

Designing Reliable Systems with Unreliable Devices

Challenges and Opportunities

Luca Benini

DEIS Università di Bologna

lbenini@deis.unibo.it

The “quantistic” revolution

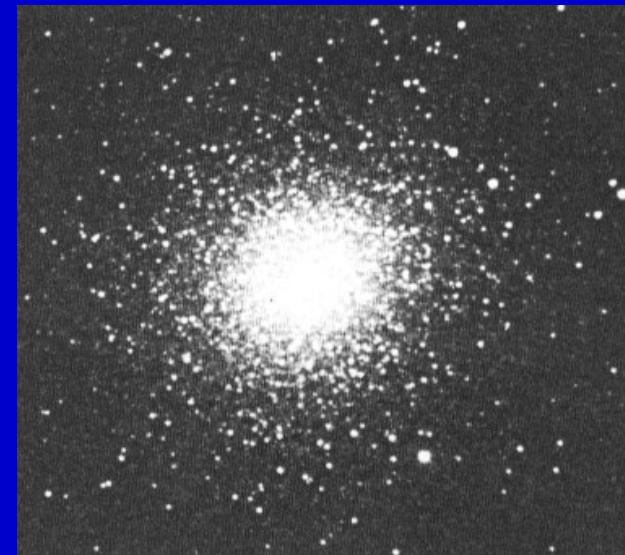
Physics at the onset of the XX century

– *Laplacian determinism*

- The future state of the universe can be determined from its present state

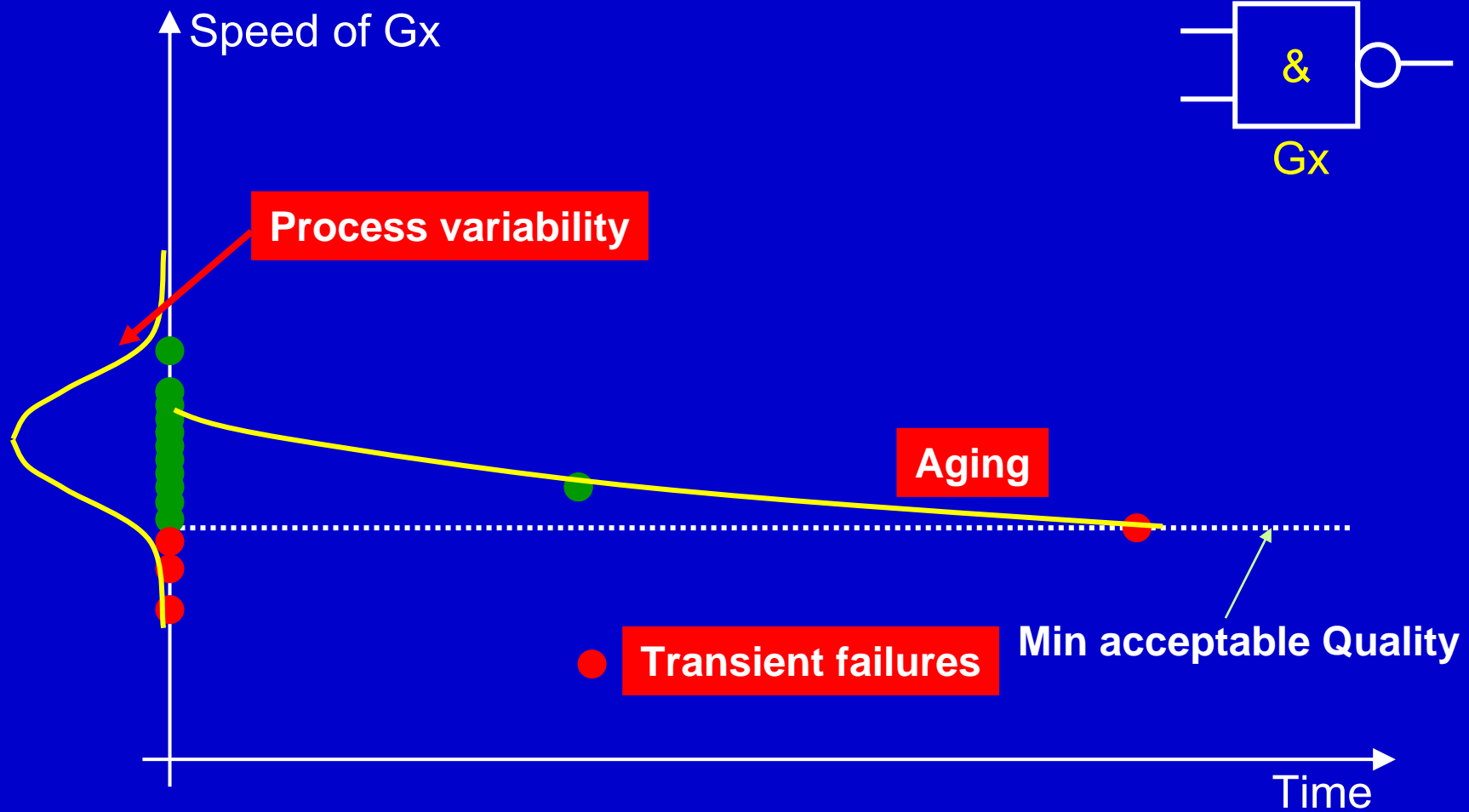
– *Quantum theory and uncertainty*

- We can neither observe nor control microscopic features with accuracy

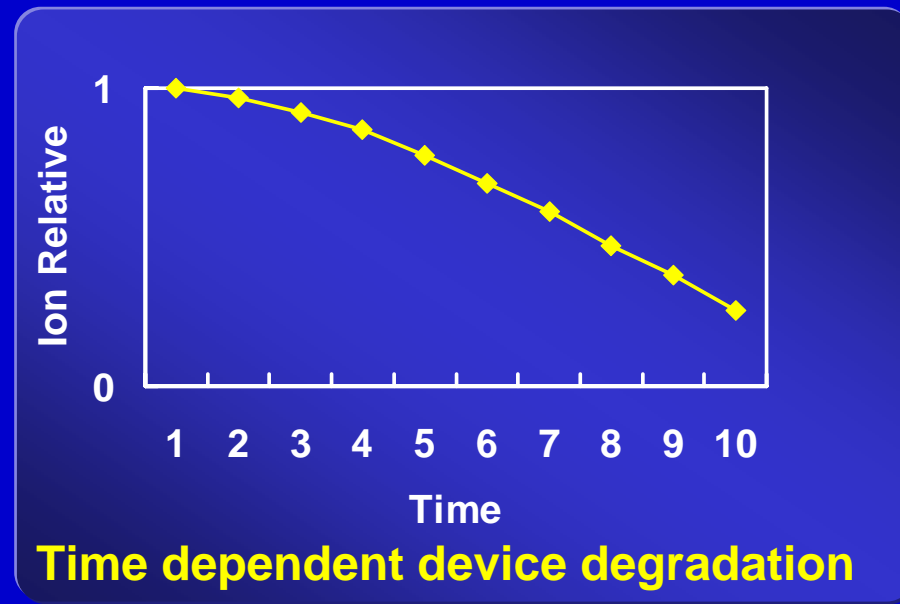
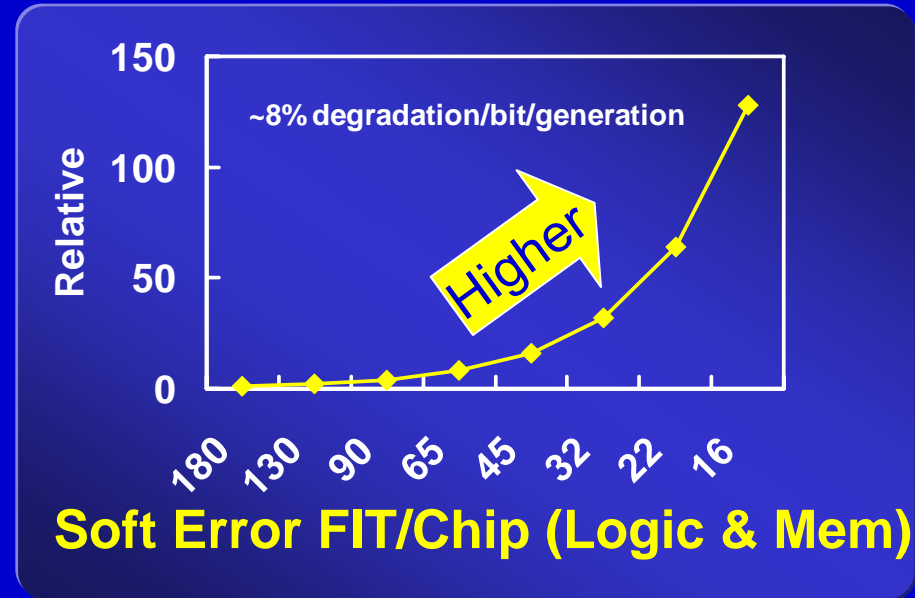
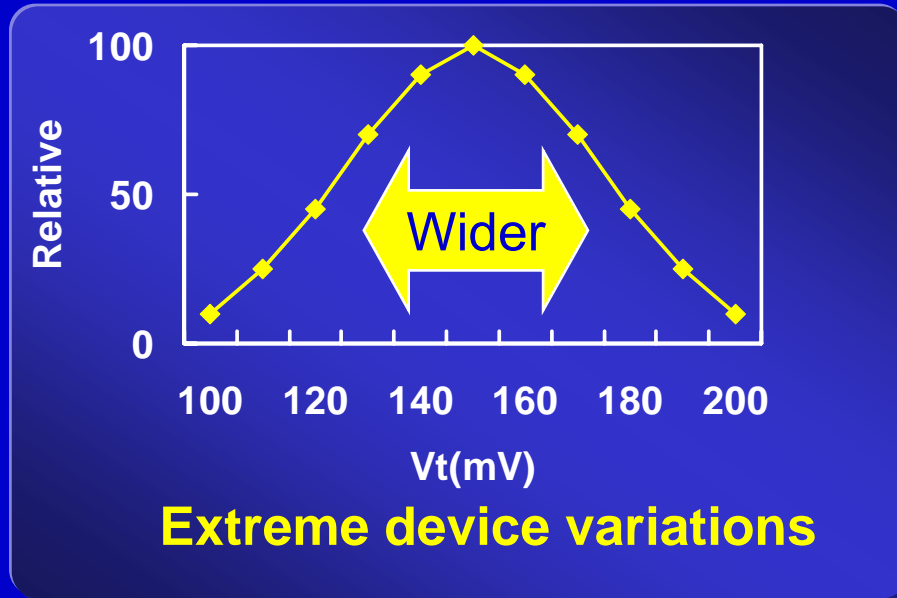


