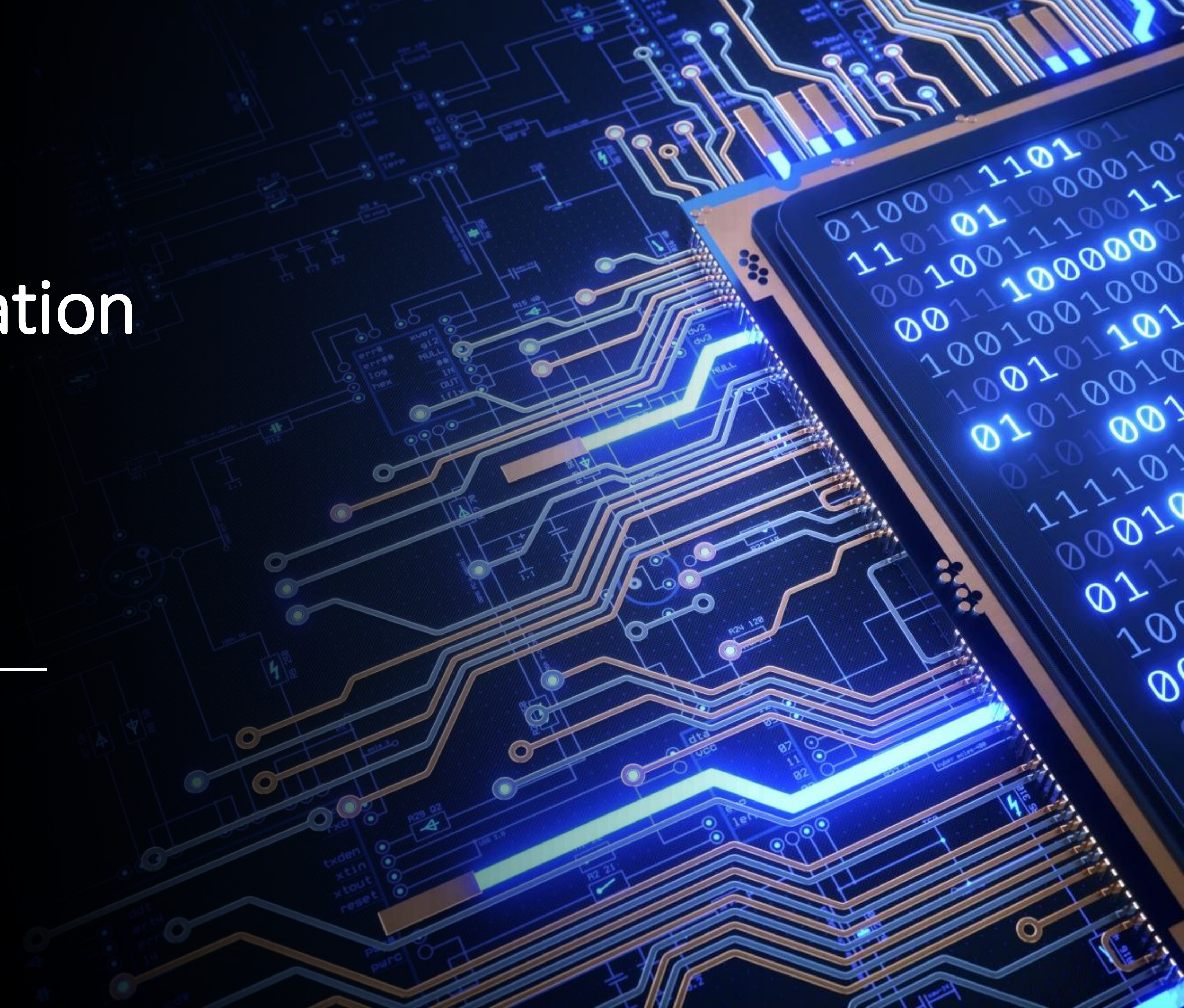
BIST implementation



Group-4

Sasanka Uppuluri(2019102036)

Siva Durga(2019102038)

Viswanadh (2019112011)

Mani Teja(2019102023)

Introduction

- BIST – Built-in Self testing
- We add additional circuits to the original function blocks to enable testing the circuit by itself
- Has a wide variety of uses including cost reduction, testing time reduction.
- Used to avoid the usage of ATE (Automated Test Equipment) which is considered rather costly.

