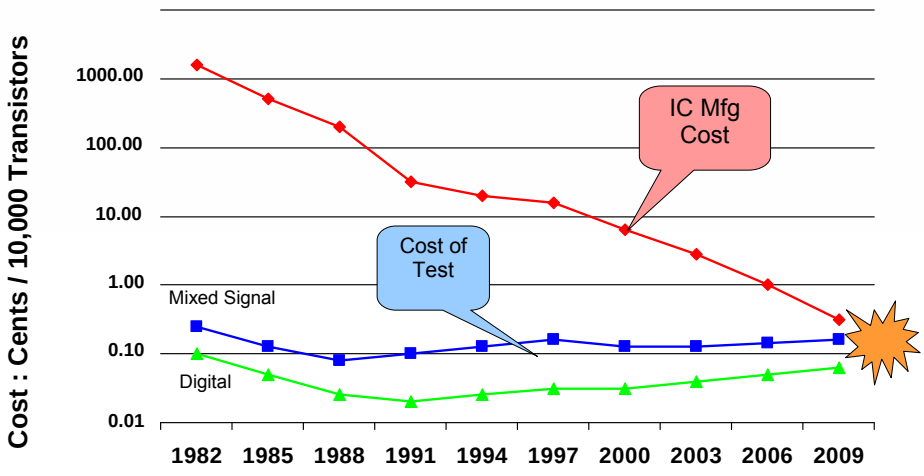
SoC Manufacturing Test

- SoC Testability Features
  - Boundary Scan
  - P1500 standard
- SoC Testing Costs
- Built-In Self Test
- Testing Mixed-Signal Components
  - "Alternate" test
- Defect Tolerance
- Error Detection and Fault Tolerance
- Loopback test of Mixed-Signal SoCs

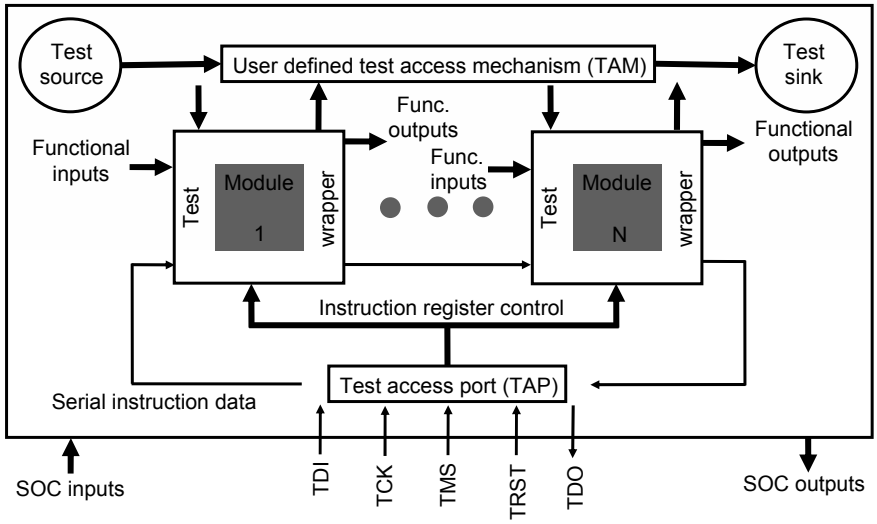
The Manufacturing Test Problem



Partitioning for SoC Test

- Partition according to test methodology:
  - Logic blocks
  - Memory blocks
  - Analog blocks
- Provide test access:
  - Boundary scan
  - Analog test bus
- Provide test-wrappers for cores

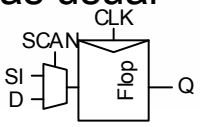
DFT Architecture for SOC

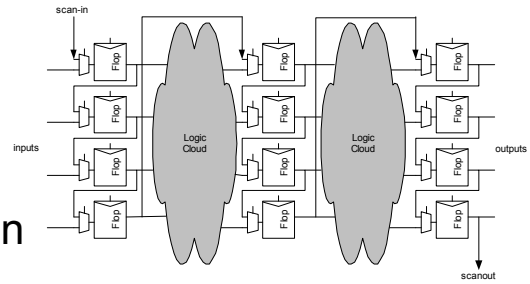


Source: Bushnell and Agrawal

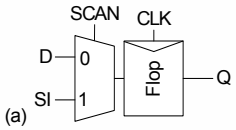
Scan

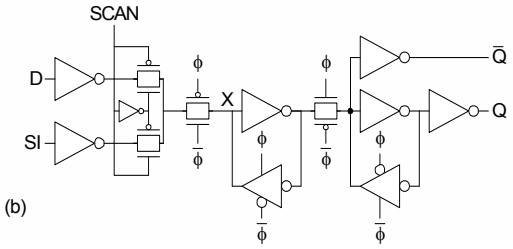
- Convert each flip-flop to a scan register
  - Only costs one extra multiplexer
- Normal mode: flip-flops behave as usual
- Scan mode: flip-flops behave as shift register
- Contents of flops can be scanned out and new values scanned in

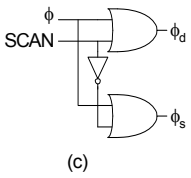


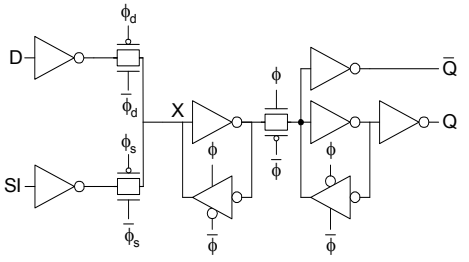


Scannable Flip-flops

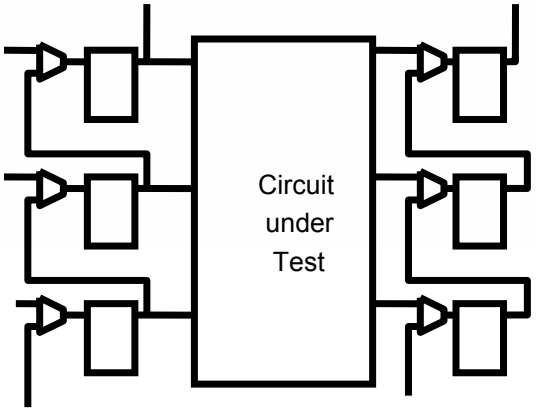








Scan Design and Delay Test



Need two patterns for delay test

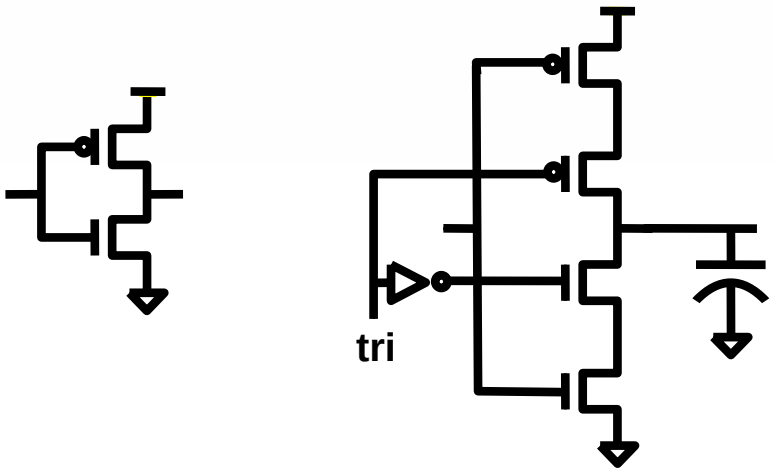
Shifting in second pattern changes state of the nodes

Solutions: Scan Shifting or Last Shift Launch

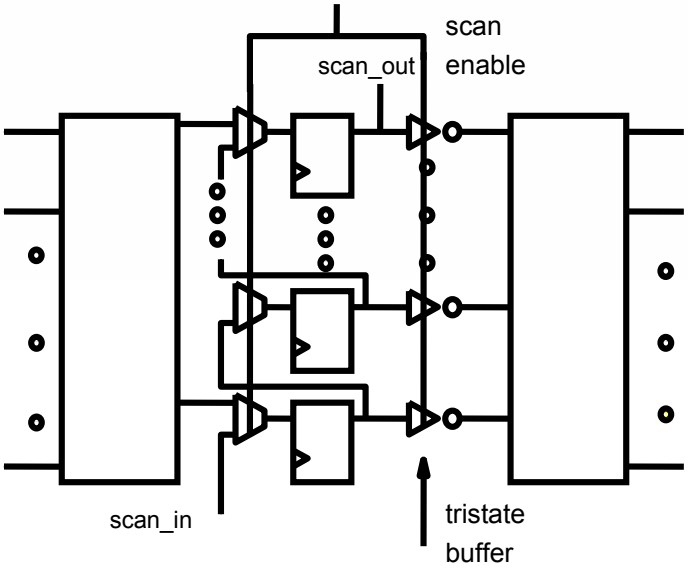
Functional Justification or Broadside Test

Tri-Scan Scheme

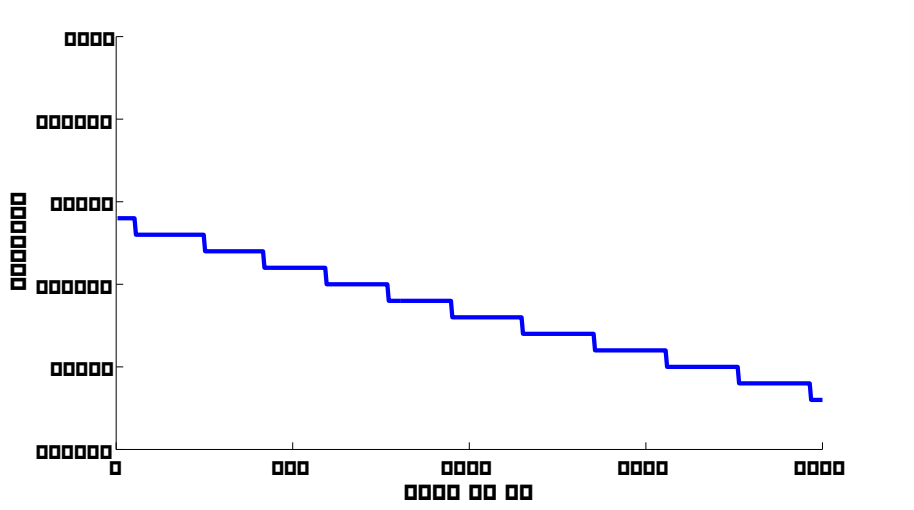
Based on state holding property of CMOS



Tri-Scan Scheme



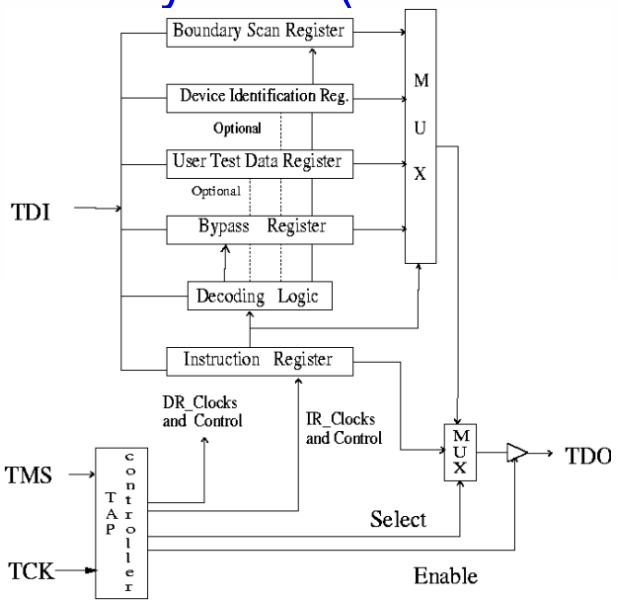
Voltage at Tri-stated output w.r.t. time



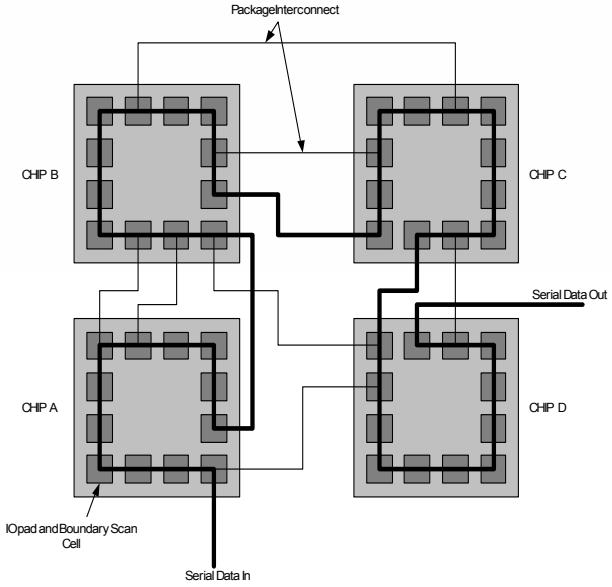
Boundary Scan

- 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 (IEEE 1149.1)



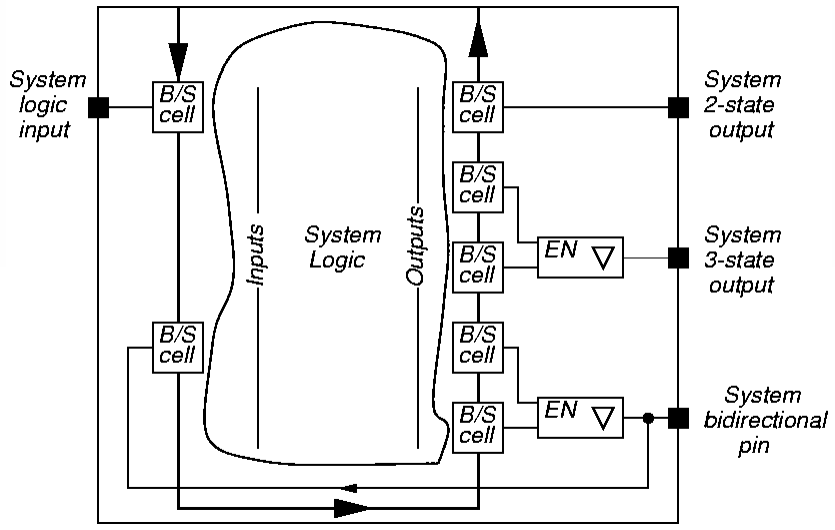
Boundary Scan Example



Boundary Scan Interface

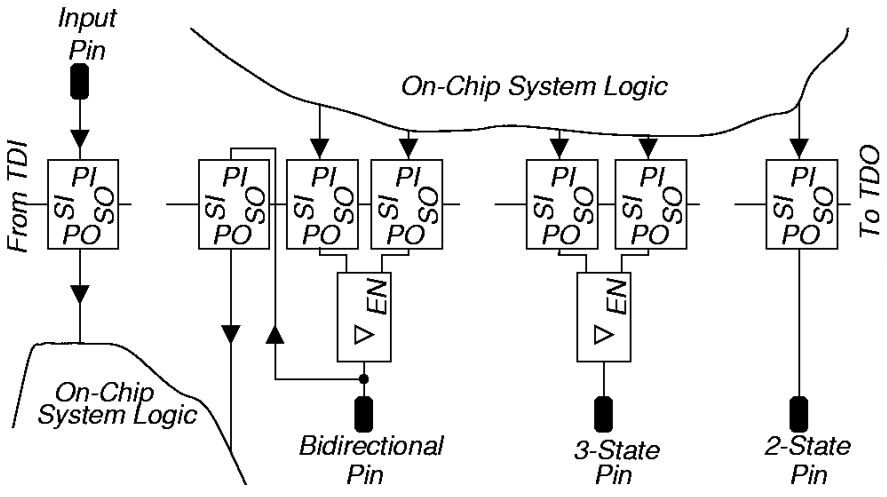
- Boundary scan is accessed through five pins
  - TCK: test clock
  - TMS: test mode select
  - TDI: test data in
  - TDO: test data out
  - TRST*: test reset (optional)
- Chips with internal scan chains can access the chains through boundary scan for unified test strategy.

