HAL
open science

Supply Voltage Glitches Effects on CMOS Circuits

Anissa Djellid-Ouar, Guy Cathébras, Frédéric Bancel

To cite this version:

Anissa Djellid-Ouar, Guy Cathébras, Frédéric Bancel. Supply Voltage Glitches Effects on CMOS Circuits. DTIS'06: Design and Test of Integrated Systems in Nanoscale Technology, Sep 2006, Tunis (Tunisia), pp.257-261. lirmm-00093365

HAL Id: lirmm-00093365

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00093365v1

Submitted on 13 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Supply voltage glitches effects on CMOS circuits

Anissa Djellid-Ouar*†, Guy Cathebras* and Frédéric Bancel†
*LIRMM - UMR5506
161, rue Ada
34492 Montpellier Cedex 5 – France
Email: Anissa.Djellid@lirmm.fr, Guy.Cathebras@lirmm.fr
†STMicroelectronics / Smartcard division
ZI de Rousset BP 2
13106 Rousset Cedex – France
Email: djellid-ouar.lirmm@st.com, frederic.bancel@st.com

***Abstract*— Among the attacks applied on secure circuits, fault injection techniques consist in the use of a combination of environmental conditions that induce computational errors in the chip that can leak protected informations. The purpose of our study is to build an accurate model able to describe the behaviour of CMOS circuits in presence of deliberated short supply voltage variations. This behaviour depends strongly on the basic gates (combinational logic, registers...) that make up the circuit. In this paper, we show why D-flip-flop are resistant to power supply glitches occurring between clock transitions and we propose an approach to evaluate the basic elements sensitivities towards faults generated by power glitches. Our aimed model will consequently be dependent on this sensitivity.**

I. Introduction

In the field of secure systems like smartcard, industrialists unceasingly improve their security mechanisms while the hackers tirelessly try to thwart them. IC' manufacturers must protect chips against security threats. Since 1998, a continuous battle is waged between manufacturers and the hackers' community. Once a new product is launched, hackers want to break it using different methods. Among them, fault injection techniques remain the less expensive ones and do not require specific equipment.

It is important to anticipate and understand as much as possible what could happen to a chip in presence of fault injection in order to find the appropriate countermeasures. The effect of faults on electronic systems has been studied since the 1970s when it was noticed that radioactive particles caused errors in chips. This led to further research on the effect of charged particles on silicon, motivated by the aerospace industry who was becoming concerned about the effect of faults in airborne electronic systems. Since then various techniques for fault creation and propagation have been discovered and researched. These techniques were firstly dedicated for testing systems' fault tolerance. When used as an attack strategy, fault injections are conduced using a combination of environmental conditions that cause a chip to produce computational errors that can leak protected data or allow an access to protected areas. One of the most advanced cracking techniques, known as the differential fault analysis (DFA), is to perturb the chip operations by taking advantage of a possible correlation between the erroneous and the correct responses. Differential Fault Analysis has been introduced by Boneh *et al.* [1] in 1997. They showed, using two cryptographic messages, one correct and one erroneous in which a fault has been introduced, that it is possible to find the cryptosystem secret key. Various mechanisms for fault creation and propagation are proposed [2], such as variations of the voltage supplied to the chip, of the clock frequency, of the temperature, etc. (Cf. [3], [4]) In this paper, the injection technique we have worked on is the glitch voltage corresponding to a transient variations of the supply voltage (V_{dd}). A glitch can be defined by many parameters among which we can name its shape, its falling and rising slopes, its low and high levels, its duration. There are already fault models describing local events (due to laser beam, heavy ion...) that may propagate throughout a circuit or a part of it [5]–[7]. In our study, we work on a global phenomenon of which effects have impacts on the entire circuit. Our objective is to construct a relevant fault model coming from analyzing the standard cells behaviour of the STMicroelectronics HCMOSM8 library ($0.18\,\mu\mathrm{m}$ technology + flash). The first results show that the most sensitive logic paths (vulnerable architectures) could be identified. The model should thus make it possible to anticipate the effect of a supply voltage glitch (its propagation) in the circuit according to the sensitivity of the standard elements. On the other hand, this fault model could be used to develop standard-cells based countermeasures.

