Fundamentals of Semiconductor Test for Physical Failure Analysis

Dr. Burnell G. West, IEEE Life Fellow
Chief Architect, Credence Systems
Fundamentals of IC Test: Course Outline

- **Session 1 – The Structure of a Semiconductor Test**
  - What to test? Why? What tests work? How do tests work?
    - Digital, Mixed Signal, and Analog Tests
    - Functional, Structural, and Alternate Tests
  - ATE Architecture and test execution
    - Patterns, failure detection and response, logging
    - Test variables – voltage, frequency, DUT temperature

- **Session 2 – Applying Tests for Physical Failure Analysis**
  - What goes on when a device fails a test?
    - Defects vs Parametric Failures (vs Design Errors)
  - How do we isolate, locate, and exhibit a defect?
    - Analyzing and modifying tests to observe a failure mode
    - Editing physical devices to demonstrate cause
    - The role of ATE in physical failure analysis – case studies

- **References**
Semiconductor Technology Overview

Process Management  Defect Detection

Pattern  Inspect  Bake  Inspect  Passivate  Probe  Singulate  Package  Test

Metalize  Implant  ?

FA  FA  FA

END USER

Semiconductor Fabrication Data Network

Wafer/Device Data
Why Test?

- Design is not perfectly reliable
  - Test can help detect and locate design errors ("bugs")
  - Test can help establish design margins

- Manufacturing is not perfectly reliable
  - Test can establish performance limits
  - Test can detect process excursions

- Things break - *where was the weak link?*
- Things wear out - *what “couldn’t take the heat”?*
- Nobody wants to *ship* bad product
- Nobody wants to *build* bad product
IC Test and Measurement Objectives

- Design debug
  - Circuit edit to support debug
- Design validation
- Device characterization

- Defect detection
  - Faulty circuit built-in repair
- Defect isolation
- Infant mortality acceleration
- Prompt process feedback

- Quality assurance through product lifetime
- Overall test cost control
  - What to test and when to test it
  - Minimize the cost of each test applied
  - Maximize the return from each test

*Test cost reduction cannot compromise test quality*
What Kinds of Test?

Launch-capture test?
Structural test?
IDDQ test?
DC parametric test?
Static Functional Test?
Analog / mixed signal test?
Adaptive test?
At-speed functional test?
Low-V functional test?
On-line test?

DUT
Test Coverage and Escapes

Sematech Study S121 (1997)

- **Scan**: 99.7% fault coverage
- **FUNC**: 52% fault coverage
- **IDDQ**: 1478 “non-operational” (of 1764 IDDQ failures; 2147 total)
Test in Physical Failure Analysis

- Observation is key
  - LVP processes
  - Photon emissions
  - Other electromagnetic or photo-optic effects

- Observation takes TIME
  - Effects are weak
  - Noise levels are large

- Tests must be LOOPED
  - Key to satisfactory defect phenomena observation
Test Operation Overview

Test Controller
- Install test
- Start test

Test Instrument
- Receive and organize test sequence/patterns
- Transmit next test data
- Transmit required test clocking
- Receive and evaluate test results

At the I/O Boundary
- Test Setup
- Test Execute
- Execute test

Inside the Chip
- Results

Evaluate test result
- Evaluate test result
- Log fail
- Identify next test data set

Incorporate the PFA Loop
- raw data
- pass
- fail
- pass
- fail
The Relevance of Digital Test

- Most of today’s devices (even SoC) are “mostly” digital
- Digital activity stimulates devices or evaluates results for almost all test types, including
  - Digital functional tests
  - Scan-based structural tests
  - Launch-Capture tests
  - BIST tests
  - IDDQ tests
  - Memory tests (and repairs)
  - Adaptive tests
The Structure of Digital Test Processes

- Principal use – sorting

- Ancillary use – defect isolation and PFA

- Tests are not necessarily designed with PFA in mind
  - So, how are tests designed?
  - Motivation, Implementation, and Adaptation to PFA
Digital Test – the classic model

Allows tests to be written simply as tables of “vectors”
with drive and strobe timing abstracted
Digital Test “Vectors”

- **Test Pattern**: 01001011 : xxxxxxxx
- **Test Vector Number**: 1 - 11001010 : xxxx0110
- **Stimulus Functional Data**: 00110101 : 11001101
- **Response Functional (or Expect) Data**: 10100101 : 01001011
- **F-data**

```
0 - 01001011 : xxxxxxxx
1 - 11001010 : xxxx0110
2 - 00110101 : 11001101
3 - 10100101 : 01001011
4 - 11010001 : 0010zzzz
5 - 00101101 : 0011zzzz
6 - 11010010 : xx1z0x1z
7 - 11010010 : xx1z0x1z
```
Digital Test “Vectors”

| 0 | 01001011 : xxxxxxxx |
| 1 | 11001010 : xxx0110 |
| 2 | 00110101 : 11001101 |
| 3 | 10100101 : 01001011 |
| 4 | 11010001 : 0010zzzz |
| 5 | 00101101 : 0011zzzz |
| 6 | 11010010 : xx1z0x1z |

fail datalog
in<0:7>out<0:7>

Digital Test System

IEEE
Reliability Society
Digital Test “Vectors” and STIL

- Most testing, today and for the foreseeable future, will be digital
  - Unambiguous specification of complex test requirements
  - Unambiguous evaluation of pass or fail result
  - *Highly computable* – growth for multiple generations

- Non-digital requirements can have digital manifestations –
  - ADC tests, BIST structures, etc

- Hence, STIL – *Standard Test Interface Language* – IEEE 1450
  - Standard description of digital patterns
  - Standard description of logic transitions – “events”
  - Standard description of basic digital test waveforms
Where Vectors Come From

- Stored Pattern ("functional") Vectors
  - Generated by hand, simulation, sometimes ATPG

- SCAN Vectors
  - Generated mainly by ATPG; require scan-chain DFT
  - Used for structural tests
  - Used for path delay tests
  - Requires launch-capture clocking

- Algorithmic generation
  - Used mainly for memory arrays

- Built-in Self Test (BIST) generators
How STIL *Event Streams* come from Vectors

- Variety of ways to represent digital tests
- All result in specific activity at each DUT pin
- Activity at each DUT pin is a *sequence of events*
  - Input pins: D1@time, D0@time
  - Output pins: T1@time, T0@time, X@time
  - I/O pins: D1@time, D0@time, DZ@time, T1@time, T0@time, TZ@time
- F-data translates to D1’s, D0’s, DZ’s or T1’s, T0’s, TZ’s, depending on I/O Definition
Event sequences to test a flip-flop

D event stream: DF@0
C event stream: DF@LE, DF_@TE
Q,QN event streams: TF@SLE, X@STE (window strobe)
Q,QN event streams: TF@strobe_time (edge strobe)

-- but we must allow for inaccuracies

Strobe Leading Edge (SLE)
Strobe Trailing Edge (STE)

D Q
C QN

F=0 F=1 F=0

inputs D C
output Q
output QN

window strobes
edge strobes

test pattern

0 - 01:01
1 - 10:10
2 - 01:01
Edge Placement Accuracy (EPA)

Event Time $T_E$

inputs $D$ $C$

output $Q$ $QN$

window strobes $F=0$ $F=1$ $F=0$

edge strobes

test pattern

0 - 01:01
1 - 10:10
2 - 01:01
Determining Edge Placement

- Edge placement is a combination of three terms
  - Program value within each vector
  - Accumulated vector start offset
  - Calibration corrections
    - Static systematic errors
    - Pattern-dependent systematic errors
    - Clock-dependent systematic errors

- So, how is it implemented?
How a Tester Timing System Works

Event sequences from F-data and Vector Type
- each pin has its own event sequence per vector

Event times adjusted by period offset
- transform vector time to clock plus interpolation
- system clock *decoupled* from vector rate

Event times further corrected by cal offset
- corrections for path length variation (skew)
- corrections for dynamic errors
Vector Time Transformation

Transform 1: Add accumulated period offset to program time
Transform 2: Add calibration offset based on event type
Transform 3: Determine clock cycle containing event time and compute interpolation value

When clock cycle occurs, trigger interpolation delay circuit...
Typical Tester Timing Path

Event Sequence Generator

F-data

Event Types

Vector

Event Times

Type

Interpolator

drive markers

Formatter

Interpolator

strobe markers

Formatter

DHI

DINH

Socket

Loadboard

Pogo

PE cable

CLOCK

fanout

MAINFRAME (moving to test head)

DUT

Time-critical paths

Time-critical devices

MAINFRAME (moving to test head)

