Avatar : Reinforcing Fault Attack Countermeasures in EDA with Fault Transformations

PRITHWISH BASU ROY, PATANJALI SLPSK & CHESTER REBEIRO

About Us

Prithwish Basu Roy is currently pursuing M.S. by Research in *Dept. Of Computer Science & Engineering, Indian Institute of Technology Madras, Chennai, India*

Dr. Patanjali SLPSK is currently a post-doctoral candidate in *University of Florida, Gainesville,US*

Dr. Chester Rebeiro is associate professor at *Dept. Of Computer Science & Engineering, Indian Institute of Technology Madras, Chennai, India*

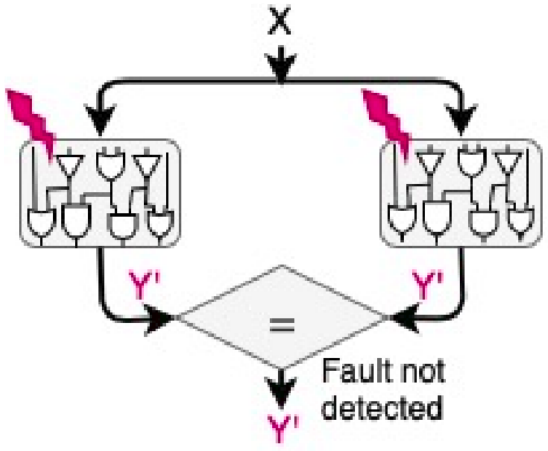
Contents

Faults and Fault Injection Techniques

- Fault – an agent that causes unexpected(erroneous) behaviour of a circuit
- Various methods of Injection of Fault
  - Clock Glitch
  - Voltage Glitch
  - Laser Beam
  - EM radiation
  - Ion Beams
  - Temperature

Spatial Redundancy - Expectation

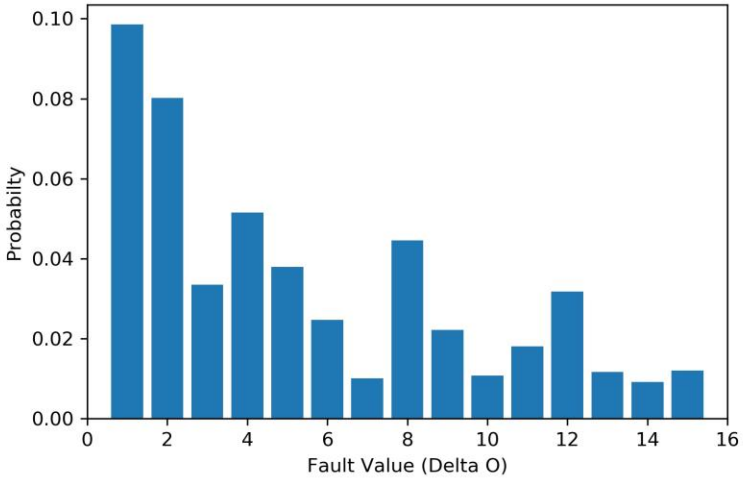
- Make one or more copy of the circuit
- Compares the output from each of them using a comparator logic
- Assumption : Chances of faults occurring are uniform
- Example : For one round of AES $f_1, f_2 \ldots\ldots, f_n$, where $n = 2^{128}-1$ faults are possible with equal probability



Typical Redundancy

Spatial Redundancy - Reality

- Reality : Some faults occur more frequently than others
- A single bit faults are more frequent than double bit faults
- Probability of single register flipping value is more than two flipping values together



