Fault-tolerant Circuit Design

Yiping Kang

Zhi Qu

Zhan Wang

Oct 24th, 2013

Outline

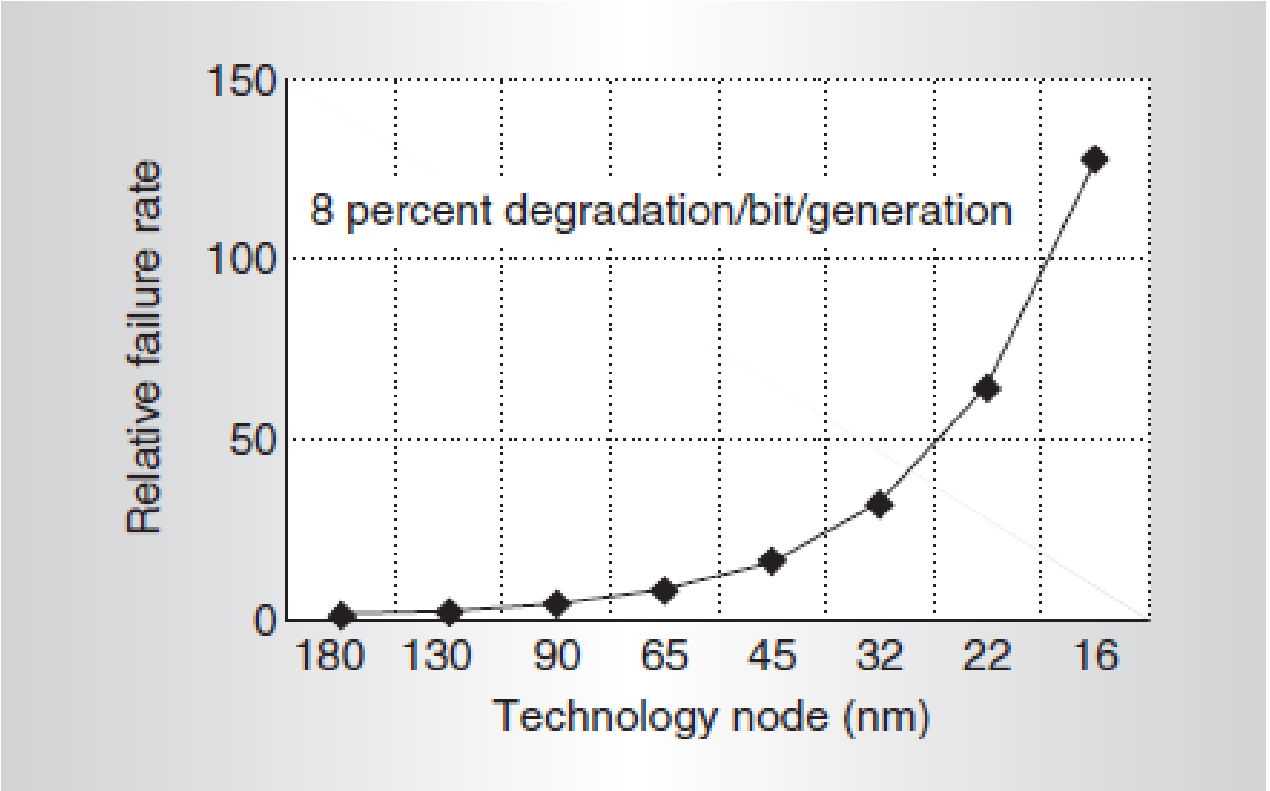
- Introduction
- Fault-tolerance Design
  - Module Redundancy
  - Error Correction Code (ECC)
  - Timing Error Detection (Razor)
- Conclusion

