Testability

Dejan Markovic
dejan@ee.ucla.edu

Admin Info: Project / Final

- Final project report (more info on classwiki)
  - Final report (up to 6 pages), due: Fri, Dec 5 (2pm)
    - Email: ee216a@gmail.com

- Office hours during the finals week
  - Mon, Dec 8, 2:00-4:00pm (56-147E Eng-IV)

- Final exam
  - Wednesday, Dec 10, 11:30am – 2:30pm
  - Closed book, you may bring two-page notes
Overview

Reading
- W&H: Chapter 9

Introduction –
- Silicon debugging is a growing problem that accompanies the increasing complexity of current designs. This lecture discusses how chips fail, how chips are tested and debugged.
- “Design for Test and Debug” is the art of adding functionality to the chip to enhance its controllability and observability so that it can be effectively debugged and tested for correct operation.
  - Controllability: the ability to set the state of internal nodes from the chip’s input pads.
  - Observability: the ability to propagate the state of internal nodes to the chip’s output pads.

Most slides in this lecture are from:
Shannon Morton, (SGI), ISSCC 2003

Do you want unhappy customers?

- Pentium FP divider bug in 1994 cost the company $450 million dollars.
  - People got fired over this!
Technology Scaling

- **IC Scaling**: 0.7x technology shrink $\Rightarrow$ 2x increase in number of internal nodes
  - From 1$\mu$m technology to 0.1$\mu$m $\Rightarrow$ 100x increase in complexity of internal state
  - Die size also increasing $\Rightarrow$ even more states
  - Only a small *relative* increase in the number of pins available for test.
  - Longer lengths of interconnect (over a mile!)
    * More layers & tighter pitches $\Rightarrow$ more IC faults

- **Complexity**
  - A combinational logic with N inputs implies $2^N$ test vectors.
  - A sequential logic with N inputs and M states implies $2^{N+M}$ test vectors.

Cost of Testing

- Man-hours required to generate sufficient test coverage (if at all possible) is *vastly* increased.

- Testing occurs at different stages and costs differently
  - Wafer, packaged chip, board, system, field
  - 10x more expensive at each level (wafer probing is $0.1/unit)

- Each part requires more time/tester, or more testers
  - 50M units at 1sec/unit $\Rightarrow$ $5$ million/year.
  - At least $2-3$ *million* for a 1000-pin tester.
  - Reduced volume $\Rightarrow$ unable to meet demand $\Rightarrow$ loss of potential revenue

- Increased risk of shipping defective parts
  - *Unhappy customers* $\Rightarrow$ loss of ongoing revenue
Why Chips Fail?

- Process defects
- Reliability failures
- Iddq failures
- Timing and noise failures
- Soft errors
- Logic design failures

Process Defect Examples

- Missing or poorly formed via (*infant mortality*)
- Hillock causing an open in upper layer metal

*Random* and *systematic* defects
- Immediate functional failures / *infant mortality*
- Some causes of process defects
  - Dust particles, Oxide/Si defects/impurities/roughness
  - Lithographic errors, Temp & chemical composition of processes
Reliability

- Failure rates of devices follow a bathtub shape
  - Infant mortality: gross defects, poor manufacturing tolerances
  - Useful life: problems arising from wear and tear, random effects
  - Wear out: slower slope than infant side, but accelerated failures

![Bathtub curve diagram](source: M. Horowitz)

Burn-in Ovens

- Accelerate the infant mortality portion of the curve
  - Push all the parts into the "useful life" region
  - Discard the ones that die and sell the rest with high confidence

- Use burn-in ovens to heat / simultaneously exercise the parts
  - Bump up temperature and voltage to get "acceleration factors"
  - Temp held to 150-200°C and voltage to 1.5x-2x nominal (typical)

- Temperature depends on burn-in oven package solution
  - Package has a thermal resistivity, e.g. 0.24 °C/W
  - Holding oven at 125°C for 100W parts means 150°C junction temp

Courtesy: M. Horowitz
Burn-in Oven Boards

- Populate a burn-in board with your parts
  - Board exercises the parts (tests and/or power virus) during burn-in

- High-power chips strain the capacity of burn-in ovens
  - You can’t put too many 100W and 100A chips on a burn-in board!

