Sustaining Error Resiliency: The IBM POWER6™ Microprocessor

Pia N. Sanda, Kevin Reick, Scott Swaney, Jeffrey W. Kellington, Prabhakar Kudva

IBM Systems Group, Poughkeepsie, NY

IBM Systems Group, Austin, TX

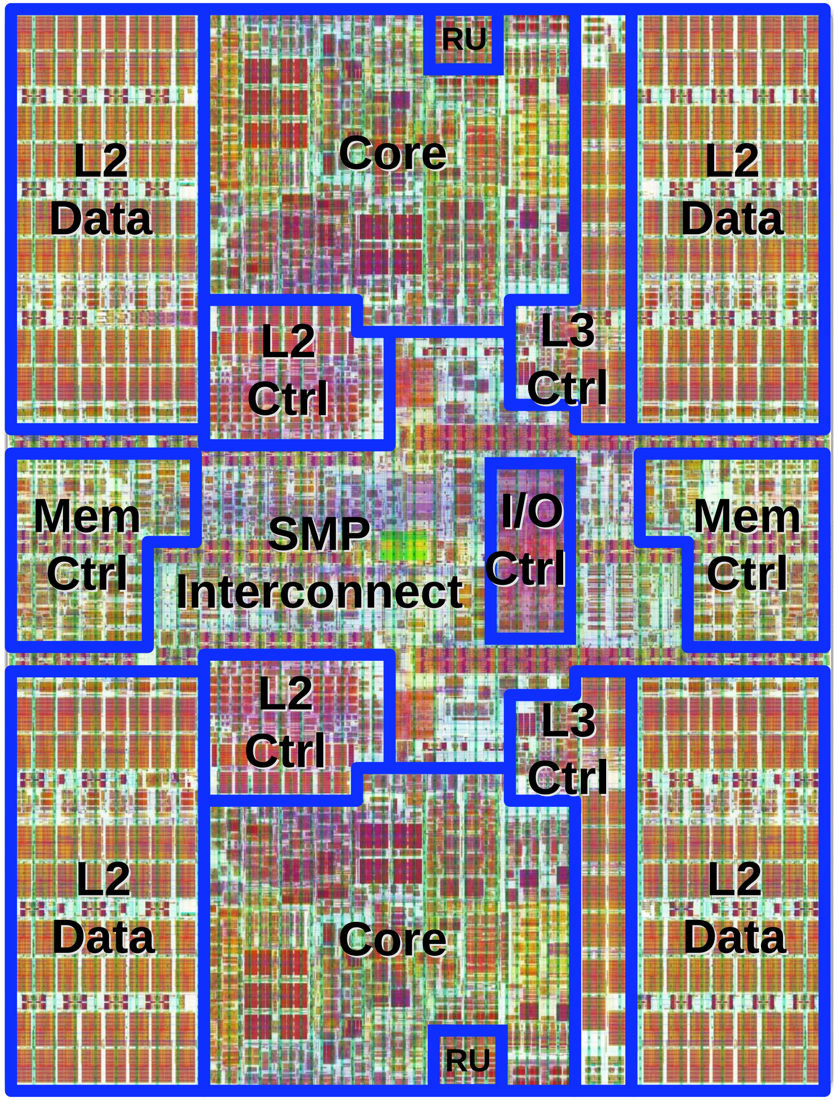
IBM Research, Yorktown Heights, NY

Outline

- POWER6^{TM} Overview
- RAS Objectives
- Describe new RAS Features
- Validation of resilience with proton beam accelerated testing
- Conclusions

POWER6 Chip Overview

- Ultra-high frequency (4.7GHz) dual-core chip
  - 7-way superscalar, 2-way SMT core
  - 9 execution units
    - 2LS, 2FP, 2FX, 1BR, 1VMX,1DFU
  - 790M transistors
  - 2x4MB on-chip L2
  - On-chip L3 directory and controller (32MB)
  - Two memory controllers on-chip
  - Recovery Unit
  - Scaleable to up to 64-core SMP systems
- Technology
  - CMOS 65nm lithography, SOI