Introduction

- Circuits are becoming more unreliable as technology scale

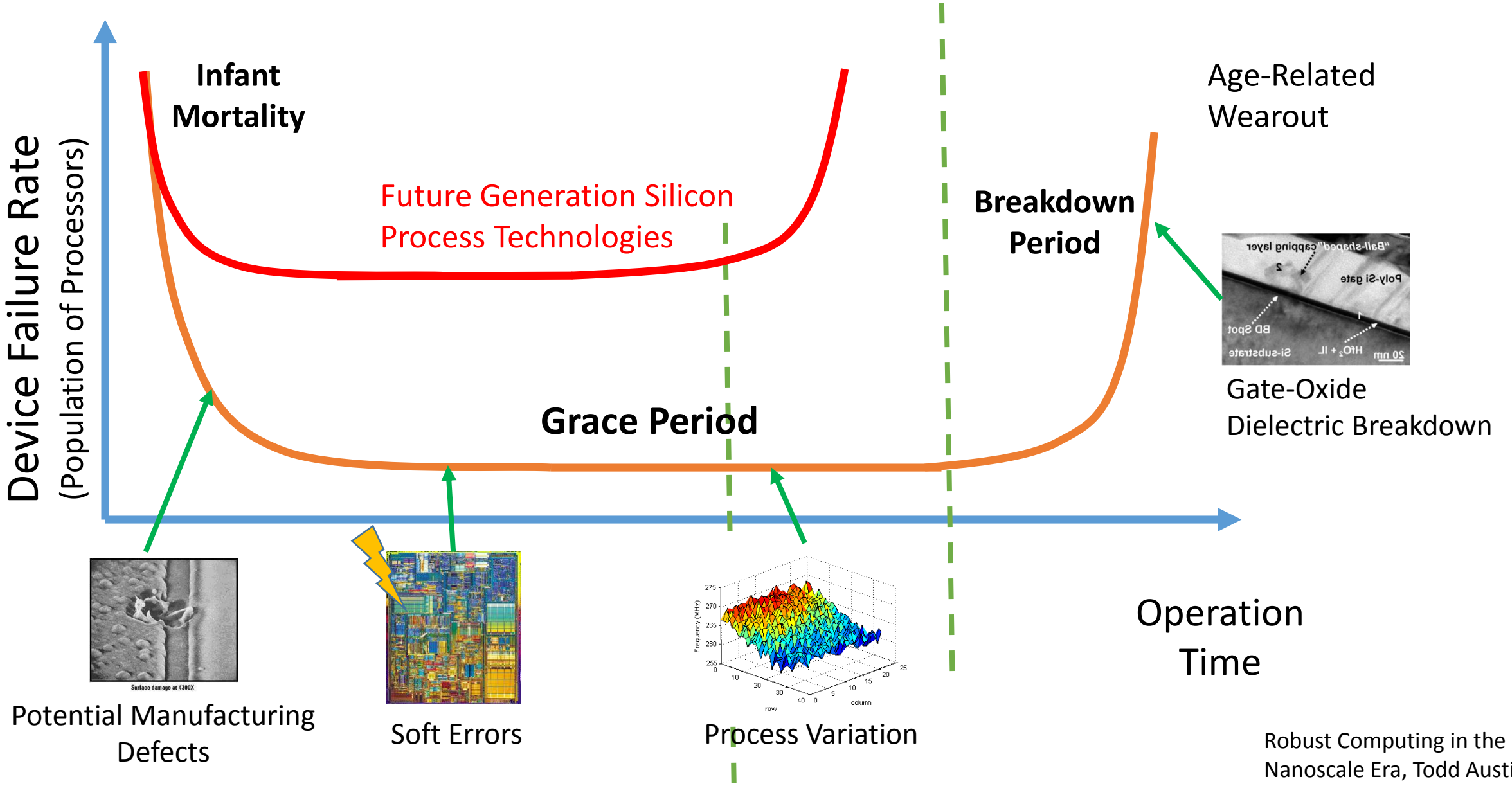
- Microprocessors are everywhere
  - Consumer Products
  - Data Centers
  - …

