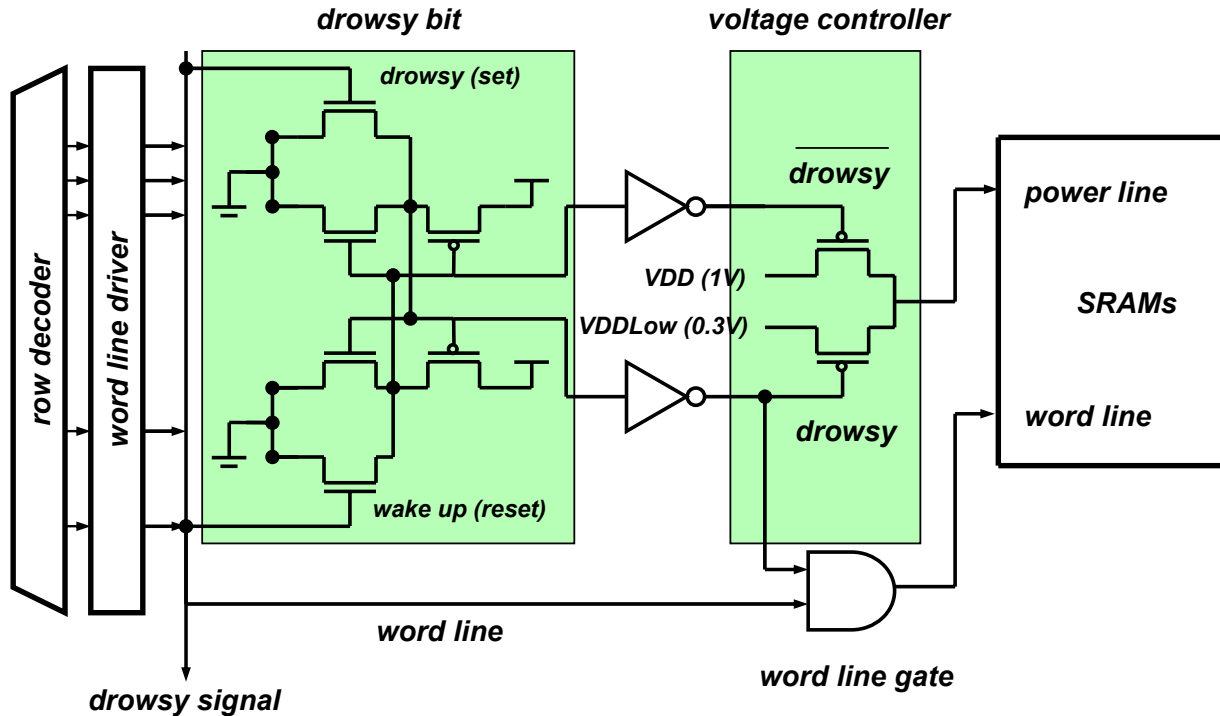
- Low supply voltage for inactive memory cells
 - Low voltage reduces leakage current too!
 - Quadratic reduction in leakage power ∴ $P_{\downarrow\downarrow} = I_{\downarrow} \times V_{\downarrow}$