Advantages of BIST

- Lower test costing
- Testing is independent of future technology
- Better fault coverage
- Shorter test time
- Easier customer support

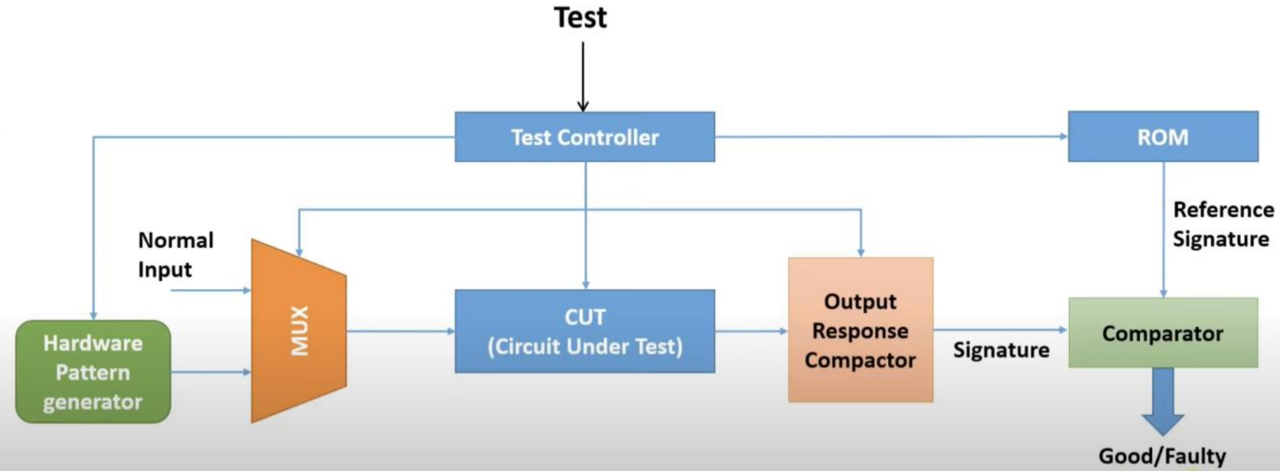


Fig: Basic architecture of BIST

Implementation

- **Test Pattern Generator and Response analyzer**
    - Test Pattern Generator and Response Analyzer are often implemented using Linear-Feedback Shift Registers (LFSR).
    - The LFSR is an n-bit shift register which pseudo-randomly scrolls between 2^n-1 values.
    - The Pseudo-Random Pattern Generator generates the necessary patterns which are fed as inputs to the CUT.
- **Circuit Under Test(CUT)** [Figure on the top-right]
- **Comparator**
    - A simple n-bit comparator has been used to compare the fault-free signature with the faulty one.