Time-critical paths

Time-critical devices

IEEE
Some Timing Error Sources

Dominant timing errors are

- Static systematic errors
- Tester clock cycle dependent systematic errors
- Pattern dependent systematic errors
- Noise

Systematic errors are (in principle) correctable

*Noise is not correctable* (but can be averaged)

---

**Can these errors be calibrated?**

- Clock generation noise: no
- Clock distribution noise: no
- Interpolator non-linearity: yes
- Marker noise: no
- Formatter noise: no
- Timing signal path dynamics: yes
- PE timing non-linearity: yes
- PE-DUT signal path dynamics: some
- Source-clocked DUT tracking error: some
- Calibration error: NO!!
Variable Test Rates

- Pin event timing can change on the fly
  - Different vector types may have different timing

- Test rates can also change on the fly
  - For match synchronization
  - For speed testing
  - For characterization and debugging
Variable Test Rates – as used for debugging

Slower

Faster
Recent Major Technology Shifts

- **Clock Multiplication**
  - High-end uProc clocks multiply input rate
  - Internal PLL controls timing (not ATE)

- **High-speed I/O’s use local clocking**
  - Forwarded Clocks
  - Reference Clocks
  - Embedded Clocks
Variable Test Rates – with smart PLL clock

Slower

Faster

Launch

Capture
Fundamentals of IC Test: Summary of Session 1

- Manufacturing processes produce unshippable defects
  - Wide variety of tests can be applied
  - Studies show different tests expose different defects
    - Difficult to know *a priori* which tests will be most effective
  - Effective ATE must apply a variety of tests in multiple conditions

- Delay defects more predominant in advanced IC processes
  - Agile timing in ATE helped identify and locate delay defects
  - Higher speeds in modern devices challenge ATE agility
    - Delta timing moving on-chip

- Yield management demands quick process feedback
  - To be addressed in Session 2
Fundamentals of IC Test: Course Outline

Session 1 – The Structure of a Semiconductor Test
- Why test? What tests work? How are tests applied?
  - Digital, Mixed Signal, and Analog Tests
  - Functional, Structural, and Alternate Tests
- ATE Architecture and test execution
  - Patterns, failure detection and response, logging
  - Test variables – voltage, frequency, DUT temperature

Session 2 – Applying Tests for Physical Failure Analysis
- What goes on when a device fails a test?
  - Design Errors vs Defects vs Parametric Failures
- How do we isolate, locate, and exhibit a circuit problem?
  - Analyzing and modifying tests to observe a failure mode
  - Editing physical devices to demonstrate cause
  - The role of ATE in physical failure analysis

References
Physical Failure Analysis: Test Loops Needed

- Defect Localization Requires Imaging
  - Frontside or backside imaging of pattern activity
  - Waveforms or circuit activity, heating, etc.

- Virtually all imaging processes require stable repeatable tests
  - Hundreds of thousands to many millions of repetitions to collect
  - Time-resolved emissions require extreme stability
    - Waveform capture with 30-ps resolution is very demanding

- How does ATE Generate test loops?
  - Functional patterns
  - Scan-based patterns
Test Operation Overview – Implementing a Loop

Test Controller
- Install test
- Start test

Test Instrument
- Receive and organize test sequence/patterns
- Apply next test vector
- Compare DUT response to table
- Execute test

Inside the DUT

- pass/fail
- fail

Log fail

Incorporate the PFA Loop
IC Test and Measurement Objectives

- Design debug
  - Circuit edit to support debug
- Design validation
- Device characterization

- Defect detection
  - Faulty circuit built-in repair
- Defect isolation
- Infant mortality acceleration
- Prompt process feedback

- Quality assurance through product lifetime
- Overall test cost control
  - What to test and when to test it
  - Minimize the cost of each test applied
  - Maximize the return from each test

*Test cost reduction cannot compromise test quality*
Why Integrated Circuit Edit?

- Vast variety of sophisticated EDA tools help designers debug their code
- Test structure are implemented into IC for post fabrication device inspection, performance & reliability assurance
- Design can be tested, simulated & debugged before fabrication
- FAB implements test structures for process control & verification
- Every step starting from Concept to Fabrication is well monitored, thoroughly tested and verified multiple times.
- “Our Silicon is going to be perfect…”
- So why CIRCUIT EDIT ????????
Need of Circuit Edit

- Test finds the device doesn’t work
  - Testing may identify design failure source
- Simulations may verify proposed fix
- BUT low confidence in making an expensive mask change
- It may take weeks to months before an ECO’d IC would be available to validate design change.
- Is it going to work ???

Therefore, FIB Circuit Edit is the solution for:
- Validating design change on Tester
- Proceeding to Mask change with confidence
- Getting working devices to customers ASAP

CE, vital for fast & cost effective Design-to-Market solution
CE Technology Roadmap

- Edit time increases as number of routing layers increase (from 3 to 10 in last 10 years) with technology nodes
- CE tools require constantly developing hardware, techniques & chemistries to address multi-layer deep sub-micron ICs
- CAD navigation is vital for effective CE solutions
Case Study

- Flipchip, 90nm Cu/low-k process
- Circuit issue was that an incorrect power supply Vdd1 was on three master phase locked loop controllers (PLLCs), i.e., the pull-up voltage on a logic gate was incorrect.
- Required connecting an M1 line to Vdd2 on M3 and cutting M1 line to isolate Vdd1 supply.
- The low beam current used for this operation made it successful.
- Once Test proved the edit worked, edit was repeated on the other 2 PLLCs & Tested
- In all eight devices were edited and Tested good.
CE Planning for in Silicon Validation

- Once CE is part of debug then 2 doors open:
  - CE to fine tune analog circuitry
  - Case study: Design planned for edit at critical points
    - M1 resistors could be added to fine-tune design
    - Test found problem in area where expected
    - CE fix employed & design advanced
      - Product development time reduced

- May not understand where risks are, but EDA tools should help

- Circuit edit enables validation of next silicon
  - When completed design team moves on to the sequel
  - Production silicon becomes prototype for new design
  - Edits validate assumptions going into new design before first silicon
IC Test and Measurement Objectives

- Design debug
  - Circuit edit to support debug
- Design validation
- Device characterization

- Defect detection
  - Faulty circuit built-in repair
- Defect isolation
- Infant mortality acceleration
- Prompt process feedback

- Quality assurance through product lifetime
- Overall test cost control
  - What to test and when to test it
  - Minimize the cost of each test applied
  - Maximize the return from each test

*Test cost reduction cannot compromise test quality*
Yield vs maturity vs complexity

Yield (%)

Relative Device Complexity

Process Maturity (Quarters)

IEEE

Reliability Society
New Technologies – New Problems

- New physical designs
  - 130 nm, 90 nm, 65 nm...
- New materials
  - SOI
  - Low K dielectric
  - Copper
  - Strained silicon

- Multi-metal interconnects

Increasing differences between simulation models and reality are making it ever more impractical to simulate fully new designs.

Timing Debug

- Problem: Motorola (Freescale) device
- Scan chain was failing at certain speeds in new SOI design
  - Structural diagnostic software was not applicable
  - Scan pattern loop length: 60 microseconds
  - Power supply: 1.8V

- EmiScope Methodology
  - Measure waveforms at key nodes to identify root-cause of failure
- C1_CLK turned off too late to block the falling transition at DIN
- EmiScope timing measurement identified a race condition
- Changing C1_CLK timing solved the problem
Resistive Interconnect Localization

- Can use imaging tools to localize a wide variety of faults
  - Traditional “static” and “hard” faults common below 250 nm
  - Design and “soft” faults increasingly common below 90 nm
Resistive Gate Localization

- Localize and Characterize interconnect faults

Design Error – Transistor Mismatch

- Design error of mismatched transistor in analog circuit
- Used EmiScope to localize fault
- Simulated observed waveforms to identify root-cause
  - Good device to bad device comparison aided analysis
Resistive Via Localization

- Information related to failure(s) was used in fault diagnosis process
  - Synopsys TetraMAX and Cadence Encounter Test Diagnostics were used to localize fault
Resistive Via Localization (cont’d)

Failure was likely related to the 2 NOR Gates
Resistive Via Localization (cont’d)

- Probe results clearly identified failing net
- Based on results, one of two vias was point of failure
Logic Debug

- Test results indicate the PAD is high while expected value is low
- Use logic state tracing to identify source of failure
## Logic Debug (cont’d)

![Logic Diagram](image)

<table>
<thead>
<tr>
<th>EmiScope Data</th>
<th>Reconstructed Logic</th>
<th>Simulated Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Typical Post-Silicon Problems

- Yield is lower than required on new technology
  - Inability to fully model circuit behavior
  - Inaccurate design models, especially for timing and jitter

- Resistive failures
  - Incomplete vias, electromigration, metal bridges, etc.

