Design for Testability

Dr. Shubhajit Roy Chowdhury,

Centre for VLSI and Embedded Systems Technology,

IIIT Hyderabad, India

Email: src.vlsi@iiit.ac.in

Books

- “VLSI Test Principles and Architectures: Design for Testability”,Laung-Terng Wang, Cheng-Wen Wu, Xiaoqing Wen
- “VLSI Testing”, Stanley Leonard Hurst
- “Electronic Design Automation”, Laung-Terng Wang, Yao-Wen Chang, Kwang-Ting (Tim) Cheng
- “System-on-Chip Test Architectures: Nanometer Design for Testability”, Laung-Terng Wang, Charles E. Stroud, Nur A. Touba
- “Testing of Digital Systems”, Jha and Gupta

Evaluation Weightages

- Mid Semester Examination 1:20%
- Mid Semester Examination 2:20%
- End Semester Examination: 20%
- Assignments: 15%
- Quiz: 5%
- Project: 20%

Why Testing?

- To determine the presence of fault(s) in a given *circuit*.
  - No amount of testing can guarantee that a circuit (chip, board or system) is fault-free.
  - We carry out testing to increase our confidence in proper working of the circuit.
- Verification is an alternative to testing, used to verify the correctness of a *design*.
  - Simulation-based approach.
  - Formal methods.
- What to test ?
  - System, Board, Chip, Gate, Switch, Interconnection, Logic, Functionality

Verification versus Testing

Verification	Testing
Verifies correctness of design.	Verifies correctness of manufactured h/w.
Performed by simulation, h/w emulation, or formal methods.	Two-part process: 1. Test generation 2. Test application
Performed once prior to manufacturing.	Test application performed on every manufactured device.
Responsible for *quality of design*.	Responsible for *quality of devices*.

Testing

- Testing is one of the most expensive parts of chips
  - Logic verification accounts for > 50% of design effort for many chips
  - Debug time after fabrication has enormous opportunity cost
  - Shipping defective parts can sink a company

- Example: Intel FDIV bug
  - Logic error not caught until > 1M units shipped
  - Recall cost $450M (!!!)

Logic Verification

- Does the chip simulate correctly?
  - Usually done at HDL level
  - Verification engineers write test bench for HDL
    - Can't test all cases
    - Look for corner cases
    - Try to break logic design
- Ex: 32-bit adder
  - Test all combinations of corner cases as inputs:
    - 0, 1, 2, 2^{31}-1, -1, -2^{31}, a few random numbers
- Good tests require ingenuity

Testing Issues

- Testing at higher levels costs higher. For e.g., testing a system is much more costly than testing a chip.
- Test application time can grow exponentially for exhaustive testing of circuits.
  - For a combinational circuit with 50 inputs, we need $2^{50} = 1.1 \times 10^{15}$ test patterns.
- Test generation of sequential circuits are even more difficult.
  - Lack of controllability and observability of flip-flops.
- Functional testing may not be adequate for the detection of physical faults.

Problem: Controllability/Observability

- Combinational Circuits:

  controllable and observable - relatively easy to determine test patterns

- Sequential Circuits: State!

