Impact and Mitigation of DRAM and SRAM Soft Errors

Charlie Slayman
OPS A LA CARTE
990 Richard Avenue, Santa Clara 95054
408-654-0499 office / 408-603-6276 cell
charlies@opsalacarte.com
Outline

• What are soft errors and why are they important for memory applications?
• What are the various sources of soft errors?
• DRAM and SRAM soft error technology trends
• Mitigation Techniques
  – Process and Material
  – Circuit Design
  – Memory System Architecture
• Conclusions
What are soft errors?

• Soft errors are any change in the output or state of a circuit that is **not permanent** and can be corrected by a simple
  – re-write
  – re-compute
  – circuit reset operation

• By contrast, hard errors are the result of some permanent (or possibly temporary)* physical change the characteristics of a device

* some types of hard errors are reversible (e.g. annealing of oxide damage)
What causes soft errors?

• Charge generated by an energetic particle due to direct or indirect ionization causing voltage and current swings in the circuit

• Direct Ionization
  – Electromagnetic (coulomb) interaction of an energetic particle with the electron cloud of the target material

• Indirect Ionization
  – An energetic particle interacts with the target material to produce one or more charged secondary particles
  – Elastic and inelastic collisions between the energetic particle and the nuclei of the target material
Sources of Energetic Particles

• Alpha Particles
  – contamination (i.e. radioactive isotopes) of IC process chemicals and packaging material

• Neutrons
  – High Energy (>1MeV) caused by cosmic ray particles (>> 1 GeV) interacting with earth’s atmosphere
  – Thermal Neutrons (~25 meV) resulting from thermalization of high energy neutrons
Why Are Soft Errors Important?

- Soft error rates can be orders of magnitude higher than hard fail rates
- As device technology scales, the amount of charge required to upset a circuit (known as critical charge, $Q_{crit}$) is getting smaller and smaller
- If proper steps are not taken, soft error rates will increase at the chip level
Brief History of Soft Errors in the Terrestrial Environment

- **1961** – Postulate that device geometry limited to 10um by cosmic rays

- **1978** - Soft errors observed due to alpha particles in device package

- **1979** – Contribution of cosmic rays to terrestrial soft errors

  – Observation of soft errors from neutrons and protons

- **1995** - Interaction of 10B with thermal neutrons

- **2000** – Soft errors catch Wall Streets attention!
  *Forbes Magazine*, Nov. 13 - Cosmic Ray Soft Errors in SRAM cache used on Sun Microsystems servers causing crashes.

- **2010** – Toyota?

Why not just eliminate the source of soft errors?

- **Alpha Particles**
  - low alpha material is very expensive
  - process and material contamination is a constant danger

- **Neutrons**
  - no practical way to shield high energy neutrons
  - thermal neutrons are a trickier subject (more on that later)
Basics of Charge Generation

- Generation of an electron-hole pair requires 3.6 eV. Each electron-hole-pair carries $2 \times 1.6 \times 10^{-19}$ C of charge.
- So 1 MeV of ion energy loss is equivalent to ~44 fC of generated charge or ~2.8 $10^5$ electron-hole pairs.
- A 1 GeV neutron could generate up to ~44 pC
- To put these numbers in perspective,
  - The storage capacitor of a DRAM cell is only 20-30 fC
  - The critical charge required to flip an SRAM cell is below 4 fC
Alpha Particles

- Classification of material
- Radioactive Decay
- Energy Spectrum
- Range of Alpha Particles
Classification of Alpha Materials

- < 0.002 cm^{-2} hr^{-1} ULA - Ultralow \( \alpha \) material
- < 0.05 cm^{-2} hr^{-1} LA - Low \( \alpha \) material
- > 0.05 cm^{-2} hr^{-1} Typical of uncontrolled material

Alpha Particles and Radioactive Decay

- Originate from nuclear decay
  - Package $^{238}\text{U}$, $^{232}\text{Th}$
  - Solder $^{210}\text{Pb}$

- Energy
  - Typically about 5 MeV
  - Max ~ 9-10 MeV

- Range
  - Typically ~ 30 μm
  - Max < 70 μm

Alpha Particle Spectra

- Alpha particles come from trace impurities of radioactive isotopes in semiconductor processing and packaging. Emission spectra from thin-film $^{238}$U, $^{235}$U and $^{232}$Th is shown here.

- Note that the energy peaks are discrete and each isotope has a unique energy spectra. This spectra is expected when alpha contamination is at the surface of the IC.
Alpha Spectra Smeared As It Looses Energy

- Energy loss as alphas exit the source lead to a continuum of lower energy particles

JEDEC JESD89A, Fig. D-2
Alpha Particle Interaction with Silicon

- Peak charge generation - 16 fC/um at ~1 MeV
- 10 MeV alpha can penetrate 70um!

