Power supply glitch attacks: design and evaluation of detection circuits

Kamil Gomina*†, Jean-Baptiste Rigaud†, Philippe Gendrier*, Philippe Candelier* and Assia Tria†
*STMicroelectronics Crolles, France Email: name.surname@st.com
†Ecole Nationale Supérieure des Mines de Saint-Etienne,
Gardanne, France Email: surname@emse.fr

***Abstract*—Techniques using modification of power supplies to attack circuits do not require strong expertise or expensive equipment. Supply voltage glitches are then a serious threat to the security of electronic devices. In this paper, mechanisms involved during such attacks are analyzed and described. It is shown that timing properties of logic gates are very sensitive to power glitches and can be used to inject faults. For this reason, detection circuits which monitor timing properties of dedicated paths are designed to detect glitch attacks. To validate these solutions, a new approach based on the study of propagation delay variation is also presented. Following this approach, the performance of detection circuits can be evaluated at design level using a standard digital design flow.**

I. Introduction

Over the past few years, security of integrated circuits has been compromised by physical attacks. The aim is to retrieve sensitive information that are processed in the circuit. These attacks can be classified in three categories: invasive, semi-invasive and non-invasive [1]. Among these attacks, non-invasive are the most accessible since they do not require expensive equipment and there are no physical damages to the chip. A non-invasive attack method consists in inducing a fault by modifying the circuit primary inputs: clock, supply voltage, etc.

Supply voltage glitches were investigated as an injection method in [2]. CMOS gates were found vulnerable to negative spikes on the power supply. In addition, it is shown in their experiments that the injection mechanism of power supply glitch was the same as clock glitch attacks. First they injected faults using clock glitch attack by reducing one specific period, then they synchronized a power glitch on this same clock cycle. They obtained the same faults with these two injection methods in 70% of cases. In this paper, different fault injection mechanisms due to power glitches are considered. It makes possible the design of countermeasure whatever the injection mechanism. To thwart power glitch attacks, our contribution is the design of detection circuits and their evaluation using frequency variation. This new approach allows designers to test the efficiency of the detection circuits at simulation level.

Recent works addressed positive glitches and their detection [3]. Here we consider only supply voltage glitches under the nominal voltage. The remainder of the paper is organized as follows: section II gives the background regarding timing paths in a synchronous circuit. The next section presents the sensitivity of logic gates towards power glitches. Section IV defines the constraints necessary for a proper detection and the last section presents the detection circuits and the methodology of characterization.

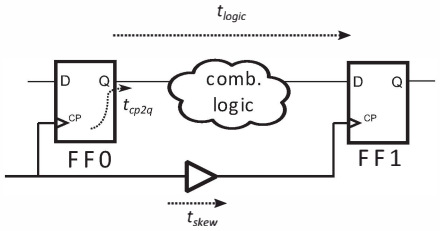


Fig. 1: Timing constraints for a digital path

II. Background: Timing based fault injection

This section reminds the timing paths in a synchronous circuit and the influence of propagation times on the circuit operation.

A. Timing requirements

Synchronous circuits operate on clock edges (rising or rarely falling edges) to validate computed data. Between two clock edges, data go through a set of combinatorial gates and must be stable an amount of time before the next clock edge (**setup time**). Likewise, the data must remain stable a short time after the next cycle, known as the **hold time**. To ensure a correct operation, every path between 2 registers must follow the timing constraints given by (1) and (2).

$$T > t_{cp2q} - t_{skew} + t_{logic} + t_{setup} \quad (1)$$

$$t_{hold} < t_{cp2q} - t_{skew} + t_{logic} \quad (2)$$

T is the clock period, t_{cp2q} is the time from the clock to the output inside the *flip-flop* (FF), t_{logic} is the delay through the combinatorial logic, t_{skew} is the time difference between the clock edge arrival at the 2 FFs, t_{setup} and t_{hold} are related to the endpoint FF (fig. 1).