II. Faults, errors and failures

Any biological or manufactured structured system tends to malfunction either because it is badly designed or because it is deteriorated. The *failure* of a system is the result of a temporary or permanent dysfunction. An *error* is the manifestation of a *fault* and can lead to the total or partial failure of the

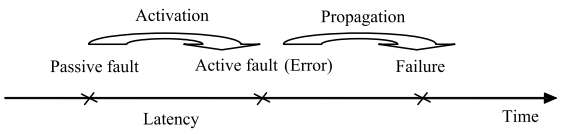


Fig. 1. From fault to failure

0-7803-9727-4/06/$20.00 ©2006 IEEE

system, if it is propagated to its outputs [8]. This mechanism is described on figure 1.

III. Flip flop behaviour

According to CMOS ICs architecture, master-slave D-flip-flops are the main part of the circuit. Gathered into registers, they are delimiting combinational logic blocks. At each end of a logic cone, we can find the output or the input of a D-flip-flop. Our objective is to analyse the intrinsic behaviour of a flip-flop and logic that surrounds it, when applying power supply glitches.

The flip-flop functionality is to memorize data for the time necessary to its treatment (one clock period). The memorized values are an image of the logical state of the circuit at a given moment. Thus if one of these values is corrupted, then the circuit moves to an erroneous logical state.

The flip-flop vulnerability towards any type of perturbation can have two origins [5], [6]:

- the logic that surrounds it;
- the internal D–latches that make it up.

Let us have a look at each of these points.

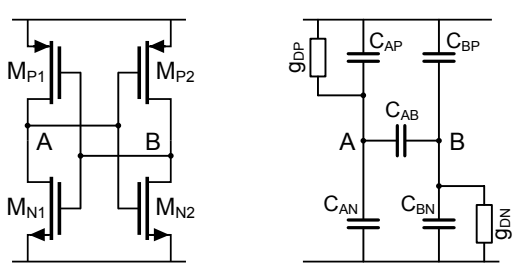
A. Combinatorial logic sensitivity

When a power glitch occurs, it has two effects on a combinatorial gate. First, the change on the supply voltage induces a modification of the timing properties of the gate, delay and output slope for instance. Of course, these timing properties modifications can only be observed if a transition occurs during the glitch. The second effect is relative to the propagation of the glitch transitions towards the output (and possibly the input) of the gate. As a matter of fact, this doesn't affect really the combinatorial logic behaviour. Indeed, this is typically a transient effect that vanishes rapidly due to the limited bandwidth of the logic gates. Only remains some kind of delay that is indiscernible of the one induced by the change in supply voltage.

So, we will retain that the effect of a power supply glitch on combinatorial logic is a modification of its timing properties.

B. D-Latch sensitivity

A D-latch has two main states: locked or unlocked. When in the unlocked state, it performs like combinatorial logic. So, the effect of a power glitch occurring during the unlocked state will be on speed, but not on a stored value that does not exist at this time!



(a) CMOS Bistable

(b) Small signal equivalent circuit

Fig. 2. Static CMOS memory point and its small signal equivalent circuit when $V_A = V_{dd}$ and $V_B = 0$. Components values are given in the text.

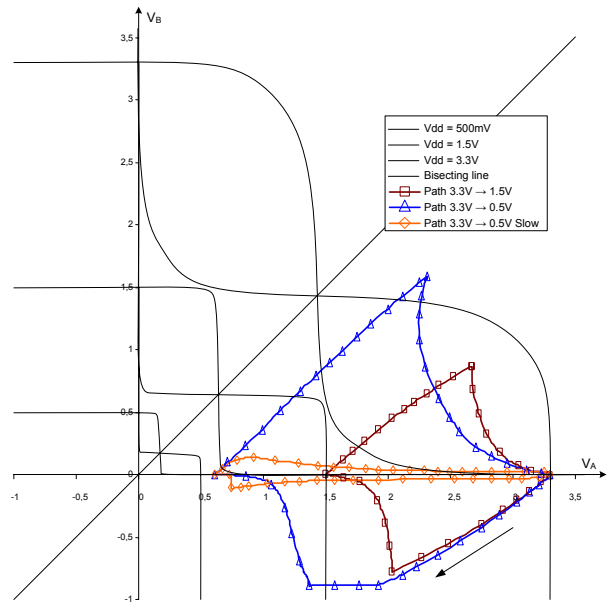


Fig. 3. CMOS bistable's operating point locus

On the other hand, the analyse of the effect of a power glitch occurring during the locked state is a bit more tricky. Figure 2(a) shows the heart of a locked D-latch: it is a CMOS bistable made of two inverters forming a loop. The bistable state is coded by the V_A and V_B voltage values.

