Protecting Non-Architectural Structures against Faults

Nikolas Ladas[*], Constantinos Kourougiannis[*], Yiannakis Sazeides[*], Veerle Desmet[†]

[*]Ξ – Computer Architecture Group
University of Cyprus

[†]Ghent University

Introduction

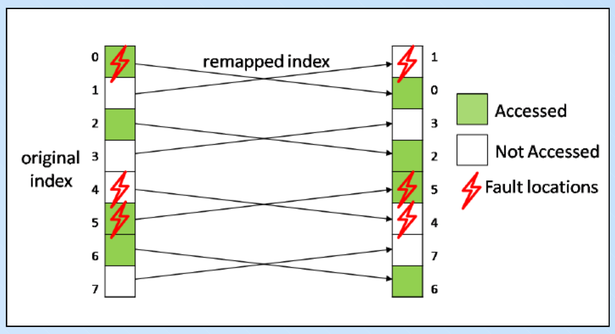
- Technology scaling enables more transistors per chip that can be used to increase processor performance
- The smaller the transistors get, the more error prone the processor becomes
- A key challenge for today's processor designers is to provide reliable operation with little or no performance degradation under the presence of faults
- Our aim is to provide solutions that mitigate the performance degradation due to faults in non-architectural structures (e.g. predictors, replacement arrays)
- We focus our study on the line predictor

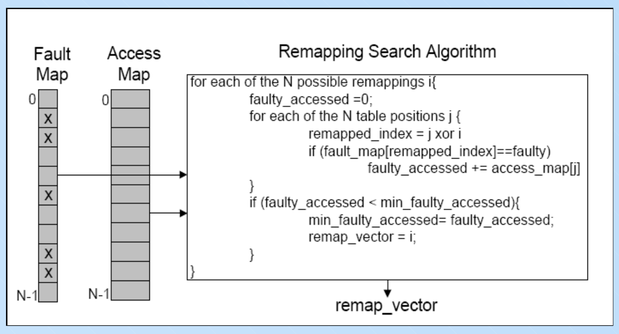
Why Protect Non-Architectural Structures

- With increasing fault frequency non-architectural resources might become a performance bottleneck
  - –Usually in high-temperature areas of the chip
  - –More susceptible to wear-out and process variation
- Some systems cannot afford delays
  - –For example time constrained real time systems
- Energy inefficient
  - –Mispredictions and incorrect replacements mean extra work for the processor
- Low voltage operation
  - –We might want to drop voltage of non-architectural structures below normal operation values to save energy
  - –This induces intermittent faults
  - –Our scheme mitigates much of the performance loss occurred due to these faults

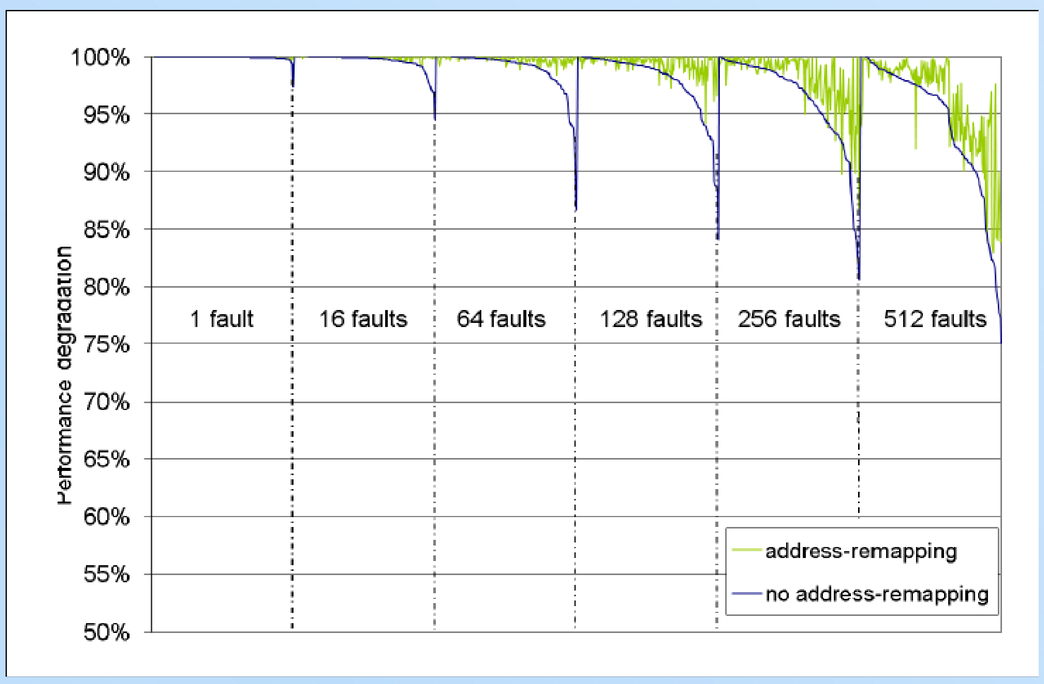
Address Remapping

- Use a built-in self test mechanism to detect faults in the line predictor and map each faulty line-predictor bit line in a fault map
  - – This is done at large intervals, for example on each system boot
- At runtime, use small counters to measure the accesses of each line predictor bit line
- Remap the line predictor bit lines in order to reduce accesses to faulty entries with high access counters
  - – Only do this if we detect frequent accesses to faulty entries
  - – Search for the best possible remapping index
- We used XOR as our remapping function





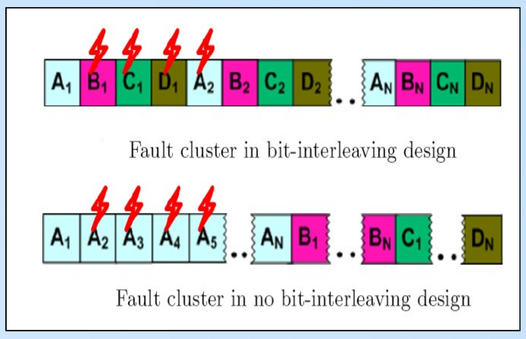
Results

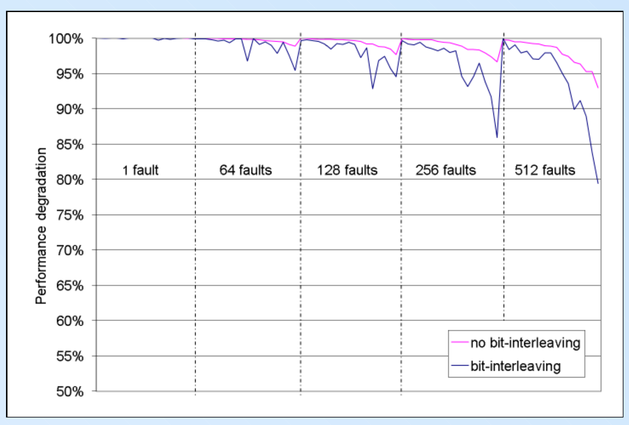


The address remapping mechanism recovers a large fraction of the performance loss occurred due to faults

Bit Interleaving

- Usually, array structures are organized in a bit interleaving manner to increase area efficiency
- In the absence of error correction codes this makes the arrays more susceptible to faults





Non-interleaved designs are more resilient against faults

Conclusion

- Faults in non-architectural structures must be addressed or they might become a serious performance bottleneck
- The use of address remapping can recover most of the performance loss occurred due to faults

Future Work

- Evaluate the effects of faults on other non-architectural structures
- Consider other remapping functions

This research is supported by the University of Cyprus, Ghent University, SARC, HiPEAC and Intel