CMOS Process Variations: A Critical Operation Point Hypothesis

Janak H. Patel
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
jhpatel@uiuc.edu

Computer Systems Colloquium (EE380)
Stanford University
April 2, 2008

Outline

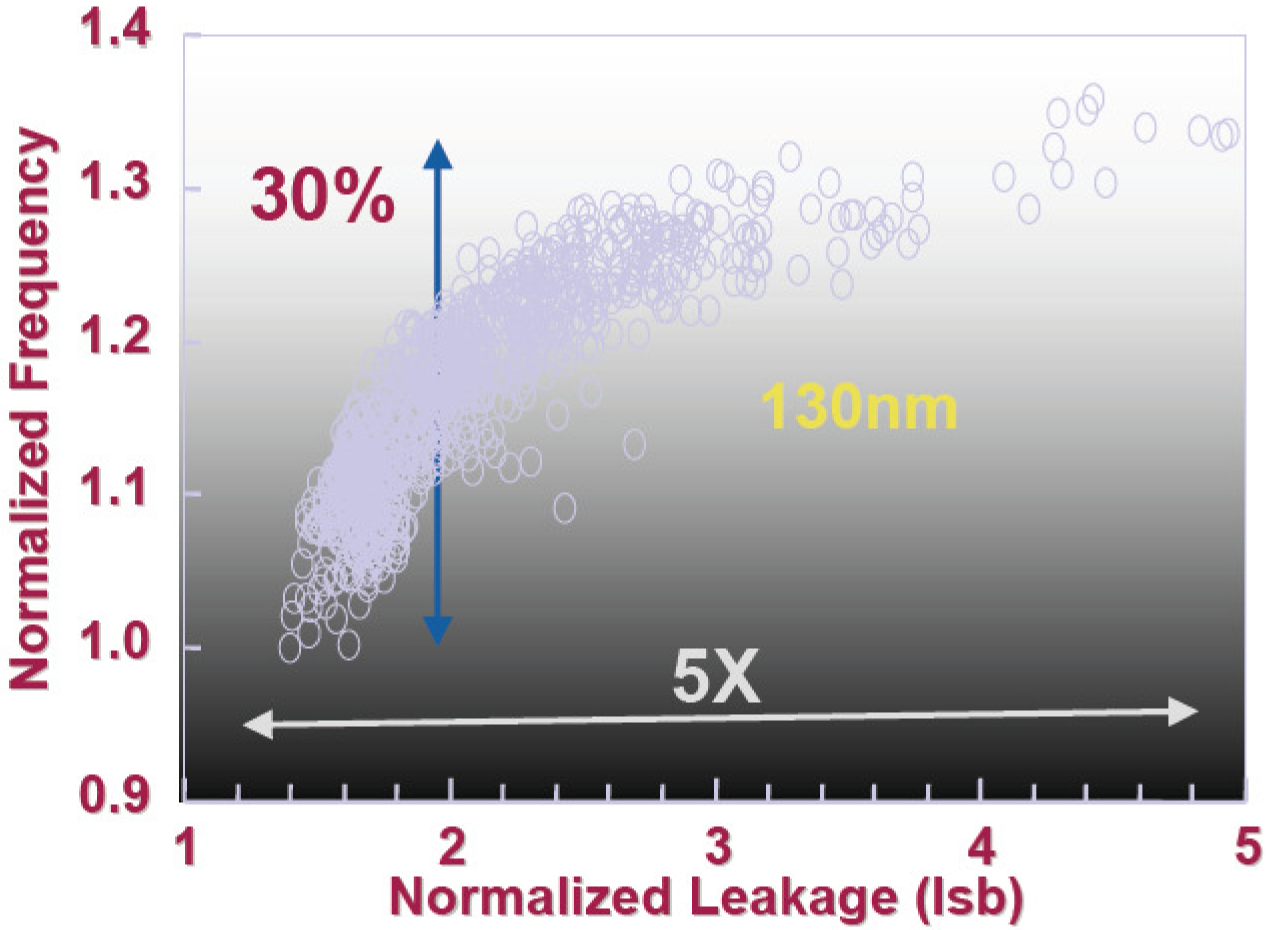
- **CMOS Process variations**
  - Current status
  - Future projections
- **A new Hypothesis on Critical Operation Point**
  - A *Thought Experiment* giving rise to the hypothesis
  - Two *Real Experiments* in support of the hypothesis
- **Potential exploits of the new hypothesis**
  - Power savings in large data-centers

Process Variations

- **Sources of Variations**
  - Gate Oxide thickness (T_{OX})
  - Random Doping Fluctuations (RDF)
  - Device geometry, Lithography in nanometer region
  - Transistor Threshold Voltage (V_T)
    - Sub threshold current, leakage, power, frequency
- **Range of Variations**
  - 100% V_T variation across a modern chip
  - 30% speed variation across a wafer
  - 100% leakage (static power) variation in a wafer

Static Variations today

(source: Shekhar Borkar, Intel)