System View of Interconnect



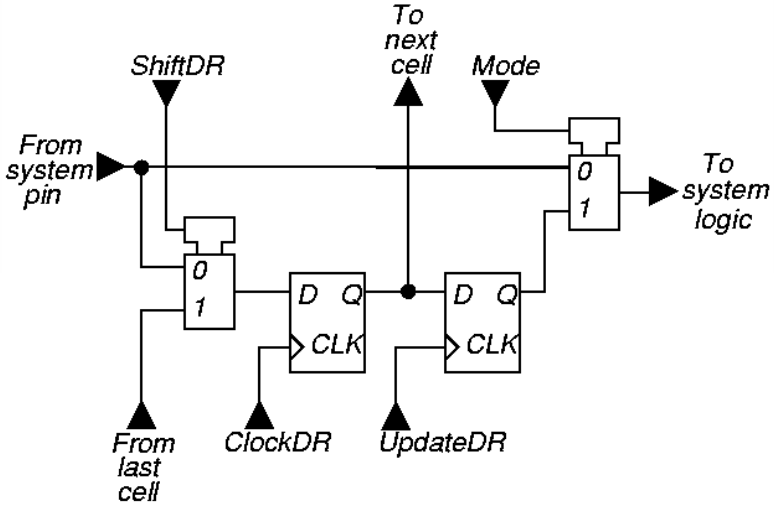
Source: Bushnell and Agrawal

Boundary Scan Chain View



Source: Bushnell and Agrawal

Elementary Boundary Scan Cell

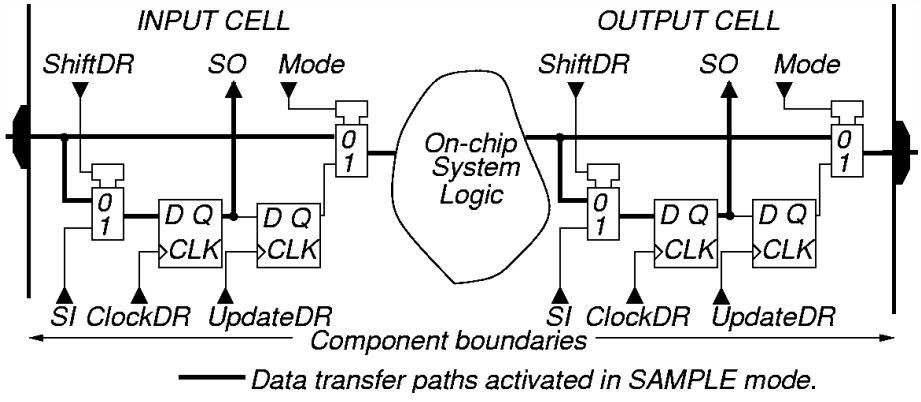


Source: Bushnell and Agrawal

SAMPLE / PRELOAD Instruction

Purpose:

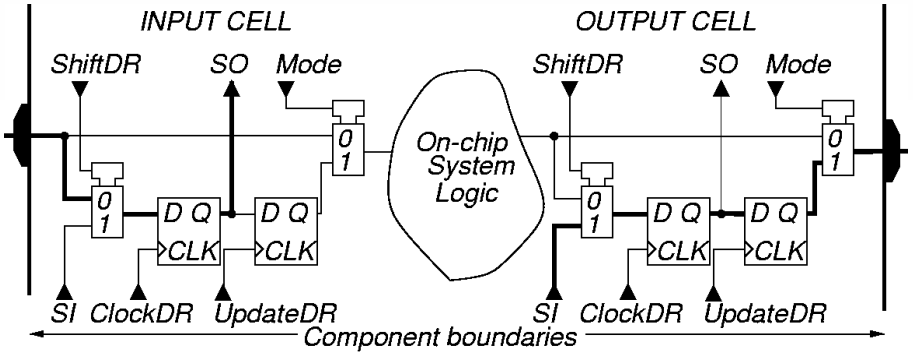
1. Get snapshot of normal chip output signals
2. Put data on bound. scan chain before next instr.



Source: Bushnell and Agrawal

EXTEST Instruction

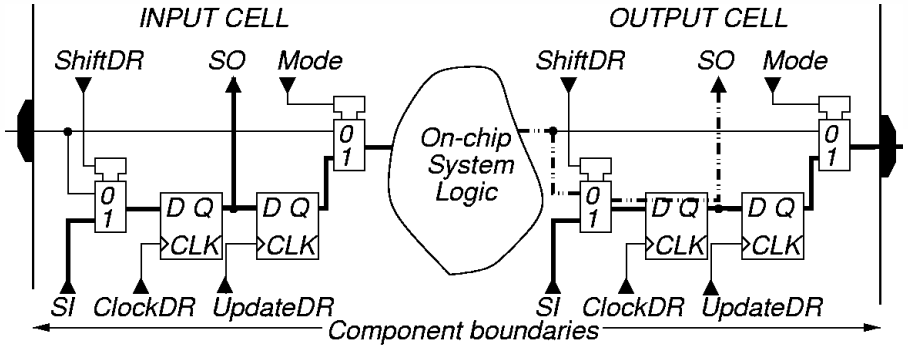
- **Purpose: Test off-chip circuits and board-level interconnections**



Source: Bushnell and Agrawal

INTEST Instruction

- **Purpose:**
  1. **Shifts external test patterns onto component**
  2. **External tester shifts component responses out**



Source: Bushnell and Agrawal

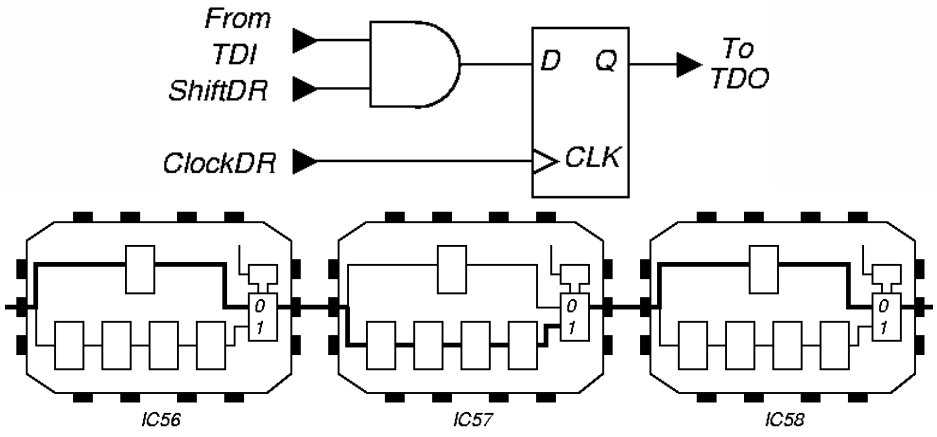
RUNBIST Instruction

- Purpose: Allows issue of BIST command to component through JTAG hardware
- Optional instruction
- Lets test logic control state of output pins
  1. Can be determined by pin boundary scan cell
  2. Can be forced into high impedance state
- BIST result (success or failure) can be left in boundary scan cell or internal cell
  - § Shift out through boundary scan chain
- May leave chip pins in an indeterminate state (reset required before normal operation resumes)

Source: Bushnell and Agrawal

BYPASS Instruction

- **Purpose: Bypasses scan chain with 1-bit register**



Source: Bushnell and Agrawal

Additional DFT Components

- Test source: Provides test vectors via on-chip LFSR, counter, ROM, or off-chip ATE.
- Test sink: Provides output verification using on-chip signature analyzer, or off-chip ATE.
- Test access mechanism (TAM): User-defined test data communication structure; carries test signals from source to module, and module to sink; tests module interconnects via test-wrappers; TAM may contain bus, boundary-scan and analog test bus components.
- Test controller: Boundary-scan test access port (TAP); receives control signals from outside; serially loads test instructions in test-wrappers.

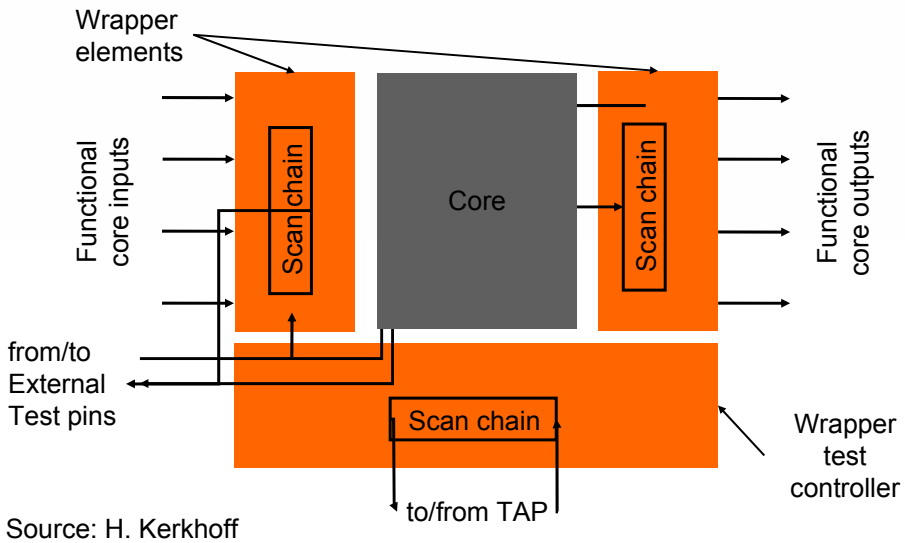
Source: H. Kerkhoff

Test Wrapper for a Core

- . Logic added around a core to provide test access to the embedded core
- Test-wrapper provides for each core input terminal
  - An external test mode – Wrapper element observes core input terminal for interconnect test
  - An internal test mode – Wrapper element controls state of core input terminal for testing the logic inside core
  - For each core output terminal
    - A normal mode – Host chip driven by core terminal
    - An external test mode – Host chip is driven by wrapper element for interconnect test
    - An internal test mode – Wrapper element observes core outputs for core test

Source: H. Kerkhoff

A Test-Wrapper



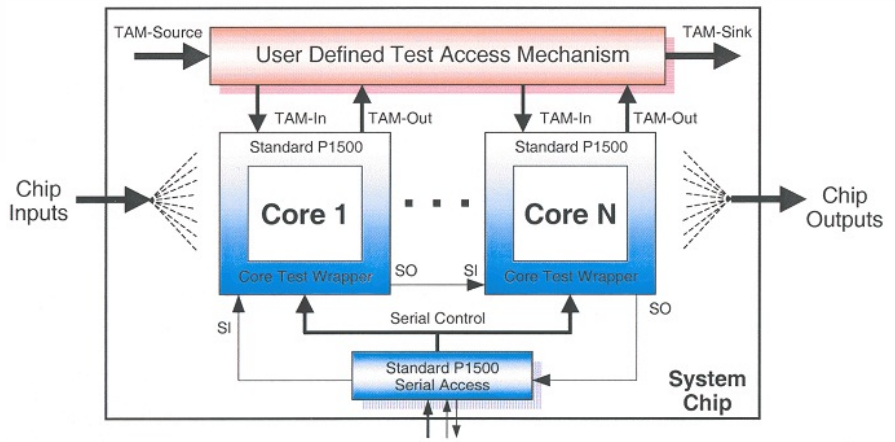
Source: H. Kerkhoff

Goals of IEEE P1500

- Core test interface between embedded core and system chip
- Test reuse for embedded cores
- Testability guarantee for system interconnect and logic
- Improve efficiency of test between core users and core providers

