Fault Injection Verification of IBM POWER6 Soft Error Resilience

P. Kudva, Jeffrey W. Kellington, Pia N. Sanda, Ryan McBeth, John Schumann, and Ron Kalla

Abstract— **Full chip statistical fault injection has been performed on a hardware emulated POWER6 platform. These results were validated against proton beam injection results. The fault isolation, error recovery and error logging capabilities of the POWER6 enabled measurement of the architectural derating factors. The proportion of errors as well as derating factors matched well between the simulated and particle beam injection cases. The simulated and particle beam induced fault injection methods are compared and contrasted. SFI and particle beam fault injection methods complement each other to provide an overall understanding of the microarchitecture error resilience.**

Keywords: reliability, SER, analysis tools, fault injection, SFI

I. INTRODUCTION

While it is well known that soft errors in logic are a concern in modern VLSI circuits [1], systematic accounting of the impact of flips at the microarchitecture level has been difficult to achieve. SER is said to be “derated” from the raw technology contributions because not all bit flips manifest into application errors. Whether or not an error propagates to the application level depends upon the location of the bit flip in a latch, logic gate, or memory cell, the time that the flip occurred, and what the chip was doing at the time of the flip. To obtain an accurate SER estimate for a given design, it is critical to accurately represent derating. Past research works have developed techniques to estimate derating factors at different levels of abstraction of the processor chip [2-5]. Fault injection techniques have been used extensively to analyze and evaluate the dependability of computer systems [13-15]. Presented here is a methodology to assess microarchitecture derating based upon simulated and particle-induced fault injection.

The method uses a whole-chip statistical fault injection (SFI) methodology and is validated by particle-induced fault injection. The studies were performed on the POWER6 dual core microprocessor built in IBM’s 65nm SOI technology. The simulated injections are performed on an Awan [6] hardware accelerator using the POWER6 RTL gate-level model. All of the functional latches in the cores were accessible by SFI. All of the functional latches on the chip were accessible by SFI. In the case of particle-induced fault injection, all of the chip, including latches, arrays, and combinational logic are accessible to being flipped. We are able to extract the errors due to upsets in the logic because the POWER6 error detection and recovery features enabled this distinction. Once validated, the methodology utilizing the SFI approach can target local effects, checker masking, and numerous experiments not possible in a particle beam environment.

In the following sections we will first describe SFI, then we will describe the proton beam experiment used for validation. We will then compare the results of the two methods. Finally, we will describe a methodology which uses particle beam testing combined with SFI to characterize SER resiliency at the microarchitecture level.

II. STATISTICAL FAULT INJECTION WITH HARDWARE ACCELERATOR

The environment for fault injection is based on hardware accelerated simulation of the entire chip. An RTL model of the system of interest, in this case a POWER6, is synthesized and loaded onto a hardware accelerator. The accelerator then behaves as though it were a hardware chip that implemented the processor model although operating at a much lower frequency.

A programmable acceleration engine which has been used extensively for performance analysis of IBM systems is Awan [6]. Awan consists of a large number of programmable Boolean function processors with a highly optimized interconnection network. The simulation speed of such hardware can be orders of magnitude faster than software based simulation. A chip processor model say POWER6 is loaded onto the accelerator and applications and vectors are run directly on the programmed hardware which will behave like a POWER6 in a cycle based simulation mode. The code and testing will run significantly faster than simulation but as expected slower than running on the actual POWER6 system.

In our experiments, latches were randomly selected for injection among all the latches in the processor core. Faults were injected at selected locations in the model using a communication interface to the simulation acceleration hardware. The fault may exist for the duration of a cycle (toggle mode) or for a larger number of cycles (sticky mode). The effect of the fault is evaluated by checking the system/processor status registers which flag errors such as checkstops, recoveries and machine errors. The effects of the faults that are detected by the Architectural Verification Program (AVP) and are not visible to the machine (such as in machine errors) are observed in special registers set by the AVP. Faults are injected in a given cycle and then clocked for a fixed number of cycles (500,000) to ensure that all possible effects of the fault including recoveries and any possible silent data corruption, if any, have been identified and serviced.

III. PROTON IRRADIATION EXPERIMENTS

The most accurate way to probe the effects of SER at the system level is to perform accelerated testing, and includes timing derating and all flips including upsets in combinational logic which are ignored by the SFI.

The 150 MeV proton beam completely and uniformly illuminated the POWER6 microprocessor chip, verified by proton radiograph. The system was capable of running the full instruction set out of cache memory and dynamic testing was performed at speed. The fluence was $8.2\text{x}10^7$ protons/cm^2/MU, where one MU represents a “monitor unit”. In one MU, an average dose of $8.2 \text{ x } 10^7$ protons would have struck a 1cm^2 sample.