- Design errors
  - Over 50% of design time is spent in verification
  - Yet – design errors are still common

- Incomplete success with ATPG methodologies
  - Incomplete test coverage
  - Incomplete fault coverage
  - Scan chain failures

They are expected to get worse below 90 nm
1st Si Failures - Design Debug Issues

1st Si Error/Flaw Distribution

<table>
<thead>
<tr>
<th>Issue</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic or Functional</td>
<td>67%</td>
</tr>
<tr>
<td>Analog Circuit</td>
<td>35%</td>
</tr>
<tr>
<td>Noise</td>
<td>29%</td>
</tr>
<tr>
<td>Slow Path</td>
<td>28%</td>
</tr>
<tr>
<td>Clocking</td>
<td>25%</td>
</tr>
<tr>
<td>Yield</td>
<td>23%</td>
</tr>
<tr>
<td>Mixed-Signal Interface</td>
<td>21%</td>
</tr>
<tr>
<td>IR Drops</td>
<td>20%</td>
</tr>
<tr>
<td>Race Condition</td>
<td>17%</td>
</tr>
<tr>
<td>Power</td>
<td>17%</td>
</tr>
<tr>
<td>Firmware</td>
<td>13%</td>
</tr>
<tr>
<td>Other</td>
<td>4%</td>
</tr>
</tbody>
</table>

At least 61% of new designs (or shrinks) required a respin

Number of Silicon Spins

- 1
- 2
- 3
- 4 or more

Sample of 251 designs from North America completed in 2002

*Collett International 4/02

Today’s IC performance requirements and leading-edge processes make it very difficult to separate design errors from process margins

*Collett International Research Inc. from Electronic Business Jun/03
EFFECTIVE TOOLS ARE STILL NOT AVAILABLE!!

Designs Having One or More Re-spins by Type of Flaw
- Logic or Functional (67%)
- Analog Circuit (35%)
- Noise (29%)
- Slow Path (28%)
- Clocking (25%)
- Yield (23%)
- Mixed-Signal Interface (21%)
- IR Drops (20%)
- Race Condition (17%)
- Power (17%)
- Firmware (13%)

COULD D.V. HAVE CAUGHT THEM ALL??

Collett International Research, April 2002
The Debug Challenge

- Huge quantities of circuitry
- Massive volumes of data to manage/analyze
- Parametric deviations causing more device failures
- Data access limited by IO bandwidth
  - faster data access = shorter debug time
- Desired data flow for debug
  - (maybe) scan in to establish starting state of entire device
  - pattern execution with IO established by pattern
  - (maybe) scan dump together with standard IO for analysis
- Multi GBPS data rate => non-deterministic patterns
  - data collection must cope with varying data
  - cannot accept fixed-protocol solutions
    - reconfigure ATE for specific protocols
    - dump bit patterns for later software protocol analysis
Differing Debug Viewpoints

- **Tester’s-eye View**
  - dataflow – both in and out
  - input waveforms
  - output waveforms

- **DUT’s-eye View**
  - high-level structure
  - RTL
  - logical circuit detail
  - topology
  - physical implementation


```cpp
#include "systemc.h"

void fft::entry()
{
    float sample[16][2];
    unsigned int index;
    while(true)
    {
        data_req.write(false);
        while( index < 16 )
        {
            data_req.write(true);
            wait_until(data_valid.delayed() == true);
            sample[index][0] = in_real.read();
            sample[index][1] = in_imag.read();
            index++;
            data_req.write(false);
            wait();
        }
        index = 0;
        //Calculate the W-values recursively
        w_real = cos(theta);
        w_imag = -sin(theta);
        while(index < len-1)
        {
            w_temp = w_rec_real*w_real - w_rec_imag*w_imag;
            w_rec_imag =  w_rec_real*w_imag + w_rec_imag*w_real;
            w_rec_real = w_temp;
            W[index][0] = w_rec_real;
            W[index][1] = w_rec_imag;
            index++;
        }
        //Begin Computation
        stage = 0;
        len = N; incr = 1;
        while (stage < M)
        {
            len = len/2;
            while(i < N)
            {
                sample[index][0] = tmp_real;
                sample[index][1] = tmp_imag;
                i = i + 2*len;
            }
            stage++;
        }
    }
}```
#include "systemc.h"

void fft::entry()
{
    float sample[16][2];
    unsigned int index;
    while(true)
    {
        data_req.write(false);
        while(true)
        {
            data_req.write(true);
            while( index < 16 )
            {
                sample[index][0] = in_real.read();
                sample[index][1] = in_imag.read();
                index++;
            }
            index = 0;
            //Calculate the W-values recursively
            w_real = cos(theta);
            w_imag = sin(theta);
            w_rec_real = w_temp;
            w_rec_imag = w_real*w_imag + w_rec_imag*w_real;
            data_req.write(false);
            while( index < 16 )
            {
                sample[index][0] = in_real.read();
                sample[index][1] = in_imag.read();
                index++;
            }
            //Begin Computation
            stage = 0;
            len = N; incr = 1;
            while (stage < M)
            {
                len = len/2;
                while( i < N )
                {
                    sample[index][0] = tmp_real;
                    sample[index][1] = tmp_imag;
                    i = i + 2*len;
                }
                stage++;
            }
        }
    }
}
DUT’s-eye View: Logic Circuit Detail

```c
#include "systemc.h"

void fft::entry()
{
    float sample[16][2];
    unsigned int index;
    while(true)
    {
        data_req.write(false);
        while( index < 16 )
        {
            data_req.write(true);
            wait_until(data_valid.delayed() == true);
            sample[index][0] = in_real.read();
            sample[index][1] = in_imag.read();
            index++;
            data_req.write(false);
            wait();
        }
        index = 0;
        //Calculate the W-values recursively
        w_real =  cos(theta);
        w_imag =  -sin(theta);
        while(index < len-1)
        {
            w_temp = w_rec_real*w_real - w_rec_imag*w_imag;
            w_rec_imag =  w_rec_real*w_imag + w_rec_imag*w_real;
            w_rec_real = w_temp;
            W[index][0] = w_rec_real;
            W[index][1] = w_rec_imag;
            index++;
        }
        //Begin Computation
        stage = 0;
        len = N; incr = 1;
        while (stage < M)
        {
            len = len/2;
            while(i < N)
            {
                sample[index][0] = tmp_real;
                sample[index][1] = tmp_imag;
                i = i + 2*len;
            }
            stage++;
        }
    }
}
```
DUT’s-eye View: The Circuit Debug Process

```
#include "systemc.h"