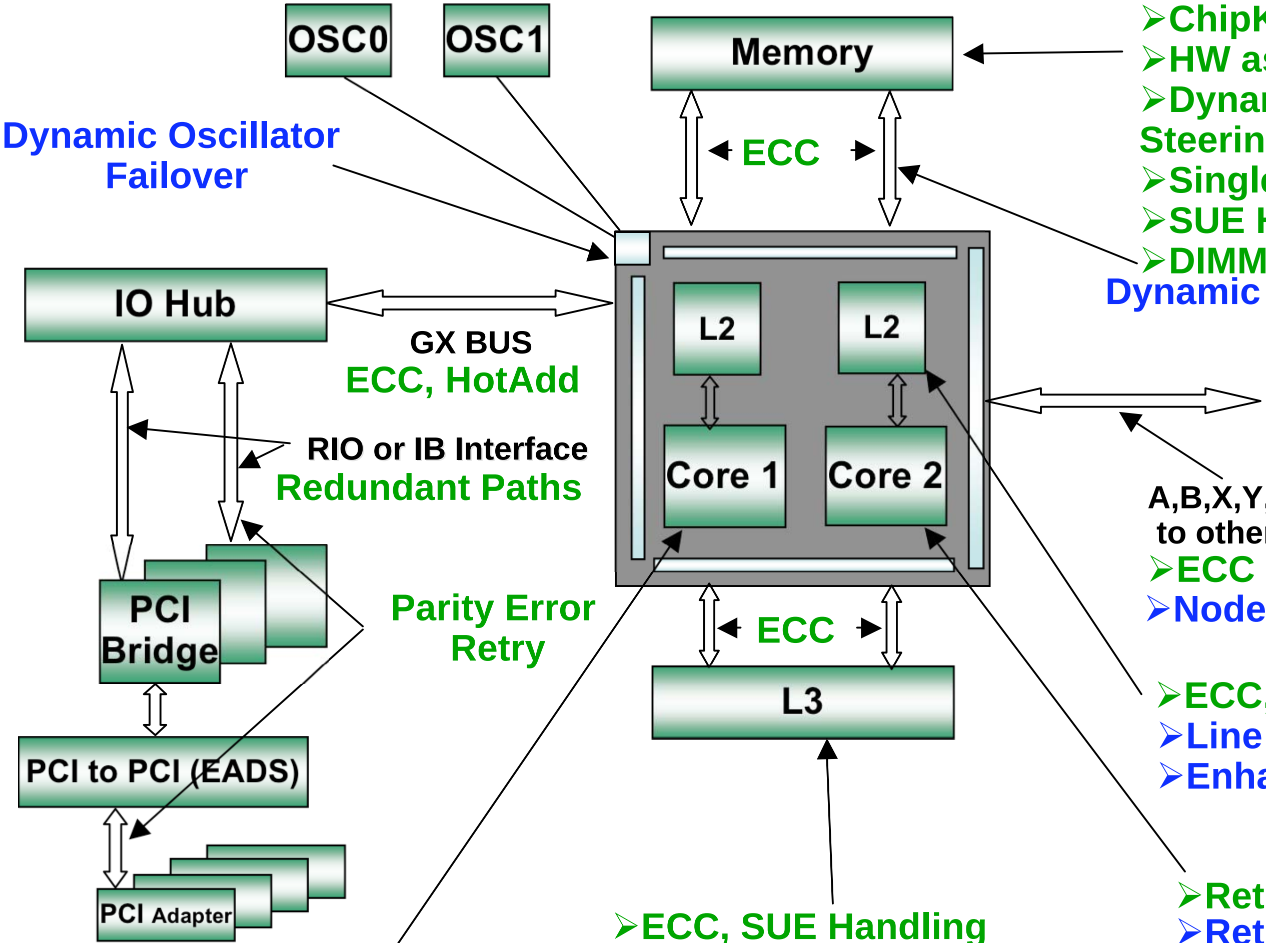


Fault-Tolerance Challenges

- **Technology Scaling**
  - Increasing rates of hard and soft errors
- **Consolidation increases risk and impact of system outage!**
  - As size of system and network increases, number of parts and interconnections increases
  - **But reliability, availability, and service costs are expected to stay the same!**

POWER6 Goal: Dramatically increase ability to recover errors without system down time

Reliability and Availability Features New to POWER6



- ChipKill Protection
- HW assisted Scrubbing
- Dynamic Redundancy, Bit Steering
- Single cell OS page deallocation
- SUE Handling
- DIMM Level Address.Ctrl ECC

Dynamic I/O Bit Line repair (eRepair)

Dynamic Oscillator Failover

ECC, HotAdd

Redundant Paths

Parity Error Retry

A,B,X,Y,Z, Fabric Bus Interface to other MCMs, Nodes

- ECC
- Node Hot Add / Repair

- ECC, SUE Handling
- Line delete
- Enhanced Cache Recovery

- Retry for I/D Cache parity errors
- Retry for SRAM/Regfile errors
- Retry for control errors

- ECC, SUE Handling
- Line delete
- Enhanced Cache Recovery

- Alternate Processor Recovery
- Partition Isolation for Core Checkstops

POWER6 RAS EXECUTION

- Error Detection and Recovery requirements were specified during the High Level Design phase
- Firmware Recovery assists specified early
- The POWER6 RAS design was a collaboration between the System p and System z processor design teams
- POWER6 shares design methodologies and macros with the System z processor
- Many of the recovery techniques used in POWER6 were initially developed for the System z processor
  - Instruction Retry
  - Alternate Processor Recovery
  - Core checkstop isolation

Functions to protect against Core errors

- Processor Instruction retry
  - Retries instructions that were affected by hardware errors
  - Protects against soft errors and intermittent errors

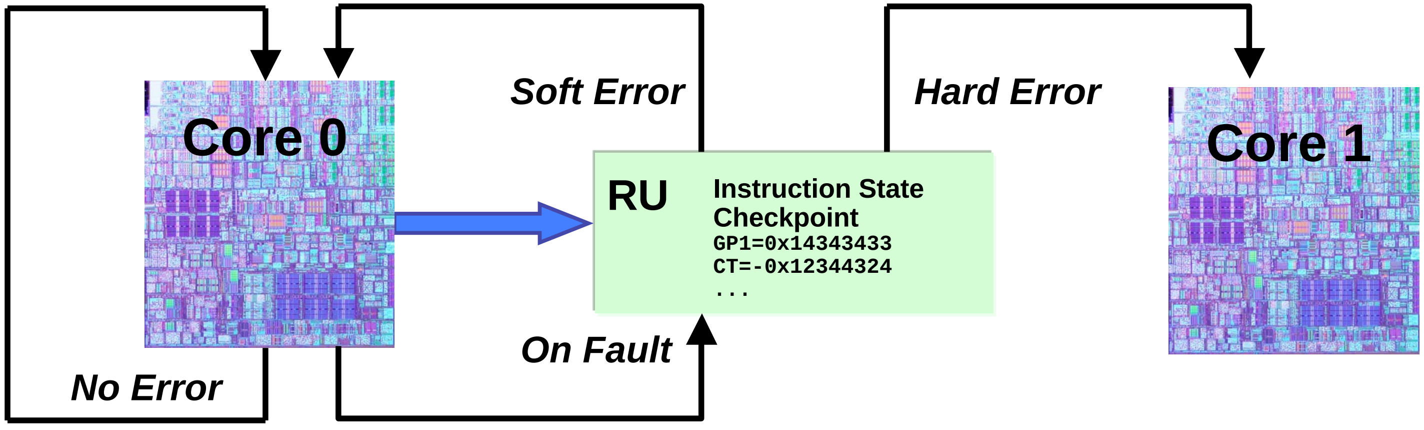
- Alternate Processor Recovery
  - If instruction retry encounters a second occurrence of the error. (i.e., Solid defect)
  - Moves workload over to an alternate/spare processor

- Processor contained checkstops
  - Limits impact of many processor logic/cmd/ctrl errors to just the processor executing the instruction

Error Detection is first step to Recovery

- 100% ECC protection for caches and interfaces
- >99% of small SRAMs and Register Files parity protected
- Dataflow protection
- Protocol checking between functional units
- Control logic protected by parity and consistency checking
- Floating Point Residue Checking
- Queue management (Underflow/Overflow)
- Architected Registers
- Store Data

Core Recovery



Non Error Case

- Core architected state is check pointed at every instruction completion
- Circuitry checked every cycle

Intermittent Error Case

- Core restarts from last check point

Hard Error Case

- Hypervisor moves workload to an alternate core

Core Checkstop

- High levels of error detection and isolation were specified early in the design cycle
- Core checkstops fall into two categories:

Recoverable

- Core Sparing moves the work to another processor

Non Recoverable

- The partition running on the core at the time of the fault is terminated
- Other partitions are not affected

Enhanced Cache Recovery

Single bit errors

- Soft errors are purged from the cache to force a refresh of the cell
- Hard errors will result in line delete.

Multi bit errors

- Hardware will purge and delete the damaged location
- Firmware will dynamically de-configure the core attached to the defective cache