B. Modification of propagation time

For both timing equations (1) and (2), a variation of t_{logic} can lead to a timing violation. The supply voltage is a parameter which modifies the propagation delay of the logic gates. The propagation through the logic directly depends on

978-1-4799-4112-4/14/$31.00

the supply voltage as presented in [4] with the propagation delay through a CMOS inverter:

$$t_{prop.r} = \frac{C_L \left[\frac{2\,|V_{th,p}|}{V_{DD} - |V_{th,p}|} + ln\left(3 - 4\frac{|V_{th,p}|}{V_{DD}}\right)\right]}{\mu_p C_{ox}(W_p/L_p)\,(V_{DD} - |V_{th,p}|)} \quad (3)$$

$t_{prop.r}$ is the rising propagation delay, C_L the output load capacitance, $V_{th,p}$ the PMOS threshold voltage, V_{DD} the supply voltage, μ_p the holes mobility, C_{ox} the gate oxide capacitance and W_p/L_p the width length ratio of the PMOS. For the falling propagation delay, $V_{th,p}$, μ_p and W_p/L_p are respectively replaced by $V_{th,n}$, μ_n and W_n/L_n which are the same parameters for the NMOS of the inverter.

By considering $|V_{th}| = 0.2 \cdot V_{DD}$ in (3):

$$t_{prop.r} = \frac{1.6 \cdot C_L}{\mu_p V_{DD} C_{ox}(W_p/L_p)} \quad (4)$$

We can deduce from (4) that the propagation time is inversely proportional to the power supply. Lowering the supply voltage results in higher delays and eventually (1) is no longer met. This method was previously used to induce setup time violation on an AES implementation [5], [6].

Supply voltage is a primary input of a circuit which is fundamental for its proper operating. Indeed, it could modify parameters of (1) and (2) and leads to a system failure. For these reasons, voltage variations effects on a circuit are investigated.

III. Logic sensitivity to power glitches

The first effect of a supply voltage variation is the modification of timing properties of logic gates especially the propagation delay if we consider (3). However, other phenomenons could occur due to the presence of a glitch on the supply rails.

A. Combinatorial logic

1) Propagation of a glitch on power rails: Let us consider a negative glitch on V_{DD} which propagates through the power grid as presented in fig. 2. When the glitch occurs on V_{DD1}, it take a short amount of time to propagate to V_{DD2} due to the parasitic resistance and capacitance of the power grid. As a result, when V_{DD1} is low and V_{DD2} still high (during t_{fault}), a logic 1 of the 1st stage could be read as a logic 0 for the next logic stage. If the next logic stage is a memory element (D-latch, FF etc.) this erroneous value can be latched resulting in a fault injection. Note that the results are the same if V_{DD1} and V_{DD2} are two different power domains and the negative glitch is applied on V_{DD1} only. This configuration is found in advanced technology nodes. The multiplication of power domains increases the complexity of analysis for security purposes.

2) Glitches on combinatorial logic output: If we consider a FF which keeps its state during two consecutive clock cycles two cases can occur: either its input D stays at its level all over the cycle or a glitch appears between two clock edges. This glitch can be caused by a difference in propagation delays of several paths. Let us see an example: fig. 3a shows a path between two FFs. The NAND gate drives two sub-paths noted