void fft::entry()
{
    float sample[16][2];
    unsigned int index;
    while(true)
    {
        data_req.write(false);
        while(index < 16)
        {
            data_req.write(true);
            wait_until(data_valid.delayed() == true);
            sample[index][0] = in_real.read();
            sample[index][1] = in_imag.read();
            index++;
            data_req.write(false);
            wait();
        }
        index = 0;
        //Calculate the W-values recursively
        w_real = cos(theta);
        w_imag = -sin(theta);
        while(index < len-1)
        {
            w_temp = w_rec_real*w_real - w_rec_imag*w_imag;
            w_rec_imag = w_rec_real*w_imag + w_rec_imag*w_real;
            w_rec_real = w_temp;
            W[index][0] = w_rec_real;
            W[index][1] = w_rec_imag;
            index++;
        }
        //Begin Computation
        stage = 0;
        len = N; incr = 1;
        while (stage < M)
        {
            len = len/2;
            while(i < N)
            {
                sample[index][0] = tmp_real;
                sample[index][1] = tmp_imag;
                i = i + 2*len;
            }
            stage++;
        }
    }
}
```
Challenges of Advanced Product FA / Debug

- Short channel – 90 nm and 65 nm design rules
  - Image and spatial resolution
  - Short channel modeling is difficult
- Dropping operating voltage levels
  - Emission signals are weaker
  - Lower productivity and effectiveness of solutions
- Materials and process challenges
  - Resistive shorts / bridges
  - Higher leakage problems
- Multiple metal interconnect
  - > 6 metal layers – “front side obstruction”
  - Capacitive effects and cross talk; difficult modeling

Need for better resolution & node level probing
Need for better sensitivity
Need for better sensitivity
Need for backside image resolution & node level probing

Physical, node level debug and FA is essential
Ever increasing performance / sensitivity is most important
Smaller process nodes (130 nm & below) are becoming more sensitive to higher R bridges.

Increasing number of resistive defects causing more Vdd, temp, & freq sensitive fails (*soft failures*).

Localizing resistive defects is very difficult using conventional techniques (mech probes).
## Effective Resistive Defect Localization Techniques

### GlobalScan Laser Stimulation Applications

<table>
<thead>
<tr>
<th>Laser λ (Effect)</th>
<th>Static Laser Stimulation (SLS)</th>
<th>Dynamic Laser Simulation (DLS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bias: CV Measure: ΔI</td>
<td>Bias: None Measure: ΔV or ΔI</td>
<td>Monitor Pass/Fail</td>
</tr>
<tr>
<td>Bias: CI Measure: ΔV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; 1100 nm (Carrier Injection)</td>
<td>LIVA/XIVA, ... Diffusions</td>
<td>XIVA/OBIC, .... Diffusions</td>
</tr>
<tr>
<td>1300 nm (Thermal)</td>
<td>TIVA/OBIRCH/XIVA Resistive vias Metal shorts Resistive opens Electromigration</td>
<td>SEEBECK Contact opens</td>
</tr>
<tr>
<td></td>
<td>SDL/CPA Marginality isolation by local heating (soft defects) Resistive interconnects Resistive bridges</td>
<td></td>
</tr>
</tbody>
</table>
## Effective Resistive Defect Localization Techniques

<table>
<thead>
<tr>
<th>Resistive defect to Vdd &amp;/or GND</th>
<th>Resistive defect (bridges) between internal nodes</th>
<th>Resistive single node interconnect defect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classic static use case</strong></td>
<td><strong>Challenging to isolate</strong></td>
<td><strong>Most difficult to isolate</strong></td>
</tr>
<tr>
<td><img src="image" alt="Resistive defect diagram" /></td>
<td><img src="image" alt="Resistive defect diagram" /></td>
<td><img src="image" alt="Resistive defect diagram" /></td>
</tr>
<tr>
<td><strong>OBIRCH:</strong> High success %</td>
<td><strong>OBIRCH:</strong> Medium success %</td>
<td><strong>OBIRCH:</strong> Ineffective</td>
</tr>
<tr>
<td><strong>DLS:</strong> Medium success %</td>
<td><strong>DLS:</strong> High success %</td>
<td><strong>DLS:</strong> High success %</td>
</tr>
</tbody>
</table>
DLS Case Study #1 - SIL on 90 nm IC

- Design meets design rules (90nm process)
- Yield <10%
- Part sensitive to:
  - Voltage
  - Temperature (would tend to Pass if heated a few degrees)
  - Frequency
- Simulations unable to isolate
  - Suspect logic block was identified but could not be verified or localized to the transistors
- EmiScope and GlobalScan used to effectively localize the defect area
- System board was used as stimulus & Pass/Fail generation
DLS Case Study #1 - SIL on 90 nm IC Data

- Green indicates Pass (passes when heated)
- Using SIL & DLS, the data clearly identifies just a few “minimum dimension” transistors in the suspect area.
- “Green stringers” follow the poly structure
- Standard air gap lens cannot provide this detail
- Data shows a sharp thermal sensitivity inside the circuit area
- Data collected in minutes

GlobalScan Thermal DLS Mode (SIL Lens)

220x SIL lens
SIL DLS image showing resolution <200 nm

Thresholding used to highlight strongest signal region
GlobalScan Thermal DLS Mode - Soft Failure in IP Memory Block

- Yield problem, intermittent read failures from an IP block
- Failure at high T, high V, low freq. (leakage suspected)
- Schematics & SPICE unavailable from IP vendor
- Results obtained in a few hours of probing time

- DLS “Fail” signals (red circles) highlighting transistor N3 as defect site overlaid with CAD (green poly) & LSM image
DLS Case Study #3 – Analyzing a Yield Problem

- Yield problem, failure identified on a frame buffer using BIST
- No fault found using EmiScope
- DLS (1340) used to localized fault
- Failure localized to a PLL circuit
  - Green circles identify the section of the Failing PLL
  - Red boxes are CAD overlay
  - PLL was never suspected as a potential problem
  - Results obtained ~1 hour

- Illustrates complimentary value of GlobalScan (DLS) & EmiScope (timing) for efficient defect localization
DLS Case #4 – Flash Memory Resistive Interconnect Defect

- Flash device with failing Read operation.
- Device pass sometimes when operated at higher Vdd.
- Used DLS to localize what was likely to be a resistive via (green dot).
Common Causes of Soft Defects

- Metallization or interconnect defects
  - Via push up or voids
  - Electro migration
  - Stress voids
  - Metal “mouse bites”
  - Granularity/Grain boundaries
  - Resistive bridging

- Inter-level Dielectric (ILD)
  - Voids
  - Metal “slivers”
  - Contamination

- Oxide Defects
  - Soft gate oxide shorts
  - Hot carrier injection

- Process variations
  - Transistor effects:
    - Vt shifts **
    - Leff shift, Weff shift
    - NMOS to PMOS length ratio
    - Diffusion resistance **
  - Interconnect
    - Metal thickness variations
    - Spacing or pitch variations
  - Gate Oxide and ILD
    - Thickness variation

- Inaccurate library models

- Noise Related effects
  - GND/Vdd Noise, Vdd sag
  - Cross talk
  - CLK/PLL jitter

Design & process margins are being pushed to the limit & no longer have distinct “boundaries”
Bare Die Electrical Fixturing Solution

- Wire bond loose die onto die carrier
- Eliminates need for multiple mechanical probes
- Simpler solution for probing unpackaged die

- DUT Carrier
  - 15x15 PGA footprint (1.55 inch square), 24 contacts
  - 1-inch square clear zone for loose die or small fragment
  - Low cost consumable

- DUT Card - backside view
  - Form factor of a “universal” 512-pin DUT card
  - Low profile ZIF socket for 15 x 15 PGA
  - Low cost board – only one needed
Fundamentals of IC Test: Summary of Session 2

- IC Test drives physical failure analysis
  - Collected test data identify potential failure locations
  - Additional localization required to correct circuit problems
    - Circuit edit to validate design error hypothesis and continue debug
    - Defect site localization to reduce / eliminate wasted milling
    - Fixturing, signal management, test program manipulation are critical

- Variety of phenomena to display circuit behavior
  - Most are weak, requiring millions of test pattern cycles
  - Must keep circuit stabilized during data acquisition
  - ATE support for physical failure analysis needs synchronization

- Nanoscale processes continue the challenge
  - Unending room for innovation in PFA tools and techniques
Fundamentals of Semiconductor Test for Physical Failure Analysis

Dr. Burnell G. West, IEEE Life Fellow
Chief Architect, Credence Systems
Fundamentals of IC Test: Course Outline

- **Session 1 – The Structure of a Semiconductor Test**
  - What to test? Why? What tests work? How do tests work?
  - Digital, Mixed Signal, and Analog Tests
  - Functional, Structural, and Alternate Tests
  - ATE Architecture and test execution
  - Patterns, failure detection and response, logging
  - Test variables – voltage, frequency, DUT temperature

- **Session 2 – Applying Tests for Physical Failure Analysis**
  - What goes on when a device fails a test?
  - Defects vs Parametric Failures (vs Design Errors)
  - How do we isolate, locate, and exhibit a defect?
    - Analyzing and modifying tests to observe a failure mode
    - Editing physical devices to demonstrate cause
    - The role of ATE in physical failure analysis – case studies

- **References**
Why Test?

- Design is not perfectly reliable
  - Test can help detect and locate design errors ("bugs")
  - Test can help establish design margins

- Manufacturing is not perfectly reliable
  - Test can establish performance limits
  - Test can detect process excursions

- Things break - where was the weak link?
- Things wear out - what “couldn’t take the heat”?
- Nobody wants to ship bad product
- Nobody wants to build bad product
IC Test and Measurement Objectives

- Design debug
  - Circuit edit to support debug
- Design validation
- Device characterization

- Defect detection
  - Faulty circuit built-in repair
- Defect isolation
- Infant mortality acceleration
- Prompt process feedback

- Quality assurance through product lifetime
- Overall test cost control
  - What to test and when to test it
  - Minimize the cost of each test applied
  - Maximize the return from each test
- Test cost reduction cannot compromise test quality
What Kinds of Test?

- Launch-capture test?
- Structural test?
- BIST?
- Analog / mixed signal test?
- At-speed functional test?
- Static Functional Test?
- IDDQ test?
- DC parametric test?
- Low-V functional test?
- On-line test?
Test Coverage and Escapes

Sematech Study S121 (1997)
Scan - 99.7% fault coverage
FUNC - 52% fault coverage
IDDQ - 1478 “non-operational”
(of 1764 IDDQ failures; 2147 total)
Test in Physical Failure Analysis

- Observation is key
  - LVP processes
  - Photon emissions
  - Other electromagnetic or photo-optic effects

- Observation takes TIME
  - Effects are weak
  - Noise levels are large

- Tests must be LOOPED
  - Key to satisfactory defect phenomena observation
Test Operation Overview

Test Setup
- Test Controller
  - Install test
  - Start test

Test Instrument
- Transmit next test data
- Transmit required test clocking
- Receive and evaluate test results

At the I/O Boundary
- Receive and organize test sequence/patterns

Inside the Chip
- Test Setup
- Test Execute
- Execute test

Results
- Test Execute
- Evaluate test result
  - Log fail
  - Identify next test data set
- Log fail

Incorporate the PFA Loop
The Relevance of Digital Test

- Most of today’s devices (even SoC) are “mostly” digital
- Digital activity stimulates devices or evaluates results for almost all test types, including
  - Digital functional tests
  - Scan-based structural tests
  - Launch-Capture tests
  - BIST tests
  - IDDQ tests
  - Memory tests (and repairs)
  - Adaptive tests
The Structure of Digital Test Processes

- Principal use – sorting

- Ancillary use – defect isolation and PFA

- Tests are not necessarily designed with PFA in mind
  
  - So, how are tests designed?
  - Motivation, Implementation, and Adaptation to PFA
Digital Test – the classic model

Allows tests to be written simply as tables of “vectors”
with drive and strobe timing abstracted
Digital Test “Vectors”

<table>
<thead>
<tr>
<th>Test Pattern</th>
<th>in&lt;0:7&gt;</th>
<th>out&lt;0:7&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>01001011</td>
<td>xxxxxxxx</td>
</tr>
<tr>
<td>1</td>
<td>11001010</td>
<td>xxxx0110</td>
</tr>
<tr>
<td>2</td>
<td>00110101</td>
<td>11001101</td>
</tr>
<tr>
<td>3</td>
<td>10100101</td>
<td>01001011</td>
</tr>
<tr>
<td>4</td>
<td>11010001</td>
<td>0010zzzz</td>
</tr>
<tr>
<td>5</td>
<td>00101101</td>
<td>0011zzzz</td>
</tr>
<tr>
<td>6</td>
<td>11010010</td>
<td>xx1z0x1z</td>
</tr>
</tbody>
</table>

Response functional (or expect) data

Test vector number

Stimulus functional data

F-data
Digital Test “Vectors”

**test pattern**

