IDDQ Testing

Sungho Kang
Yonsei University
Outline

- Introduction
- Defect Detection with IDDQ
- Fault Detection with IDDQ
- IDDQ Test Patterns
- IDDQ Instrumentation
- IDDQ Limit Setting and Characterization
- Design Consideration of IDDQ Testing
- Delay Faults and IDDQ Testing
- Defect Diagnosis with IDDQ Testing
- Conclusion
What is IDDQ?

- IDDQ Testing is the process of measuring IDDQ
- IDDQ is the IEEE symbol for the quiescent power supply current in an MOS circuit.
  - In our cases we are referring to IDDQ in CMOS ICs
- Perfect CMOS circuits have IDDQ values typically less than 100 nA
  - There is no direct conducting path between Vdd and Vss
- Most CMOS IC defects elevate IDDQ several orders of magnitude greater than non-defective circuit
- IDDQ Testing is the most sensitive way to detect the majority of CMOS defects
Limitation of Logic Testing

- Gate to source short in N1 of a NAND gate
- Voltage testing based on stuck-at fault model cannot fully detect transistor level defects (about 40%)
Limitation of Logic Testing

❖ Voltage waveform of output Y

❖ Use current rather than voltage
IDDQ : Quiescent Current

- When CMOS is not actually switching, one transistor in CMOS pair is always off.
- $I_{ddq}$ refers to the quiescent power supply current drawn by CMOS circuits in stable states.
- Draws only a leakage current on the order of nA.
- $I_{ddq}$ can become as high as several mA.
High IDDQ Indicates a Defective IC

❖ When CMOS is not actually switching, one transistor in CMOS pair is always off
❖ Draws only a leakage current on the order of nA
❖ IDDQ can become as high as several mA
Why IDDQ Testing?

- Chips with transistor leakage faults may pass even 100% fault graded functional production tests
- Reliability studies involving life test
  - Burn-in testing
  - Leakage faults could develop into hard stuck-at faults
  - Source of long time failure
- As intensity and complexity of VLSI are increased
  - Reliability becomes very important
  - Aerospace, Automobile
Advantages of IDDQ Testing

- **Direct Observability**
  - >>50% of transistors tested with only a few vectors
- **Detection of defects that do not cause functional failure**
  - Identification of subtle defects and failure mechanisms in addition to those that affect logic functions
- **Greatly increased detection of common physical defects**
  - Gate oxide shorts, Interconnect shorts, Interconnect opens
- **100% coverage of stuck-at faults for many designs**
- **Reduced test vector count**
- **Simplified test vector generation and fault simulation**
- **No additional on-chip circuitry required**
Three Board Test Strategies

**Functional Testing**
- This approach tries to stimulate the product as a customer would use it.

**Logic Structural Testing**
- This approach was started as a method of allowing computer algorithms to recognize failures and to automatically generate test patterns.
- Fault model testing does not ask whether the part functions, but rather is it free of faults that would cause logical failure.

**Physical Defect Testing**
- This approach lists all defects that can occur, then performs tests specific for each defect.
- This is the only approach that leads to zero defects in testing.
What is Function Testing?

- Test programs often evolve from the design stage
  - Designers exercise their part to see if it works
- The stimuli that designers use to verify the design becomes the core of the final test program used in production
- For perhaps the majority of electronic products in the world
  - Test programs are developed for complex products in this manner
- The functional test set originated by the designer is a very poor foundation for the testing of complex product
- Functional testing is a major contributor to high defect escape levels
Why Functional Tests Inadequate?

- Why are functional tests inadequate for detecting random detects?
- They can not duplicate all customer possibilities in complex electronic systems
  - It is an intractable problem
- Motorola estimated that it would take a least 2 million years to fully functionally test the 8-bit 6800 microprocessor
- An equally strong reason for not relying on functional test patterns is that many CMOS defects will escape a functional test set and will cause customer failure
What is Structural Testing?

- Structural testing identifies failure modes and devises quantifiable tests that are tailored for detection of these failure modes
- Structural testing asks not whether the design is correct, but is the hardware without flaw
- Structural testing allows the test program to be quantified as to how many of the possible defective situations have been examined in the product under test
Fault Models

- Fault models use an abstract assumption of how a product fails
- Computers can understand a fault model but have greater difficulty dealing with actual defects
- Fault modeling preceded physical defect testing and for partly historic reasons is the more dominant structural method

Classifications:
- Stuck-at
- Bridging
- Transistor stuck-on
- Transistor Stuck-off (CMOS stuck-open)
- Delay
- Leakage
Testing for Zero Defects

- All test sets must be able to evaluated against their ability to detect real defects
- A strong failure analysis effort must be maintained in order to determine exactly what the real defects are
- Examples of real CMOS IC defects
  - Gate oxide shorts
  - Power supply bridging defects
  - Signal Node bridging defects
  - Leaky PN junctions
  - Punchthrough
  - Parasitic transistors
  - Open drain and source defects
  - Open gate defects
## Test Method Success for Defects

**P-poor; F-fair; G-good**

<table>
<thead>
<tr>
<th>DEFECT</th>
<th>FUNCTIONAL</th>
<th>STUCK-AT</th>
<th>STUCK-OPEN</th>
<th>$I_{DDQ}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gate shorts</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>G</td>
</tr>
<tr>
<td>Bridges</td>
<td>F</td>
<td>F-G</td>
<td>G-F</td>
<td>G</td>
</tr>
<tr>
<td>Parastics &amp; p-n leakage</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>G</td>
</tr>
<tr>
<td>Punch through</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>G</td>
</tr>
<tr>
<td>Open drain</td>
<td>P</td>
<td>F</td>
<td>G</td>
<td>F</td>
</tr>
<tr>
<td>Open gate</td>
<td>F</td>
<td>G</td>
<td>G</td>
<td>F</td>
</tr>
<tr>
<td>Open t-gate</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>G</td>
</tr>
</tbody>
</table>
Recent IDDQ Achievements

- >50% reduction in IC production cost due to improved screening of defects at wafer test/sort
- Major assembly line and customer field quality improvement after implementation for scan ASICs with >99% stuck at fault coverage
- >50% drop in burn-in rejects and resultant ability to reduce burn-in
Types of CMOS Defects Mechanisms

- **Shorts**
  - Electrical connections between circuit nodes that are not usually zero ohms

- **Opens**
  - Breaks in the interconnect materials

- **Degradations**
  - Defects mechanisms that may alter circuit performance and may produce shorts or opens

- **IDDQ Leakage**
  - Can include all of the above
Types of CMOS Short Defects

Gate Oxide Ruptures
- Destructive breakdown of a particular spot in the transistor thin oxide dielectric, leading to a relatively low impedance connection between the gate and the area underneath the gate

Bridged Interconnect Material
- May include any combination of metal polysilicon or diffusion material inadvertently connected to another interconnect material

Punchthrough
- The drain/substrate depletion region extends through the channel length and reaches the source

Parasitic Transistor Leakage
- Drain to source conduction caused by inadvertent channel inversion of a parasitic transistor

Leaky PN Junctions
- High leakage current or low breakdown voltage of reverse bias PN junctions caused by defects such as stacking faults or metallic ion contamination
PMOS Gate Oxide Short Properties

- Ig versus Vgs curves are nonlinear, even n-poly gate to n-substrate shorts
- Elevated IDDQ is found in at least one logic state
- Degradation in logic signal strength is found similar to that of nMOS gate oxide shorts
What about Gate Oxide Shorts?