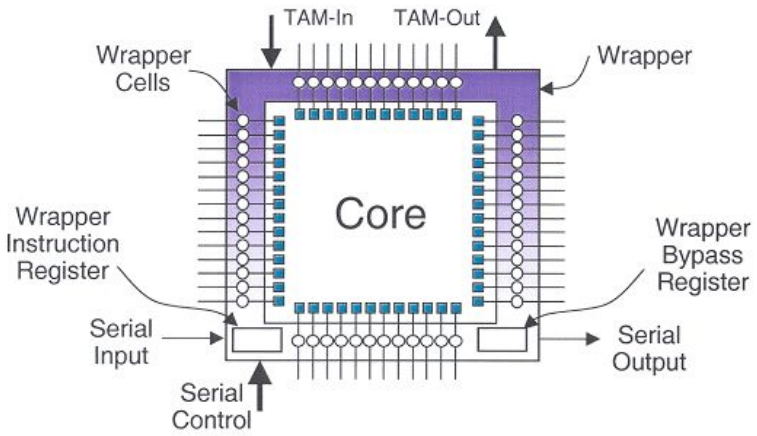
Source: H. Kerkhoff

Set-up of P1500 Architecture



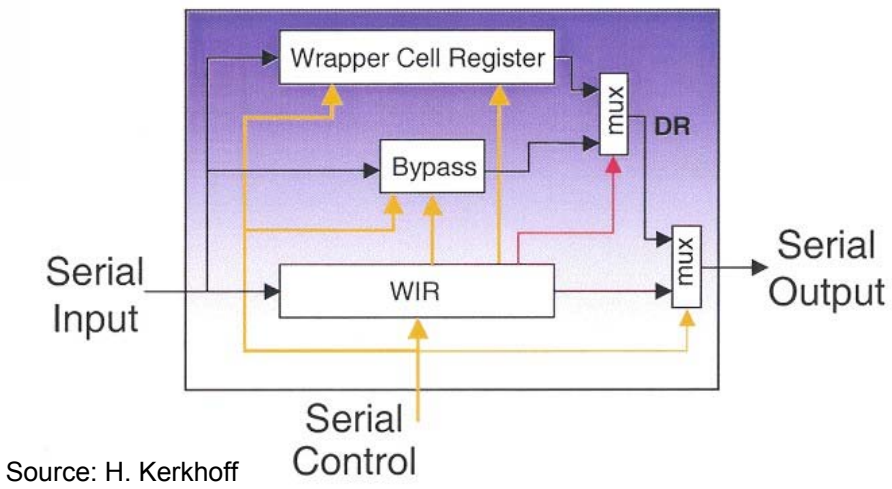
Source: H. Kerkhoff

Core including Wrapper Cells



Source: H. Kerkhoff

Wrapper Registers for P1500



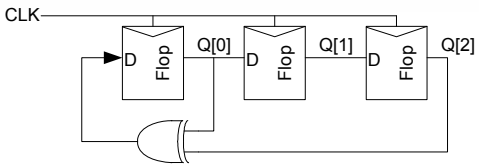
Source: H. Kerkhoff

Built-In Self Test (BIST)

- Increasing circuit complexity, tester cost
  - Interest in techniques which integrate some tester capabilities on the chip
  - Reduce tester costs
  - Test circuits at speed (more thoroughly)
- Approach:
  - Compress test responses into "**signature**"
  - **Pseudo-random (or pseudo-exhaustive) pattern generator (PRG)** on the chip
- **Integrating pattern generation and response evaluation on chip – BIST**

Pseudo-Random Sequences

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

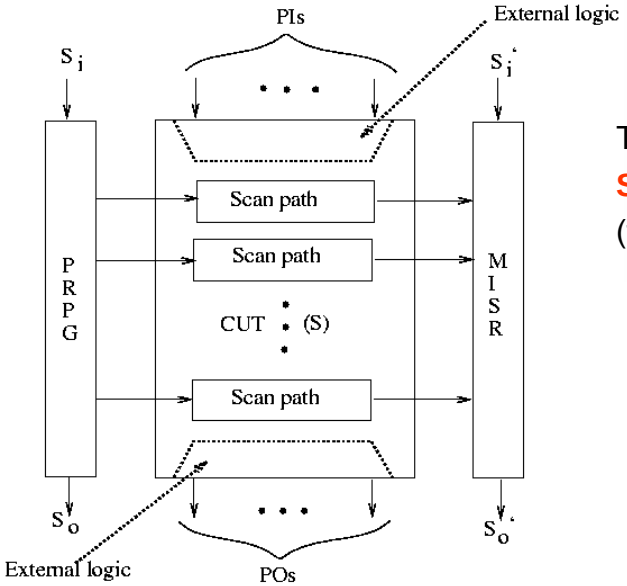


Step	Q
0	111
1	110
2	101
3	010
4	100
5	001
6	011
7	111

(repeats)

Can also be used to **compress** test responses

Example of BIST

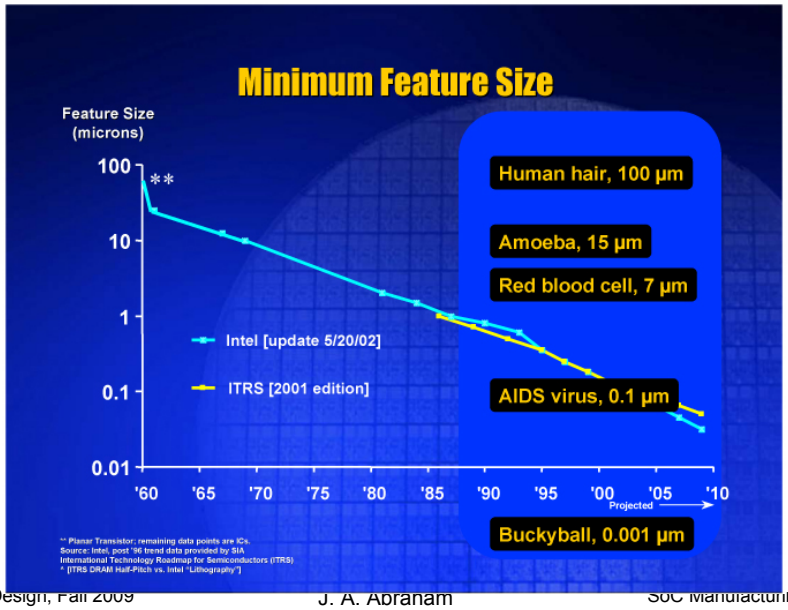


Technique called **STUMPS** (from IBM)

Why is Conventional Test Successful?

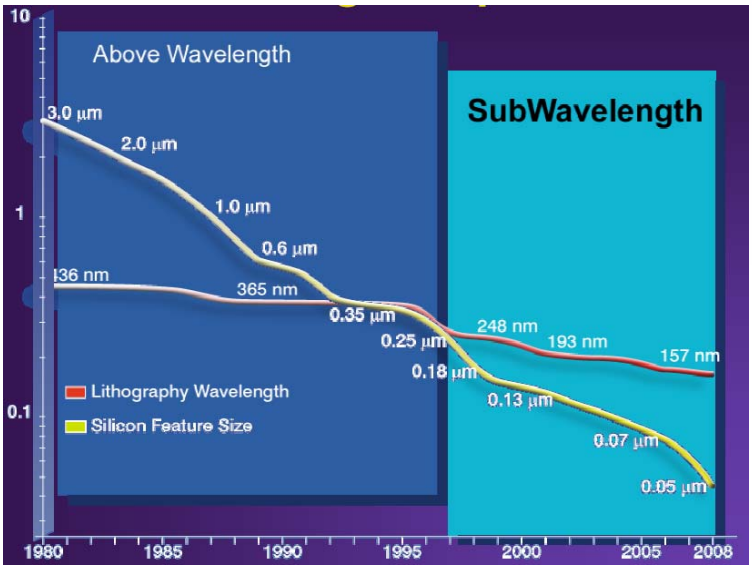
- Two innovations have allowed test to keep up with complex designs
- The stuck-at fault model
  - the model allows structural test generation, with a number of faults which is linear in the size of the circuit
- Partitioning the circuit
  - partitioning the circuit (with scan latches for example), alleviates the test problem so that test generation does not have to deal with the entire circuit
- Do these two assumptions hold for Deep SubMicron (DSM) circuits?

IC Technology

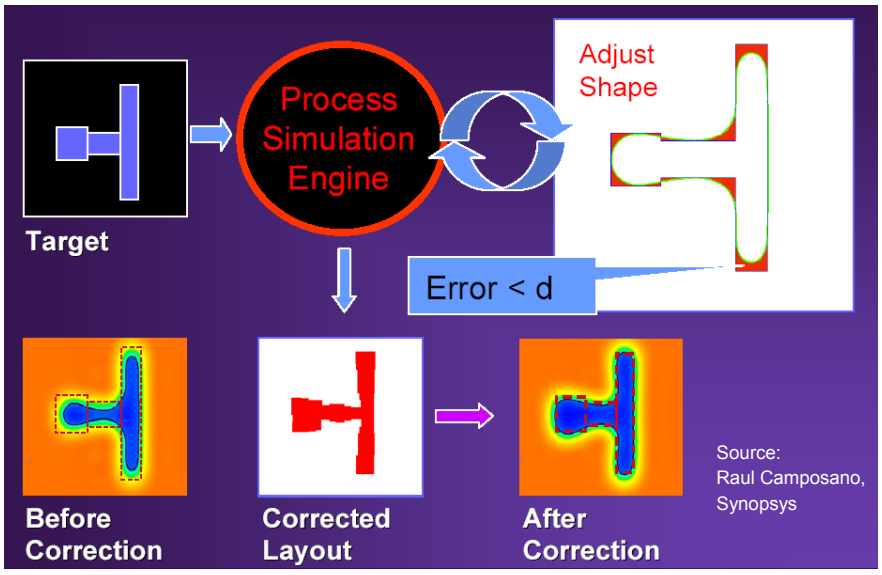


Features Smaller than Wavelengths

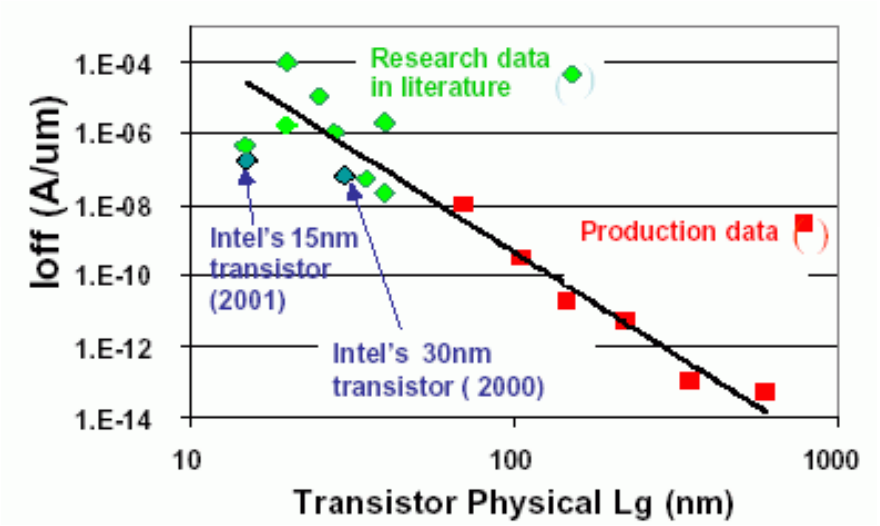
Source:
Raul Camposano,
Synopsys



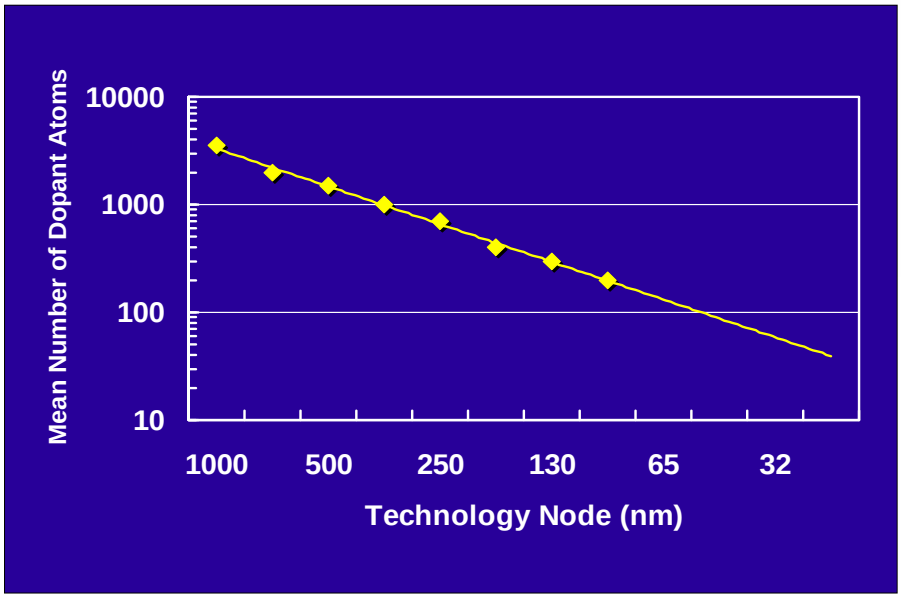
Optical Proximity Correction (OPC)



Increased Leakage



Random Dopant Fluctuations



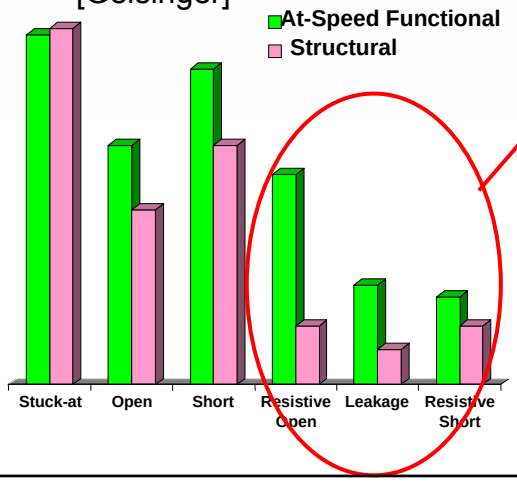
Defects in DSM Technologies

- Experiments on real chips (e.g., Stanford University)
  - Stuck-at tests do not detect some defects unless they are applied at speed
- Resistive opens comprise the bulk of test escapes in one production line
  - Likely in copper interconnect – cause delay faults
- Delay faults identified as the cause of most test escapes on another line
  - Speed differences of up to a factor of 1.5 can exist between fast and slow devices - problems with "speed binning"
  - Increasing possibility of shorts and crosstalk

Effects on Chip?

- Change in delays of paths
- Effects could be distributed across paths

[Gelsinger]



At-speed functional tests better for delay defects

Solution:

At-Speed tests

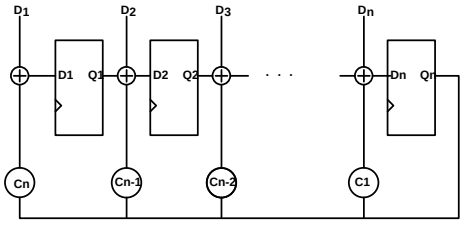
Tester Cost

Applied "Native Mode"?