supply voltage for normal mode

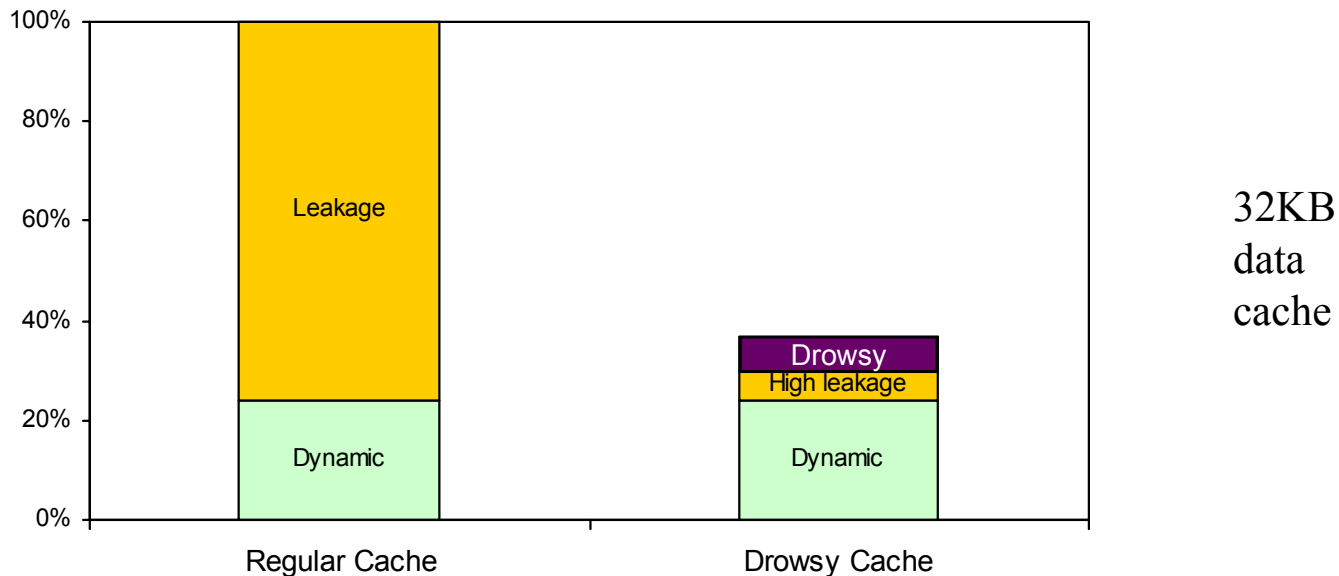
supply voltage for drowsy mode



Drowsy Cache Line Architecture



Energy Reduction



- High leakage: lines have to be powered up when accessed
- Drowsy circuit
 - Without high v_t device (in SRAM): 6x leakage reduction, no access delay.
 - With high v_t device: 10x leakage reduction, 6% access time increase.



Power and Design Uncertainty

■ Increasing uncertainty with process scaling

- Inter- and intra-die process variations
- Temperature variation
- Power supply drop
- Capacitive and inductive noise

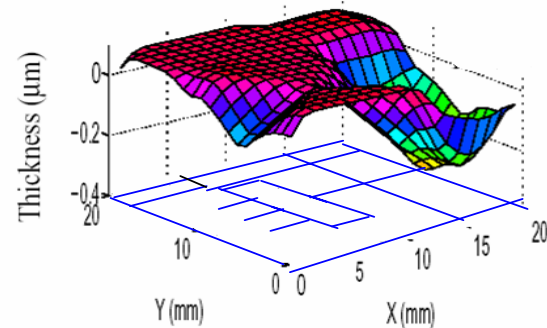
■ Impact on traditional design:

- Addressing worst-case variation in design requires *large safety margins*
- Higher energy / lower performance
- Reduced yield
- Difficulty in design closure

■ Key Observation: worst-case conditions also highly improbable

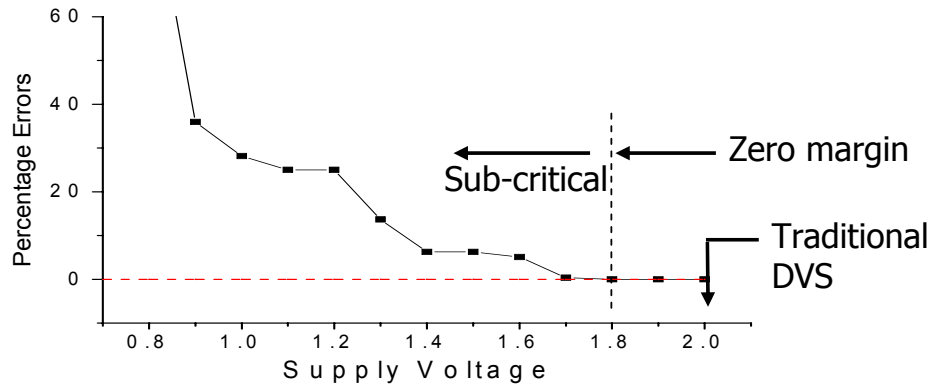
- Significant gain for circuits optimized for common case
- Efficiency mechanisms needed to tolerate infrequent worst-case scenarios

Intra-die variations in ILD thickness



Reducing Voltage Margins with Razor

- **Goal:** reduce voltage margins with *in-situ* error detection and correction for delay failures