Similar situation for CMOS design in XXI century

Failures over time

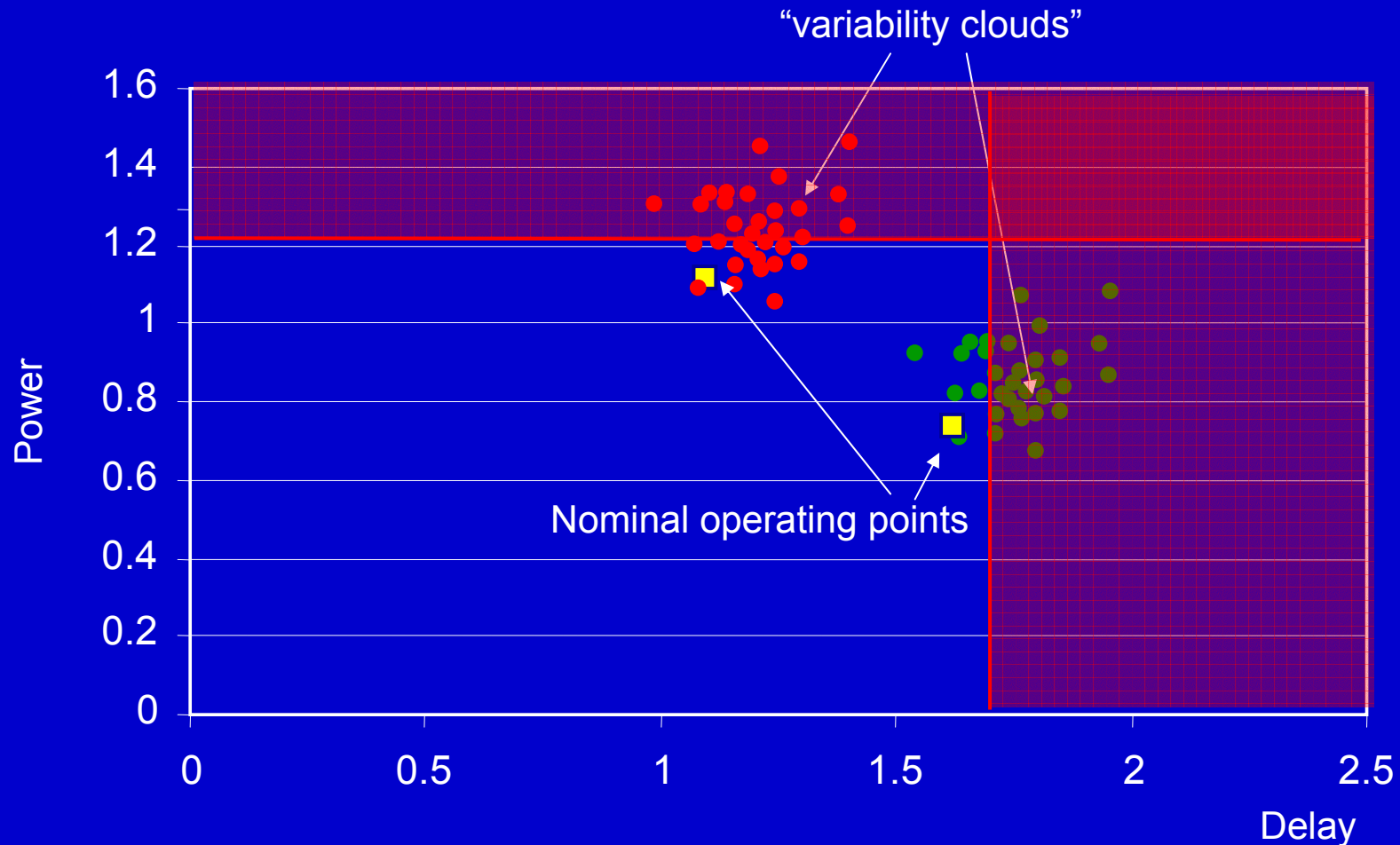


Quantitative view



[Borkar05]

Multi-dimensional variability



Design margins do not solve the problem!

Technology Outlook

High Volume Manufacturing	2004	2006	2008	2010
Technology Node (nm)	90	65	45	32
Integration Capacity (BT)	2	4	8	16
Delay = CV/I scaling	0.7	~0.7	>0.7	De
Energy/Logic Op scaling	>0.35	>0.5	>0.5	Ene
Bulk Planar CMOS	High Probability			
Alternate, 3G etc	Low Probability			
Variability	Medium			Hig
ILD (K)	~3	<3		Redu
RC Delay	1	1	1	1
Metal Layers	6-7	7-8	8-9	0.5