- The actual electrical properties of the short vary with transistor type and geometric location on the gate
  - n-poly to p-channel and n-poly to p* diffusion cause a PN junction diode
  - n-poly to n* diffusion causes an ohmic short
  - n-poly to n-substrate causes a nonlinear IV response probably caused by the action of channel inversion
- The effect of gate shorts on logic voltages is to weaken them and not generally to cause low frequency logic error
- The absolute method of detection measures IDDQ for the two logic states that place the input gate node in both the logic 1 and logic 0 states
Gate Oxide Short Detection

- Data show a strong reliability risk in passing circuits with gate oxide shorts even through they appear to meet all functional specifications.
- IDDQ testing with 100% node state activity is the only way to guarantee detection of gate oxide shorts.
  - IDDQ can be < 0.5uA at Vdd = 5V.
- 63% of the ICs with gate shorts were not detected by the functional test set.
- Of the initial 35 ICs with gates shorts, 31% were reliability problems.
- Of the 22 ICs that escaped the functional test patterns, 50% caused reliability failure.
- Dynamic screens are very efficient for activating gate short defects.
CMOS Bridge Defects

- Bridging defects usually refer to electrical shorts between interconnect material.
- The frequency of bridge defects is fab site and wafer lot dependent, but is a significant defect.
- Bridge defects can be difficult to detect; they can be detected with logic vectors or IDDQ Measurements.
CMOS Bridge Defects

A    B    C    D    E    V_{BR}    V_{OUT}    IDDQ
0    0    0    0    0    5.00    0.00    805pA
5    5    5    5    5    0.00    5.00    105pA
0    0    0    5    5    3.79    0.00    509uA
0    0    5    5    5    2.83    0.08    537uA
0    5    5    5    5    0.98    5.00    317uA
5    5    5    0    5    1.74    4.55    356uA
5    5    5    0    0    3.81    0.00    345uA
The intermediate bridge voltage quickly acquires a strong logic strength after passing through one logic load gate.

Elevated IDDQ occurs in at least one logic state.

A functional (or SAF) test set will detect a failure in at least one logic state if observability paths are correct.

The IDDQ test is more efficient because it does not require observability at a primary output node.
Strategy for Bridge Defect Detection

- Identify from the mask layout the potential node pairs that have a reasonable probability of bridging by a defect
- CAE tools are capable of this, but have not yet been demonstrated for the bridge problem. Therefore manual inspection is unfortunately all that you presently have
- Use the paired node defect table to generate test vectors that drive these nodes to opposite logic states and then measure IDDQ for each vector
Each of the following open circuit categories produce different electrical effects and require different test strategies:

- Open transistor gate
- Open transistor drain or source
- Transmission gate open circuit

Open circuit defects defy conventional logic response testing and significant cases exist of escapes to the next level of assembly and from field failures.

The open circuit may occur during fab from particle contamination or can occur from a reliability failure mechanism.

Metal stress voiding and electromigration are two major open circuit reliability concerns.
Open Transistor Gate Responses

- Many open circuit defects have small clefts such as stress voids and electromigration
- These small clefts exert a different electrical response than expected for a large open circuit defect
- Significant electrical coupling of the signal has been found by three independent labs
- Why do circuits still operate with open circuit defects?
  - The mechanism is not coupling capacitance across the narrow clefts as calculations show sub-femtofarad values
  - CMOS circuit data show that the open circuit electrical response acts as a low frequency filter with variable upper cutoff frequencies from a few KHz to the MHz region - The mechanism is not capacitive coupling
Detection of Transistor Gate Open

- The primary electrical response to a gate open circuit is that of a timing or delay fault
- A secondary electrical response is elevation in IDDQ for certain designs
- The delay fault approach is very cumbersome considering the combinations of delay paths in a typical VLSI circuit
- The IDDQ method has been useful for gate open circuit detection
- Drain/Source Open circuit defects
  - An open circuit in the drain or source of a transistor can cause a memory effect on the CMOS circuit: Stuck-open fault
  - Any defect that doesn’t allow charge to pass in a transistor causes a stuck open fault; it is very difficult to guarantee detection of this fault
CMOS Stuck-Open Faults

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0*</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1*</td>
</tr>
</tbody>
</table>

VDD

OPEN

C

A

B
CMOS Stuck-Open Faults

- The defect density of stuck-open faults is not accurately known and is a function of wafer processing variables.
- The defect does act as a classic stuck-open fault at tester clock frequencies (> 1MHz).
- The effected drain node is put in a high impedance state for a given test vector and that node then drifts with a long time constant (2-5 s) to a steady state value.
- IDDQ was elevated by two mechanisms:
  - A slow drift of the node in the high-Z state turned on the load gate. This reached a value of 100 uA in about 20ms.
  - A faster effect was found that elevated IDDQ in about 2ns. The effect was due to drive contention between two gates.
Detection of Stuck-Open Faults

- ATPG of a 2-vector sequence is a guaranteed approach, but intractable for VLSI
- IDDQ elevation occurs, but is a function of the logic design and a chance activation of the defect
- Scrambling (randomizing) a functional test vector set can increase the probability of detection i.e. A binary up-count is the worst case order to detect a stuck-open fault
- Layout can be modified to decrease the probability of a metal open
- The attack on the stuck-open fault should concern itself with combining IDDQ measurements, layout design for reliability techniques and a functional test set
Memory Testing with IDDQ

- Gate oxide shorts are a strong concern in memories
  - A large gate oxide area exists
- IDDQ testing in memories is efficient for many types of defects; gate shorts, stuck cells, or bridged nodes
- An IDDQ measurement of the checkerboard and inverse checkerboard patterns provides rapid coverage of the majority of transistors (but not all)
- A memory node-state test is relatively short, O(4n)
- IDDQ measured at the checkerboard and inverse checkerboard patterns will detect all gate shorts in the four-transistor cross-coupled effort
- To detect gate shorts in the write access transistors, IDDQ must be measured at each phase of the write signal
- IDDQ is very powerful for detecting memory defects in CMOS SRAM’s with a minimum of effort
All short circuit defects are most efficiently and exactly detected by the IDDQ test measurement.

It is necessary that a test vector activate the defect (controllability) and IDDQ measurement be taken.

Functional or SAF test sets will not defect many of these defects except by chance since the primary effect of these short defects can be either weakening of logic voltages or a bridge masking effect.

None of the open circuit defects are guaranteed detectable by the IDDQ test, but neither are functional or stuck-at fault test sets.

The IDDQ test increases the probability of open circuit defect detection with the advantage of a small test set.

