

Reliability Challenges and Measures at Architecture Level

Munish Jassi
Technische Universität-München

REFERENCES:

- Intel Corp.
- IBM Thomas J. Watson Research Center
- University of Illinois at Urbana-Champaign

Outline

- Terminologies.
- Introduction
 - Paradigm shift from Device to System.
- Sources for unreliability:
 - Reliable functionality vs. Reliable fabrication.
- Reliability Measures:
 - Circuit level vs. Logic level techniques.
- Architecture Level Control-Techniques
 - Memory vs. Combinational logic reliability measures.
 - Future reliability measure.
- Conclusions and Take Aways

Terminologies

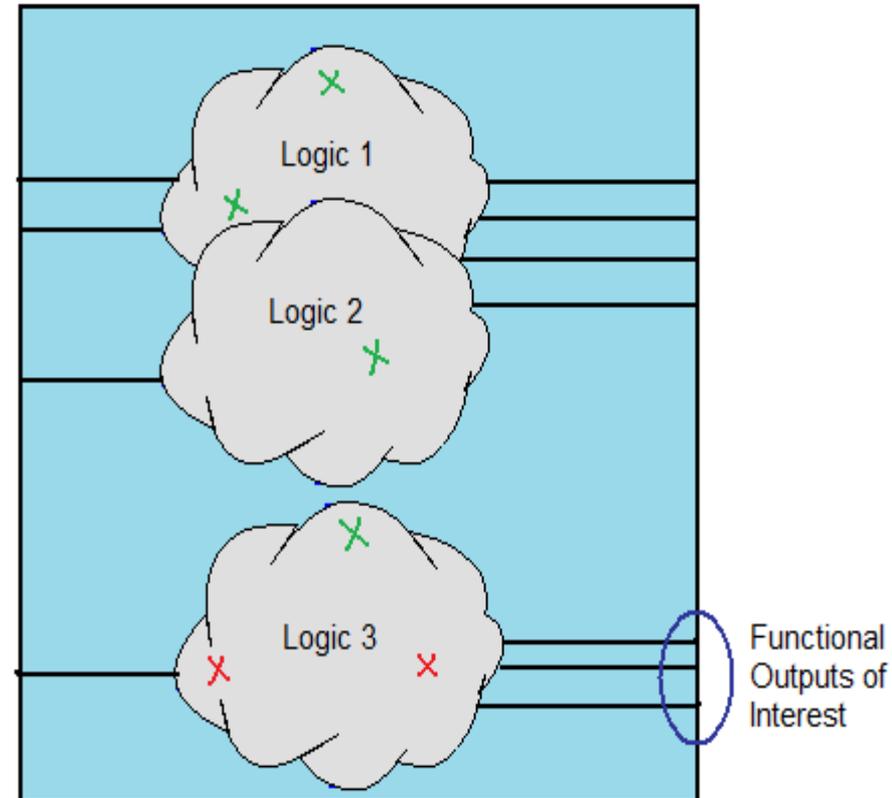
- Soft Error and SER(Soft Error Rate): Single Event Upset (SEU)
- Hard Error: Physical wearout, fabrication defects.
- Availability: Mean Time To Failure (MTTF)
- Servability: Mean Time To Repair (MTTR)
- Concurrent Error Detection (CED): Error Detection vs. Error Correction.
 - Error-Correcting Code (ECC).
 - Cyclic Redundancy Checks (CRCs).
 - Parity Check.
- Worst Case Analysis (WCA) vs. Statistical Analysis.

Introduction

- Why Reliability is becoming more and more important ?
 - Device shrinking.
 - Higher Clock Rates.
 - Logically correct implementations alone cannot ensure correct program execution.
- Importance of Architecture level Reliability Measures.
 - Studies suggest most low-level errors don't translate to errors in the application's outcome.
 - Traditional Accelerated Aging(Burn-In) Tests becoming questionable.
 - Traditional measures of WCA, are becoming over-constrained.

Introduction

- Not all Soft-errors are critical!
- Only those SE which propagate to desired outputs are of concern.
- Research results shown, between 3.7% and 10.4% of faults in sequential logic manifest as errors at the processor pins.
 - Kalbarczyk et al, IEEE trans.



Sources of Unreliability

- What's making Systems more unreliable?
 - Device Shrinkage.
 - Lower Junction capacitances.
 - Increasing dominance of parasitic effects: Subthreshold Curr.
 - Heat Flux: V-T variations.
 - Fabrication Variations: Reliable Manufacturing

System Reliability Improvement Techniques

- Circuit Level Reliability Measures
 - Forward Body Bias.
 - Transistor Sizing for Critical paths.
 - Conservative Design Practices: CMOS more robust than Dynamic Logic.
- Logic Level Reliability Measures
 - Self Checking circuits.
 - Redundant Latches/FFs.