FMAX statistical analysis

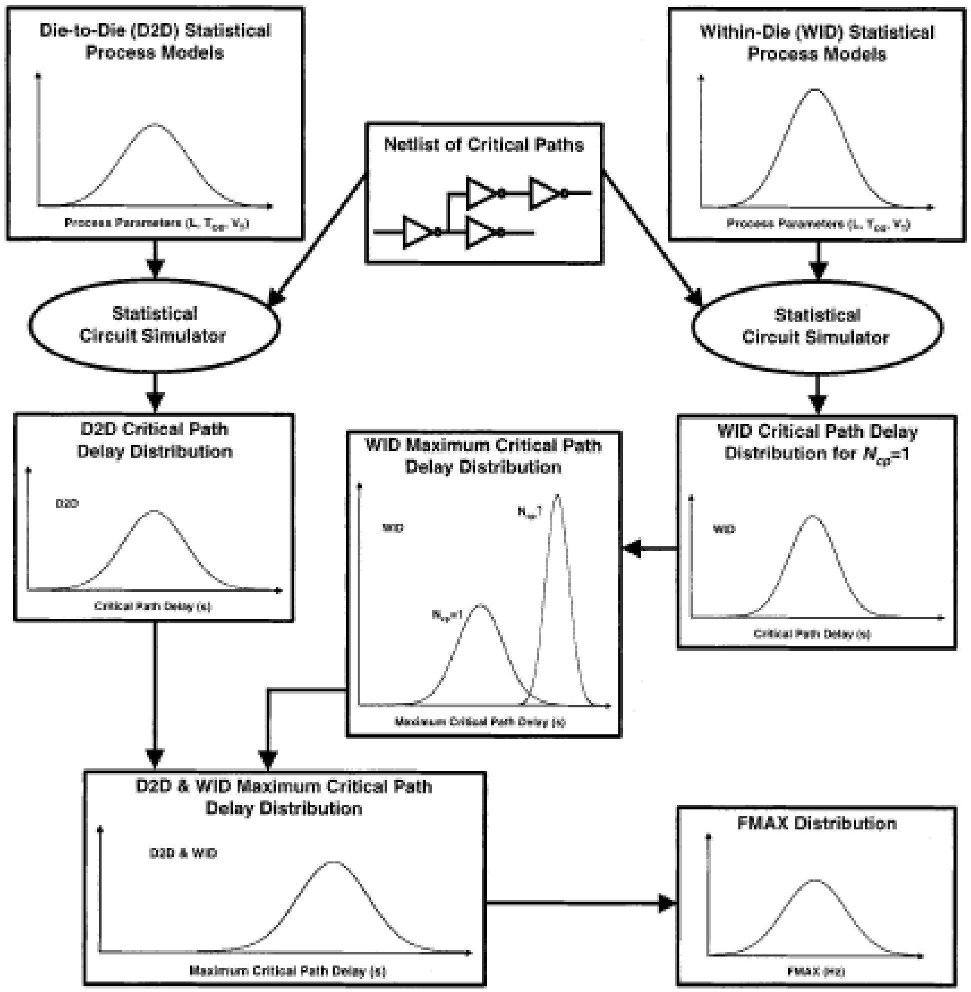
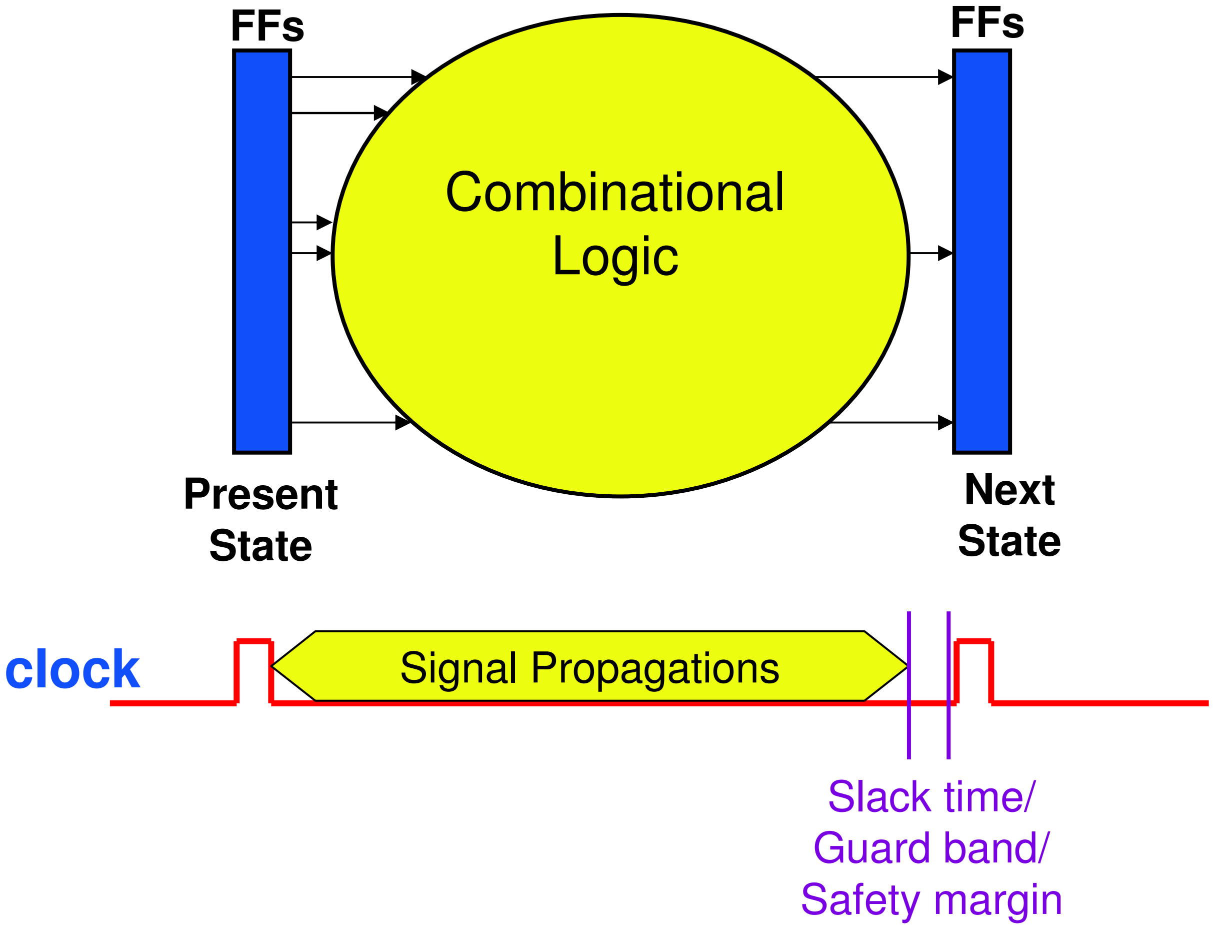

Die-to-Die (D2D) Statistical Process Models
Process Parameters (L, T_OX, V_t)
Netlist of Critical Paths
Within-Die (WID) Statistical Process Models
Process Parameters (L, T_OX, V_t)
Statistical Circuit Simulator
Statistical Circuit Simulator
D2D Critical Path Delay Distribution
D2D
Critical Path Delay (s)
WID Maximum Critical Path Delay Distribution
WID
N_cp↑
N_cp=1
Maximum Critical Path Delay (s)
WID Critical Path Delay Distribution for N_cp=1
WID
Critical Path Delay (s)
D2D & WID Maximum Critical Path Delay Distribution
D2D & WID
Maximum Critical Path Delay (s)
FMAX Distribution
FMAX (Hz)

Fig. 1. Flowchart for describing the FMAX distribution. N_{cp} is the number of independent critical paths on a chip.