Certain defects do not elevate IDDQ nor are they efficiently detected by a functional type test set.
ICs That Have High IDDQ

- Reliability concerns
  - These ICs often have lower reliability (early functional failure)
  - High current states may cause premature battery failure

- Quality/Yield concerns
  - Current may be symptomatic of a significant problem
  - Causal defects/mechanisms may worsen in time
  - Ignoring defects produces
    - Inaccurate quality/performance measurements/metrics
    - Incorrect test/experiment conclusions
Fault Detection with IDDQ

- Several Faults can be used as test metrics
  - Stuck-at faults
  - Bridging faults
  - Stuck-open faults
  - Stuck-on faults
  - Delay faults
- Faults are considered because
  - Many test standards evolved from faults not defects
  - Computer simulation programs typically use faults
  - Legal and vendor/customer requirements often specify fault coverage
SAF Detection with IDDQ Testing

- SAFs are defined for CMOS ICs as logic failures that occur when a logic gate input or output node is bridged to one of the power supply rails with a zero ohm connection.
- SAFs are time invariant and therefore cause failures at any frequency of operation.
- SAFs have been the backbone of DFT throughout the 1970-80 era and despite advances in defect knowledge, the SAF is still a required metric by many customers.
- A significant observation was that SAF coverage can be obtained with IDDQ testing using test vector sizes that may be 1% of conventional vectors.
Detection of SAFs

Fault Detection

[Diagram showing a circuit with labels for nodes A, B, N_A, N_B, P_A, P_B, and I_{DDQ}.]
Detection of Redundant Faults

- A B C D E F E s-a-1 detection
- 0 0 0 0 0 0 Yes
- 1 1 1 1 1 1 No
## Detection of Multiple Faults

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>D s-a-1 detection</th>
<th>E s-a-0 detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

![Logic Diagram](image-url)
Comments on SAF Testing

- The SAF is a poor metric for CMOS circuits, but it lingers in the industry
- The IDDQ test easily satisfies most SAF requirements
- Previously untestable circuits, such as typical microprocessors can now be evaluated for those customers who demand it
- Many who try the IDDQ test technique for SAF coverage have discovered the additional advantages of IDDQ testing such as
  - 100% gate oxide short coverage
  - Better bridge and delay defect coverage
  - Detection of some open circuit defects
  - Detection of certain design properties
  - Detection of certain process problems
Detection of Bridging Faults

- Bridging faults and bridging defects are similar except that bridging faults assume a zero ohm defect, but bridging defects assume that any impedance is possible.
- Logic (voltage) detection problems are the same as those for other types of defects
  - Identification of plausible node bridges
  - Generation of a test vector that activates the defect (controllability)
  - Setting of observation paths to observe logic response
- IDDQ testing eliminates the 3rd requirement and greatly simplifies the test.
## Logic vs Current Testing

### Bridging faults detected with voltage sensing

<table>
<thead>
<tr>
<th>ISCAS85</th>
<th>Patterns</th>
<th>Faults</th>
<th>Missed</th>
<th>%detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>c17</td>
<td>6</td>
<td>74</td>
<td>1</td>
<td>98.7%</td>
</tr>
<tr>
<td>c432</td>
<td>53</td>
<td>3968</td>
<td>174</td>
<td>95.6%</td>
</tr>
<tr>
<td>c499</td>
<td>58</td>
<td>7806</td>
<td>362</td>
<td>95.4%</td>
</tr>
<tr>
<td>c880</td>
<td>67</td>
<td>7909</td>
<td>99</td>
<td>98.8%</td>
</tr>
<tr>
<td>c1355</td>
<td>87</td>
<td>12103</td>
<td>62</td>
<td>99.5%</td>
</tr>
<tr>
<td>c1908</td>
<td>108</td>
<td>14203</td>
<td>338</td>
<td>97.6%</td>
</tr>
</tbody>
</table>

### Bridging faults detected with current sensing

<table>
<thead>
<tr>
<th>ISCAS85</th>
<th>Patterns</th>
<th>Faults</th>
<th>Missed</th>
<th>%detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>c17</td>
<td>5</td>
<td>75</td>
<td>0</td>
<td>100.0%</td>
</tr>
<tr>
<td>c432</td>
<td>40</td>
<td>4086</td>
<td>56</td>
<td>99.59%</td>
</tr>
<tr>
<td>c499</td>
<td>99</td>
<td>8167</td>
<td>1</td>
<td>99.99%</td>
</tr>
<tr>
<td>c880</td>
<td>27</td>
<td>8006</td>
<td>2</td>
<td>99.97%</td>
</tr>
</tbody>
</table>
Transistor Bridging Faults

- A CAD tool called QUIETTEST selects a small subset of IDDQ vectors from a large set of previously generated SAF vectors.
- This IDDQ subset detects six leakage faults per transistor.
- Fgs : the leakage fault between gate and source
- Fgd : the leakage fault between gate and drain
- Fsd : the leakage fault between source and drain
- stuck-on
- Fbs : the leakage fault between bulk and source
- Fbd : the leakage fault between bulk and drain
- Fbg : the leakage fault between bulk and gate
Detection of CMOS Stuck-Open

- CMOS stuck-open faults shows that IDDQ has a significant probability of detecting CMOS stuck-open faults due to
  - Transistor contention caused by the memory effect of the SOF
  - Floating nodes caused by the SOF turn on CMOS transistor pairs
- Consider SOFs as high likelihood non-target faults of IDDQ testing
Delay faults denotes the collection of all defects that cause a circuit fail at a frequency less than that of a defect-free circuit.

Delay faults will pass functional tests at DC.

Delay faults are an increasing concern as we build products whose clock drops from the 100ns period to the 5-10ns period.

What does the relatively slow speed IDDQ test have to do with the detection of delay faults?

A large percentage of defects that cause delay faults are sensitive to detection by the IDDQ test.
IDDQ Testing and Delay Faults

- Delay faults are caused by defects that do not elevate IDDQ such as
  - Increased resistance in interconnects such as in the lines themselves or in the vias
  - Increased capacitance in interconnects such as defective bulging of parallel tracks
  - Shifted transistor thresholds
- For very high speed applications, we begin to worry about the statistical spread of what were previously considered normal variations
Defect Testing and Fault Models

- Goal for quality improvement is increased detection of defects
  - So test goal is high defect coverage
  - Simple effective IDDQ testing can be based on defect behavior for target defects
- Fault models are usually considered necessary for test generation and grading
  - A model is only as good as its ability to represent physical defects
  - The SAF model is poor for CMOS ICs
  - Alternatives include transistor short models and leakage fault models
Summary of Fault Detection

- IDDQ detects SAFs with substantially reduced TPs
  - For conventional test patterns
    - $P(\text{detection}) = P(\text{control}) \times P(\text{observation})$
  - With IDDQ test patterns
    - $P(\text{detection}) = P(\text{control})$
- IDDQ detects logic gate bridging faults more efficiently than SAF vectors, but the reduction in test patterns is not as dramatic
- IDDQ detects transistor stuck-on faults with a test pattern reduction that is on the order of 0.5%
- IDDQ is a tool to reduce stuck-open faults, but the fault coverage depends on the type of physical defect; 100% coverage is not guaranteed for all types of defects
- IDDQ removes a large fraction of defects that cause delay faults
IDDQ Test Patterns