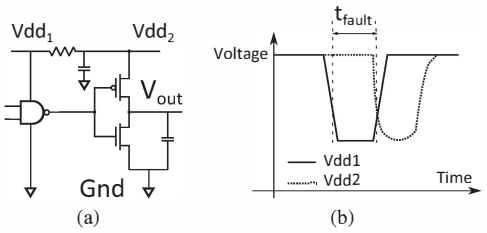


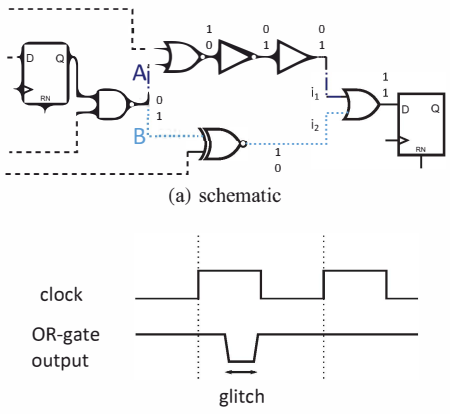
Fig. 2: Propagation of a power glitch

A and B, then these paths are recombined to drive the FF input. Gates output values are indicated for two consecutive clock cycles. For the path A, the OR-gate input i_1 switches from 1 to 0 and the opposite for path B (input i_2). As a result the OR-gate output final value is 1 for these two cycles. However, path B may be faster than path A as it is composed of a single gate. When the OR-gate input i_2 switches from 1 to 0, the gate output is low until the new data arrives at i_1. A glitch appears for a short period of time (fig. 3b).

This situation is likely to happen for all combinatorial gates as soon as both gate inputs are changing and the output state is the same as the previous one. The modification of the propagation delay due to a negative glitch would eventually move this glitch to the setup time of the next FF and induce a timing violation.

B. Sequential logic sensitivity

The behavior of a sequential element toward negative voltage glitches was presented in [7]. A D-latch which is a bi-stable element composed of two inverters driving each-over is analyzed in presence of different negative glitches. They demonstrate that a negative glitch cannot toggle the value of a D-latch in a locked state. The voltage variation on one node of the bi-stable is always smaller than the variation on the other node so the glitch causes a transient variation on both nodes. Actually, the value stored in a bi-stable element can be changed only when V_{DD} becomes negative. This case will not be considered here.



(a) schematic

(b) timing diagram

Fig. 3: Example of glitch at the flip-flop input

IV. Timing detection

A. Principle

We want to design a circuit which is able to detect any timing violation due to modifications of the operating conditions of logic cells. Two kinds of detection can be considered: the violation is detected within the same clock cycle or it can be periodic (test over several cycles), the detection may occur few cycles after the violation. In case of an attack, the detection must occur as soon as possible to prevent the attacker from exploiting his injections. The detection should happen as soon as a path in the circuit is violated if there is no margin for the detector. Otherwise, the detection should occur before a violation. The time difference between the critical path and the detector path is defined as the margin.

B. Margin for timing detection

One of the difficulties to detect a timing violation on a circuit is to calibrate the detector to ensure a sufficient margin. The margin is the time difference between the circuit critical path and the detector path as shown in fig. 4. Indeed in the timing equations, setup and hold times are also voltage dependent, as a result, the margin could change according to the supply voltage. t_{setup} and t_{hold} define the uncertainty time windows in which the FF output is not deterministic [8]. When a timing violation occurs in the circuit, we have to verify that the detector is not anymore in the uncertainty window.

Let us consider a path of the circuit to protect and the detector path. We suppose that the flip-flops at the endpoint for theses two paths are identical (same t_{setup}, t_{hold} and t_{cp2q}). Figure 4 describes this situation: here the circuit path as well as the detector path meet the timing constraint defined by (1). However, the detector delay is greater than the circuit one as expected. In case of a negative voltage glitch, the propagation time will be longer. By making the assumption that the delay reduction is uniform, the detector will be in the violation window first. Let us denote $t_{s\_h} = t_{setup} + t_{hold}$. Here the margin is greater than t_{s_h} so by the time the circuit path enters the violation window, the detector path will have been beyond this window therefore the violation will be detected. Formally, the constraint to respect is given by (5). In this equation, t^c are the timings related to the circuit to protect and t^d are related to the detector.

$$\begin{aligned} t_{det} &= t^d_{cp2q} + t^d_{logic} - t^d_{skew} \\ t_{cir} &= t^c_{cp2q} + t^c_{logic} - t^c_{skew} \\ t_{det} - t_{cir} &> t_{hold} + t_{setup} = t_{s\_h} \end{aligned} \tag{5}$$

Usually different types of flip-flops are used within the same circuit so the setup and the hold time depends on each path. In this case the greatest setup and hold time must be chosen to check the margin. A particular case is when the hold time is negative. This corresponds to a time window before the rising clock edge. t_{s_h} is then equal to $max(t_{setup}, |t_{hold}|)$ and (5) is still valid. Also note that the constraint defined by (5) must be met whatever the operating conditions (process, voltage, temperature) to ensure a proper detection. To reduce the minimum required margin, the setup and hold time of the FF at the endpoint of every paths must be as small as possible.

