Test Synthesis

Krzysztof Kuchcinski
Dept. of Computer Science, LTH

Outline

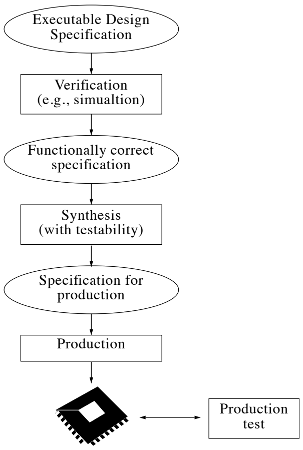
Digital system testing

Design for testability
- Test points
- Scan path
- Built-In Self-Test (BIST)

High-level test synthesis
- Testability analysis
- Testability improvement transformations
- Partitioning

Digital system testing

Testing activities



Executable Design Specification

Verification (e.g., simualtion)

Functionally correct specification

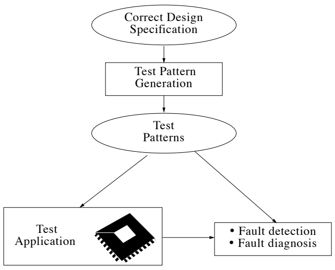
Synthesis (with testability)

Specification for production

Production

Production test

Production Testing Activities



Correct Design Specification

Test Pattern Generation

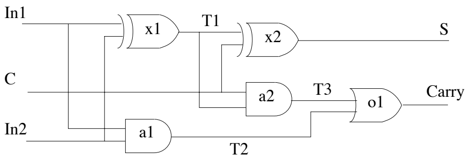
Test Patterns

Test Application

- Fault detection
- Fault diagnosis

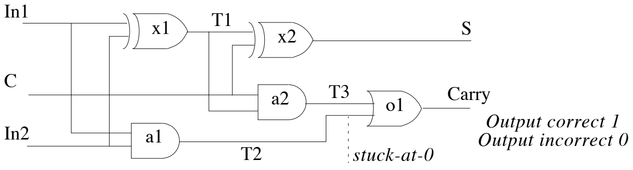
- Test pattern generation
  - ATPG (Automating Test Pattern generation)
  - BIST (Built-In Self Test)
- Testability improvements used (scan, BIST, etc.)

Digital System Testing



- General strategy is to apply a test pattern to primary inputs and observe primary outputs
- Other variations of the basic technique exist, for example I_{ddq} (Integrated Circuit Quiescent Current), BIST
- Test pattern generation is NP-complete problem
- Fault model required for systematic test generation procedure

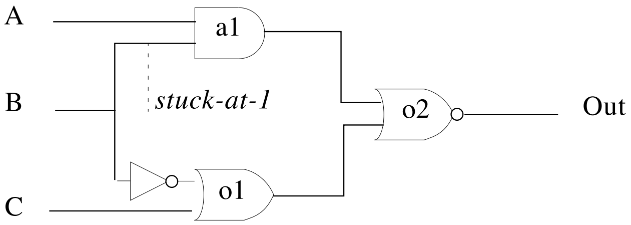
Test Pattern Generation



- **stuck-at-0** and **stuck-at-1** fault model,
- *fault sensitizing*- finds an input pattern that produces a value 1 (0) at the faulty line in case of stuck-at-0 (stuck-at-1) fault,
- *fault propagation*- find additional input patterns which will produce different values at the primary outputs for the correct and faulty circuit.

Test Pattern Generation (cont'd)

- *Reconvergent fanout*- different paths from the same line reconverging at the same component

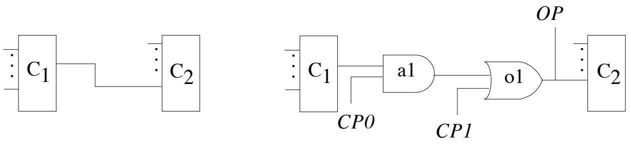


Design for testability

Design for Testability

- ad hoc techniques
  - test points
  - partitioning
- structured techniques
  - scan path
  - BIST
- the main goal is to improve *controllability* and *obesrvability* of selected lines in a design
  - Controllability indicates the relative difficulty of setting a particular line to a specific value from primary inputs
  - Observability indicates the relative difficulty of propagating a value assigned to a particular line to a primary output
- Controllability and observability reflect the two phases of test generation, namely test sensitizing and test propagation