Can use low-cost testers

Native-Mode Built-In Self Test

- Functional capabilities of processors can be used to replace BIST hardware – **(UT research, published in ITC 1998)**
- Application to self-test of processors at Intel – **FRITS method applied to Pentium 4, Itanium (Published in ITC 2002)**



Hardware for MISR

Software implementation of MISR

```
for each data value Di {
   Shift_Right_Through_Carry(S);
   if (Carry) { S = XOR(S, polynomial) }
   S = XOR(S, Di)        }
```

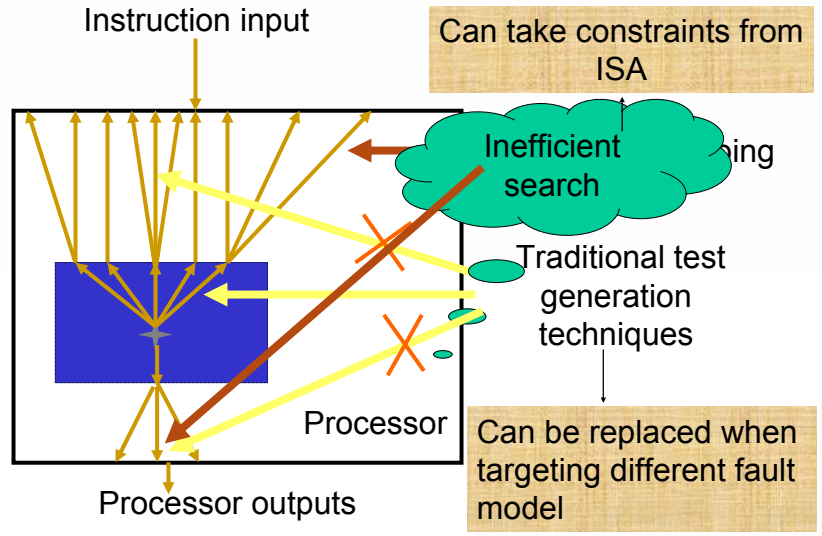

Native-Mode Self Test for Processors

- Random instructions can be run from cache and results compressed into a signature
- Implementation in Intel FRITS system showed benefits for real chips (Pentium 4, Itanium)
- Technique can be used for self-test of an embedded processor in a System-on-Chip
- Is it possible to now use this processing capability to test other modules (digital, analog/mixed-signal and RF) on the SoC?
  - First, can the processor test be improved to detect realistic defects, e.g., small delays?

Are Random Tests Sufficient?

- Intel implementation involved code in the cache which generated random instruction sequences
- Interest in generating instructions targeting faults
- Possible to generate instruction sequences which will test for an internal stuck-at fault in a module (Gurumurthy, Vasudevan and Abraham, ITC 2006)
- In order to deal with defects in DSM technologies, need to target small delay defects
- Recent work: automatically generate instruction sequences which will target small delay defects in an internal module (Gurumurthy, Vemu, Abraham and Saab, European Test Symposium (ETS) 2007)

Hierarchical approach to instruction mapping



Instruction mapping using bounded model checking

- Uses symbolic model verifier's (SMV) bounded model checking option (BMC)
  - Provides verification result up to a given bound
  - Accepts properties written in linear temporal logic (LTL)
  - Generates a counterexample if property fails
- Expresses the controllability and observability constraints in LTL
- Extracts instruction sequence from the counterexample

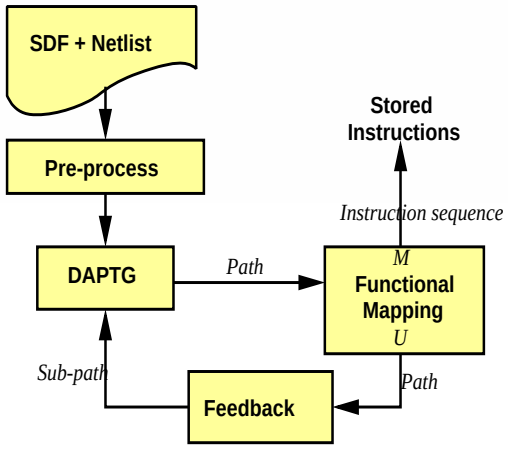
Application to stuck-at faults

- Used a commercial ATPG engine at the module level
- Mapped sequences generated by the ATPG engine
- No feedback
  - No additional effort if the sequence generated for a fault is not mappable
- Targeted hard-to-detect faults with this approach
- Able to achieve 82% fault coverage
  - Up from 68% through random instructions

Test for Small Delay Defects

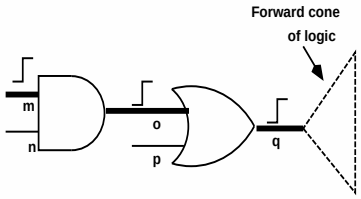
- Weighted random instructions will give good coverage for hard defects
- Need to test **paths** in the circuit to detect small delay defects
- However, the number of paths in a circuit can be exponential in the number of nodes
- Solution: **test the longest path through every node**
  - This will detect the smallest possible delay increase which will cause the circuit to fail
- Total number of tests linear in number of nodes

Automatic Generation of Instruction Sequences for Small Delay Defects



- Phase 1: all paths above delay threshold
- Phase 2: longest paths through all nodes
- Delay-Based ATPG: generate “TRUE” paths above given delay threshold
- Functional mapping: use verification engine
- Feedback: heuristics to speed up search

Functional Mapping - Illustrated




```
L1   #define legal (insn_input == `add) || (insn_input == `sub)

L2   if (`legal&&(m==0)&&(o==0)&&(p==0)&&(q==0)) begin
L3      prop = 1;
L4      wait(1);
L5      if (`legal&&(m==1)&&(o==1)&&(p==0)&&(q==1)begin
L6         prop = 0;
L7         wait(1);
L8         `All_outputs_equal;
L9      end
L10  end
```

Opcode Constraints

insn_input == `add OR insn_input == `sub

Path excitation constraints

m → rising

o → rising

q → rising

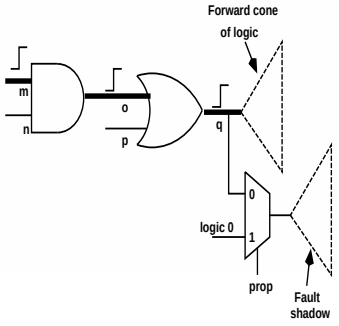
Robustness constraints

p is stable zero

Lines L2 and L5 hold these constraints (Antecedent of the property)

Functional mapping - Illustrated

```
L1   #define legal (insn_input == `add) || (insn_input == `sub)
L2   if (`legal&&(m==0)&&(o==0)&&(p==0)&&(q==0)) begin
L3      prop = 1;
L4      wait(1);
L5      if (`legal&&(m==1)&&(o==1)&&(p==0)&&(q==1)begin
L6         prop = 0;
L7         wait(1);
L8         `All_outputs_equal;
L9      end
L10  end
```



Observability constraints based on Boolean difference

- Model transformed to generate fault shadow logic
- Faulty value propagated through fault shadow logic
- 'prop' signal introduced to make sure faulty value is propagated only when path is excited
- Assertion expresses that outputs in original and fault shadow logic are always equal

Functional mapping - Illustrated

- Transformed model and property given to BMC
- Counterexample, if produced, satisfies the excitation, controllability and robustness constraints
- Fails the assertion → some output is different between faulty and correct logic
- Values for 'insn_input' in counterexample trace gives the required instruction sequence

Feedback

- Many paths generated by DATPG are not functionally feasible
- Many non-functional paths have a common non-functional sub-path
- Process of identifying the maximal non-functional sub-path in a given path is time consuming - $O(n^2)$ iterations needed
- Fact - first few nodes in consecutive paths produced by DATPG are generally the same

```
Find_subpath(path,N) {
//Get sub-path starting at input half
//the size of current sub-path
  subpath = getsubpath(N/2);
  if (functionally_controllable(subpath) {
    return N;
  } else {
    Find_subpath(N/2);
  }
}
```

→ Gives a non-functional sub-path in *O(log n) iterations*

Experimental setup

- OR1200
  - Open source RISC processor
  - 5 stage pipeline
  - Source code and documentation available from www.opencores.org
  - Synthesized using TSMC's 0.18u Artisan technology

No. of instructions in OR1200 ISA	92
No. of combinational	15878
No. of sequential elements	1594

Results

Phase1: Threshold 80% of clock

Paths	Yes	No	Timed out
27424	15118	12106	200

Phase2: Results for some modules

Module	Yes	No	Rejected Sub-paths	N(%)
ctrl	1826	29191	68087	91
alu	1427	16985	2716	100
lsu	970	4077	3744	100
wbmux	1146	2285	2118	100

N → Node coverage efficiency

Percentage of nodes for which mapping produced a test or rejected all paths given by DATPG

Yes – Functionally feasible

– Not functionally feasible

N	96%
Average mapping time	18.85secs

Test Access Mechanisms for SoC Test

- Non-functional access
  - Uses a kind of access to core not allowed during the normal functional operation
  - Generally based on scan chains or other design for test (DFT) structures
  - Can also use the embedded processor as the test source/sink → Needs wrappers around the core under test
- Functional access
  - Embedded processor is the test source/sink → *No DFT structures or wrappers around the cores*

55

Non-functional TAMs

- Boundary scan based
  - Uses the JTAG/boundary scan mechanism to load/capture the tests
  - Slow since the access is serial
- Direct access based
  - Direct access to core test pins given through external pins
  - Faster
  - High overhead to route the access pins and also multiple pins required

56

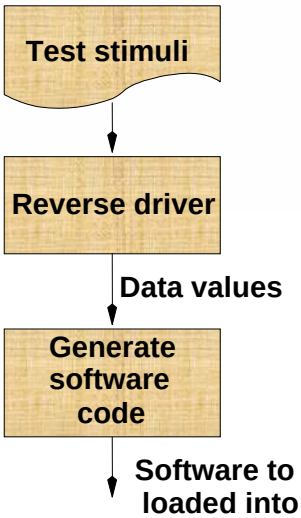
Functional TAMs for Testing Cores

- Software-Based Self Test: Use the intelligence of the embedded processor to test the SOC
- At-speed tests are possible
- Cores in the SOC can be of three kinds
  1. White box -- internals visible, structure changeable
  2. Grey box – all the internals visible, but structure of the core cannot be changed
  3. Black box – no internals visible, no change can be made on the core
- Any methodology for testing black box cores should not depend on knowledge of the core's internals

57

Approach to Testing Cores

- Uses functional TAM
- Uses pre-existing vectors
- Generates software to be loaded on to the embedded processor



- Gurumurthy, Sambamurthy and Abraham, Int'l Test Synthesis Workshop (ITSW) 2008

58

Pre-Existing Vectors

- If using a core bought from vendor
  - Vectors might also be provided by the vendor
- Reusing a core
  - Vectors from the previous use
- Newly designed core
  - Validation vectors
- Only constraint: these vectors must be functional test patterns for the core

59

Reverse Driver

- Parses the vector sequence to generate the data set to be sent to the core being tested
- Is specific to each core – as many as the number of driver programs
- Only overhead involved
- Generates the output in a format readable by the driver program

60

Reverse Driver – Illustration

- Peripheral core communicating with external environment (send/receive 32-bit data)
- Five 8-bit registers addresses 0 – 4
  - Register 0 – Control
  - Registers 1 to 4 – Data

Address	Data
0x00	0x07
0x01	0x54
0x02	0xDF
0x03	0x71
0x04	0x78

Reverse Driver →

Send at speed rate 1
Data 0x0754DF7178

61

Software Generation

- Use the driver program associated with each core being tested
- Driver programs
  - Software code that actually talks with the non-processor cores
  - Know about the bus protocol
  - Generally able to take in the data to be sent to the core or read back data from the core
  - Developed as part of designing the SOC

62

Coverage Measurement

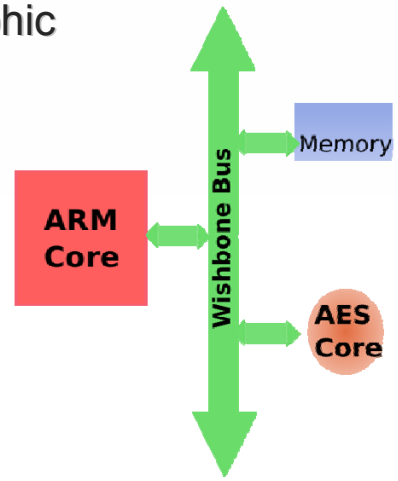
- Simulate the SOC using the software generated
  - Platform used SOC validation can be used
- Monitor the core boundaries to capture the pin data
- Fault simulate the core with the captured data

63

Experimental Setup

- Implemented a SOC containing ARM core, AES cryptographic core and a Wishbone bus interface (Verilog)
- AES 128-bit data/key encryption/decryption from *www.opencores.org*

Validation vectors:
Set of random values encrypted and decrypted



64

Experiment Results

Details about the synthesized AES core

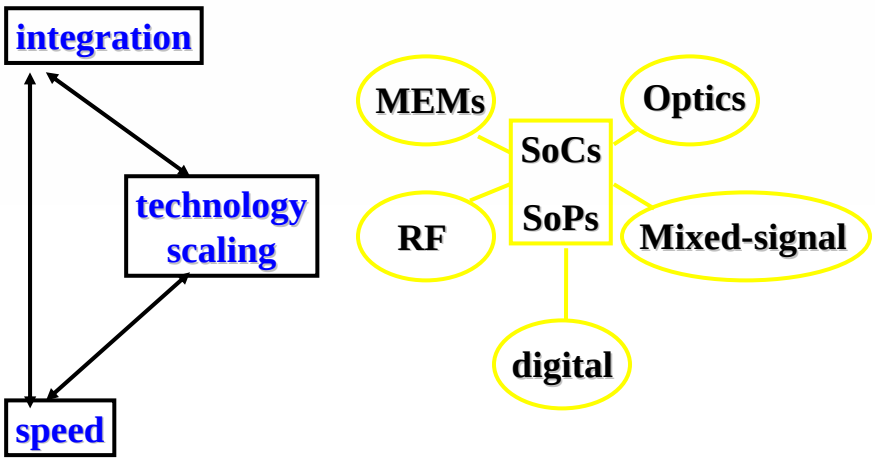
No. of inputs	69
No. of outputs	33
No. of sequential elements	9225
No. of combinational elements	1119
No. of stuck-at faults	64070