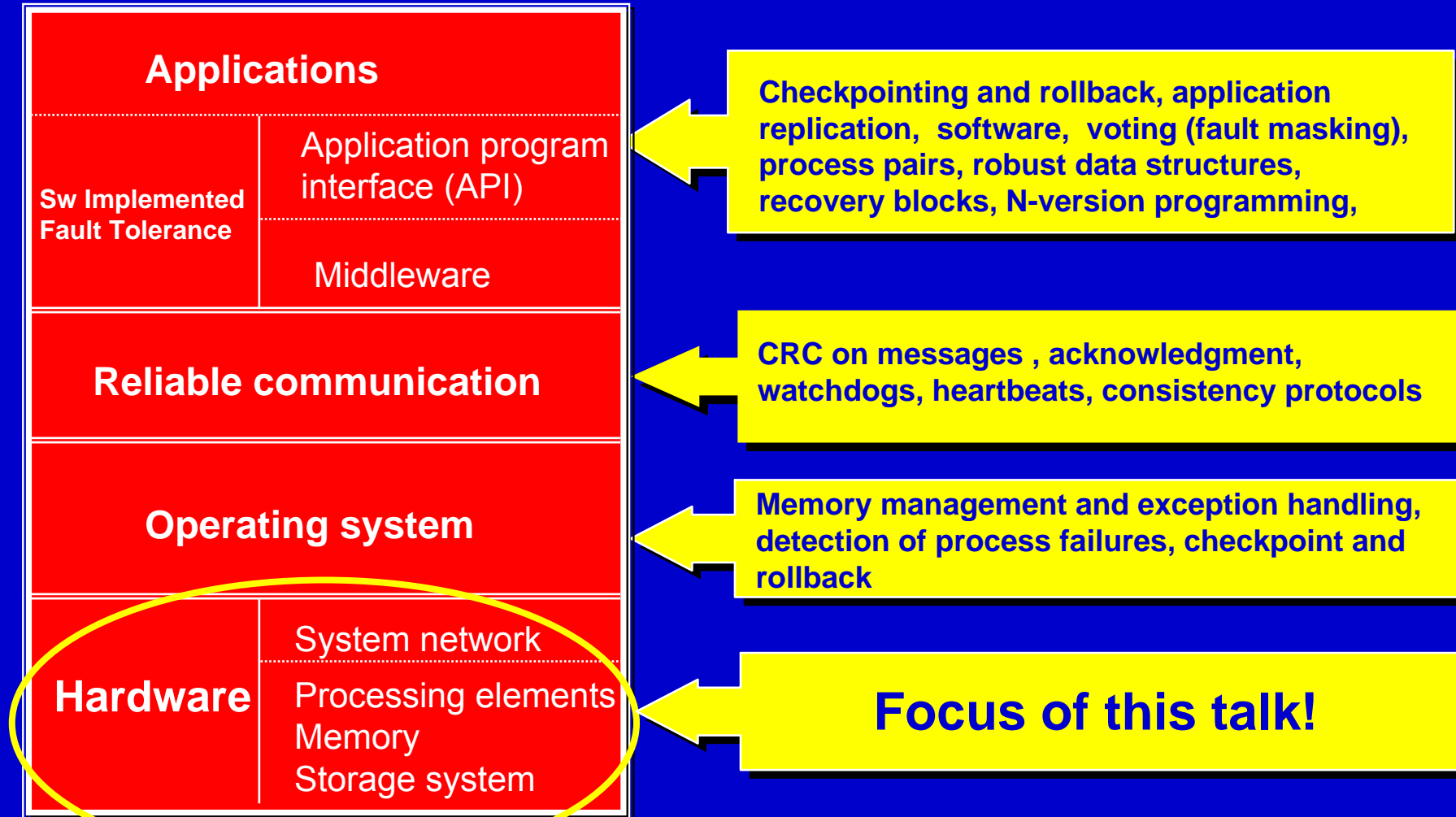
Variations are increasing!

Implications

- Extreme variations (Static & Dynamic) will result in unreliable components
- Impossible to design reliable system as we know today
 - Static & dynamic **variations**
 - Transient **errors** (Soft Errors)

Reliable systems with unreliable components
Resilient circuits, architectures, systems

Reliability is a system issue



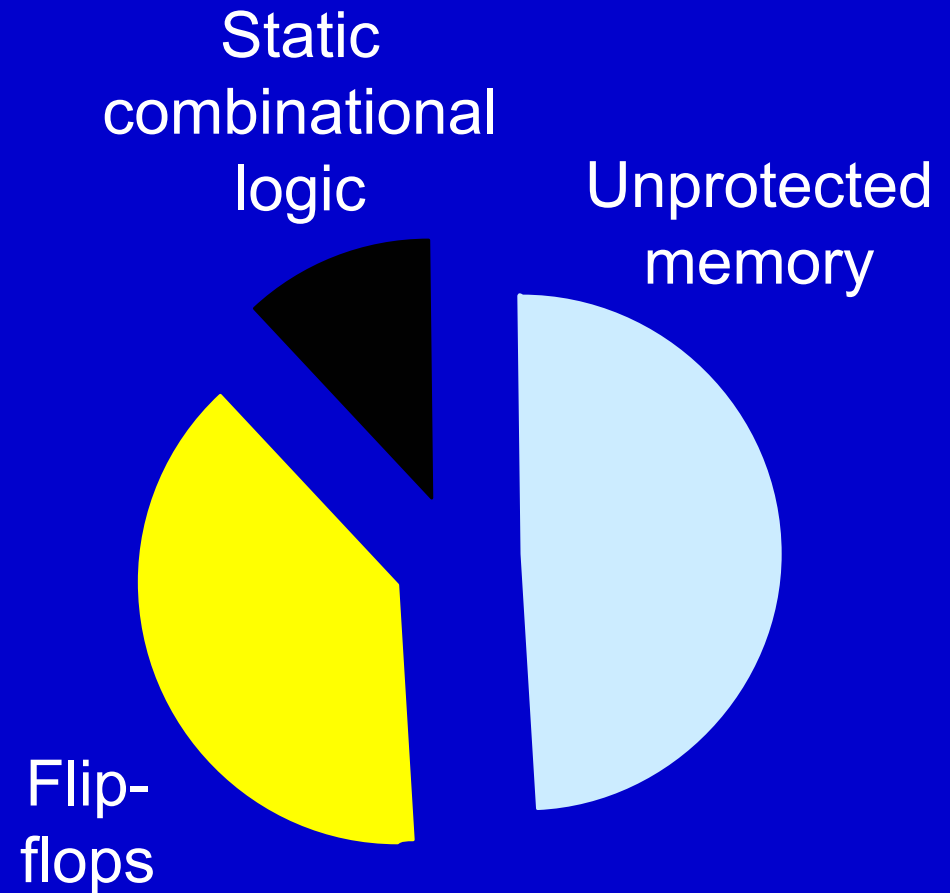
[Iyer]

Parsimonious Robustness

- Robustness with less HW overhead than traditional DUP, TMR techniques.
- Guiding principles
 - Focus on high-impact reliability threats
 - Exploit existing HW redundancy
 - Provide protection only for HW structures with high **architectural vulnerability**
 - Architectures have a degree of “self-healing” capability: e.g. SEU in a register that is rewritten before being read

