

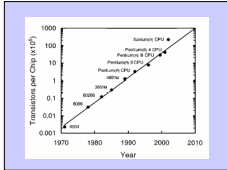
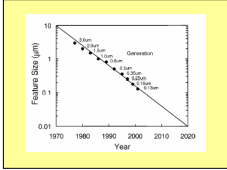
Fault-Tolerant Computing at PASCAL



University of California, Irvine



High-performance microprocessor design is challenging



- Increase issue width
- Fill pipeline stalls
- Increase clock frequency by technology scaling and increasing number of pipeline stages
- Custom circuit design



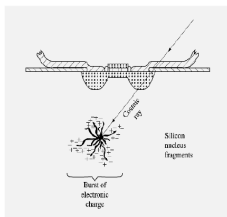
What is the billion-transistor architecture of the future?

Fault-tolerant computing – hard problem

Fault-tolerant computing is as old as the electronic digital computer: how do we build reliable computer from unreliable components?
The problem has always been here but is becoming more significant because technology scaling, architectures of growing complexities.

Radiation-induced soft errors

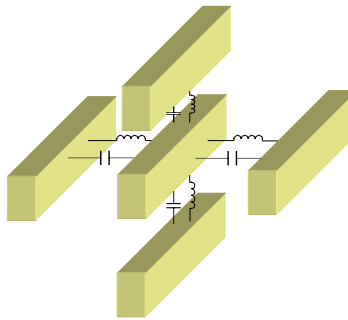
- Single Event Upsets (SEUs) caused by energetic particles are more severe because of lower device capacitance and supply voltage.
- On storage units SEUs result in bit-flips
 - On combinational logic particle strikes generate pulses



From "IBM experiments in soft fails in computer electronics (1978-1994)"

On-chip interconnect-induced crosstalk

- On-chip wires are becoming taller and more densely packed
- Increased crosstalk



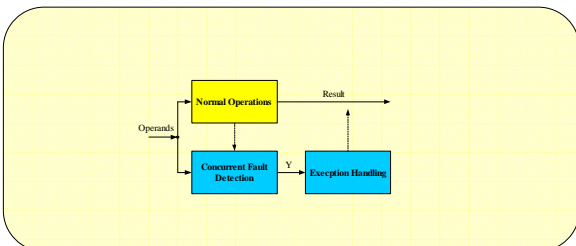
Design faults

The rapid increase in the complexity of high-performance microprocessors has led to continuing post-design discoveries of numerous design faults.

Harder problem – Can we have both high-performance and fault-tolerance?

High-performance is the ultimate goal of microprocessor designer, how to add fault-tolerant features in? We believe one way to go is dealing with faults during runtime.

Concurrent fault-detection + Exception Handler for fault-recovery



What we have done?

- Control flow error is the fault model that specifies violation of instruction sequencing
- Partitioning programs into basic blocks
- Checking for control flow error between basic blocks