Consequences

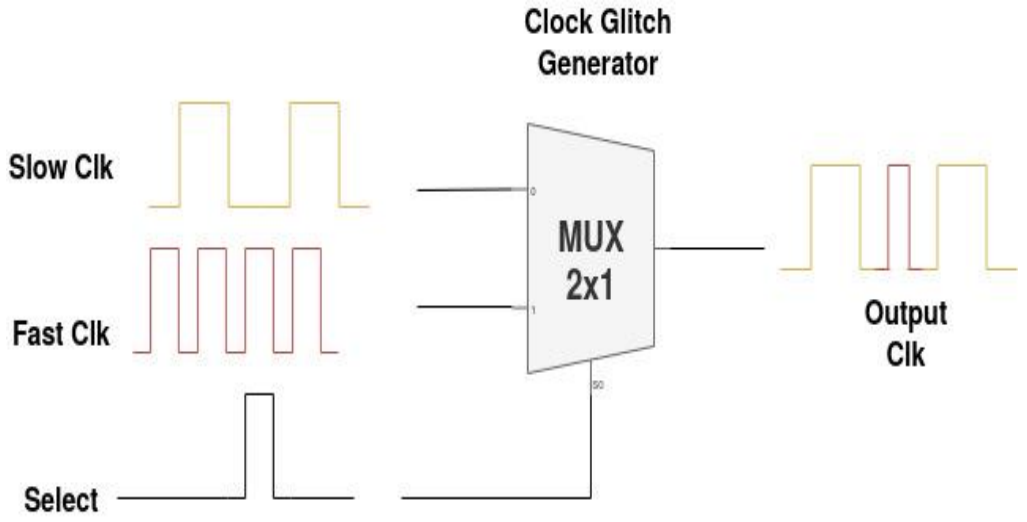
Attackers trying to inject faults which are more frequent

Bypassing the comparator logic becomes feasible

Attacks like Differential Fault Analysis can now be easily carried out

Clock Glitch Attack

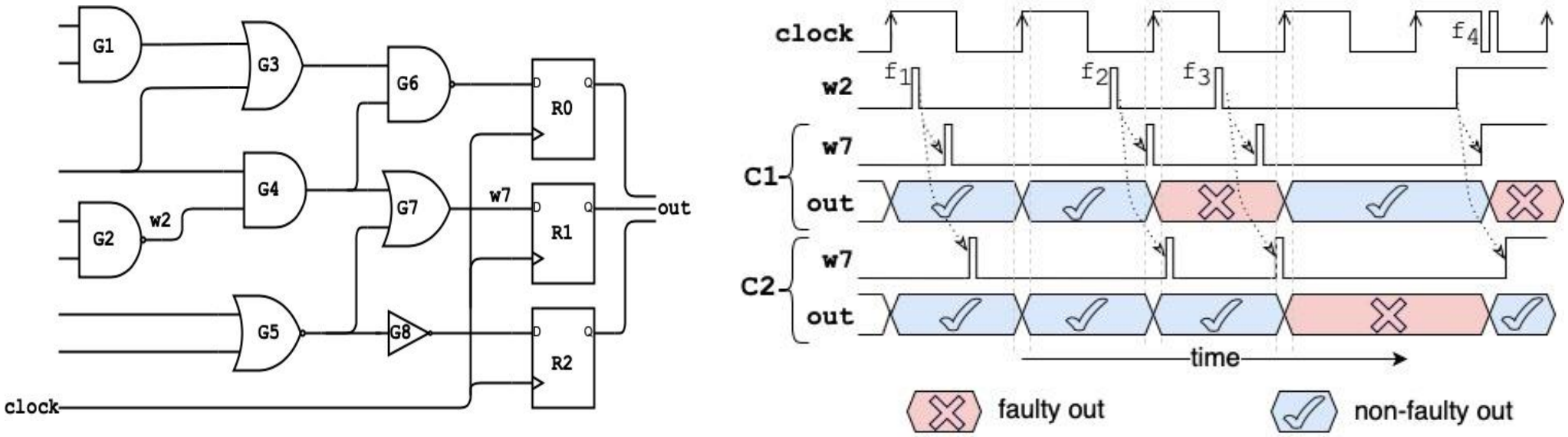
- Uses a (2x1) MUX that has fast and slow clocks as inputs
- MUX output is switched from fast to slow for a short duration
- May lead to setup time violation in the registers
- Registers don't latch the values properly
- Leads to a non-deterministic state