```
in<0:7> out<0:7>
0  01001011 : xxxxxxxx
1  11010101 : xxxx0110
2  00110101 : 11001101
3  10100101 : 01001011
4  11010001 : 0010zzzz
5  00101101 : 0011zzzz
6  11010010 : xx1z0x1z
```

**fail datalog**

```
in<0:7> out<0:7>
3  10100101 : 01010011
4  11010001 : 0010zzzz
6  11010010 : xx1z0x00
```
Digital Test “Vectors” and STIL

- Most testing, today and for the foreseeable future, will be digital
  - Unambiguous specification of complex test requirements
  - Unambiguous evaluation of pass or fail result
  - *Highly computable* – growth for multiple generations

- Non-digital requirements can have digital manifestations –
  - ADC tests, BIST structures, etc

- Hence, STIL – *Standard Test Interface Language* – IEEE 1450
  - Standard description of digital patterns
  - Standard description of logic transitions – “events”
  - Standard description of basic digital test waveforms
Where Vectors Come From

- **Stored Pattern ("functional") Vectors**
  - Generated by hand, simulation, sometimes ATPG

- **SCAN Vectors**
  - Generated mainly by ATPG; require scan-chain DFT
  - Used for structural tests
  - Used for path delay tests
    - Requires launch-capture clocking

- **Algorithmic generation**
  - Used mainly for memory arrays

- **Built-in Self Test (BIST) generators**
How STIL *Event Streams* come from Vectors

- Variety of ways to represent digital tests
- All result in specific activity at each DUT pin
- Activity at each DUT pin is a *sequence of events*
  - Input pins: D1@time, D0@time
  - Output pins: T1@time, T0@time, X@time
  - I/O pins: D1@time, D0@time, DZ@time, T1@time, T0@time, TZ@time
- F-data translates to D1’s, D0’s, DZ’s or T1’s, T0’s, TZ’s, depending on I/O Definition
Event sequences to test a flip-flop

D event stream: DF@0
C event stream: DF@LE, DF@TE
Q,QN event streams: TF@SLE, X@STE (window strobe)
Q,QN event streams: TF@strobe_time (edge strobe)

-- but we must allow for inaccuracies

<table>
<thead>
<tr>
<th>test pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 – 01:01</td>
</tr>
<tr>
<td>1 – 10:10</td>
</tr>
<tr>
<td>2 – 01:01</td>
</tr>
</tbody>
</table>
Edge Placement Accuracy (EPA)

**Event Time** $T_e$

**inputs**
- $D$
- $C$

**output**
- $Q$
- $QN$

**window strobes**

**edge strobes**

---

**test pattern**
- 0 - 01:01
- 1 - 10:10
- 2 - 01:01
Determining Edge Placement

- Edge placement is a combination of three terms
  - Program value within each vector
  - Accumulated vector start offset
  - Calibration corrections
    - Static systematic errors
    - Pattern-dependent systematic errors
    - Clock-dependent systematic errors

- So, how is it implemented?
How a Tester Timing System Works

Event sequences from F-data and Vector Type
- each pin has its own event sequence per vector

Event times adjusted by period offset
- transform vector time to clock plus interpolation
- system clock *decoupled* from vector rate

Event times further corrected by cal offset
- corrections for path length variation (skew)
- corrections for dynamic errors
Vector Time Transformation

*period one*  
*period two*

\[
\begin{array}{c}
t1_1 \quad t1_2 \\
\hline
\hline
e1 \quad e2 \\
\hline\hline
t2_1 \quad t2_2 \\
\hline
\hline
\end{array}
\]

*clock*

\[
\begin{array}{c}
c1_1 \quad c1_2 \\
\hline
\hline
c2_1 \quad c2_2 \\
\hline
\hline
\end{array}
\]

\[
\begin{array}{c}
v1_1 \quad v2_1 \\
\hline
\hline
v1_2 \quad v2_2 \\
\hline
\hline
\end{array}
\]

*Transform 1*: Add accumulated period offset to program time  
*Transform 2*: Add calibration offset based on event type  
*Transform 3*: Determine clock cycle containing event time and compute interpolation value

*When clock cycle occurs, trigger interpolation delay circuit - - -*
Some Timing Error Sources

Dominant timing errors are
- Static systematic errors
- Tester clock cycle dependent systematic errors
- Pattern dependent systematic errors
- Noise

Systematic errors are (in principle) correctable

Noise is not correctable
(but can be averaged)

Can these errors be calibrated?
- Clock generation noise: no
- Clock distribution noise: no
- Interpolator non-linearity: yes
- Marker noise: no
- Formatter noise: no
- Timing signal path dynamics: yes
- PE timing non-linearity: yes
- PE-DUT signal path dynamics: some
- Source-clocked DUT tracking error: some
- Calibration error: NO !!
Variable Test Rates

- Pin event timing can change on the fly
  - Different vector types may have different timing

- Test rates can also change on the fly
  - For match synchronization
  - For speed testing
  - For characterization and debugging
Variable Test Rates – as used for debugging

Slower

Faster
Recent Major Technology Shifts

- Clock Multiplication
  - High-end uProc clocks multiply input rate
  - Internal PLL controls timing (not ATE)

- High-speed I/O’s use local clocking
  - Forwarded Clocks
  - Reference Clocks
  - Embedded Clocks
Variable Test Rates – with smart PLL clock

- Slower
- Faster

Launch | Capture
Fundamentals of IC Test: Summary of Session 1

- Manufacturing processes produce unshippable defects
  - Wide variety of tests can be applied
  - Studies show different tests expose different defects
    - Difficult to know a priori which tests will be most effective
    - Effective ATE must apply a variety of tests in multiple conditions

- Delay defects more predominant in advanced IC processes
  - Agile timing in ATE helped identify and locate delay defects
  - Higher speeds in modern devices challenge ATE agility
    - Delta timing moving on-chip

- Yield management demands quick process feedback
  - To be addressed in Session 2
Fundamentals of IC Test: Course Outline

- Session 1 – The Structure of a Semiconductor Test
  - Why test? What tests work? How are tests applied?
    - Digital, Mixed Signal, and Analog Tests
    - Functional, Structural, and Alternate Tests
  - ATE Architecture and test execution
    - Patterns, failure detection and response, logging
    - Test variables – voltage, frequency, DUT temperature

- Session 2 – Applying Tests for Physical Failure Analysis
  - What goes on when a device fails a test?
    - Design Errors vs Defects vs Parametric Failures
  - How do we isolate, locate, and exhibit a circuit problem?
    - Analyzing and modifying tests to observe a failure mode
    - Editing physical devices to demonstrate cause
    - The role of ATE in physical failure analysis

- References
Physical Failure Analysis: Test Loops Needed

- Defect Localization Requires Imaging
  - Frontside or backside imaging of pattern activity
  - Waveforms or circuit activity, heating, etc.

- Virtually all imaging processes require stable repeatable tests
  - Hundreds of thousands to many millions of repetitions to collect
  - Time-resolved emissions require extreme stability
    - Waveform capture with 30-ps resolution is very demanding

- How does ATE Generate test loops?
  - Functional patterns
  - Scan-based patterns
Test Operation Overview – Implementing a Loop

Test Controller
- Install test
- Start test

Test Instrument
- Receive and organize test sequence/patterns
- Apply next test vector
- Compare DUT response to table

Inside the DUT
- Execute test

Incorporate the PFA Loop

Log fail
IC Test and Measurement Objectives

- Design debug
  - Circuit edit to support debug
- Design validation
- Device characterization

- Defect detection
  - Faulty circuit built-in repair
- Defect isolation
- Infant mortality acceleration
- Prompt process feedback

- Quality assurance through product lifetime
- Overall test cost control
  - What to test and when to test it
  - Minimize the cost of each test applied
  - Maximize the return from each test
- Test cost reduction cannot compromise test quality

Early Silicon

Production Testing

Always
Why Integrated Circuit Edit?

- Vast variety of sophisticated EDA tools help designers debug their code
- Test structure are implemented into IC for post fabrication device inspection, performance & reliability assurance
- Design can be tested, simulated & debugged before fabrication
- FAB implements test structures for process control & verification
- ... 
- Every step starting from Concept to Fabrication is well monitored, thoroughly tested and verified multiple times.
- “Our Silicon is going to be perfect…”
- So why CIRCUIT EDIT ??????

Designs are getting more complex
Clocks are running faster
Interference, cross talk, L & C are hard to predict
Core voltages are dropping
Pre-silicon testing is getting extremely tedious
Simulations are taking excessively longer time-wise
Designs can not be verified 100%
Feature sizes are in the nano-meter range
Thinner gate-oxides are causing unexpected leakage
Routing layers are increasing in number
Dielectric & metal layers are getting thinner
Ultra-high doping precision is needed for actives
High accuracy is needed for Analog ICs
Packaging can stress sensitive devices
...

Therefore, Silicon may not be perfect!!!
Need of Circuit Edit

- Test finds the device doesn’t work
  - Testing may identify design failure source
- Simulations may verify proposed fix
- BUT low confidence in making an expensive mask change
- It may take weeks to months before an ECO’d IC would be available to validate design change.
- Is it going to work ???
- Therefore, FIB Circuit Edit is the solution for:
  - Validating design change on Tester
  - Proceeding to Mask change with confidence
  - Getting working devices to customers ASAP
- CE, vital for fast & cost effective Design-to-Market solution

Not all suggested and successful Circuit Edits work, if they did then Circuit Edit would be redundant.

Circuit edit ROI readily justified on basis of reducing mask costs

Mask cost savings related to whether design ECO moves forward

  If every design ECO correct, then wasteful to do CE?

Can justify doing CE anyway

CE enables further debug & diagnostics including ATE or system board testing

Maybe a 2\textsuperscript{nd} issue beyond the “known”? 

Design ECO should be executed by CE so retesting before mask change
CE Technology Roadmap

- Edit time increases as number of routing layers increase (from 3 to 10 in last 10 years) with technology nodes
- CE tools require constantly developing hardware, techniques & chemistries to address multi-layer deep sub-micron ICs
- CAD navigation is vital for effective CE solutions
Case Study

- Flipchip, 90nm Cu/low-k process
- Circuit issue was that an incorrect power supply Vdd1 was on three master phase locked loop controllers (PLLCs), i.e., the pull-up voltage on a logic gate was incorrect.
- Required connecting an M1 line to Vdd2 on M3 and cutting M1 line to isolate Vdd1 supply.
- The low beam current used for this operation made it successful.
- Once Test proved the edit worked, edit was repeated on the other 2 PLLCs & Tested
- In all eight devices were edited and Tested good.