System Recovery of Cache UEs

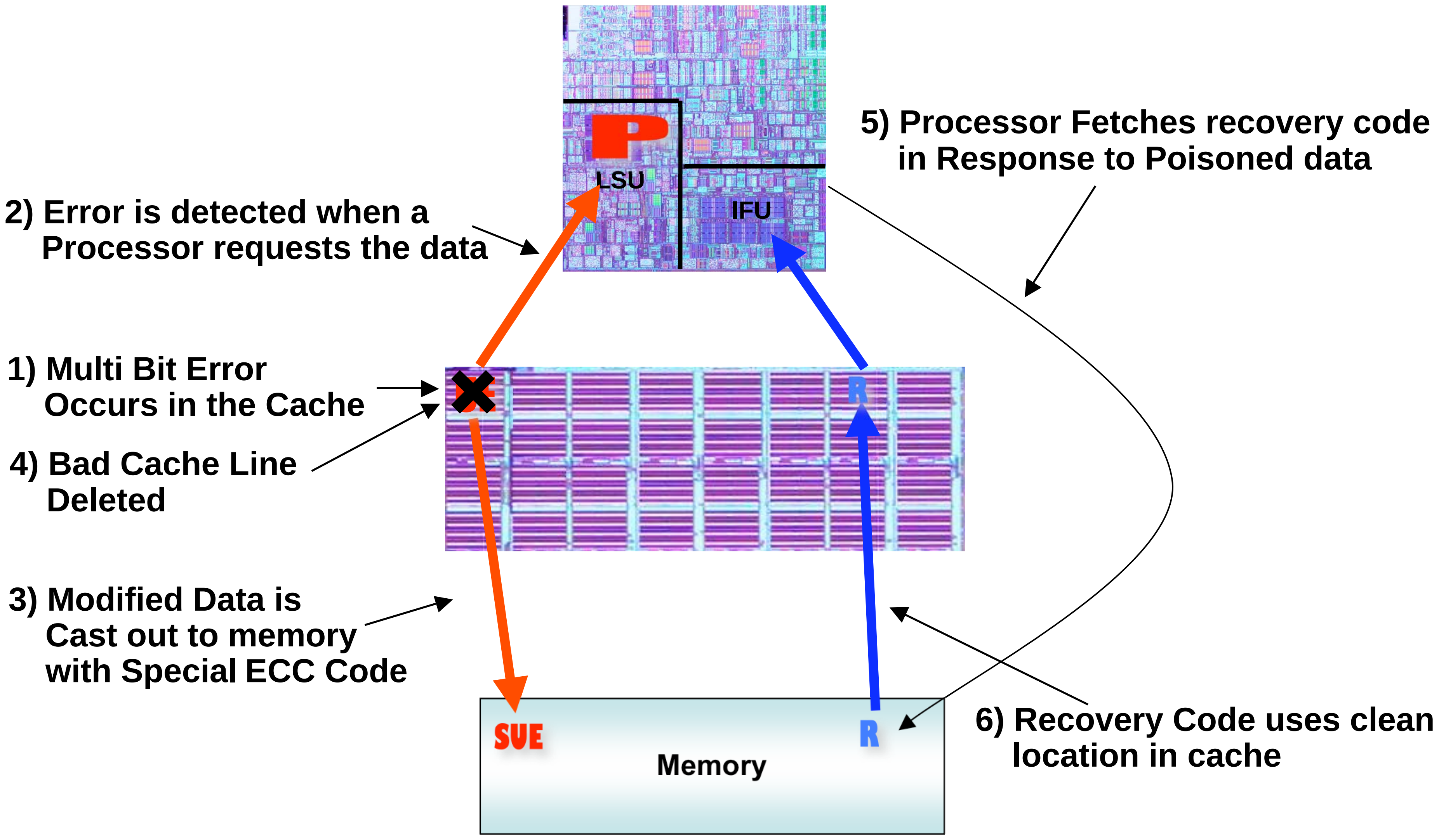
Problem

- POWER6 systems employ System Recovery Code for Uncorrectable Errors detected in the Cache Hierarchy
- If cache location is damaged, the same code being used to recover the initial error could be damaged as well

Solution

- POWER6 has automatic purge and delete for L2 and L3 Cache UEs
- Non-modified lines are re-fetched from Main Store and recovered transparently
- Modified lines are frequently contained to affected application, occasionally resulting in partition outage.

Enhanced Cache Recovery



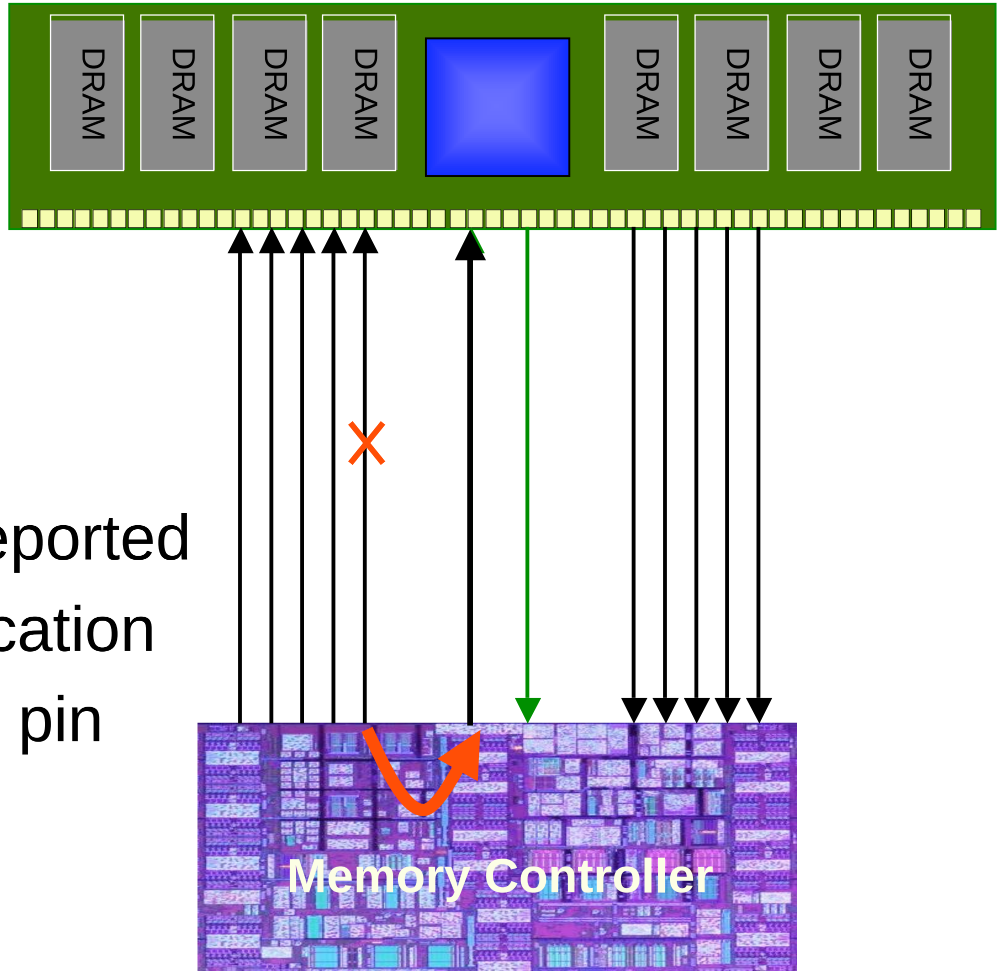
Dynamic I/O Bit Line repair (eRepair)