- IDDQ testing refers to the measurement of quiescent power supply current in CMOS circuits to cover many types of defects.
- An IDDQ test pattern set consists of test vectors used for IDDQ measurement.
- If the CUT contains defects targeted by an IDDQ test vector, the CUT draws excessive quiescent current for the test vector.
- IDDQ test patterns are also used for defect diagnosis and can be lead to specific defective devices in the IC.
Requirements for IDDQ Test Patterns

- A test vector must create one or more low resistance paths from Vdd to Vss in the presence of target defect(s).
- The acceptable number of IDDQ test vector is several orders of magnitude less than what may be reasonable for voltage based logic testing.
Obtaining IDDQ Test Patterns

- Use the entire functional or structural test pattern and measure IDDQ for each test vector
  - Every Vector IDDQ Test Patterns
- Select a subset of the functional or structural test pattern and measure IDDQ for selected test vectors
  - Selective IDDQ Test Patterns
- Generate test patterns specifically for IDDQ testing to supplement the traditional functional or structural testing
  - Supplemental IDDQ Test Patterns
Every Vector IDDQ Test Patterns

- Functional or structural test patterns are used
  - the only modification necessary is addition of control statements to slow down the test application rate

- Advantages
  - No additional test generation effort or software is required
  - No fault model is necessary; coverage for all defects that are sensitized and to which IDDQ is sensitive are covered
  - Effective for both combinational and sequential circuits

- Disadvantages
  - Coverage of defects for which IDDQ testing is effective is constrained by the quality of functional or structural test patterns
  - Defect coverage is not quantified
  - For dynamic circuits, it may not be allowable to slow down testing during the application of at least some of the vectors
  - Circuit is not tested at system speed
  - The number of test vectors is usually too large for the methodology to be practical in large volume production
Selective IDDQ Test Patterns

- A subset of the functional or structural test patterns is selected; testing is done at normal speed except for slowing down to allow IDDQ measurement for selected test vectors.
- Fault models must be used to make the process of IDDQ test vector selection feasible and appropriate software must be developed.
- Mao et al (ICCAD 90)
  - The QUIETEST system was implemented to select test patterns to cover gate oxide shorts and transistor stuck-on faults by IDDQ measurements.
- Vandris et al (ITC 91)
  - A mixed functional/IDDQ testing methodology for CMOS transistor faults.
  - IDDQ measurements are employed to cover transistor stuck-on faults that cannot be caught by voltage testing.
The fault model includes:
- Leakage faults (intended to cover gate oxide shorts and transistor stuck-on faults)
- Weak faults (intended to cover transistor stuck-open faults which cause weakened logic voltage at a node)

It is assumed that the modeled faults do not change the logic values at circuit nodes although they may affect logic levels.

A hierarchical logic simulation technique is employed for selecting IDDQ test vectors; no fault simulation is necessary.

The number of selected IDDQ test vectors is often less than 1% of the total number of provided logic test vectors, yet they give the same coverage against modeled faults as would Every Vector IDDQ testing.
QUIETEST Leakage Fault Model

- The system considers six leakage faults for each transistor; they are modeled as permanent, resistive shorts.
- $F_{gs}$: the leakage fault between gate and source
- $F_{gd}$: the leakage fault between gate and drain
- $F_{sd}$: the leakage fault between source and drain (stuck-on)
- $F_{bs}$: the leakage fault between bulk and source
- $F_{bd}$: the leakage fault between bulk and drain
- $F_{bg}$: the leakage fault between bulk and gate
Leakage Fault Detection

- IDDQ measurement at the circuit state shown will catch the Gate-Source leakage fault in N2.
- IDDQ measurement leads to detection of a leakage fault only if the two nodes across the fault are at opposite logic levels with driving strengths.
- Fault sensitization and fault propagation are both implicit in the direction condition.
Selecting IDDQ Test Patterns

- A straightforward procedure could be
  - Use logic simulation to capture the logic values at the terminals of each transistor upon the application of every test vector
  - Determine which leakage faults would be caught by IDDQ measurement at each test vector
  - Select a small subset of test vectors to cover as many leakage faults as possible

- Practical problems
  - The models used for logic simulation may not all be at transistor level
  - Transistor level simulation of VLSI circuits may be beyond the capacity (memory and speed) of logic simulators
  - Even if transistor level simulation is feasible, the amount of data required to be captured and analyzed will be beyond reasonable limits
QUIETEST Solution

- QUIETEST employs a hierarchical a leakage fault analysis methodology
  - Each logic component type is first characterized by QUIETEST to relate logic levels at its inputs (and state nodes for sequential circuits) to the detection of leakage faults at transistor level
  - QUIETEST then uses gate level logic simulation to capture only the inputs and state nodes of logic components at each vector and uses the characterization data to infer the detection of leakage faults
Logic Component Characterization

- Each logic component type is individually logic simulated at transistor level using exhaustive stimulus

Example

- Time I1 I2 O1 A
- 99 0 0 1 Z
- 199 0 1 1 0
- 299 1 0 1 1
- 399 1 1 0 0
Leakage fault table is generated based upon analysis of logic simulation results

Example

<table>
<thead>
<tr>
<th>N1</th>
<th>N2</th>
<th>P1</th>
<th>P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
</tbody>
</table>

Example:
- 12 GDSSGG GDSSGG GDSSGG GDSSGG
- BBBDDS BBBDDS BBBDDS BBBDDS
- 00 uduudd uuuuuu duuudd duuudd
- 01 uduudd duuudd duuudd uuuuuu
- 10 ddduuu uduudd uuuuuu duuudd
- 11 duuudd duuudd uduudd uduudd

Only 3 patterns are required

(0,1), (1,0), (1,1)
### IDDQ Test Subset Selection

#### Gate level logic simulation results for an example circuit

<table>
<thead>
<tr>
<th>Time</th>
<th>I1</th>
<th>I2</th>
<th>X1</th>
<th>O1</th>
<th>O2</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>199</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>299</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>399</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>499</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>599</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>699</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>799</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>899</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>999</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1099</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1199</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1299</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1399</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Test Subset Selection Criteria

IDDQ test subset for the example circuit

<table>
<thead>
<tr>
<th>Time</th>
<th>I1</th>
<th>I2</th>
<th>X1</th>
<th>O1</th>
<th>O2</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>199</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>299</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>399</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>499</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>599</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>699</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>799</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>899</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>999</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1099</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1199</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1299</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1399</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

QUIETEST is able to select 5 of the 14 vectors without loss of coverage
Requirements

- Layout vs schematic checks should not allow swapping of input terminals or order of transistors in stacks; this can invalidate vector selection by QUIETEST

Example
Impressive results have been obtained for IDDQ test pattern selection for two industrial VLSI circuits.

In each case QUIETEST selected less than 1% of the functional test vectors for IDDQ testing which would still provide as much leakage fault coverage as would be achieved by IDDQ measurements on 100% of the vectors.