Focused Ion Beam (FIB) instrument are used to physically edit ICs
  - Original CE concept proposed electron beams
  - FIB more controllable & efficient
FIB instrument are able to perform cuts & deposits, as “Fab-In-a-Box”
  - FIB traces work around fabrication as per design layout
  - FIB & fab have ~same challenges
CE addresses modifications mainly in Al, Cu / LowK and SiO₂
Edits can be performed on packaged device, bare die or wafer
Multiple edits can be performed on a single IC
ICs can be edited from Front-side or from Backside (FlipChip) Si
CE Planning for in Silicon Validation

- Once CE is part of debug then 2 doors open:
  - CE to fine tune analog circuitry
  - Case study: Design planned for edit at critical points
    - M1 resistors could be added to fine-tune design
    - Test found problem in area where expected
    - CE fix employed & design advanced
    - Product development time reduced
- May not understand where risks are, but EDA tools should help
- 2. Circuit edit enables validation of next silicon
  - When completed design team moves on to the sequel
  - Production silicon becomes prototype for new design
  - Edits validate assumptions going into new design before first silicon

New materials present new challenges to CE

- Recently Low-k and Cu has been a challenge for CE
- Cu grains etch non-uniformly & tend to redeposit in the vicinity
- Low-k is very fragile and can cause to expose circuitry very quickly
- Solution: FIB provides chemistry to even out Cu removal without re-deposition while protecting the die

FIB exposed Cu dummy fill metal with porous CDO (Carbon Doped Oxide) low-k

Note: Irregular porosity distribution

CAD navigation had been a powerful tool for Circuit Edit

- Dummy fill metal generally added by foundry due to process limitations
- Merged CAD with dummy metal often not available
- Dummy CAD beneficial to edit planning & executing
IC Test and Measurement Objectives

- Design debug
  - Circuit edit to support debug
- Design validation
- Device characterization

- Defect detection
  - Faulty circuit built-in repair
- Defect isolation
- Infant mortality acceleration
- Prompt process feedback

- Quality assurance through product lifetime
- Overall test cost control
  - What to test and when to test it
  - Minimize the cost of each test applied
  - Maximize the return from each test
- Test cost reduction cannot compromise test quality
New Technologies – New Problems

- New physical designs
  - 130 nm, 90 nm, 65 nm...
- New materials
  - SOI
  - Low K dielectric
  - Copper
  - Strained silicon

Increasing differences between simulation models and reality are making it ever more impractical to simulate fully new designs.

To support the high integration, high speed, low current required, and tight minimum features size, today’s state of the art technology involves shrinking gate dimensions, multiple level interconnect, and host of new materials. All these changes are extremely difficult to model and simulate. Compact models for sub-0.18 um and below are extremely complex and often do not accurately represent the behavior of high-speed circuits [1, 2]. Indeed, accurate SOI models are still emerging[3, 4]. With each new technology generation there is a race between the process maturation and the circuit design community. While designers are busy creating the new design, the process engineers are still adjusting the process parameters. As a result, the correlation between simulated results and physical implementation is weak, especially for early silicon.

There are also growing class of device abnormalities which do not manifest themselves as hard failures, but rather as “soft failures”. These failures manifest themselves only under certain conditions of temperature, frequency, or voltage[5]. Many failures analysis labs rely on physical evidence for their analysis, but many new failure conditions are non-visual[6].

This difference between simulation and modeling and the actual performance on the chip has been growing more and more pronounced from generation to generation, and has increased the need for analysis of finished devices to optimize performance. Given the importance of physical design verification, how efficient are your models?
Timing Debug

- Problem: Motorola (Freescale) device
- Scan chain was failing at certain speeds in new SOI design
  - Structural diagnostic software was not applicable
  - Scan pattern loop length: 60 microseconds
  - Power supply: 1.8V

- EmiScope Methodology
  - Measure waveforms at key nodes to identify root-cause of failure
C1_CLK turned off too late to block the falling transition at DIN
- EmiScope timing measurement identified a race condition
- Changing C1_CLK timing solved the problem

A. Input data sends falling edge through NMOS at (A)
B. Simultaneously, falling edge of PCH at (B) turns off PMOS
C. Correct data sent into Latch at (C) [C1CLK clock is on]
D. Later, Hatch data is flipped at (D)
E. Why? As C1CLK falls, DIN also falls
F. A rising edge on PCH at (F) propagates to DIN before the C1CLK falls.

Solution: delay PCH
Resistive Interconnect Localization

- Can use imaging tools to localize a wide variety of faults
  - Traditional “static” and “hard” faults common below 250 nm
  - Design and “soft” faults increasingly common below 90 nm
Resistive Gate Localization

- Localize and Characterize interconnect faults

Design Error – Transistor Mismatch

- Design error of mismatched transistor in analog circuit
- Used EmiScope to localize fault
- Simulated observed waveforms to identify root-cause
  - Good device to bad device comparison aided analysis
Resistive Via Localization

- Information related to failure(s) was used in fault diagnosis process
  - Synopsys TetraMAX and Cadence Encounter Test Diagnostics were used to localize fault

The schematic above highlights the flip-flops where the failure is occurring.

The failing flop information was fed into both Synopsys TetraMAX and Cadence Testbench (now called Encounter Test Design Edition) for fault modeling and diagnosis.
Failure was likely related to the 2 NOR Gates

- The 2 NOR gates were then probed. An abnormal emission waveform was captured from both gates. The signals were slow persistent emission waveforms, suggesting a very slow input signal to both of the NOR gates.
- From experience the problem looked to be resistive in nature.
Resistive Via Localization (cont’d)

- Probe results clearly identified failing net
- Based on results, one of two vias was point of failure

On investigation of the layout we found that the 2 NOR gates were fed by a common metal segment. One of two VIAs on this segment was exhibiting abnormal resistivity.
The example describes how to trace a typical signal through a logic path to pinpoint the root cause of a problem. The symptom of the failure is that the tester reports that PAD is high while the expected value is LOW.

The failure analysis starts at PAD and probes backwards through the logic. He compares the expected (simulated) waveforms to the data measured by the EmiScope. If no simulation information is available, he could probe the same signals on a good part and on a bad part. In this case simulation data is available to which he compares the EmiScope data. He is looking for a gate with the correct inputs but the wrong outputs, indicating that there is something wrong with that circuit.

The EmiScope produces current emission pulses. The nMOS produces an emission spike when the output passes from high to low, while the pMOS produces a photon emission spike when the output passes from low to high. The strength of emission of the pMOS is lower than the nMOS because holes (the majority carrier for p-doped silicon) have lower mobility than electrons (the majority carrier for n-doped silicon).
Discussion

The EmiScope probes signals from switching transistors. In this example, we are probing transistors (both nMOS and pMOS), which are driving the lines associated with the signal.

The probe locations and results are discussed below:

A. The reconstructed waveform doesn't match the simulated data. The error must be upstream from the probed signal.

B. The reconstructed waveform still doesn't match the simulated waveform. The error is further upstream.

C. The waveform matches the simulated signal. The error is not in the path before C, so we stop probing this path.

D. The waveform does not match simulation.

E. The waveform still doesn't match simulation.

F. The waveform does match simulation, suggesting that something is going wrong at the flip-flop.

G. The reconstructed waveform shows that CLK edge is too early, so the wrong data from F is being passed through the latch.
Typical Post-Silicon Problems

- Yield is lower than required on new technology
  - Inability to fully model circuit behavior
  - Inaccurate design models, especially for timing and jitter

- Resistive failures
  - Incomplete vias, electromigration, metal bridges, etc.

- Design errors
  - Over 50% of design time is spent in verification
  - Yet – design errors are still common

- Incomplete success with ATPG methodologies
  - Incomplete test coverage
  - Incomplete fault coverage
  - Scan chain failures

  They are expected to get worse below 90 nm
Today’s IC performance requirements and leading-edge processes make it very difficult to separate design errors from process margins.
Why didn't it work??

EFFECTIVE TOOLS ARE STILL NOT AVAILABLE!!

COULD D.V. HAVE CAUGHT THEM ALL??

Designs Having One or More Re-spins by Type of Flaw
- Logic or Functional (67%)
- Analog Circuit (35%)
- Noise (29%)
- Slow Path (28%)
- Clocking (25%)
- Yield (23%)
- Mixed-Signal Interface (21%)
- IR Drops (20%)
- Race Condition (17%)
- Power (17%)
- Firmware (13%)

Collett International Research, April 2002
The Debug Challenge

- Huge quantities of circuitry
- Massive volumes of data to manage/analyze
- Parametric deviations causing more device failures
- Data access limited by IO bandwidth
  - faster data access = shorter debug time
- Desired data flow for debug
  - (maybe) scan in to establish starting state of entire device
  - pattern execution with IO established by pattern
  - (maybe) scan dump together with standard IO for analysis
- Multi GBPS data rate => non-deterministic patterns
  - data collection must cope with varying data
  - cannot accept fixed-protocol solutions
    - reconfigure ATE for specific protocols
    - dump bit patterns for later software protocol analysis
Differing Debug Viewpoints

- **Tester's-eye View**
  - dataflow – both in and out
  - input waveforms
  - output waveforms

- **DUT's-eye View**
  - high-level structure
  - RTL
  - logical circuit detail
  - topology
  - physical implementation
DUT's-eye View: High-Level Structure (1)
```c
#include "systemc.h"

