Lecture 16: Circuit Pitfalls and Design for Test

Slides courtesy of Deming Chen

Slides based on the initial set from David Harris
Outline

- Variation
- Reliability
- Circuit Pitfalls
- Testing
- Fault Models
- Observability and Controllability
- Design for Test
- Boundary Scan

Readings: 7.1-7.3; 15.1-15.7
Variation

- Process
  - Threshold
  - Channel length
  - Interconnect dimensions
- Environment
  - Voltage
  - Temperature
- Aging / Wearout
Process Variation

- **Threshold Voltage**
  - Depends on placement of dopants in channel
  - Standard deviation inversely proportional to channel area

- **Channel Length**
  - Systematic *across-chip linewidth variation* (ACLV)
  - Random line edge roughness (LER)

- **Interconnect**
  - Etching variations affect w, s, h
Spatial Distribution

- Variations show spatial correlation
  - Lot-to-lot (L2L)
  - Wafer-to-wafer (W2W)
  - Die-to-die (D2D) / inter-die
  - Within-die (WID) / intradie

- Closer transistors match better

Courtesy M. Pelgrom
Environmental Variation

- **Voltage**
  - $V_{DD}$ is usually designed $\pm 10$
  - Regulator error
  - On-chip droop from switching activity

- **Temperature**
  - Ambient temperature ranges
  - On-die temperature elevated by chip power consumption

![Graph](image)

<table>
<thead>
<tr>
<th>Standard</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commercial</td>
<td>0 °C</td>
<td>70 °C</td>
</tr>
<tr>
<td>Industrial</td>
<td>−40 °C</td>
<td>85 °C</td>
</tr>
<tr>
<td>Military</td>
<td>−55 °C</td>
<td>125 °C</td>
</tr>
</tbody>
</table>

Courtesy IBM

[Harris01b]
Aging

- Transistors change over time as they wear out
  - Hot carriers
  - Negative bias temperature instability
  - Time-dependent dielectric breakdown
- Causes threshold voltage changes
Process Corners

- Model extremes of process variations in simulation
- Corners
  - Typical (T)
  - Fast (F)
  - Slow (S)
- Factors
  - nMOS speed
  - pMOS speed
  - Wire
  - Voltage
  - Temperature

<table>
<thead>
<tr>
<th>Corner</th>
<th>Voltage</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>1.98</td>
<td>0 °C</td>
</tr>
<tr>
<td>T</td>
<td>1.8</td>
<td>70 °C</td>
</tr>
<tr>
<td>S</td>
<td>1.62</td>
<td>125 °C</td>
</tr>
</tbody>
</table>
Corner Checks

- Circuits are simulated in different corners to verify different performance and correctness specifications

<table>
<thead>
<tr>
<th>Corner</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>nMOS</td>
<td>pMOS</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>S</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>S</td>
</tr>
</tbody>
</table>
Monte Carlo Simulation

- As process variation increases, the worst-case corners become too pessimistic for practical design
- Monte Carlo: repeated simulations with parameters randomly varied each time
- Look at scatter plot of results to predict yield
- Ex: impact of $V_t$ variation
  - ON-current
  - leakage

![Scatter plot of results](image)
Noise

Sources
- Power supply noise / ground bounce
- Capacitive coupling
- Charge sharing
- Leakage
- Noise feedthrough

Consequences
- Increased delay (for noise to settle out)
- Or incorrect computations


Reliability

- **Hard Errors**
  - Oxide wearout
  - Interconnect wearout
  - Overvoltage failure
  - Latchup
- **Soft Errors**
- **Characterizing reliability**
  - Mean time between failures (MTBF)
    - \# of devices \times\ \text{hours of operation} / \text{number of failures}
  - Failures in time (FIT)
    - \# of failures / thousand hours / million devices
Accelerated Lifetime Testing

- Expected reliability typically exceeds 10 years
- But products come to market in 1-2 years
- Accelerated lifetime testing required to predict adequate long-term reliability

![Graph showing DC lifetime 10% ion vs Vdd stress]
Electromigration

- "Electron wind" causes movement of metal atoms along wires
- Excessive electromigration leads to open circuits
- Most significant for unidirectional (DC) current
  - Depends on current density $J_{dc}$ (current / area)
  - Exponential dependence on temperature

- Black’s Equation: $MTTF \propto \frac{e^{E_a/kT}}{J_{dc}^n}$

- Typical limits: $J_{dc} < 1 – 2$ mA / $\mu$m$^2$
Self-Heating

- Current through wire resistance generates heat
  - Oxide surrounding wires is a thermal insulator
  - Heat tends to build up in wires
  - Hotter wires are more resistive, slower

