Challenges of Future: Robust Design against Soft Errors with Low Power

Prof. Dhiraj Pradhan
Department of Computer Science
University of Bristol, UK
http://www.cs.bris.ac.uk/~pradhan/

University of BRISTOL

Overview

- Background
- Introduction
- Soft Error Problem
- Memory Solution
- Logic Solution
- Architecture Level
- Conclusion

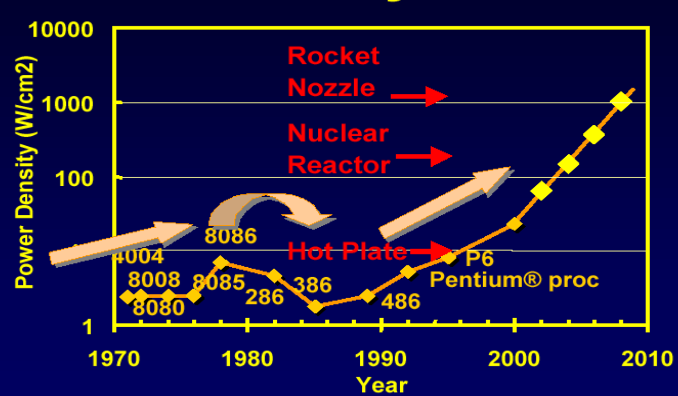
University of BRISTOL

Our Recent Related Publications

1. A Defect Tolerance Scheme for Nanotechnology Circuits. *IEEE Trans. On Circuits and Sytems I*, . December 2007 .
2. A Graph-Based Unified Technique for Computing and Representing Coefficients over Finite Fields . *IEEE Transactions on Computers*, 56(8)., pp. 1119–1132. August 2007.
3. Reliable Network-on-Chip Based on Generalized de Bruijn Graph. *DATE 2008.*
4. *An Efficient Technique for Synthesis and Optimization of Polynomials in GF(2m), ICCAD 2006.*
5. *"Efficient Method to Tolerate Multiple Bit Upsets in SRAM Memory". ETS 2007.*
6. *LPRAM: a novel low-power high-performance RAM design with testability and scalability, IEEE TCAD. May 2004*
7. *Simultaneous scheduling and binding for low gate leakage nano-complementary metal-oxide-semiconductor data path circuit behavioural synthesis. IET Computers & Digital Techniques, 2008.*
8. *Single Error Correcting Finite Field Multipliers over* $GF(2^m)$*. VLSID 08 .*
9. *GfXpress: An Efficient Technique for Synthesis and Optimization of Polynomials in* $GF(2^m)$*, IEEE TCAD 2008.*

University of BRISTOL

Power density will increase



Power density too high to keep junctions at low temp

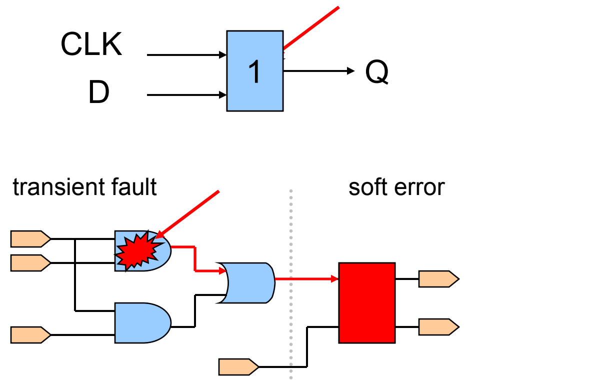
intel.