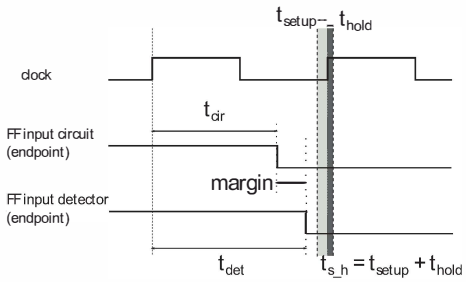


Fig. 4: Margin for secure detection

C. Defining circuit timing constraints

The detector timing calibration is based on the Static Timing Analysis (STA) performed after synthesis or after a complete placement and routing. However, without specific constraints, the timing distribution is not preserved from one routing to another making difficult the detector calibration. To do so, circuit timing constraints are defined to be able to set the margins (5) and the detector delay. The idea is to have a single critical path to monitor in order to detect a timing violation.

Thus, let us focus on the variation of the propagation delay of a gate according to the supply voltage. In the case of an inverter, the propagation delay is defined by (4). The variation of the propagation delay according to the supply voltage is given by:

$$\frac{\mathrm{d}(t_{prop})}{\mathrm{d}V_{DD}} = -\frac{1.6 \cdot C_L}{\mu C_{ox}(W/L){V_{DD}}^2} \tag{6}$$

Both the propagation time and its variation according to V_{DD} depend on the ratio C_L/W. Let us consider the reciprocal number of the propagation delay $f_{prop.} = 1/t_{prop.}$. In this case the variation of $f_{prop.}$ is given by (7), with $k = 1.6 \cdot L/(\mu \cdot C_{ox})$.

$$\frac{\mathrm{d}(f_{prop})}{\mathrm{d}V_{DD}} = \frac{\mu C_{ox}(W/L)}{1.6 \cdot C_L} = \frac{1}{k} \cdot W/C_L \tag{7}$$

f_{prop} is proportional to V_{DD} as a result the variation of f_{prop} is constant and depends on the ratio W/C_L. In the case of a path of N inverters, let f_{path} be the reciprocal of the propagation delay, f'_{path} its derivative, C_i and W_i the output load and the transistor width of inverter i $(1 \leq i \leq N)$. Note that W_i could be either $W_{n,i}$ or $W_{p,i}$ depending on which delay is considered (rising of falling). Thus:

$$f_{path} = \frac{1}{\sum_{i=1}^{N} \frac{k}{V_{DD}} \frac{C_i}{W_i}} \tag{8}$$

$$\begin{aligned} f'_{path} &= \frac{\mathrm{d}}{\mathrm{d}V_{DD}} \left(\frac{1}{\frac{k}{V_{DD}} (C_1/W_1 + \cdots + C_N/W_N)} \right) \\ &= \frac{1}{k} \frac{1}{\sum_{i=1}^{N} \frac{C_i}{W_i}} \end{aligned} \tag{9}$$

Since each term of the denominator of (9) is positive, the more cells, the smaller the derivative. As the slope of the curve

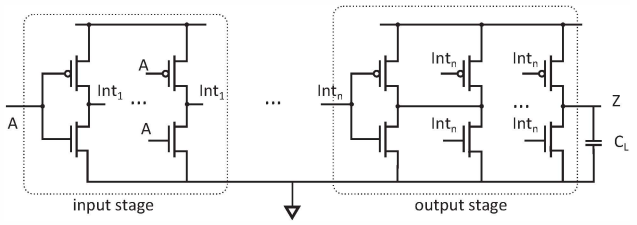


Fig. 5: Input and output stage of a gate

representing f_{path} vs. V_{DD} is the derivative defined by (9), two results can be given:

- the slope (f'_{path}) decreases with the number of cells.
- a faster path has a greater slope.