- Self-heating limits AC current densities for reliability
  - Typical limits: $J_{\text{rms}} < 15 \text{ mA} / \mu\text{m}^2$
Overvoltage Failure

- High voltages can blow out tiny transistors
- *Electrostatic discharge* (ESD)
  - kilovolts from static electricity when the package pins are handled
- *Oxide breakdown*
  - In a 65 nm process, $V_g \approx 3$ V causes *arching* through thin gate oxides
- *Punchthrough*
  - High $V_{ds}$ causes depletion region between source and drain to touch, leading to high current flow and destructive overheating
Soft Errors

- In 1970’s, DRAMs were observed to randomly flip bits
  - Ultimately linked to alpha particles and cosmic ray neutrons
- Collisions with atoms create electron-hole pairs in substrate
  - These carriers are collected on p-n junctions, disturbing the voltage

[Baumann05]
Radiation Hardening

- Radiation hardening reduces soft errors
  - Increase node capacitance to minimize impact of collected charge
  - Or use redundancy
  - E.g. dual-interlocked cell

- Error-correcting codes
  - Correct for soft errors that do occur
Circuit Pitfalls

- Detective puzzle
  - Given circuit and symptom, diagnose cause and recommend solution
  - All these pitfalls have caused failures in real chips
Bad Circuit 1

- Circuit
  - 2:1 multiplexer

- Symptom
  - Mux works when selected D is 0 but not 1.
  - Or fails at low $V_{DD}$.
  - Or fails in SFSF corner.

- Principle:
  - 
  - 
  - 

- Solution:
  - Use transmission gates, not pass transistors
Bad Circuit 2

- Circuit
  - Latch

- Symptom
  - Load a 0 into Q
  - Set $\phi = 0$
  - Eventually Q spontaneously flips to 1

- Principle:
- Solution:
Bad Circuit 3

- Circuit
  - Pseudo-nMOS OR

- Symptom
  - When only one input is true, \( Y = 0 \).
  - Perhaps only happens in SF corner.

- Principle:

- Solution:
Bad Circuit 4

- Circuit
  - Latch

  ![Latch Diagram]

- Symptom
  - Q stuck at 1.
  - May only happen for certain latches where input is driven by a small gate located far away.

- Principle:
  - Series resistance of D driver, wire resistance, and tgate must be much less than weak feedback inverter.

- Solutions:
  - Check relative strengths
  - Avoid unbuffered diffusion inputs where driver is unknown
Circuit Pitfalls

CMOS VLSI Design 4th Ed.

Bad Circuit 5

- Circuit
  - Latch
  - Symptom
    - Q changes while latch is opaque
    - Especially if D comes from a far-away driver

- Principle:
- Solution:
Testing

- Testing is one of the most expensive parts of chips
  - Logic verification accounts for > 50% of design effort for many chips
  - Debug time after fabrication has enormous opportunity cost
  - Shipping defective parts can sink a company

- Example: Intel FDIV bug (1994)
  - Logic error not caught until > 1M units shipped
  - Recall cost $450M (!!!)
Logic Verification

- Does the chip simulate correctly?
  - Usually done at HDL level
  - Verification engineers write test bench for HDL
    - Can’t test all cases
    - Look for corner cases
    - Try to break logic design
- Ex: 32-bit adder
  - Test all combinations of corner cases as inputs:
    - 0, 1, 2, $2^{31}-1$, -1, -$2^{31}$, a few random numbers
- Good tests require ingenuity
Silicon Debug

- Test the first chips back from fabrication
  - If you are lucky, they work the first time
  - If not...

- Logic bugs vs. electrical failures
  - Most chip failures are logic bugs from inadequate simulation
  - Some are electrical failures
    - Crosstalk
    - Dynamic nodes: leakage, charge sharing
    - Ratio failures
  - A few are tool or methodology failures (e.g. DRC)

- Fix the bugs and fabricate a corrected chip
Shmoo Plots

- How to diagnose failures?
  - Hard to access chips
    - Picoprobess
    - Electron beam
    - Laser voltage probing
  - Built-in self-test

- Shmoo plots
  - Vary voltage, frequency
  - Look for cause of electrical failures
Manufacturing Test

- A speck of dust on a wafer is sufficient to kill chip
- Yield of any chip is < 100%
  - Must test chips after manufacturing before delivery to customers to only ship good parts
- Manufacturing testers are very expensive
  - Minimize time on tester
  - Careful selection of test vectors
Manufacturing Failures