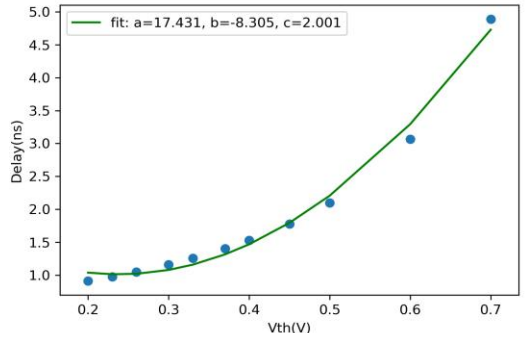
Delay – Key to Setup time violation

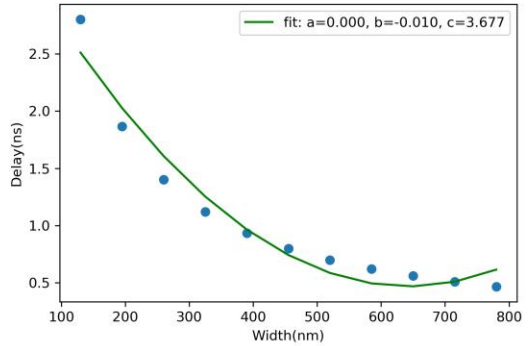
- Setup time – The minimum time required for the flop to latch a change in signal
- Arrival of signal before setup time of the flop ensures correct operation
- Attackers try to manipulate the arrival time of the signal or the position of incoming clock edge
- A designer can only control the arrival time of the circuit
- Arrival Time in turn is determined by the delay of the gates in the path

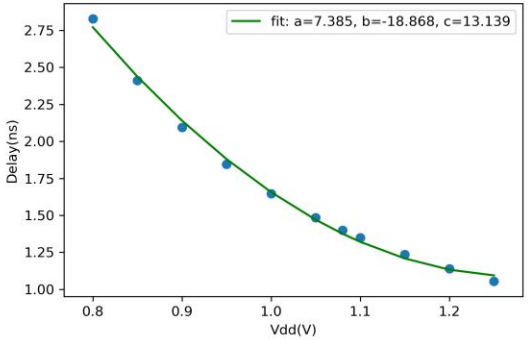
Delay – Key to Setup time violation



Analysis of how faults are propagated(right) through different realizations(C1,C2) of the same circuit C(left). C1 is faster compared to C2.







Can we control Delay while designing? Yes!

- Delay is governed by three factors which a designer has control over
  - Threshold Voltage(proportional to transitional delay)
  - Gate Size(Increase in W decreases output transitional delay)
  - Supply Voltage(inversely proportional to propagation delay)

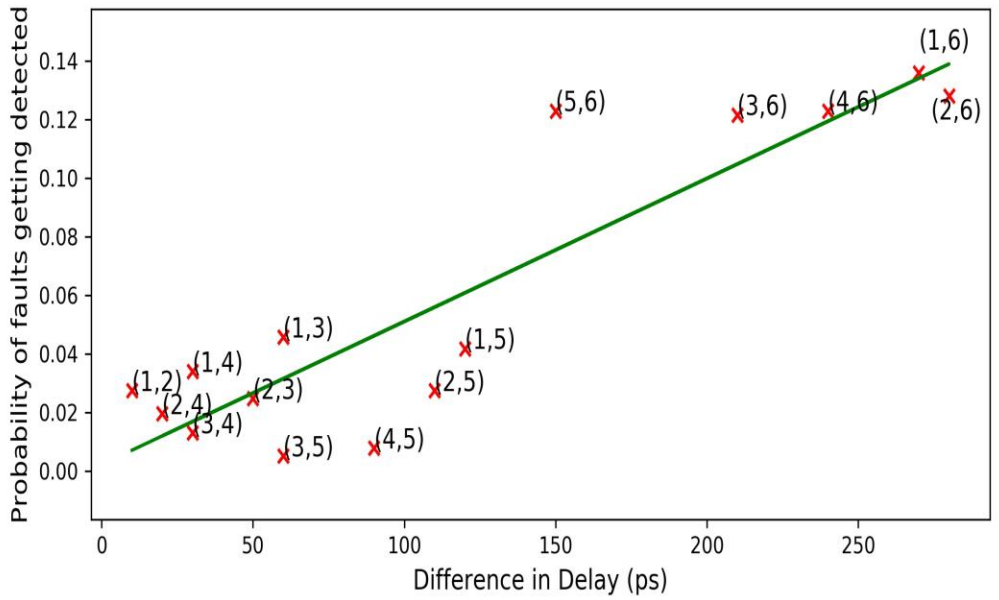
Arrival Time and Fault

- Assumption : Attacker is trying to insert a glitch just before the actual clock pulse arrives
- A register mistakes it as a clock edge and latches wrong value
- If a signal arrives very early compared to the glitch such that (Glitch Time - A.T.(signal) > setup time) then there will be no error
- Similarly, a signal arrives so late such that (Glitch Time - A.T.(signal) < setup time) will always result in a fault

Difference in Arrival Time and Probability of Fault Detection

Difference in Arrival Time and Probability of Fault Detections are linearly correlated. 6 different configuration of a PRESENT S-Box has been used to collect the data.