void fft::entry()
{
    float sample[16][2];
    unsigned int index;

    while(true)
    {
        data_req.write(false);
        while(index < 16)
        {
            data_req.write(true);
            wait_until(data_valid.delayed() == true);
            sample[index][0] = in_real.read();
            sample[index][1] = in_imag.read();
            index++;
            data_req.write(false);
            wait();
        }
        index = 0;

        //Calculate the W-values recursively
        w_real = cos(theta);
        w_imag = -sin(theta);
        while(index < len-1)
        {
            w_temp = w_rec_real*w_real - w_rec_imag*w_imag;
            w_rec_imag =  w_rec_real*w_imag + w_rec_imag*w_real;
            w_rec_real = w_temp;
            W[index][0] = w_rec_real;
            W[index][1] = w_rec_imag;
            index++;
        }

        //Begin Computation
        stage = 0;
        len = N; incr = 1;
        while (stage < N)
        {
            len = len/2;
            while(i < N)
            {
                sample[index][0] = tmp_real;
                sample[index][1] = tmp_imag;
                i = i + 2*len;
            }
            stage++;}
    }
}
DUT's-eye View: RTL

```c
#include "systemc.h"

//Begin Computation
while (stage < M)
len = N; incr = 1;
data_req.write(false);
sample[index][1] = in_imag.read();
wait_until(data_valid.delayed() == true);
```

```c
while(index < len-1)
{ w_real = cos(theta);
index++;
W[index][1] = w_rec_imag;
w_rec_real = w_temp;
w_temp = w_rec_real*w_real - w_rec_imag*w_imag;
}
```

```c
sample[index][0] = tmp_real;
```

```c
void fft::entry()
{ float sample[16][2];

while(true)
unsigned int index;

while(true)
{ data_req.write(false);
unsigned int index;

{ data_req.write(true);

{ index = 0;
while( index < 16 )
{ wait();
sample[index][0] = in_real.read();
wait_until(data_valid.delayed() == true);
len = len/2;
incr = 1;
data_req.write(false);
index++;
wait_until(data_valid.delayed() == true);
}
}

{ sample[index][0] = in_real.read();
wait_until(data_valid.delayed() == true);
len = len/2;
incr = 1;
data_req.write(false);
}

{ sample[index][0] = in_real.read();
wait_until(data_valid.delayed() == true);
len = len/2;
incr = 1;
data_req.write(true);
}

stage = 0;
//Calculate the W-values recursively

W[index][0] = w_rec_real;
w_rec_real = w_temp;
w_rec_imag = w_rec_real*w_imag + w_rec_imag*w_real;
w_temp = w_rec_real*w_real - w_rec_imag*w_imag;
W[index][1] = w_rec_imag;
}

stage++;
}
```

```c
w_imag = -sin(theta);
while(i < N)
{ w_imag = -sin(theta);
index++;
w_rec_real = w_temp;
w_rec_imag = w_rec_real*w_imag + w_rec_imag*w_real;
w_temp = w_rec_real*w_real - w_rec_imag*w_imag;
index++;
W[index][1] = w_rec_imag;
w_rec_real = w_temp;
w_rec_imag = w_rec_real*w_imag + w_rec_imag*w_real;
w_temp = w_rec_real*w_real - w_rec_imag*w_imag;
}
```

```c
sample[index][1] = tmp_imag;
```

```c
sample[index][0] = tmp_real;
```

```c
while(i < N)
{ i = i + 2*len;
}
```

```c
W[index][0] = w_rec_real;
w_rec_real = w_temp;
w_rec_imag = w_rec_real*w_imag + w_rec_imag*w_real;
w_temp = w_rec_real*w_real - w_rec_imag*w_imag;
W[index][1] = w_rec_imag;
```
#include "systemc.h"