Results

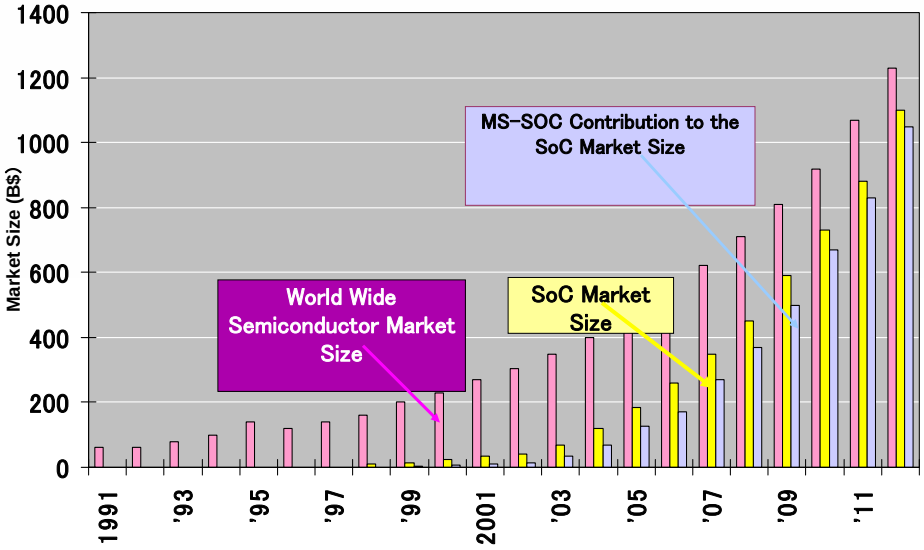
	Size (bytes)	Fault coverage	Original coverage	No. of cycles	Original cycles
Test1	7808	90.01	90.26	6700	6373
Test2	9128	90.15	90.35	7816	7435
Test3	10432	90.20	90.44	8932	8496

65

Technology Roadmap



System-on-Chip Market Size



New test problem: dealing with embedded mixed-signal blocks

Testing Mixed-Signal SoCs

- Analog test issues
- Analog test bus
- "Alternate" tests
- System-level Built-In Self Test
- Testers and test application

Testing Analog/Mixed-Signal/RF Circuits

- Have to deal with continuous signals
- ***Customers want a guarantee of specifications***
- ***A defect may or may not affect the desired behavior of a chip***
- ***Tests are for the specifications, not for defects***
- ***Similar trend in digital: testing for distributed path delays***
- ***Test costs are very high if every specification has to be tested***

"Alternate Tests"

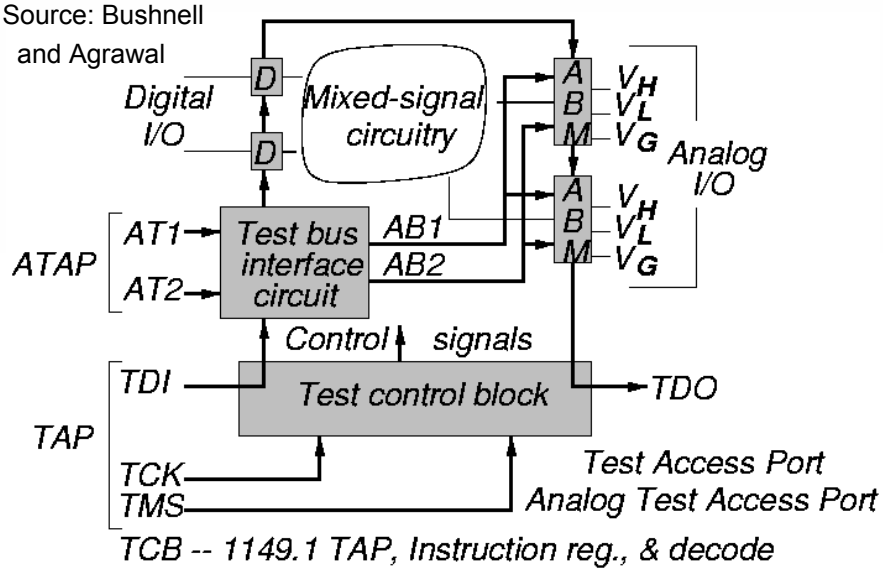
Analog Test Bus (IEEE 1149.4)

- PROs:
  - Usable with digital JTAG boundary scan
  - Adds analog testability – both controllability and observability
  - Eliminates large area needed for analog test points
- CONs:
  - May have a 5% measurement error
  - C-switch sampling devices couple all probe points capacitively, even with test bus off – requires more elaborate (larger) switches
  - Stringent limit on how far data can move through the bus before it must be digitized to retain accuracy

Source: Bushnell and Agrawal

Analog Test Bus Diagram

Source: Bushnell and Agrawal

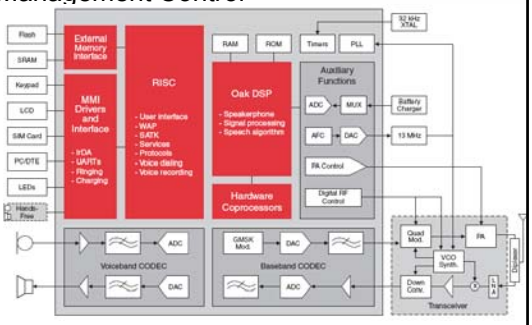


Typical Mixed-Signal Test Program

- Open-Shorts
  - Detected wirebond and packaging issues
- Leakage - Test Input and Tri-State pads
- DC Levels - Vol, Voh, Vih, Vil of pads
- Digital Tests: SCAN Tests, Memory, Functional Test (@ speed, high speed IO)
- Current tests – Dynamic, Special Modes, Standby, Iddq
- Mixed-Signal Test
  - PLL, DAC, ADC, OpAmps, Filters, References, Mixers
  - Gain Stages, Impedance Match, PA, LNA . . .

Example Device

- Cell-Phone Handset Baseband IC
  - Digital
    - Flash, SRAM, Keypad, LCD, SIM Card, PC & LED interfaces
    - RISC Controller, DSP, Hardware Co-processors
    - Embedded SRAM
    - Embedded ROM
    - Transceiver and Power Management Control
  - Control DAC/ADC
    - PA, Transceiver
  - Voice CODEC
  - Baseband DAC, ADC
  - PLL
  - Timer
  - Voltage References



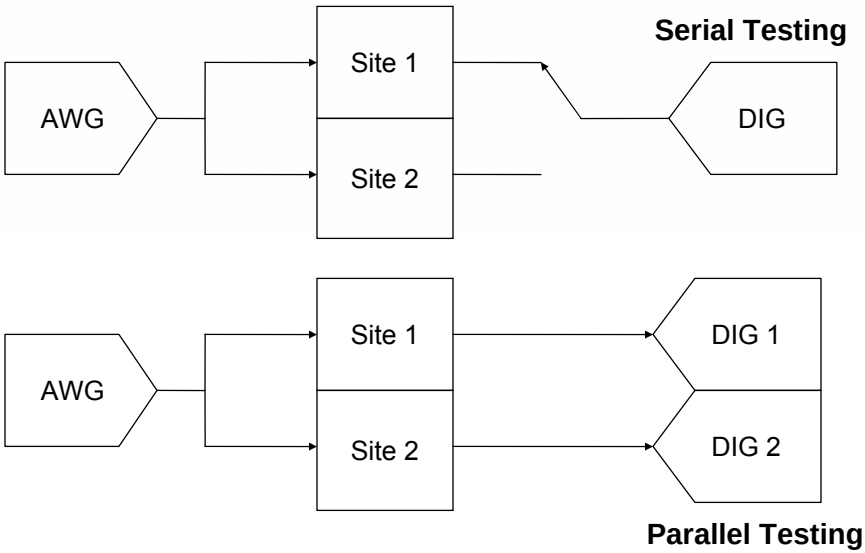
Single Site Test Program

- Tester: Teradyne Catalyst
- Capital Cost: $2.3 million
- Test Times: 10 Seconds
  - Test time profile:
    - **DAC/ADCs: 35%**
    - **Digital & Memory 30%**
    - **Idd 17%**
    - **Leakage & O/S: 8%**
    - **Reference tests: 5%**
    - **PLL: 3%**
    - **Test overhead: 2%**

Multi-Site Test

- Multi-Site Test – testing more than 1 device at a time
- Parallel Tests – testing of multiple devices simultaneously
  - Assumes no resource limitations
- Serial Tests – testing executed one site at a time because of resource limitations

Multi-Site Testing



Multisite Mixed-Signal Testing

- Multi-site Functional, Scan, Memory, leakage & continuity test efficiencies are typically high: 85-98%
  - Resource per-pin / site
    - Per Pin PMU - leakage & continuity
    - Functional pattern memory behind each pin
    - SCAN Capability, Memory Test Option (per site)
- Mixed-Signal efficiencies are typically driven by resource constraints
  - dedicated vs. shared instruments
  - which drive parallel vs. serial execution

Multi-Site Tester

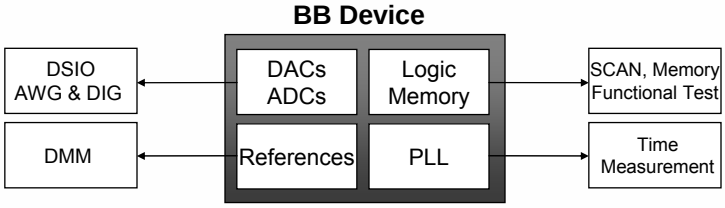
- A single site tester
  - High Resolution AWG & Digitizer
  - Time Jitter Digitizer
  - High Bandwidth AWG & Digitizer
  - DMM
  - Digital Pins with PMU Memory

Test Coverage

Voice Codec
GP DACs & ADCs
PLL
BB codec
References
Digital &
Leakage & OS

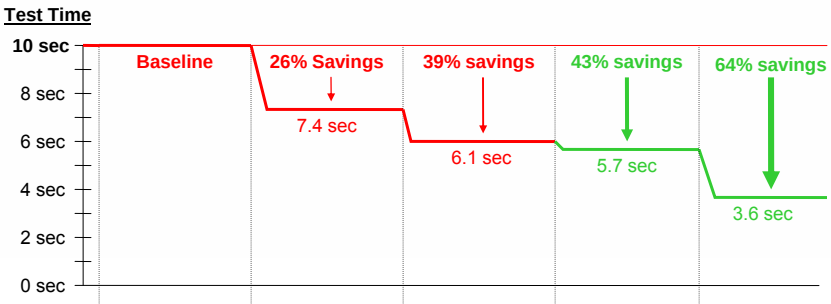
- Instrument requirements for quad site Parallel Testing
  - 4x if each resource is not shared
  - Costly purpose build tester, with instruments shared

Multi-Site Test Program



Test Time Profile	%TT	case 1	case 2
– DAC/ADCs:	35%	shared	dedicated
– Digital & memory	30%	dedicated	dedicated
– Idd:	17%	dedicated	dedicated
– Leakage & O/S:	8%	dedicated	dedicated
– Reference tests:	5%	shared	shared
– PLL:	3%	shared	shared
– Test overhead:	2%	shared	shared

Multi-Site Productivity Improvement



Sites:	Single	Dual	Quad	Dual	Quad
Resoursing:	Shared	Shared	Shared	Dedicated	Dedicated
AWG:	1	1	1	1	1
DIG:	1	1	1	2	4

Parallel Test

Cell Phone SOC (different device)

	Multi-Site Efficiency	Tester Cost	Ratio
Single site	1.00	1.00	1
Dual site non-shared	1.73	1.32	1.3
Qual site non-shared	3.00	2.16	1.4

Multi-Site efficiency goes up faster than tester cost.

Hard Drive Read Channel Device

Typical device: DAC & ADC, AGC, Filters, Thermal Sensor, OpAmps, ROM, PLL, Digital Signal Processing, Small Memory

# Sites	Catalyst	Efficiency	Tiger	Efficiency
Single	3.19 sec	--	2.59 sec	--
Dual	3.86 sec	79.1%	2.99 sec	84.7%
Triple	4.53 sec	79.0%	3.40 sec	84.5%
Quad	5.18 sec	79.2%	3.80 sec	84.5%

Digital test:	169ms	single site - Catalyst
Leakge test:	128ms	single site - Catalyst
Idd test:	60ms	single site - Catalyst
Signal Processing Overhead:	291ms	single site - Catalyst

All other test mixed-signal in nature

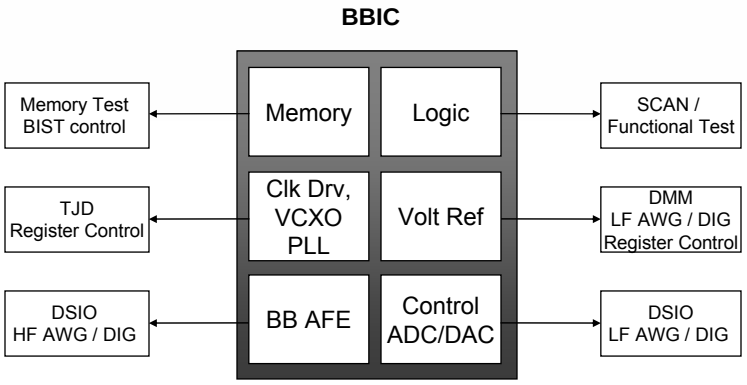
Pipeline Test Example

Site 1	O/S -PE	LKG -PE	Func -PE	PLL -TJD	DAC -Dig	ADC -AWG	Codec -LFAC
Site 2	O/S -PE	LKG -PE	Func - PE	Codec -LFAC	PLL -TJD	DAC -Dig	ADC -AWG
Site 3	O/S -PE	LKG -PE	Func -PE	ADC -AWG	Codec -LFAC	PLL -TJD	DAC -Dig
Site 4	O/S -PE	LKG -PE	Func -PE	DAC -Dig	ADC -AWG	Codec -LFAC	PLL -TJD

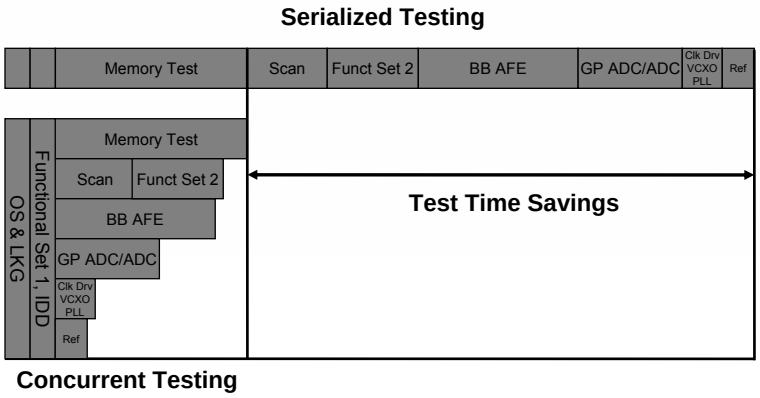
Concurrent Test

(Device Parallelism)

- Maximize number of functional blocks tested simultaneously on a single die.