On figure 3 we have first represented, in the $(V_A V_B)$ plane, the static transfer function of each inverter[1] for three V_{dd} values: 3.3 V, 1.5 V and 0.5 V. Next, we plotted on the same figure the operating point locus ($V_B = f(V_A)$) for three different power glitches. The first (square ticks) is an abrupt (10 ps fall and rise time) variation of the supply from $V_{dd} = 3.3\,\mathrm{V}$ to $V_1 = 1.5\,\mathrm{V}$ and return (Cf. left part of figure 5). For the second (triangular ticks), the amplitude was increased in order to reach $V_1 = 0.5\,\mathrm{V}$. Last, the third (diamond ticks) looks like the second but with slower falling and rising edges: $T_r = T_f = 1\,\mathrm{ns}$. In the three cases, the operating point starts from $(V_A = V_{dd}, V_B = 0)$ and moves along its locus in the direction indicated by the arrow.

Looking at these loci, we can say that none of these power glitches is able to tilt the bistable. Indeed, to have a toggle of the bistable, the operating point must cross the first bisecting line of the $(V_A V_B)$ plane. Let us prove that these examples are not particular cases. We can distinguish two kinds of arcs on an operating point cycle:

- when the operating point leaves a stable point ($V_A = V_{dd}, V_B = 0$ for instance) it follows a quasi straight line roughly parallel to the first bisecting line;
- when the power supply is stable, the operating point moves towards the closest stable point.

[1] $0.35\,\mu\mathrm{m}$ CMOS technology

Since the topology of the CMOS bistable is symmetric, the first bisecting line is a symmetry axis and a ridge in the potential map of the system. Thus, the "natural" evolution of the operating point towards the closest stable point will never imply the crossing of the first bisecting line.

But what about the evolution of the operating point during the transitions of V_{dd}? Let us have a look at the small signal equivalent circuit of figure 2(b). The component values are:

$$C_{AN} = \frac{C}{2} \quad C_{BN} = C \quad C_{BP} = \beta\,\frac{C}{2} \quad C_{AB} = \frac{\beta+1}{2}\,C$$
$$C_{AP} = \beta\,C \quad g_{DP} = \gamma\,g \quad g_{DN} = g$$

where

$$C = C_{ox} W_n L \qquad g = K_n\,\frac{W_n}{L}\,(V_{dd} - V_{tn})$$
$$\beta = \frac{W_p}{W_n} \qquad \gamma = \beta\,\frac{K_p\,(V_{dd} - |V_{tp}|)}{K_n\,(V_{dd} - V_{tn})}$$

C is the gate capacitance of the N-channel transistors. It is distributed using the *Meyer* capacitance model ($C_{gb} = C$ for off state, $C_{gs} = C_{gd} = C/2$ for triode region and $C_{gs} = 2C/3$ for saturation). Assuming an identical length for N and P-channel transistors, β is the configuration ratio. It is typically chosen between 1 and 3. The parameters g and γg are respectively the conductance of the N and P-channel transistors in "on" state with $|V_{GS}| = V_{dd}$ and $V_{DS} = 0$ (triode region). These last values are derived from the *Schichman and Hodges* (spice level 1) model. Obviously, K_n and K_p are the transconductance parameters of the transistors, while V_{tn} and V_{tp} are their threshold voltages.

Using these notations, we can derive the transmittance from the supply to the A and B nodes:

$$\frac{V_A(s)}{V_{dd}(s)} = \frac{1 + \frac{2\beta+2\beta\gamma+3\gamma}{2\gamma}\frac{C}{g}s + \frac{5\beta^2+7\beta}{4\gamma}\frac{C^2}{g^2}s^2}{1 + \frac{2+3\beta+2\beta\gamma+3\gamma}{2\gamma}\frac{C}{g}s + \frac{5\beta^2+11\beta+5}{4\gamma}\frac{C^2}{g^2}s^2}$$

$$\frac{V_B(s)}{V_{dd}(s)} = \frac{\frac{2\beta+1}{2}\frac{C}{g}s + \frac{5\beta^2+4\beta}{4\gamma}\frac{C^2}{g^2}s^2}{1 + \frac{2+3\beta+2\beta\gamma+3\gamma}{2\gamma}\frac{C}{g}s + \frac{5\beta^2+11\beta+5}{4\gamma}\frac{C^2}{g^2}s^2}$$

From these transmittances, we are only interested by their limits when $s \to \infty$, since this give us the ratio between ΔV_A or ΔV_B and ΔV_{dd} for ideal rectangular glitches.