An architectural verification program (AVP) was used in the test which logged each error detected on the chip with the exception of the level 2 caches (L2). The POWER6 contains two ECC protected 4MB L2 caches and we did not take the time to log the single bit upsets in these arrays to allow more time to study the strikes in the cores. All errors were logged by fault isolation bit (FIR) and all recovery actions (several thousand in the cores alone) were reported. In this way, a full account of errors by type and action resulted. With the knowledge of the latch flip rate by static calibration, the ratio of system level events to the static flips which would have occurred in the test interval measures the effects of derating. In addition to recovered and checkstop events, the AVP was designed to detect incorrect architected state in the microprocessor cores. For this study, AVP detected incorrect machine state errors are categorized as Silent Data Corruption (SDC) [5].

In these measurements the L3 cache was disabled so the system would not be spending resources recovering from SEUs in the L3. Functional testing was done without instructions or data going out to the L3.

IV. DERATING CLASSIFICATION AND BEAM RESULTS

Once a flip appears in a latch, the effect might vanish as the AVP progresses as shown by vector (a) in Figure 1. It may also be corrected (b), it may result in a system checkstop (c), or it may result in an incorrect architected state (d). Some of the inject faults may vanish. The percentages for (a) through (d) are the actual percentages measured by proton testing. If the architected state is corrupted, this condition can be further identified as those that 1) do not impact the application, 2) are detected by software and 3) Cause silent data corruption (SDC).

The soft error derating factor is defined as the ratio of flips taken to the number of events impacting application result. Derating may be further classified as logic derating and timing derating.

The hardware acceleration based simulation methodology is primarily intended to evaluate the ‘logic derating’ of the machine. Logic derating is the probability that a soft-error upset in the design will impact the behavior of the machine. Logic derating is dependent primarily on the microarchitecture and the application.

The logic derating of a chip needs to be estimated in order to accurately determine the vulnerability of the chip. The logic derating in our case is further classified according to their impact on the behavior of the chip. In particular, using each of our methodologies, we are interested in knowing the percentage of latch flips that result in effects in each of the categories for different portions of the chip.

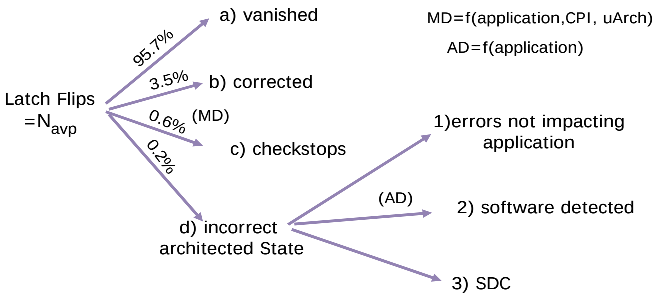

Figure 1. Classification of derating terms

We next discuss the SFI simulations that mimic the beam experiment which show that the hardware acceleration based simulation fault injection methodology compares well with the radiation approach. We will then describe how the methodology may be used to perform targeted injections and to identify portions of the design where robustness may be improved.

V. SFI RESULTS

Random fault injection was performed within the model on the accelerator while running the same AVP that was used in the proton radiation experiments. Bit flips were introduced in randomly selected latches across the chip. In one experiment, bit flips were introduced at regular time intervals during the application run. Table I shows the SFI and beam results. The beam experiment was conducted over two days, flipping ~1750 latches (determined by static latch calibration). The beam flux had to be throttled down to (i) allow SRAM recoveries including the 8MB’s of L2 to complete and (ii) allow logging of core errors including those occurring in the L1 caches. The SFI on the other hand targeted just the latches (and not the SRAMs) so many more cycles and many more latch flips could be simulated than the actual beam measurement.

The Table shows that the SFI measurements closely matched the results of the actual proton irradiation of the chip. The vanished category represents those injections that have no effect on the AVP test. For SFI, we know how many latches were injected. For the beam experiment, we know the latch flip rate from static testing and compute the number of flips that would have occurred in the time interval of the test. 96.19% of the SFI injections vanished, and 95.6% were found to vanish according to the beam measurements. The 0.6% difference could be accounted for by inaccuracies in the cycle simulator, as the Awan simulation does not accurately reflect timing derating and effects of SER strikes on combinational logic. The recovery events also matched very well (3.6% for SFI vs. 3.5% for beam). Checkpoint and incorrect architected state are also close. Since the primary use of the SFI methodology is to understand the relative sensitivities of the chip from a microarchitectural derating standpoint, the timing derating effects are ignored in our microarchitecture evaluation experiments. We are primarily looking for a strong correlation the proton irradiation experiments and the SFI approach. The very small difference indicates very good agreement between measurement and simulation and indicates that ignoring combinational logic strikes for our 65nm technology is warranted.

	Fault	**Proton**
Total Flips	28815	1748
a) Vanished	96.19%	95.90%
b) Corrected	3.58%	3.50%
c) Checkstop	0.93%	0.60%
d) Bad Arch. State	0.37%	0.22%
1) No Impact	0.08%	0.05%
2) SW Detected	0.03%	0.00%

Table 1. Comparison between Proton Irradiation and Fault Injection Experiments

VI. FUTURE WORK AND PROPOSED METHODOLOGY

Now that the full chip hardware emulation SFI method for POWER6 has been validated against POWER6 beam experiments, a wealth of variations can be simulated to obtain details of the interaction of latch flips to the microarchitecture and workloads. Targeted injections are being performed on the various macros of the chip, as well as on the various types of latches. Error detection and recovery modes can also be adaptively adjusted to study the effectiveness of the error handling. Such experiments will be valuable in the development system SER models for next generation computing systems. Additional results from this further work will be available by the conference publication date.

