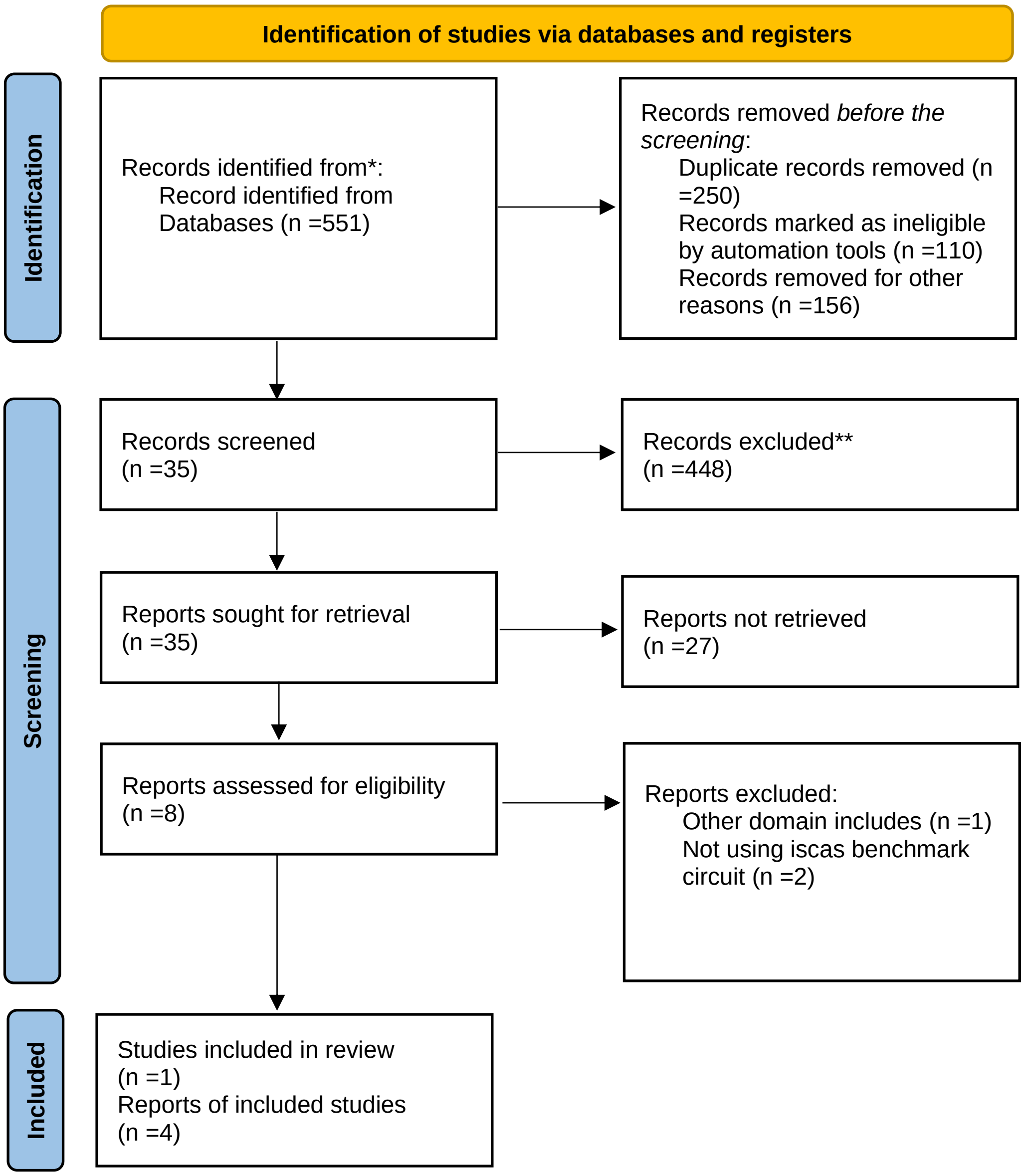
NAME: Siddharth Singh Upadhyay **Roll no.:** 2021VLSI-06

Subject: Research Methodology



Identification of studies via databases and registers

Identification

Records identified from*:
Record identified from Databases (n =551)

Records removed *before the screening*:
Duplicate records removed (n =250)
Records marked as ineligible by automation tools (n =110)
Records removed for other reasons (n =156)

Screening

Records screened
(n =35)

Records excluded**
(n =448)

Reports sought for retrieval
(n =35)

Reports not retrieved
(n =27)

Reports assessed for eligibility
(n =8)

Reports excluded:
Other domain includes (n =1)
Not using iscas benchmark circuit (n =2)

Included

Studies included in review
(n =1)
Reports of included studies
(n =4)

Hardware Trojan Detection by increasing transition Probability using Insertion in Rare net

Siddharth Singh Upadhyay

M.Tech, VLSI & Embedded system

ABV-IIITM, GWALIOR (M.P)

Abstract**: hardware-based vulnerabilities are present due to outsourcing during the time of various stages in IC fabrication. There may be a chance some virulent circuit is present in the IC are hardware trojan horses. Government, securities, defence agencies, and industries raise concerns about HTH. Here we are using some techniques where we stimulate or increase transition in the functionality of trojans. Here we target to significantly rise trojan-activity and decrease the activation time of the trojan circuit. Rises the probability of net transition using some circuitry changes with tri-buffer and scan registers. And the transitions numbers are increasing rare net in the iscasc'89 benchmark circuit. We are using some tri-state buffer circuits with a scan register to stimulate the transition probability. Also, here we are using Xilinx vivado to check the circuit output with or without insertion.**

Keywords: Xilinx vivado, HTH detection, transition probability, tri-state buffer, Weighted signal probability.