- **Proposed Approach:**

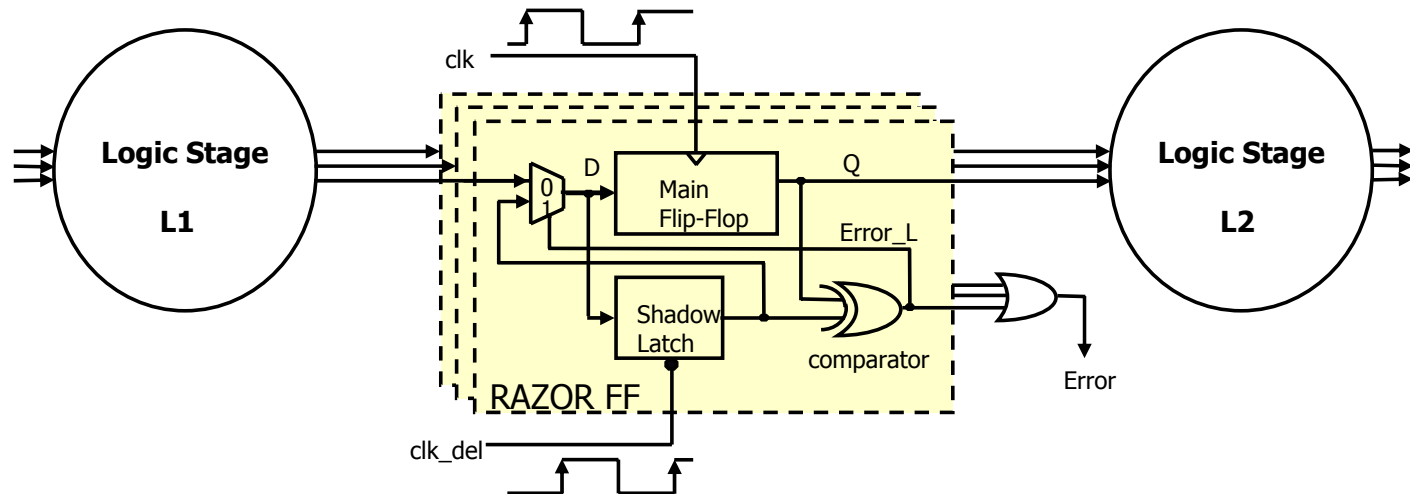
- Tune processor voltage based on error rate
- Eliminate safety margins, purposely run *below* critical voltage
 - Data-dependent latency margins
 - Trade-off: voltage power savings vs. overhead of correction

- Analogous to wireless power modulation & power/reliability trade-offs in DSP



Razor Flip-Flop Implementation

- Compare latched data with *shadow-latch* on delayed clock



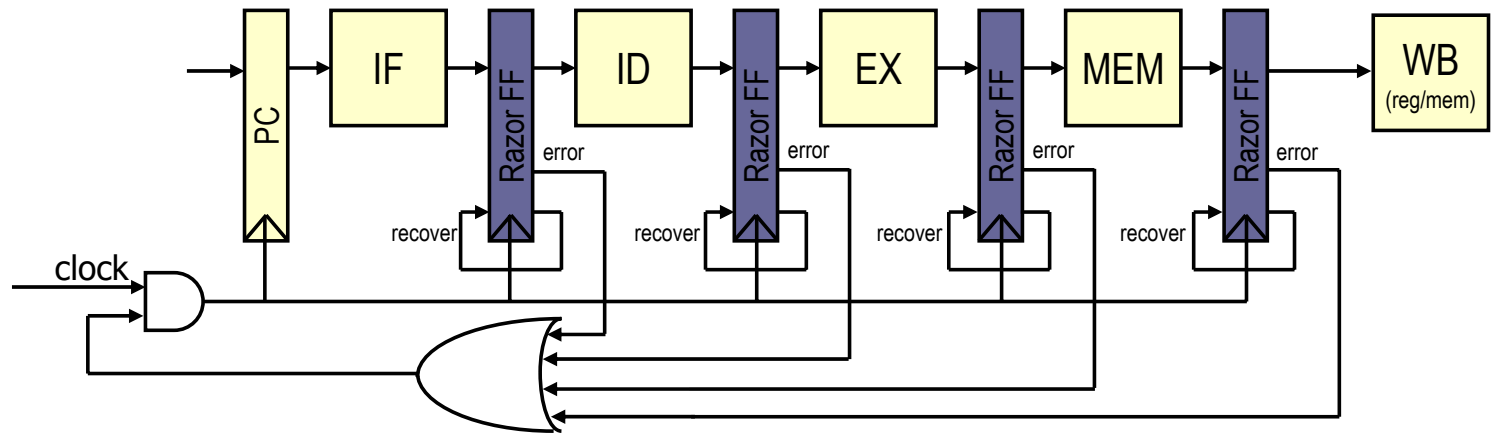
- Upon failure: place data from shadow-latch in main latch
 - Ensure shadow latch always correct using conservative design techniques
- Key design issues:
 - Maintaining pipeline forward progress
 - Recovering pipeline state after errors
 - Short path impact on shadow-latch
 - Meta-stable results in main flip-flop
 - Power overhead of error detection and correction