Logic Soft Errors

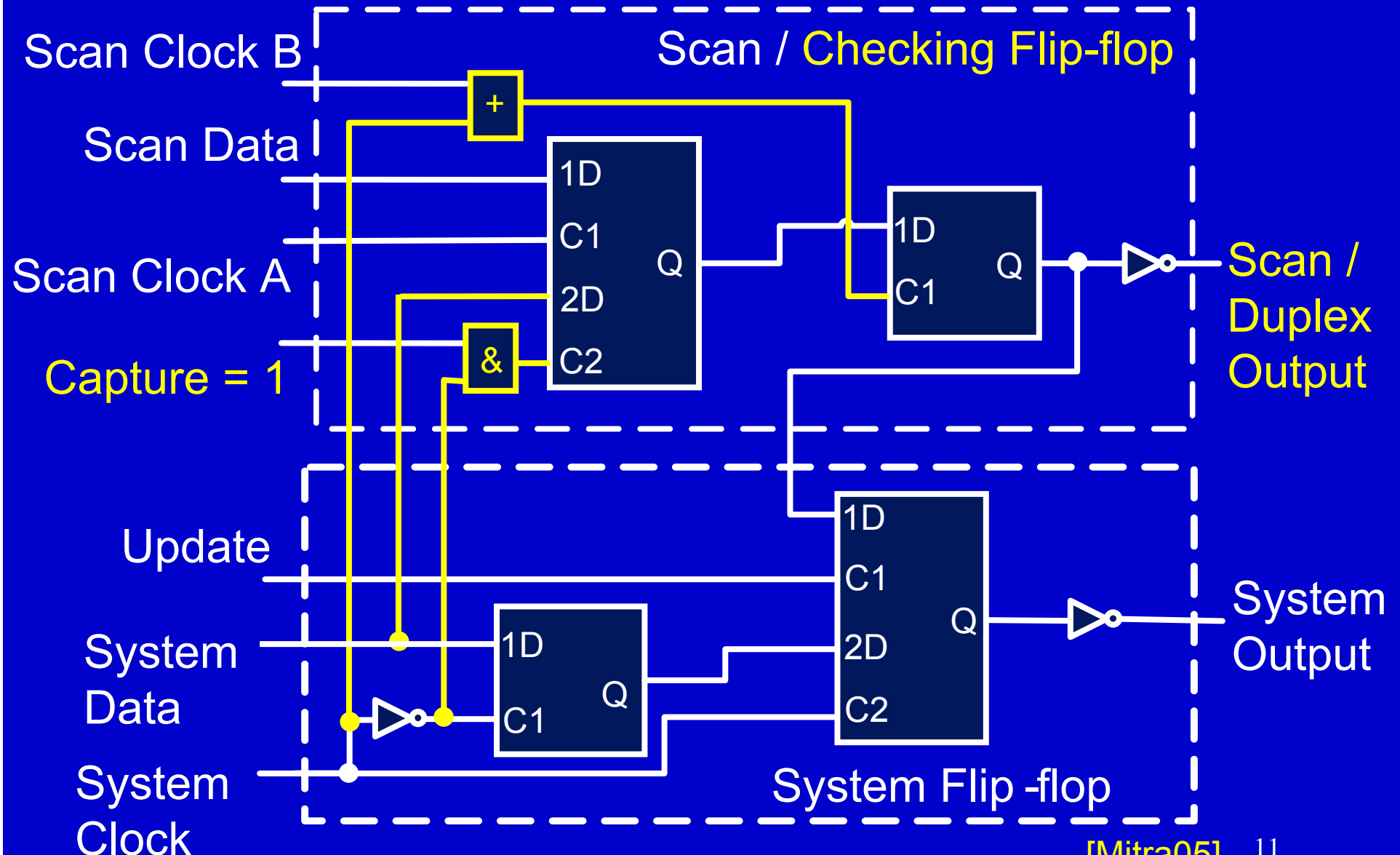
- Soft errors affecting
 - Flip-flops
 - Latches
 - Combinational logic
- Memory soft errors
 - Much more work
 - ECC approaches



Soft Error rate contributions

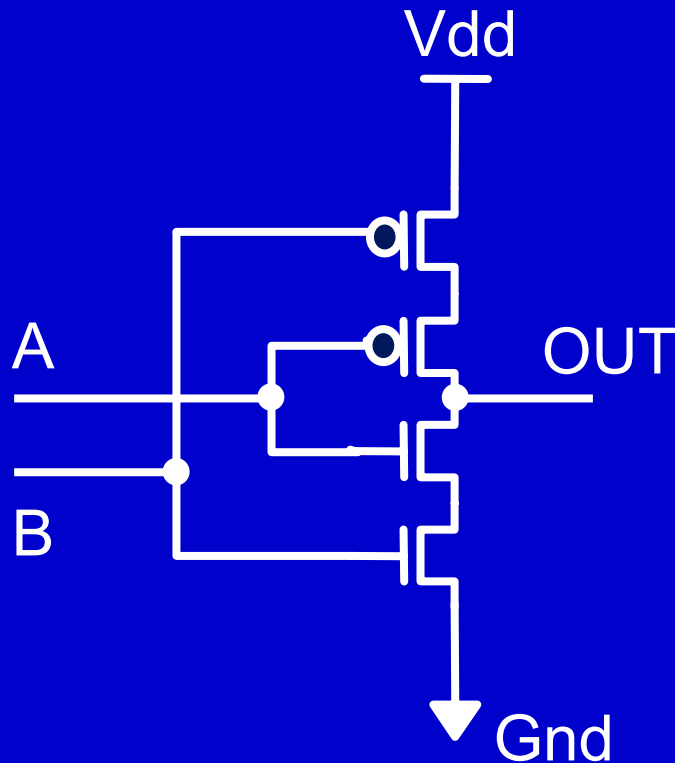
[Mitra05]