Memory Data and Control

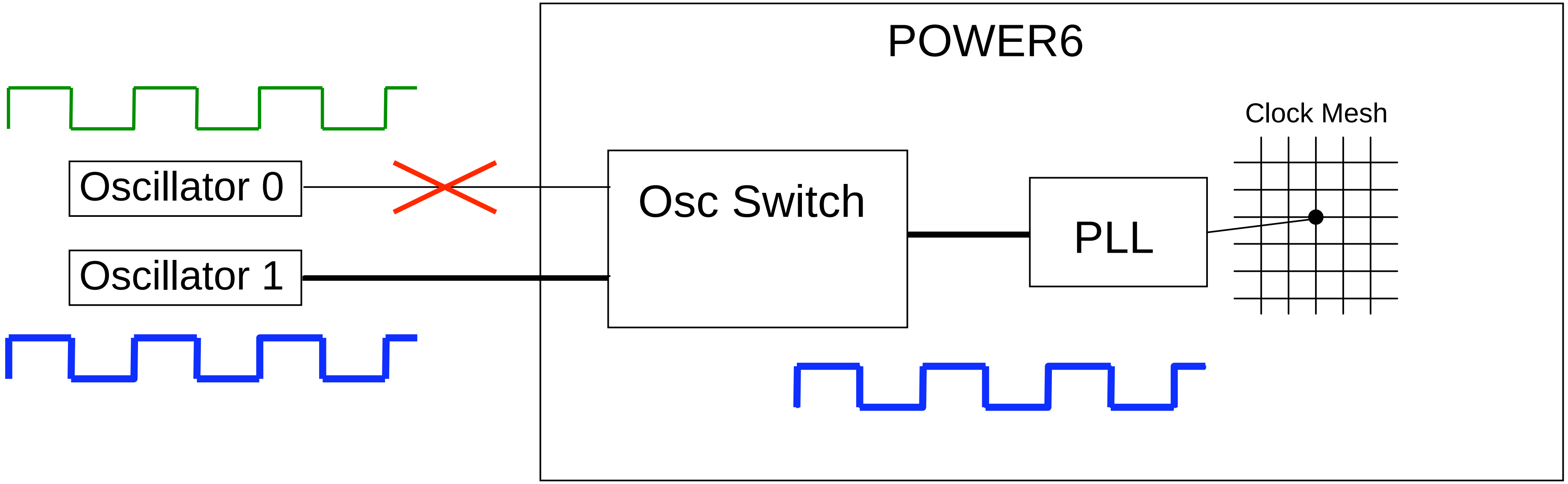
- Protected with ECC
- Spare Pins added

If a pin breaks

- Correctable errors are reported
- Transparent to the application
- Data redirected to spare pin



Dynamic Oscillator Failover



- System running on Oscillator 0
- Fault Detected on Oscillator 0
- Switch to Oscillator 1 with no disruption to system operation. Eliminates a single point of failure from the system.

Beam Verification of RAS features

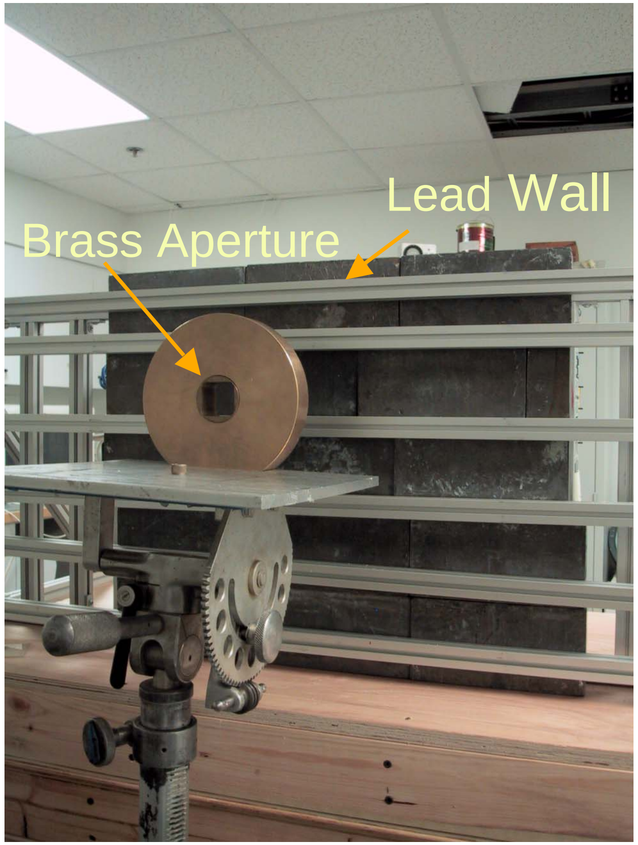
- POWER6 resilience was verified in running systems using two methods of accelerated testing:

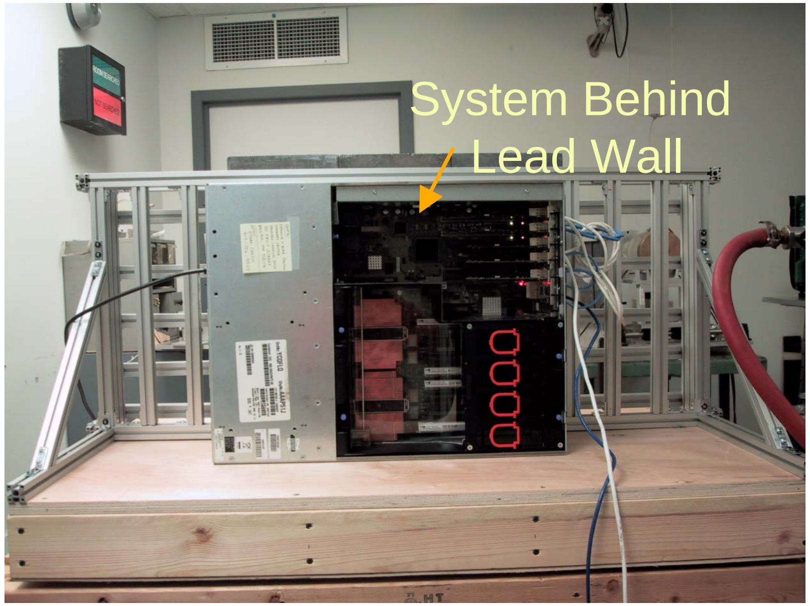
1. Alpha Particle emitters were added to the chip underfill. Used to test Latch and Array resilience
2. Proton Beams were fired through the chip to test the resilience to high energy radiation

POWER6 System in Beamline

- During the Proton Beam experiments 5662 events were recorded.