Introduction:

In the IC fabrication, design, and manufacturing process various levels, stages like [1] Design integration, RTL netlist, verification, physical synthesis, layout, test, and PCB assembly in these stages may be design stages or foundry stages add some corrupted circuits which stole information during using these IC. An attacker uses many techniques like reverse engineering also side-channel analysis or counterfeiting to extract secret information.

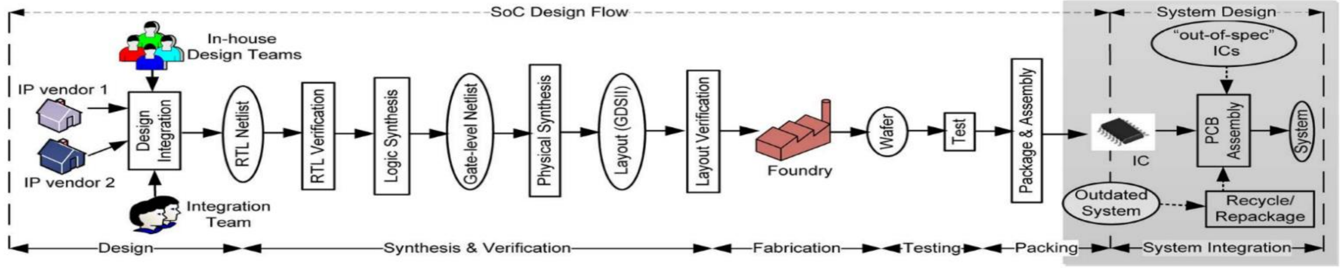


Fig.1.:- various stages where trojan inserted in the IC

When signal transitions from one different state to the next state in a single cycle is called (TP) transition probability. In this, we calculate the probability before trojan insertion or after trojan insertion.

The inserted trojan circuit in the rare nets has less number of transitions in signals and converts them. For the (design for security) DFS adds some circuits that help to increase the TP of signals. which helps to find out the HT present on the Integrated circuit. Increasing the TP triggers the malicious circuit which has some other activities which we can detect. Increasing the TP of the signal we do some other parametric analysis or maybe direct trigger the payload circuit.

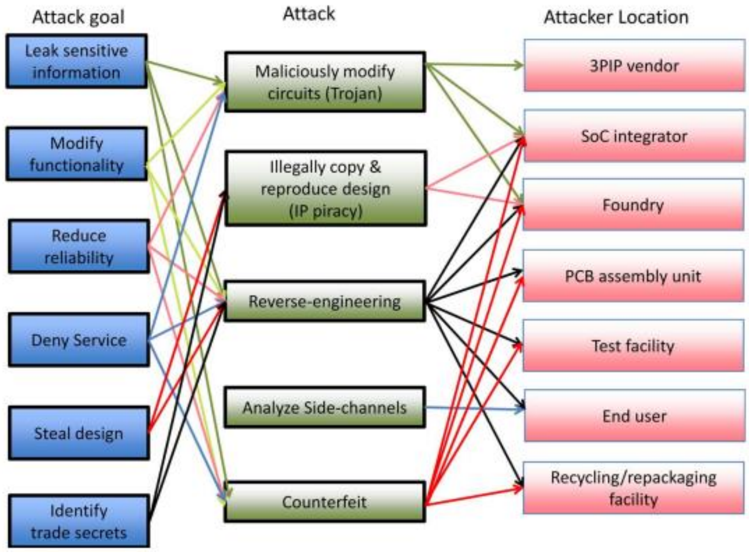
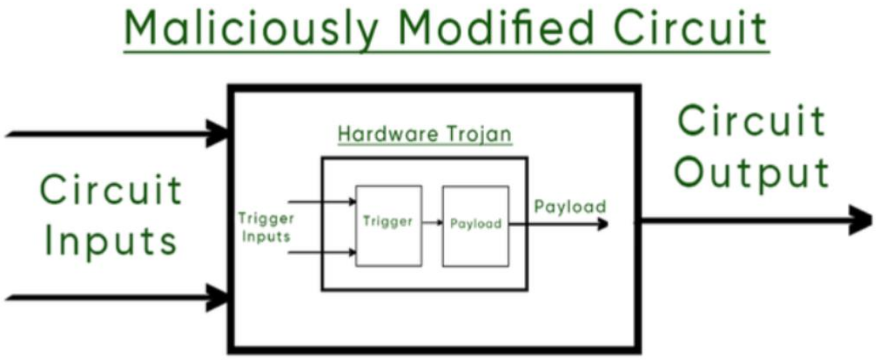


Fig.2: - [1] attack, attacker location, and goal of the attack.