Source: Bowman, K.A.; Duvall, S.G.; Meindl, J.D., "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *Solid-State Circuits, IEEE Journal of* , vol.37, no.2, pp.183-190, Feb 2002

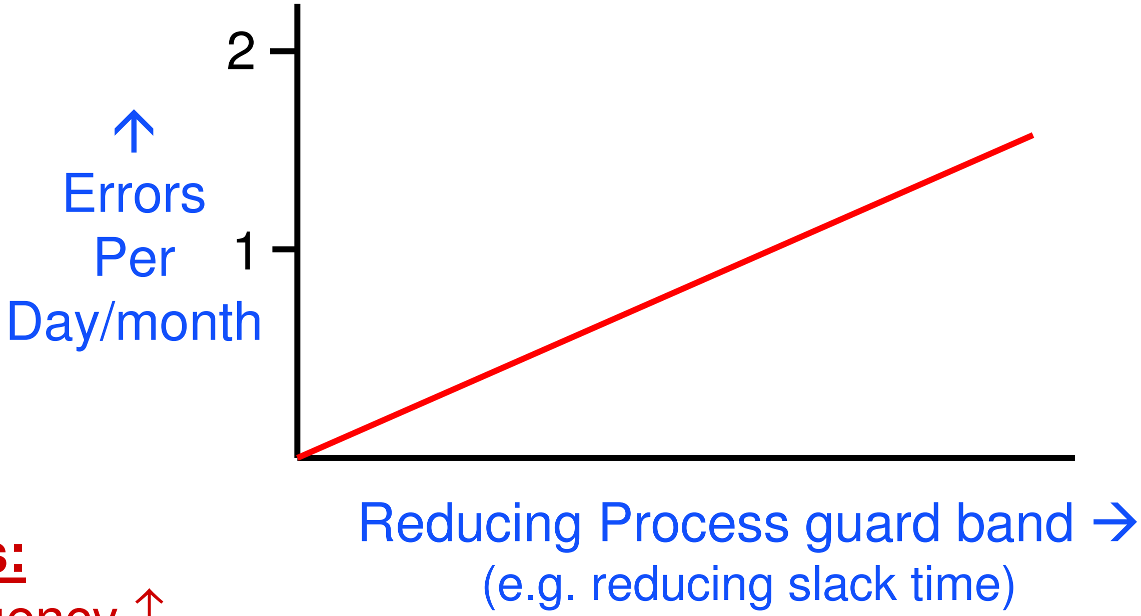
Process Variations and Slack Time



Slack Time Reduction

Process Variations ↑
Clock Frequency ↑
Supply Voltage ↓
Ambient Temperature ↑
Gate and Pin Switching rate ↑
Years of Aging ↑

Errors and Process Variations



Parameters:

Clock Frequency ↑
Supply Voltage ↓
Ambient Temperature ↑
Gate and Pin Switching rate ↑
Years of Aging ↑
Process Variations ↑

Errors:

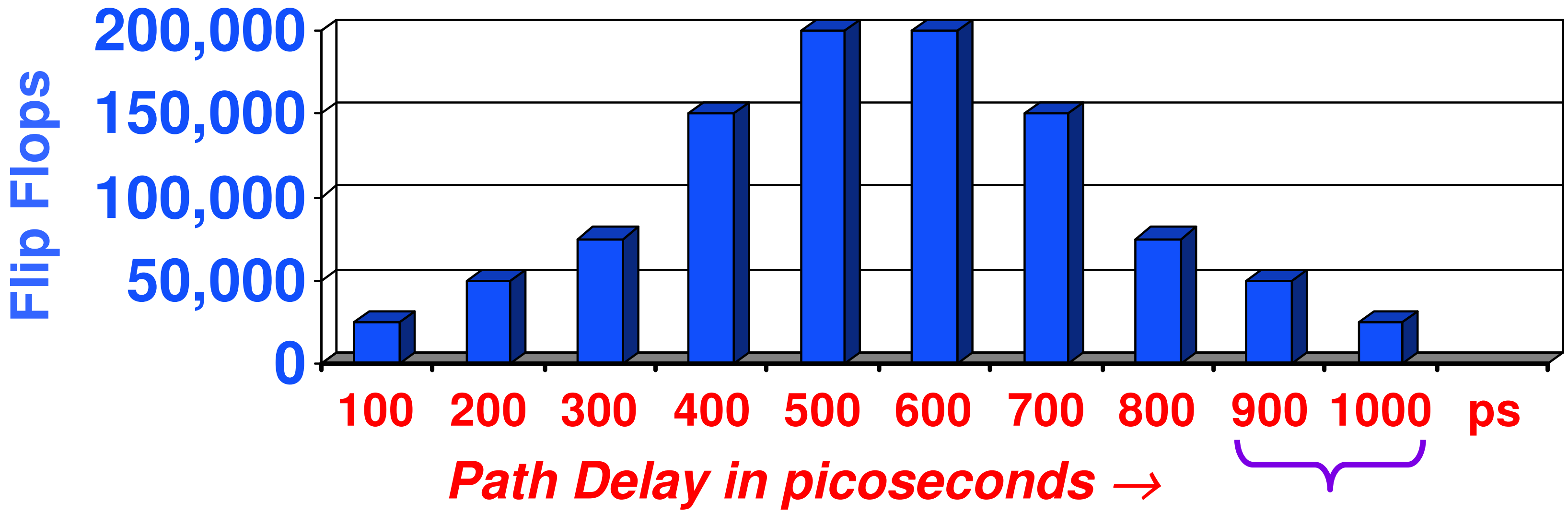
All are *timing errors*
No spontaneous bit flips

Protecting against process variations

- **If the error rate from added delays remains relatively small, we can utilize some of the established techniques**
  - iROC, Razor, Biser etc.
  - Error coding – Parity codes, Arithmetic codes, Residue codes, Parity prediction, Algorithm-based fault-tolerance, TMR etc.
  - Time redundancy like RESO
- **What if the error rate is massive?**
- **Are massive errors possible in a good chip?**