Iddq Failures

- Excessive standby current (through S/D & gate too)
  - Beyond that expected from sub-threshold leakage
    - Pseudo-nmos structures (typically disallowed)
    - Current references (analog circuitry)
  - May indicates process defect having caused a short

- Tested during bring-up & burn-in
  - May become less relevant as leakage (sub-Vth & gate) is increasing rapidly with each technology
    - May also be swamped by large L2/L3 SRAM leakage
Timing & Noise Failures

- Unforeseen **critical path** on chip
  - Perhaps only under certain bizarre conditions
  - Example: Insufficient differential into low-swing SA’s
  - Particularly relevant to shrinks (interconnect scaling)

- **Glitch** resulting from excessive RLC noise
  - *Timing push-out* to await settling of glitch
    - Perhaps only under certain bizarre conditions
  - *Functional failure* if driven into state-holding circuitry
  - *Reliability failure* due to excessive coupling

Soft Errors - What are they?

- Alpha particle’s (He nucleus) released primarily from radioactive isotopes in lead (C4 bumps)
  - Cosmic rays of $\approx$GeV energy levels
- Collisions with Si atoms generate $e^-$/hole pairs
- $e^-$ are swept across PN junction reducing $V_d$
Soft Errors - Memory

- Memory is usually the focus
  - Dense; Low capacitance

- ECC codes are employed on large arrays to enable sufficient detection and correction.
- Parity checks used to enable sufficient detection
- C4 bumps may be prohibited over arrays, but...

*Inject current into S/D region and measure how much charge \( Q_{crit} \) is needed to flip the cell*

Soft Errors - Logic

- Logic is also prone to soft errors:
  - Dynamic nodes and latches may flip state
  - Static nodes may create a glitch that gets latched

- If ECC is employed on the arrays, even with C4 bumps over them, then logic soft-errors may govern overall FIT rate (failures in \( 10^9 \) hours)
**Logic Design Failures (Debug)**

- Incorrect wiring into a gate
- Incorrect gate in a logic function
- Incorrect algorithm in the micro-architecture
- BUBBLE ERRORS!
  - Particularly between top or 2nd level blocks

![Logic Diagram]

\[ Z = (A \cdot B + C) \cdot D + E \cdot F \]

\[ \bar{x} \cdot \bar{y} = \bar{x} + \bar{y} \]

---

**VLSI Testing**

- Testing is expensive: VLSI testers cost $1-5M
  - Volume manufacturing requires large number of testers
  - Test contributes 20-30% of the total chip cost

Types of testing

<table>
<thead>
<tr>
<th>Step</th>
<th>Error Source</th>
<th>Test Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Design flaws</td>
<td>Design verification ✓</td>
</tr>
<tr>
<td>Prototype</td>
<td>Design flaws ✓</td>
<td>Functional test ✓ (&quot;silicon debug&quot;) ✓</td>
</tr>
<tr>
<td></td>
<td>Prototype flaws ✓</td>
<td></td>
</tr>
<tr>
<td>Manufacture</td>
<td>Physical defects</td>
<td>Manufacturing test</td>
</tr>
<tr>
<td>Shipping</td>
<td>Mfg. test, transport</td>
<td></td>
</tr>
<tr>
<td>System integration</td>
<td>Same</td>
<td>Functional test +</td>
</tr>
<tr>
<td>Service</td>
<td>Stress, age</td>
<td>diagnosis</td>
</tr>
</tbody>
</table>

- Wafer / packaged chip / board / system / field  
  
*Courtesy: B. Nikolic*
Testing Tools

- Several common “external” tools
  - Probing
    - Land a probe onto a top metal square (10x10 µm)
  - Imaging
    - Infrared – detect clock transitions
    - E-beam – reflected electrons from different voltage wires is different.
  - Iddq – monitor supply current
  - FIB – focussed ion beam
    - Can cut or deposit metal to correct wiring errors (shorts or opens).