5,651 (99.8%) Full Recovery, transparent
10 (0.19%) Resulted in a Partition Outage
1 (0.01%) Resulted in a System Outage
Equivalent to >1,000,000 years of execution*





Taxonomy of Soft Error Effects

Machine Derating (MD)

Application Derating (AD)

Injected Faults

= N_{IF}

a) Vanished

b) Recovered

c) Checkstop

d) Incorrect Architected state

1) Errors not Impacting system

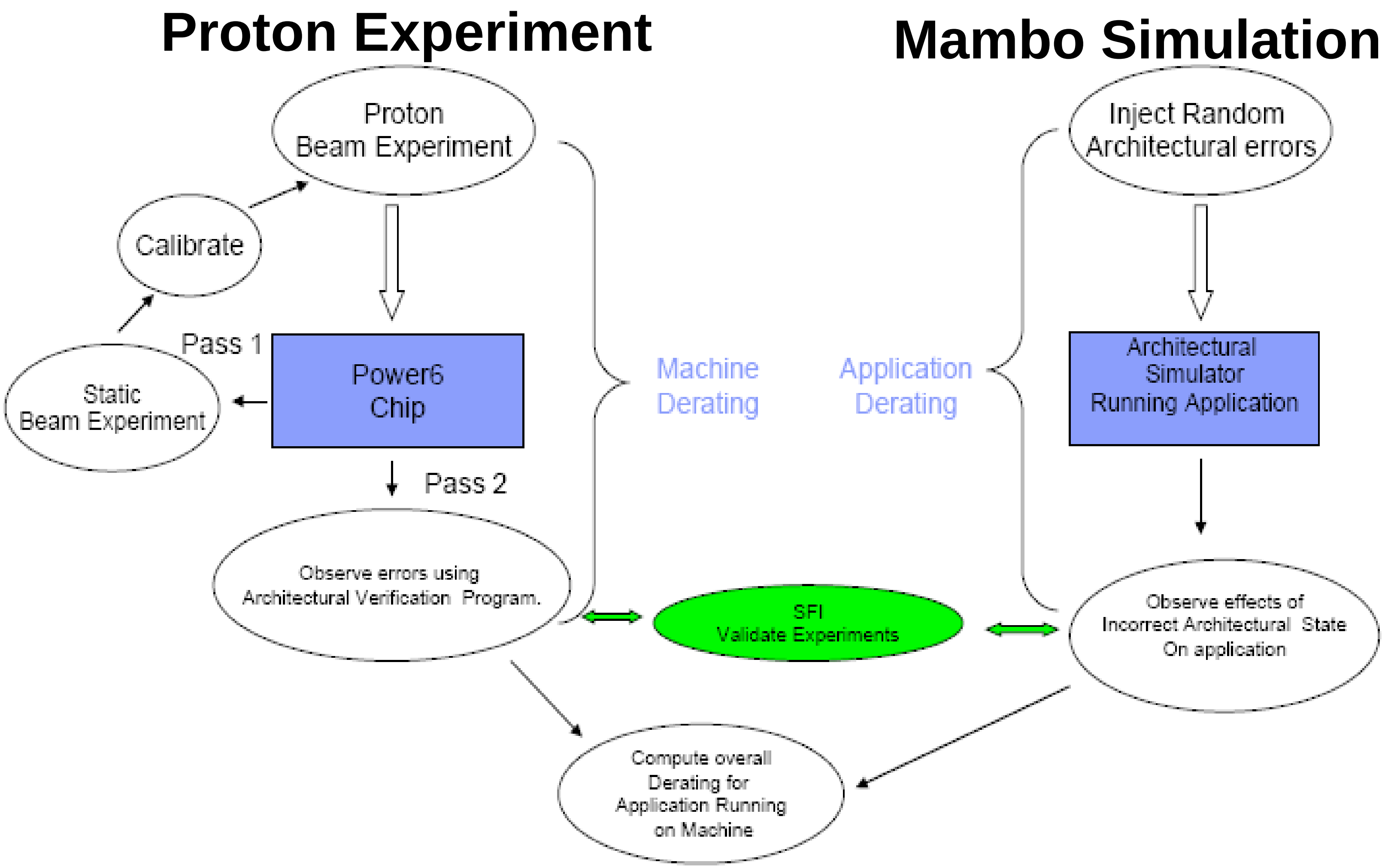
2) Software detected

3) SDC

MD depends on microarch, Instruction mix

AD depends on application

Process Flow for Derating Analysis



Processor Core Testing

- Static testing 2x2 matrix (L1;L2/0's;1's)

  Scan in bit stream pattern to L1 or L2 latches

  0's or 1's
- Irradiate the processor with a fixed amount of protons

  Scan out latch values

  Compare input and output values

  Count flips
- Functional testing:

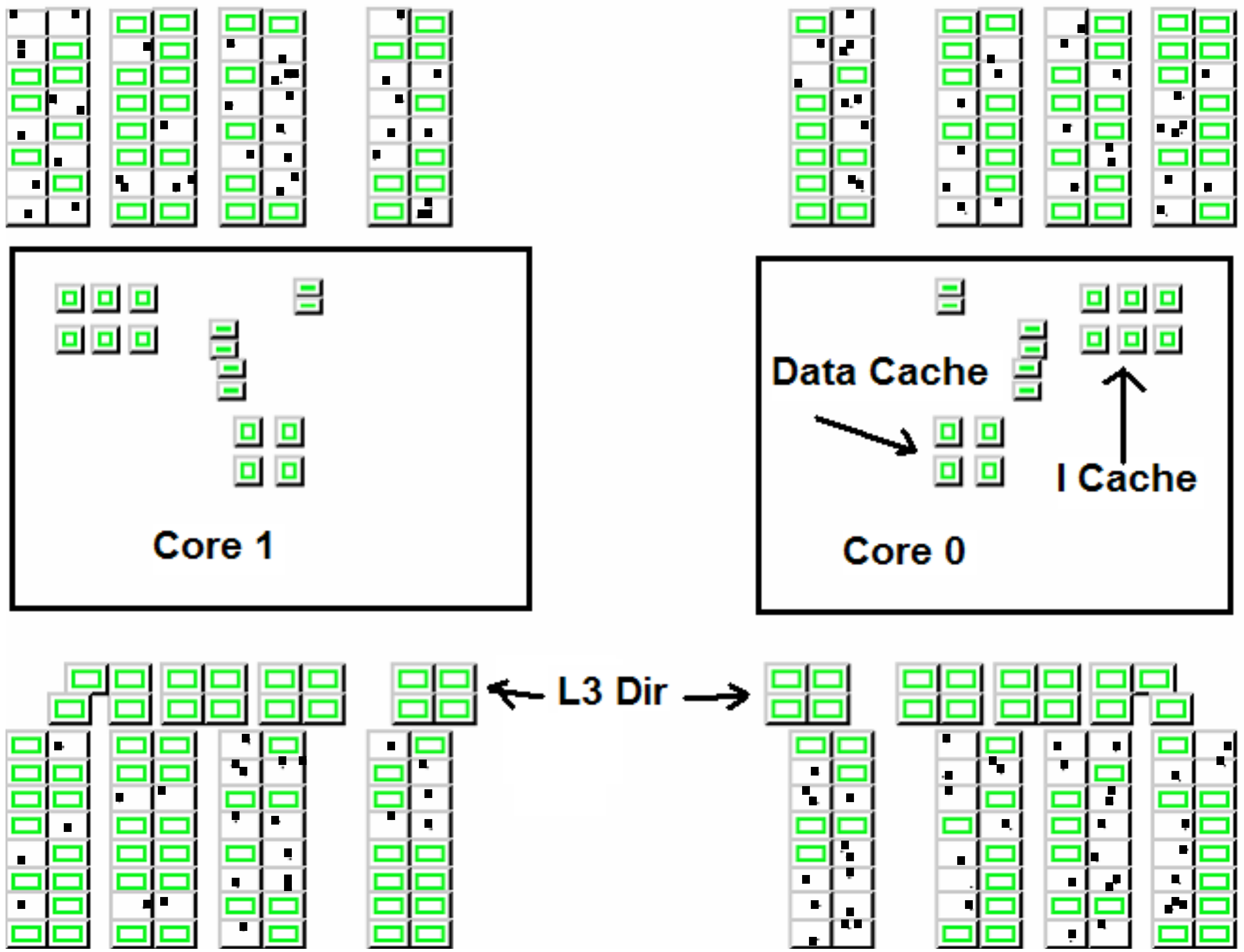
  Start an exerciser to run an instruction stream on processor

  Start irradiating the processor and check for fails.