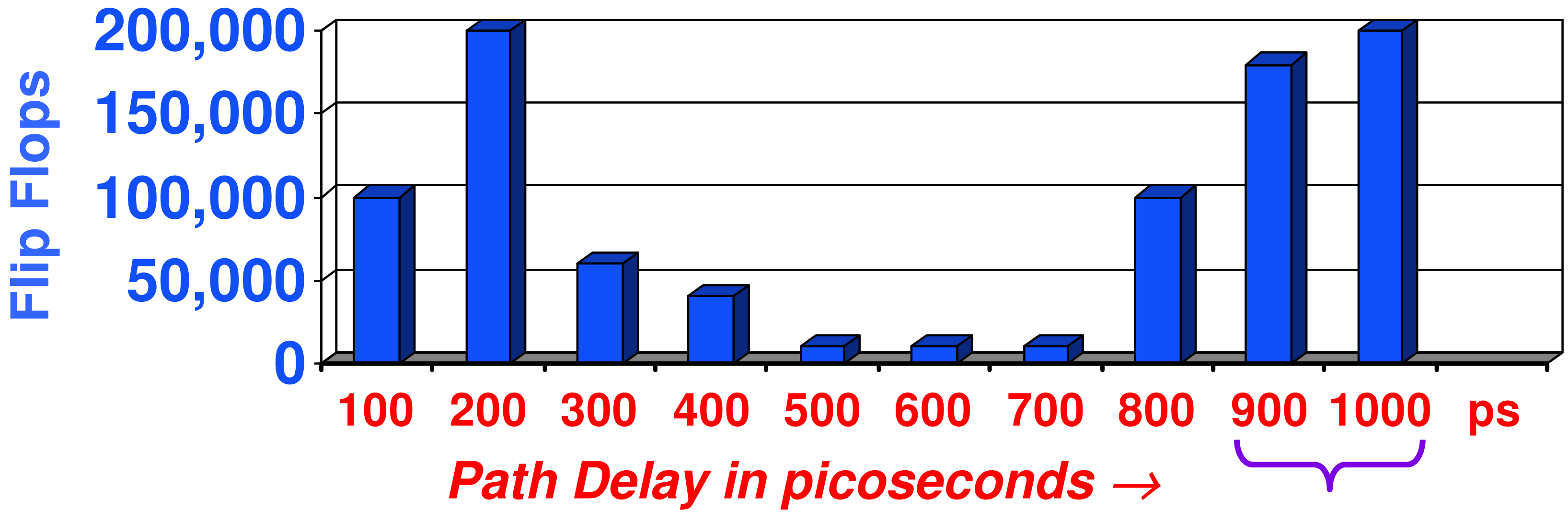
How many flip-flops on critical-paths?

- Consider a 1-Ghz chip with a million flip-flops
- Let us divide the 1ns Clock period in to 1000 bins
- Put a FF in bin *p* if the longest path at its input has a delay of *p* picoseconds
- How many FFs are in bins 900ps to 950ps?

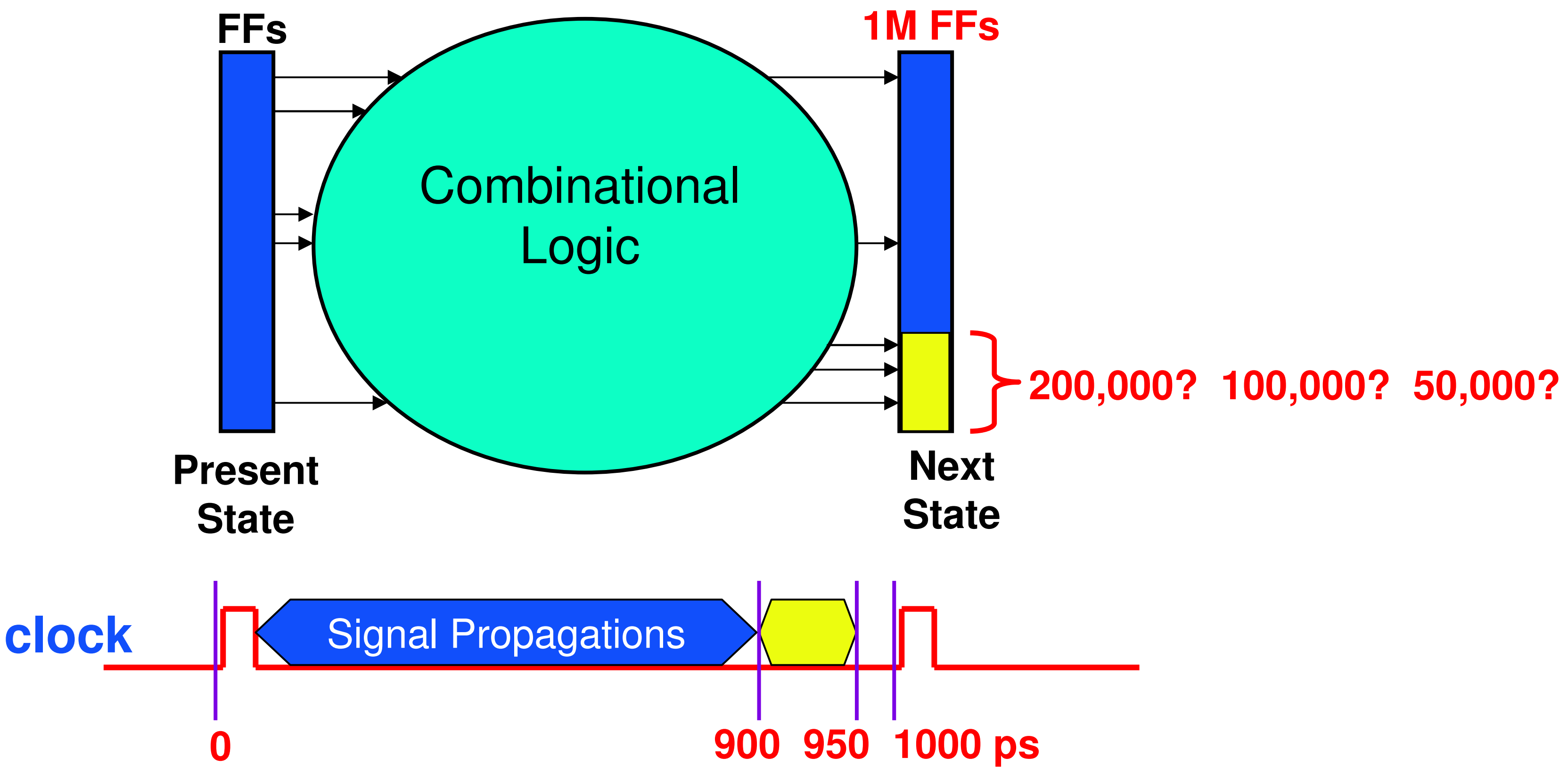


How many flip-flops on critical-paths?