  Turn into combinational circuits or use self-test

- Memory: requires complex patterns

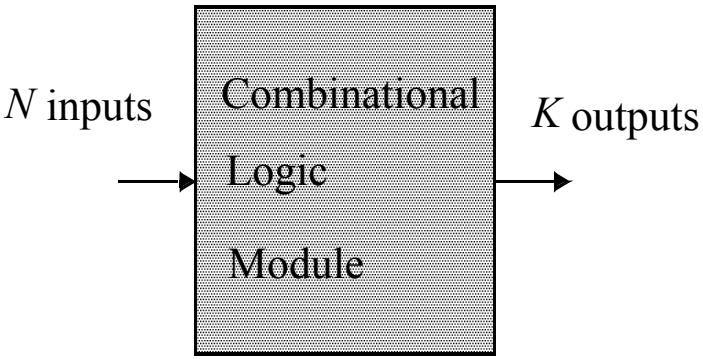
  Use self-test

Observability & Controllability

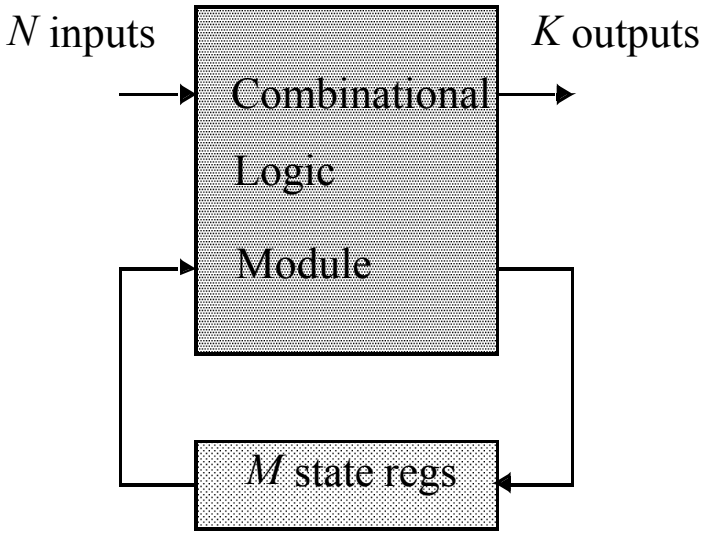
- *Observability*: ease of observing a node by watching external output pins of the chip
- *Controllability*: ease of forcing a node to 0 or 1 by driving input pins of the chip

- Combinational logic is usually easy to observe and control
- Finite state machines can be very difficult, requiring many cycles to enter desired state
  - Especially if state transition diagram is not known to the test engineer

Design for Testability



(a) Combinational function



(b) Sequential engine

2^N patterns **2^{N+M} patterns**

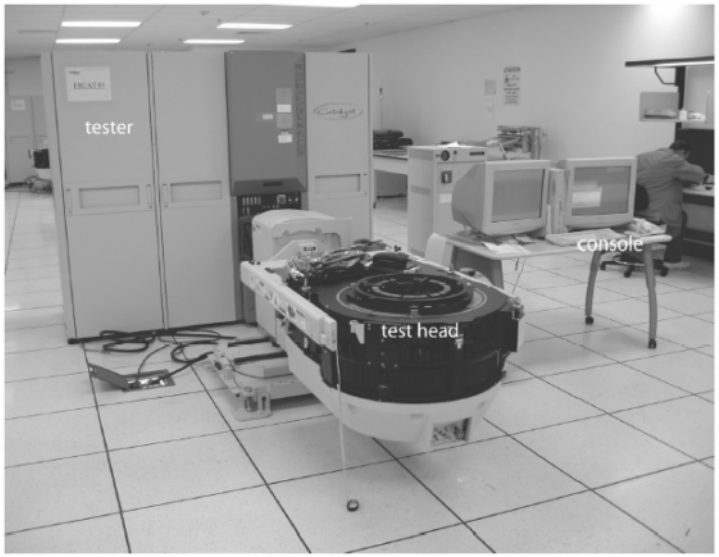
Exhaustive test is impossible or unpractical

Some real defects in a chip

- **Processing Faults**
  - missing contact windows, parasitic transistors,
  - oxide breakdown
- **Material Defects**
  - bulk defects (cracks, crystal imperfections)
  - surface impurities (ion migration)
- **Time-Dependent Failures**
  - dielectric breakdown, electron migration
- **Packaging Failures**

Manufacturing Test

- A speck of dust on a wafer is sufficient to kill chip
- *Yield* of any chip is < 100%
  - Must test chips after manufacturing before delivery to customers to only ship good parts
- Manufacturing testers are very expensive
  - Minimize time on tester
  - Careful selection of *test vectors*



Test Pattern Generation

- Manufacturing test ideally would check every node in the circuit to prove it is not stuck.
- Apply the smallest sequence of test vectors necessary to prove each node is not stuck.