Source: Shekhar Borkar, Intel

16

University of BRISTOL

The Soft Error Problem

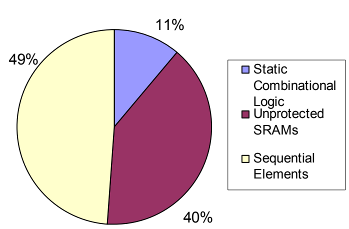


A 1->0 change at the above AND gate output may not cause soft error

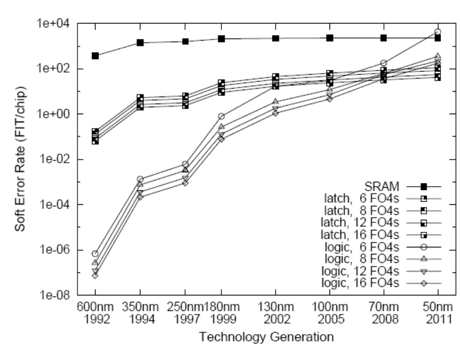
University of BRISTOL

Soft Error Rate Trends

Soft Error Rate Contributions



Mitra 2005

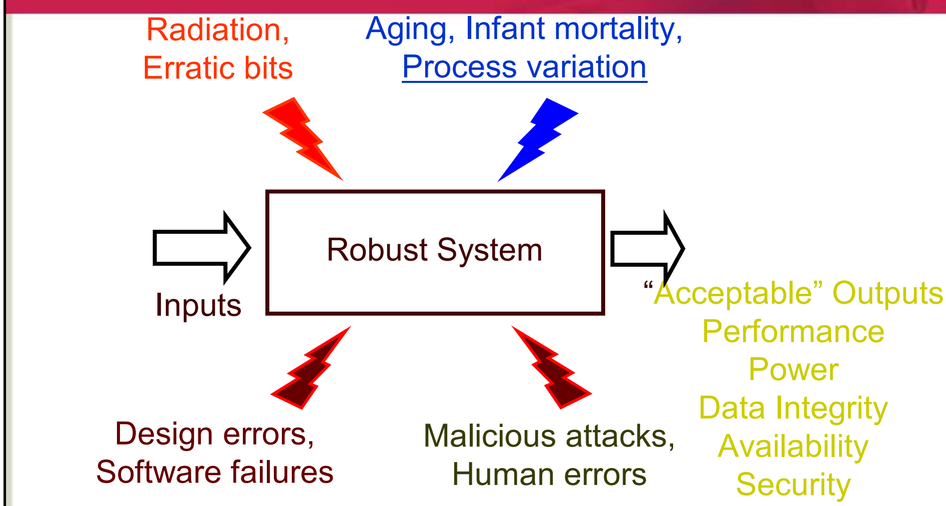


Shivakumar 2002

Increasing contribution of faults in combinational logic to the overall soft error rate

University of BRISTOL

Robust Systems in Scaled CMOS



Source: S Mitra

University of BRISTOL

Solution?