- Consider a 1-Ghz chip with a million flip-flops
- Let us divide the 1ns Clock period in to 1000 bins
- Put a FF in bin *p* if the longest path at its input has a delay of *p* picoseconds
- How many FFs are in bins 900ps to 950ps?



How many flip-flops on critical-paths?



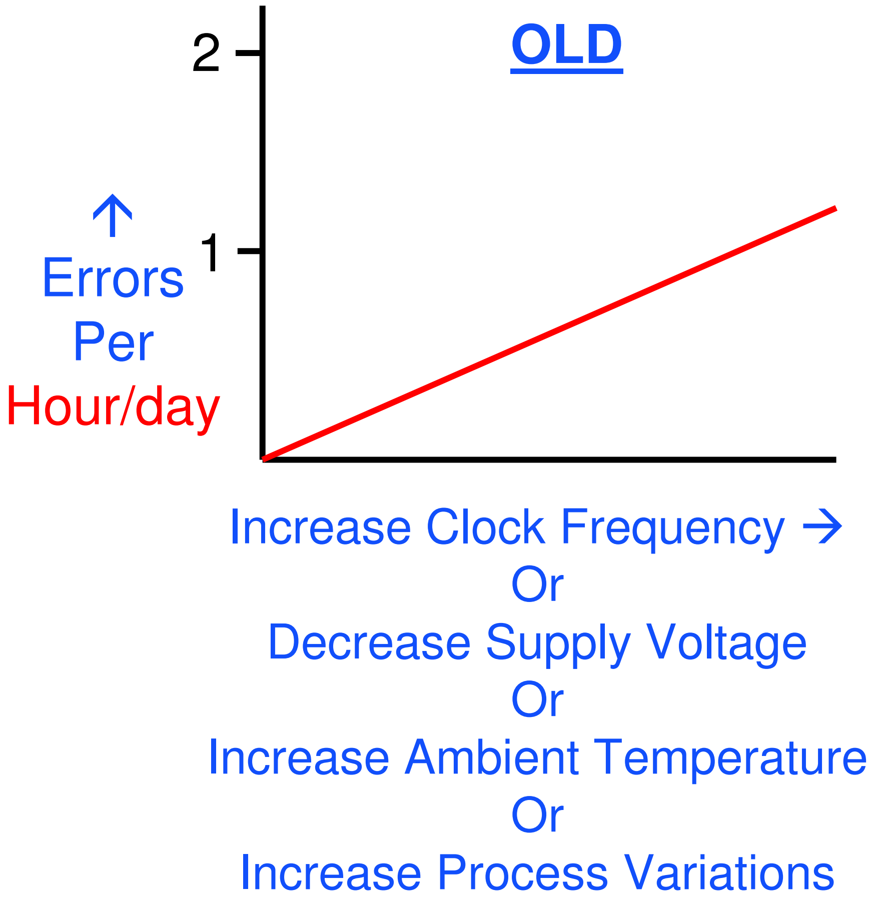
A Thought Experiment

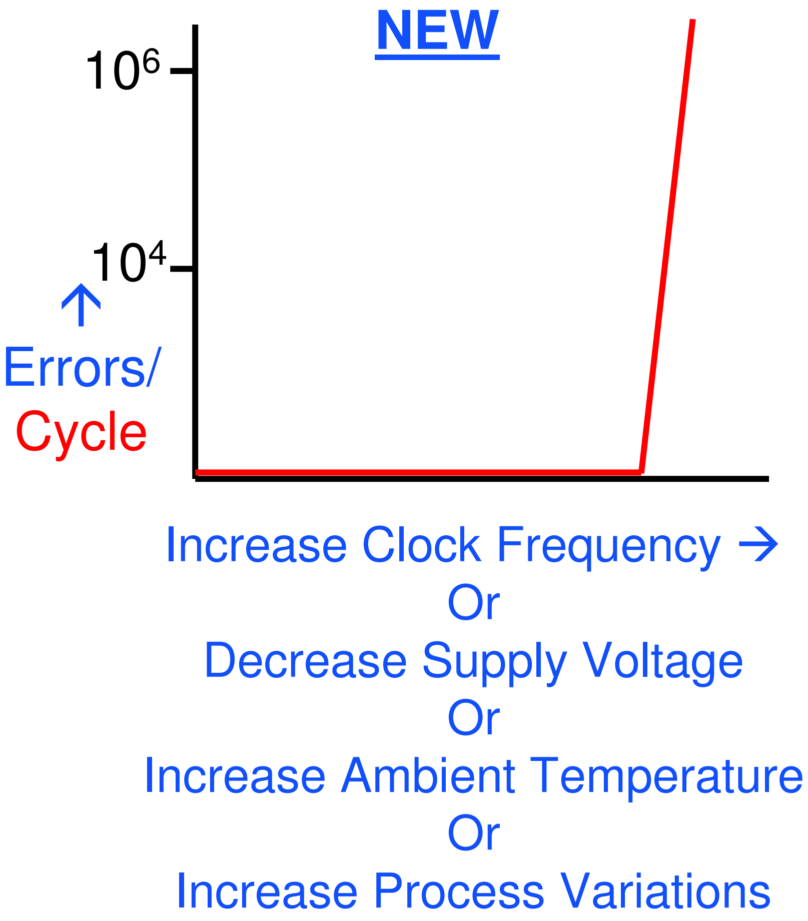
- **Let us conservatively assume 100,000 ffs are on critical paths (10% of total)**
- **Consider any of the following factors that reduce the slack time of these ffs.**
  - Increase clock frequency (reduce cycle time)
  - Decrease supply voltage (increases gate delays)
  - Add years of aging (gates get slower with age)
  - Increase process variations (larger sigma)
- **Assume just 10% of critical ffs get its inputs late this cycle**
  - This implies 10,000 flip-flops produce errors in a single clock cycle!
- **Massive number of errors result in a few clock cycles**

Do your own Thought Experiment!

- **Total Number of Flip-Flops: 400,000**
  - Only 5% of these are on critical paths: 20,000 FFs
  - Only 1% of these receive critical signals: 200 FFs
  - In 10 consecutive clock cycles: 2000 errors!
- **Do your own Thought Experiment**
  - Estimate number of FFs on critical paths from timing analysis or synthesis report. Guesstimate, % of active signals.
  - How many errors in 10, 100 or 1000 consecutive clock cycles?
  - Is there any scenario that doesn’t lead to a catastrophic failure in an extremely short time?