In DFT [5] (design for testing) we exploit the rare nets with transition probability set near the threshold. In process of trojan detection, dummy scan flip flop is inserted to increase the transition probability up to the threshold. In this improved work, various other techniques are used where using 2 in 1 mux also with weighted signal probability.



[4] Weighted signal probability is found by dividing the number of ways to achieve the needed outcome by the number of total possible outcomes.

Here weighted signals are provided in a different or separate node of IC to increase the transition probability using a tri-state buffer to their threshold value and find out the trojan presence in the IC with minimum DFT insertion.

Proposed method:

In the proposed method we just want to increase the transition probability of the circuit using some algorithm. we know that only TP is maximum when signal probability goes to 0.5. the relation between transition probability and signal probability is

$$TPi \quad = SPi *(1-SPi)$$

Here i represent the ith net.

In this method, [2] we use buffer instead DFFS or 2to1 mux both have 6 transistors inserted in the net but for reducing spatial effect per DFT we use tristate buffer which has 4 transistors minimum circuit insertion.

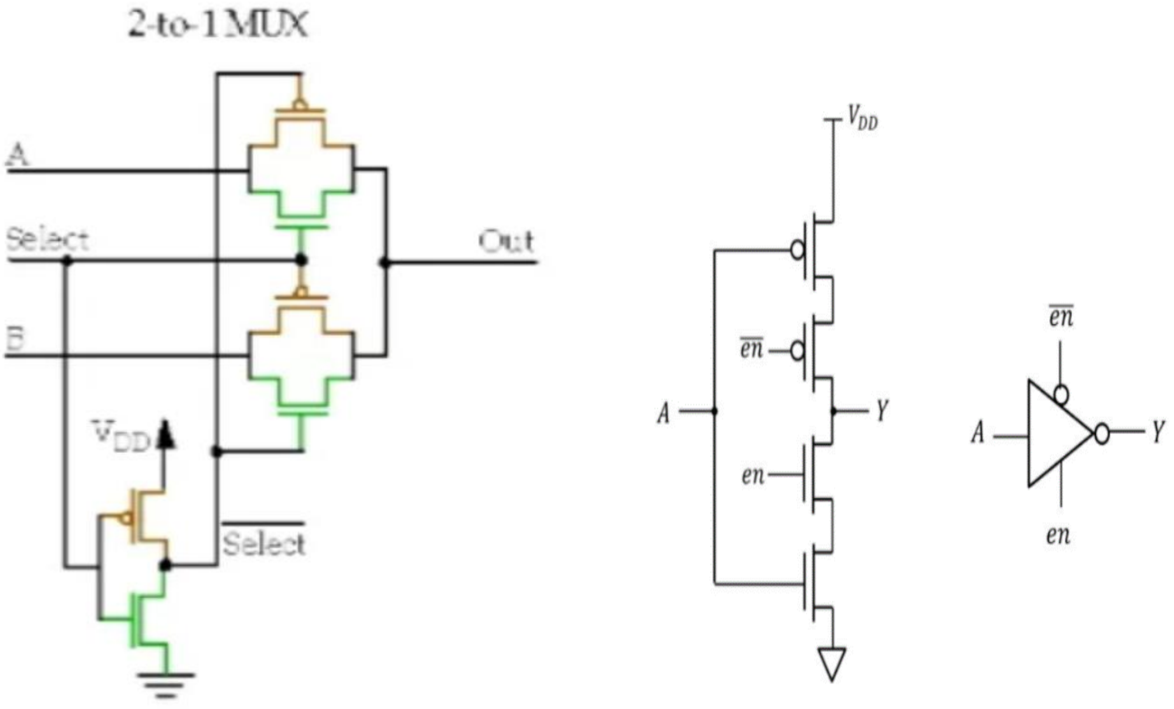


Fig.3: - number of transistors present in 2in1 or tristate buffer in the circuit.

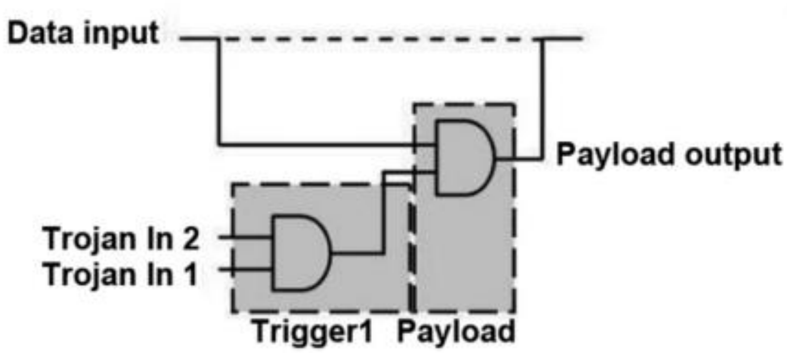


Fig.4.: - Trigger with payload circuit

Test architecture:

tri-state buffer is serially connected with the scan register used to improve the circuit. Tri-state buffer with scan register put on the node net were having the lowest transition probability .and [4] weighted signal probability using LFSR.