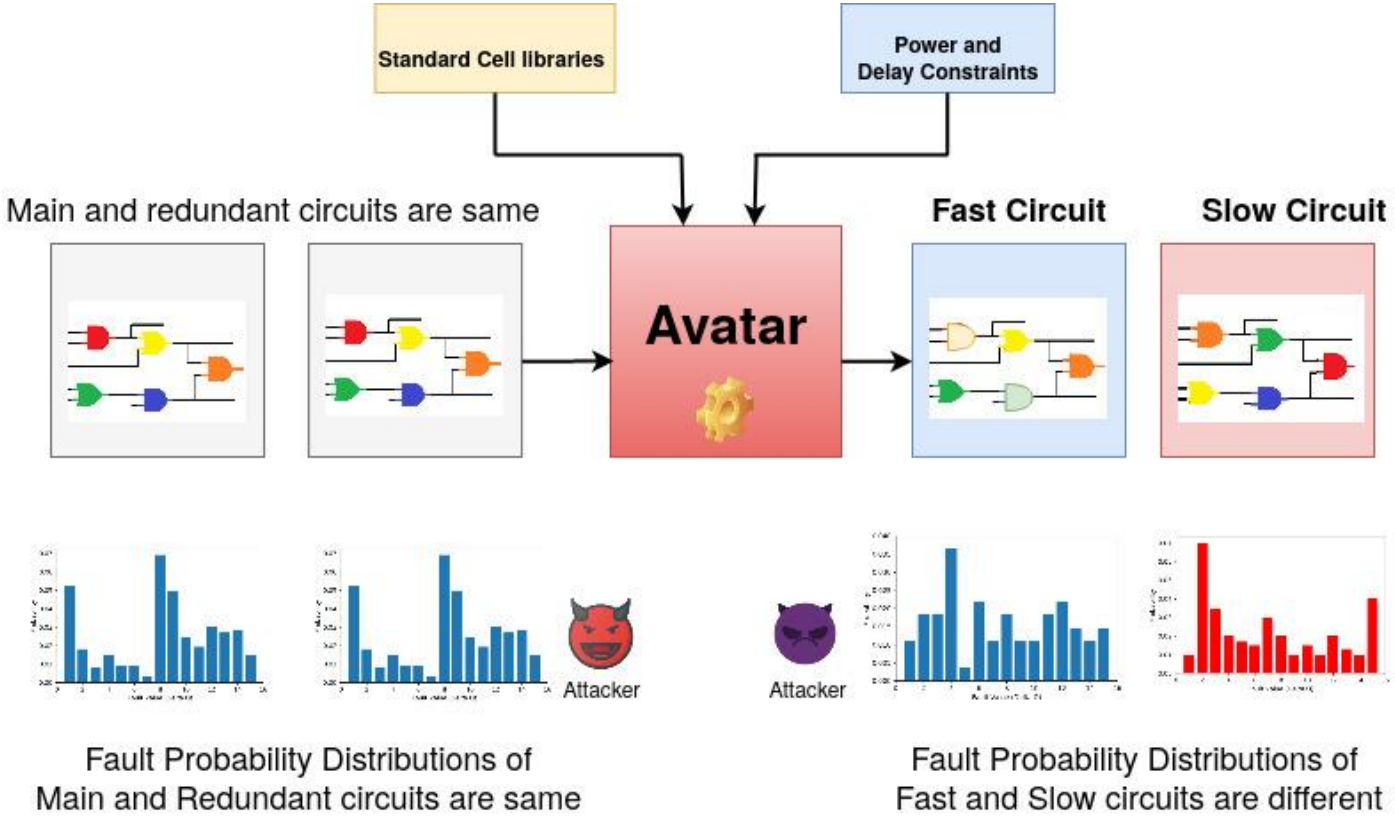
A.T. are in the order of C1<C2<C4<C3<C5<C6



Naïve solution and its limitations

- Since the probability of fault getting detected and the difference in delay are linearly correlated
- Making one circuit the fastest and the other circuit the slowest will cause maximum detection of faults
- What about the overheads?
  - Power ↑
  - Performance(due to slow circuit) ↑
- Conclusion : **Naïve solution will blow the overheads off the roof**

Avatar to the rescue



Avatar to the rescue

- Takes Standard Cell Library as input
- Takes the netlist as in input
- Takes the user's input for Max Leakage Power that can be allowed
- Takes the user's input for the maximum performance overhead that can be allowed
- Uses two modules **AvatarFast** and **AvatarSlow** to speed up one circuit and slow down the other one
- After each successful gate replacement Avatar verifies whether any of the user defined constraint is met or violated
- If violated the replacement is immediately reverted back

How it works?