A new hypothesis



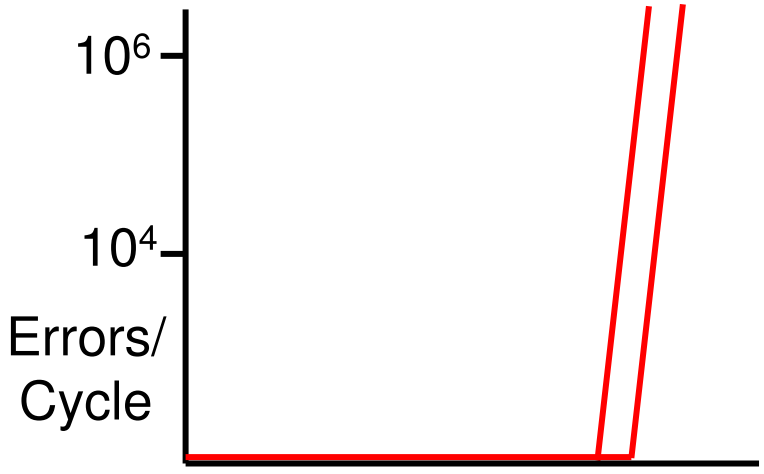


Hypothesis of Critical Operation Point

- **In large CMOS circuits there exists a *Critical Operating Frequency* F_C and *Critical Voltage* V_C for a fixed ambient temperature T, such that**
  - Any frequency above F_C causes massive errors
  - Any voltage below V_C causes massive errors
  - Any frequency below F_C or voltage above V_C , no process related errors occur
- **In practice, F_C and V_C are not single points, but are confined to an extremely narrow range for a given ambient temperature T_C**

F_C and V_C : Points or a Range?

- **During a systematic search for the critical point, one will find a point when the system crashes**
- **Critical point varies in a very narrow range from one experimental search to another, most likely due to temperature variations**
- **Practically it is impossible to control the junction temperature of each transistor to a precise number T_c**

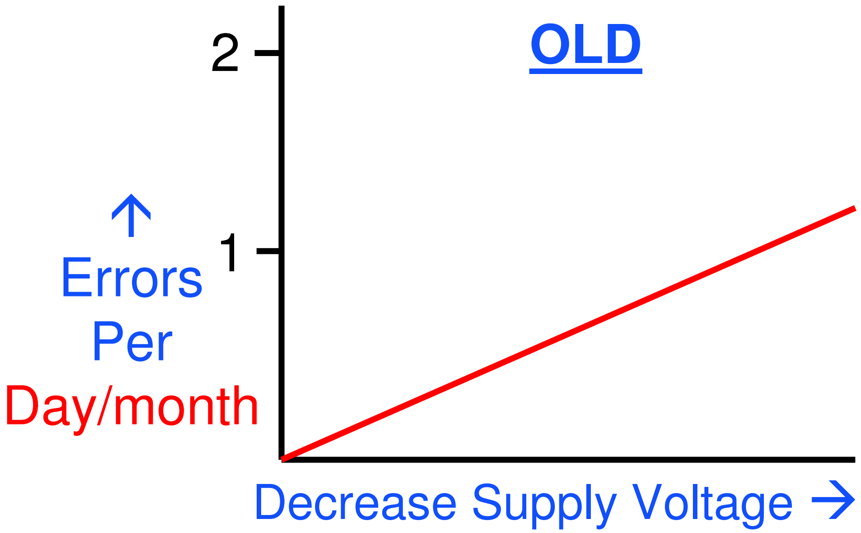


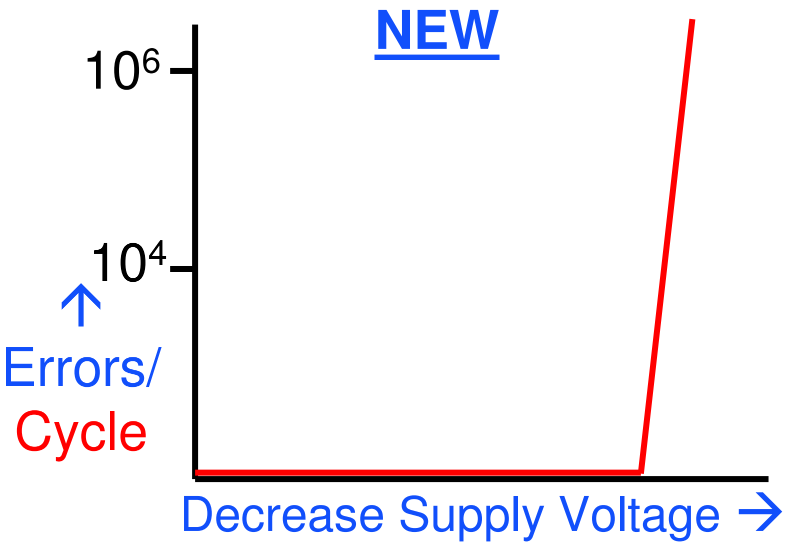
Outcome of two distinct Experiments on the same chip

Experiments to disprove the hypothesis

- **Subject a large chip to slowly increasing frequency or slowly decreasing supply voltage**
  - At each step, exercise the chip extensively and monitor continuously for any errors
- **Two microprocessors were set up for detecting errors in the presence of reduced supply voltage**
  - PowerPC 750, 2.5V, 233MHz
    - C-program to exercise and monitor for errors
  - Pentium-M, 1.308V, 2GHz
    - Third-party program to keep the cpu 100% busy and report errors (more like a power virus!)

Experiment to find which of these two?