- Better not to break, huh…



Shekhar Borkar, Intel Fellow

Microprocessor Reliability



Robust Computing in the Nanoscale Era, Todd Austin

Error Classes

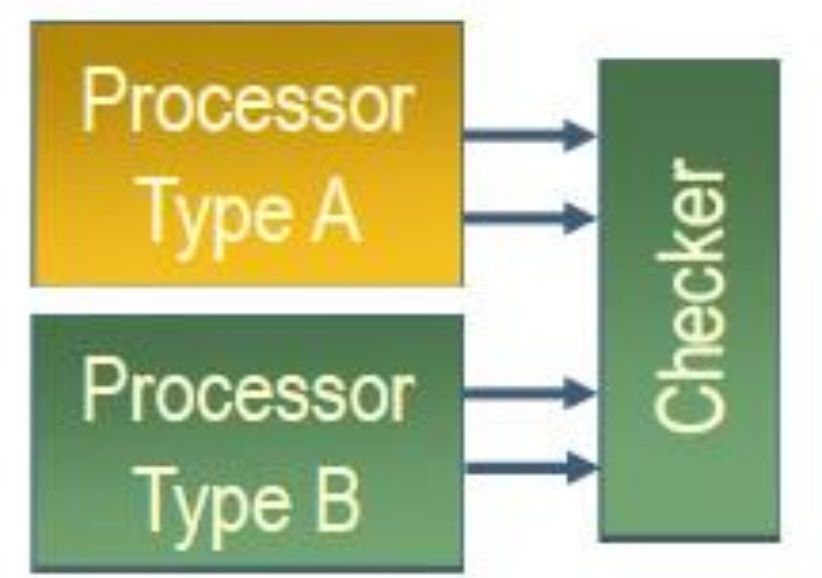
- Permanent Fault (hard fault)
  - Physical damage
  - Can NOT be reverted
  - E.g, latent manufacturing defects
- Transient Fault (soft error)
  - Single-event upsets (SEU)
  - Particle strikes
  - Interconnect noises

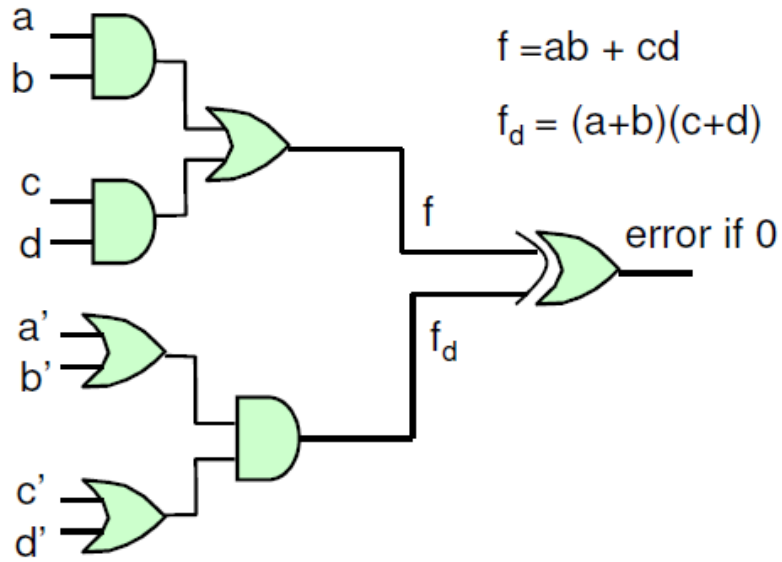
Fault-tolerance Design

- DMR, TMR
- Error Detection Code
- Timing Error Detection and Recovery

DMR Error Detection

- Context: **Dual-modular redundancy for computation**
- Problem: **Error detection across blades**





Triple Modular Redundancy (von Neumann)

- ***M*** are identical ***modules*** or ***black boxes***
- ***V*** is called a ***majority organ*** by Von Neumann. (a *voting* circuit)

