Reliability Challenges and Measures at Architecture Level

Munish Jassi
Technische Universitat Muenchen

JASS-2011
Moscow

1. Introduction

Following Moore's Law, chip designers are pushing more and more transistors into the chips, and bringing transistor dimensions closer to the physical limits. As the chips are becoming denser and faster, the secondary effects which were negligible in the earlier technologies have started playing significant roles for the current state of the art technologies. Because of these secondary effects, chips are becoming more prone to soft and hard-errors, thus robustness of future chips is becoming more of a concern. In the beginning of VLSI era chip area was the concern for Industry and academia, by the turn of 21st century power-consumption became an add-on concern, now experts from Industry and academia are speculating that in future reliability will play as important role as power and area is playing for the current generation. There are various levels at which reliability measures can be taken: logical level, device level, and architectural level. With chips becoming more and more complex architectural level measures are becoming quite important. This article will focus on the need of architectural level reliability improvement measures, available techniques and future requirements.

2. Terminologies

Before we dig further into the challenges about reliability, here we discuss couple of terminologies which will be used though this text.

Reliable behaviour means error free functioning of a chip. There can be two categories of errors which can cause erroneous outputs, soft errors (transient error), and hard errors. Soft-errors are a one time event, system give incorrect output for limited number of iterations and then come back to the normal correct behaviour. This can occur because of reasons like incorrect capturing by the sequential elements, toggling of bits in the memory elements etc. Soft-errors are also called Single Event Upsets (SEU). Another related term is Soft Error Rate (SER), which is the rate of occurrence of Soft-errors. Hard-errors are because of physical damage occurring in the circuitry. This damage can occur because of wearout or malfunction of some hardware component.

For quantification of reliability there are two important terms, Mean Time To Failure (MTTF), and Mean Time To Recovery (MTTR). MTTF accounts for the time duration over which device is functioning as desired before its failure. MTTR is the time required to recover the system to functional state once it started malfunctioning.

In this article we will be discussing about various schemes to overcome the errors occurring in the systems for improving reliability of chips. There has already been various error correction schemes successfully deployed in other domains like wireless/wired communication, networking etc. Similar learning can be further used to correct error occurring in the chips. Schemes like Cyclic Redundancy Checks (CRC) and Parity Check are heavily used for Error Detection. Schemes like Error-Correcting Coding (ECC) are in use for error detection and correction. Similar techniques are under research for there usability in improving system reliability.

3. Sources of Unreliability

Static variability:

As the devices are shrinking, variability of device properties are increasing. First main source of the variability is random dopant fluctuations[1], channel underneath the oxide gate is doped with dopants to adjust the threshold voltage. At 1um technology the dopant count was in thousands, but at 65-32nm technologies it has dropped to few tens to less then 100 dopant atoms. So even the transistors sitting next to each other shows observable variations in their properties.

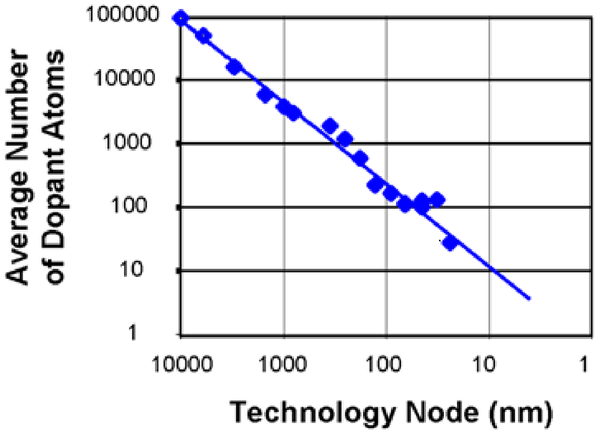


Figure 1 : Random dopant fluctuation, Intel Technology Journal [6]

Second source of variability is lithography. Beyond 0.25um technology sub-wavelength lithography is being used for patterning. Sub-wavelength lithography means wavelength of light is more then the minimum feature size for manufacturing, which leads to effects like line edge roughness and others, resulting in variations in the fabricated devices.

The effect of these static variations is visible in term of ~30% variation in operating frequencies and 5-10 times variation in leakage power which corresponds to ~50% variation in total power consumption. This variation further mean unreliable chip outcomes.

Dynamic variability:

There is another source of variation having time or context dependency. In a functioning chips there are certain regions on the chip which are more active than other regions. These regions dissipate more power and cause temperature variation. Similarly, high activity regions extract more current causing voltage drops across regions. Because of these variations transistor's speed varies from one region to another, causing potential functional failures.

Another reason is because of device shrinkage. It becomes critical for SER because shrinkage comes with decrease in junction capacitances and since alfa particles, cosmic rays are the main source of soft-error, with decreasing junction capacitance devices are becoming more and more susceptible to soft-errors.