Experimental Set-UP

- **A Single-Board-Computer with PowerPC 750**
  - 233MHz, 2.5V Power Supply
  - A Hewlett-Packard E3631A Power Supply
    - Digital control in units of 10 miliVolts steps
  - A Blow-Drier to raise the ambient temperature
- **A Program written to stress all major functional blocks**
  - Tried to maximize execution rate (load)
  - Tried to maximize logic switching rate
  - Every operation was checked against known good values and instantly reported for any error

“Stressing” PowerPC 750 (233 MHz)

Routine	Operations per loop	Number of loops	Total Operations	Approx. Running Time	**Approx. Operations Per Second**
Register Unit	40	8,000,000	320,000,000	6.34 s	**50.47x10^{6}**
Instruction Fetch Unit	32	8,000,000	256,000,000	92.04 s	**2.78x10^{6}**
Integer Addition	40	8,000,000	320,000,000	9.35 s	**34.22x10^{6}**
Integer Subtraction	40	8,000,000	320,000,000	9.12 s	**35.09x10^{6}**
Integer Multiplication	58	8,000,000	464,000,000	18.21 s	**25.48x10^{6}**
Integer Division	50	8,000,000	400,000,000	33.72 s	**11.86x10^{6}**
Logical AND	20	8,000,000	160,000,000	0.71 s	**225.35x10^{6}**
Logical OR	20	8,000,000	160,000,000	0.64 s	**250.00x10^{6}**
Logical XOR	20	8,000,000	160,000,000	0.71 s	**225.35x10^{6}**
Integer Unit 2	40 adds & multiplies	8,000,000	640,000,000	48.75 s	**13.13x10^{6}**
Floating Point Add	20	8,000,000	160,000,000	0.82 s	**195.12x10^{6}**
Floating Point Subtract	20	8,000,000	160,000,000	0.82 s	**195.12x10^{6}**
Floating Point Multiply	20	8,000,000	160,000,000	0.83 s	**192.77x10^{6}**
Floating Point Divide	20	8,000,000	160,000,000	0.82 s	**195.12x10^{6}**
Branch Processing Unit	7	8,000,000	56,000,000	6.09 s	**9.20x10^{6}**
Load/Store Unit	320 loads, 192 stores	80,000	40,960,000	13.24 s	**3.09x10^{6}**
Data Cache	2	3,300,000	6,600,000	15.97 s	**0.41x10^{6}**

Results of Lowering Supply Voltage Power PC-750 µP

Chip No.	No. Tests	Critical Supply Voltage V_C	Observations	
			System Hangs	Program Crashed
1	45	1.99 V – 2.10 V	31	14
2	35	2.00 V – 2.08 V	26	9
3	25	2.10 V – 2.29 V	18	7
4	25	2.08 V – 2.20 V	17	8

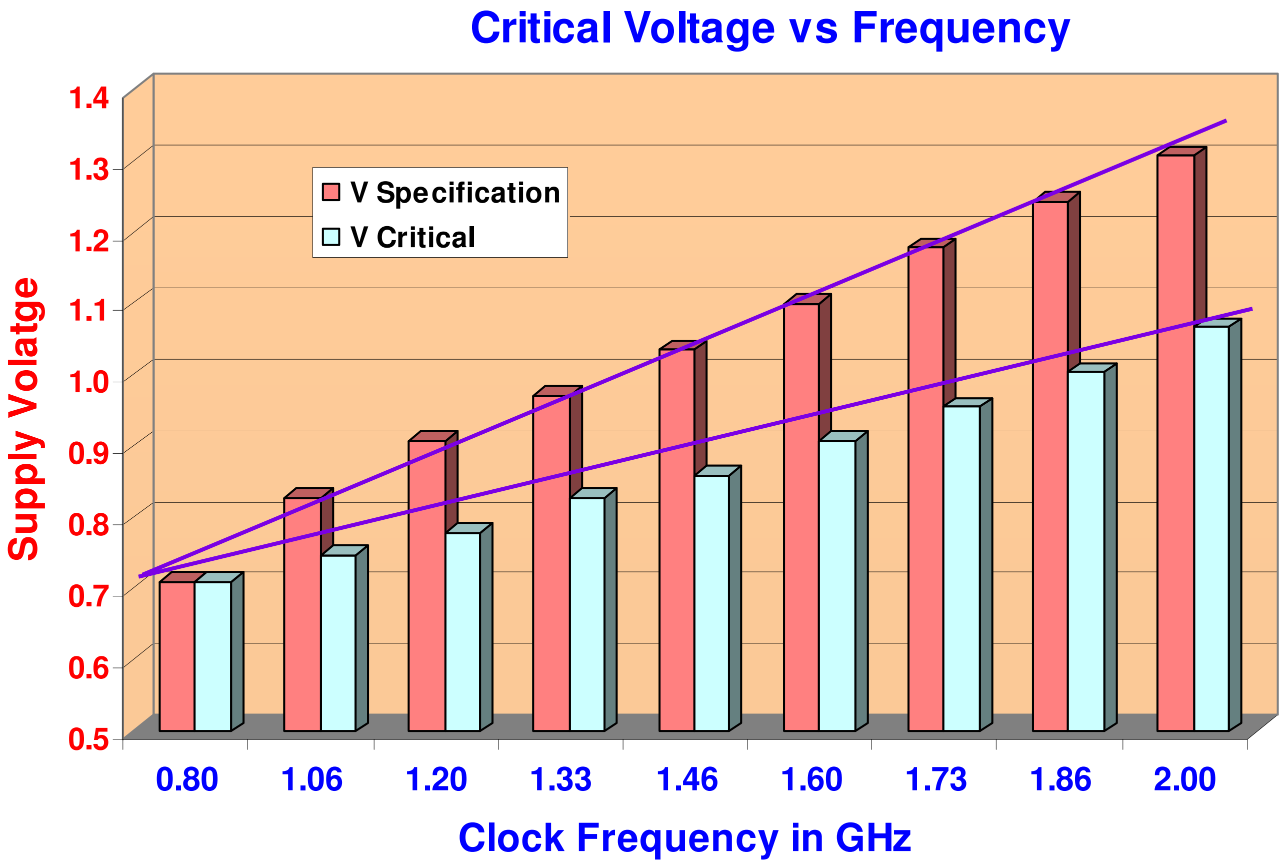
Nominal Supply Voltage of 2.5 V is reduced in steps of 1/100th Volt with clock frequency constant at 233MHz

No Data Error was ever Observed at user visible Registers!