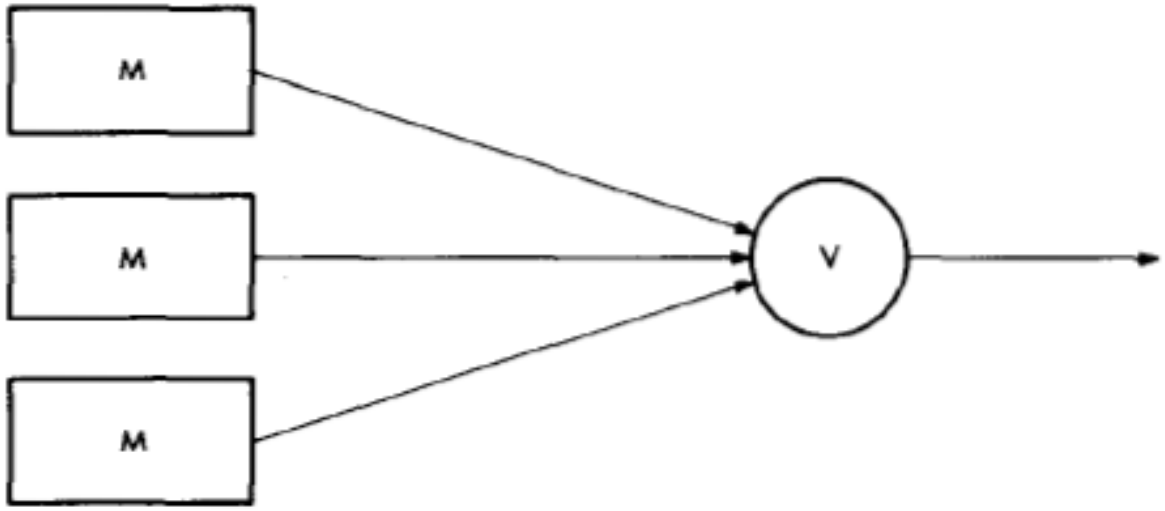


Figure 1 **Triple redundancy as originally envisaged by Von Neumann.**

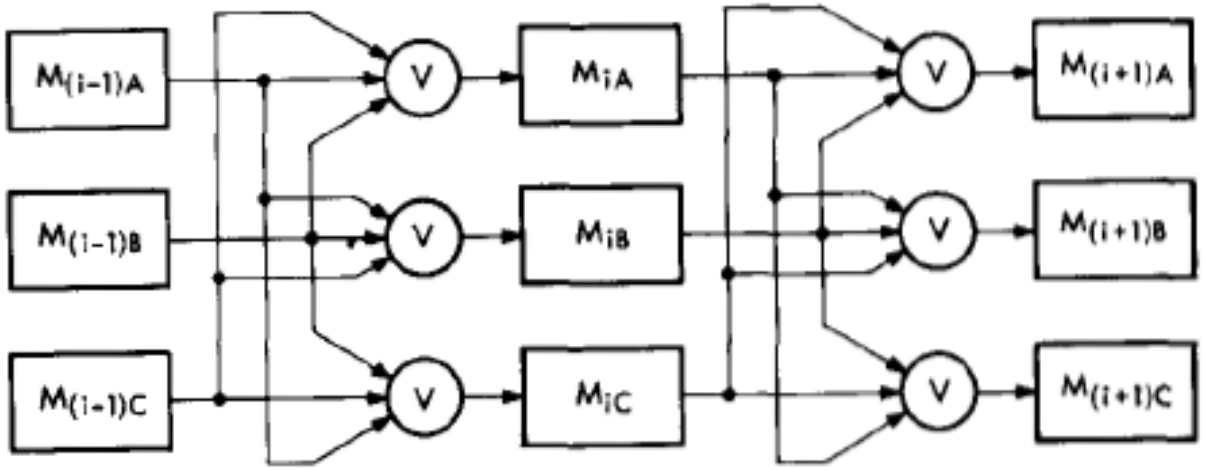


Figure 2 **Triple-modular-redundant configuration.**

Error Correction Code

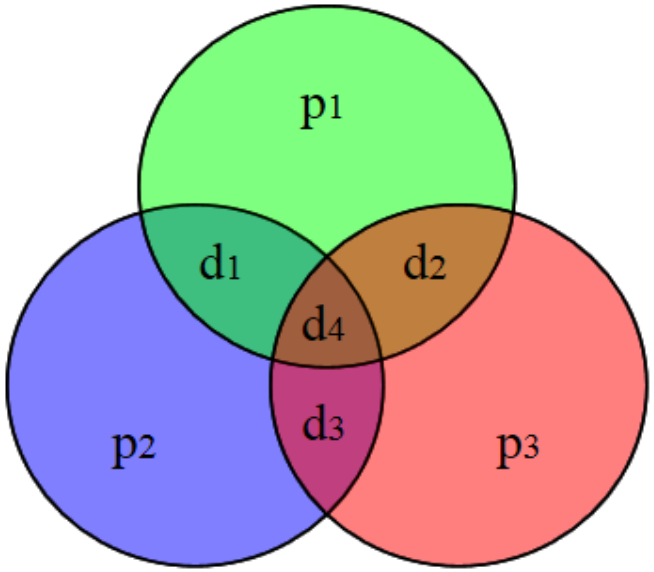
- Assumption: only one bit is incorrect
- Checking bit: (i.e. 1, 2, 4, 8, 16 …)
- Each checking bit can check several other bits
- Several checking bit can check one single bit