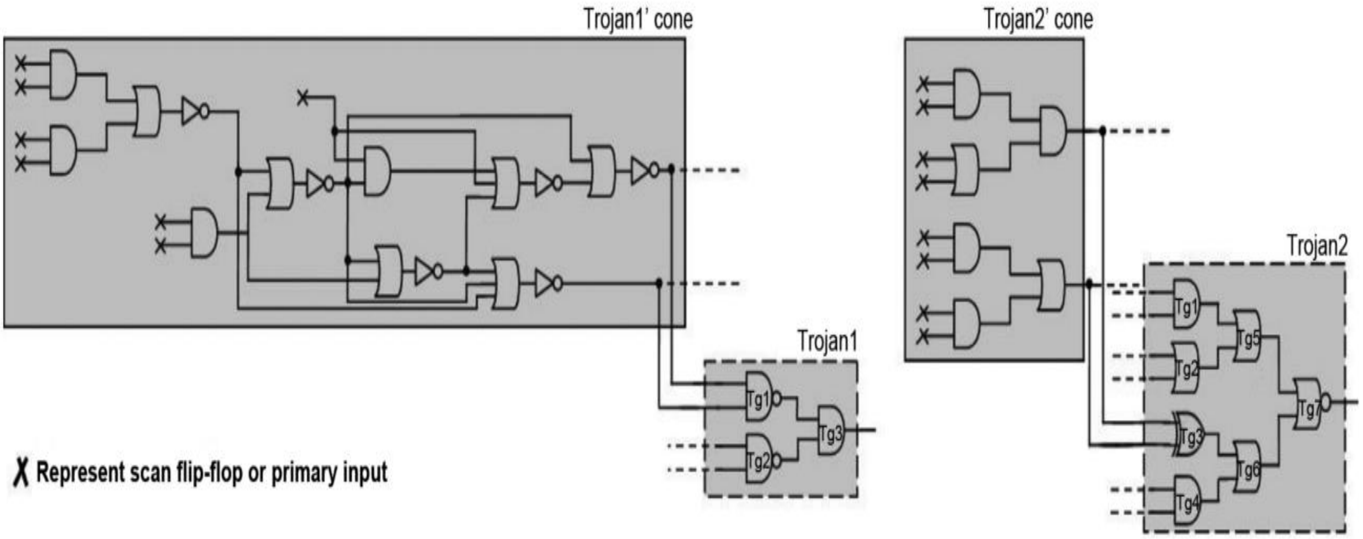


Fig.5.: - trojan cone in the insertion circuit.

Exhaustive way evaluated when some insertion point in IC, overall variance or mean of transition probability for each WSP present in the IC.

Here we choose that point of WSP where the mean of TP is the maximum value or variance is the minimum value.