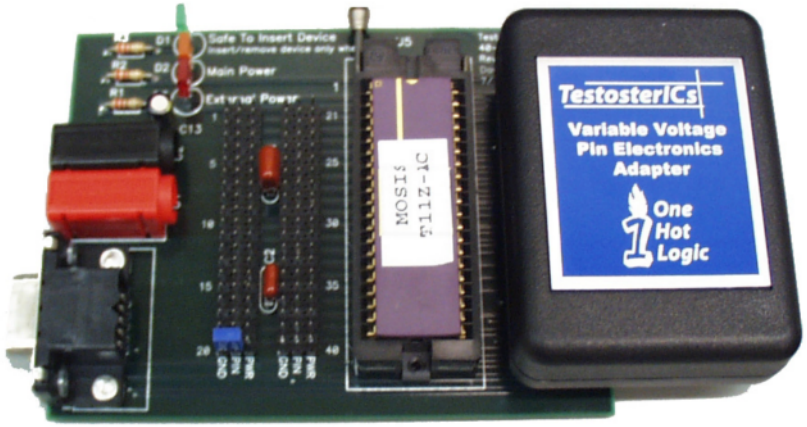
- Good observability and controllability reduces number of test vectors required for manufacturing test.
  - Reduces the cost of testing
  - Motivates design-for-test

Testing Your Chips

- If you don't have a multimillion dollar tester:
  - Build a breadboard with LED's and switches
  - Hook up a logic analyzer and pattern generator
  - Or use a low-cost functional chip tester

TestosterICs

- Ex: TestosterICs functional chip tester
  - Designed by clinic teams and David Diaz at HMC
  - Reads your IRSIM test vectors, applies them to your chip, and reports assertion failures



Fault, Error and Failure

- **Fault**: A physical defect within a circuit or a system
  - May or may not cause a system failure
- **Error**: Manifestation of a fault that results in incorrect circuit (system) outputs or states
  - Caused by faults
- **Failure**: Deviation of a circuit or system from its specified behavior (caused by an error)
  - Fails to do what it should do
- **An Example**: a car with a flat tyre
  - Fault : pin puncture in the tyre
  - Error : Erroneous state of air pressure in the tyre
  - Failure : Car cannot be driven safely

Manifestation of faults

- **Two Types of Faults:**
  - **Permanent: Faults that change the functional behavior of a system permanently.**
    - **Incorrect connections in ICs, PCBs**
    - **Incorrect IC masks**
    - **Functional design errors**

    EASY TO DETECT
  - **Non-permanent: They occur at random times, and affect the system's functional behavior for finite, but unknown periods of time.**

    DETECTION & DIAGNOSIS IS DIFFICULT

Manifestation of faults

- Two classes of *non-permanent faults*:
  - Transient Faults: Caused by environmental conditions such as cosmic rays, α particles, humidity, pressure, vibration, etc.
    - Example: Bit changes in RAMs caused by α-radiation (called *soft errors*; no permanent damage).
  - Intermittent Faults:
    - Caused by non-environmental conditions, such as loose connections, ageing components, critical timing, etc.
    - Behave like permanent failure for the duration of the failure and can be detected by continuously repeating the test.

Fault Coverage and Defect Level

- **Fault Coverage (T):**
  - The measurement of the ability of a set of tests to detect a given class of faults that may occur on the device under test. (determines *test quality*)

$$T = \frac{\text{No. of detected faults}}{\text{No. of possible faults}}$$

- **Defect Level (DL):**
  - The fraction of shipped parts that are defective.