$$\frac{\Delta V_A}{\Delta V_{dd}} = \frac{5\beta^2 + 7\beta}{5\beta^2 + 11\beta + 5} \qquad \frac{\Delta V_B}{\Delta V_{dd}} = \frac{5\beta^2 + 4\beta}{5\beta^2 + 11\beta + 5}$$

We can see on table I that ΔV_A and ΔV_B are always smallers (in absolute value) than ΔV_{dd}. So, in theory, the falling edge of a negative power supply glitch could never toggle the bistable, since the operating point cannot reach the first bisecting line. As a matter of fact, there is a possibility to tilt the bistable: the drain bulk junction of the N-channel transistor prevents V_A and V_B to descend under 0.7 V (one can notice this by looking at figure 3). Therefore, a huge negative power supply glitch, larger than V_{dd}, but very short, should push the operating point beyond the first bisecting line, toggling the bistable. Nevertheless, this kind of power supply glitch is much more like a power supply interruption and it should certainly trigger the power-on reset structures of the circuit too. So, we will not consider this possibility.

Last, we must consider the rising edge of the negative glitch. In that case we have V_A and V_B growing simultaneously. Noticing that the ratio:

$$\frac{\Delta V_B}{\Delta V_A} = \frac{5\beta + 4}{5\beta + 7}$$

is always lower than one, we can affirm that the path of the operating point diverge from the first bisecting line. As a consequence, the operating point will not be pushed on the first bisecting line by the rising edge of the glitch.

C. Conclusion

We have shown here that a reasonable power supply glitch (whose amplitude doesn't exceed the permanent supply voltage) cannot tilt a locked D-latch. Since a D-flip-flop contains two D-latches among which one is locked between the clock transitions, it appears that a D-flip-flop cannot be tilted by a power glitch whose transitions doesn't coincide with clock edges. As a consequence, the only way to induce errors in sequential logic using power supply glitches is the temporary modification of the timing properties of the gates[2] in order to violate some timing constraints at the boundaries of the combinatorial logic blocs.

In the following section, we will describe the simulations that can be done to analyse the effects of power supply glitches on a logic cone and its output register. We will next summarize the main results we have obtained.

IV. Simulations

Figure 4 illustrates the typical simulation circuit chosen for power supply glitch effect analysis on a flip-flop (the flip-flop under analysis is the right one).

The simulations are performed using Eldo (V5.6) simulator, from Mentor Graphics, with Philips MOS9 Model for the

[2]Here, gates is taken in the wide sense, i.e. designating the combinatorial and the sequential gates.

β	0.5	1	1.5	2	2.5	3	5
$\Delta V_A/\Delta V_{dd}$	0.40	0.57	0.66	0.72	0.76	0.80	0.86
$\Delta V_B/\Delta V_{dd}$	0.28	0.43	0.53	0.60	0.65	0.69	0.78
$\Delta V_B/\Delta V_A$	0.68	0.75	0.79	0.82	0.85	0.86	0.91

TABLE I

HIGH FREQUENCY TRANSMITTANCES VALUES AS A FUNCTION OF β

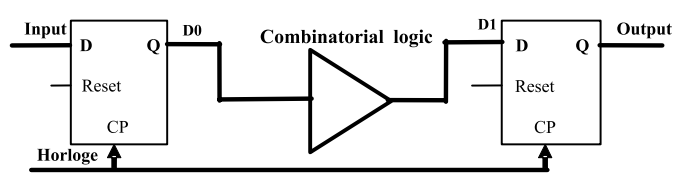


Fig. 4. Flip-flop typical usage

transistors. Glitches are modeled as trapezoidal pulses as shown on figure 5 (negative glitch on the left and positive glitch on the right).

Glitch parameters were varied throughout simulations in order to analyze the behaviour of the master-slave D-flip-flop. We tried to apply the power supply glitch during stable states of the clock and around its transitions. The results showed that, in agreement with the theoretical results given above, the D-flip-flop itself is insensitive to power glitches: after the end of the glitch, Q output reaches its correct state. In fact, the only qualitatively perceptible effect was a slight modification of the hold, setup and delay times.

On the other hand, the effect of the power supply glitch on combinatorial logic is more evident: it modifies the delay of the logic path. For larges negative glitch amplitudes, we can talk of a true "freeze" of the signal propagation during the glitch.