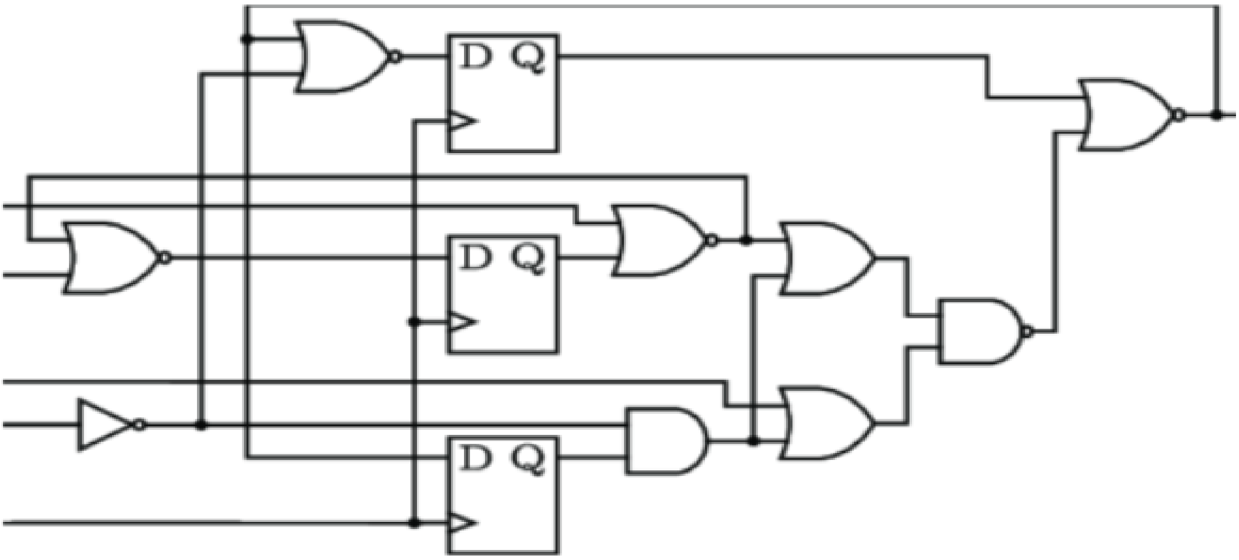


Fig.6: -iscas’89 S27 benchmark circuit

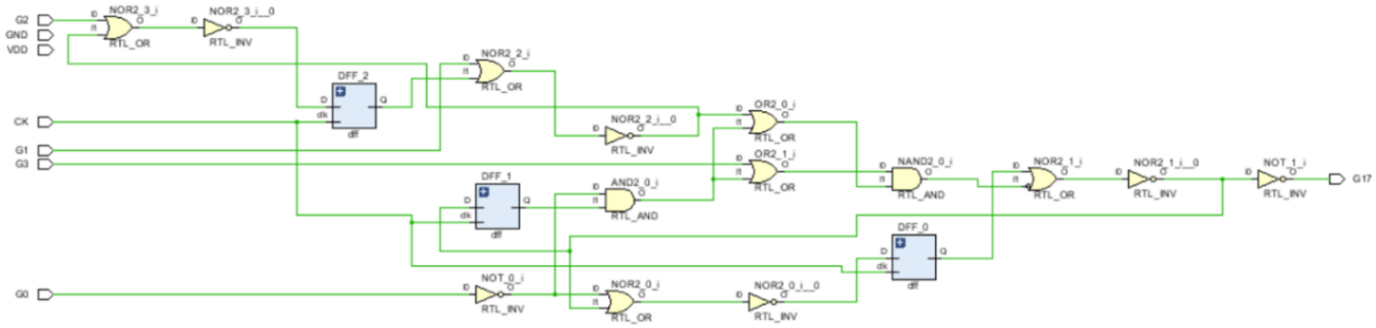


Fig.7: - schematic for iscas’89 circuit

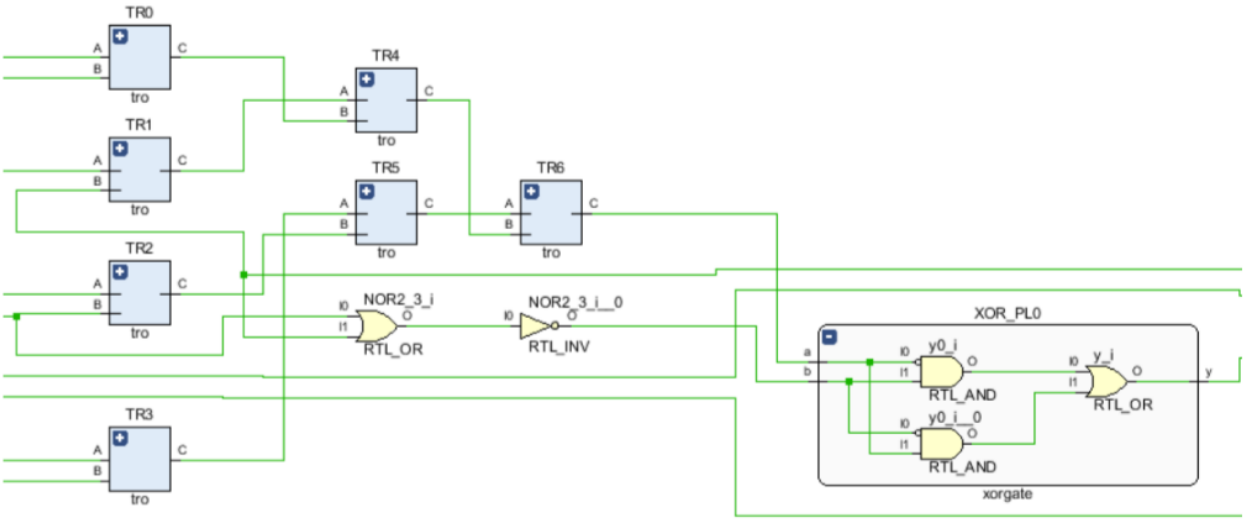


Fig.8: - insertion circuit with tristate buffer and xor gate payload.