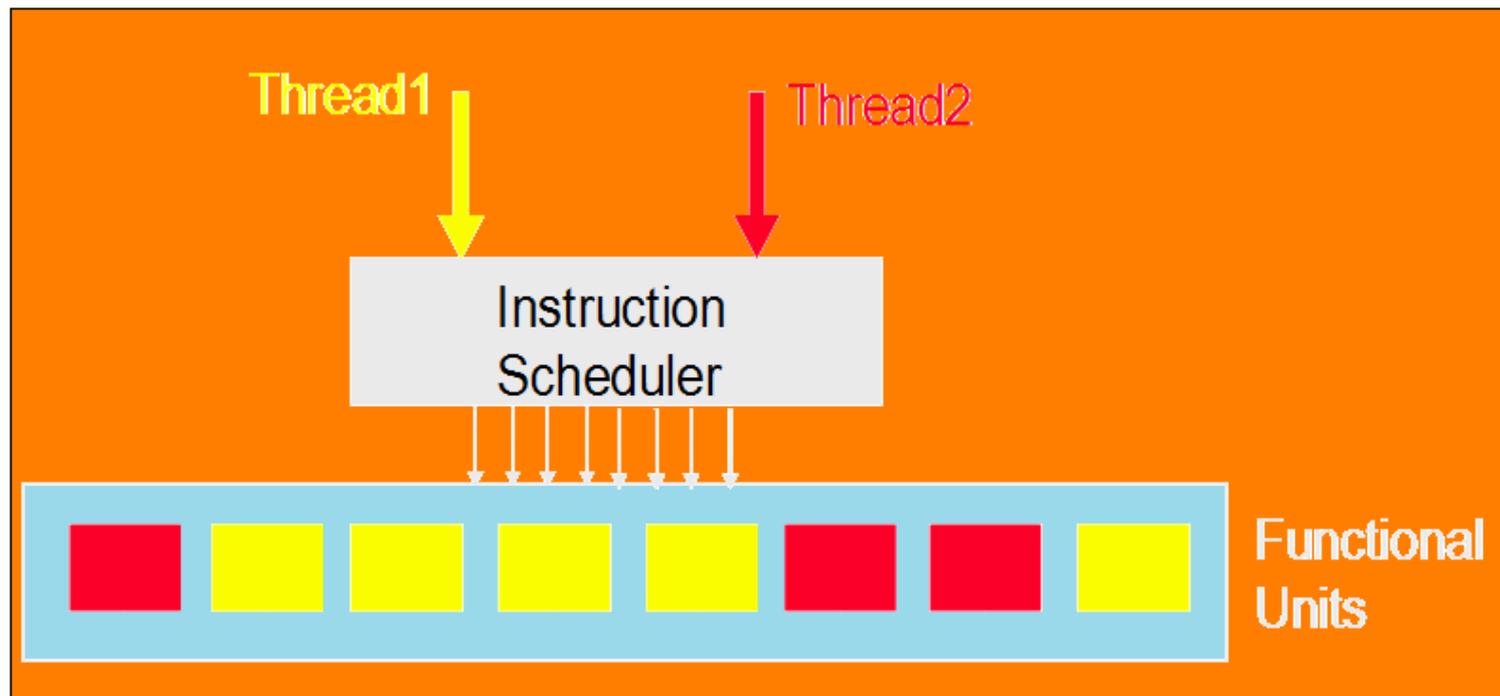
Architecture-Level Reliability Control Techniques

- Redundancy*: Information, Hardware, or Time
 - Information Redundancy
 - Error-Correcting Code (ECC): Addition Storage of Encoded Data.
 - Typically for Memories, caches and Register files.
 - Hardware Redundancy
 - Modulars: Resource Overhead, Excessive Power, Performance Hit.
 - Time Redundancy
 - Avoid Area overhead.
 - High performance overhead, high error detection latency.

**** Why Redundancy is acceptable!***

Architecture-Level Reliability Control (cont.)