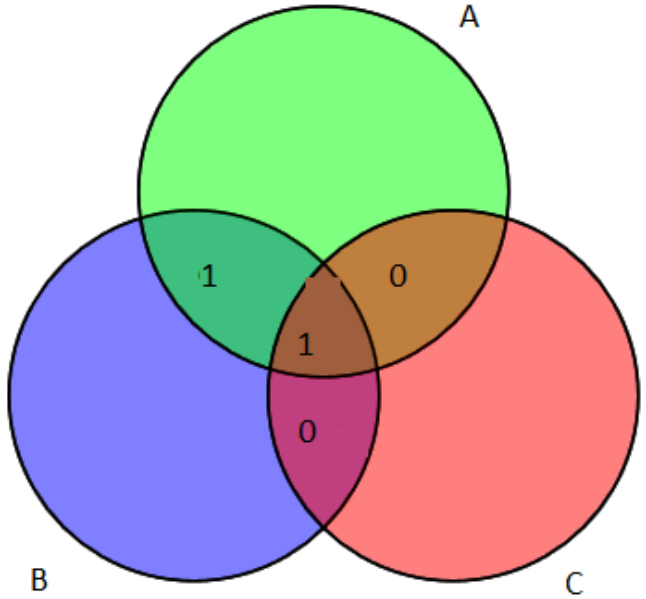
(Puzzled?)

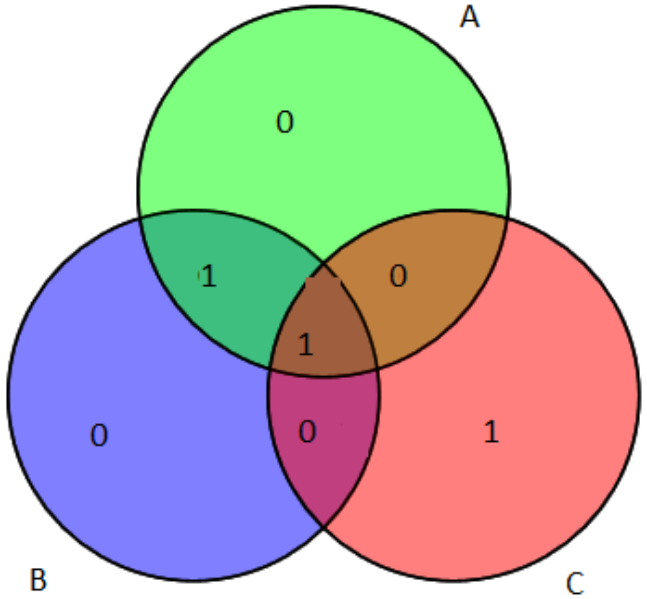
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Code	0	0	1	0	1	1	1	0	0	0	0	0	1	0	1	1	0	1	1	1	0

Error Correction Code

- Even Parity: make the number of bit 1 in checking range even.
- Simple illustration: Venn Diagram







Check bit: p1 p2 p3
Data bit: d1 d2 d3 d4

http://en.wikipedia.org/wiki/File:Hamming(7,4).svg

Error Correction Code

- Assign slot: 1111000010101110 -> xx1x111x0000101x01110

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Code	x	x	1	x	1	1	1	x	0	0	0	0	1	0	1	x	0	1	1	1	0

$$\text{Checking } Bit = 2^m$$

$$m = 0, 1, 2...$$

Error Correction Code

- Assign checking bit:
- 1=1
- 2=2
- 3=1+2
- 4=4
- 5=1+4
- 6=2+4
- 7=1+2+4
- …

$$Each\ Bit = \sum 2^m$$

$$m = 0, 1, 2...$$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Bit 1	X		X		X		X		X		X		X		X		X		X		X
Bit 2		X	X			X	X			X	X			X	X			X	X		
Bit 4				X	X	X	X					X	X	X	X					X	X
Bit 8								X	X	X	X	X	X	X	X						
Bit 16																X	X	X	X	X	X
Original code	0	0	1	0	1	1	1	0	0	0	0	0	1	0	1	1	0	1	1	1	0

