Silent Data Corruption: Causes and Mitigation Techniques

Cristian Constantinescu

October 9th, 2008, LANL
Outline

• What is Silent Data Corruption (SDC)?

• What events can lead to SDC?
  – Silicon faults
  – Hardware design bugs
  – Firmware/Software bugs

• Can the computing industry avoid SDC?

• Can users detect and recover from SDC?

• Can users avoid SDC?

• Summary
What is Silent Data Corruption (SDC)?
SDC Definition

SDC occurs when incorrect data is delivered by a computing system to the user without any error being logged

The most famous example of SDC:

• Pentium floating-point unit bug
  – Discovered by Prof. Thomas Nicely in 1994
  – Cause: missing entries in the division lookup table
What Events Can Generate SDC?
Silicon Faults
Main Classes of Silicon Faults

- **Permanent faults**, e.g. bridges, opens
  - Reflect irreversible physical changes
  - Occur at the same location, are always active

- **Transient faults**, e.g. particle induced single event upsets (SEU), noise, electrostatic discharge (ESD)
  - Induced by temporary environmental conditions
  - Occur at different locations, at random time instances

- **Intermittent faults**, e.g. manufacturing residues, oxide breakdown
  - Occur at the same location
  - May be activated and deactivated
  - Induce bursts of errors

**Transient and intermittent faults are the most common source of SDC**
Transient Faults
Particle Induced SEU

- Intel Itanium processor
  - Neutron accelerated test
  - Test site: LANL
  - Workload: Linpack benchmark
  - 1 SDC event out of 40 detected errors

Source: C. Constantinescu et al, DSN 2005

- Fujitsu SPARC64 V processor
  - Neutron accelerated test
  - Test site: RCNP, Osaka
  - Workload: galgel benchmark
  - 1 SDC event (MTBSDC ~1 Million years at sea level)

Source: H. Ando et al, DSN 2008
Electrostatic Discharge

• Four servers produced by two manufacturers underwent operational ESD tests
  – Workload: Linpack benchmark
  – Air discharges 2kV – 15kV
  – Contact discharges 2kV – 8kV
  – 30 test points for each server
  – 20 positive and 20 negative discharges per test point

• One server experienced SDC for 15kV air discharges targeting the front disk bay
  – On line replacement of a hard drive may lead to SDC

Source: C. Constantinescu, RAMS 2005
Intermittent Faults
Fault/Error Data Collection Study

• Servers from two manufacturers were instrumented to collect errors
  – Manufacturer A: 193 servers, 16 months
  – Manufacturer B: 64 servers, 10 months

• Examples of reported errors
  – Memory
  – Front side bus

• Failure analysis performed when possible

Source: C. Constantinescu et al, SELSE 2006
Server Instrumentation

HAL – hardware abstraction layer

MCH – machine check handler

CI – component instrumentation

Instrumentation validated by fault injection
Corrected Memory Errors

- 310.7 server years

- Servers experiencing intermittent faults: 16 out of 257, i.e. 6.2%

- Corrected single-bit errors (SBE) induced by intermittent faults: 12990 out of 16069, i.e. 80.8%
Typical Signatures of Memory Intermittent Faults

Daily number of corrected SBE reported by two servers
- SBE corrected by ECC
Memory Errors – Failure Analysis

Example

SBE induced **intermittently** by poly residue, within memory chips

Source: Hynix Semiconductor
Process Variations

• Increasingly difficult to accurately control device parameters
  – Channel length and width
  – Oxide thickness
  – Dopant profile

• Intra-die variations, e.g., different transistor voltage threshold within the same SRAM cell
  – Intermittent failure of read/write operations

• Interconnect dimensions may vary
Crosstalk

• Aggressor line may induce a pulse into a victim line

• Timing violations due to crosstalk
  – Signal speedup – adjacent lines switch in the same direction
  – Signal delay– adjacent lines switch in opposite directions

• Process, Voltage and Temperature (PVT) amplifies crosstalk

• Timing violations increase with frequency
Ultra-thin Oxide Faults

• Ultrathin oxide reliability
  – Rate of defect generation decreases with supply voltage
  – Tunnel current increases exponentially with decreasing gate oxide thickness

• Soft breakdown (SBD)
  – Intermittent fluctuating current, high leakage
  – SBD examples
    • Erratic erasure of flash memory cells
    • Erratic fluctuations of Vmin in SRAM

![Graph showing Vmin vs Time for 90nm technology SRAM](image-url)

Source: M. Agostinelli et al, IEDM 2005
Negative Bias Temperature Instability (NBTI)

- At SiO₂ – Si interface, Si chemical bonds are satisfied by hydrogen annealing
- Si – H bonds may be broken by
  - Negative bias of PMOSFET
  - High operating temperature
- Breakage leads to reduced drain current
- Over time, increased number of broken Si – H bonds leads to transistor degradation
- Aggressive oxide scaling increases the likelihood of Si – H breakup