Researchers expect ~8% increase in SER per logic state bit with each technology generation [5]. Increase in SER with increasing technology nodes is shown in figure 2. It should be noted that at 16nm SER will be more then 100 times than that at 180nm.

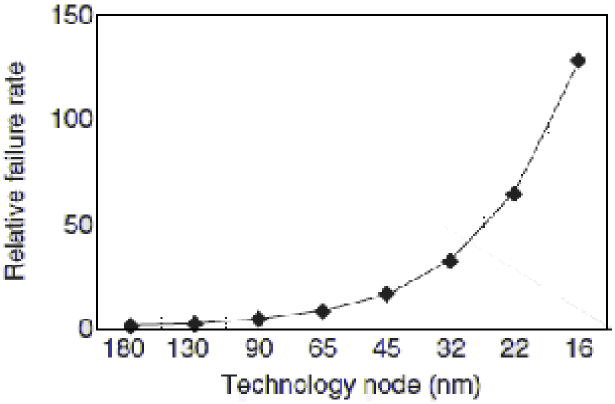


Figure 2: SER of a chip is increasing by 8% per technology generation.

Also, as the chips are becoming more faster, dynamic effects like glitches and signal settling time are also becoming important. With decreasing timeperiods, lesser time is available for signal settling after transition. This can lead to incorrect capturing of logical signal value.

4. Architecture-level Reliability

There are multiple reasons to look at unreliability problem from architecture level perspective. First and the main reason being, chips are becoming bigger and more complex. Looking at the complexities of current state of art chipsets it becomes almost impossible to find the solution at device or gate level. Not all the soft-error which occur at device or gate level translate to errors at the functional outputs. As described in figure 3 errors which falls in the fan-in cone of desired outputs ports are of interest for us. Research shows that only between 3.7% to 10.4% of faults in sequential logic translate to errors at processor pins [4].

Second reason being, traditionally lifetime estimation of the chip and filtering faulty chips is based upon accelerated burn-in testing. But because gate leakage current increases exponential with increase in temperature, leakage power becomes exponentially high. With higher leakage power playing part (which in reality will have negligible role at actual operating voltages and temperatures), modelling of device degradation based upon biasing current (which is the primary source of device degradation) becomes difficult leading to incorrect lifetime estimations, making burn-in test obsolete.

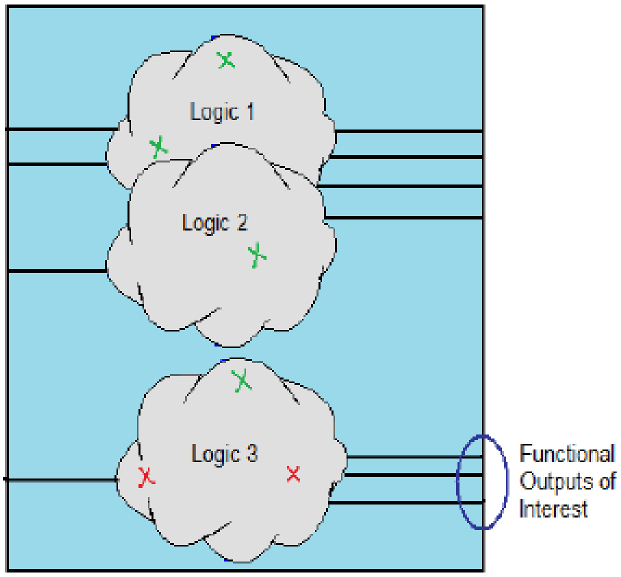


Figure 3: Not all the faults at sequential logic translate to error at output ports.

Another reason is that till now WCA (worst case analysis) is used for taking care of reliability related concerns. In this worst case scenarios are considered and extra timing margins are put during timing closure. But for 32nm and below, variations across multiple corners have exponentially increased, so excessive design margins would overfix the design. Also it is becoming difficult to do timing fixes simultaneously for all the corners.

5. System-reliability improvement techniques

There are three main levels at which reliability measures are taken care: circuit level, logic level and architecture level [2].

Circuit-level Reliability measures:

One way of improving reliability at circuit level is by using forward-body bias. Forward-body bias increases the junction capacitance and as discussed earlier increasing the junction cap improves the logic value retain capability of logic elements.

Another important approach is doing transistor hardening. Of all the timing paths in the design, only the variation in the timing critical paths directly effect the operating frequency of the chip. If delay in non-critical paths increases because of variations it is highly unlikely that it become critical and effect the operating frequency. So, by making gates which belong to the critical paths more variation resistant (transistor hardening) one can improve the overall reliability of the system. One approach for doing transistor hardening is oversizing the transistors.

Also, conservative design practices like using cmos based approach in place of dynamic logic improves the system reliability.