Test Points

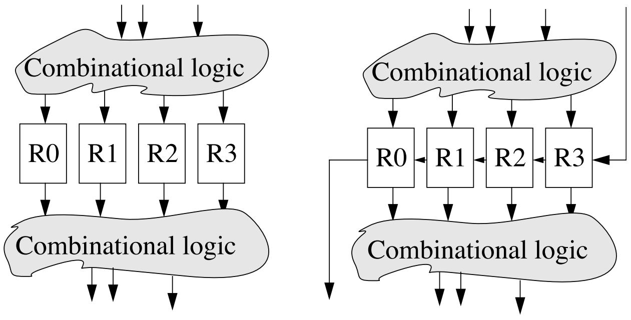


original design

0/1-injection and observation

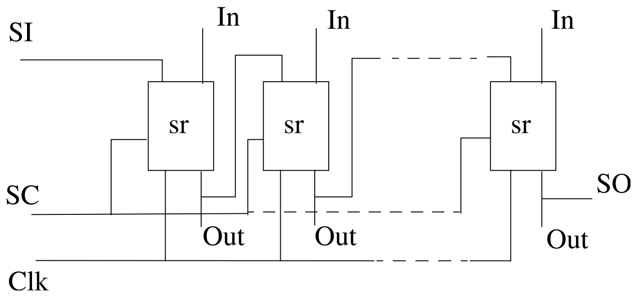
- very simple method
- large overhead in terms of lines (can be alleviated a little bit by using addressing of test points and multiplexing observation points)
- can be used to lines (not only to registers)

Scan Path

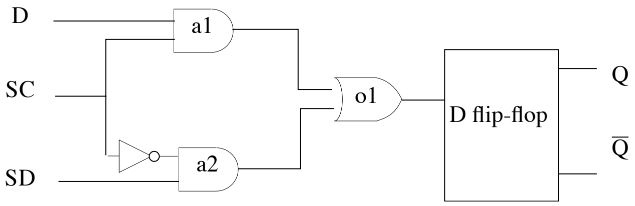


- full scan reduces the sequential test generation problem to a combinational one
- Boundary scan (IEEE Std 1149.1-1990),
- Full vs. partial scan.

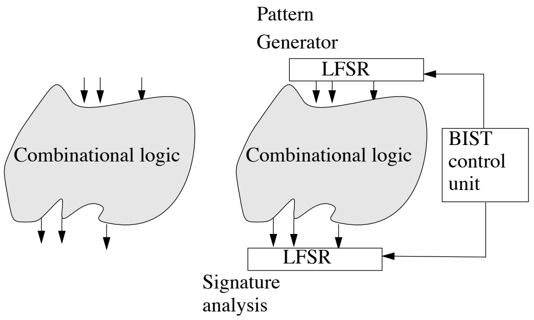
Scan Path



Scan Path (cont’d)



BIST



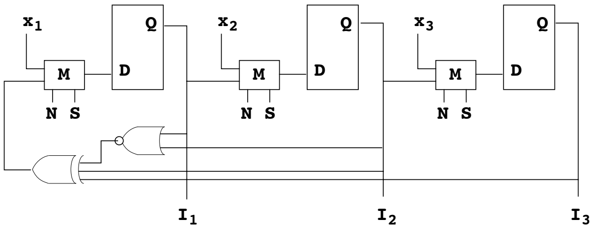
Problems:

- alliasing,
- random pattern resistance,
- . . .

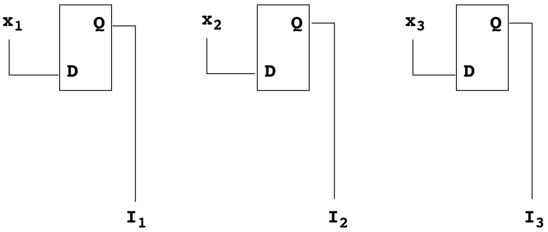
Reconfigurable 3-bit LFSR module

M=

N	*S*	*Mode*
1	x	Normal Operation
0	0	Input Generator
0	1	Signature Analyzer

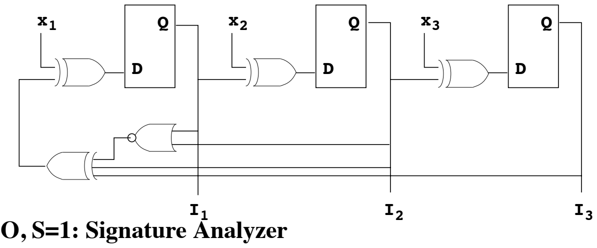


Reconfigurable 3-bit LFSR module (cont’d)