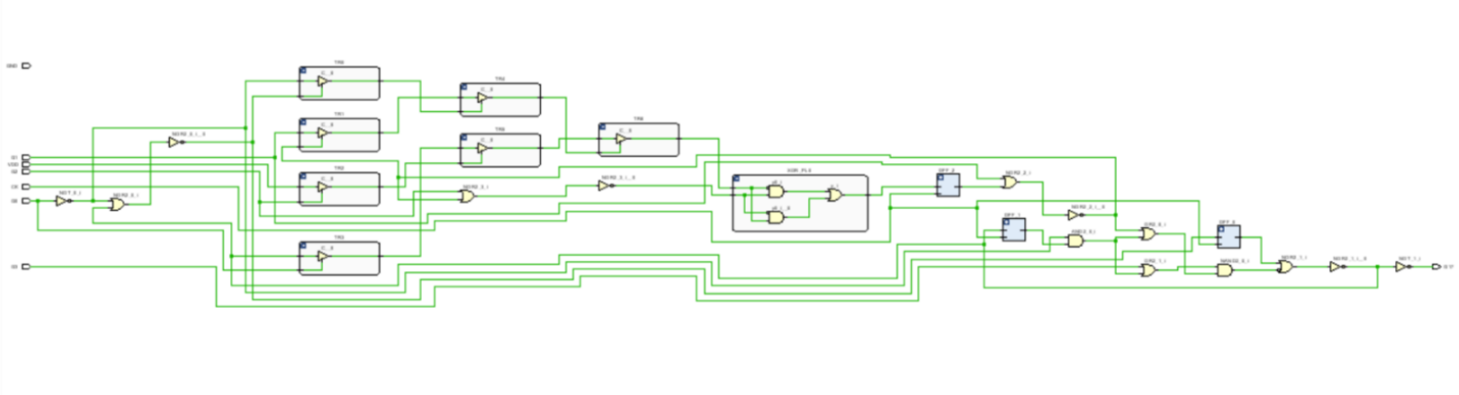
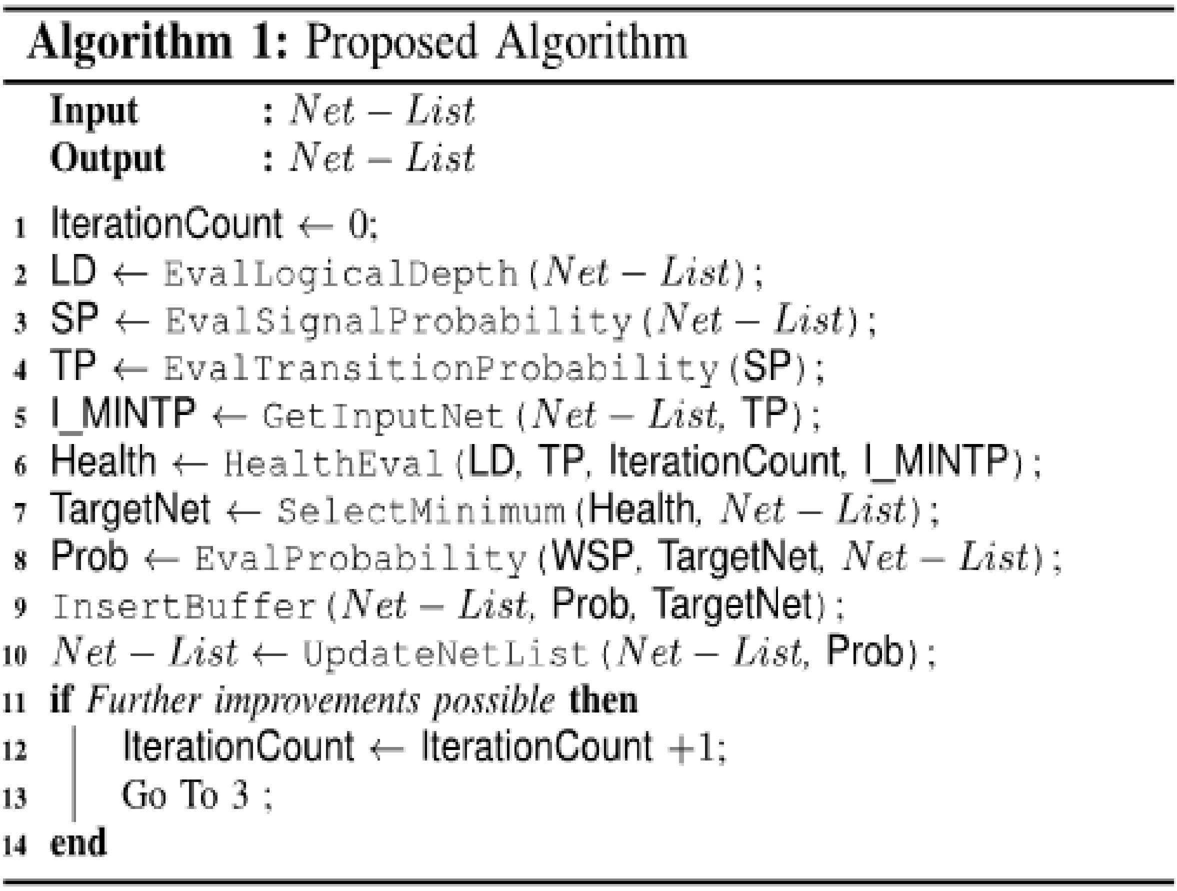


Fig.9: -This is a schematic for the iscas’89 benchmark circuit with insertion.

After inserting the buffer into the circuit, we make a new circuit that has a trigger circuit with buffer and payload with xor gate.