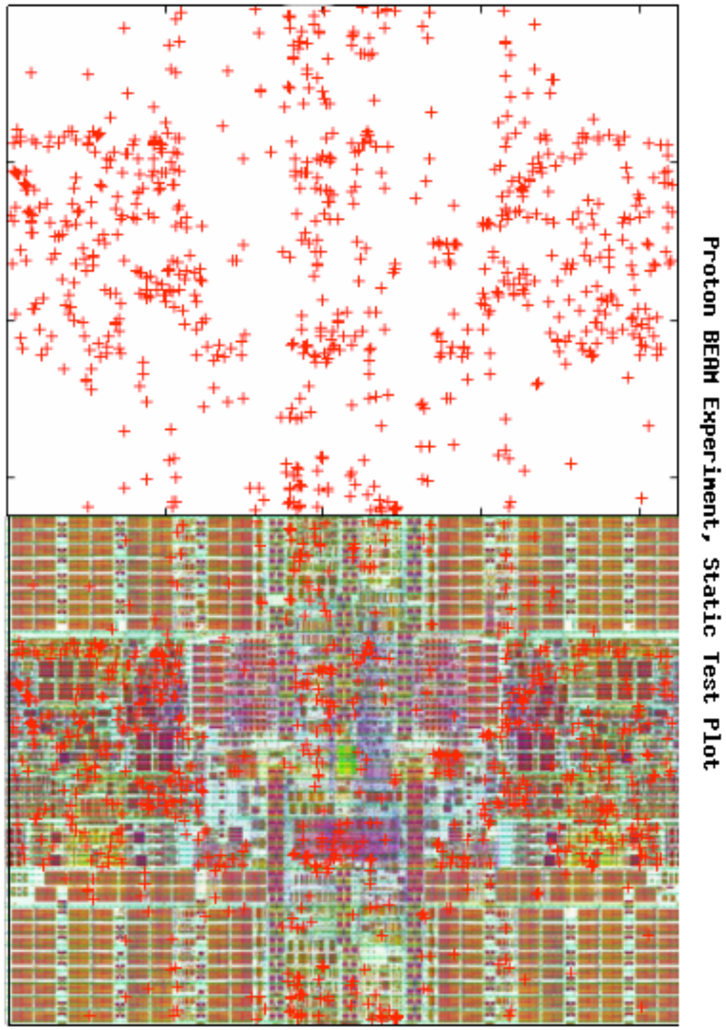
  Stop irradiation when fail occurs

  Collect rings and trace arrays to determine root cause.

Static SRAM Test Result



Map of Static Latch Flips



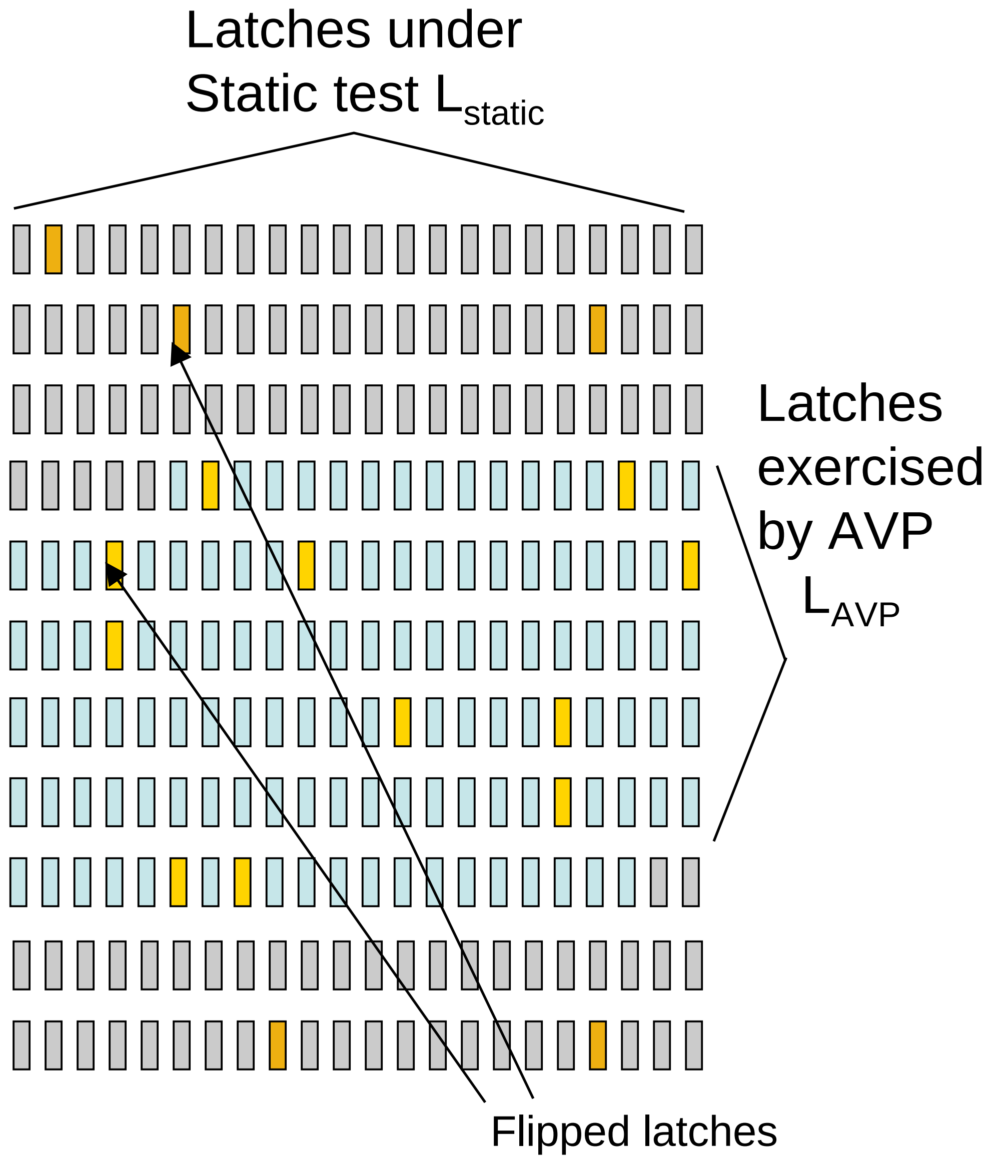
Functional Exerciser

- Exerciser runs random sequence of instructions used to validate hardware
- Intermediate results are saved
- Instruction loops repeated and final and all intermediate states are compared
- Enable recovery (field mode)
- Exercise stops when a failure occurs
- Check FIRs (Fault Isolation Registers)
- Redundant loop miscompare indicates SDC event