-Different solutions for different components

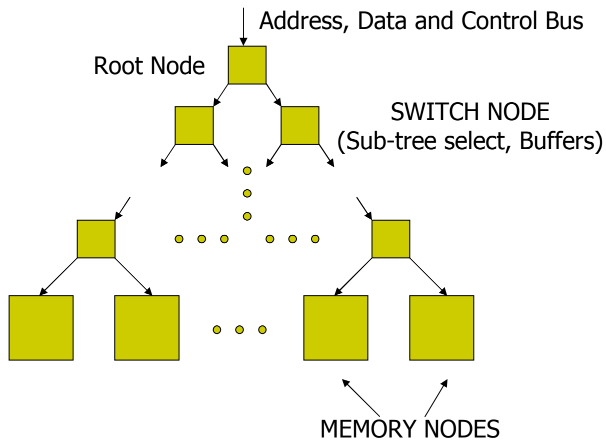
-Memory

-Logic

-Subsystem

University of BRISTOL

Low Power Soft Error Tolerant RAM Architecture



University of BRISTOL

Error Tolerant Low Power RAM

- Consider a 64 MB RAM
- This can be partitioned into 16 4MB RAM Modules

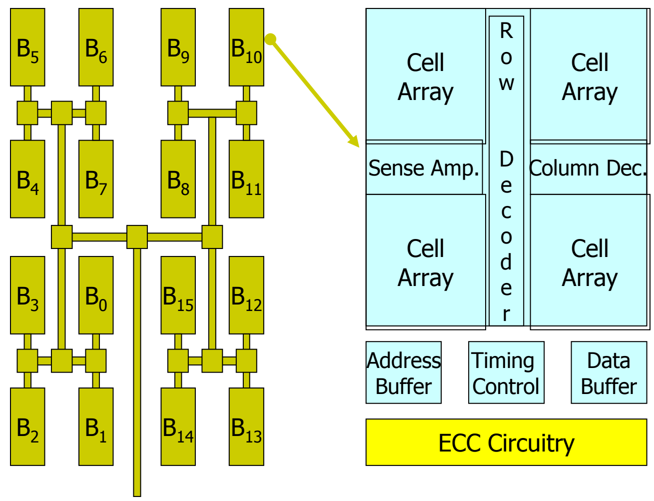
The 16 leaf nodes B_0, B_1…B_{15} connected in a binary tree and laid out as a H-Tree as shown next.

Assume the rows in each module are encoded using different classes of codes shown below

- Hamming Code denoted as H
- Matrix-Product code denoted as M
- Reed Muller Codes

University of BRISTOL

Error Tolerant Low Power RAM



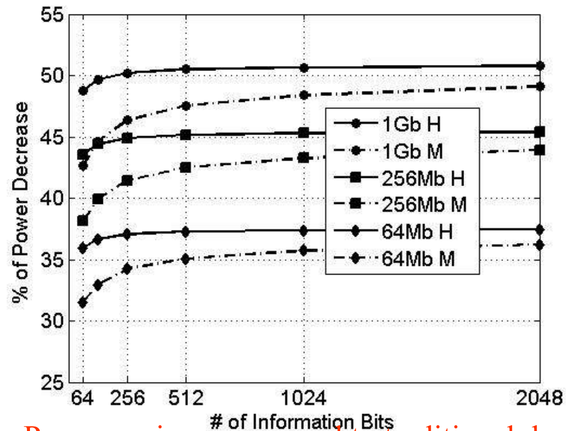
Different ECC Effectiveness

- **Hamming Code denoted as H(single error correction)**
- **Matrix-Product code denoted as M(double errror correction)**
- **Reed Muller Codes(two or more error correction)**

A row is segmented into Different sized data bits and encoded

University of BRISTOL

Reduction in Power Consumption compared to traditional design

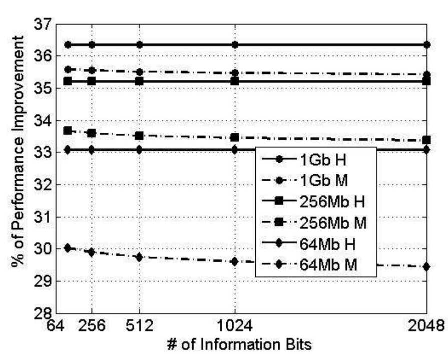


H- Uses Hamming
M- Uses Matrix product code

Power savings compared to traditional design
Guaranteed protection against certain number of Soft Errors

University of BRISTOL

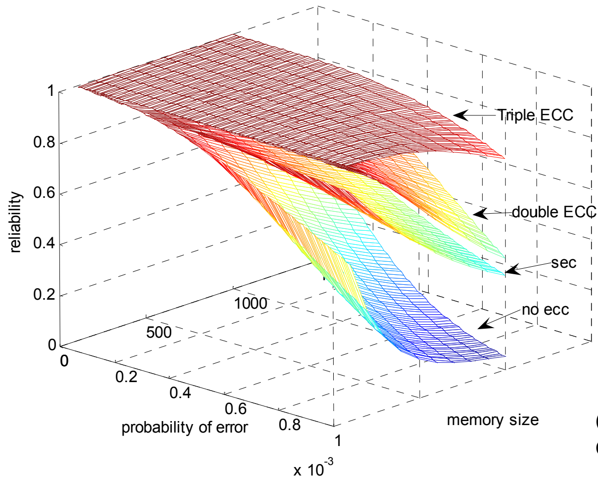
Performance improvement along with soft error correction



Delay is reduced compared to traditional design
Soft Error Protection Incorporated

University of BRISTOL

Reliability Results



(no. information bits
On z-axis)

