DESIGN FOR TESTABILITY

Generating tests for large circuits is very time consuming. One way to get around this problem is to constrain or modify the design to make test generation easier.

Most DFT techniques are targeted to sequential circuits where test generation in general is a difficult problem.

If testing is not considered during the design phase very low fault coverages can result in addition to high test generation times ⇒ DFT saves money + time!

DFT techniques can be divided into:
- ad hoc techniques
- structured design techniques
- self-test + build-in testing
The objective of DFT is to improve the controllability and observability of internal circuit nodes so that the circuit can be tested effectively:

**Controllability:**

The ability to set or reset internal nodes from the primary inputs

**Observability:**

The ability to observe the value of an internal node at the primary outputs

These are two good measures of the testability of a circuit, i.e. how easy it is to test. Design for testability attempts to improve circuit testability by making the internal nodes more controllable and observable.
AD-HOC DFT TECHNIQUES

(I) Test Points

(a) ORIGINAL CIRCUIT

(b) "JUMPER": EXTERNAL TEST EQUIPMENT CAN OBSERVE AND/OR INJECT LOGIC VALUES ON AIA'.

(c) Φ-CONTROLLABILITY PLUS OBSERVABILITY

(d) Φ-1-CONTROLLABILITY

(e) 0-L-CONTROLLABILITY VIA THE USE OF A MULTIPLEXER
Using test points require a significant number of I/O pins. To alleviate the problem we use the following trade-offs:

Multiplexing observation points

Controllability via a demultiplexer

Time sharing I/O ports
(II) Initialization

Design circuits so that it is easy to initialize them.

(III) Counter and Shift Register Partitioning

\[ R = R_1 + R_2 \]

Without testability feature...

... with testability feature.

Savings:

16-bit counter needs \( 2^{16} = 65536 \) clock cycles.

"Partitioned" in 2 8-bit counters it needs \( 2 \cdot 2^8 = 512 \) clock cycles.
(IV) Partitioning of large combinational structures

INSERT MULTIPLEXERS BETWEEN THE LARGE COMBINATIONAL STRUCTURE \( C = C_1 + C_2 \):

- \( T = 00 \) normal operation
- \( T = 01 \) tests \( C_1 \) as inputs to \( C_1 \) are \( A \) and \( C' \) and it outputs to \( F \) \( G' \)
- \( T = 10 \) tests \( C_2 \)
Control points usually are:
- control, address, data buses
- enable/hold to microprocessors & memories
- data select to MUXes
- clock, presets to FFs, counters, shift registers

Observation points usually are:
- stem lines with high Fanout
- redundant signal lines
- outputs of MUXes
- FFs, counters, shift registers
- address, control, data buses
DFT WITH SCAN REGISTERS

TEST POINTS ARE EXPENSIVE IN TERMS OF I/O PINS. SCAN REGISTER (MADE WITH SCAN CELLS) IS A REGISTER WITH BOTH SHIFT AND PARALLEL-LOAD ABILITY. ITS SCAN CELLS CAN BE USED AS OBSERVATION AND/OR CONTROL POINTS.

TRADE OFF: Saves I/O pins increases test time and area overhead

SCAN CELL

\[ \overline{N/I}T = 0 \text{ loads from } D \]
\[ \overline{N/I}T = 1 \text{ loads from } S_i \]

SCAN REGISTER

\[ \overline{N/I}T = 1 \text{ shifts from } S \]
\[ \overline{N/I}T = 0 \text{ loads in parallel} \]

SCAN IN: loading from \( S_{in} \)
SCAN OUT: reading from \( S_{out} \)
Simultaneous Controllability and Observability

Original Circuit

Two variants of simultaneous controllability and observability (depending on MUX's select line)

Non-Simultaneous Techniques

Observability

Observability through compaction

Controllability
Run PODEM on combinational structure assuming $Y$ and $E$ are pseudo inputs/outputs. You get a sequence of:

$$(x_1, y_1) \ (x_2, y_2) \ \ldots \ \text{test vectors} \ \text{responses}$$

$$(2_1, e_1) \ (2_2, e_2) \ \ldots$$

To test the device:

Scan in $y_1$ and at $k^\text{th}$ clock cycle apply $x_1$.