Algorithm 1: The Avatar algorithm

Input: $\mathbf{N}$: The netlist of the circuit, S : maximum delay allowed; P : maximum power allowed; $\mathbf{L}$: standard cell library

Output: $(\mathbf{N}_s, \mathbf{N}_f)$ slow and fast variants of $\mathbf{N}$

1 $\mathbf{N}_s \leftarrow \mathsf{AvatarSlow}(\mathbf{N}, \mathbf{L}, S)$

2 $P_{gain} = (2 \times \mathsf{Power}(\mathbf{N})) - \mathsf{Power}(\mathbf{N}_s)$

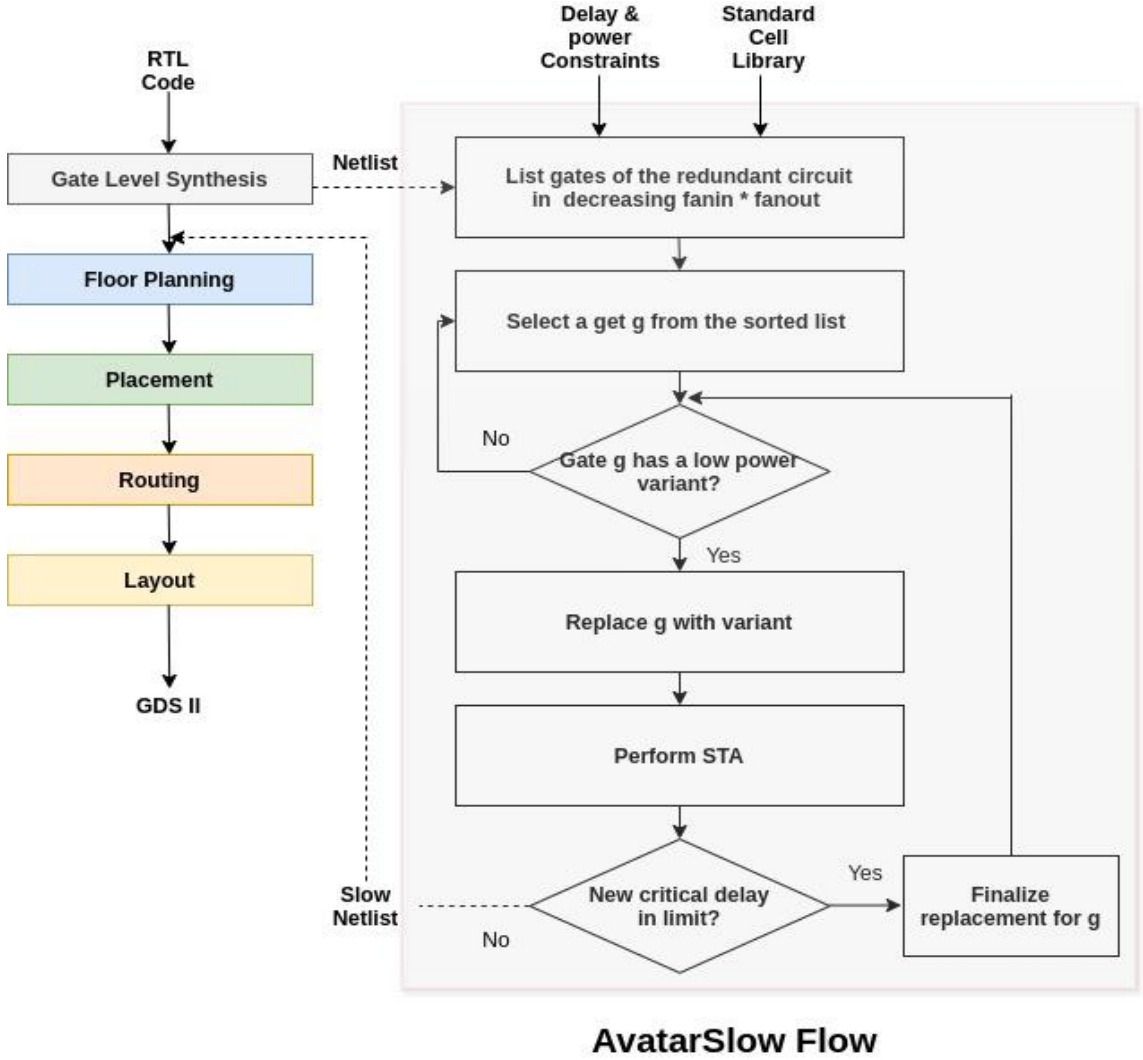
3 $P_{leeway} = P + P_{gain}$

4 $\mathbf{N}_f \leftarrow \mathsf{AvatarFast}(\mathbf{N}, \mathbf{L}, P_{leeway})$

5 **Return** $(\mathbf{N}_s, \mathbf{N}_f)$

AvatarSlow - How it works?