<table>
<thead>
<tr>
<th>Circuit</th>
<th># of Transistors</th>
<th># of Functional Test Vectors (Full Test Set)</th>
<th>Leakage FC (Full Test Set)</th>
<th># of Selected IDDQ Test Vectors (Selected Test Set)</th>
<th>Leakage FC (Selected Test Set)</th>
<th>Size of Selected Test Set (Percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit 1</td>
<td>7584</td>
<td>74880</td>
<td>94.82%</td>
<td>370</td>
<td>94.82%</td>
<td>0.49%</td>
</tr>
<tr>
<td>Circuit 2</td>
<td>40948</td>
<td>310294</td>
<td>84.80%</td>
<td>1709</td>
<td>84.80%</td>
<td>0.55%</td>
</tr>
</tbody>
</table>
While NMOS transistors can successfully pass logic 0, they degrade the voltage level for logic 1; the opposite is true for PMOS transistors.

Example and circuit simulation results:
- **defect**: VI1 VA VO1
- **P1 open**: 5.0V 3.6V 0.1V
- **N1 open**: 0.0V 1.5V 4.8V

Although the logic levels are significantly degraded, the logic values are still interpreted correctly by the driven stages. Such defects are often incorrectly classified as undetectable although in actual circuits they can cause malfunction due to increased propagation delays, decreased noise immunity, etc.
QUIETEST Weak Fault Model

- If a node has normal logic 0(1) voltage in a defect free circuit and a weakened logic 0(1) voltage in the defective circuit, the effect of the defect is modeled as a weak 0(1) fault at the node.
- In the example circuit, if transistor P1(N1) is stuck-open, node A will have a weak 1(0) fault.
  - Degraded voltage levels due to tristated driving stages are not considered.
  - Individual weak faults due to single defective transistor are targeted.
Example

Not all nodes can have weak faults; on the other hand some nodes may have the possibility of wither weak-0 or weak-1 faults
Some nodes may have the possibility of only weak-0(1) faults
Some nodes may have the possibility of more than one weak-0(1) faults
QUIETEST does not allow the weak-q fault at node A when transistors P2 and P3 are both stuck-open
Detection of Weak Faults

- Example and circuit simulation results
- If either transistor P1 and N1 is stuck open, the fault can be detected if IDDQ is measured for test vectors that cause logic 0 and logic 1 to be applied to the transmission gate input.

<table>
<thead>
<tr>
<th>defect</th>
<th>VI1</th>
<th>VA</th>
<th>IDDQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 open</td>
<td>5.0V</td>
<td>3.6V</td>
<td>56uA</td>
</tr>
<tr>
<td>N1 open</td>
<td>0.0V</td>
<td>1.5V</td>
<td>94uA</td>
</tr>
</tbody>
</table>
Detection of Weak Faults

Example circuit

For transistor P1 or N1 stuck open, the fault detection cannot be guaranteed by measuring IDDQ for vectors that cause logic 0 and logic 1 applied to the transmission gate input.

Unlike leakage faults, sensitization of weak faults and propagation of weak fault effects must be separately addressed.
A weak fault is sensitized by a test vector if it establishes a degraded logic level due to transmission of logic 0(1) through a PMOS(NMOS) transistor alone.

The fault effect must be propagated by establishing logic values that complete a conducting path from Vdd to Vss through the partially turned on complementary transistors due to the weakened logic level.
QUIETEST Weak Fault Tables

- While the sensitization of a weak fault exclusively depends upon the I/O states of the logic component with the weak fault, the propagation of the fault effect may depend upon:
  - I/O states of the component with the weak fault
  - I/O states of the driven component
  - Both of the above

- Up to three weak fault tables may be necessary for each logic component:
  - Weak fault detection table
  - Weak fault sensitization table
  - Weak fault propagation table
A logic component, its exhaustive logic simulation results and weak fault detection table for weak-0(1) faults at node A

<table>
<thead>
<tr>
<th>I1</th>
<th>I2</th>
<th>O1</th>
<th>F0</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>d</td>
<td>u</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>u</td>
<td>d</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>d</td>
<td>u</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>u</td>
<td>d</td>
</tr>
</tbody>
</table>
Good results have been obtained for IDDQ test pattern selection for two VLSI circuits.

<table>
<thead>
<tr>
<th>Circuit</th>
<th># of Weak Faults</th>
<th># of Test Vectors (Full Test Set)</th>
<th>Weak FC (Full Test Set)</th>
<th># of Selected IDDQ Test Vectors (Selected Test Set)</th>
<th>Weak FC (Selected Test Set)</th>
<th>Size of Selected Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit 1</td>
<td>1923</td>
<td>74880</td>
<td>85.3%</td>
<td>263</td>
<td>85.3%</td>
<td>0.35%</td>
</tr>
<tr>
<td>Circuit 2</td>
<td>1497</td>
<td>113856</td>
<td>86.7%</td>
<td>397</td>
<td>86.7%</td>
<td>0.35%</td>
</tr>
</tbody>
</table>

In each case QUIETEST selected less than 1% of production test vectors for IDDQ testing which would still provide as much weak fault coverage as would be achieved by IDDQ measurements on 100% of the vectors.

For circuit 1, 80.2% of the weak fault IDDQ vectors were common with leakage fault IDDQ vectors; for Circuit 2, 95% of the vectors were common.
Selective IDDQ Test Patterns

- A subset of the functional or structural test pattern set is selected for IDDQ testing of modeled faults

Advantages
- Small number of test vectors (often under 1%) can be selected without loss of coverage; methodology is practical for large volume production IDDQ testing
- Fault coverage is quantified
- Effective for both combinational and sequential circuits
- Circuits can be tested at system speed except for selected vectors
- Dynamic circuits can be accommodated

Disadvantages
- Fault model is necessary; coverage shows IDDQ test effectiveness for only those defects that have a representation in the fault model
- Coverage is constrained by the quality of functional or structural logic test pattern
- Software for test vector selection is required
Detection of Inter-Gate Bridging

QUIETEST supports IDDQ test pattern selection for bridging faults between user specified node pairs

- Only inputs and outputs of logic components may be included in the bridging fault list consisting of node pairs
- The intent is to provide coverage for bridging faults between long, parallel lines such as bridges between adjacent bit lines in a PLA AND plane
- Ideally the bridging fault list should be extracted form the layout

QUIETEST attempts to select a small number of vectors to drive the nodes in each specified node pair to opposite logic levels with driving strengths.

![Diagram of bridge fault](image)

- 0
- P1
- N1
- Bridge
- P2
- N2
- 1
Detection of Bridges to Vss or Vdd

QUIETEST has been enhanced to select IDDQ test subsets that cover bridges to Vdd/Vss

A bridge to Vdd(Vss) can be detected by measuring IDDQ while attempting to drive the bridged node to logic 0(1)

If the functional or structural test pattern toggles every node, QUIETEST can do IDDQ test subset selection for 100% coverage of bridges to Vdd/Vss

Initial results show that inmost cases the vectors selected to cover bridges to Vdd/Vss are a subset of the vectors selected to cover leakage faults
This methodology for obtaining Selective IDDQ test patterns has been proposed by Vandris et al (ITC 91).