Error Resilient FFs



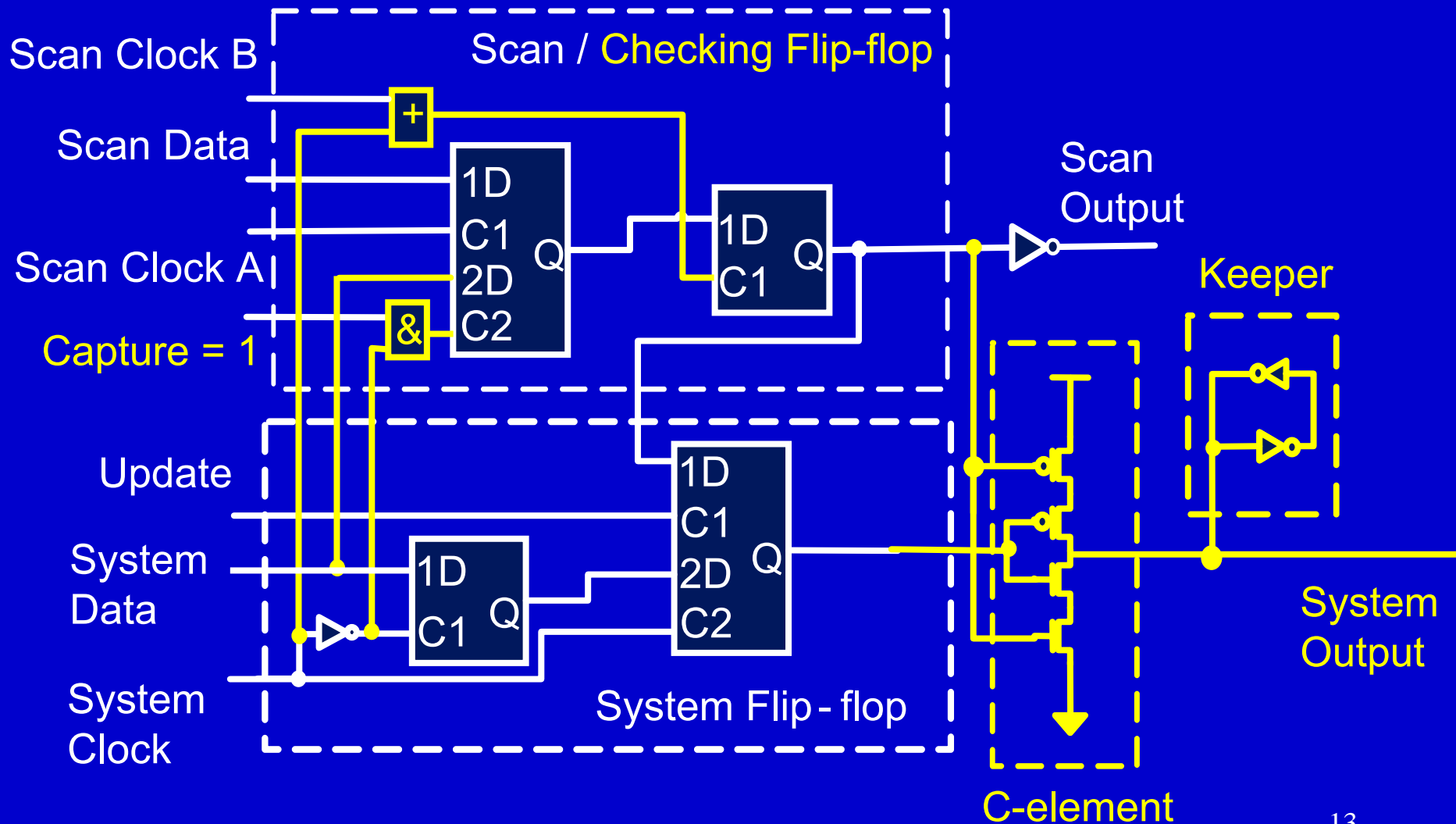
C-element

- Extensive use in asynchronous circuit design



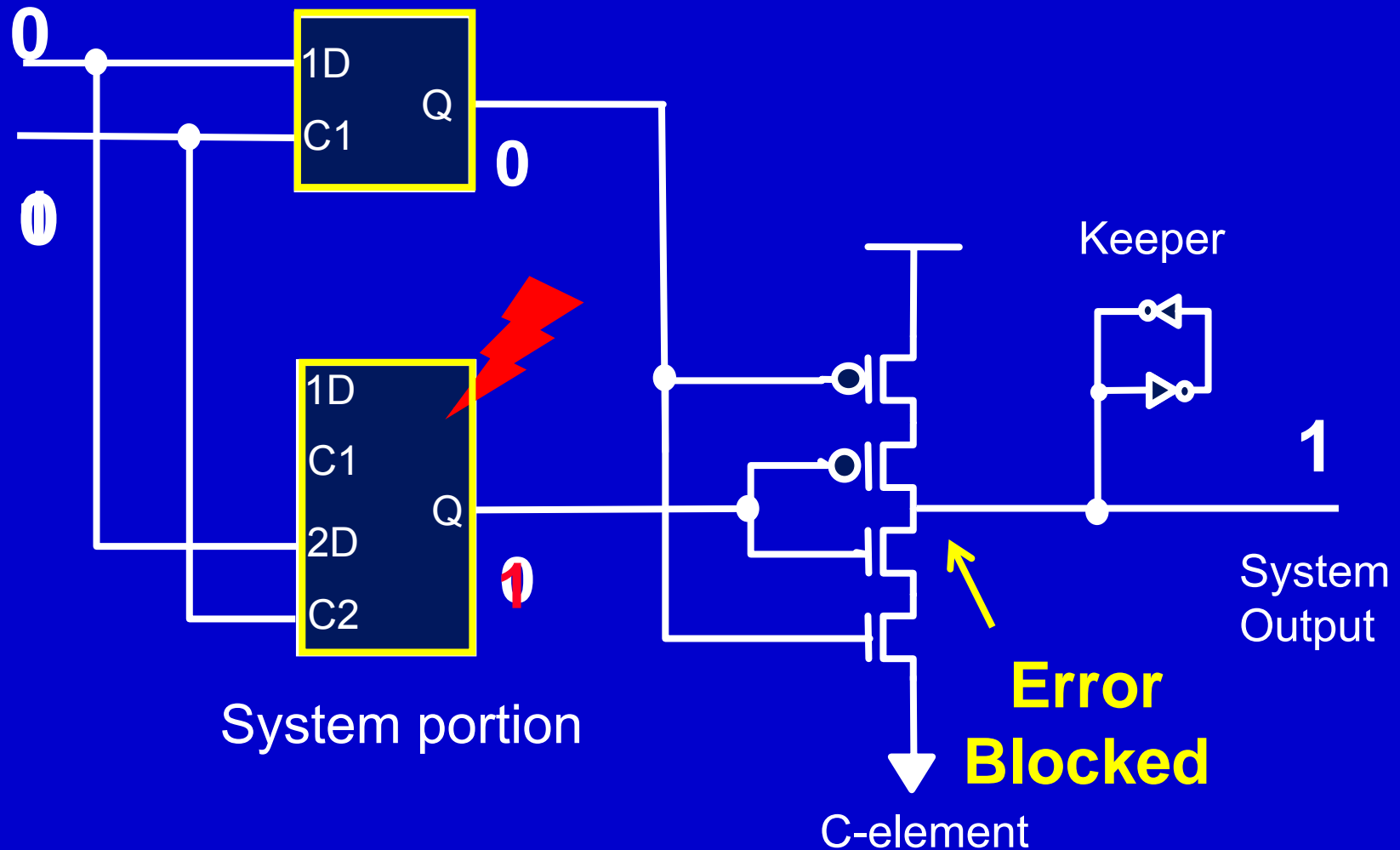
A	B	OUT
0	0	1
1	1	0
0	1	Previous value retained
1	0	Previous value retained

Error Blocking Design



Error Blocking Operation

Scan / Checking portion

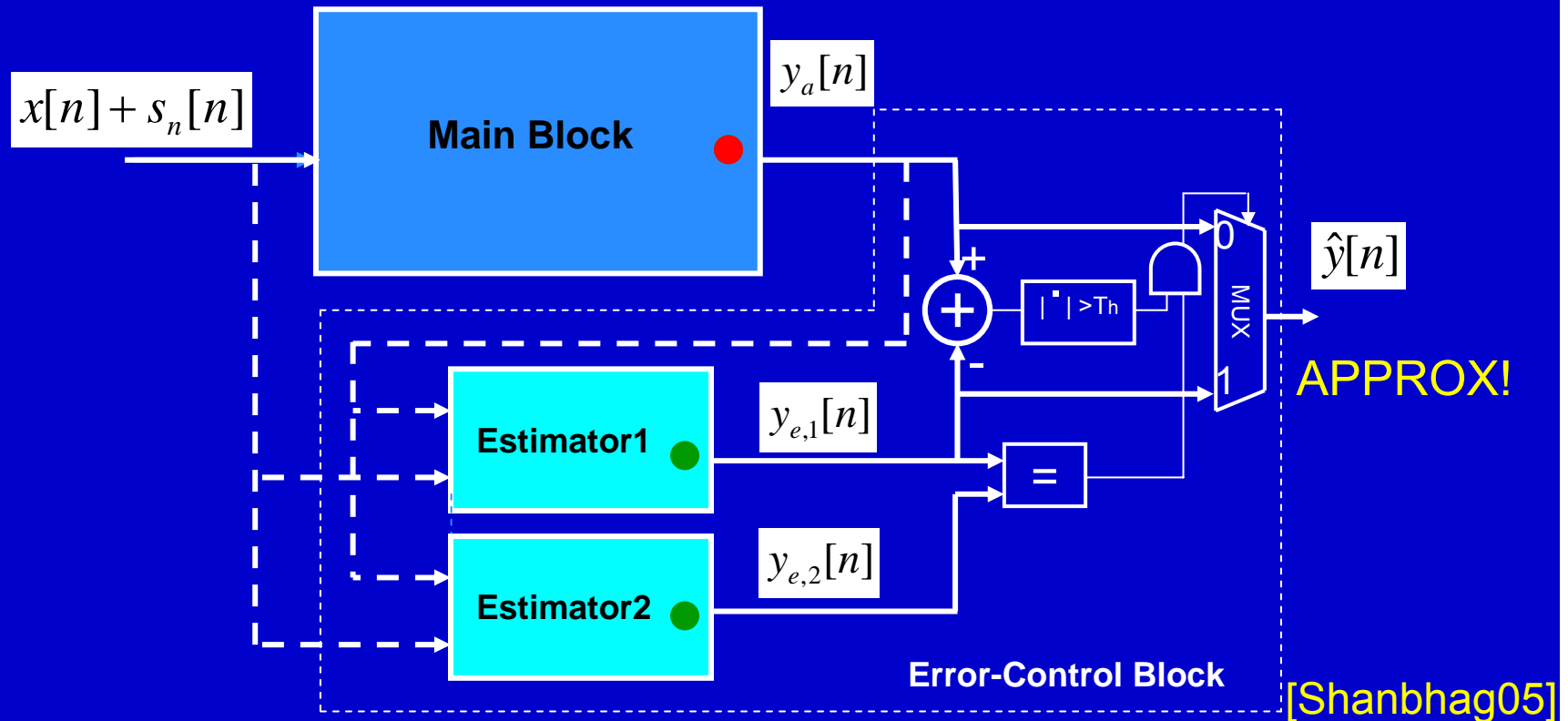


Error Blocking Characterization Results

- Flip-flop soft error rate reduction: at least 20X
- Chip-level analysis: 25% flip-flops protected
 - Selected by fault injection

Power penalty (Error resilient mode)	3 – 4.5 %
Power penalty (Economy mode)	1.6 %
Performance penalty	0 – 1.5 %
Area penalty	0.1 %
Error recovery support	Not required