- “Internal” tools
  - Design features to aid test and debug.

Manufacturing Test

- Employed as part of the production flow to screen out defective parts
  - Test patterns applied to the die/package to test for correctness against an ideal (fault-free) model
    - May also test for basic delay faults/characterization
  - Burn-in tests to weed out parts that are prone to early failure (infant mortality) in the field
  - Explicit Design For Test structures usually required

- Chip design must already be functionally sound
Functional Test

- Employed as part of the *bring-up flow* to debug the chip’s electrical, logical, and timing functionality
  - Evolving tests from basic ROM ⇒ Icache load & executing programs that fit wholly within caches
  - To full multi-processor high-end applications in a variety of “real” user systems
  - How do you identify the cause of a program failure?
  - Explicit Design For Debug structures are helpful

- Chip design is *becoming* functionally sound

---

Debugging Concepts

- Additional knobs available to debugging
  - Raising and lowering the temperature.
    - Simple lab – freeze spray and heat gun.
  - Raising and lowering the supply voltage.
  - Increase and decrease the cycle-time.
    - Adjusting the duty cycle (clock compression)

- Common fixes
  - Slow logic – raise supply or increase cycle time.
  - Race condition – increase temperature
  - Leakage – lower supply
Debugging a Chip

- Run parts on tester and exercise the Clk shrink mechanisms
  - Move clock edges to test speedpath theories

- Also vary the voltage and frequency
  - Obtain “schmoo” plots
  - Named (and misspelled) after the Li’l Abner comic strip (1940’s)
    - One of the first schmoo plots looked round and bulbous

A “shmoo” (plural: shmoon)
Resembles a type of plot used by EEs
(who can’t spell and call it a “schmoo”)

www.deniskitchen.com

Courtesy: M. Horowitz

Schmoo

- Sweeping the supply and operating frequency (often at various temperatures).
  - Selectively done as a manufacturing test.
- Can be an excellent way to determine problems.
- Example of common schmoo

Typical

Finger:
FSM Init

Brick:
FSM Init

Wall:
Coupling
Charge sharing

Floor:
Leakage

Reverse:
Speedpaths
Leakage

Finger:
Coupling

Coupling
Manufacturing Test Flow

- **Raw Materials**: Sliced, polished, processed, according to a great design
- **Wafer Probe**: Probe card attached to ATE steps across each die: Basic functionality. Speed Paths.
- **Laser Repair**: Zap fuses to enable redundant blocks & improve yield. Repeat wafer tests. Mark bad die.
- **Packaged Parts**: Tests for more detailed functionality, speed binning.
- **Burn-in**: V/T stress over time, then re-test. Weeds out infant mortality failures.
- **System Test**: Real apps, I/O, Power, Performance, etc.

Cost of Doing DFT

- Increased area ⇒ less die/wafer & lower yield
  - Increased cost per good die ✔
  - Increased cap ⇒ hurts power & performance ✔

- Extra logic may be a part of the chip’s critical path
  - Reduced operating frequency ⇒ reduces revenue

- Test logic itself has a risk of introducing a bug
  - What tests the test logic?

- The cost of doing DFT is far easier to quantify than the cost of not doing DFT
Scan-based Test

- Scan chains offer observability and controllability
  - Observability: can stop the chip and read all the states
  - Controllability: can stop the chip and set all the states
  - Can trace back to find source of errors

Building Scan Chains

- Idea: add parallel path to each flip-flop
  - Extra capacitance / area (typically <5% of the total chip area)
  - Ensure scan inputs can overwrite the state
  - Ensure scan doesn’t interfere with regular operation (backwriting)
  - Trend is to have scanable flip-flops (libraries typically have scan FFs)
Fault Models

- Defects manifest in a variety of ways and may require the application of different circuit models to test for their presence.