The previous example with a path composed of inverters is a specific case because for these gates the input stage is also the output stage. The other gates are composed of at least two stages: an input and an output stage (fig. 5). Actually, the propagation delay of a gate is the sum of each stage propagation time. The frequency and its derivative depends on the ratio W/C of each stage where C is the next stage input capacitance and W is the transistor width of the current stage (either W_p or W_n). Finally, the results obtained for a path of inverters can be extended to a path composed of any kind of gate. Two parameters are available to set a path delay : increasing the number of gate or modifying f'_{path}. Depending on the circuit to protect, it is not always possible to increase the number of gates because of the minimum operating frequency. In this case, the designers can modify f'_{path} instead by choosing the gate that compose the path to ensure the detection. This method enables more fine-tuning. According to this analysis, f_{path} will be considered in the following section to evaluate its variation according to voltage (f'_{path}).

V. Timing detection circuits

A. Related work

Many systems have been suggested to detect timing errors due to static (process) or dynamic (temperature, voltage, aging etc.) variations.

Razor I [9] introduces in each critical path a shadow latch. The latch sampling occurs at the falling clock edge such that the shadow latch has additional time to hold the correct value. An error is detected when the main FF output differs from the shadow latch output. This solution requires to control the clock duty cycle, a metastability detector and delayed clock which increases design complexity and implementation. Razor II [10] is an improvement of the first one. A transition detector indicates if the data arrive late after the rising clock edge. Razor II does not require a metastability detector however, it requires the control of the clock duty cycle too.

Canary FF [11] also introduces a shadow FF with a delay element to have a longer delay than the functional path. The main FF and the canary FF output are compared to detect a timing error. This technique has some advantages over razor FF: the delayed clock is not required and the violation should occur in the canary FF first, thus protecting the main FF.

In [12], [13] the authors presented two Error Detecting Sequential (EDS) circuits. The first circuit detects a late transition of the input data and triggers an error signal. The second one uses an additional master-slave FF to sample the data at two different time and the output is compared to detect an error. Both these circuits use a latch instead of a master-slave FF in the data path. The possible metastability of the output is handled by the error signal path. A tunable replica circuit (TRC) is suggested in [13], [14]. A delay tree which is longer than the circuit critical path is used to detect a timing error. This method requires post-silicon calibration increasing the test time.

The other solutions integrate a sensor to detect data transition in an amount of time before the setup time [15], [16]. They used a smaller number of transistors for the edge detector in comparison with razor I or II solutions.

All these methods are suitable for detecting progressive variation but not for voltage attacks which are unpredictable and can involve large modifications of the supply voltage.

B. Circuit description

The detection circuits below are designed using the same process: the gates used to set the delay are chosen to optimize f'_{path}.

1) Parallel delay lines (PDL): The idea is to use multiple paths to determine the evolution of the delay in the circuit to monitor. Each path of the detector has a different propagation delay: the first path is the fastest one and the last is the slowest one as shown in fig. 6. The slowest path is designed to be the longest reached within the given specifications (for instance a slow process, and the lowest voltage and temperature). As a result if the state of FF1 differs from FFN, timing violation might occur in FFN and an error is triggered. This method requires at least two paths, one for the fastest path and another for the slowest one. However, with several path it is possible to evaluate the delay variations by observing the output of the set of FF. Obviously path N must have a greater delay than the critical path of the circuit to be violated first. Choosing the margin according to (5) is required to avoid a metastability detector : FFN output will be already stable when the FF of the critical path becomes metastable.