Algorithmic view

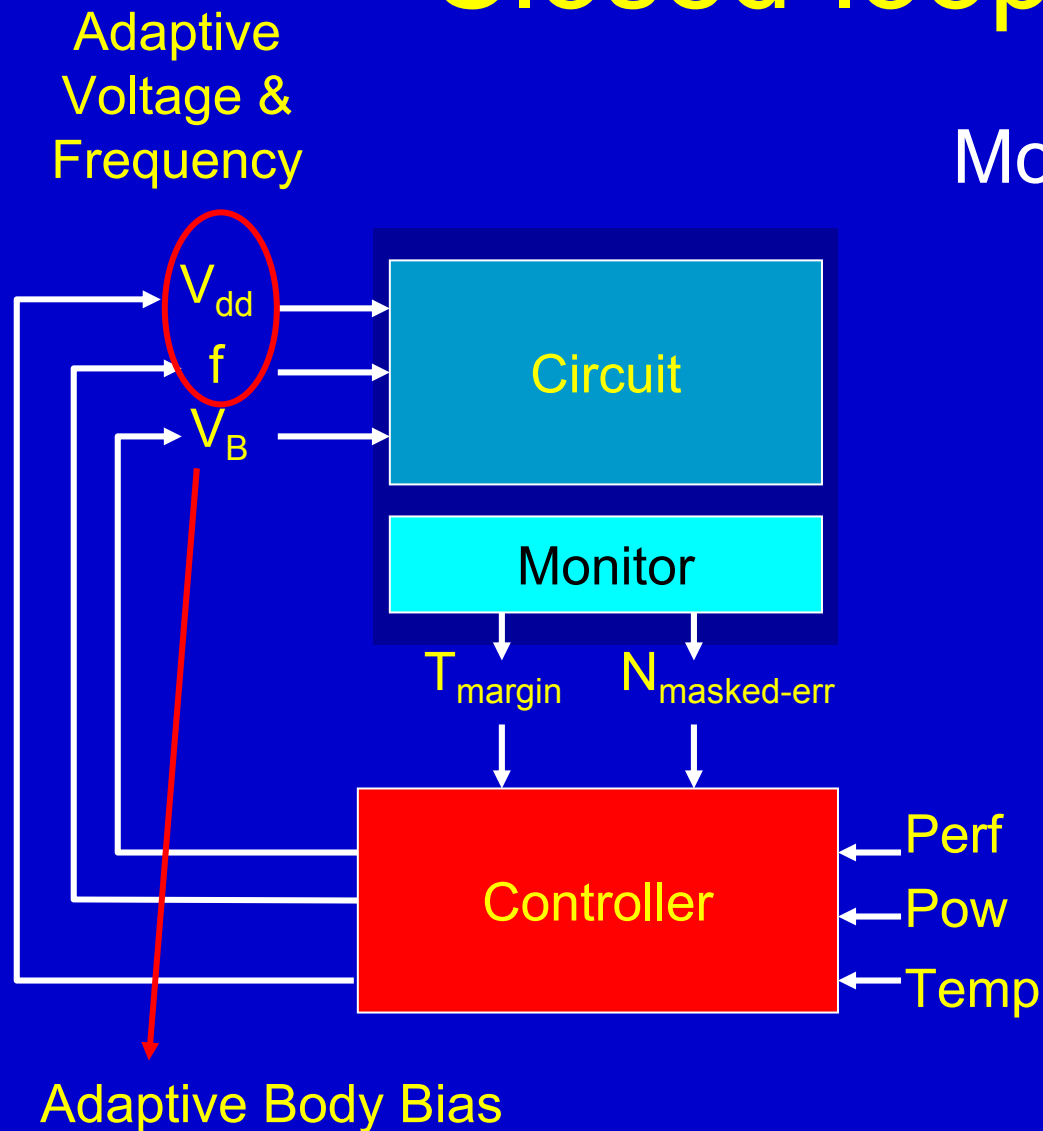


- Algorithmic error tolerance
 - Use two estimators with small approximation error T_h
 - Savings over TMR = 2.5X
- Suitable for “soft functional constraints”

Dynamic Reliability Management

- Online adaptation to enhance robustness
 - **Monitor** -sample events that correlate with failures
 - **Actuator** -control the HW operation mode
 - **Controller** -decide how to set actuators in response to monitor's data
- Effective for “soft” variations and failure mechanisms that can be effectively monitored and predicted.
- Increasingly integrated with other forms of resource management (e.g., power and quality-of-service).

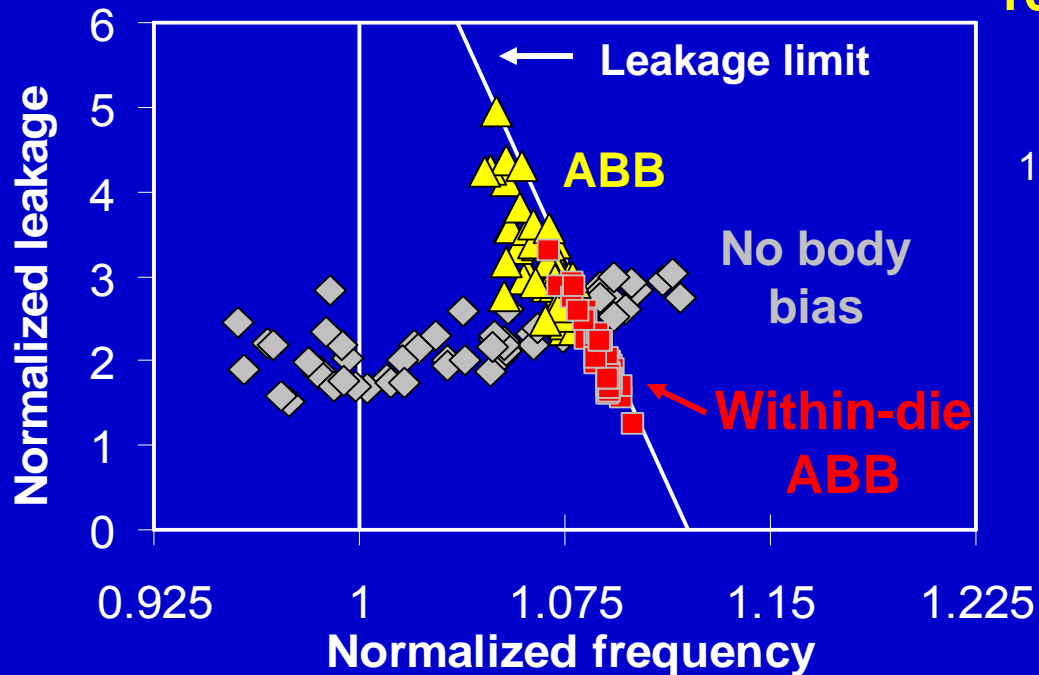
Closed-loop Model



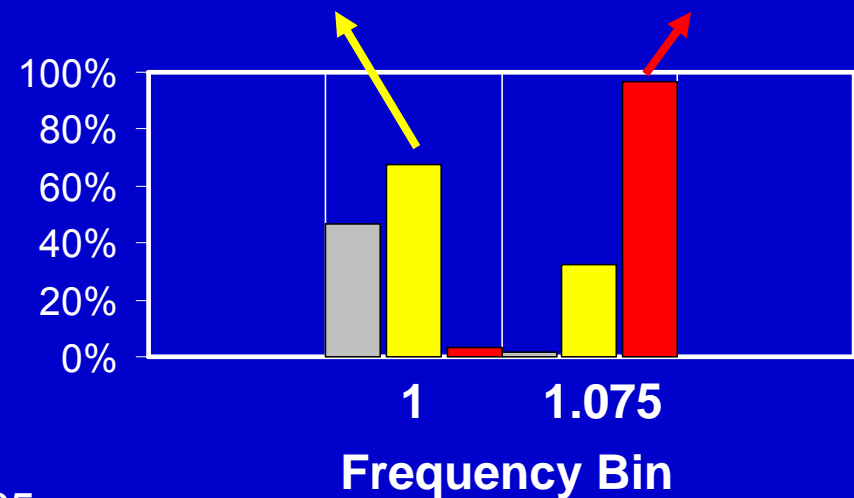
Mode of operation

- **Post-fabrication:** regulation is performed once in post-fabrication testing → easiest & lowest overhead but limited adaptation
- **Off-line:** regulation is performed periodically, while system is in “test mode”
- **Online:** monitors and controller operate in parallel with the circuit