<table>
<thead>
<tr>
<th>Fault Model</th>
<th>Effect on Circuit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Stuck-at Line</td>
<td>Single logic node stuck at 0 or 1</td>
<td>Most common -- focus of this section</td>
</tr>
<tr>
<td>Multiple Stuck-at Line</td>
<td>Multiple logic nodes stuck at 0 or 1</td>
<td>Vast majority covered by SSL model [13]</td>
</tr>
<tr>
<td>Stuck-open</td>
<td>Logic node floats (X)</td>
<td></td>
</tr>
<tr>
<td>Bridging</td>
<td>Logic node becomes AND(x,y) or OR(x,y) or X depending on drive strengths</td>
<td>May not be very well covered by SSL (83/95 to 51/98) [14]</td>
</tr>
<tr>
<td>Delay</td>
<td>Gate or Path delay is increased</td>
<td></td>
</tr>
<tr>
<td>Coupling</td>
<td>Transition on node X causes Delay Fault on Y, or alters function Y=F(x,y)</td>
<td>bitlines, buses, register files</td>
</tr>
<tr>
<td>Pattern-Sensitive</td>
<td>Complex space-time dependence</td>
<td>RAM arrays</td>
</tr>
</tbody>
</table>

> 98.5% coverage desired

ATG Time

- Time required to generate a test vector set for all SA faults is a function of:
  - ATG technique & heuristics for decision-making
  - Number of Primary Inputs & Outputs
  - Number and size of sequential structures
  - Number of equivalent SA faults
  - Depth of logic from PI to PO (esp. for sequential!)

- Circuits with \(\approx 10^6\) gates or \(\approx 10^3\) latches may be too large to test with suitable SA coverage and in a reasonable amount of time.
Scan Methodologies

- **Full Scan**: Every latch in the design is a scan latch
  - Do NOT scan simple pipeline stages

- **Partial Scan**: A selection of latches are scannable
  - Where low SA coverage is identified
  - Sequential logic: counters, data forwarding paths
  - Important data buses: PC, load/store bus

- **Scan Islands**: No scan within blocks, but a ring of scan latches on the I/O surrounds the block
  - More applicable to debug
  - Good for shrinks

AMD’s K6 Flip-Flop

- Non-overlapping scan clocks ⇒ no race & looser to route
- No additional logic in the D→Q path (via ckplse) ⇒ good performance

CLK must be inactive during scan
Intel's McKinley Flip-Flop

- Similar in principle to the AMD scan latch, but with a single clock
- No additional logic in the D→Q path (via ckplse), but o/p has extra load
- Scan operation results in true dynamic nodes ⇒ need to be cautious of noise, and a minimum scan rate is necessary due to leakage

Self-Test

- Becoming more important with increasing chip complexity and larger modules
Built-In Self-Test (BIST)

- The capability of a circuit to test itself
  - Minimal external requirements (ck, si, so, control)

Pros | Cons
--- | ---
At-speed testing of circuitry | Small area/delay penalty
Removes (or reduces) time and effort required for ATG | May be difficult for faults insensitive to random patterns (64-bit NOR)
Reduces tester time & ports, thereby saving $$$ | Aliasing in output compression introduces risk in error detection
Independent of fault model | |

General BIST Architecture

- Pattern Generator *may* need to be initialized
- Error Status could be as simple as PASS / FAIL

- One BIST controller may govern multiple CUTs
  - This interfaces to the tester
  - I/O to/from controller is serial to reduce wiring
4-bit LFSR as a Pattern Generator

- All XOR functions can also be moved to the far RHS of LFSR
  - Enables regularity in datapaths
  - Example on next slide
- Easily incorporated into scan chain

Characteristic Polynomial: \( X^4 + X^2 + 1 \)

<table>
<thead>
<tr>
<th>#</th>
<th>p4</th>
<th>p3</th>
<th>p2</th>
<th>p1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Signature Analyzers

- **One’s counter**: n-bit counter x m-bit inputs
- **Transition counter**: n-bit counter x m-bit inputs
- **Single Input LFSR**: n-bit LFSR x m-bit inputs