Algorithm:

in this algorithm,[3] we store the lowest TP on the I_MINTP array. this algorithm stops where the mean value of TP and variance value also stop increasing or decreasing in insertion.



Algorithm 1: Proposed Algorithm

```
Input  : $Net - List$
Output : $Net - List$
1  IterationCount ← 0;
2  LD ← EvalLogicalDepth($Net - List$);
3  SP ← EvalSignalProbability($Net - List$);
4  TP ← EvalTransitionProbability(SP);
5  I_MINTP ← GetInputNet($Net - List$, TP);
6  Health ← HealthEval(LD, TP, IterationCount, I_MINTP);
7  TargetNet ← SelectMinimum(Health, $Net - List$);
8  Prob ← EvalProbability(WSP, TargetNet, $Net - List$);
9  InsertBuffer($Net - List$, Prob, TargetNet);
10 $Net - List$ ← UpdateNetList($Net - List$, Prob);
11 if Further improvements possible then
12     IterationCount ← IterationCount +1;
13     Go To 3 ;
14 end
```

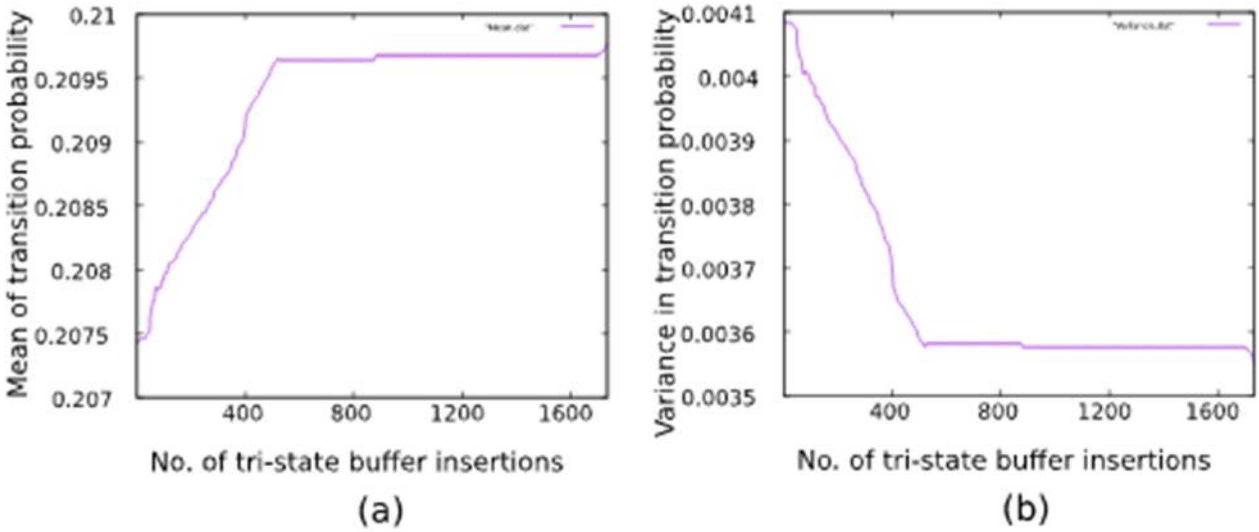
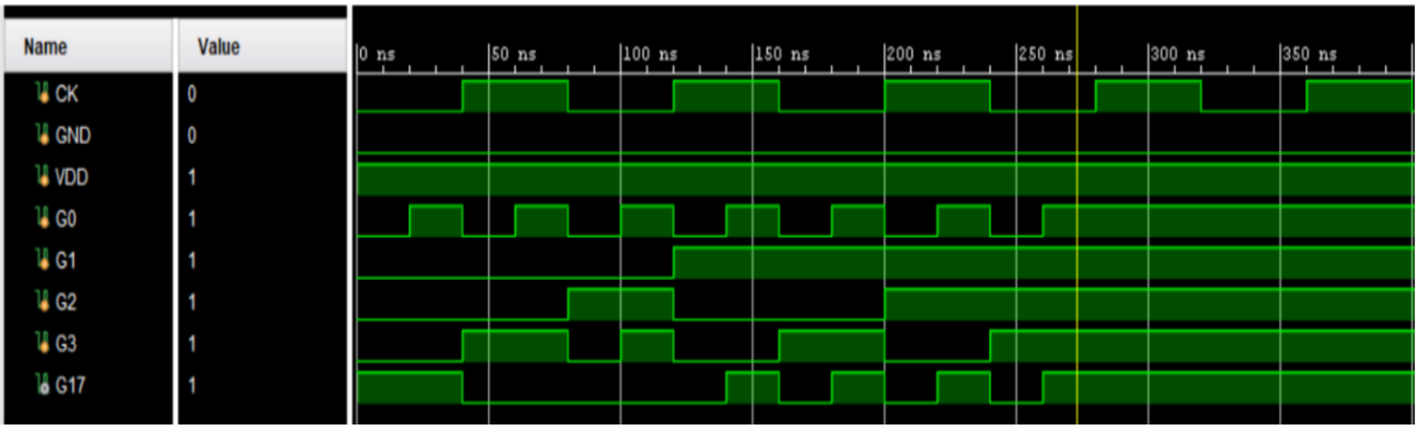


Fig.10.: - [3] this is the experimental result of TP when insertion of a tri-state buffer.