2) Tunable replica circuit (TRC): TRC uses a new path in the circuit which combinational logic is composed of a delay tree. The logic made by series of buffers or inverters is set to be the new critical path of circuit to protect. However, this path must follow (1) and (2). An example of TRC is given in fig. 7. Depending on the selectors, it is possible to fine-tune the delay post-silicon. The main advantage of this

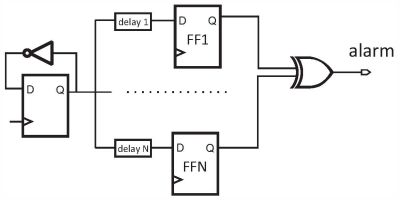


Fig. 6: Parallel delays lines

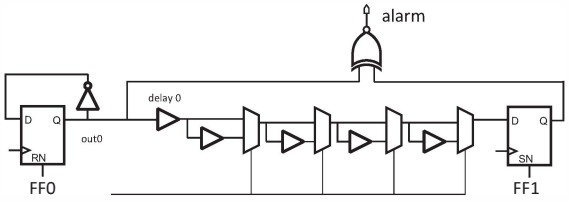


Fig. 7: Tunable replica circuit with 4 multiplexers

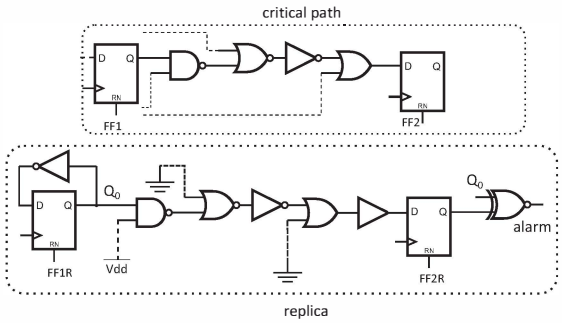


Fig. 8: Critical path replica

circuit is the delay reconfiguration which could be used to cover unexpected variations after manufacturing. The main difference with the TRC in [13] is the buffers sizing to set f'_{path}. Figure 7 describes the operating of the circuit in case of longer propagation delays. When there is no attack, FF0 and FF1 output are inverted. If the data arrive late at FF1 input, its output does not change until the rising clock edge as a result FF0 and FF1 are equal triggering the alarm.

3) Critical path replica (CPR): This method is based on the introduction of a new path with placement and timing constraints. The principle is to duplicate the path including the startpoint and the endpoint FF to monitor and detect a violation on this duplicated path. The same logic cells are used to recreate the exact same path by respecting each cell fan-out as much as possible. In addition, this path needs to be always active: each cell input pin is connected to the ground or the power supply to make the path active. Figure 8 shows an example of critical path and its replica. The startpoint FF provides a toggling signal which propagates through a replica of the combinatorial logic composing the critical path.

The critical path replica must be the first path to violate the timing constraints in case of attacks. Therefore its propagation delay should be slower than the original critical path. Several ways to make the replica slower can be considered. The fan-out of one or more gate can be increased. Another possibility is to add a delay element in the combinatorial logic. It can be a series of buffers or inverters. Here we decided to add a delay element because it is the simplest way to add a accurate delay to ensure the proper margins.

C. Simulation results

To evaluate these detectors, they are integrated in an AES circuit. Each detector is set to trigger an alarm when its timing constraint is violated. The circuit under test is designed on 28 nm bulk CMOS technology. The characteristics of the circuit are given in table I. PDL and TRC presented in the previous section are implemented to reduce their size and to be able to compare all of them consistently. Thus, there are only two lines in the parallel delay line detector and four multiplexers in the TRC. The replica is already at a minimum size since it is composed of a single path.