void fft::entry()
{
    float sample[16][2];
    unsigned int index;
    while(true)
    {
        data_req.write(false);
        while( index < 16 )
        {
            data_req.write(true);
            wait_until(data_valid.delayed() == true);
            sample[index][0] = in_real.read();
            sample[index][1] = in_imag.read();
            index++;
            data_req.write(false);
            wait();
        }
        index = 0;
        //Calculate the W-values recursively
        w_real =  cos(theta);
        w_imag =  -sin(theta);
        while(index < len-1)
        {
            w_temp = w_rec_real*w_real - w_rec_imag*w_imag;
            w_rec_imag =  w_rec_real*w_imag + w_rec_imag*w_real;
            w_rec_real = w_temp;
            W[index][0] = w_rec_real;
            W[index][1] = w_rec_imag;
            index++;
        }
        //Begin Computation
        stage = 0;
        len = N; incr = 1;
        while (stage < M)
        {
            len = len/2;
            while(i < N)
            {
                sample[index][0] = tmp_real;
                sample[index][1] = tmp_imag;
                i = i + 2*len;
            }
            stage++;
        }
    }
}
DUT’s-eye View: The Circuit Debug Process
Challenges of Advanced Product FA / Debug

- Short channel – 90 nm and 65 nm design rules
  - Image and spatial resolution
  - Short channel modeling is difficult
- Dropping operating voltage levels
  - Emission signals are weaker
  - Lower productivity and effectiveness of solutions
- Materials and process challenges
  - Resistive shorts / bridges
  - Higher leakage problems
- Multiple metal interconnect
  - > 6 metal layers – “front side obstruction”
  - Capacitive effects and cross talk; difficult modeling

Need for better resolution & node level probing
Need for better sensitivity
Need for better sensitivity
Need for backside image resolution & node level probing

- Physical, node level debug and FA is essential
- Ever increasing performance / sensitivity is most important
Smaller process nodes (130 nm & below) are becoming more sensitive to higher R bridges.

- Increasing number of resistive defects causing more Vdd, temp, & freq sensitive fails (soft failures).
- Localizing resistive defects is very difficult using conventional techniques (mech probes).
Effective Resistive Defect Localization Techniques

GlobalScan Laser Stimulation Applications

<table>
<thead>
<tr>
<th>Laser $\lambda$ (Effect)</th>
<th>Static Laser Stimulation (SLS)</th>
<th>Dynamic Laser Simulation (DLS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bias: CV Measure: $\Delta I$ Bias: CI Measure: $\Delta V$</td>
<td>Bias: None Measure: $\Delta V$ or $\Delta I$ Monitor Pass/Fail</td>
</tr>
<tr>
<td>$&lt; 1100$ nm (Carrier Injection)</td>
<td>LIVA/XIVA, ... Diffusions</td>
<td>XIVA/OBIC, ... Diffusions</td>
</tr>
<tr>
<td>$1300$ nm (Thermal)</td>
<td>TIVA/OBIRCH/XIVA Resistive vias Metal shorts Resistive opens Electromigration</td>
<td>SEEBECK Contact opens</td>
</tr>
</tbody>
</table>
Effective Resistive Defect Localization Techniques

Resistive defect to Vdd &/or GND

**Classic static use case**

Resistive defect (bridges) between internal nodes

**Challenging to isolate**

Resistive single node interconnect defect

**Most difficult to isolate**

<table>
<thead>
<tr>
<th>Technique</th>
<th>Vdd &amp;/or GND</th>
<th>Internal Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBIRCH:</td>
<td>High success %</td>
<td>Medium success %</td>
</tr>
<tr>
<td>DLS:</td>
<td>Medium success %</td>
<td>High success %</td>
</tr>
</tbody>
</table>

OBIRCH: Ineffective

DLS: High success %
DLS Case Study #1 - SIL on 90 nm IC

- Design meets design rules (90nm process)
- Yield <10%
- Part sensitive to:
  - Voltage
  - Temperature (would tend to Pass if heated a few degrees)
  - Frequency
- Simulations unable to isolate
  - Suspect logic block was identified but could not be verified or localized to the transistors
- EmiScope and GlobalScan used to effectively localize the defect area
- System board was used as stimulus & Pass/Fail generation
DLS Case Study #1 - SIL on 90 nm IC Data

- Green indicates Pass (passes when heated)
- Using SIL & DLS, the data clearly identifies just a few “minimum dimension” transistors in the suspect area.
- “Green stringers” follow the poly structure
- Standard air gap lens can not provide this detail
- Data shows a sharp thermal sensitivity inside the circuit area
- Data collected in minutes

GlobalScan Thermal DLS Mode (SIL Lens)

220x SIL lens
SIL DLS image showing resolution <200 nm

220x SIL lens
Thresholding used to highlight strongest signal region

500nm
**GlobalScan Thermal DLS Mode - Soft Failure in IP Memory Block**

- Yield problem, intermittent read failures from an IP block
- Failure at high T, high V, low freq. (leakage suspected)
- Schematics & SPICE unavailable from IP vendor
- Results obtained in a few hours of probing time

DLS “Fail” signals (red circles) highlighting transistor N3 as defect site overlaid with CAD (green poly) & LSM image
DLS Case Study #3 – Analyzing a Yield Problem

- Yield problem, failure identified on a frame buffer using BIST
- No fault found using EmiScope
- DLS (1340) used to localized fault
- Failure localized to a PLL circuit
  - Green circles identify the section of the Failing PLL
  - Red boxes are CAD overlay
  - PLL was never suspected as a potential problem
  - Results obtained ~1 hour

- Illustrates complimentary value of GlobalScan (DLS) & EmiScope (timing) for efficient defect localization
DLS Case #4 – Flash Memory Resistive Interconnect Defect

- Flash device with failing Read operation.
- Device pass sometimes when operated at higher Vdd.
- Used DLS to localize what was likely to be a resistive via (green dot).
Common Causes of Soft Defects

- **Metallization or interconnect defects**
  - Via push up or voids
  - Electro migration
  - Stress voids
  - Metal “mouse bites”
  - Granularity/Grain boundaries
  - Resistive bridging

- **Inter-level Dielectric (ILD)**
  - Voids
  - Metal “slivers”
  - Contamination

- **Oxide Defects**
  - Soft gate oxide shorts
  - Hot carrier injection

- **Process variations**
  - Transistor effects:
    - $V_t$ shifts
    - $L_{eff}$ shift, $W_{eff}$ shift
    - NMOS to PMOS length ratio
    - Diffusion resistance
  - Interconnect
    - Metal thickness variations
    - Spacing or pitch variations
  - Gate Oxide and ILD
    - Thickness variation

- **Inaccurate library models**

- **Noise Related effects**
  - $GND/V_{dd}$ Noise, $V_{dd}$ sag
  - Cross talk
  - CLK/PLL jitter

Design & process margins are being pushed to the limit & no longer have distinct “boundaries”

Soft Defect Localization (SDL) on ICs - M Bruce, V Bruce, Eppes, Wilcox, E Cole, Tangyunyong, C Hawkins - ISTFA 2002
Bare Die Electrical Fixturing Solution

- Wire bond loose die onto die carrier
- Eliminates need for multiple mechanical probes
- Simpler solution for probing unpackaged die

**DUT Carrier**
- 15x15 PGA footprint (1.55 inch square), 24 contacts
- 1-inch square clear zone for loose die or small fragment
- Low cost consumable

**DUT Card - backside view**
- Form factor of a “universal” 512-pin DUT card
- Low profile ZIF socket for 15 x 15 PGA
- Low cost board – only one needed
Fundamentals of IC Test: Summary of Session 2

- IC Test drives physical failure analysis
  - Collected test data identify potential failure locations
  - Additional localization required to correct circuit problems
    - Circuit edit to validate design error hypothesis and continue debug
    - Defect site localization to reduce / eliminate wasted milling
    - Fixturing, signal management, test program manipulation are critical

- Variety of phenomena to display circuit behavior
  - Most are weak, requiring millions of test pattern cycles
  - Must keep circuit stabilized during data acquisition
  - ATE support for physical failure analysis needs synchronization

- Nanoscale processes continue the challenge
  - Unending room for innovation in PFA tools and techniques