- **Multiple Input LFSR**: m-bit LFSR
Spare Gates

- Post-silicon edits can be done using Focused Ion Beams
  - Remove or add wires

- FIB cannot add new devices, but
  - Designers can throw some extra devices in the layout
  - Need to put them in the schematics too (remember, LVS…)

- Spare gates are basic cells with grounded inputs
  - They don’t do anything normally (except take up space)
  - You can insert them using a FIB edit layer
  - Mixture of Inv, Nand, Nor, Flops…
  - Plan on inserting these in your blocks, wherever you have room
  - Some companies call them “happy gates”

Electronic “Optics” Can Look at Chips

- Scanning electron microscope looks at chips in a vacuum
  - Useful for defect analysis, not really for tests during chip operation

Courtesy: M. Horowitz
E-beam Probing and Controlling

- E-beam probing is a technique that requires face access
  - Shoot electrons at the chip and measure reflected electrons
  - Grounded metals look bright; high-voltage metals look dark
  - Can probe metals this way to find out their voltages
  - Can also pulse e-beams at higher energy to charge up nodes
    • Mild form of controllability to go along with observability

Focused Ion Beam (FIB) for Chip Edits

- FIB allows post-fabrication edits on silicon
  - Used to check if a proposed fix will actually work
  - Very expensive (~$400/hr), so don’t do it unless you need to
    • Usually 3-5 hours per “normal” fix / one chip at a time
- FIB edits can be additive or subtractive
  - Cut wires or lay down new wires
- FIB used to be from the top of the chip only
  - Today can also be used for backside FIB (for flip-chip)
FIB Repair

- FIB is typically used to etch & deposit metals to make repairs for bug fixes. Example:
  - Through your debug features, you believe the signal below needs to be AND’ed with another signal

![Diagram of signal processing](image)

- Simulations seems to confirm this, but to be sure you’d like to try the repair on a real part to see if it truly does fix the bug
- *Spare gates need to be included in the design*

FIB Etch & Deposition Process

- FIB image of metal lines to be connected
- CAD overlay to identify locations of etch & deposition
- Final view of repair after FIB with chemical gases

*Images courtesy of Accurel*
Example of a FIB Job

- FIB milling and lifting of sample

Another FIB Example

Source: Stinson, Intel
**Testing in University Environment**

*I/O hardware library, automated FPGA flow*

- I/O lib
- RTL
- Custom tool 3
- FPGA backend

- Simulink
- Hw lib

- Custom tool 2
- ASIC backend

- Power Area

*FPGA implements ASIC logic analysis*

[D. Markovic, C. Chang, B. Richards, H. So, B. Nikolic, R.W. Brodersen, CICC’07]

---

**FPGA Based ASIC Verification**

- ASIC
- I/O
- TB

*Goal: use Simulink testbench (TB) for ASIC verification*

- Develop custom interface blocks (I/O)
- Place I/O and ASIC RTL into TB model

*Simulink implicitly provides the testbench*
Custom Interface Blocks (I/O)

- Sw / Hw interfaces
  - Regs, FIFOs, BRAMs
- External interfaces
  - GPIO ports
- A/D & D/A
  - Analog subs.
- Debugging
  - Signal gen.
  - Hw scope

Fully automated RTL flow for ASIC verification

Simulink Test Model
Initial Verification Strategy

- Testbench model on the FPGA board
  - Test vectors entered through RS232
- Block read / write operation
  - Custom `read_xps, write_xps` commands

*Real-time performance bounded by GPIO*

---

Example: SVD Test Model

*Emulation-based ASIC I/O test*
Example: FPGA Based ASIC Verification

Real-time at-speed ASIC verification

Test Example: Measured Functionality

4x4 MIMO channel tracking

Up to 10 b/s/Hz with adaptive PSK
FYI: EE216B / Spring 2009

Timed dataflow

DSP algorithm

Architecture Optimization

Speed Power Area

SysGen
B-box

HDL

FPGA backend

Hardware co-simulation

ASIC backend

EEM216A

EEM216A  /  Fall 2008
D. Markovic  /  Slide 55