TABLE I: Circuit under test characteristics

Area	0.14 mm^2
Gate count	15000
Operating freq.	100 MHz
Technology	CMOS 28 nm

Simulations of supply voltage glitches require to modify dynamically the circuit power supply which is not possible using usual back-annotated simulation flow. Indeed, back-annotated simulations are based on timings measured at a static supply voltage. This is why we will focus on the variation of propagation delays in addition to static timings. Figure 9 presents the results given by the different detectors. The maximum frequency (the reciprocal of the propagation time) of the critical path as well as the timing detection circuits are reported for static voltages from 0.6 to 1 V. First of all, the critical path delay is less than the entire set of detectors whatever the supply voltage. Then, we can also compare the frequency variation of all these solutions. Frequency variation according to voltage is given by the slope of each straight line (table II). The critical path has greatest frequency variation followed by the critical path replica, the parallel delay lines and lastly the TRC. By reporting both the propagation delays and the frequency variation, we ensure that for any variation in the supply voltage (amplitude and width), the combinatorial path of the detectors are slower than those of the circuit, especially the critical path. Indeed, each detector max. frequency moves along the straight line defining its variation according to the voltage in fig. 9. As a result, if the curves do not intersect, the max. frequency rankings are not modified and the detector path which has the lowest max. frequency violates (1) first. Finally, by considering both frequency and its variation, any solution presented here detects a violation before the circuit timing failure.

Among the detectors, the critical path replica is the one which has the closest max. frequency and variation to the critical path. Note that the replica delay can be even closer to the critical path but this difference is set to ensure the margin constraints. The other solutions have greater detection thresholds, therefore, they are more prone to false positive

TABLE II: Detectors frequency variation

Circuit	Slope (GHz.V^{-1})
TRC	0.4241
PDL	0.4672
CPR	0.5180
C. path	0.5948

TABLE III: Margins for the replica based timing detector

Voltage (V)	Required margin (ns)	Reported margin (ns)
0.60	3.69	11.36
0.85	0.27	1.03
0.90	0.21	0.83
0.95	0.17	0.68
1.00	0.15	0.58

TABLE IV: Comparison of the detection circuits

Circuit	PDL	CPR	TRC
Area	medium	low	medium
design complexity	low	high	low
Post Si calibration	no	not required	tunable delay
Delay control (CAD)	medium	high	low

detection. Table III indicates the required margins (t_{s_h} in section IV-B) for the critical path replica at different supply voltages. These margins enable to take into account the minimum required value defined by (5), and on the other hand, some delay variability after manufacturing.

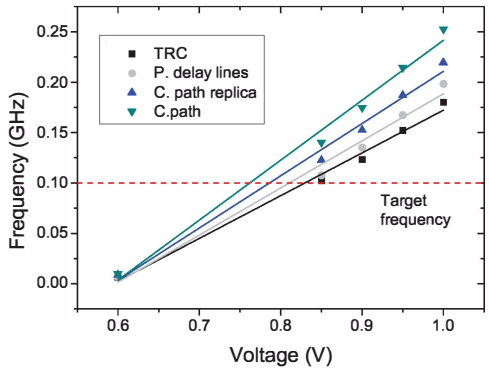


Fig. 9: Detector characterization using frequency variation

All the previous solutions are able to detect a supply voltage glitch under the nominal voltage. The comparison of these solution is given in table IV. As the CPR is composed of a single path, it requires less area than the others. But its complexity is greater because the critical path must be recreated using the same cells, it requires many steps and scripts for automation. Regarding the delay control, the PDL needs to be finely tuned at CAD level while the accuracy of the CPR is based on replica method. For the latter, margin is set by adding extra cell. On the other hand, the TRC can minimize the effort of tuning delays at CAD level since tuning can be done on silicon. To summarize, the CPR requires much effort on design to be efficient, whereas the TRC needs less effort at CAD level. The PDL effort is between these two but requires to have accurate timing models. Finely CPR appears as a solution with high accuracy and a low overhead.

VI. Conclusion

This work presented the vulnerability of CMOS circuits to negative power glitch attacks. Based on logic cell sensitivity, three detection circuits were designed. Our approach was to investigate the variation of the path frequency to define the gate combination implemented in the detection circuits. It points out that this variation is related to ratio between the transistor width W and the output capacitance C_L. The validation process considers the frequency and its variation to make sure that no functional paths violate their timing constraints before the detectors. The critical path replica seems to be an efficient solution to detect glitch attacks. A complete characterization of these solutions on silicon are ongoing. The results will be presented in a further work.