Memory size 64Mbits

University of BRISTOL

Logic Synthesis for Low Power and Soft Error Robustness using Finite Field Theory

University of BRISTOL

Finite Field

- Also called *Galois Field,* denoted by GF(N)
- $N = p^k$, p is a prime number, and k an integer
- Each element is a k-tuple
- There are exactly N elements in the field (0, 1,…, N-1) or $(0, 1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{N-1})$ where , α - primitive element

University of BRISTOL

Finite Field

- Two operators
  - '+' called addition operation forming Abelian Group
  - '.' called multiplication operation forming Abelian group.
- Additive and multiplicative identities 0 and 1 respectively
- Additive and multiplicative inverse exists for each element

University of BRISTOL

GF(4) Elements

$[0, 1, \alpha, \alpha^2] = [0, 1, \alpha, \beta]$

Elements can be represented in $GF(2^n)$ as 2-tuples over GF(2) as shown below

University of BRISTOL

Example GF(4)

Let $\alpha^2 = \beta$

Thus we have 4 elements 0, 1, α and β

GF(4)	2-tuple of GF(2)	Polynomial Representation
0	0 0	0
1	0 1	1
α	1 0	x
β	1 1	(x+1)

University of BRISTOL

Addition and Multiplication

- Addition and Multiplication in GF(4)

+	0	1	α	β
0	0	1	α	β
1	1	0	β	α
α	α	β	0	1
β	β	α	1	0

*	0	1	α	β
0	0	0	0	0
1	0	1	α	β
α	0	α	β	1
β	0	β	1	α

- β is multiplicative inverse of α, and vice versa.

Addition

The addition of any two elements in GF(4), can be performed as addition of polynomials over GF(2)

For example, consider $\alpha + \beta$

$\alpha = x$ and $\beta = x + 1$

$\boldsymbol{\alpha + \beta = x + x + 1 = 2x + 1}$

Since $2x = 0$ over $GF(2^2)$

$\boldsymbol{\alpha + \beta = x + x + 1 = 1}$

Multiplication

Multiplication is done by using polynomials mod a primitive polynomial P(x)

Let

$P(x) = x^2 + x + 1$

Thus

$\alpha * \beta = x(x + 1) = x^2 + x$

$(x^2 + x) \bmod x^2 + x + 1 = 1$

So

$\alpha * \beta = 1$

$\alpha^{-1} = \beta$, $\beta^{-1} = \alpha$

α and β are inverse of each other

University of BRISTOL

An Expansion Theorem (Pradhan 1978)

Theorem: *Any n-variable function $f(x_1,\ldots,x_i,\ldots,x_n)$ in GF(N) can be expanded as,*

$$f(x_1, \quad , x_i, \quad , x_n) = \sum_{e=0}^{N-1} [1 - (x_i - \delta(e))^{N-1}] f_{x_i = \delta(e)}$$

- *where $\delta : I_N \rightarrow GF(N)$ is a one-to-one mapping, with $I_N = \{0,1,\ldots,N-1\}$, and $\delta(0)=0$.*
- $f_{xi=\delta(e)}$ is called the *cofactor* of f with respect to $x_i = \delta(e)$

- The term $[1 - (x_i - \delta(e))^{N-1}]$ is a *multiple-valued literal* in GF(N).

University of BRISTOL

Special Case

- Let $N = 2$
- Then,

$$
\begin{aligned}
f(x_1, \ \ , x_i, \ \ , x_n) &= \sum_{e=0}^{1} [1-(x_i - \delta(e))] f_{x_i=\delta(e)} \\
&= [1-(x_i-\delta(0))] f_{x_i=\delta(0)} + [1-(x_i-\delta(1))] f_{x_i=\delta(1)} \\
&= [1-x_i] f_{x_i=\delta(0)} + x_i f_{x_i=\delta(1)} \\
&= \bar{x}_i f_{x_i=\delta(0)} + x_i f_{x_i=\delta(1)}
\end{aligned}
$$

But this is Shannon's expansion theorem!
i.e., the expansion theorem reduces to Shannon's theorem in GF(2).