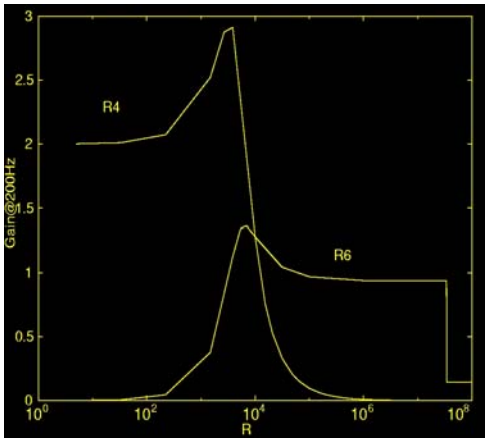
Concurrent Test



- Typical test time savings: 30-40%
  - Test scheduling not optimum
  - Signal Processing Overhead can't be pipelined
  - Sequencer limitations

Nonlinear Fault Effects Are Complex

-

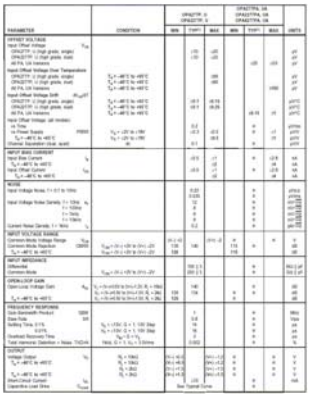


Source: Chatterjee

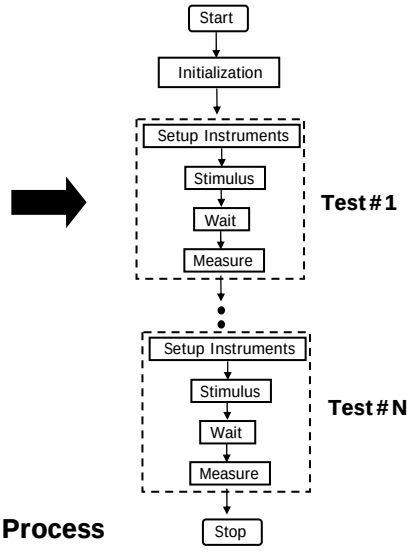
How do we define a failure?

ANY COMBINATION OF CIRCUIT/PROCESS PARAMETERS THAT CAUSES ONE OR MORE OF THE CIRCUITS SPECIFICATIONS TO BE VIOLATED IS DEFINED TO BE A FAILURE

Conventional Serial Testing



Datasheet



Source: Chatterjee

Serial Test Process

New *Alternate Test* Approach

Replace

Expensive specification tests, fully or partially

With

Low-cost, easy-to-perform alternate tests

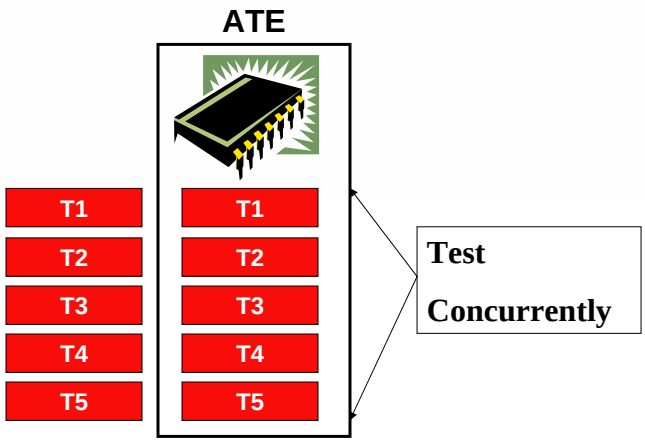
Such that

No yield loss

Same coverage as specification tests

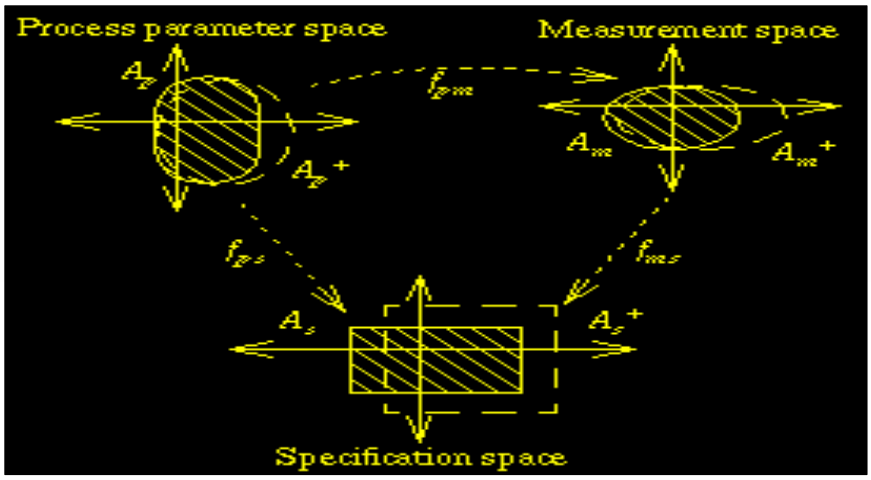
Source: Chatterjee

Benefit: Eliminate Test Bottleneck



Source: Chatterjee

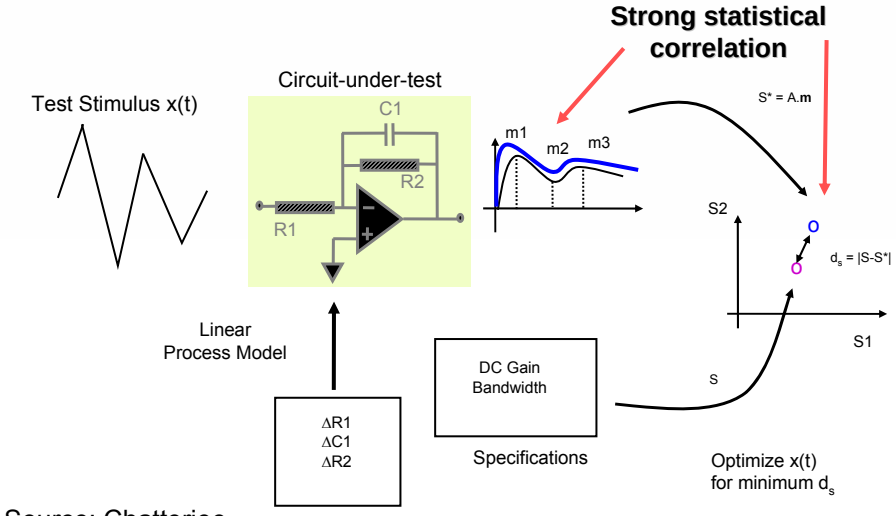
"Alternate Tests"



Mapping between measurement and specification spaces is derived using regression (MARS)

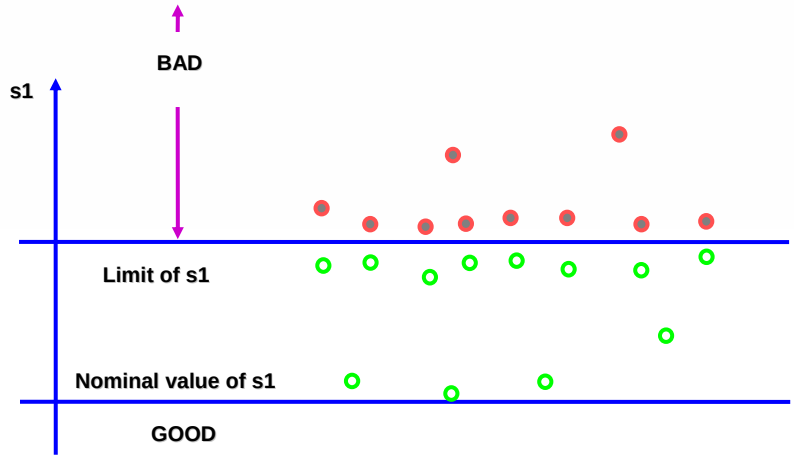
Source: Chatterjee

Transient Alternate Test



Source: Chatterjee

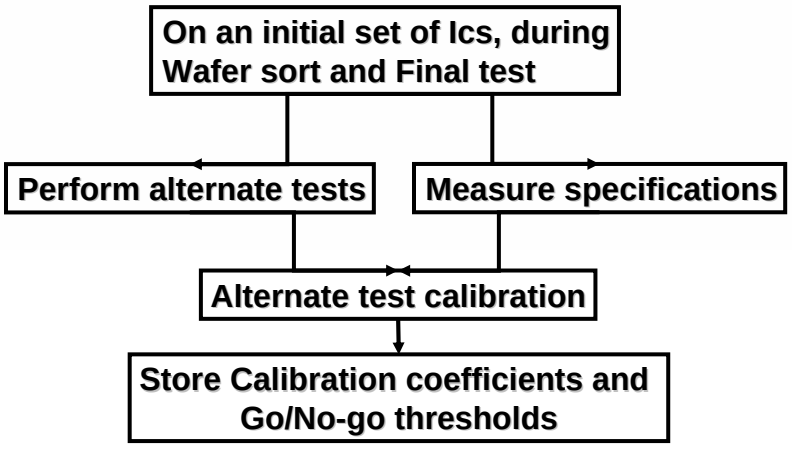
Coverage Modeling: How Good is the Alternate Test ?