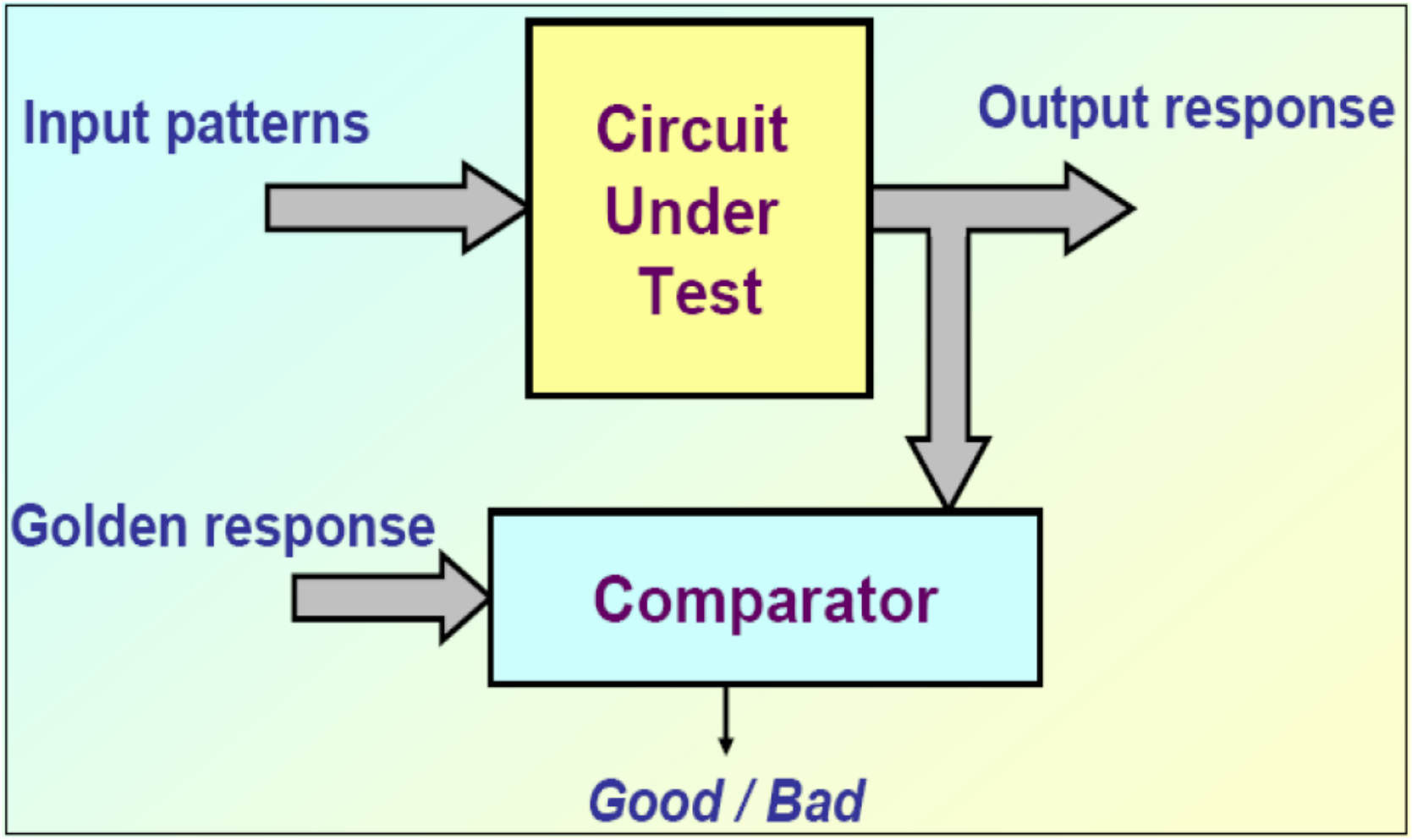
$$DL = 1 - Y^{(1-T)}$$

where Y is the yield, and T is the fault coverage

How to test?

- **Fault Modeling**
  - Identify target faults
  - Limit the scope of test generation
- **Test Generation**
  - Automatic or Manual
- **Fault Simulation**
  - Assess completeness of tests
- **Testability Analysis**
  - Analyze a circuit for potential problem on test generation
- **Design For Testability**
  - Design a circuit for guaranteed test generation
  - Introduce both area overhead and performance degradation

Basic Testing Principle



Questions?