The fault model consists of single stuck-at faults, transistor stuck-on faults and transistor stuck-open faults.

Faults detectable by structural or functional non parametric voltage based testing are simulated using the functional or structural test pattern set.

Transistor stuck-on faults deterministically undetectable by logic testing are considered for detection by IDDQ testing.

Accurate electrical evaluation of the switch level circuit state is employed.

- The number of faults determined to be detectable by logic testing is increased.
- The number of faults requiring IDDQ measurement is decreased.
Fault Preprocessor

- The fault preprocessor generates single stuck-at faults, transistor stuck-on faults and transistor stuck-open faults.
- It then uses a fault collapsing procedure to reduce the number of faults to be simulated to determine logic testing fault coverage.
- An N-transistor stuck-on (stuck-open) fault is considered equivalent to a s-a-1(0) fault on its gate input; P-transistor individual gate stuck-at faults are similarly collapsed.
- It is assumed that stuck-on faults in parallel transistors with equal conductances are equivalent, and the stuck-open faults in series transistors are equivalent.
- The fault preprocessor uses transistor group dominance to determine the detectability of a class of transistor stuck-on faults by logic testing.
Transistor Group Dominance

- Every transistor group is modeled as a pull-up network and a pull-down network of N-transistors
- A dominance attribute is statistically assigned to every transistor group based on an electrical calculation of the network conductances that take into the account transistor device characteristics and interconnections
- A group has
  - N-dominance if its pull-down network overpowers the pull-up network when both are on
  - P-dominance if its pull-up network overpowers the pull-down network when both are on
  - No dominance if neither network overpowers the other when both are on
- When a group is non-dominant, a dynamic dominance evaluation is performed during fault simulation based on the instantaneous input values
Dominance allows accurate determination of the detectability of a transistor stuck-on fault by logic testing and is used to deduce the equivalence of a class of a transistor stuck-on faults to node stuck-at faults.

The strength dominance fault collapsing rules.

IDDQ monitoring need only be used for covering transistor stuck-on faults which are either undetectable or only potentially detectable by logic testing.
Consider a static CMOS inverter

- A: input, B: output
- There 8 faults: P1 stuck-on, P1 stuck-open, N1 stuck-on, N1 stuck-on, A s-a-1, A s-a-0, B s-a-1, B s-a-0
- Separate faults on the branches of A need not be modeled
- Using the gate level fault collapsing A s-a-1(0) is equivalent to B s-a-0(1)

If the inverter is P-dominant, P1 stuck-on is equivalent to B s-a-1 and N1 stuck-on is undetectable by logic testing

If the inverter is N-dominant, N1 stuck-on is equivalent to B s-a-0 and P1 stuck-on is undetectable by logic testing

If the inverter is non-dominant, both P1 and N1 stuck-on faults are only potentially detectable by logic testing
Selection of IDDQ Test Vectors

- Transistor stuck-on faults that are undetectable or only potentially detectable by logic testing are considered for detection by IDDQ monitoring
  - Upon the application of a test vector, the fault simulator determines if simultaneous pull-up and pull-down transistor paths are created due to the fault; the first vector to do so is selected for IDDQ testing
- IDDQ test vector selection is done during fault simulation
- Simulation results for an ALU, accumulator and multiplier show encouraging results
  - Test sets generated for logic testing based upon traditional single stuck-at faults were used
  - Transistor fault coverage was found to be considerably less than the single stuck-at fault coverage when only logic testing was employed
  - Addition of IDDQ testing consistently improved the transistor fault coverage by as much as 25%
Supplemental IDDQ Test Patterns

- Test vectors are generated specifically for IDDQ testing
  - logic testing is done at full system speed and is supplemented by IDDQ testing for the generated pattern
- Fault models must be used to make the process of IDDQ test vector generation feasible and appropriate software must be developed
- Bollinger et al (ITC 91) have developed a methodology that supports IDDQ test generation for bridging faults
- Chen et al (ITC 91) have proposed an approach for generating tests for switch level circuits using both IDDQ and logic test generation; node stuck-at, transistor stuck-open and transistor stuck-on faults are considered
Test Generation of Bridging Faults

- Based on traditional ATPG tool
  - Excite error signal at assumed fault site
  - Propagate error signal to the output of component, not to PO
  - Modify fault simulation
- Conventional stuck-at fault can also be detected
  - Can detect more stuck-at faults than logic testing
  - Easier to generate test patterns than logic testing
- Test vector compaction is very important
  - To reduce test time
Test Generation of Bridging Faults

- Bollinger et al (ITC 91) have proposed an IDDQ test generation methodology for CMOS bridging faults.
- A switch-level circuit model is employed to allow for direct representation of both intra-gate and inter-gate unrestricted bridging faults.
- A graph based approach is used for accurate representation of switch level circuits that corresponds directly to circuit topology and the ability to easily represent bridging faults as graph perturbations.
- Computational overhead of switch level analysis is counteracted by the formulation of a modular, hierarchical approach.
- Only combinational static CMOS logic is considered.
A hierarchical circuit model is used to conserve memory resources and to restore the computational complexity and processing time.

The circuit is described as an interconnection of primitive modules which correspond to structures such as simple and complex CMOS gates.

For every primitive module type one switch level structural description is maintained and every fault-free instance can map into it.

Each primitive module instantiation requires only enough storage for the instance connectivity and the set of external nodes.

Gate level descriptions are also attached to primitive modules; they are used in efficient implementation of the backward line justification and forward implication routine.
Test Generation

- **Sensitization condition**
  - The nodes with a bridging fault should be driven to opposite logic levels

- **Switch-level path tracing**
  - A depth first search method is used to determine faulty modules’ input logic values to create a path from Vdd to Vss through the bridge

- **Backward implication**
  - The input logic values at faulty modules is not available, switch level routine is used

- **Topological ordering**
  - The distance of primitive modules from primary inputs is used as a heuristic during backtracing

- The algorithm has been used to generate tests for randomly selected bridging faults in combinational ISCAS circuits
Supplemental IDDQ Test Patterns

- Test vectors are generated specifically for IDDQ testing of modeled faults

**Advantages**
- Small number of effective test vectors can be generated to supplement logic testing
- May also be employed to supplement Selective IDDQ test patterns
- Fault coverage is quantified
- Coverage is independent of the quality of logic test pattern
- Logic testing can be done at system speed

**Disadvantages**
- Fault model is necessary; coverage shows effectiveness for only those defects that have a representation in the fault model
- Practicality of test generation methods for large sequential circuits is yet to be established
- Software for test vector generation is required
Summary