GOAL OF ATPG: MAXIMIZE SENS OF TEST TO PROC/CKT PARAMETERS AT SPEC BOUNDARIES

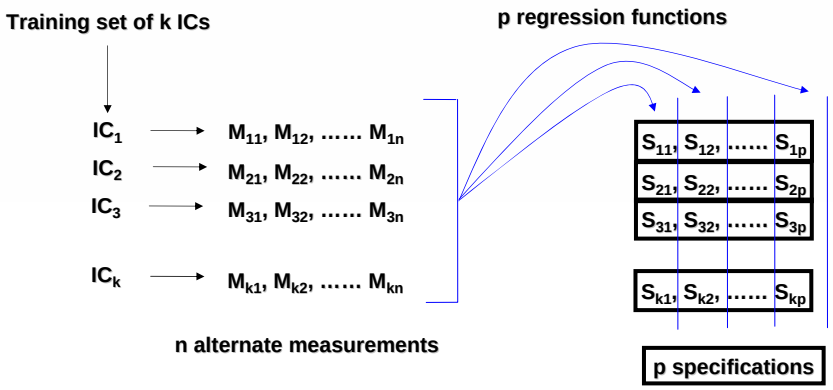
Source: Chatterjee

Test Calibration Procedure



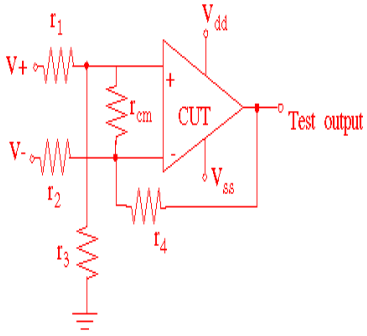
Source: Chatterjee

Test Calibration Using Regression

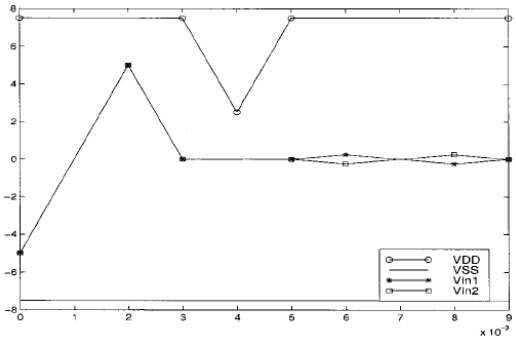


Source: Chatterjee

Application 1 : LM7101



Test Configuration

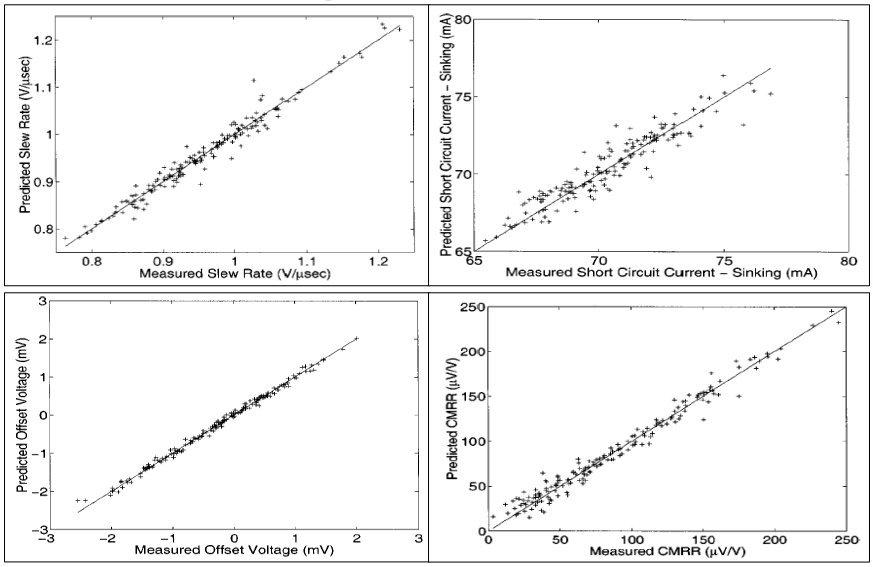


Signature Test Waveforms

4X test time reduction

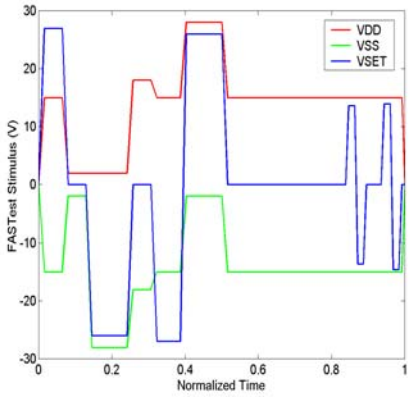
Source: Chatterjee

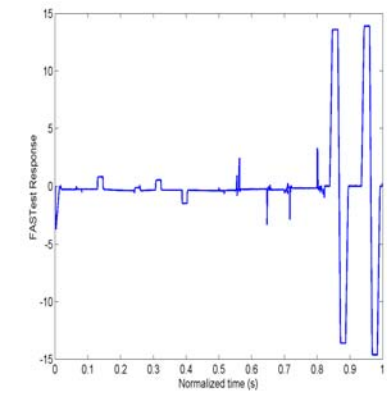
Concurrent Signature Test: Performance



Source: Chatterjee

Application 2: Precision Opamp

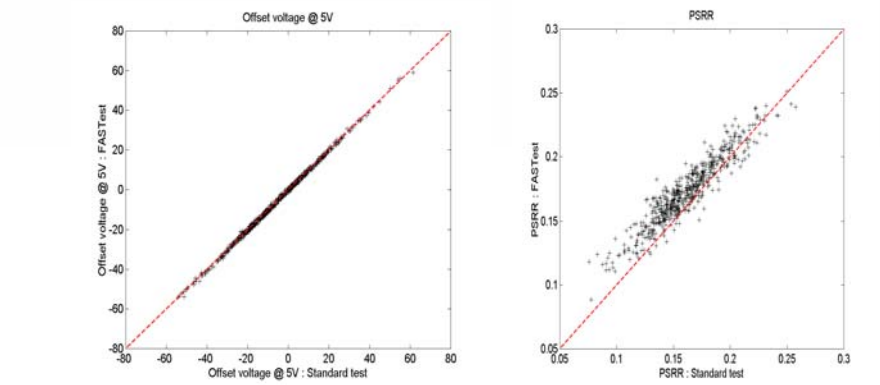




>3X test time reduction

Source: Chatterjee

Comparison with standard tests



Source: Chatterjee

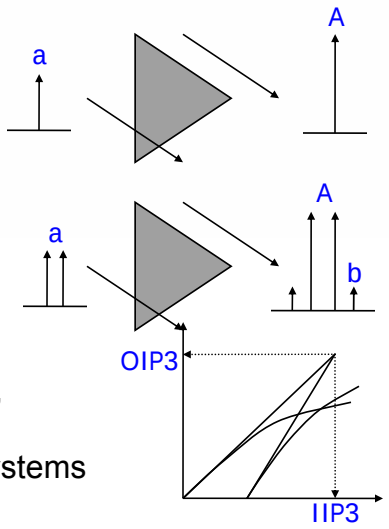
RF System Specifications

- Transmitter
  - System Gain, Gain Flatness, System IIP3, System NF, ACPR, Dynamic range, Modulation quality (Spectral mask), PA switching
- Receiver
  - System Gain, System IIP3, System NF, LO Stability, Sensitivity, BER/FER, EVM

Source: Chatterjee

Test setup : Gain and IIP3

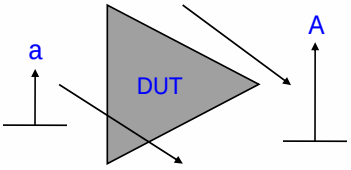
- Gain test setup
  - Single tone input
  - Gain = $20\log_{10}(A/a)$ dB
- IIP3 test setup
  - Two tone test
  - IIP3 = $a\sqrt{(A/b)}$
  - Non-linearity test for amplifiers, mixers
  - Also measured for complete systems

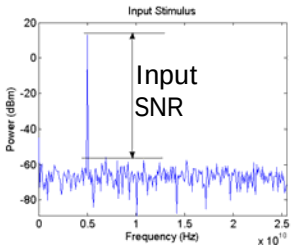


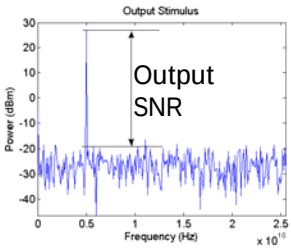
Source: Chatterjee

Test setup: Noise Figure

- SNR measurement
  - Signal to noise ratio
- Noise Figure = SNR_{in}/SNR_{out}
- Measure of noise added by the DUT



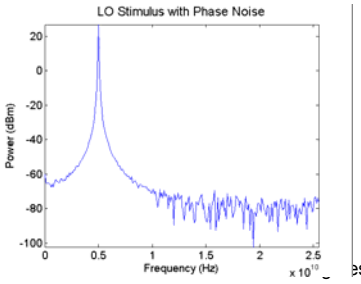




Source: Chatterjee

Other Specifications

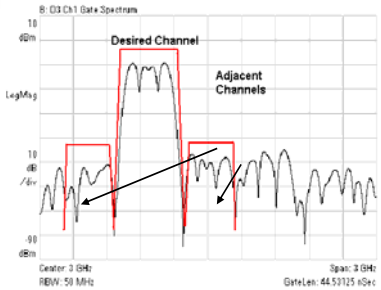
- Conversion Gain (CG)
  - CG = (IF o/p power)/(RF I/p power) dB
- LO Rejection
  - Isolation of LO signal to IF output
- Phase noise
  - Measure of PSD
  - Measured in dBc/Hz (ratio w.r.t. carrier/BW)



Source: Chatterjee

Transmitter Specifications

- Adjacent Channel Power Ratio (ACPR)
  - Amount of energy spilled to adjacent bands
  - Non-linearity measure of transmitter
- Dynamic Range
  - Range of power within which transmitter operates reliably
- Gain Flatness
- Modulation Quality
  - The spectral shape of the modulated signal

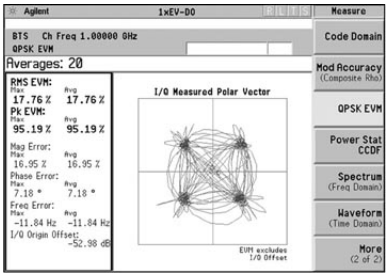


Source: Chatterjee

Receiver Specifications

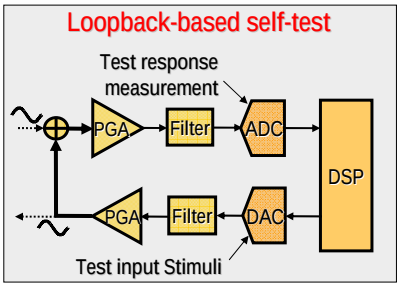
- LO Stability
  - Stability of LO frequency w.r.t. time and environment
- Sensitivity
  - Minimum signal level that the system can detect with acceptable SNR
- BER/FER
  - Error in received bits/frames
- Error Vector Magnitude
  - Quality of modulation
  - Denoted in %

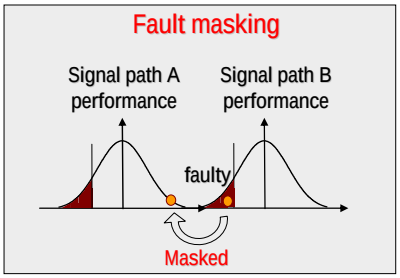
Source: Chatterjee



Loopback-Based Self-Test

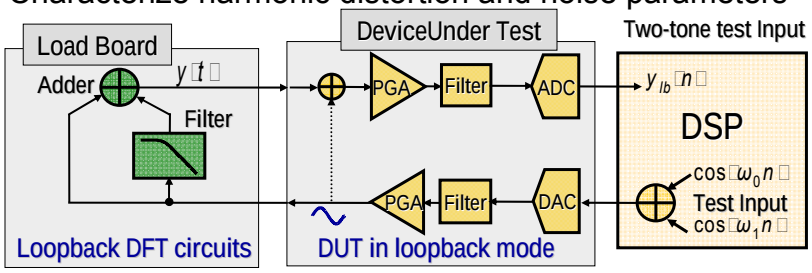
- Test several blocks at the same time using loopback
  - Measure the combined performance of path
- Advantages
  - Reduced test time
  - No performance degradation from insertion of test points
- Limitations
  - Combined response of non-functionally related paths
    - Distortion and noise of signal paths are additive
    - Fault masking
    - Misclassification
- *Need to extract performance parameters of individual signal paths*



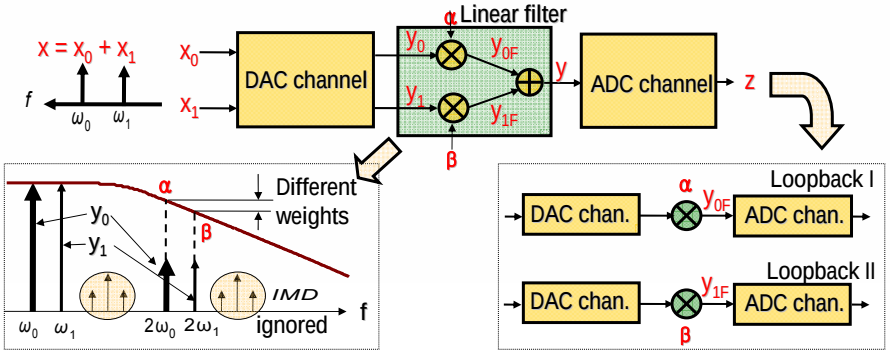


Loopback + DFT Scheme

- Provide dynamic performance parameters of individual signal paths
  - Avoid yield loss due to fault masking
- DFT circuitry on loadboard or on chip
  - Implement analog filter and adder as DFT circuitry
    - Reduce silicon cost with minimal pin count (2 dedicated pins)
    - Compatible with existing loopback scheme
  - Characterize harmonic distortion and noise parameters



Performance Parameter Extraction

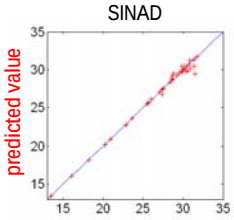


- Loopback response: sum of input/output channel performance
  - Loopback = performance (DAC) + performance (ADC)
- Excite ADC channel with unknown input (output of DAC)
  - Scaled by known filter magnitude response – by different scaling factors
  - Analyze correlation between the obtained loopback responses
    - Loopback I = scaling factor α * performance (DAC) + performance (ADC)
    - Loopback II = scaling factor β * performance (DAC) + performance (ADC)

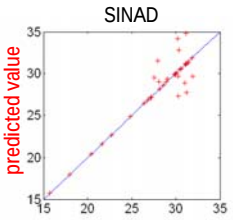
Validation: Hardware Measurements

- Broadband modem IC
  - Tx/Rx data rates upto 80MSPS
- Programmable 3-pole filter
  - Bypassed in normal mode
- Faults injected by
  - Reconfiguring Tx/Rx gain
  - Sweeping power supplies and input amplitude

	Prediction Errors	
	DAC channel	ADC channel
SNR	0.32dB	1.35dB
THD	0.31dB	0.63dB
SINAD	0.25dB	0.94dB

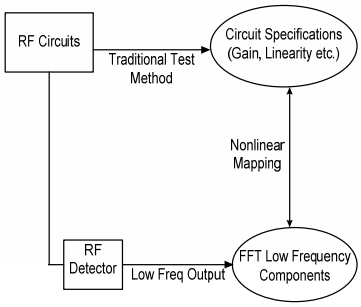


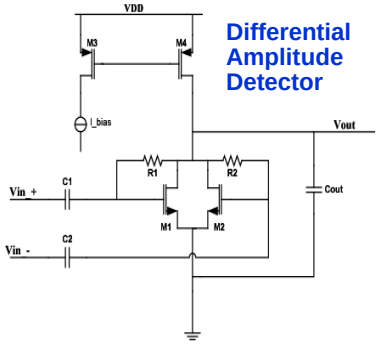
(a) DAC channel



(b) ADC channel

RF Built-In Test using Amplitude Detectors





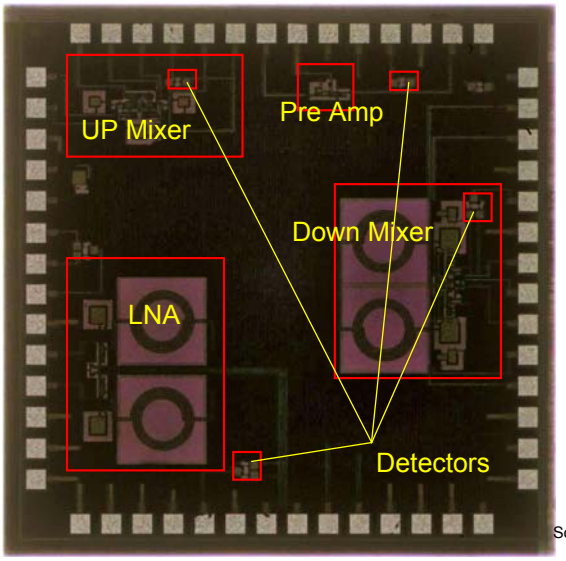
Differential Amplitude Detector

- Alternate test methodology
- High input impedance (7.6KOhm@1GHz) for detector
- Detector output mapped to RF circuit specifications
- Low frequency output signal (sampling frequency of 10MHz for mixer test, DC for amplifier test)
- Strong correlations with RF circuit parameters

Die Photo of 940 Mhz Transceiver

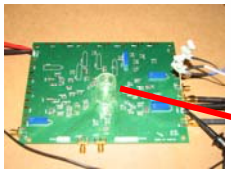
(UMC 0.18µ CMOS)

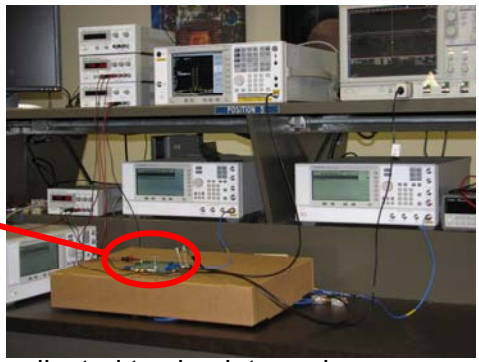
10 MHz output from sensors used to predict specifications



Measurement Setup

- Agilent E8257D Signal Generator
- Agilent E4448A Spectrum Analyzer
- Tektronix DPO 7104 Digital Oscilloscope



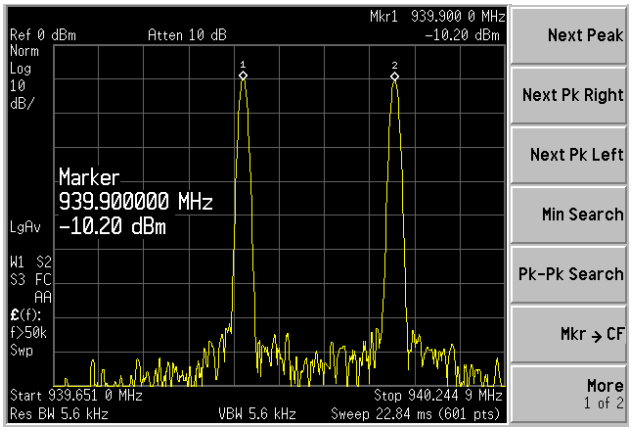


- Many tuning knobs designed
- Almost every bias point can be adjusted to simulate real case chip variations

  (mismatch, supply drop etc.).