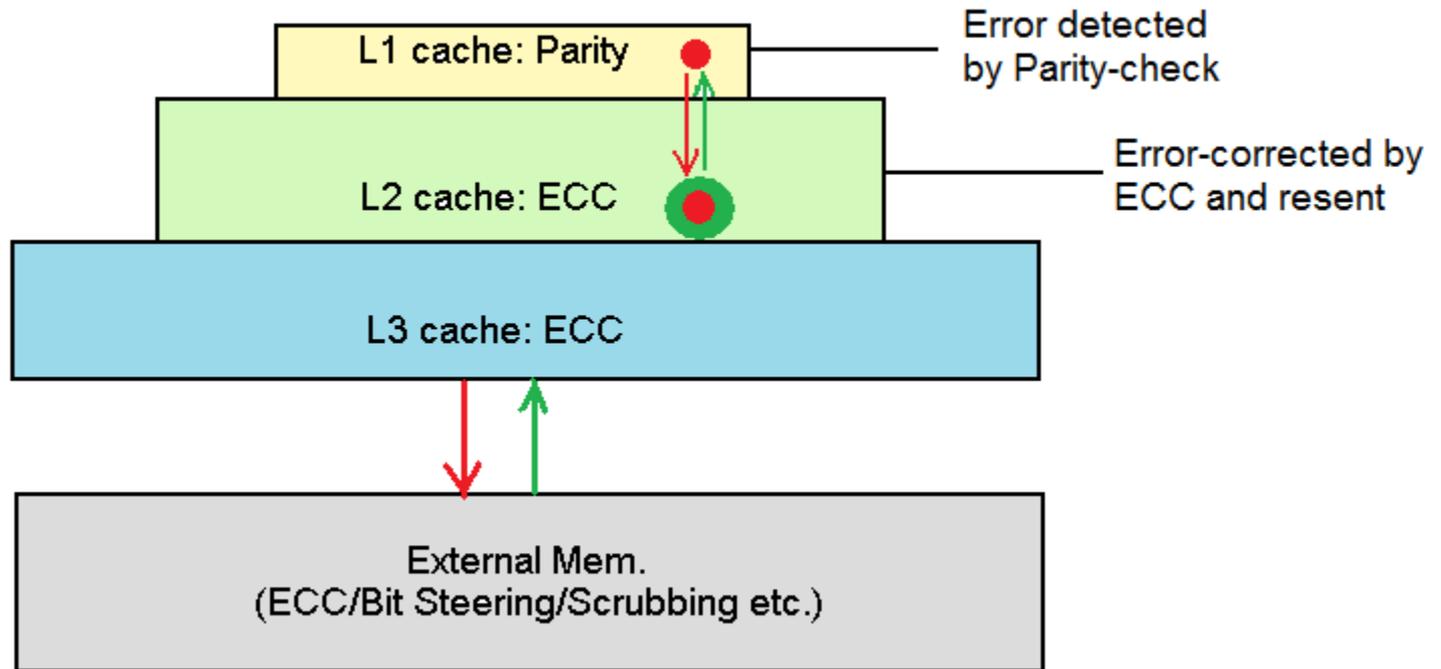
- Instruction Duplication.
- Redundant Multi-Threading (RMT).



Architecture-Level Reliability Control (cont.)

- Memory Errors Detection and Correction
 - Information Redundancy is widely used approach.
 - Optimized approach for Memory hierarchy
 - Parity check for Lower level Cache.
 - Single/Multi Error Correction for Higher-level.
 - Data bit interleaving (Bit Scattering)
 - Periodic Scrubbing: Avoid Multiple Errors.
 - Bit Steering.

Architecture-Level Reliability Control (cont.)



Architecture-Level Reliability Control (cont.)

- Memory Errors Detection and Correction.

Feature	Intel P6 Family	AMD Hammer	Intel Itanium	IBM S/390 G5	IBM Power4
Internal Regs	Parity	No Protection	No Protection	ECC	Parity
L1 Data	Parity	I-Cache : Parity I-Data: ECC	Parity	Parity; Store Buffer protected by ECC	Parity
L2 Data	ECC	ECC	8-bit ECC/ 64-bit Data Parity	ECC	ECC
L3 Data	N/A	N/A	8-bit ECC/ 64-bit Data Parity	N/A	ECC
Buses	ECC on CPU-L2 bus	No Protection	No Protection	No Protection	Databus:ECC Address&Contr ol bus:Parity

Architecture-Level Reliability Control (cont.)

- Combinational and I/O Error detection and Correction
 - Soft-error-tolerant hardened FFs.
 - For dynamic errors: Razor (DVS).
 - Protection coding on Data-paths.
 - Built-in soft-error-resilience: C-element.
 - IBM ChipKill: Multibit error correction.
- Soft Error: Checkpoint based Instruction Call back.
- Hard Error: Checkpoint based Core replacement.