Bragg Peak (max energy loss)

JEDEC JESD89A, Fig. D-3

Alpha particle starts at right and moves to left as it looses energy
High Energy Neutrons

- The high energy neutron flux extends beyond 1GeV!!!
- The integrated neutron flux at sea level (NYC):
  - 1 to 10MeV → ~6 neutrons/cm²-hr
  - > 10MeV → ~14 neutrons/cm²-hr
Thermal neutron spectra

- Peak at 2.5e-8 MeV (i.e. 25 meV) are the thermal neutrons
- Flux varies depending on high energy background and environment: Typically 1 to 10 n/cm²-hr at sea level

Goldhagen, 2008

Normalized to NYC using JESD89A Eqn. A.2
Why Are Thermal Neutrons Important?

- $^{10}$B nucleus has a large capture cross section for thermal neutrons
- $^{10}$B is present in
  - Boro-phospho-silicate glass (BPSG) planarizing layers
  - PMOS Source-Drain Implants
- ($^{10}$B + thermal neutron) is equivalent to an “alpha particle time bomb” imbedded in the IC

From Baumann (10/31/00)
Neutron Shielding Is Not An Option

- Shielding is not a viable option for high energy neutrons
- Concrete is not a good shield for thermal neutrons, but a material rich in $^{10}$B could work

Cell Flip in SRAM

- Particle strike on nmos or pmos inverters can flip “1” to “0” or vice versa and is frequency independent
- Particle strikes in bit line during read/write is frequency dependent

• Particle strike on storage cell or select transistor can discharge cell and is frequency independent
• Particle strike on bit line during read/write operation is frequency dependent

Neutron Strikes on DRAM Logic

cell upsets (single and multi-cell)  logic upsets (kcell)

L Borucki  G Schindlbeck and C Slayman, IRPS 2008
180nm → 90nm DRAM Soft Errors

- Multi-cell upsets due to charge collection from nearest neighbor cells
- Logic upsets can be comparable to multi-cell upsets

L Borucki, G Schindlbeck and C Slayman, IRPS 2008
Single Event Latch-up

- Though not unique to memory, single event latch-up can be difficult to deal with by any mitigation technique

H. Puchner, et al., IRPS 2006
Memory Soft Error Trend Per Bit

- DRAM soft error rates are trending downwards because cell capacitance **IS NOT** scaling
- SRAM soft error rates are remaining roughly flat
Memory Soft Error Trend Per Chip

- SRAM cell packing density is increasing more rapidly than soft error rate per cell is falling off
Example of Soft Error Rates to Anticipate - Cache

- Intel 7500 Series Xeon Processor with 24MB L3$
- SRAM soft error rates $\sim 1e^{-4}$ to $1e^{-3}$ FIT/bit
- This translates to 20,000 to 200,000 FIT or 0.2 to 2 errors/year per CPU (sea level, NYC)
Example of Soft Error Rates to Anticipate – Main Memory

• Up to 250 GB of main memory can be supported by an Intel 7500 Xeon CPU socket
• DRAM error rates are dropping below 1e-9 to 1e-8 FIT/bit
• This translates to 2,000 to 20,000 FIT for main memory or 0.02 to 0.2 errors/year
• About 10x less than the L3 example in the previous slide
SOFT ERROR MITIGATION TECHNIQUES

• From the previous examples, mitigation of soft errors is clearly mandatory for high reliability systems with large main memory and cache

• Mitigation techniques can be divided into two broad categories:
  – Reduce the raw soft error rate through silicon process, material and design/layout techniques
  – Let soft errors happen and then deal with them through system RAS features (architectural)
Mitigation by Reducing the Soft Error Rate – Silicon Process

• Engineer implant profiles to reduce charge collection
  • Reduction in soft error rate is modest
  • Other requirements (speed, power and area) are higher priority
• Increase the critical charge of the circuit
  • Must be balanced against speed, power and area requirements
• Silicon on Insulator (SOI) to eliminate charge collection from substrate
  • Only modest reduction (~2x) in soft error rate unless body ties are used (which blows up the layout area)
  • Factors other than soft error rate reduction will determine if a design moves from bulk to SOI
Mitigation by Reducing the Soft Error Rate – Material Selection

- **Low alpha materials**
  - Added material cost
  - Control of contamination can be tricky

- **Eliminate $^{10}$B**
  - Chem-Mechanical Polishing (CMP) to replace BPSG
  - Recent work indicates elimination of BPSG might not be sufficient.*
  - $^{11}$B isotopic separation for pmos source-drain implants might be required.

- **Die coat and underfill to shield transistors from alpha particles**
  - Make sure it is thick enough
  - Make sure it is ultra-low alpha

* Olmos et al, IRPS 2006, Wen et al, IRPS 2010,
Mitigation by Reducing the Soft Error Rate – Design/Layout Techniques

- Keep-out areas on IC
  - Separate sensitive memory from lead bumps in flip chip packaging

- Robust cell designs
  - Dual interlocked cell (DICE) – charge collection at multiple nodes required to upset the cell. As technology scales below 45nm, charge sharing is becoming a problem.
  - Layout aware (LEAP) – this technique uses layout to minimize charge sharing
  - Triple modular redundancy (TMR) – big overhead penalty
  - Internal trench cell – used in some DRAM designs
  - Added cell capacitance – used in some radiation robust SRAM designs