- An IDDQ test vector must create one or more low resistance paths from Vdd to Vss in the presence of target defects.
- There are three approaches to obtaining IDDQ test patterns: Every Vector, Selective, and Supplemental.
- So far the most practical approach for industrial use seems to be Selective IDDQ testing.
  - Has been shown to be effective for several fault types.
  - Has been successfully employed for 40,000 transistors sequential circuits.
  - Production IDDQ test time seems to be economical.
- As more effective supplemental IDDQ test generation approaches for sequential circuits are developed, Selective IDDQ testing may be replaced or augmented with Supplemental IDDQ vectors.
The diode resistor method has been replaced with other techniques for new ICs.
The FET-resistor method was used for several years but its limit is around 10 uA.
An OP amp circuit has been evaluated and also has a limit around 10-15 uA.
The bit-current option has a sensitivity to about 0.5uA; its measurement rate at 1 uA is about 10-15 KHz.
The Keating circuit is sensitive (pA-uA range), but as with all circuits runs slow for low current measurements; at 1 uA, the measurement rate may be about 5 KHz.
Off-Chip Measurement

Instrumentation

- DUT
- \( V_{\text{DD}} \)
- \( I_{\text{DD}} \)
- \( V_O \)
- \( I_{\text{DDQ}} \)
- \( R_m \)
- \( 5V \)

No Defect

Defect

\( I_{\text{DDQ}} \)

\( V_O \)

\( i_{\text{DD}} \)

\( I_{\text{DDQ}} \)

\( \text{TIME} \)
Diode Circuit

- Current sensing with Diode

![Diode Circuit Diagram]

- Circuit Under Test
- Voltmeter
- To DC
- To power supply
Current sensing with NMOS
FET circuit

Overcome Noise Problem

Circuit Under Test

To DC voltmeter

To power supply

\[ c_1 \]
FET circuit

-speed up

![Circuit diagram]

- Circuit Under Test
- To power supply
- c1
- To sample and hold
The Keating Circuit

A circuit is measured the decay in Vdd if the power supply pin to the DUT was opened

Diagram:
- TESTER PS
- 5V
- VDD
- I_DD
- C_DD 1 nF
- Logic
- DUT

Graph:
- VO
- TIME
- NO DEFECT
- I_DDQ DEFECTS
On-Chip Circuit

- High speed testing
- Detection of small currents
- Use of conventional test equipments
- Easy to test

- Area overhead
- Performance degrade
  - Output voltage
  - Operation speed
- Partition required
CMU On-Chip Circuit

- Issq measurement with sensor virtual ground replacing IC’s DUT ground connection
- Main sensing element is a diode-connected lateral bipolar transistor; get logarithmic response
On-Chip Circuit

- Reduce overhead and performance degrade
On-Chip Circuit

- Measure dynamic and static currents together
- Reduce performance degrade
- Difficult to generate test patterns
OP-Amp On-Chip Circuit

- Negligible performance degrade
- Difficult to make an accurate OP-amp
Summary

- IDDQ instrumentation down to the 10-15 uA range is quite easy to install using the FET/resistor or OP amp circuits.
- IDDQ limits on the order of 1uA are slightly more difficult; the Keating floating Vdd node circuit looks good or can use very sensitive instrumentation with a measurement resistor.
- All instrumentation methods have rate vs sensitivity tradeoffs:
  - at IDDQ = 1uA, ftest = 3-5kHz
  - at IDDQ = 10uA, ftest = 50kHz
- On-chip circuits will probably grow in development; higher rates and more sensitivity are reported.
The 1mA limit setting is very coarse, catching gross leakage defects and CMOS stuck-at faults.

- It is better than nothing but perhaps too much.

Some have used the RMS current of the power supply.

- This provides an indication of power consumption with frequency, but not much about random defects.

The 20-100 uA range is easily obtainable with rapidly assembled IDDQ instrumentation.

- The defect detection is still coarse in this range, but you get many bridges, gate shorts, stuck-opens, etc.
- There are probably many fab houses that put out circuit quality in this range.
1 uA is a compromise limit between good quality, an IDDQ tester rate of about 10-15 KHz, and acceptable yield for a good fab house.

A gate short was observed to have 0.5 uA at a 5V.

0.5 uA strains most present capabilities, but some individual wafer lots have shown this low level.

1 uA is also a reasonable upper limit for battery operated circuits.

1 uA IDDQ measurement instrumentation requires a little care, but is done.

The 50 nA limit for an IC is slightly above what would usually be considered a leakage defect-free circuit.

If you set IDDQ at this level, the yield out of fab will probably go to zero.
Characterization

- Histogram data can be used to evaluate quality vs cost
- You will find that many very leaky circuits pass all functional test vectors at speed
  - Do not be compromised by this observation and the “Who cares. It works.” attitudes
- Histogram data provide a first look at your capabilities
  - You can set levels and begin initial screening at this point or take further data by comparing split lots on an experimental basis
- In some cases leaks have an associated reliability defect and in other cases not
  - Gate shorts are major reliability risks
  - Leakage may be symptomatic of a serious root cause
- You have no economic way to separate out these mechanisms at test
- A dynamic burn in will show many more defects
What is the risk of accepting ICs that pass function, but fail IDDQ limits?

A life test study at Ford Microelectronics observed the failure response of ICs that were grouped into different IDDQ leakage ranges (McEuen, VLSI Test Symp. 91)

The data shows that about 8.3% reliability failures occur for the IDDQ leakage values studied
Summary of Reliability

- SANDIA data specifically show that gate shorts are a significant reliability risk
  - The part may leak, show functionality, but subsequently fail
- The Ford data show that 21-51% of ICs with high IDDQ pre-burn-in failed functional tests after burn-in
- Some leakage mechanisms might not cause reliability failures, but their presence masks the ability to detect those leaky defects that do cause failure
Ideal Design for IDDQ Testability

- Fully CMOS Design
- 100% static
  - Nodes always driven to Vdd/Vss except when switching
  - No minimum clock frequency
- 100% complementary
  - Nodes driven by complementary P/N networks
Non-static Design Examples

- **Nodes that are not latched**
  - Precharge circuits

**Buses with undriven states**

![Diagram showing control signals and internal bus connections](image)
Non-Static Design Alternatives

Testing
- Perform IDDQ test prior to drift of node from Vdd/Vss
- Example: 80C51 microcomputers

Design
- Weak p-channel half latch or bus holder latch
Non-Complementary Design Options

Testing

- Measure IDDQ for input pull-up resistor or IDDQ for input pull-down resistors
- Measure IDDQ(ISSQ) only for those vectors where pull-up(pull-down) nodes are 1(0)

\[ \text{INPUT} \quad 0/1 \]

- PULL-UP \quad I_{\text{SSQ}} = 0

\[ \text{V}_{\text{DD}} \quad I_{\text{DDQ}} \]

\[ \text{V}_{\text{DD}} \quad I_{\text{DDQ}} = 0 \]

\[ 0 \quad 1 \]
Non-Complementary Design Options

Design

- Use latches to hold inputs to 0/1 when undriven
- Use complementary enhancement mode P(N) networks instead of pull-up (pull-down) resistors or depletion mode transistors
Non-Complementary Design Options

Design

Separate power for peripheral and core logic

[Diagram of IC with separate power supplies for inputs, outputs, and core logic]
Non-Complementary Design Options

Mixed Signal ICs

![Diagram of Mixed Signal ICs with labels: V_{DD}, V_{CC}, V_A, CMOS, BIPOLAR, ANALOG]
Suggested Cures