Centralized Pipeline Recovery Control

Cycle: 5

inst6

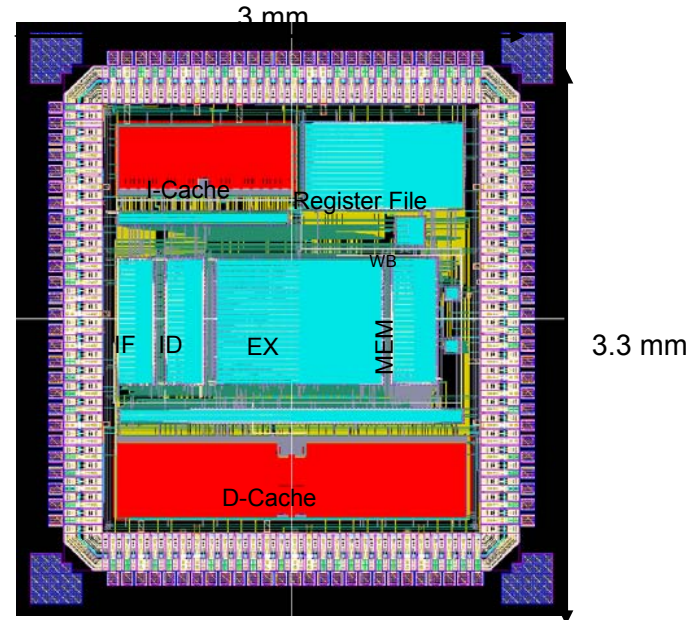


- Once cycle penalty for timing failure
- Global synchronization may be difficult for fast, complex designs