Use Even Parity to determine the value of checking bit

Xx1x111x0000101x01110 -> 00101111000010110110

Error Correction Code

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Bit 1	X		X		X		X		X		X		X		X		X		X		X
Bit 2		X	X			X	X			X	X			X	X			X	X		
Bit 4				X	X	X	X					X	X	X	X					X	X
Bit 8								X	X	X	X	X	X	X	X						
Bit 16																X	X	X	X	X	X
Original code	0	0	1	0	1	1	1	0	0	0	0	0	1	0	1	1	0	1	1	1	0
Wrong code 1	0	0	1	0	0	1	1	0	0	0	0	0	1	0	1	1	0	1	1	1	0

00101111000001011011110 -> 00100110000010110111110

Error Correction Code

- Step 1: Check the Even Parity Invariant:

Check bit	Number of 1s	Result
1	5	Odd
2	6	Even
4	5	Odd
8	2	Even
16	4	Even

- Step 2: Ignore correct bit

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Bit 1	X		X		X		X		X		X		X		X		X		X		X
Bit 2		X	X			X	X			X	X			X	X			X	X		
Bit 4				X	X	X	X					X	X	X	X					X	X
Bit 8								X	X	X	X	X	X	X	X						
Bit 16																X	X	X	X	X	X

- Step 3: Find the mutual wrong bit

	1	4	5
Bit 1	X		X
Bit 4		X	X

Detected bit 5 is wrong!

- Step 4: Invert bit 5

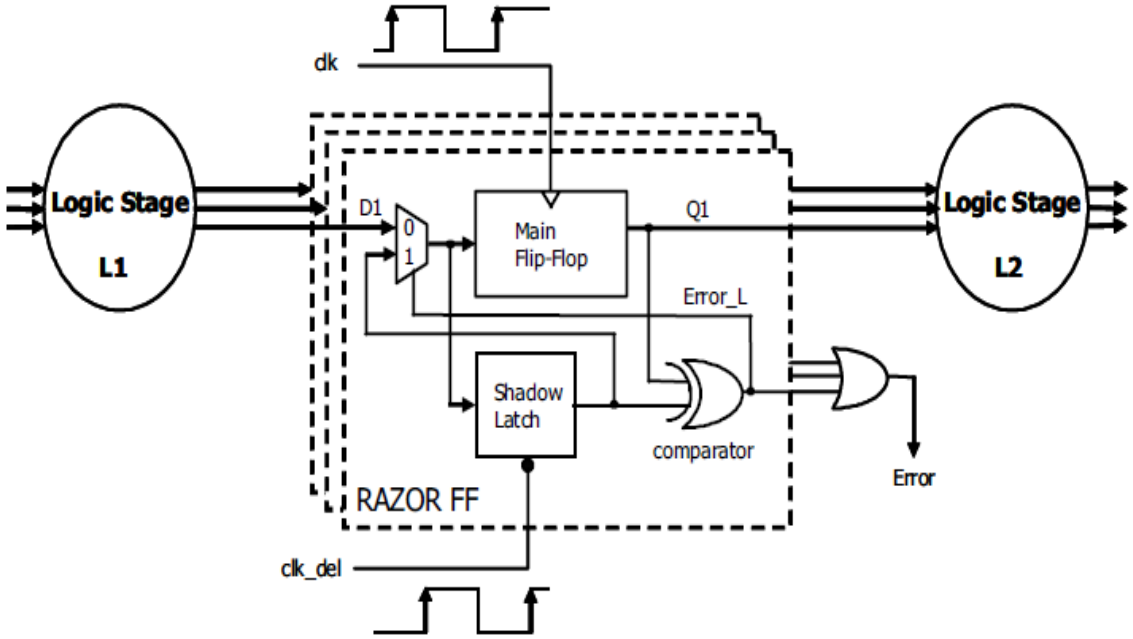
001001100000101101110 -> 001011100000101101110

