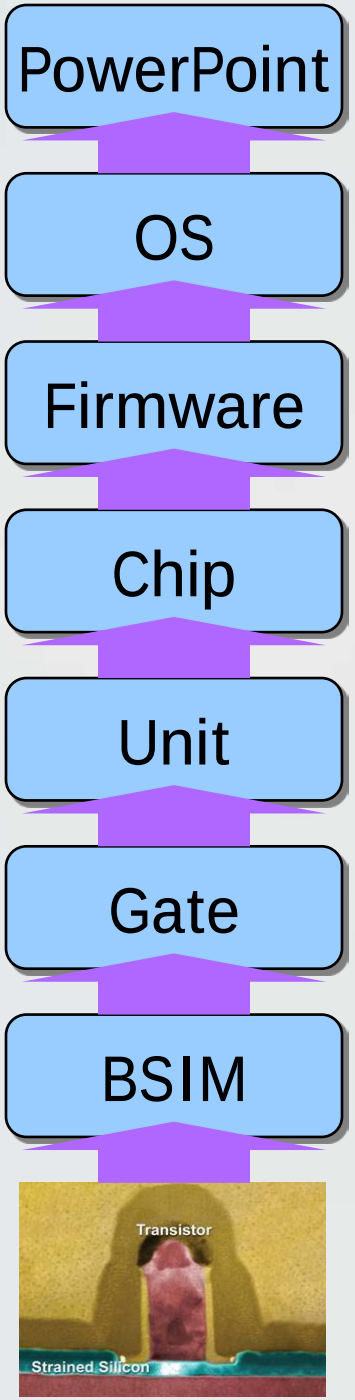
Filtering…
Breakout Sessions

Session 1, scribe: Sani Nassif nassif@us.ibm.com
Session 2, scribe: Nick Carter nicholas.p.carter@intel.com

Overall

- First session struggled in the “bottom” layers.
  - Circuits, gates, architecture…
- Second session focused on the “upper” layers.
  - Software.
- Note: this happened with absolutely no planning on anyone’s part!



First Session

- Error detection should be the main focus.
  - Once detected, it is relatively straightforward to do something about it. Many existing examples…
- Each level in the system handles and (perhaps) hides errors from those above it…
  - Things like Bit Error Rate specifications define a contract for how much error can be expected or tolerated.
- Observation: logging of errors and exposing via appropriate interfaces to higher levels is a useful facility.

Timely Filtering

- Faults are typically easier to solve “near” and “soon”.
- Waiting too long, or assuming the error has to propagate too far up means we may potentially lose the chance to fix it.
  - So filtering needs to be cognizant of the time scale at which things happen.
- A hard drive going bad is different from a latch missing an edge.

Efficient Filtering

- Efficient in what?
  - Design/Verification/Test time? No
  - Number of transistors? No
    - (some who are still stuck in the 90nm + world disagreed).
  - Power… Yes!
- The cost of error detection and filtering is very application dependent.
  - Implementations have to be such that they can be turned off completely, i.e. being zero overhead.

Second Session

- Applications will have different needs in terms of resilience and configurability.
- Q: when a hardware fault occurs, what get exposed to the software? How can this interface be managed?
- Observation: some (sophisticated) applications can directly handle faults at the user level.
  - Example: database re-try.

Fault Detection Cost

- What is the cost of fault detection and diagnosis?
  - This also came up in the open mike session…
- How much can software help by identifying critical code and data segments?
  - How would such information be passed generically?
- Observation: systems are often more (most?) vulnerable when doing fault recovery…

Challenges (1)

- Detection in higher levels (Software) was not discussed -had too much fun at the μ-arch and below.
- One major challenge is being able to do complete system level fault simulation in a way that allows optimization of the error filtering across all the levels.
  - Optimization for cost, power, resilience, etc…
  - Getting overall efficiency in a cross-level solution means that we MUST retain low level accuracy in high level views!
  - Difficult…

Challenges (2)

- An application focus needs to deal with a broad diversity of application needs. (open mike, again).
- Cost of recovery is clearly application dependent.
- Are there a set of golden benchmark cases of software resilience with specified performance, power, and robustness measures?
  - If not, does it make sense to create such cases?

Challenges (3)

- Error detection for SRAM, various regular arrays, and data paths is challenging but doable.
- Error detection for random control logic is harder.
  - Example: introducing parity prediction…
- Design Automation Challenge: automate the efficient insertion of parity prediction and management logic.
  - This is already done, by hand, in current machines. We should be able to make quick progress on it!

Opportunities

- Are we doing all we can to analyze existing fail data to get estimates and trends?
  - Something beyond planes and satellites…

- Example: mainframes (IBM Z class machines) routinely log failures in detail. Does it make sense to harvest this data and use it to drive more realistic metrics and trends?

Opportunities

- Creating a strawman resilience roadmap as part of the upcoming ITRS document can help drive research forward.

- DFM section already includes variability trends and hints at the morphing of extreme variability into “permanent” faults.
  - Should be possible to extend to more general resilience.
  - Who wants to help?