Figure 6 shows how this effect on combinatorial logic can be converted into a bit flip when the added delay is larger than the time slack of the path leading from the output of a flip-flop to the input of another (topology of figure 4). For this example, the parameters are as follow:

- Clock Period : $PCP = 6\,\mathrm{ns}$
- The delay between CP and $D1$ (the longest path of the combinational logic plus the propagation delay of the first flip-flop) is $T_p = 2.5\,\mathrm{ns}$; consequently, the corresponding time slack is $T_S = 3.5\,\mathrm{ns}$;
- The negative power supply glitch has : $V_{dd} = 1.65\,\mathrm{V}$, $A = 0.65\,\mathrm{V}$, $\delta = 5\,\mathrm{ns}$ and $T_r = T_f = 0.5\,\mathrm{ns}$.

To be more precise, we can see on the right part of figure 6 that, due to the slowing down of the circuit, the delay between $D0$ and $D1$ is increased beyond PCP the clock period. Consequently, when occurs the active clock edge following the glitch, the second flip-flop still "see" on its input the "previous" value of $D1$. At the end of this clock cycle, even if a new propagation between $D0$ and $D1$ takes place, the second flip-flop output reaches its correct state: the bit flip duration was only one cycle.

Obviously, some conditions must be satisfied to be able to get a bit flip at the output of a register. To precise them, we must define some notations. The aim of figure 7 is to remind

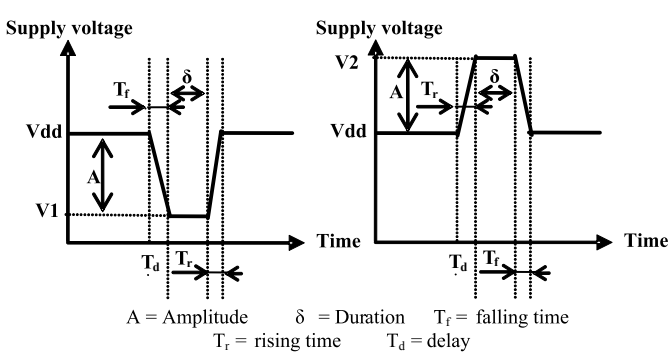


Fig. 5. Power glitch characteristics

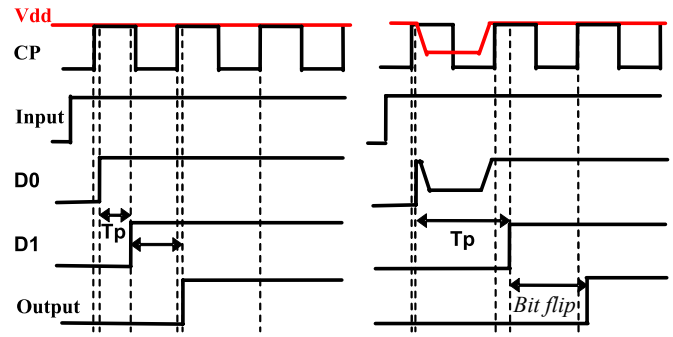


Fig. 6. Example of power supply glitch effect

the main timing definitions used when designing a data path (signal names are referred to figure 4):

- *Delay* is the max propagation time of the combinational logic between $D0$ and $D1$.
- *SetUp* is the minimum time to be respected, between the settling of the D–input and the active edge of the clock, to have a proper operation of the flip-flop.
- *Thold* is the minimum time to be respected, between the active clock edge and a variation of the D–input, to have a proper operation of the flip-flop. Usually, $Thold = 0$.
- *Slack* is the difference between the clock period and the sum of *Delay*, *SetUp* and the own delay time of the flip-flop (defined as the delay between the active clock edge and the settling of the corresponding Q–output).

Two main conditions must be met to be able to expect a bit flip for a given register output:

- The slack of the critical path leading to the corresponding flip-flop input must be shorter than the delay increase induced by the power supply glitch.
- A true (not virtual) transition must be propagating along the considered critical path at the time of the glitch occurrence.

Of course, these are only necessary conditions, they are not sufficient. As an example, the input transition of the considered flip-flop can occurs in the "Setup-Hold" period surrounding the active clock edge, leading to "only" an error probability.