- AvatarSlow reconfigures gate to make the redundant circuit slow
- It makes sure the delay constraint is not violated
- Records the change in power caused by the replacement
- Any reduction in leakage power can be used up by AvatarFast



AvatarSlow Flow

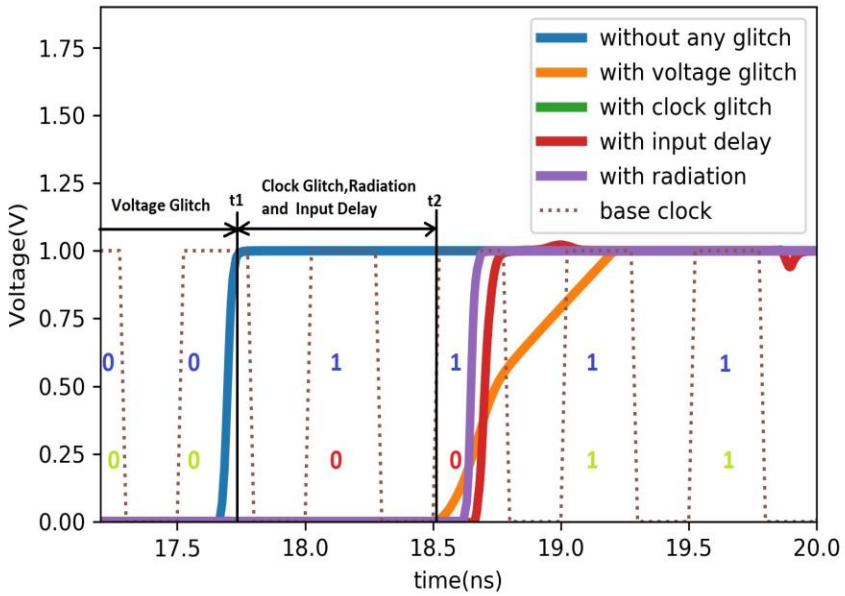
Results and Inferences

- Avatar works better on large block ciphers because it gets to tune more gates
- SPN ciphers with inherent higher diffusion properties are best matched with Avatar to detect faults

	Simon (64/32)bit	PRESENT 64-bit	CLEFIA 128-bit	AES 128-bit
Critical Delay(ns)	0.4 (0.3)	0.7 (0.5)	1.7 (1.3)	1.1 (0.7)
Power(μW)	2.2 (2.2)	23 (23)	119 (119)	194 (197)
ΔArea Utilization	7%	7%	4.3%	3%
DFA & DFIA Detection	81% (77.4%)	84.2% (79.1%)	94% (81.8%)	95% (55%)
SIFA Vulnerability	2.9 (3.4)	4.5 (4.6)	2.81 (4.5)	0.01 (1.2)
Runtime (hours)	0.05	5	23	92

Avatar against Voltage Glitch and Ion Beam Attacks

- **Voltage Glitch**
  - A sudden glitch in the voltage can lead to delayed signals from gate
  - A delayed arrival can lead to a setup time violation
  - **Ion Beam**
  - Ion Beam used to inject precise faults in the circuits
  - Focused ion beams can cause bit flips resulting in erroneous output
  - **The effect of a voltage glitch can be recreated using a clock glitch vice versa**
  - **The effect of a ion beam can be recreated using a clock glitch vice versa**



Summary

- Double redundancy used to protect against Fault Injection Attacks doesn't work as expected
- Bias in occurance of some faults over the other is responsible
- Attacks like Differential Fault Analysis, Statistical Ineffective Fault Analysis becomes easy
- Avatar is the solution
- Avatar analyses the relation between Threshold Voltage, Supply Voltage, Gate size vs Delay
- Difference in Arrival Time of signal increases probability of faults getting detected
- Avatar reconfigures the gates of main and redundant circuit such that one circuit becomes fast and the other one becomes slow so that their behaviour to similar faults are maximally different