GFMODD

- Galois Field Multiple Output Decision Diagram (GFMODD)
- Based on Finite Field

University of BRISTOL

An Example 2-Bit Multiplier

Bit level 2x2 multiplier

X		Y		Z			
x_1	x_0	y_1	y_0	z_3	z_2	z_1	z_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

word level 2x2 multiplier

X	Y	Z	
		Z_1	Z_2
0	0	0	0
0	1	0	0
0	α	0	0
0	β	0	0
1	0	0	0
1	1	0	1
1	α	0	α
1	β	0	β
α	0	0	0
α	1	0	α
α	α	1	0
α	β	1	α
β	0	0	0
β	1	0	β
β	α	1	α
β	β	α	1

University of BRISTOL

2-Bit Multiplier Galois Expression

- Inputs partitioned as $X = (x_1, x_0)$, $Y = (y_1, y_0)$
- Outputs partitioned as $Z_1 = (z_3, z_2)$, $Z_2 = (z_1, z_0)$
- GFMODD yields

$$Z_1 = \alpha X^3Y^2 + \alpha XY^2 + X^3Y + \alpha X^2Y + \beta XY + \beta X^3Y^3 + \alpha X^2Y^3 + XY^3$$
$$Z_2 = \beta X^2Y^2 + \beta X^2Y + \beta XY^2 + \alpha XY$$

Gfxpress (TCAD 2008) uses an algorithm using the Polynomial description as input to synthesize low power designs. This has been recently augmented to incorporate soft Error correction

University of BRISTOL

Power Aware Soft Error Robust Multiplier Design

Pimitive Poly	Synopsys Tool Only (Area,Delay,Power)	(EXOR,AND)	Our Tool and Synopsys tool (Area,Delay,Power)
131	(30866.1,7.7,39.1)	(48,49)	(1699.9,1.3,2.4)
137	(30356.5,8.3,39.9)	(50,49)	(1709.6,1.6,2.5)
143	(33375.6,8.0,40.3)	(67,49)	(1925.7,1.9,3.0)
145	(33469.1,8.1,42.3)	(50,49)	(1703.1,1.7,2.5)
157	(35020.6,7.4,42.7)	(65,49)	(1964.4,2.1,3.1)
167	(34978.7,7.7,39.7)	(65,49)	(1964.4,1.7,3.2)
171	(39023.2,8.5,49.1)	(64,49)	(1906.3,1.4,2.9)
185	(34797.9,8.7,42.1)	(66,49)	(2003.1,1.4,3.0)
191	(34843.1,8.8,41.0)	(75,49)	(2077.3,1.8,3.2)
193	(30321.0,8.0,38.8)	(48,49)	(1725.7,3.2,2.7)
203	(33875.6,8.0,42.0)	(64,49)	(1922.5,1.5,3.0)
211	(33440.1,8.2,40.3)	(64,49)	(1935.4,1.7,3.1)
213	(33488.5,8.0,38.9)	(65,49)	(2009.6,1.8,3.3)
229	(34578.6,8.2,41.3)	(67,49)	(1980.5,1.8,3.2)
239	(34123.9,9.9,41.5)	(72,49)	(2041.8,1.6,3.1)
241	(32298.3,8.7,36.7)	(61,49)	(1899.9,1.6,3.0)
247	(36704.1,7.2,45.0)	(72,49)	(2106.3,1.6,3.2)
253	(33782.0,8.3,41.3)	(74,49)	(2106.3,2.0,3.2)

Area in 10^{-6} mm^2,
delay in ns,
power in microW at 1.8V

University of BRISTOL

Power Aware Soft Error Robust Adder Design

Adder	Synopsys Only (area,delay,power)	(a,m)	With our technique (area,delay,power)
2-bit	(119.3,0.3,115.7)	(4,5)	(106.4,0.4,113.1)
3-bit	(197.8,0.7,214.3)	(11,6)	(216.1,0.81,241)
4-bit	(274.2,1.21,301.8)	(17,8)	(316.1,1.08,378.2)
5-bit	(383.8,1.22,468.5)	(23,10)	(416.1,1.4,515.8)
6-bit	(493.5,1.58,590)	(29,12)	(509.6,1.72,627.5)
7-bit	(638.7,1.61,791.1)	(35,14)	(609.3,2.04,752.8)
8-bit	(687.0,2.0,809.4)	(41,16)	(709.6,2.38,879)
9-bit	***(11877,3.7,13800)***	***(47,18)***	***(809.6,2.69,1000)***

An order of magnitude improvement!