- Remove the pullups
- Use p-channel pass transistor in series with the pullups - control transistor on/off state from TAP controller signal
- Measure ISSQ instead of IDDQ - might work at IC test, but still high current at board test
- 1149.2? No pullups
- Test sequence - load a vector; stop TCK and open up TMS and TDI; take reading (slow)
Other Factors

- Ideally IDDQ for defect-free ICs should be due to junction leakage (fundamental limitation for junction-isolated technologies)
- Avoid design or technology features that cause IDDQ to increase above junction leakage
  - Low threshold voltage, punchthrough, parasitic field oxide transistor subthreshold leakage, isolation structure leakage, etc.
- The leakage due to these can be greater than junction leakage and have a stronger dependence on temperature, voltage, or radiation than junction leakage
Summary

- The ideal design for IDDQ testing is a fully-static, fully-complementary CMOS IC.
- IDDQ testing can be performed on some ICs that are not full CMOS designs.
- For low IDDQ, the CMOS IC design and fabrication technology should be implemented and controlled so that IDDQ is dependent primarily on reverse-biased junction leakage.
Delay Faults

- The collection of all defects that cause a circuit to fail at a frequency less than that of a defect-free circuit
- For very high speed applications, we begin to worry about the statistical spread of what were previously considered normal variations

What does the relatively slow speed IDDQ test have to do with detection of delay faults?

- A large percentage of defects that cause delay faults are sensitive to detection by the IDDQ test
Delay Faults

- A significant trend exists to manufacture electronic systems that are much faster than the typical 5-10 MHz systems of the 1980’s.
- These newer systems run from 20-200 MHz with the 100 MHz workstation leading off the decade of the 1990’s.
- We will look at a category of defect that causes timing errors, but does not elevate IDDQ.
- Two reasons why we devote delay faults:
  - It is very difficult to guarantee the coverage of timing faults for circuits of any complexity intractability situation.
  - Testers themselves cannot run at these component speeds because of fixturing problems and the inability to run as fast as the products that they are testing.
Why Difficult to Detect?

- Timing defects lie on signal paths; the total number and length of all signal paths is very large.
- Many CMOS timing defects do not alter the signal symmetrically with respect to rise and fall time.
- This last possibility really blows up the defect search since we must now examine both rise and fall times instead of just sending a single transition signal down a path and measuring the delay.
IDDQ testing will eliminate many delay defects.

However there is a subset of delay defects that can only be detected by at-speed testing.

- Elevated series of resistance in interconnect lines such as can often occur in vias
- Altered transistor thresholds, elevated interconnect capacitance are other such defects
- Any defect that slightly weakens logic threshold voltages but not enough to turn on normally-off transistors

Delay faults represent the most difficult situation to detect at any level of assembly

- Most of the defects that we have studied can be 100% detected with new test methods
- Delay faults will grow in concern as clock periods shrink and circuit complexity continues to grow

At-speed detection of high speed timing faults will most likely have to be done without the aid of testers
Defect Diagnosis

- The objective is to locate defects in VLSI circuits
- Defect diagnosis is used
  - To help improve yields when a new design or process is introduced
  - During high volume production as a process monitor
  - Later in life cycle to identify weak points by analysis of failed chips
- This presentation deals with location of resistive shorts in CMOS circuits
Resistive Short Location

- Common techniques are hot-spot detection, waveform proving and emission microscopy
- IDDQ monitoring is an effective alternative for location of resistive short in CMOS circuits
Diagnosis by Every Vector

An every vector IDDQ test pattern

<table>
<thead>
<tr>
<th>test vector</th>
<th>detected faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>f1, f3, f4, f5</td>
</tr>
<tr>
<td>v2</td>
<td>f2, f4, f5</td>
</tr>
<tr>
<td>v3</td>
<td>f2, f5</td>
</tr>
<tr>
<td>v4</td>
<td>f1, f2, f3</td>
</tr>
<tr>
<td>v5</td>
<td>f1, f3, f5</td>
</tr>
</tbody>
</table>

Under the assumption of a single fault

- If v1 and v2 exceed IDDQ limit, the fault can be diagnosed to be f4
- If v1, v4 and v5 exceed IDDQ limit, the fault is one of f1 and f3

An implementation was effective in locating transistor terminal to terminal shorts and bridges between unrelated nodes; it was useful in isolating a reduced list of possible short locations to speed physical analysis
Diagnosis by Selective

- **A selective IDDQ test pattern**
  - Choose v1, v2 and v5

- **Under the assumption of a single fault**
  - If v2 exceeds IDDQ limit, the fault can be diagnosed to be f2
  - If v1 and v2 exceed IDDQ limit, fault is one of f4 and f5

- **The QUIETEST system includes an implementation for diagnosis of the six modeled resistive shorts per transistor; the diagnosis tool has been found effective in particular situations**
Aitken (ITC 91) has proposed that since IDDQ monitoring alone provides only one observation point, it may be advantageous to couple it with output logic observations for more effective diagnosis.

- For every transistor, 6 resistive shorts are considered.
- Supplemental IDDQ test patterns are used.
- During simulation, the output of a logic gate is assigned X if it is determined that a high IDQ state would occur due to a resistive short in a transistor feeding the output.
- Single or multiple faults are allowed; diagnosis is performed by comparing three value simulation results and the observed IDDQ and logic values.
- The method is effective for combinational or full scan sequential circuits.
DIVA Diagnosis Algorithm

- Falling vectors (that exceed IDDQ limit) are simulated for each defect in the defect list.
- Simulation results are compared with observed values and the defect list is divided into numbered groups:
  1: defects whose predicted output values match all observed output values.
  2: defects whose predicted output values match the observed output values on all outputs which can be reached from the defect site.
  3: Defects whose predicted failures match observed output failures on all outputs which can be reached from the defect site.
  4: Defects whose predicted failures match some observed output failures on outputs which can be reached from the defect site.
  5: Defects whose predicted failures do not match any observed output failures.
- Defects in group 1 are most likely to be the cause of observed failures while defects in group 5 cannot alone be the cause; defects in remaining groups may be the cause with decreasing likelihood.
Summary

- IDDQ monitoring is an effective alternative for location of resistive shorts in CMOS circuits
- Every Vector, Selective, and Supplemental IDDQ test patterns can all be used for diagnosis
  - Selective test patterns are usually worse than Every Vector test pattern in their diagnosis capability
  - Supplemental test patterns can be better or worse in their diagnosis effectiveness compared to the other types of IDDQ test patterns
- When Selective IDDQ testing is used in production, Every Vector IDDQ testing may still be employed for better fault diagnosis on falling chips
- Combined use of logic observations and IDDQ monitoring may improve diagnosis capability
Conclusion

- Most CMOS IC defects elevate IDDQ several orders of magnitude greater than non-defective circuit
- IDDQ Testing is the most sensitive way to detect the majority of CMOS defects
  - Direct Observability
  - Greatly increased detection of common physical defects
  - 100% coverage of stuck-at faults for many designs
  - Reduced test vector count
  - Simplified test vector generation and fault simulation
  - No additional on-chip circuitry required
- Need research on current sensing and test set compaction