To measure derating compare SDC rate to reference rate For latches

- Observe number of static latch flips per MU
  - Scan known patter into latches
    - Clocks off, 1's and 0's
  - Turn on beam for MU_{static} monitor units
  - Observe number of static latch flips $N_{IF\text{-}static}$
- Scale by fraction of latches exercised by AVP

$$\frac{N_{IF\text{-}AVP}}{MU_{AVP}} = \frac{N_{IF\text{-}static}}{MU_{static}} * \frac{L_{AVP}}{L_{static}}$$

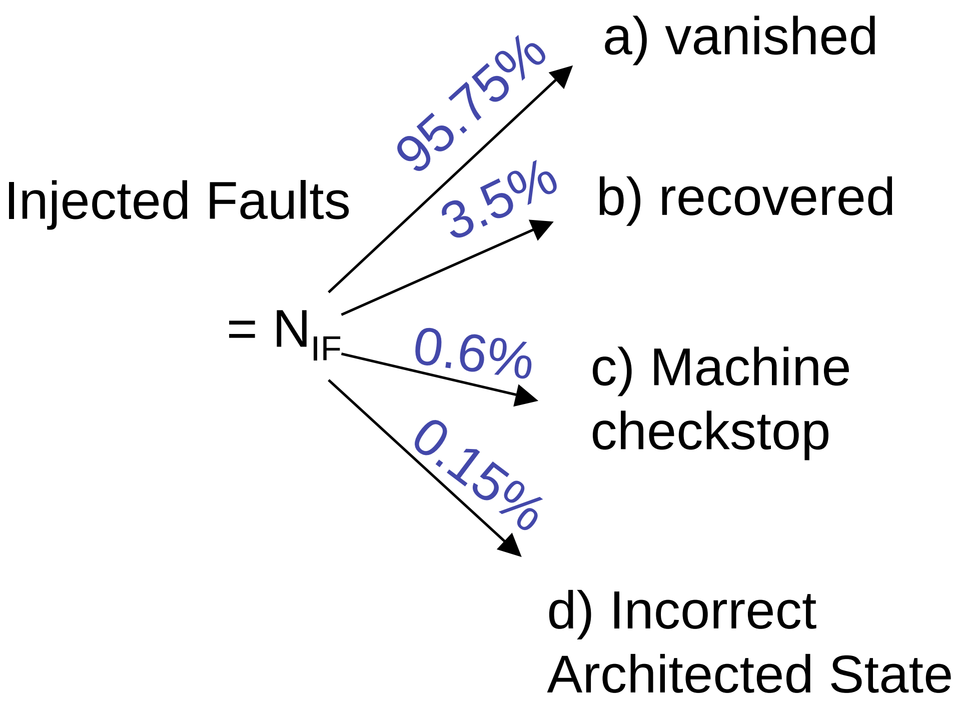


Functional Test

- Suppose AVP (architectural verification program) detects 100% of the incorrect architected state

  → Machine derating (MD) for an application with similar instruction mix and CPI is

  100x 1/0.15 = 667



Mambo Experiment - Method

- Inject incorrect architected state errors in application running on software simulator (Mambo)
- Observe outcome of injected flip
- Performed on AVP and benchmark (bzip2)

Incorrect Architected State

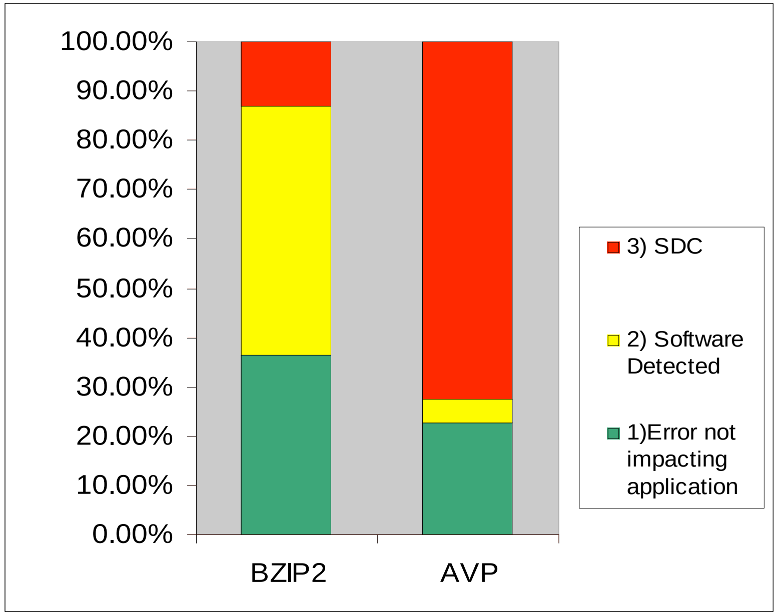
1) Errors not Impacting application

2) Software detected

3) SDC

Mambo Simulation - Results

- AVP is only 75% effective
  - Uplift vector d) by 100/75 = 1.3X (0.15% uplifted to 0.20%)
  - MD changes from 667X to 667/1.3 ~500X
- Bzip2 has an application derating (AD) of 100/15 = 6.7X
- MD is transferable between applications with similar instruction mix and CPI



Overall Derating for BZip2 on POWER6

Machine Derating (MD)

Injected Faults

= N_{IF}

95.70% ~~95.75%~~ → **a) Vanished**

3.5% → **b) Recovered**

0.6% → **c) Checkstop**

0.20% ~~0.15%~~ → **d) Incorrect Architected state**

MD =100/0.20=500X

Application Derating (AD)

0.2%*35%=0.07% → **1) Errors not Impacting system**

0.2%*50%=0.3% → **2) Software detected**

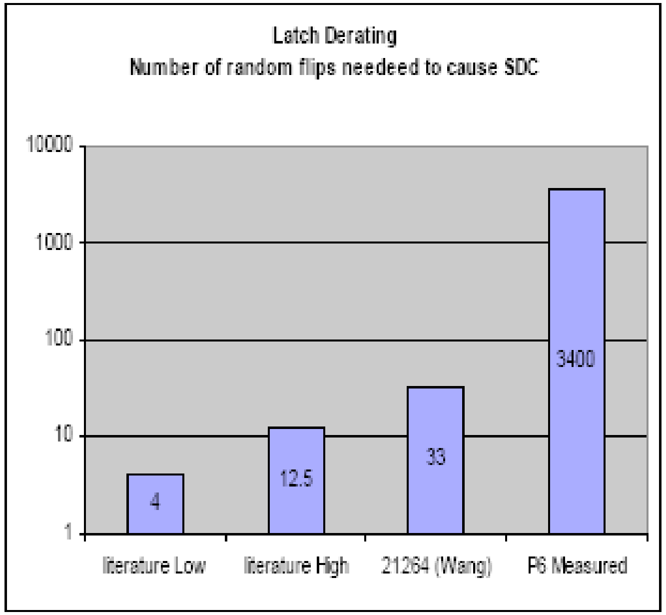
0.2%*15%=0.03% → **3) SDC**

AD = 0.20/0.03 = 6.7X

Overall Derating: 500 * 6.7 = 3400X

Comparisons to Published Work

- POWER6 provides 100X greater soft error protection than previously published results
  - *Wang et. al. performed similar experiment on 21264
    - No latch protection
  - Validated with statistical fault injection
- Results show that, on average, 3400 random latch flips are required to cause 1 SDC event
- Expected latch flips due to soft errors over the POWER6 program lifetime is much less than 3400
- Therefore, POWER6 customer's data is protected.