After clock cycle load $e_1$ in $R_S$ and observe $2_1$, etc.
ISOLATED SERIAL SCAN

SCAN REGISTER IS NOT IN DATA PATH AS BEFORE (see figure below)

IF $|R_s| = |R|$ THEN WE HAVE FULL ISOLATED SCAN:
- accommodates on-line testing vs. full serial
- hardware overhead

NON-SERIAL SCAN

Improves scan-in/scan-out time but increases hardware overhead
SCAN CELL DESIGNS

Most of the times it is useful to separate system clock from scan clock:

![Diagram of MUX and MD-SRL circuits]

To avoid performance degradation due to MUX the following 2-port shift register latch was designed by IBM:

![Diagram of two-port latch]

**Note:** only C-B or A-B clocks can be enabled at any time!
Level Sensitive Scan Design

The following double-catch design uses the design cell developed by IBM and shown before.

There other LSSD schemes with less gates, more clocking etc. Final decision is based upon the # of added combinational logic to the circuit (trade-off).
A sequential circuit is level sensitive if its steady state response to any input state change is:
- independent of its inertial delays
- independent of the order which inputs change

IBM's discipline achieves this with LSSD.
Design rules:
- All internal latch storage must be implemented with hazard-free latches
- The two latches in the SLR must be controlled by two non-overlapping clocks
- All (respective) clocks for the same latch in the SLR must originate from same clock source

Advantages:
Design is race-free, hazard immune, and LSSD is not intrusive on design process

Disadvantages:
Latches are complex, overhead 4-20% in chip area, introduces 4 extra I/O pins and all timing is controlled by external clocks
SYSTEM LEVEL TESTING

TEST AND MAINTENANCE PROCESSOR

A HIERARCHICAL DFT APPROACH
MULTIPLE TEST SESSIONS

TESTING IN OVERLAPPED MODE:

Test them as partly as one block of logic and partly as separate blocks.

- Test C₁ and C₂ with 20 patterns 12 bits wide - **12 × 20 clock cycles**
- Test C₁ with remaining 80 patterns each 8 bits wide - **80 × 8 clock cycles**

Total: 640 clock cycles
MULTIPLE TEST SESSIONS

![Diagram of test sessions]

Testing in "Together Mode":

- $C_1$ requires 100 patterns
- $C_2$ requires 20 patterns

Total: $100 \times 12$ clock cycles

We ignore time to load $R_2$ and $R_4$ as well as scanning out final result.
MULTIPLE TEST SESSIONS

TESTING IN "SEPARATE MODE":

While $C_1$ is tested, $C_2/R_3/R_4$ are ignored (and vice versa)

Testing $C_1$: $8 \times 100$ clock cycles

Testing $C_2$:
redirect $R_4$ to a PO and pad input test patterns with 4 don't cares
$8 \times 20$ clock cycles

Total: 860 clock cycles vs. 1200 before
CASE STUDY: BALLAST PARTIAL SCAN

CLOUD: A MAXIMAL REGION OF COMBINATIONAL (NON-EMPTY) LOGIC

\[\text{BALANCEDSEQUENTIALCIRCUIT (or B-STRUCTURE):}
\]
\[\text{WHEN FOR ANY TWO CLOUDS IN THE CIRCUIT}
\]
\[\text{ALL SIGNAL PATHS GO THROUGH THE SAME NUMBER}
\]
\[\text{OF REGISTERS.}
\]

\[\Rightarrow \text{Circuit above is B-STRUCTURE below they're not}
\]
**PROPOSITION**

Any sequential circuit can be a B-STRUCTURE by replacing a set of appropriate registers into scan registers (NP-complete problem)

**DEFINITIONS**

- $C_B$ is the combinational equivalent of B-STRUCTURE $S_B$ when FFs are replaced by wires
- Largest # of registers on any path $S_B$ between two clouds is depth of $S_B$

Let $t_1, t_2, ..., t_n$ vectors testing $C_B$, where (intuitively): $t_i = t_i^{PI} + t_i^{PPI} \rightarrow$ scan inputs

**ALGORITHM: Testing $S_B$ (depth $d$)**

Scan $t_i^{PPI}$ and apply $t_i^{PI}$

Hold values above and clock circuit $d$ times

Place scan path in normal mode and clock it once

Observe POs

Repeat w/ next vector
(a) original circuit
(b) R3/R6 selected as scan registers
(c) replacing scan registers with PIs/POs
(d) CB (depth = 2)

Clock cycle 1: results at R1/R2
Clock cycle 2: results at R5/R4
Clock cycle 3: results at POs