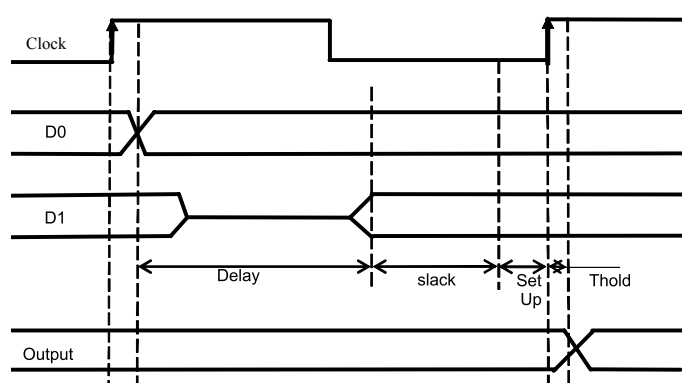


Fig. 7. Data path timing definitions

More important, the clock tree distributing the clock signal to the various flip-flop, with heavy constraints on the clock-skew, is also a combinational path whose propagation delay is influenced by the power supply voltage variations... This effect tends to extend the slack to the critical path when applying negative V_{dd} glitches. On the other hand, this extension of slack is "eaten" from the slack of the critical paths of the next clock cycle...

V. Conclusions

We have shown theoretically and verified by simulation that D-flip-flop are resistant to V_{dd} glitches whose transitions occur between clock edges. The first main important result found in this study, come from that V_{dd} glitches induce timing violations in combinational logic cones, which surround flip-flops. Consequently, we are focusing our work on the behaviour of the combinational logic.

On the other hand, simulations show the relationship between the delay in the combinational logic and the clock frequency. If the propagation time throughout the combinational is proportional to the clock period (T_p/PCP close to 1), the probability of observing errors increases significantly.

The clock tree is also a combinational logic. However, in a given design, the propagation time throughout the most critical combinational logic path (with the less significant slack), is significantly most important than the delay between the root clock and the leaf cells (synchronous cells). So, at first sight we think that we can carry on our work using an ideal clock. But, this will require to be validated on true (hardware) examples.

VI. Perspectives

The following of this study will lead to extract criteria of flip-flop faults' sensitivity that is bound, from now on, to the combinational logic. We are thus studying the propagation of a glitch throughout logic cones. We carry out this study on simple structures providing the same function, but having different architectures (length, reconvergent paths, different type of standard cells: inverting and non inverting, etc.).

To do so, our methodology is the following:

- For each structure, different glitches are applied (positive, negative, varying timing parameters of the glitch) to extract the most pertinent glitches if there are any.
- Confront the responses of each structure under the same environmental conditions. This allows the evaluation of the structure vulnerability compared with another one.

The forecasted objectives are

1) Assignment of a criterion of flip-flop sensitivity (depending on its logical cone) for faults' injection simulations at the logical level.
2) Counter measures: definition of design rules to avoid vulnerable combinational logic structures. To do so, we will have to confront the built model to experimental measurement on silicon. Theses results will be validated on a test chip designed for functional verification implemented in HCMOSM8 technology. It contains a specific block composed by the entire library's standard cells, which are connected to a scan chain allowing the functional verification.

References

[1] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the importance of checking cryptographic protocols for faults," in *Proc. Advances in Cryptology - EUROCRYPT '97: International Conference on the Theory and Application of Cryptographic Techniques*, ser. Lecture Notes in Computer Science, vol. 1233. Konstanz, Germany: Springer-Verlag, May 1997, pp. 37–51.

[2] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The sorcerer's apprentice guide to fault attacks," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 370–382, Feb. 2006.

[3] R. J. Anderson and M. G. Kuhn, "Low cost attacks on tamper resistant devices," in *Proc. 5th International Workshop on Security Protocols*, ser. Lecture Notes in Computer Science, vol. 1361. Paris, France: Springer-Verlag, Apr. 7–9, 1997, pp. 125–136.

[4] S. Skorobogatov and R. Anderson, "Optical fault induction attacks," in *Proc. Cryptographic Hardware and Embedded Systems - CHES 2002: 4th International Workshop*, ser. Lecture Notes in Computer Science, vol. 2523. Redwood Shores, CA, USA: Springer-Verlag, Aug. 13–15, 2002, pp. 2–12.

[5] J.-M. Dutertre, "Circuits reconfigurables robustes," thèse de doctorat, Université de Montpellier 2, oct 2002.

[6] T. Monnier, "Durcissement de circuits convertisseurs a/n rapides fonctionnant en environnement spatial," thèse de doctorat, Université de Montpellier 2, oct 1999.

[7] D. Leroy, S. J. Piestrak, F. Monteiro, and A. Dandache, "Modeling of transients caused by a laser attack on smart cards," in *Proc. IOLTS 2005, 11th IEEE International On-Line Testing Symposium*, 2005, pp. 193–194.

[8] L. Anghel, "Test des circuits intégrés, cours de l'école d'électronique numérique IN2P3 ENSERG-INP Grenoble," 2003, unpublished.