Logic-level reliability measures:

At logic-level, approaches like self-checking circuits which contains additional logic to detect and correct errors can be used. One such approach is by using redundant

FF/latches. In this rather then using only one latch to capture output of sequential logic, multiple latches capture the same output and then the captured value is compared for the correctness of the data. Here C-elements can be used for comparison. C-element has multiple input ports and single output. If only all inputs have identical values, input value is transferred to the output, otherwise the previous output value is retained.

6. Architecture-level reliability measures

Redundancy:

Redundancy is the most usable form of reliability measures in current generation of chips. There are four types of redundancies which find there applicability in sequential logic, memories or combinational elements: Information redundancy, hardware redundancy, time redundancy, and space redundancy.

Information redundancy is about storing additional redundancy information in the form of encoded data. This additional information is further used for error detection and correction schemes using ECC or parity check. This is mainly used for correcting soft-errors in memories, caches, and register files.

Hardware redundancy (HR) deals with hard-errors. In this measure, logic which is more prone to hard-errors are provided with addition duplicate hardware which take-over the original hardware once a hard-error is detected in the original hardware. Another way HR being utilized is using multiple hardwares (duplicated) that runs in parallel and the output is selected via majority voting (Modulars). HR comes up with lot of resource overhead, excessive power, and performance hit. This approach is mainly suitable for logical blocks.

Time redundancy is one way to overcome resource overhead which comes with HR. In this one input is executed multiple times and the outputs are compared. Again, this comes with penalty of high performance overhead and high error detection latency.

Last type of redundancy is spatial redundancy which utilized the flexibility provided by compilers. Here one approach is instruction duplication, in which compiler issue duplicated instructions and the outputs are compared for the correctness. Another state of art approach is using Redundant multithreading (RMT). In this duplicated instructions are issues in different threads and executed. Processing of RMT is shown in figure 4, in this Thread1 and Thread2 consist of identical instruction which are further scheduled by an instruction scheduler.

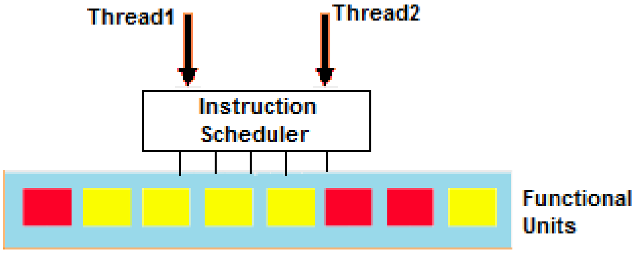


Figure 4: Redundant multithreading (RMT), Thread1 & Thread2 contains identical instructions

In the current designs, memories are much more densely packed then the logic, and with further device shrinkage transistor densities in memories is further increasing. This make the memories much more susceptible to errors. There has been different approaches used in industry to provide protection for memories and Logic.

For memories, information redundancy has been the widely accepted approach. Periodic scrubbing is also used to avoid multiple errors. Bit steering is another hardware based approach where memories are provided with additional bitlines. In this if multiple errors are found on single bit line, then memory content of that bit-line is transferred to an additional bitline with corresponding address mapping.

The hierarchical nature of caches in modern processors is exploited by utilizing different schemes at various cache levels. If the cache structure is like the data at higher level cache is also available at lower level cache, then at higher level cache schemes with lesser overhead and at lower level more complicated schemes can be used. One approach is to use error detection schemes at L1 cache, and error detection and correction scheme at L2 cache, so once an error is detected at higher level, the correct data can be retrieved from the lower cache level (figure 5).

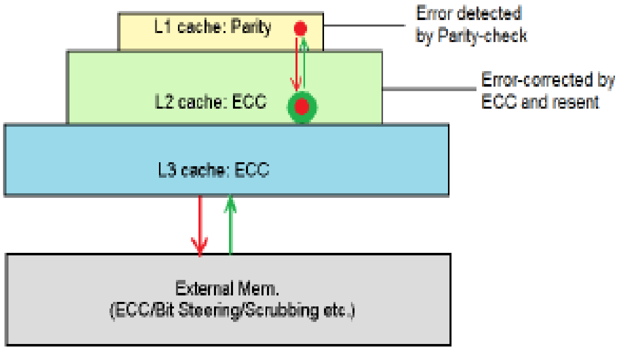
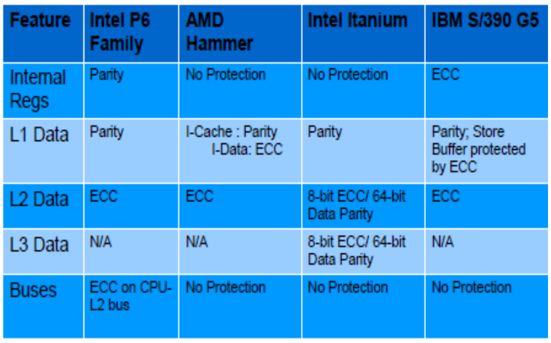

