Testing and its Current Practice

- To meet user’s quality requirements.
- Testing aims at the detection of physical faults (production errors/defects and physical failures).

Automatic Test Equipment
Definition

- Testing means to examine a product to ensure that it functions and exhibits the properties and capabilities it was designed to possess.

- Testing is an important activity in the life-cycle of an electronics product.

The Life-Cycle of an Electronics System
Outline

- The test problems
- Fault modeling
- Design for testability
- High-level synthesis for testability
- BIST

Main Difficulties in Testing

- Miniaturization -> Physical access difficult or impossible.
- Increasing complexity -> Large amount of test data.
- No. of I/O ports remains constant -> Long test application time.
- High speed -> High demand on tester’s driver/sensor mechanism and more complicated failure mechanism.
- Power consumption -> heating problem.
- Testing accounts up to 50% of product development efforts.
Composition of Costs of Testing

■ Cost of test equipment (hardware):
  – A test controller (usually a computer).
  – Interface drivers and receivers and cable-connections.
  – System of probe-contacts.
  – A controlled environment.

■ Cost of software supports:
  – Test pattern generation programs.
  – Test-design verification procedures (fault simulation and analysis).

■ Testing time
  – Test development time.
  – Test application time.

Types of Testing

■ Production test — testing of individual product to check whether faults are introduced during the manufacturing phase. It is assumed that the design is correct.

■ System test — testing of the product in the environment where it is operating to ensure that it works correctly when interconnected with other components.

■ Operation and maintenance test — testing of the product in the field for diagnosis or "preventive" purpose.

■ Prototype test — testing to check for design faults during the system development phase. Diagnosis is required.
Causes of Incorrect Function of an Electronic System

- Design errors — usually consistent
- Fabrication (manufacturing) errors
  - often consistent, e.g., wrong components
  - usually operator errors
- Fabrication (manufacturing) defects
  - inconsistent, e.g., impurity of materials
- Physical failures
  - wear-out
  - environmental factors

The Stuck-At Fault Model

- It is assumed that every fault in a circuit changes its functionality as if some nodes in the circuit were steadily tied to either logic 0 or logic 1.

A single stuck-at model is usually used. That is, we assume that only one node at a time is tied to 0 or 1.
The Single Stack-At (SSA) Fault Model

- Number of possible faults is small (2N).
- Capable of representing many different physical faults.
- Tests produced for SSA faults also detect many other faults.
- Technology independent.
- Facilitate the use of boolean algebra mathematics to be used for deriving test.
- SSA fault coverage remains an established metric for test quality.

Bridging Faults

- Technology dependent.
- Faults are related to the placement of the components.
- Fault activation and propagation are more difficult.
Design for Testability (DFT)

- To take into account the testing aspects during the design process so that more testable designs will be generated.
- Advantages of DFT:
  - Reduce test efforts.
  - Reduce cost for test equipment.
  - Shorten time-to-market.
  - Increase product quality.
- Limitations:
  - Hardware overhead, 5-30%, and performance degradation.
  - Increased design complexity.

Ad Hoc DFT Techniques

- Test point insertion:

  ![Diagram](a) original design  ![Diagram](b) 0/1-injection and observation

  - Very simple.
  - Can be used for both registers and lines.
  - Large overhead of I/O ports (→ multiplexing and addressing)
Ad Hoc DFT Techniques (Cont’d)

- Initialization: design circuits to be easily initializable.
- Clocking: generate clocks off chip or provide a way to control/bypass internal clock.
- Redundancy must be avoided or controlled.

$z = ab + bc + ac = ab + ac$

A Hazard-Free Circuit

Ad Hoc DFT Techniques (Cont’d)

- Sequential depth and feedback: high sequential depth should be avoided.
- Combinational feedback loops should be avoided since such loops result in additional hidden storage elements.
- Partitioning of large combinational circuits.
Structural DFT Techniques

Can be applied to any design in a systematic way:

- Full scan.
- Partial scan.
- Built-in self-test (BIST).

Scan Techniques

- Sequential circuits have poor controllability and poor observability.

If all FFs are included in the scan path (full scan), the problem of ATPG for sequential circuits is reduced to ATPG for combinational ones.
## Partial Scan

- To reduce the number of FFs to be included in the scan chain.
- **Advantages:**
  - reduced area overhead.
  - reduced test application time.
- **Disadvantages:**
  - Design time may increase.
  - Need a sequential ATPG tool.
  - Need a method for selecting the partial scan subset.