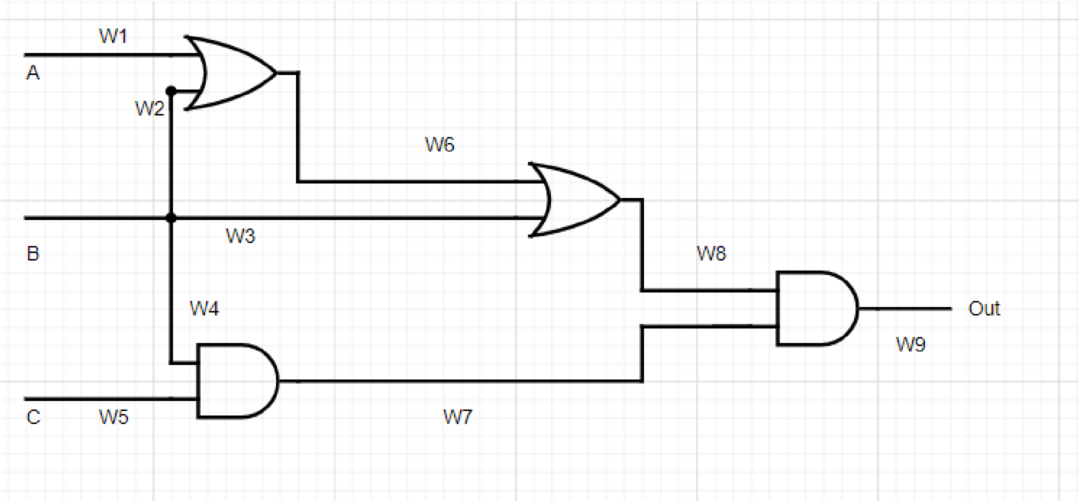


Fig: Circuit Under Test

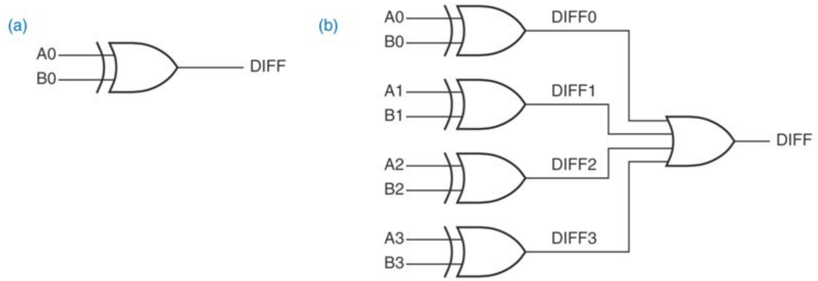
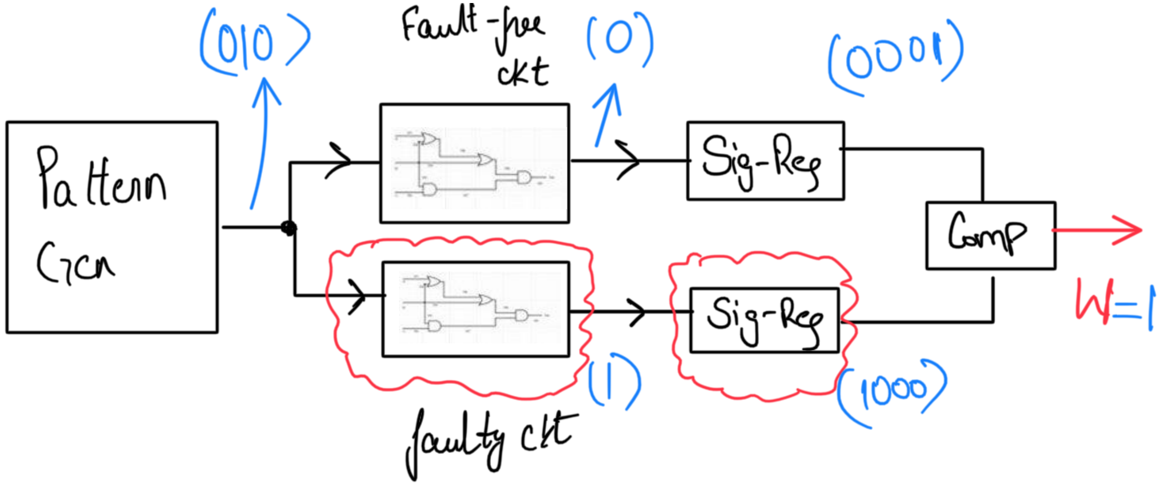


Fig: Comparator

Working and Results

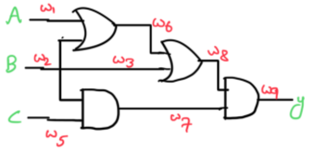
- - If there's a s-a-f in faulty circuit (here w7 is stuck at 1) then output's get toggled, setting the comparator output to 1 which indicates that the CUT is faulty



Error	Comp Output (W)
Not Present	0
Present	1

Work Done

- Implemented an ideal Online-non concurrent BIST.
- Consists of five main modules: Implemented each in verilog
- Implement a flexible combinational circuit where stuck-at-faults can be given as arguments to the module instead of hardcoding.
- Prepared a wrapper ALU to integrate all the individual modules and designed a testbench for the same
- Testbench includes 3 different cases: I) ckt with 0 saf, ii) fault-free ckt, iii) ckt with 1 saf



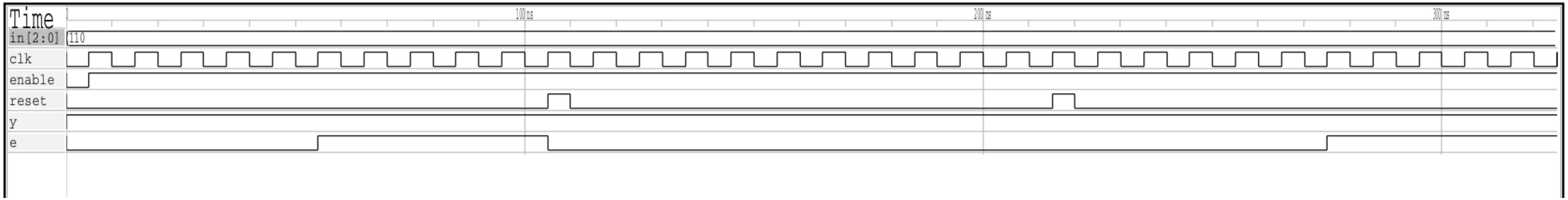
Truth table

A	B	C	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

stuck at faults

w_1 = check[i]

Val= stuck at fault
location value


Time
100 ns
200 ns
300 ns
in[2:0]
110
clk
enable
reset
y
e