- Memory Cell Interleaving
  - Multi-cell upsets only appear as multiple single bit errors (not multi-bit)
  - Allows for simpler ECC codes
Keep-Out Area for Flip Chip Lead Bumps

Underfill can help attenuate high angle alpha particles

Keep sensitive devices (e.g. minimum design-rule SRAM) away from Pb bumps

• Three orders of magnitude reduction in soft error rate using internal trench cell capacitor


SC = stacked capacitor

TEC = trench capacitor with charge stored externally

TIC = trench cell with charge stored internally
SRAM CELL DESIGN – Extra Capacitance

- Increase in cell capacitance increases critical charge ($Q_{\text{crit}}$) required to flip cell

Memory Cell Interleaving

- Physical interleaving of SRAM or DRAM into different logical checkwords
- Multi-cell upset will not lead to multi-bit errors
- Single error correct – double error detect (SEC-DED) codes can handle multi-cell upset from energetic neutrons
Mitigation of Soft Errors - Architectural

- Process requires detection of error followed by some technique to correct it
- Detection and correction can be built into hardware, software or combination
- Examples
  - **Parity codes** are simple and effective at detection of single bit errors, but cannot correct the data.
  - **Discard corrupted data and fetch a clean copy**. Common for parity protected cache where a clean copy of data exists in main memory.
  - **Error correction codes (ECC)** for memory range from simple single bit to powerful sixteen bit capabilities.
  - **Memory scrubbing** - primary function is to detect latent errors in memory. It is used in conjunction with some form of ECC.
  - **Memory mirroring** allows for simpler ECC code but doubles the size of memory.
Mitigation of Soft Errors - Error Correction Code

- ECC for memory ranges from none (PCs) to powerful multi-bit correction codes (high-end servers) depending on the system design
- Other failure mechanisms already require ECC (DRAM – weak bits, SRAM – read disturb, FLASH – Vt shift)
- Effective technique with very modest area, power and speed penalties compared to techniques like TMR or Memory Mirroring
Example of Error Correction Codes

<table>
<thead>
<tr>
<th>Data Bits</th>
<th>SEC-DED</th>
<th>SBC-DBD (b=4)</th>
<th>DEC-TED</th>
</tr>
</thead>
<tbody>
<tr>
<td>check</td>
<td>overhead</td>
<td>check</td>
<td>overhead</td>
</tr>
<tr>
<td>bits</td>
<td></td>
<td>bits</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>27%</td>
<td>22</td>
</tr>
<tr>
<td>32</td>
<td>7</td>
<td>18%</td>
<td>39</td>
</tr>
<tr>
<td>64</td>
<td>8</td>
<td>11%</td>
<td>72</td>
</tr>
<tr>
<td>128</td>
<td>9</td>
<td>7%</td>
<td>137</td>
</tr>
</tbody>
</table>

- Single Error Correct-Double Error Detect (SEC-DED) Codes are common in cache designs.
- Single Byte Correct-Double Byte Detect (SBC-DBD) are more common in server main memory.
  - Many IBM (chipkill) / Intel (single device data correction – SDDC) / Fujitsu (extended ECC) designs use SBC-DBD where b=4/8/16 bits.
  - Error involving all the I/O of a single DRAM can be corrected.
- Note that as word size increases (data bits), overhead for ECC (checkbits) decreases.
ECC Registered Dual Inline Memory Module (RDIMM)

- Only 1/9 = 11% overhead in DRAM count
- “Chipkill /SDDC/Extended ECC capability” handles all multi-cell and logic errors since an alpha or neutron strike only effects a single chip
Fully-buffered (FBDIMM) and Buffer on Board (BoB) Design

- CRC protection on memory controller ↔ FBDIMM link for signal integrity
- ECC protection on DRAM for soft errors (2 out of 18 DRAM = 11%)
Conclusions

- **Source of soft errors** – alpha particles, high energy neutrons and thermal neutrons can all be significant
  - Alpha particles can be reduced by requiring purer materials but cost can be significant
  - Neutrons can’t be shielded and must be dealt with at the process and design level

- **Scaling trends**
  - FIT/bit for SRAM is not trending down as fast as packing density is growing
  - FIT/bit of DRAM is trending down because cell capacitance is not scaling. But this means logic errors in DRAM will become more significant
  - Soft errors are observable in large scale memory

- **Many mitigation techniques exist**
  - process, materials and layout have their place
  - there can be significant trade-offs with other features (power, area, performance and cost)
Conclusions (cont.)

• **ECC is the most effective mitigation technique**
  - Efficient/fast codes to detect (but not correct) when duplicates of data exist (e.g. parity and CRC)
  - More traditional Hamming codes (e.g. SEC-DED and Chipkill) to protect critical data

• **For large memory:**
  - ECC codes are the most powerful and cost effective way to deal with soft errors
  - The wider the word, the less ECC overhead
  - You probably already need ECC to deal with other fault mechanisms (weak bits, noise, etc.) anyway

• **For small cache (or registers):**
  - Soft error rates might be insignificant
  - Parity protection might be sufficient
  - Don’t design at the bleeding edge (they aren’t taking up that much real estate anyway)