COURSE WORK

Design for Testability

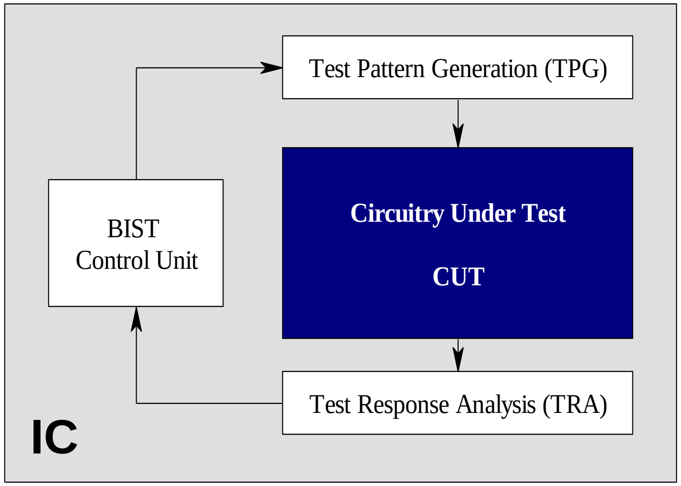
Course Work. Investigations of BIST

The tasks of the course work

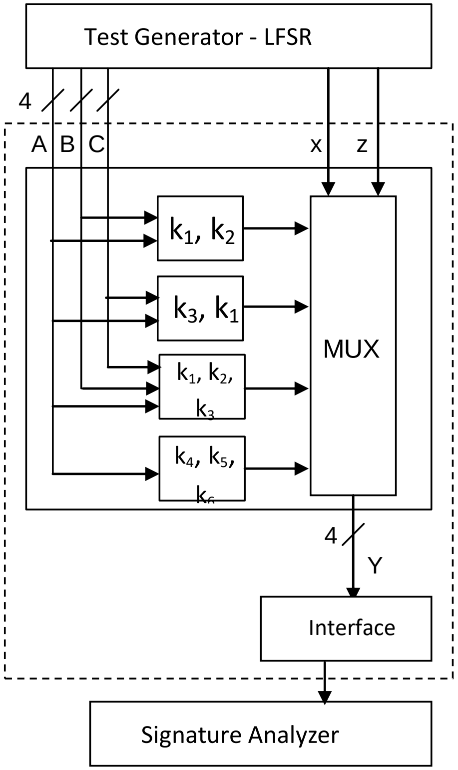
- Design of a circuit
- Evaluation of the testability of the circuit
- Redesign for testability
  - Control points selection, optimization
  - Scan path, optimization
- Built-in self-test. Design of solutions
- Experimental research

Course Work. Introduction

- **In-circuit**
  - **Test pattern generation**
  - **Response verification**
- **Pseudorandom test generation,**
  very long tests
- **Hybrid test solutions**
- **Response compression**



Course Work. Description of the Circuit



1. Design of a combinational circuit for the following functionality

If $x = 0, z = 0$, then $Y = k_1A + k_2B$, else

if $x = 0, z = 1$, then $Y = k_3A - k_1C$, else

if $x = 1, z = 0$, then

$Y = (k_1A \vee k_1B \wedge k_2C) \oplus (k_3C \vee \text{NOT } (k_3A) \wedge k_1B)$,

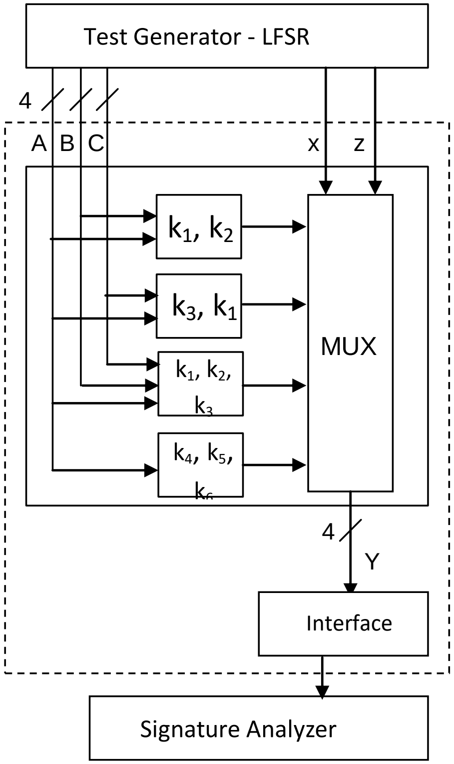
if $x = 1, z = 1$, then $Y = k_4A^2 + k_5A + k_6$