Post-fab Adaptive Body Bias



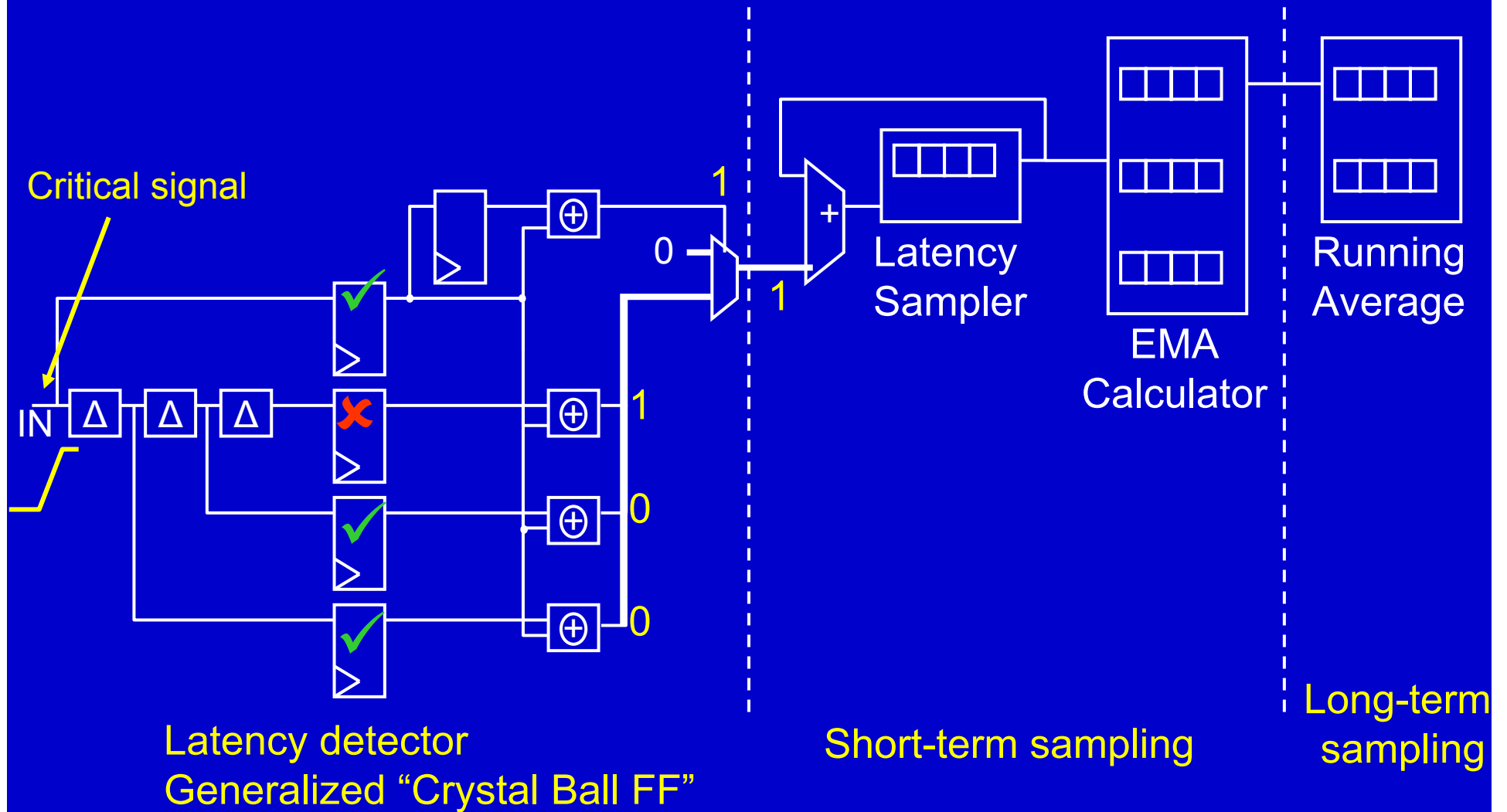
100% Performance Yield **97% Highest Frequency Bin**



[Intel]

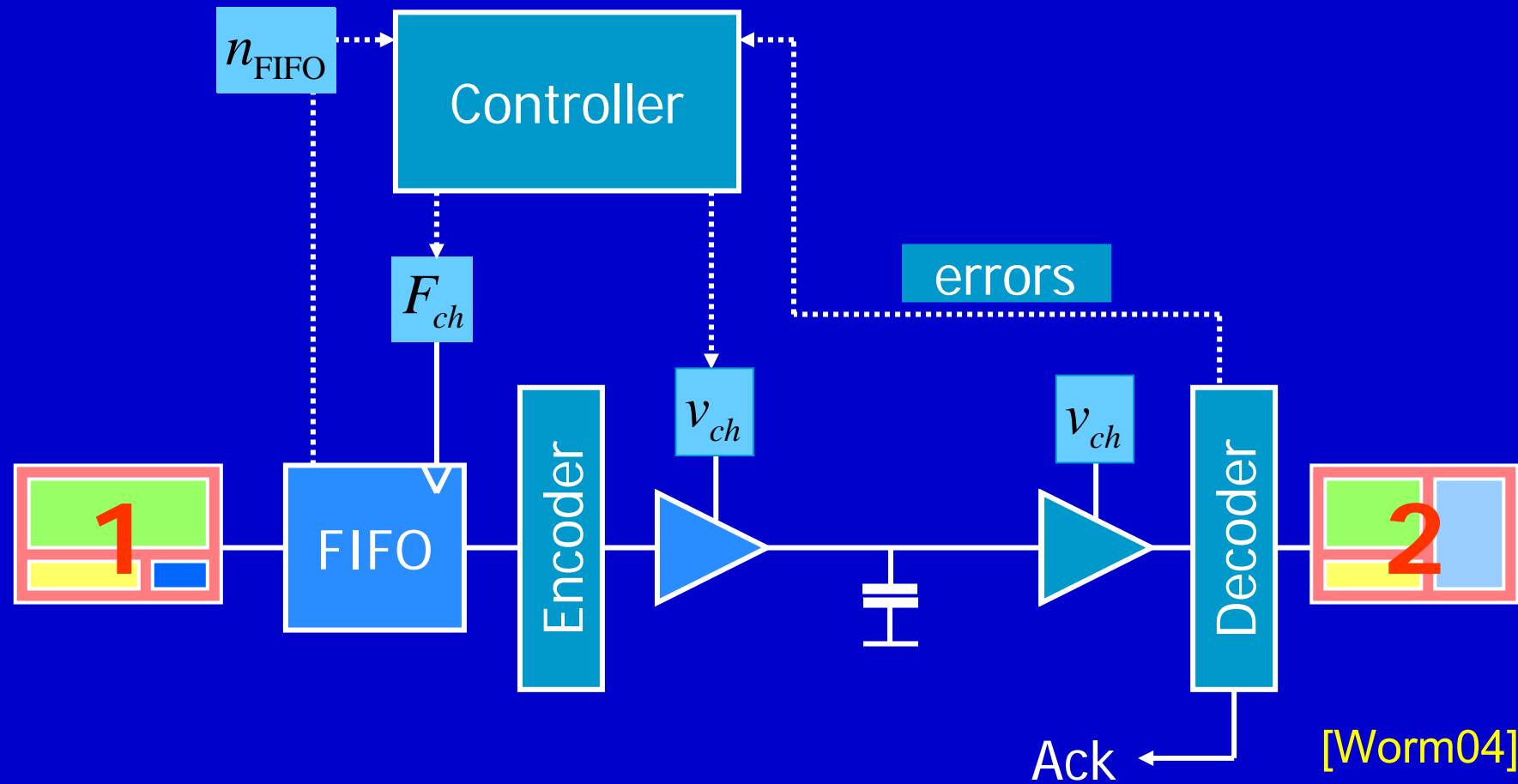
ABB reduces variation in FMAX
WID-ABB: Moves ~100% of dies into highest frequency bin

Anatomy of an Aging Monitor



[Blome06]

