Modeling Architectural Vulnerability

- Primary type of transient fault in modern CPUs is a single-bit flip due to cosmic rays
  - Radioactive materials in chip packages are no longer a problem—can be shielded against, screened out
  - Cosmic rays also cause permanent faults; not considered
- Low incidence, ~ 1 event/year/cm^2 of chip, but nearly unavoidable
- This paper: Which single-bit flips cause visible errors?

Digression: Cosmic rays

- High-energy subatomic particles that hit the earth
- Ultimately caused by things out in space, like the sun
- At sea level, 97% neutrons, 10^5 neutrons$/$cm$^2/$year
- Neutrons must hit an atomic nucleus to have any effect

J. F. Ziegler (1996) “Terrestrial cosmic rays.”
IBM Journal of Research and Development, 4(1) 19–40.

Digression: Cosmic rays

- High-energy subatomic particles that hit the earth
- Ultimately caused by things out in space, like the sun
- At sea level, 97% neutrons, 10^5 neutrons/cm^2/year
- Neutrons must hit an atomic nucleus to have any effect
- Crude model of a chip as a sheet of silicon atoms, 1 micron thick
- Only one in 240,000 neutrons hits a nucleus. . .
- . . . but *every hit causes a bit flip*
- Some variation with circuitry: CMOS less sensitive than bipolar, DRAM less sensitive than logic

J. F. Ziegler (1996) "Terrestrial cosmic rays."
IBM Journal of Research and Development, 4(1) 19–40.

Categorizing single-bit flips

- An *ACE bit* is a bit that will affect the outcome of the calculation
- An *un-ACE bit* is a bit that won't
- If we only care about the final outcome, many bits are un-ACE

Categorizing single-bit flips

- An *ACE bit* is a bit that will affect the outcome of the calculation
- An *un-ACE bit* is a bit that won't
- If we only care about the final outcome, many bits are un-ACE
- Microarchitectural un-ACE bits:
  - Idle circuits (unused cache lines, etc)
  - Mis-speculated instructions
  - Predictor state
  - Dead values

Categorizing single-bit flips

- An *ACE bit* is a bit that will affect the outcome of the calculation
- An *un-ACE bit* is a bit that won't
- If we only care about the final outcome, many bits are un-ACE
- Microarchitectural un-ACE bits:
  - Idle circuits (unused cache lines, etc)
  - Mis-speculated instructions
  - Predictor state
  - Dead values
- Architectural un-ACE bits:
  - NOP, prefetch, hint instructions (all fields but opcode)
  - Predicated-false instructions
  - Dynamically dead instructions
  - Masked bits (e.g. or-ed with zero)

The architectural vulnerability factor

- AVF of a bit $=$ probability of a visible error if that bit flips
- Equal to fraction of execution time that bit is ACE
- AVF of a structure is just average AVF of all its bits
- Can also be computed as bandwidth-latency product of ACE bits through that structure
- This allows use of a performance model (e.g. a cycle accurate simulator) to estimate AVF

Example: instruction queue

- $\sim$ 100 bits per entry in IA64 instruction queue
- 25 bits always ACE (5 control, 7 opcode, 6 predicate, 7 dest register)
- Other 75 bits assumed ACE iff *instruction* is ACE
- For SPEC2000, average 45% ACE instructions
  - Largest categories of un-ACE instructions: NOPs (26%), dynamically dead (19.6%), prediated false (6.7%)
- Works out to 28% AVF for instruction queue *bits*

Advantages

- Identifies vulnerable structures (high AVF)
- Also identifies underutilized structures (low AVF)
- Consistent with particle bombardment experiments
- Consistent with statistical fault injection
- Easier than either
  - Needs only a performance model—can be done early
  - Requires fewer experiments
  - Gives breakdown of *why* bits do not affect results

Disadvantages

- Does not model permanent faults
  - Also caused by particle strikes (at much lower rates)
  - Many other causes, more common (electromigration, thermal cycling)
- Actual incidence of bit flips is ignored
  - If you're going to worry about bit flips you have to do that
  - But often "don't worry about it" is fine
- Using benchmarks to tune correctness
  - True of any statistical error study—only as good as its sample