DVS and Timing Error

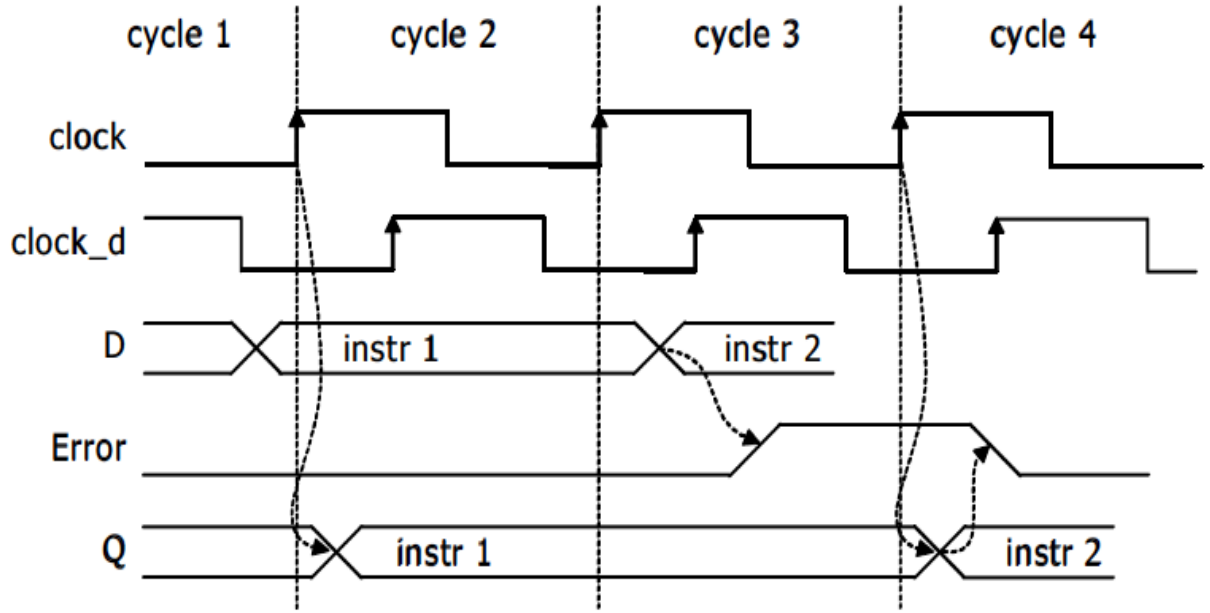
- Sometimes processor don’t need to operate a high frequency
  - 1080P video playback vs MP3 playback
- Dynamic Voltage Scaling (DVS)
  - Scale down clock frequency and supply voltage during non-critical instructions
  - Critical voltage – lowest voltage under which the processor can ensure correct operation
  - Margins added to consider process and ambient variations
  - Limit the degree of energy reduction
- Timing errors
  - Computational logic does not finish during the intended clock cycle
  - Next rising edge loses the value

Razor [Austin/Blaauw/Mudge]

- *In-situ* detection and correction of timing errors



(a)



(b)

Razor, Micro'03

Razor [Austin/Blaauw/Mudge]

- *In-situ* detection and correction of timing errors
  - Tune supply voltages dynamically according to error rates
  - Low error rates -> computation finished too quick -> lower voltage
  - High error rates -> clock period constraints are being violated -> increase voltage

  - Eliminate margins(process variation,temperature,dopant variations,coupling noise)
    - There variations may be data-dependent
    - Processor can operate under sub-critical voltage

Razor [Austin/Blaauw/Mudge]

- Meta-stability-tolerant design
  - Voltage hovers near Vdd/2
  - Have a meta-stability checker
  - Tow inverters that switch at different voltage levels
  - Error signal is double latched to detect a panic signal

- Up to 64.2% of energy savings
  - With little performance overhead (less than 3%)

Conclusion

- Circuit reliability becomes a bigger issue as technology scales
- Fault-tolerant Design
  - Double/Triple Module Redundancy
  - Error Correction Code
  - Timing Error Detection and Recovery (Razor)
- Research focus in architecture and circuit

References

- R. E. Lyons and W. Vanderkul, "*The Use of Triple-Modular Redundancy to Improve Computer Reliability*". IBM JOURNAL APRIL 1962
- Structured Computer Organization, 6th Edition, by Tanenbaum and Austin
- Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation, Dan Ernst et.al. MICRO'03
- Designing Reliable Systems From Unreliable Components: The Challenges of Transistor Variability and Degradation, Shekhar Borkar, Intel Fellow
- Robust Computing in the Nanoscale Era, Todd Austin, University of Michigan
- http://www.sourceresearch.com/newsletter/ESD.cfm?emART