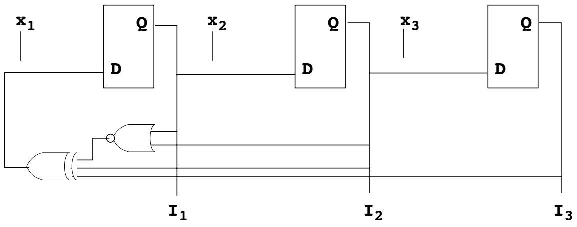
N=1: Normal operation

Reconfigurable 3-bit LFSR module (cont’d)



N=O, S=1: Signature Analyzer

Reconfigurable 3-bit LFSR module (cont'd)



N=O, S=0: Input Generator

Software-based test BIST

- To reduce the hardware overhead cost in the BIST applications the hardware LFSR can be replaced by software.
- Attractive to test SoCs, because of the availability of computing resources directly in the system (a typical SoC usually contains at least one processor core).
- The TPG software is the same for all cores and is stored as a single copy.
- All characteristics of the LFSR are specific to each core and stored in the ROM .
- For each additional core, only the BIST characteristics for this core have to be stored.

High-level test synthesis

High-Level Test Synthesis

- Testablility analysis
  - testability measures
  - testability calculation algorithm
- Testability transformations
  - partial scan insertion
  - test points insertion
  - partitioning

Testability measures

- Controllability and observability are assigned to lines in the design
- They have a combinational and sequential factors CC, SC, CO, SO
- CC and CO are between 0 and 1, where 1 is the best controllability/observability
- SC and SO are natural numbers representing number of clock cycles needed to control/observe a line
- heuristics are developed to find components controllability transfer factor (CTF) and observability transfer factor (OTF)
  - CTF reflects the probability of setting a value at a unit's output by randomly exercising its inputs.
  - OTF reflects the probability of observing a unit's input by randomly exciting its other inputs and observing its outputs.

Testability measures (cont'd)

- controllability of an combinational component
  $CC_{out} = CC_{in} * CTFU$
  $SC_{out} = SC_{in} + clk(S_i)$
  $clk(S_i)$ is the number of clock cycles needed for data operation at a functional unit.

Testability Analysis Algorithm

```
for all primary inputs do
  $CC_{prim\_in} = 1; SC_{prim\_in} = 0;$
do
  select a next unit U;
  -- calculate controllability at a unit U's outputs by
  -- using the average controllability calculated
  -- at its inputs
  if U is a combinational unit then
    $CC_{out} = CC_{in} * CTFU; SC_{out} = SC_{in} + clk(S_i);$
  if U is a sequential unit then
    $CC_{out} = CC_{in} * CC_{cond(S_i,S_j)}; SC_{out} = SC_{in} + clk(S_i, S_j);$
  if U is involved in a feedback loop then
    $CC_{out} = CC_{out} * CC_{loop}; SC_{out} = SC_{out} + SC_{loop};$
until all primary outputs are reached;
```

Testability Improvement Transformations

Partial scan insertion

- calculate a testability evaluation for registers as a average testability (combinational and sequential)
- select registers which are in a data path loop; selected registers must be in different loops,
- if the testability still does not satisfy requirements select a number of registers with the worst testability,
- re-calculate testability measures and repeat previous phase if necessary

Test points insertion

- Insertion of T-cells on similar principles as for registers

Partitioning

- The partitioned circuit works in two modes: normal and test mode,
- The main objective of data path partitioning is to use testability analysis results and other heuristics to find partitioning boundaries and isolate data communication among partitions in the test mode.
- The main procedure
  - selecting partitioning boundaries, and
  - identifying partitions by clustering a set of components surrounded by the partitioning boundaries.

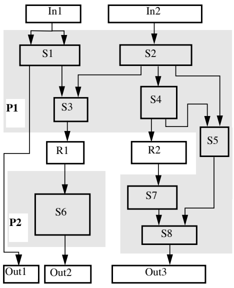
Partitioning (cont'd)

Selection of registers and/or lines to be boundary components:

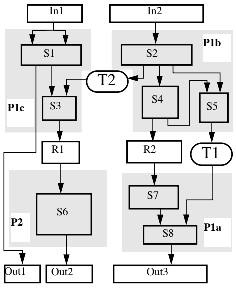
- break feedback loops identified by the testability analysis algorithm
- select registers or lines with the worst testability
- eliminate fanout points
- BIST heuristic

LUND UNIVERSITY

Partitioning example



preliminary partitioning



further partitioning

LUND UNIVERSITY

LUND UNIVERSITY

LTH

FACULTY OF ENGINEERING