L1 cache: Parity
L2 cache: ECC
L3 cache: ECC
External Mem.
(ECC/Bit Steering/Scrubbing etc.)
Error detected by Parity-check
Error-corrected by ECC and resent

Figure 5: Different schemes at various cache levels



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

Figure 6: Error-correction schemes at different cache levels in some modern processors

For combinational logic there are various approaches being used, hardened FF is one of these approaches. For dynamic errors, research is going on approaches like Razor FFs. The idea behind Razor FF is to dynamically control the supply voltage, lowering the supply voltages increases the errors, but those errors are recovered by error correction logic. Razor FF provide a trade-off between power and errors.

Another modern approach focuses on multibit error-correction. IBM found that multibit error accounts to significant contribution towards device failures. They have developed a technique to detect and correct multibit errors, unlike ECC which is a single error detection and correction scheme. When a single error is detected, an Instruction call back occurs at a checkpoint. If error is detected multiple times, then it is treated as hard-error and hardware replacement (duplicated hardware) occurs.

7. Future Reliability Measures

As we have seen till now, reliability is becoming as critical as power or area. It becomes important that various schemes get available which could be utilized at different stages during chip design flow. From designer's perspective, they need some automation tools that can visually show problematic regions in the chip, based upon which designer can use appropriate corrective mechanism (redundancy, ECC, parity etc.). Present technologies are not in place to handle the complex reliability requirements of future chipsets, large research effort from EDA industry and academia is required to develop comprehensive Top-down frameworks for designers with efficient toolsets for reasonable failure-rate estimations.

PHASER: Toolset for Transient Errors [3]

PHASER tool is one of the first steps towards providing automation for reliability challenges. It provides graphical platform for designers to look at the vulnerable regions in the chip and adopt corrective actions. Tool provide iterative approach for resilience against errors at different abstraction levels (figure 7). It gives the designer option of choosing among the various error resilience approaches eg. Sub-module duplication, ECC, parity, RMT. With this designer have flexibility to either go for S/W or H/W based error handling, hence can optimize for resource utilization.

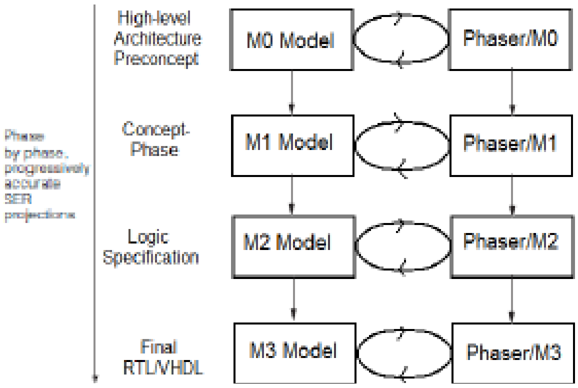


Figure 7: PHASER tool, provides iterative measures at various abstraction levels

RAMP: Toolset for hard-fault analysis

RAMP is a tool meant for permanent-fault analysis. It models the different wearout mechanisms and does the wearout profiling for chip based upon these models. RAMP takes in cycle accurate application behaviour, power and temperature information, and chip floorplan information as inputs. Tools gives out FIT and MTTF values for various components of the chips as output, based upon which designer can take decisions concerning Performance, Power and wearout reliability.

8. Conclusions

This article reviews the recent trends in the problem of unreliability of modern and further chips. We discussed why reliability is becoming more of a concern as technology is moving towards higher generations. We discussed why architectural level measures are required to tackle reliability challenges for future chips. The main reason for going towards architectural level measures is that only very few proportion of all the faults occurring at sequential logic level translate to errors at the functional outputs. Eventhough there are various solutions available at device, logical and architecture level, lot of effort is still needed to handle the reliability issues for bigger and more complex future chips. EDA efforts are required to help designers visualize the hotspots regions on the chip and take appropriate corrective actions.

9. 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

[4] G.P. Saggese et al., An Experimental Study of Transient Faults in Microprocessors, tech. report, Center for Reliable and High-Performance Computing, Univ. of Illinois, Urbana-Champaign, 2005.

[5] P. Hazucha et al., “Neutron Soft Error Rate Measurements in a 90-nm CMOS Process and Scaling Trends in SRAM from 0.25-µs to 90-nm Generation,” Proc. Int’l Electron Devices Mtg. (IEDM 2003), IEEE Press, pp. 21.5.1-21.5.4.

[6] Kelin J. Kuhn et al "Critical sources of variation in the 45nm generation", Intel Technology Journal, June, 2008