Source: H. Kufuoglu, M. A. Alam, IEEE TED 2006
**Activation of Intermittent Faults**

<table>
<thead>
<tr>
<th>Voltage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.70V</td>
<td>*******************************************</td>
</tr>
<tr>
<td>1.45V</td>
<td>*******************************************</td>
</tr>
<tr>
<td>1.20V</td>
<td>*******************************************</td>
</tr>
</tbody>
</table>

Voltage and frequency shmoo

- Faults may be activated by
  - Voltage
  - Frequency
  - Workload
  - Temperature
Activation of Intermittent Faults

- Four corner environmental test
- Workload: Linpack benchmark
- Intermittent faults activated by voltage and temperature variations

Failure analysis results
  - Memory controller setup and hold-time violations

Source: C. Constantinescu, RAMS 2005
Hardware design bugs
PCI-E Design Bug Example

- CRC error detected in inbound packet header
- Packet is discarded and transaction is retried
- New (correct) packet discarded again due to 1st design bug
- Data is read from inbound buffer
- Underflow condition in inbound buffer not signaled due to 2nd design bug
- Data is read from empty buffer = SDC

C. Constantinescu et al, SELSE 2006
Firmware/Software bugs
SDC due to FW/SW

• Validation of Intel Teraflop “supercomputer”, 1996
  - Workload: Linpack benchmark
  - Faults injected in data signals induce SDC
  - Root cause: FW did not enable processor bus ECC

• Solaris 7 and 8 OS bug, 2001
  - SDC may occur when attempting dynamic reconfiguration
Can the computing industry avoid SDC?
Two Schools of Thought

- **Fault avoidance**
  - Improved manufacturing processes, better materials, improved design, validation and testing

- **Fault tolerance**
  - Error detecting and correcting codes
  - Checking for invalid states and protocol violations
  - Self checking circuits
  - Residual arithmetic
  - Space redundancy – particularly attractive for multi-core processors
  - Time redundancy - e.g. redundant multithreading
  - Voting
  - Failure prediction - can take advantage of specifics of the intermittent faults
  - Graceful degradation – e.g. interconnect width
Fujitsu SPARC64 V Microprocessor

- Data integrity protection
  - ECC or parity on all caches
  - Byte parity on ~86% of flip-flops and latches
  - Residue checking and parity for ALU
  - Checkpointing and retry for recovery

Source: H. Ando, DSN 2008
IBM G5/G6 Microprocessor

- Mirrored Instruction and Execution units

- Compare outputs in n-1 instruction pipeline stage
  - No error: update checkpoint array (register content and instruction address into R-unit) in last pipeline stage and continue normal execution
  - Error detected: Reset CPU (except R-unit), purge cache and its directory, reload last correct state from checkpoint array, retry

- Transient faults are recovered from

- Error threshold used for intermittent faults

- Permanent faults require activation of a spare CPU under OS control

Source: L. Spainhower, T. A. Greg, IBM JR&D, 1999
Can users detect and recover from SDC?
User Level SDC Detection and Recovery

• Several applications/tests may detect SDC
  – Linpack benchmark derives residues
  – Galgel benchmark analyses fluid oscillatory instability
  – Repeated transfer of large files
    – Error detection by verifying checksum or comparison

• Space redundancy: run the same application on two independent partitions and compare results

• Time redundancy: run the same application twice and compare results

• Employ checkpointing for recovery
Can users avoid SDC?
Measures to Lower the Likelihood of SDC

- Tight procurement requirements
  - What Reliability, Availability, Serviceability (RAS) features are provided?
    - Data integrity protection
    - Error reporting
    - Error recovery
  - How are RAS features validated?
    - Is extensive fault/error injection done?
  - What environmental tests are performed?
    - What is the workload?
    - What is the sample size?
Measures to Lower the Likelihood of SDC

• Continuously analyze error logs
  – Failure prediction
    • Correlate errors with workload and environmental conditions
    • Replace nodes that experience high error rates (both corrected and uncorrected errors)
  – Request manufacturer to perform failure analysis on failing nodes
  – Request corrective actions, if necessary

• Periodic testing of the systems
  – Run tests able to detect SDC (e.g., Linpack) on a “test partition”
  – Change environmental conditions (e.g., turn off fans in one node at a time; monitor temperature and Linpack results)
  – Rotate test partitions
Summary

• Computer technology is a two edge sword
  – Lower dimensions and voltages and higher frequencies have led to tremendous performance gains
  – Transient and intermittent faults have become a serious challenge to developers and manufacturers
  – Increased complexity raises design and validation issues
  – SDC is a real threat, especially in the case of large HPC systems

• Fault avoidance, by itself, will not be able to ensure the desired reliability

• FAULT TOLERANCE is becoming a requirement for HPC