* 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.

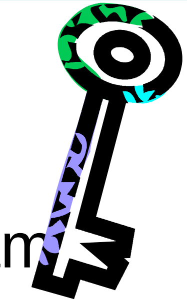
POWER6 Derating Advantage

- Hardware error detection and correction logic
  - SRAM and Regfile cells are protected
  - Error detection and recovery on data flow logic
  - Control checking provides fault detection and stops execution prior to modification of critical data
- Soft error mitigation techniques
  - Extensive clock gating prohibits faults injected in non-essential logic blocks from propagating to architected state
  - Critical state held in soft error resistant latches
- Recovery Unit prevents costly system outages
  - 81% of non-vanished latch flips were recovered
  - 99.96% of core SRAM and Regfile errors were recovered

Statistical Fault Injection: Limitations of Traditional Simulation

- Simulating small portions of the design does not accurately reflect system level derating
- If simulating a full system, it is difficult to simulate a large number of cycles which are required to allow for recoveries and self-correct at the system level
- Traditional simulation uses small architectural verification tests rather than realistic workloads. Realistic workloads require full system models that can simulate all aspects of system behaviors (OS, firmware, Full RAS pathways etc).

Hardware Acceleration

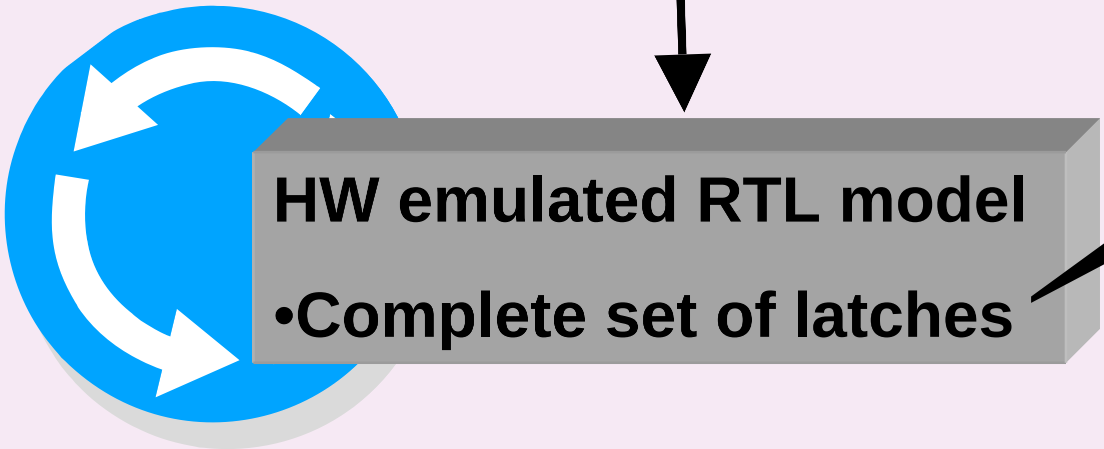


- Hardware emulation is the key to reproducing beam experiments. It's benefits include:
  - Full system models for SER
  - Observability and controllability of design latches and status registers
  - Allows use of realistic workloads and operation in real world environments
  - Simulation of a large number of cycles
  - Statistically significant subset of latch flips can be made
  - Provides analysis and understanding of RAS behavior at all levels: logic, microarchitecture and system level

AwanNG Hardware Acceleration

- Awan consists of a large number of programmable boolean processors
- Highly optimized interconnection network
- Simulation speed orders of magnitude greater than software based simulation
- Used extensively for verification in IBM including: BlueGene, P-series, Z-series etc
- Allows a broadside scan of fault injection to specific latches on an given cycle while the chip is executing a program
- Chip internals are observable and controllable via communication interface and used for status monitoring

Load program and Data



Inject flip randomly into set of latches
- e.g. all latches to simulate beam expt.

Cycle Accurate Simulation
- Inject at a random cycle
- stop at e.g. 500,000 cycles

Output all events
- Incorrect state
- Checkstops
- Corrected errors
- Vanished

HW Emulated SFI Method Reproducing Beam Experiment

SFI Results for POWER6 System

	SFI	Proton Beam
Injected Flips	16,817	1748
a) Vanished	94.98%	95.75%
b) Corrected	3.7%	3.5%
c) Checkstop	0.9%	0.6%
d) State Mismatch	0.42%	0.15%

Conclusions

- World Class RAS Depends on:
  - Hardware
    - Error Detection
    - Error Isolation
    - Error Recovery
  - Firmware
    - Error logging and thresholding
  - Hypervisor
    - Intelligent policy decisions for different error scenarios
  - Tightly interlocked design between hardware, firmware and Hypervisor
- Small investment in chip real estate provided resilience to a wide range of soft and hard errors
- Use of fault injection to validate recovery effectiveness proved to be valuable
- POWER6 continues best of breed UNIX Processor and System RAS

References

- IBM POWER6 Microarchitecture: H. Q. Le, W.L. Starke, J.S. Fields, F.P. O'Connell, D.Q. Nguyen, B.J. Ronchetti, W.M. Sauer, E.M. Schwartz, and T.M. Vaden, IBM JR&D, vol 51 no. 6, p639 (2007) http://www.research.ibm.com/journal/rd/516/le.html
- IBM z10: Charles Webb, Hot Chips 2007, IEEE Micro, to appear (2008)
- IBM POWER6 Reliabily: M.J. Mack, W.M. Sauer, S.B. Swaney, and B.G. Mealey, IBM JR&D, vol 51 no. 6, p763 (2007)
- Soft Error Resilience of the IBM POWER6 Microprocessor: P.N. Sanda, J.W. Kellington, P. Kudva, R. Kalla, R.B. McBeth, J. Ackaret, R. Lockwood, J. Schumann, and C.R. Jones, vol.52 no.3, p275 (2008)
- Fault-Tolerant Design of the IBM pSeries 690 Using POWER4 Processor Technology: D.C. Bossen, A. Kitamon, K.F. Reick, M.S. Floyd, IBM JR&D, vol 46, issue 1 (2002).

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

eServer
IBM
IBM Logo
z/OS
zSeries
z9
z900
z990
POWER
POWER5+

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Intel is a trademark of Intel Corporation in the United States, other countries, or both.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

Red Hat, the Red Hat "Shadow Man" logo, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc., in the United States and other countries.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

SAP and Netweaver are registered trademarks of SAP AG in Germany and in other countries.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.