Coefficients k_i can be found on the next slide

Coefficients for the Course Work Versions

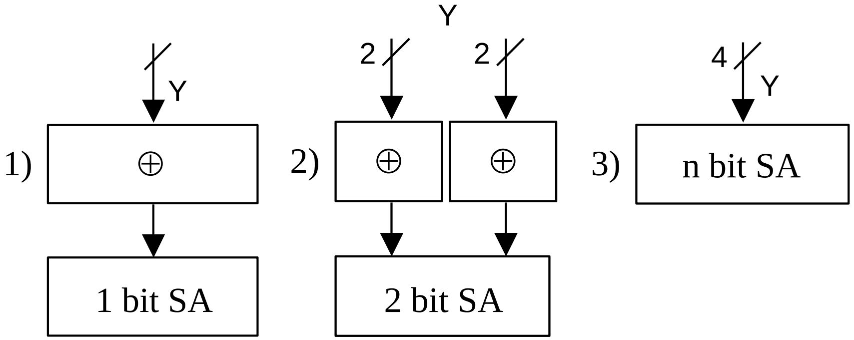
Vers. No.	k_1	k_2	k_3	k_4	k_5	k_6	Vers no.	k_1	k_2	k_3	k_4	k_5	k_6
1	1	1	1	0,1	0,2	0,5	8	1	1	1	1,5	0,1	0,5
2	1	1	0	0,1	0,2	1,0	9	1	1	0	1,5	0,1	1,0
3	1	0	1	0,1	0,2	2,0	10	1	0	1	1,5	0,4	2,0
4	1	0	0	0,1	0,2	3,0	11	1	0	0	1,5	0,4	3,0
5	0	1	1	0,1	1,0	0,5	12	0	1	1	1,5	0,8	0,5
6	0	1	0	0,1	1,0	1,0	13	0	1	0	1,5	0,8	1,0
7	0	0	1	0,1	2,0	2,0	14	0	0	1	1,5	1,5	2,0
15	1	1	1	0,2	0,4	1,0	20	0	0	1	0,5	0,4	1,0
16	1	1	1	0,1	0,4	1,2	21	0	0	1	1,0	0,8	1,5
17	1	1	0	0,2	0,4	0,5	22	0	1	1	0,5	1,2	0,5
18	1	0	1	0,2	0,4	1,0	23	0	1	1	1,0	0,3	1,0
19	0	0	1	0,3	2,0	1,0	24	1	0	1	0,4	1,0	1,5

Course Work. Design of Interface Versions



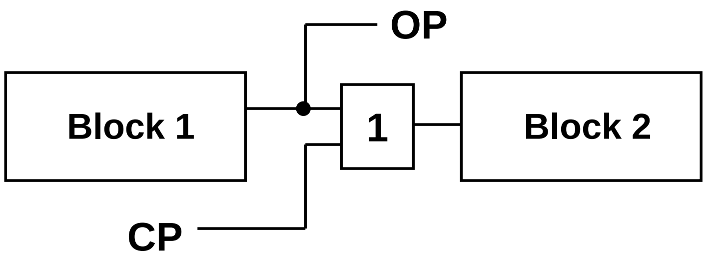
2. Use three different interface versions for experiments: 1 bit, 2-bit and 4- or more bit interfaces for respective n-bit Signature Analyzers

The types of interface:



Course Work. Design of a Testable Circuit

3. Enter the designed gate-level (AND, OR, NOT) combinational circuit into the computer, using CADENCE circuit editor
4. Generate test patterns with Turbo-Tester (TT) ATPG. If the fault coverage is 100%, remove one or more patterns from the test set, so that at least two faults remain undetected.
5. Improve the testability of the circuit to reach again 100% fault coverage with the updated test set