Adaptive low-power transmission scheme



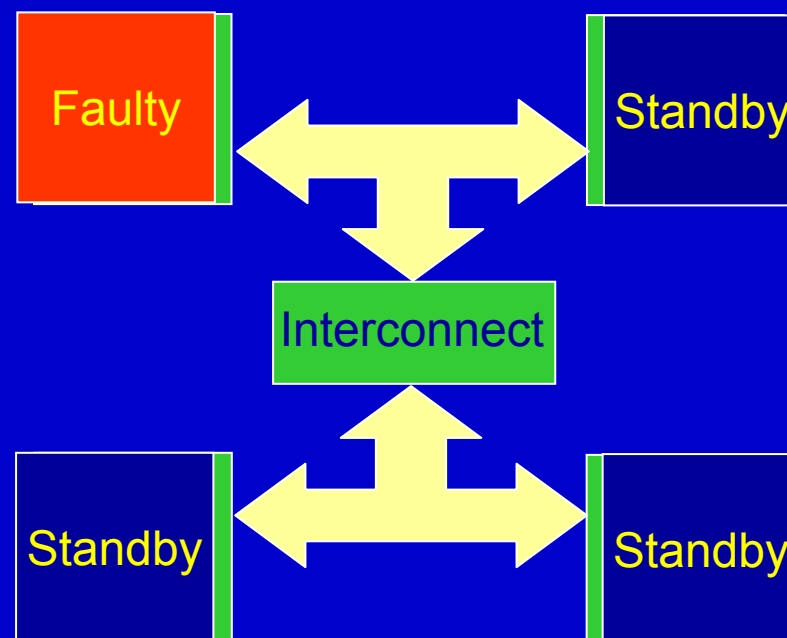
[Worm04]

Poly-core reliability

- Lower clock frequency, less aggressive design & lower dynamic power density → Reliability↑
- Much functional redundancy & freedom for allocating & scheduling workloads in a reliability-aware fashion
- Space and time redundant execution is possible
- Software controllers for a complete system view

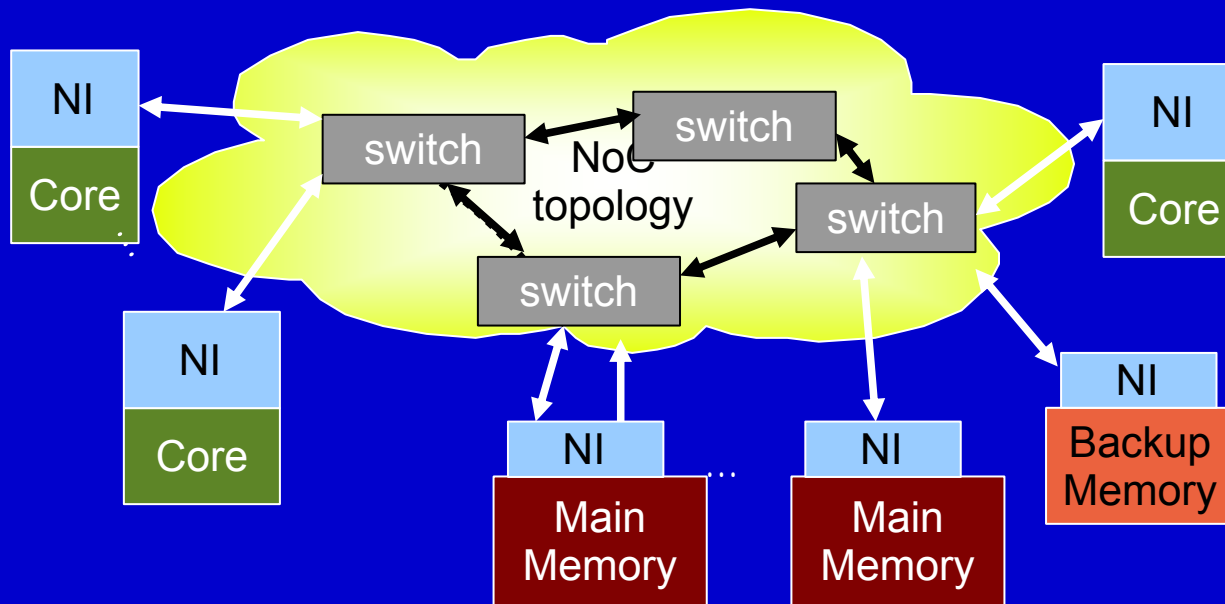
Component redundancy

- Use stand-by components to replace faulty ones
 - Provide for temporary or permanent back-up
 - Exploit power/thermal management
 - Load sharing
- A programmable and flexible interconnection is required



Memory Reliability enhancement

- Application-Level: Partition application data to **critical** Vs **non-critical**
- Use **back-up memories** for critical data
- Use **NoCs** to achieve **transparent** back-ups

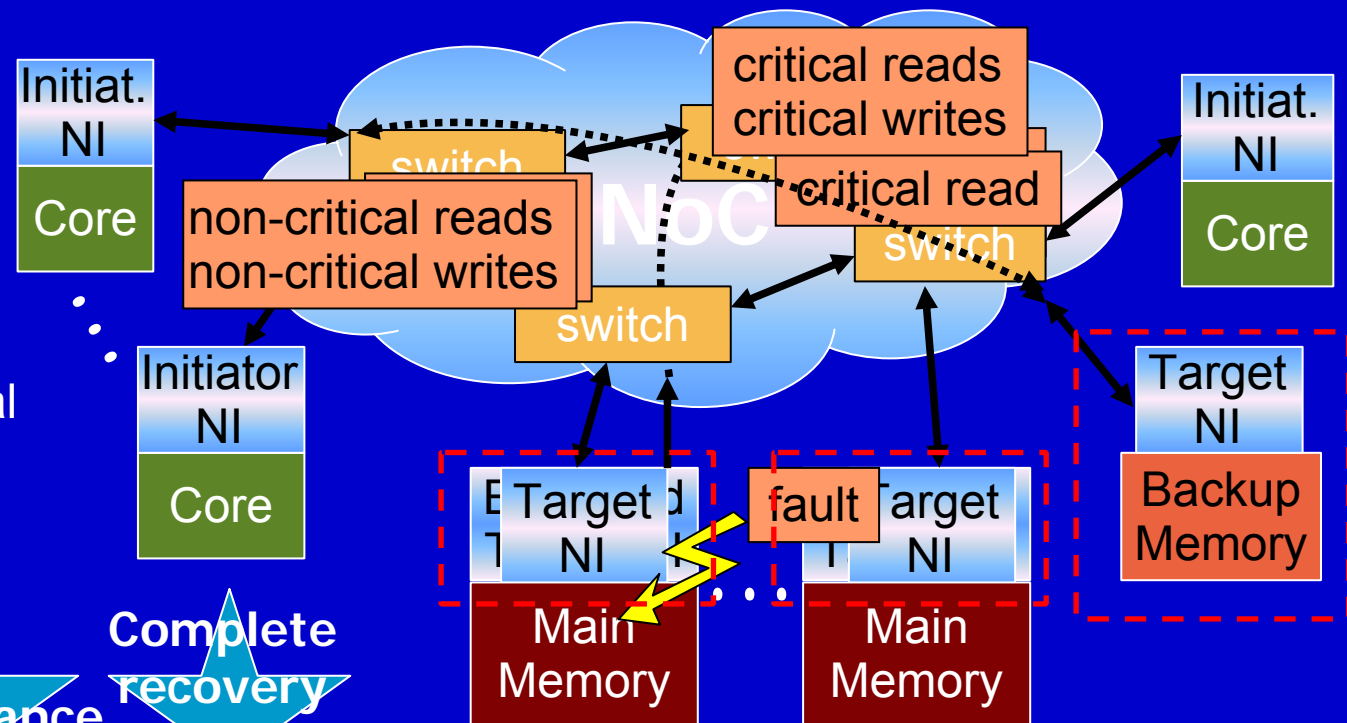


Masking unreliable memories

- Use slower (reliable) memory as backup for critical data
- Only modifications in NIs, transparent to users

- Mechanism:

- 1) Always forward critical writes
- 2) Upon faults, critical reads forwarded
- 3) Separation of critical & non-critical traffic



Negligible
area
overhead

No
performance
degradation

Complete
recovery
against
errors

Summary

- Technology robustness is decreasing
- Holistic approaches are needed to deal with increasing uncertainties in design
- Hardware-only solutions will not be sufficient
- Bio-inspired paradigms in the future?