## The Basic Elements of HLTS

- Methodology selection — the selection of the overall test strategies, structures, and design constraints.
- Testability-driven synthesis and test structure insertion — the selection and insertion of testability-enhancement structure.
- Design verification — the verification that the inserted test structure and the CUT operate as intended.
The Transformational Approach

- Intermediate representation capturing the design during the whole synthesis process.
- Systematic testability analysis procedure.
- Transformational approach to design.
- Formalization of the whole synthesis problem as a global optimization problem.
- Use efficient optimization heuristics.

Transformational Approach Principle

Traditional Approach

Allocation → Scheduling → Binding → Test insertion

Transformation-Based Approach

Performance
Cost
Power
Testability
...

Optimization
Testability Analysis

- Quantitative testability (controllability/observability) measure reflects:
  - test generation cost,
  - difficulty of achieving high fault coverage, and
  - test application cost.

- An algorithm should accept the design representation and generate the controllability and observability of the basic units, taking into account:
  - the structure of a design,
  - the depth from I/O ports for access, and
  - the characteristics of the functional blocks used.

Testability Analysis Example

```
+-----------------+   Difficult to control
|     S_0          |                          |
|   S_1            |                          |
|   S_2            |                          |
|     S_3          |                          |
|     S_4          |                          |
|     S_5          |                          |
|     C            |                          |

      Difficult to observe
```

```
L_1, S_0
L_2, S_3
L_3, S_3
L_4, S_3
L_5, S_3
L_6, S_3
L_7, S_7
L_8, S_4
L_9, S_3
L_10, S_3

Out
```

```
C_1
C_2
```

In

```
L_11, S_2
L_14, S_2
```

```
’0’
L_1, S_1
L_2, S_3
L_3, S_3
L_4, S_5
L_5, S_7
L_6, S_7
L_7, S_7
L_8, S_4

≥
```

```
+’0’
```

```
S_0 S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8
S_9
```
Testability Improvement Techniques

- Low-impact testability enhancement.
- High-impact testability enhancement.
- Selection of testability improvement techniques is based on:
  - Testability analysis results.
  - Basic test strategy.
  - Test requirement.
  - Global optimization.

Design Transformation Examples

- Merger transformation -> data path allocation.

- Selection of nodes to merge was traditionally based on connectivity/ closeness, resulting in difficult-to-test designs:
  - Nodes with good controllability and bad observability are usually merged together, since they are close to the primary inputs.
  - Nodes with good observability and bad controllability are also merged together.
  - Many loops are generated.
Testability-Guided Data Path Allocation

- Testability analysis results are used to guide the selection of merged nodes.
- The objective is to generate a data path with good controllability and observability for all of the nodes.

**Controllability/Observability-Balance Allocation:**

- Merge nodes with good controllability and bad observability to node with good observability and bad controllability.
- => Low-impact testability enhancement.

Partial Scan Insertion

- The most difficult-to-test register is selected.
- Module binding is done to change it into a scan register.
- The implication in terms of hardware overhead and performance degradation will immediately be checked.
- If the testability requirement still cannot be satisfied, repeat the above process.
- To speed up the algorithm, several registers can be selected at each iteration. They should be in different loops.
**A Simple Example**

![Diagram of a simple example](image)

**Control Part**
- P1, P3, C1, C2, P8, P9, P10

**Data Path**
- #255
- #1
- #0
- P6
- C1
- C2
- P8
- P11

**Built-In Self Test (BIST)**

- **Solution:** Dedicated built-in hardware for embedded test functions.
- No need for expensive ATE.
- At-speed testing.
- Current test possible.
- Support O&M.
- Support field test.

Source: LogicVision
Built-In Self-Test (BIST)

- BIST means that the circuit provides the capability to test itself.

Main BIST Issues to be Considered

- How to exploit existing circuits for BIST purpose so as reduce hardware overhead?
  - For example, a counter can be used as a test pattern generator.
- How to share the same BIST components for different purpose/modules?
- How to test the BIST logic itself.
- How to optimize the BIST design with the rest of the circuits to avoid performance degradation.
Conclusions

- Testability must be taken into account at all stages of the design/synthesis process.
- Early testability consideration prevents costly design iterations.
- Transformation-based approach to HLS for testability is very efficient:
  - BIST is expected to play a prominent role in the future testable design.
  - The key to successful testing lies in the design process!