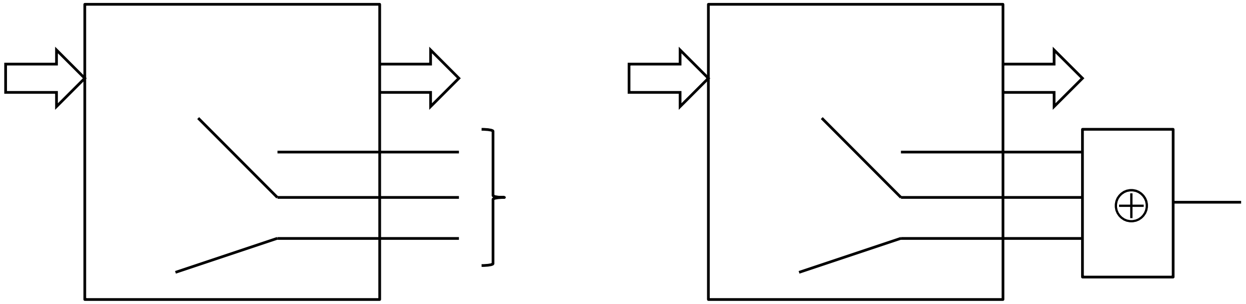
1- controllability:
CP = 0 - normal working mode
CP = 1 - controlling Block 2 with signal 1

Course Work. Observability Investigation

6. Analyze two different testability improvement solutions:

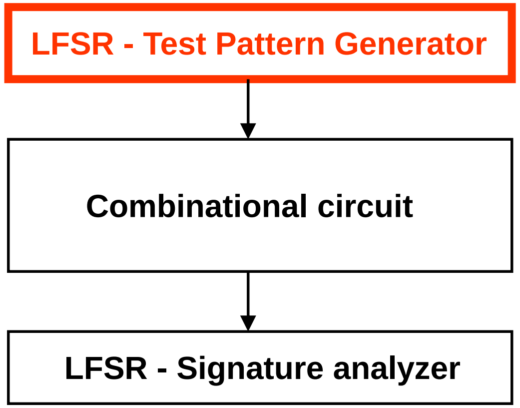
- Separate pins for all observability points
- Single joint pin for all observability points

Draw the graphics for both cases for the function *P* = *f*(*T*) where *P* is fault coverage, and *T* is test length



Course Work. Design of a Test Generator

BILBO - Built- In Logic Block Observer:


LFSR - Test Pattern Generator

Combinational circuit

LFSR - Signature analyzer

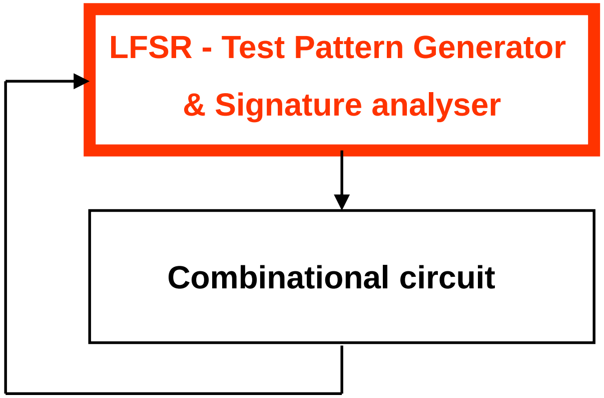
7. Generate test patterns by the BILBO tool for 10 different polynomials, and find the best structure for the LFSR

   Report for all 10 experiments the maximum achievable fault coverage, and fix the minimum test length needed for that

   Calculate the increase of the circuit size (in number of 2-input gates) due to adding of the self-test circuitry

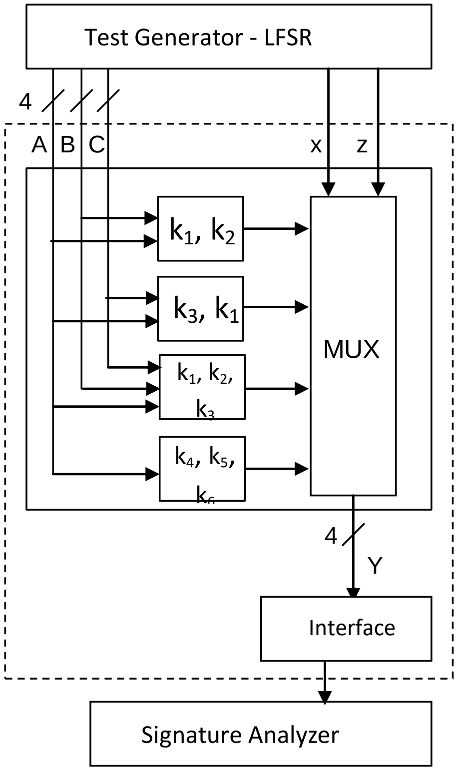
Course Work. Design of a Test Generator

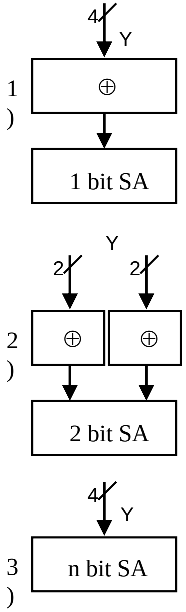
CSTP - Circular Self-Test Path:



8. **Repeat the previous task for the case of using CSTP ("Circular Self Test Path") for self-test purposes**

Course Work. Design of a Signature Analyzer





9. Carry out experiments with the best test set found in task 7 for 4 different Signature Analyzers: 1-bit, 2-bit, 4-bit, and 8-bit

Calculate the fault coverages

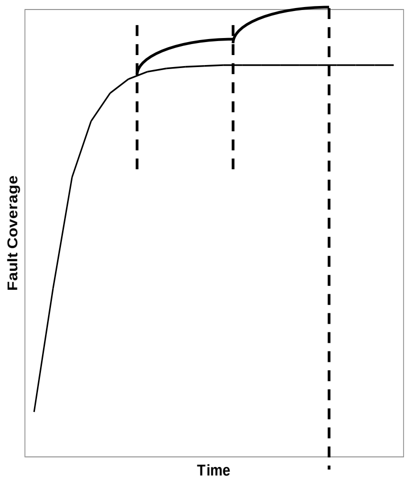
Draw the graphic P = f(SA), where P is
the fault coverage and
SA – is the number of bits in the Signature Analyzer

Draw 4 graphics P = f(T), for 4 SA
cases, where
T – is the test length 5, 10, 15, 20 etc. up to P = 100%
Explain the graphics

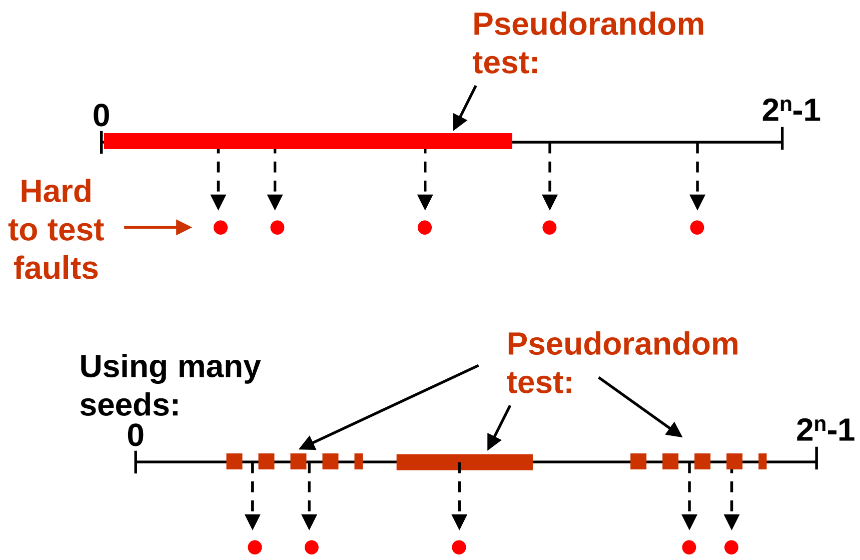
Course Work. Store-and-Generate BIST

The main motivations of using random patterns are:

- low generation cost
- high initial efeciency



Problem: low fault coverage

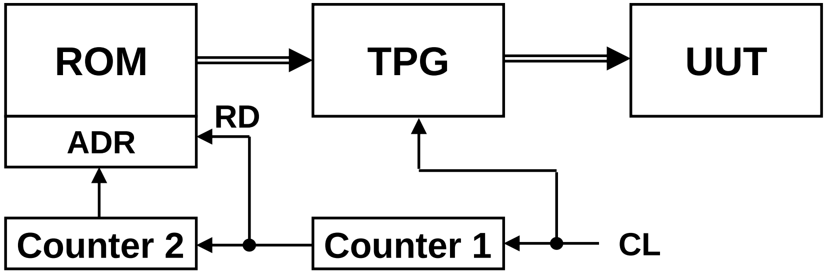


Long PR test:

Using many seeds:

Course Work. Store-and-Generate BIST

10. Synthesize an optimal BIST, using "store & generate“ architecture. Chose for that the best BILBO structure and ja the 100% test with length N. Minimize the number of seeds to be stored in the memory



11. Compare the results in tasks 4, 5, 7, 8 and 10. Which solution is the best and why? Draw the block-level final structure of the selected best BIST solution.
12. Present a report of the course work.