VII. RELATED WORK

Derating has been estimated using fault injection. [2,3,5]. Wang et. al. uses both random latch sampling across the simulated chip and time sampling across the application run was performed. Nguyen et. al. used fault injection on Intel processor models [3] and has shown the relative sensitivities of design blocks.

Particle beam testing has been reported already on microprocessors [10, 11]. Constantinescu reported on system crashes for an Intel microprocessor running applications under Windows OS irradiated by a neutron beam. Cakici et. al. reported proton irradiation studies of POWER5 microprocessor which was a precursor study used to develop the unique logging and methodology used here.

No known work exists that has validated a whole chip extensive hardware emulator based SFI model with accelerated testing.

VIII. SUMMARY AND CONCLUSIONS

We have shown a full chip statistical fault injection (SFI) methodology based on a hardware emulated POWER6 microprocessor and validated the method against actual proton beam accelerated testing results. For latch flips, the proportion of vanished, checkstops, and incorrect architected state were delineated and match well between SFI and beam-induced fault injection. While SFI does not replace beam testing, once validated, SFI does far more targeted "experiments" than possible in an actual irradiation environment.

The methodology presented here can be used to perform fault modeling as well as the analysis of the effects of such faults both at the logic circuit level as well as the system level. We hope to use this tool to derive higher level models and abstractions of designs for reliability and soft error sensitivity.

ACKNOWLEDGMENTS

The authors would like to thank Pradip Bose for his support. Sertac Cakici, Jared Day, Scott Swaney and Kenneth Wright are gratefully acknowledged for their pioneering efforts on POWER5 beam experiments. We would like to thank Wolfgang Roesner for the very valuable comments and suggestions for changes in the paper.

REFERENCES

[1] 2nd Workshop on System Effects of Logic Soft Errors, April 2006, www.selse.org.

[2] Nicholas J. Wang, Justin Quek, Todd M. Rafacz, and Sanjay Patel, "Characterizing the Effects of Transient Faults on a High-Performance Processor Pipeline," Proc Intl. Conf. Dependable Systems and Networks, pp. 61-70, 2004.

[3] Arigit Biswas, Razvan Cheveresan, Joel Emer, Shubhendu S. Mukherjee, and Ram Rangan., "Computing architectural vulnerability factors for address-based structures", 32nd International Symposium on Computer Architecture, pp. 532 – 543 (2005).

[4] Xiaodong Li, Sarita V. Adve, Pradip Bose and Jude A. Rivers, "SoftArch: An Architecture-Level Tool for Modeling and Analyzing

Soft Errors", Proc Intl. Conf. Dependable Systems and Networks, (2005).

[5] H.T.Nguyen, Yoad Yagil, Norbert Seifert, Mike Reitsma, Chip Level Soft Error Estimation Method, IEEE Transactions on Device and Materials Reliability, Vol 5, No 3, Pages 365-380

[6] J. Ludden, et. al. Functional Verification of the POWER4 microprocessor and POWER4 multiprocessor systems, IBM J. Res. & Dev. Vol 46 No. 1 January 2002

[7] S. Kim and A.K.Somani, Soft Error Sensitivity Characterization for microprocessor dependability enhancement strategy, Proceedings of the International Conference on Dependable Systems and Networks (DSN), 2002, 416-425

[8] J. Kellington, R.McBeth, P. Sanda, R. Kalla, Proton Irradiation Analysis of Microarchitecture Derating of IBM Power6 System. 2nd Workshop on System Effects of Logic Soft Errors , www.selse.org

[9] Shubhendu S. Mukherjee, Christopher Weaver, Joel Emer, Steven Reinhardt, and Todd Austin, "A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor", Proc. 36th Int. Symp. on Microarchitecture (2003).

[10] Cristian Constantinescu, "Neutron SER Characterization of Microprocessors", Proc. Int. Conf. on Dependable Systems and Networks (2005).

[11] Sertac Cakici, Pia Sanda, Kenneth Wright, Scott Swaney, Ethan Cannon and Jared Day, "Proton Irradiation Studies Single Event Upsets in IBM POWER5+ System" SELSE II (2006).

[12] Vicky Wong and Mark Horowitz, "Soft Error Resilience of Probabilistic Inference Applications", SELSE II (2006).

[13] Mei-Chen Hsueh, Timothy Tsai, Ravishankar Iyer, "Fault Injection Techniques and Tools", IEEE Computer, April 1997, pp 75-81

[14] Fabien Faure, Raoul Velazco, Paul Peronnard, "Single-Event-Upset-Like Fault Injection: Comprehensive Framework", IEEE Transactions on Nuclear Science, Vol 52, No. 6, December 2005 pp 2205-2209

[15] Edward Czeck, Daniel Siewiorek, "Observations on the Effects of Fault Manifestation as a Function of Workload", IEEE Transactions on Computers, Vol 41, No. 5, May 1992, pp 559-565