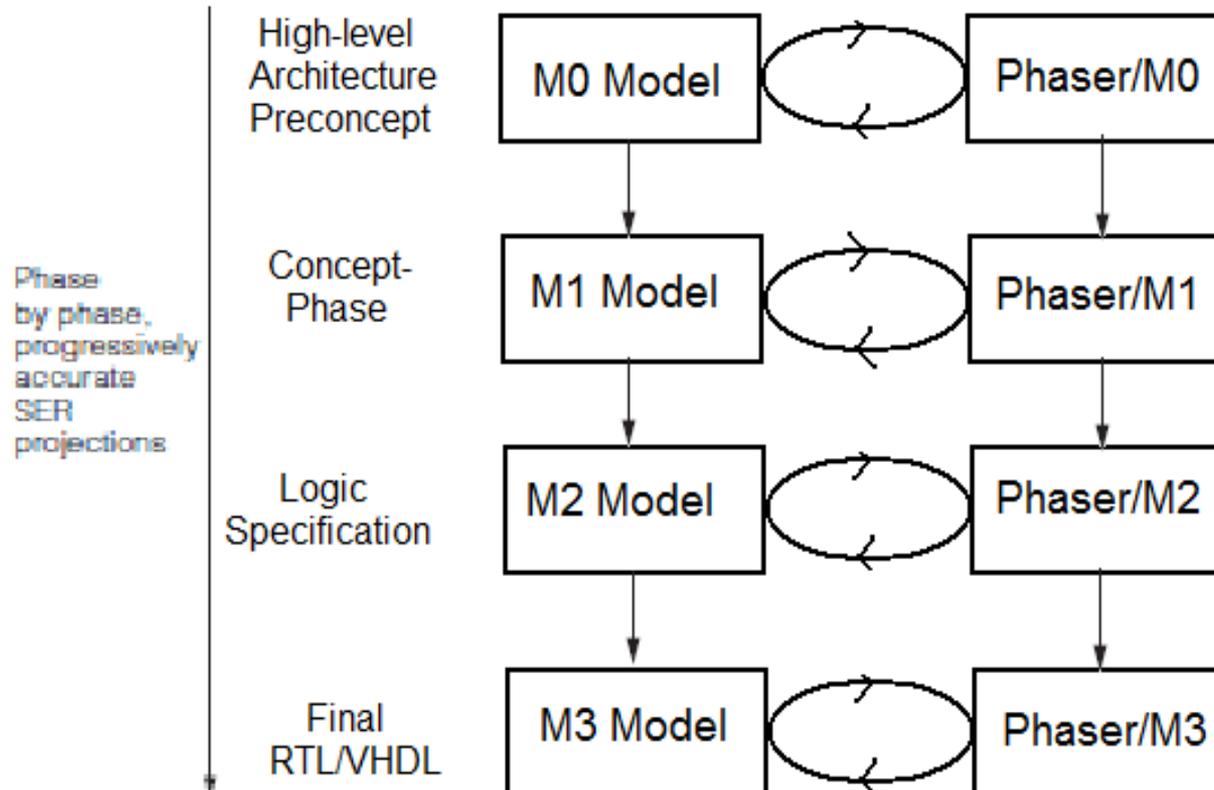
Future Reliability Measures

- Reliability as important as Performance and Power.
- Present techniques unable to handle Future complexities.
- Comprehensive Top-Down framework for designers.
- MTTF (failure rate) driven chip-design methodologies.
- Efficient tool-set for reasonable failure-rate estimations.
- Innovations at Circuit-Logic-Architecture-Software level solutions.

Future Reliability Measures: PHASER (cont.)

- PHASER: Toolset for Transient Errors.
 - SER vulnerability maps: SER Hotspot regions.
 - Iterative improvements at different abstraction levels.
 - Trade-off b/w HW recovery vs. SW error handling.
 - Intelligent adoption of appropriate error resilience approach.
 - Sub-module duplication, ECC, parity, RMT.
- STEPS:
 - Separate full-chip into sub-modules.
 - Generate SER value for sub-units.
 - Scale raw-SER with microarchitectural residency factor.
 - Microarchitectural trade-off is done to different units to achieve desired Total-SER.

Future Reliability Measures: PHASER (cont.)



Future Reliability Measures: RAMP (cont.)

- RAMP: Toolset for permanent-fault analysis.
 - Device models for different wearout mechanisms.
 - Takes in Cycle accurate application behaviour, Power and Temperature information, and chip floorplan.
 - Tool outputs FIT and MTTF values for all components of chip.
 - Designer takes decisions concerning Performance, Power, wearout reliability.

Conclusions and Take-Aways

- Reliability becoming as critical as Power and Performance for future Systems.
- Not all faults translate to System-Failure: Motivation behind Architecture level reliability Measures.
- Present Reliability Improvement techniques (Circuit-Logic-Architecture level) and Future Trends (PHASER, RAMP) were discussed.
- Radical Research efforts are being put for Future Reliability Improvements.
- EDA tools need to improve on more accurate Reliability estimations, to support Designers.

References

- [1] S. Borkar, “Designing Reliable systems from unreliable components: the challenges of transistor variability and degradation”, IEEE Computer Society, 2005.
- [2] Ravishankar K. Iyer, Nithin M. Nakka, Zbigniew T. Kalbarczyk, Subhasish Mitra, “Recent advances and new avenues in hardware-level reliability support”, IEEE Micro, Nov-Dec 2005.
- [3] Jude A. Rivers, Prabhakar Kudva, “Reliability Challenges and System Performance at the Architecture Level”, IEEE Design & Test of Computers, 2009

Danke Schön!
Спасибо!