More recent Experiment

- **Processor: Pentium-M, speed step technology**
- **Rated at 2GHz at core voltage of 1.308V**
- **Experiment**
  - While keeping cpu 100% busy at 2GHz, reduced the voltage in steps of 16mV
  - Third party software claimed to report errors
- **Reduced voltage 15 steps down to 1.068 with no errors**
- **At the next step down to 1.052V, cpu crashed**
- **No errors observed – only crashes!**
- **Similar results at seven other frequencies**

Experiment on Pentium-M



Some Remarks on Experiment

- **Possible explanation for the observations**
  - A modern processor has a large number of flip-flops that are *not user visible*
    - e.g. Pre-fetch buffers, history tables, reservation stations, write buffers, and state controllers for everything from moving instructions and data to controlling a cache
  - Control Logic fails simultaneously with ALU datapath
  - Massive errors in control and data in a single cycle
  - Instruction flow is completely disrupted. Therefore no error could be reported. **Catastrophic failure!**

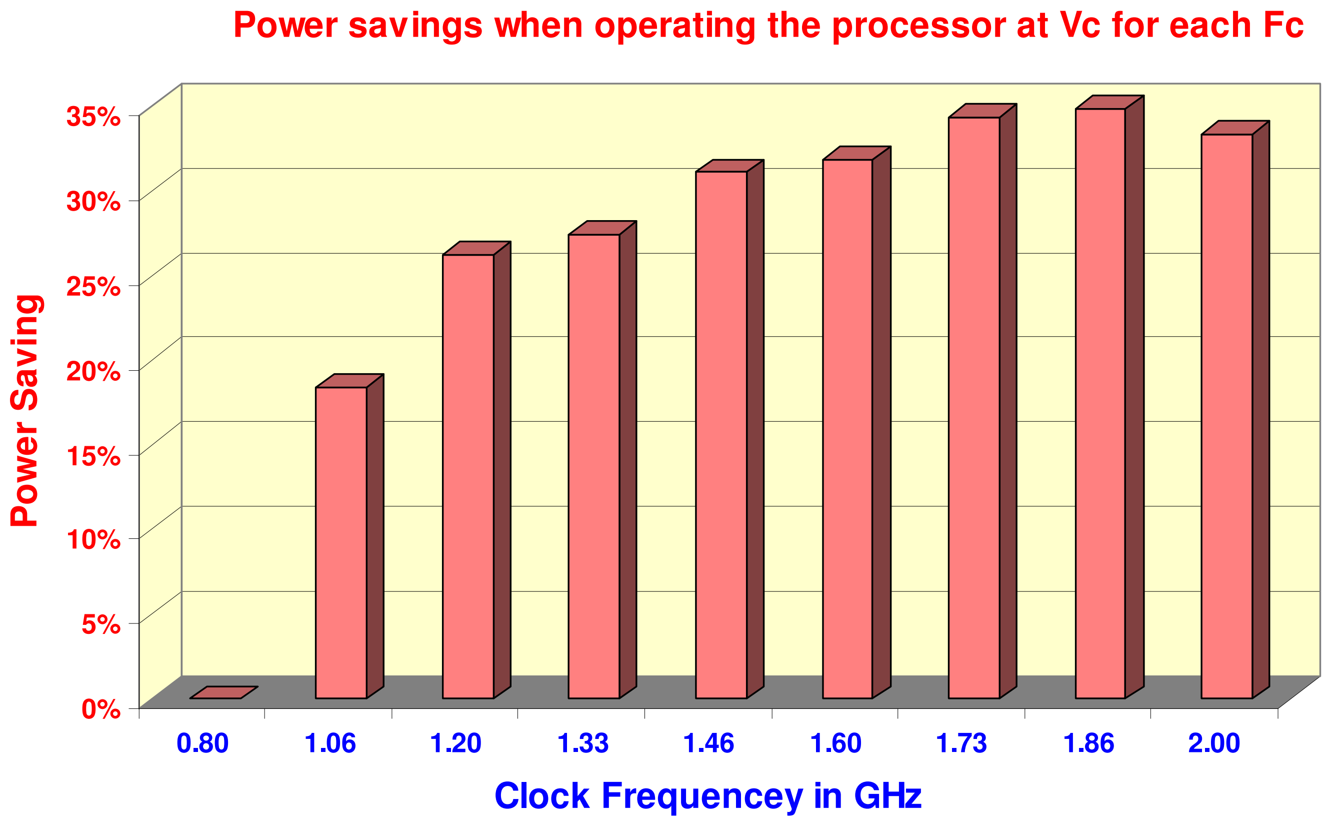
Personal Remarks

- **CMOS technology is robust now and will continue to be so for the foreseeable future**
  - Process Variation related errors if any, must be massive
  - No industry can survive with massive failures
  - Process variations must remain bounded within some reasonable limits
- **Moore's Law continues to hold!**
  - 45nm with (HiK+MG) has lower RDF and T_{OX} variations than 65nm [Kelin J. Kuhn, Reducing Variation in Advanced Logic Technologies: Approaches to Process and Design for Manufacturability of Nanoscale CMOS, IEDM 2007.]

Exploiting Process Variations

- **If the *"critical operation point hypothesis"* holds**
  - Above critical frequency **F_C** massive failure occurs, below this point error-free operation results
  - Below critical supply voltage **V_C** massive failure occurs, above it error-free operation results
- **In data-centers with 1000's of μPs, operating each μP with the lowest V_C for a given frequency can save lots of power**
- **As the number of cores approach 100 or more, it would be imperative to use different voltage-frequency pair (F_C, V_C) for each core on the same die**

Dynamic Power Savings in Pentium-M



Future Research

- **Need to verify the proposed hypothesis with more experiments or simulations**
- **Off-line Test**
  - To determine several critical frequency-voltage pairs ($\mathbf{F_C}$, $\mathbf{V_C}$) for each die and possibly each core on the die
- **On-line Test**
  - To establish new frequency-voltage pairs ($\mathbf{F_C}$, $\mathbf{V_C}$) in the field at the time of deployment
  - To monitor aging, since ($\mathbf{F_C}$, $\mathbf{V_C}$) may shift with age
- **Self-Test**
  - Self Calibrate periodically to arrive at current ($\mathbf{F_C}$, $\mathbf{V_C}$)

Questions? Comments?

OUT & BACK TRAIL
RUGGED 7.6 MILE
CLEAR LAKE TRAIL
RUGGED 3 MILE
FOOT RACE STARTS ON ROAD
HIKING TRAIL STARTS HERE
CLEAR POND TRAIL
WARNING! BIKERS USE BIKE TRAIL ONLY!
SIBERIAN EXPRESS
1150
866
ROAD RUNNERS
2001

SIBERIAN EXPRESS
867
KENDRICK ROAD RUNNERS