Experimental result:

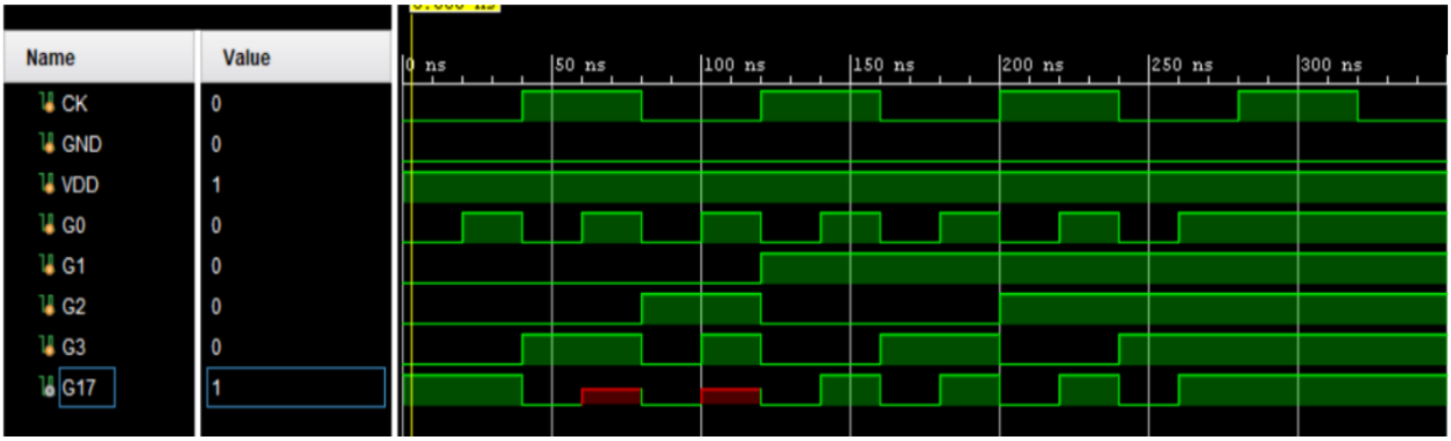
here we are using the iscas'89 benchmark circuit which output detects without insertion or with insertion and we find out some trojan in the system in the output wave.

We choose some node points where after insertion the transition probability are increase so we easily detect the trojan present in the circuit. Here we compare two modules' output without insertion or with insertion and then find out some output changes or faults in the nets.



Name	Value
CK	0
GND	0
VDD	1
G0	1
G1	1
G2	1
G3	1
G17	1

Fig.11.: -This is iscas'89 circuit output without insertion



Name	Value
CK	0
GND	0
VDD	1
G0	0
G1	0
G2	0
G3	0
G17	1

Fig.12.: -This is the iscas'89 circuit output with insertion.

As we see in the output, we find out some fault in the G17 line after insertion because the transition probability increase. so we detect the fault and say some differential functional circuit present in the circuit.

Conclusion:

here we are using a direct triggering method to find out the trojan in the IC system. The second method is side-channel analyses which increase the transition probability and shows the presence of any hardware trojan on the integrated circuit. their heuristic analysis, we increase the probability of TP transition and also the WSP after the insertion of the tri-state buffer.

We see the improvement of the results in the detection of trojans in the circuit.

References:

[1]-M. Rostami, F. Koushanfar and R. Karri, "A Primer on Hardware Security: Models, Methods, and Metrics," in Proceedings of the IEEE, vol. 102, no. 8, pp. 1283-1295, Aug. 2014, doi: 10.1109/JPROC.2014.2335155

[2]-H. Salmani, M. Tehranipoor and J. Plusquellic, "A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time," in IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 20, no. 1, pp. 112-125, Jan. 2012, doi: 10.1109/TVLSI.2010.2093547.

[3]-T. Dhar, S. K. Roy and C. Giri, "Hardware Trojan Detection by Stimulating Transitions in Rare Nets," 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID), 2019, pp. 537-538, doi: 10.1109/VLSID.2019.00124.

[4]-B. Zhou, W. Zhang, S. Thambipillai, J. Teo Kian Jin, V. Chaturvedi and T. Luo, "Cost-efficient Acceleration of Hardware Trojan Detection Through Fan-Out Cone Analysis and Weighted Random Pattern Technique," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 5, pp. 792-805, May 2016, doi: 10.1109/TCAD.2015.2460551.

[5]-M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," in IEEE Design & Test of Computers, vol. 27, no. 1, pp. 10-25, Jan.-Feb. 2010, doi: 10.1109/MDT.2010.7.