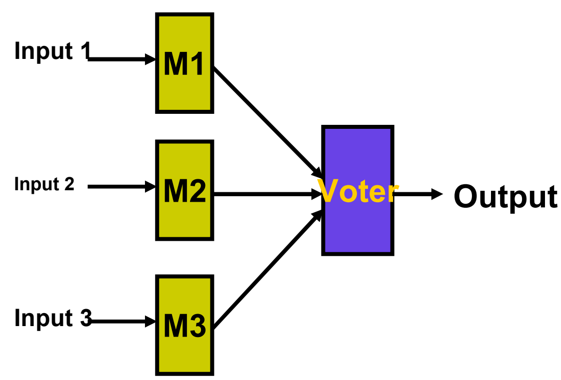
University of BRISTOL

Architecture Level Solution

University of BRISTOL

Traditional Soft Error Tolerance at the system Level

- TMR
  - 3X power



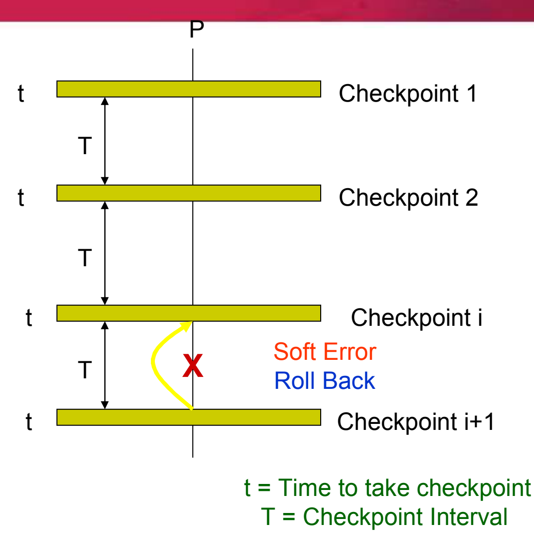
Power Dissipation is too high and Errors In Two or More Modules are not tolerated

University of BRISTOL

Rollback Recovery

- Checkpoints are taken at regular intervals
- When an error is detected the program is rolled back to the last checkpoint and re-executed
- Loss of performance. One checkpoint interval for every occurrence of soft error.
- Loss of Power
- Limited to one check point interval

Rollback Recovery

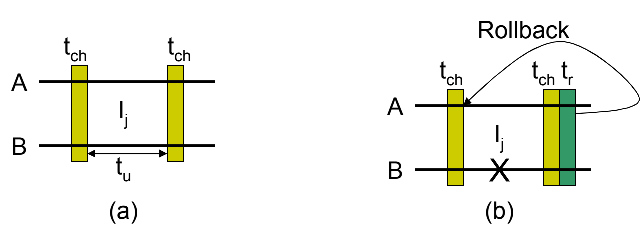


How to detect faults

- Diagnostic Program (single module)
- Time overhead of running the diagnostic program.
- Fault coverage a problem
- Duplicated System
- Requires two modules
- Comparison
- Fault coverage can be good

University of BRISTOL

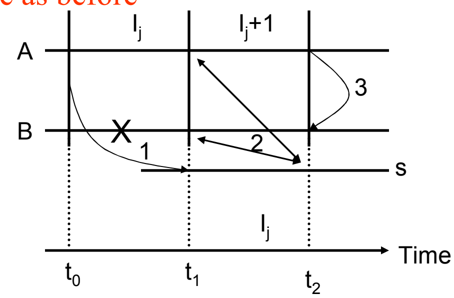
Traditional Duplex Rollback Schemes



University of BRISTOL

A Novel Power Aware Soft Error Tolerant Scheme (Roll forward)