References

[1] S. P. Skorobogatov, "Semi-invasive attacks-a new approach to hardware security analysis," *Technical report, University of Cambridge, Computer Laboratory*, 2005.

[2] L. Zussa, J.-M. Dutertre, J. Clediere, and A. Tria, "Power supply glitch induced faults on FPGA: an in-depth analysis of the injection mechanism," in *On-Line Testing Symposium (IOLTS), 2013 IEEE 19th International*, pp. 110–115, 2013.

[3] K. Gomina, P. Gendrier, P. Candelier, J.-B. Rigaud, and A. Tria, "Detecting positive voltage attacks on cmos circuits," in *Proceedings of the First Workshop on Cryptography and Security in Computing Systems*, CS2 '14, (New York, NY, USA), pp. 1–6, ACM, 2014.

[4] L. Zussa, J.-M. Dutertre, J. Clédière, B. Robisson, and A. Tria, "Investigation of timing constraints violation as a fault injection means," in *27th Conference on Design of Circuits and Integrated Systems (DCIS)*, (Avignon, France), 2012.

[5] N. Selmane, S. Guilley, and J. L. Danger, "Practical setup time violation attacks on AES," in *Dependable Computing Conference, 2008. EDCC Seventh European*.

[6] S. Bhasin, N. Selmane, S. Guilley, and J. L. Danger, "Security evaluation of different AES implementations against practical setup time violation attacks in FPGAs," in *HOST '09*, pp. 15–21, July.

[7] A. Djellid-Ouar, G. Cathebras, and F. Bancel, "Supply voltage glitches effects on CMOS circuits," in *International Conference on Design and Test of Integrated Systems in Nanoscale Technology DTIS*, pp. 257 –261, Sept. 2006.

[8] J. Horstmann, H. Eichel, and R. Coates, "Metastability behavior of CMOS ASIC flip-flops in theory and test," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 1, pp. 146–157, 1989.

[9] S. Das, S. Pant, D. Roberts, S. Lee, D. Blaauw, T. Austin, T. Mudge, and K. Flautner, "A self-tuning DVS processor using delay-error detection and correction," in *VLSI Circuits, 2005. Digest of Technical Papers. 2005 Symposium on*, pp. 258–261, 2005.

[10] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. Bull, and D. Blaauw, "RazorII: in situ error detection and correction for PVT and SER tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, 2009.

[11] Y. Kunitake, T. Sato, H. Yasuura, and T. Hayashida, "Possibilities to miss predicting timing errors in canary flip-flops," in *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1–4, 2011.

[12] K. Bowman, J. Tschanz, N. S. Kim, J. Lee, C. Wilkerson, S. Lu, T. Karnik, and V. De, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 49–63, 2009.

[13] K. Bowman, J. Tschanz, S. Lu, P. Aseron, M. Khellah, A. Raychowdhury, B. Geuskens, C. Tokunaga, C. Wilkerson, T. Karnik, and V. De, "A 45 nm resilient microprocessor core for dynamic variation tolerance," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 194–208, 2011.

[14] A. Raychowdhury, J. Tschanz, K. Bowman, S.-L. Lu, P. Aseron, M. Khellah, B. Geuskens, C. Tokunaga, C. Wilkerson, T. Karnik, and V. De, "Error detection and correction in microprocessor core and memory due to fast dynamic voltage droops," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 208–217, 2011.

[15] B. Rebaud, M. Belleville, E. Beigne, M. Robert, P. Maurine, and N. Azemard, "On-chip timing slack monitoring," in *2009 17th IFIP International Conference on Very Large Scale Integration (VLSI-SoC)*, pp. 89–94, 2009.

[16] B. Das and H. Onodera, "Warning prediction sequential for transient error prevention," in *2010 IEEE 25th International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, pp. 382–390, 2010.