- Metal 1 Shelving
- Metal 5 film particle (bridging defect)
- Open defect
- Spongy Via2 (Infant mortality)
- Metal 5 blocked etch (patterning defect)
- Spot defects “Co” Defect under Gate
- Metal 1 missing pattern (open at contact)

SEM images courtesy Intel Corporation
Stuck-At Faults

- How does a chip fail?
  - Usually failures are shorts between two conductors or opens in a conductor
  - This can cause very complicated behavior

- A simpler model: *Stuck-At*
  - Assume all failures cause nodes to be “stuck-at” 0 or 1, i.e. shorted to GND or $V_{DD}$
  - Not quite true, but works well in practice
Examples

Stuck-At-1
SA1 Fault

Stuck-At-0
SA0 Fault

SA1

SA0
Observability & Controllability

- **Observability**: ease of observing a node by watching external output pins of the chip
- **Controllability**: ease of forcing a node to 0 or 1 by driving input pins of the chip

- Combinational logic is usually easy to observe and control
- Finite state machines can be very difficult, requiring many cycles to enter desired state
  - Especially if state transition diagram is not known to the test engineer
Test Pattern Generation

- Manufacturing test ideally would check every node in the circuit to prove it is not stuck.
- Apply the smallest sequence of test vectors necessary to prove each node is not stuck.

- Good observability and controllability reduces number of test vectors required for manufacturing test.
  - Reduces the cost of testing
  - Motivates design-for-test
Test Example

SA1  SA0

☐ A₃
☐ A₂
☐ A₁
☐ A₀
☐ n1
☐ n2
☐ n3
☐ Y

☐ Minimum set:

Minimum set: \{0100, 0101, 0110, 0111, 1010, 1110\}
Design for Test

- Design the chip to increase observability and controllability

- If each register could be observed and controlled, test problem reduces to testing combinational logic between registers.

- Better yet, logic blocks could enter test mode where they generate test patterns and report the results automatically.
Scan

- Convert each flip-flop to a scan register
  - Only costs one extra multiplexer
- Normal mode: flip-flops behave as usual
- Scan mode: flip-flops behave as shift register

- Contents of flops can be scanned out and new values scanned in
Scannable Flip-flops

(a)  

(b)  

(c)  

Design for Testability  CMOS VLSI Design 4th Ed. 38
ATPG

- Test pattern generation is tedious
- Automatic Test Pattern Generation (ATPG) tools produce a good set of vectors for each block of combinational logic
- Scan chains are used to control and observe the blocks
- Complete coverage requires a large number of vectors, raising the cost of test
- Most products settle for covering 90+% of potential stuck-at faults
Built-in Self-test

- Built-in self-test lets blocks test themselves
  - Generate pseudo-random inputs to comb. logic
  - Combine outputs into a syndrome
  - With high probability, block is fault-free if it produces the expected syndrome
PRSG

- **Linear Feedback Shift Register**
  - Shift register with input taken from XOR of state
  - *Pseudo-Random Sequence Generator*

<table>
<thead>
<tr>
<th>Step</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Flops reset to 111
BILBO

- Built-in Logic Block Observer
  - Combine scan with PRSG & signature analysis

<table>
<thead>
<tr>
<th>MODE</th>
<th>C[1]</th>
<th>C[0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Test</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Reset</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Normal</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Boundary Scan

- Testing boards is also difficult
  - Need to verify solder joints are good
    - Drive a pin to 0, then to 1
    - Check that all connected pins get the values
- Through-hold boards used “bed of nails”
- SMT and BGA boards cannot easily contact pins
- Build capability of observing and controlling pins into each chip to make board test easier
Boundary Scan Example
TestosterICs

- TestosterICs functional chip tester
  - Designed by clinic teams and David Diaz at HMC
  - Reads your test vectors, applies them to your chip, and reports assertion failures
Testing Your Class Project

- Presilicon Verification
  - Test vectors: corner cases and random vectors
  - HDL simulation of schematics for functionality
  - Use 2-phase clocking to avoid races
  - Use static CMOS gates to avoid electrical failures
  - Use LVS to ensure layout matches schematic
  - Don’t worry about timing

- Postsilicon Verification
  - Run your test vectors on the fabricated chip
  - Use a functional chip tester
  - Potentially use breadboard or PCB for full system
Summary

- Circuits suffer from a variety of pitfalls
  - Process variation; noise; various reliability issues
- Essential to check circuits for pitfalls
  - Reliability and testing are two sides of the same coin
- Think about testing from the beginning
  - Simulate as you go
  - Plan for test after fabrication (post-silicon debug)
- “If you don’t test it, it won’t work! (Guaranteed)”