Outline

- Introduction and Terminology
- Test Generation for Single Stuck-At Faults
- Functional Testing
- Software Testing
- Design For Test (DFT)
- Built-In Self-Test (BIST)

Functional Testing

- Testing a system at a higher level of abstraction
  - If a system works at higher level, then you don't have to worry about the lower level
  - The user only sees the high level abstraction, not transistors
- For example, instead of testing for every possible stuck-at in a Pentium4, test to see if it boots Windows
  - If it boots Windows, you might believe that it is OK
- Other examples?
- Functional testing can be done at many levels
  - E.g., you could probably consider stuck-at testing to be functional if you were only worried about logic gate functionality

Why Use Anything BUT Functional Testing?

- If functional testing subsumes lower-level testing, why do we bother with low level testing for stuck-ats?
- Reason #1: Devising functional tests with good fault coverage isn't easy
- Reason #2: Fault location and diagnosis --- during fabrication, if you notice that a lot of faults are in the same place, you might be able to adjust the fabrication process to make that place less of a problem
- Any other reasons?

Validation in the Real World

"Validating the Pentium 4 Microprocessor," by Bob Bentley and Rand Gray (Intel), DSN 2001