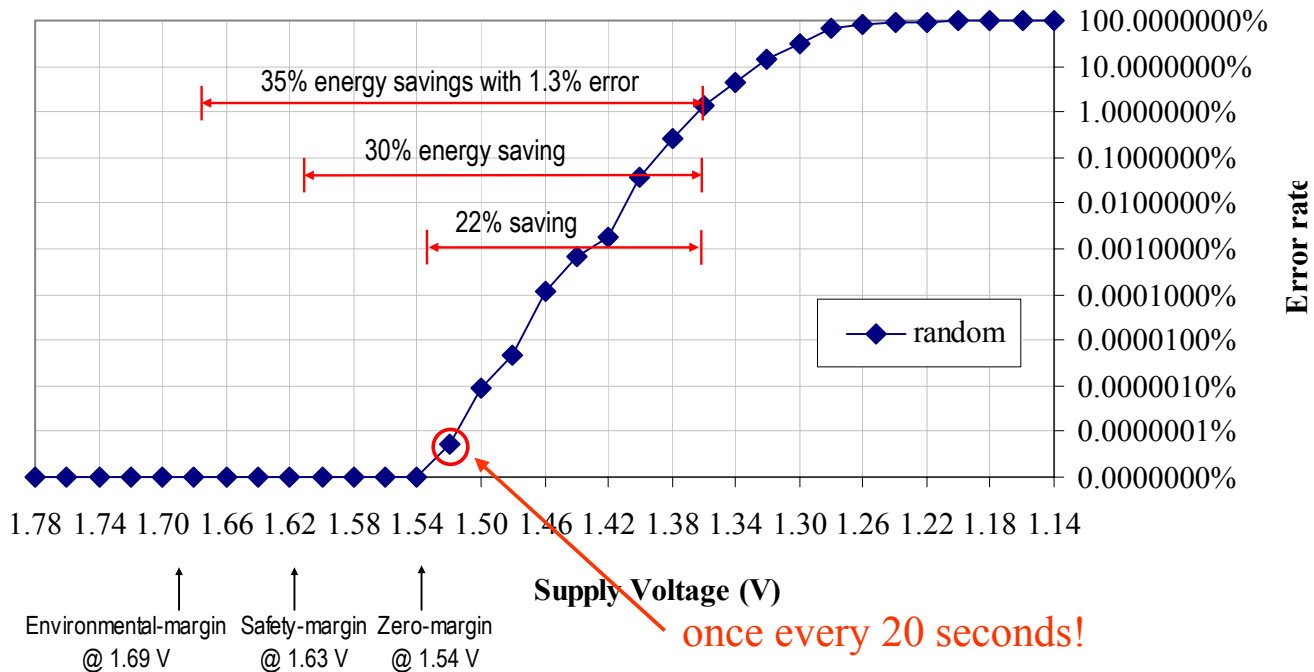
Razor Prototype Implementation

- 4 stage 64-bit Alpha pipeline
 - 200MHz expected operation in 0.18 μ m technology, 1.8V, ~500mW
- Razor overhead:
 - Total of 192 Razor flip-flops out of 2408 total (9%)
 - Error-free power overhead:
 - Razor flip-flops: < 1%
 - Short path buffer: 2.1%
 - Recovery power overhead:
 - Razor latch power overhead: 2% at 10% error rate
 - Additional power overhead due to re-execution of instructions



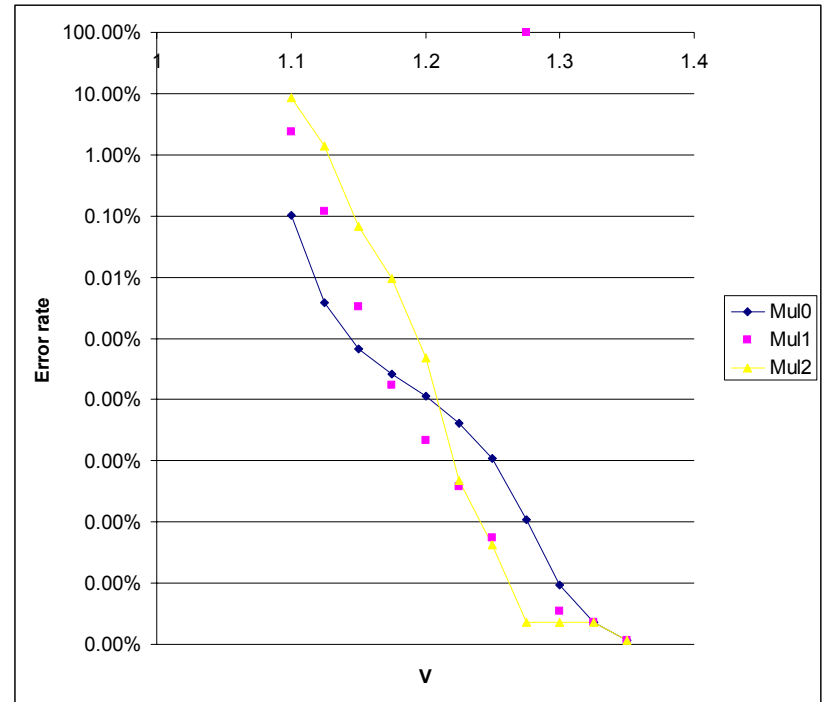
Error Rate Studies – Empirical Results

18x18-bit Multiplier Block at 90 MHz and 27 C



Multiplier Bit-flip Analysis

- Error rates similar to initial experiment (right)
- Occasional multi-bit flips require more complex error correcting schemes
- Razor overhead much lower than other schemes for full fault coverage



Summary for Razor

■ Razor benefits:

- Efficient in-situ timing error detection and correction for worst-case timing failures
- Eliminate process, environmental, and safety margins necessary in DVS
- Data dependent speculation for sub-critical voltage operation
- Allow design for common case – “better than worst case design”

■ Other applications:

- Over-clocking for performance improvement (2x shown among hobbyists)
- Clock skew tuning to off-set process / ambient variations
- Automatic adjustment to process variation



Conclusions

- Reducing power is the #1 issue facing designers of digital systems
 - true even if they are not mobile
- Dynamic power will continue to be a challenge
- Static power will to although there are partial technological solutions on the horizon
 - high-k dielectrics – oxide leakage
 - finFETs – subthreshold leakage
 - Feature size reductions – dynamic
- Trends that may help reduce power will also introduce uncertainty in the process of manufacturing chips – *the next major challenge*