Spare is released at time t_2 and the system continues to operate as before



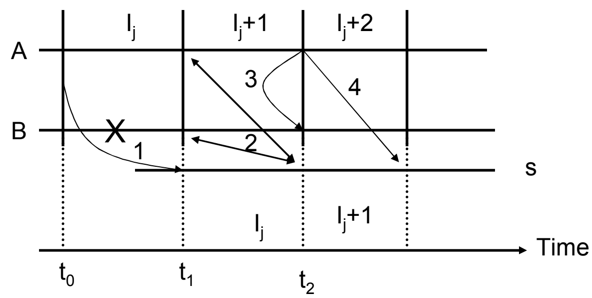
1: Copy state to the spare
2: Compare state of the spare with the state of A and B
3: Copy state from A to B
X A fault

Roll forward Cont.

- Spare is activated for a short duration (a single check point interval) when there is a soft error.
- This scheme tolerate all single transient/intermittent faults with minimum loss of performance and power.
- This scheme also tolerates hard faults and various multiple transient faults

Roll Forward Scheme with additional checking of the spare

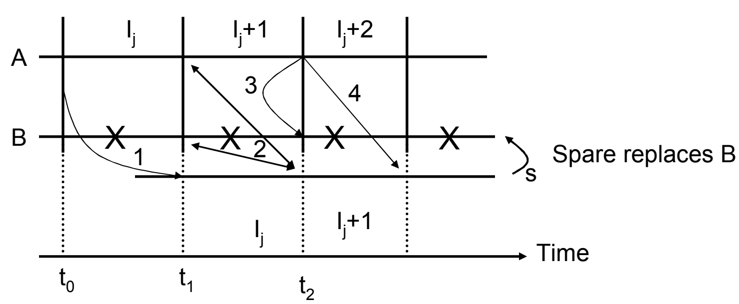
Better Reliability with Minimal loss of Performance with soft errors



4: Compare state of the spare and module A

University of BRISTOL

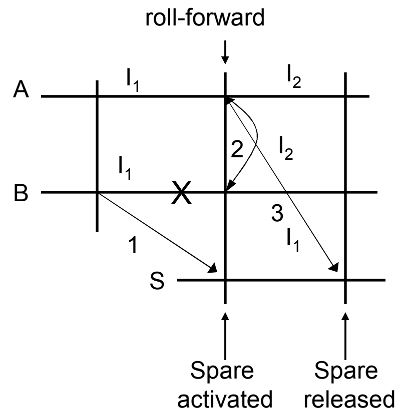
Roll Forward Scheme with a Permanent fault in B



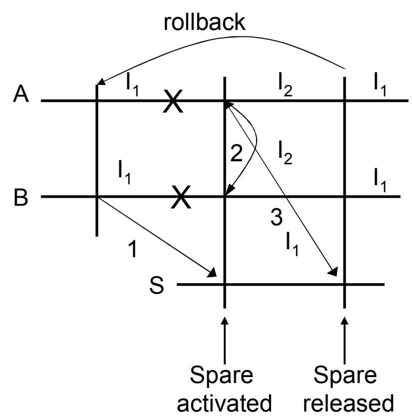
Spare replaces faulty module

University of BRISTOL

Roll forward with minimal Power/performance penalty



Double failures can force a roll back. The additional power required still remains the same



Future Challenges

- Develop new Design Paradigm: Soft Error Tolerant Low Power Design
- Hierarchical Soft Error Tolerance with low Power
- Design for Testing under Process Variation
  -It is a major challenge because variability may cause soft error
- Impact of workload and speed on soft error rate
- Trade-off of Soft error robustness againt power

University of BRISTOL

Other ongoing Research in Our Group

- Memory Cell design for robustness against Soft Error
- Logic Synthesis for Low power using TED
- Logic Error Correction using Mutliple Parity bits
- Soft Error Tolerant Lowpower Sensor Networks

University of BRISTOL


BRISTOL
Thank You
University of
BRISTOL