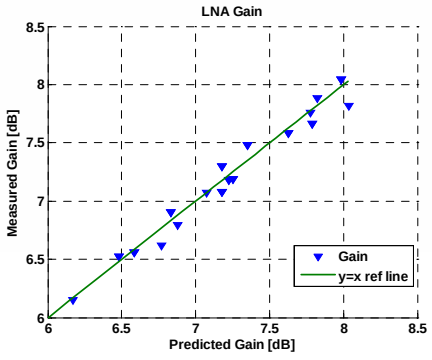
Test Procedure: Input Signal

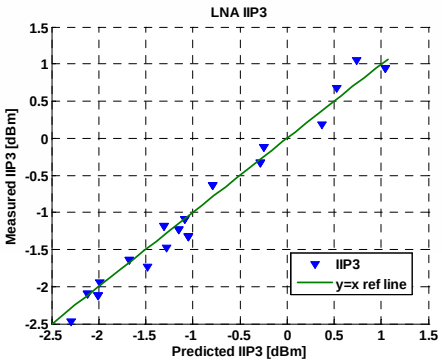
- Two tone signal of 939.9MHz and 940.1MHz at -10dBm



- Tone signal selection is not limited to these values
- Depends on chip applications and specifications to be tested

Measurement Results: LNA in RX



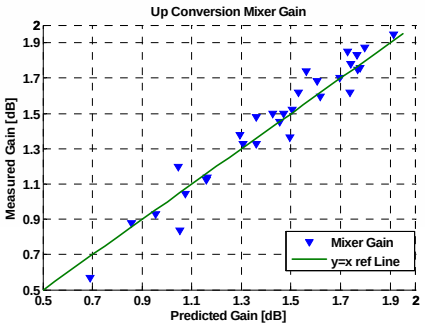


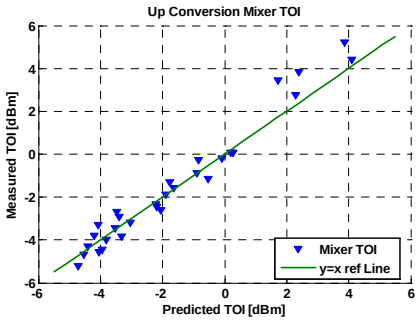
	LNA Gain	LNA IIP3
RMS Error	0.09 dB	0.15 dBm
Relative Error	4.8%	4.4%

$$RMS_{error} = \sqrt{\frac{1}{N}\sum (P_{true} - P_{estimated})^2}$$

$$Relative_{error} = \frac{RMS_{error}}{Variation\ Range}$$

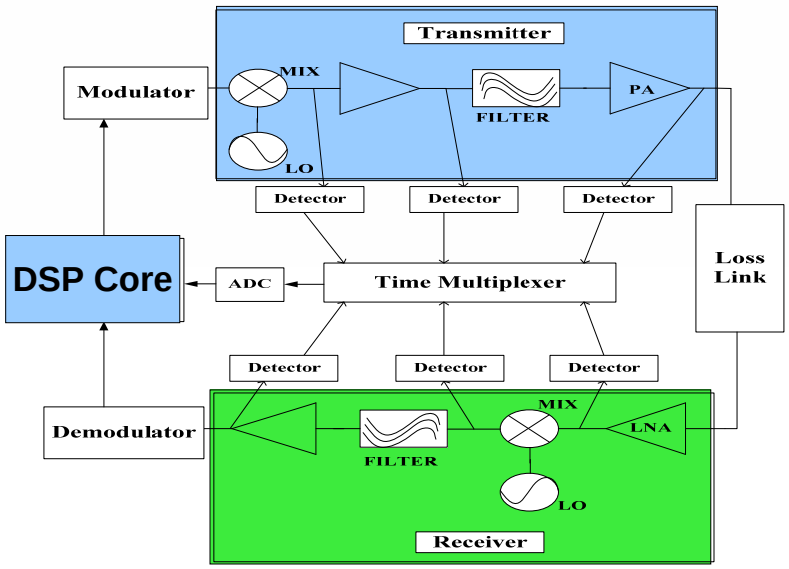
Measurements Results: Up Mixer in TX





	Mixer Gain	Mixer TOI
RMS Error	0.09 dB	0.61 dBm
Relative Error	6.3%	5.9%

Loopback RF Test



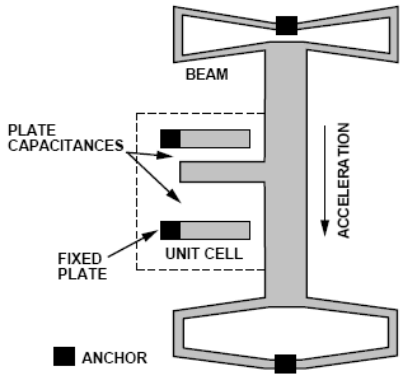
Testing Non-Electrical Modules – MEMS

- Develop new ways of characterizing and testing MicroElectroMechanical (MEM) systems
  - Use gravity to provide mechanical input
- Reduce test time and cost by using electrical tests to characterize and test mechanical subsystem
  - Correlate electrical and mechanical tests
- Develop and validate approach with measurements on commercial MEM accelerometer

Target Accelerometer

- Analog Devices ADXL204
- Dual-axis
- Full scale reading of +/- 1.7 g, 0g => 1.65V
- Saturates beyond full scale – non-linear response
- MEM Capacitive Transducer

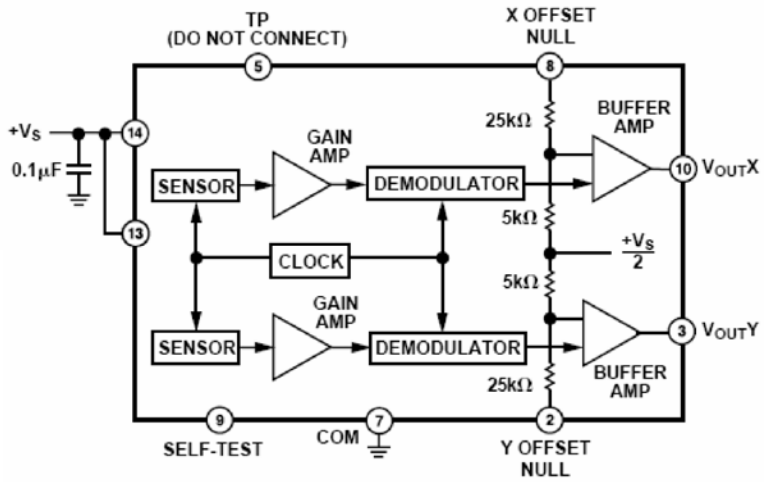
Capacitor Plates



- Distance between capacitive plates varies with acceleration
- C varies with 1/d

Source: ADXL204 Product Manual, Analog Devices Inc.

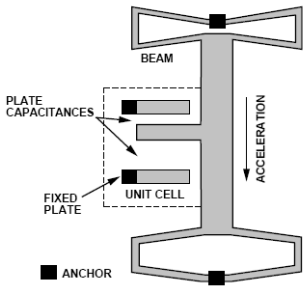
Block Diagram



Source: ADXL204 Product Manual, Analog Devices Inc.

Accelerometer Function

- Simple first-order model
- Variation in plate spacing due to displacement, d (caused by an applied acceleration), produces a voltage on output pin
- $\delta V / d$

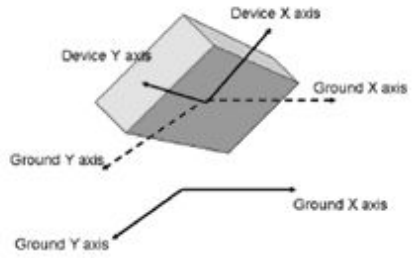


Conventional Mechanical Stimulus

- "Shaker": standardized acceleration generator
  - Compare output voltage with expected value from standard acceleration
  - Expensive
- Turn-table: centrifugal force
  - Centrifugal acceleration from rotation
  - Also expensive

Our Mechanical Stimulus

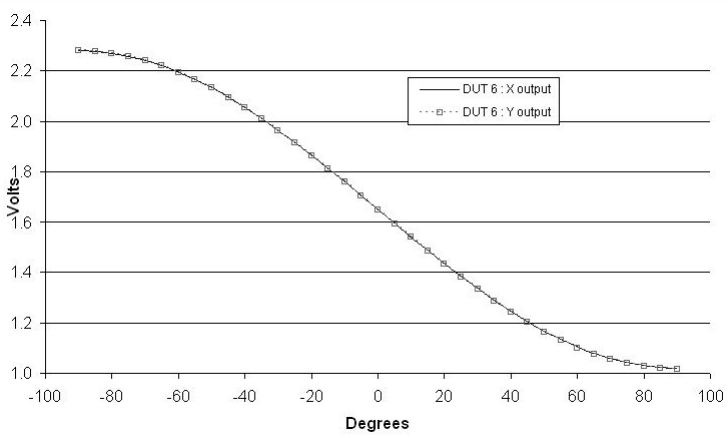
- Uses fact that the DUT is a dual axis device
- Tilt device to change acceleration due to gravity ($g=9.81 m/s^2$) on different axes
  - 1.65 V when horizontal



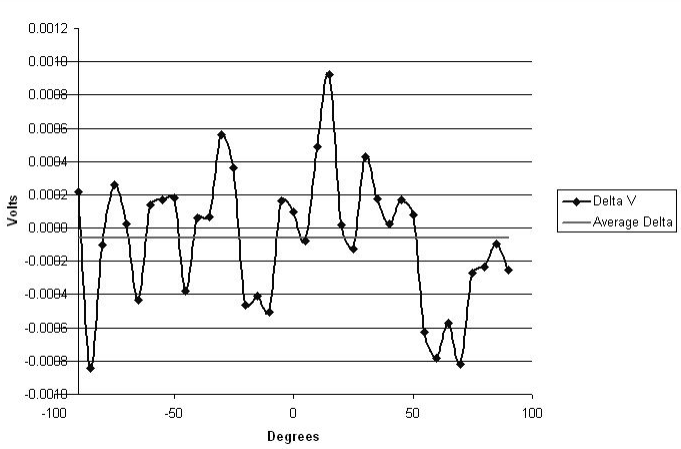
Methodology

- Measure V_x and V_y for $\theta x=0^o$, $\theta y=0^o$
- Change the orientation of the DUT physically, imparting a gravitational acceleration component on it
- $g_{x\_eff} = g(\sin\theta_x+\cos\theta_x)$
- $g_{y\_eff} = g(\sin\theta_y+\cos\theta_y)$
- Obtain marker points and interpolate to obtain a curve

Mechanical System Calibration



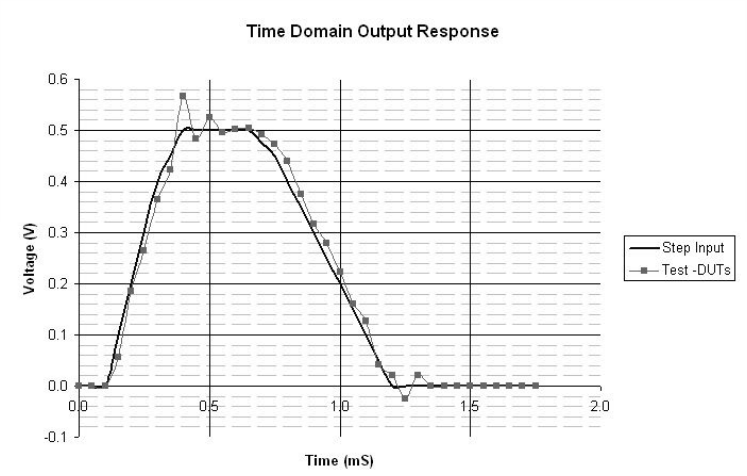
Error in Mechanical Stimulus



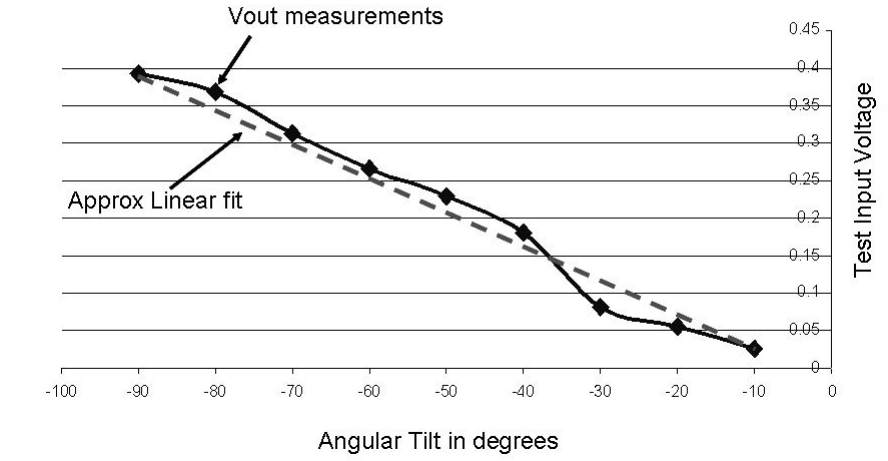
Calibration using Electrical Test

- Unit has test input pin connected to capacitor elements
- Applied electrical signal produces electrostatic force, producing displacement
- Resulting change in output voltage
- Measurements of result of input step
  - 10,000 runs for each DUT
  - National Instruments platform

Averaged Results of Input Step



Test Input versus Angular Tilt



Electrical vs. Mechanical Tests