

A SECURE PROGRAMMABLE ARCHITECTURE WITH A DEDICATED TECH-MAPPING ALGORITHM: APPLICATION TO A CRYPTO-PROCESSOR

Taha Beyrouthy¹, Laurent Fesquet¹, Alin Razafindraibe¹, Sumanta Chaudhuri², Sylvain Guille², Philippe Hoogvorst², Jean-Luc Danger², Marc Renaudin¹

¹ TIMA Laboratory, Institut National Polytechnique de Grenoble, 46 Avenue Félix Viallet 38031 Grenoble France

² Ecole Nationale Supérieure des Télécoms, 46 rue Barrault, 75634 Paris cedex 13 France

email: taha.beyrouthy@imag.fr

ABSTRACT

With worldwide communications, information technology and confidentiality have become a major issue for exchanging and securing data. Nevertheless the ASIC high costs and the frequent updates in cryptographic standards used in security applications such as homeland security or banking have made the ciphering algorithms on an embedded FPGA (e-FPGA) co-processor a viable alternative. This paper presents a secure e-FPGA architecture dedicated to security applications, which is able to support asynchronous crypto-processor implementations in order to defeat side-channel attacks. The proposed architecture is natively robust against attacks such as differential power analysis (DPA) or clock based fault attacks. The article also presents a specific tech-mapping algorithm for implementing asynchronous dual rail functions on this e-FPGA. This algorithm respects the security constraints to be robust against DPA. Electrical simulations on a sensitive crypto-processor module validate the proposed approach.

1. INTRODUCTION

While synchronous circuits design has reach a high level of performance, the clock distribution issues have become a real problem to cope with, therefore asynchronous circuits are more and more used in order to remove this clock-tree as well as dealing with the power consumption overhead which drastically increases with frequency. Moreover, the asynchronous circuits appear to be an interesting alternative to their synchronous counterparts for implementing cryptosystems [19][20]. In fact these latter are very sensitive to the so-called Side-Channel Attacks (SCAs) which aim at illegally retrieving secret information contained in cryptographic systems. At the same time programmable circuits have proved their validation role in the logic design flow and their high level of flexibility and performance. Therefore combining asynchronous logic with programmable circuits will be very helpful to explore and experiment the ability of asynchronous circuits, with or without specific countermeasures, to resist against power attacks and Fault Attacks.

In the literature, several architectures of programmable asynchronous circuits have been proposed [4] [5] and [6]. From the flexibility point of view, most of them are dedicated either to a specific asynchronous circuit style (PGA-STC [8], PAPA[11], Achronix FPGAs [25] or to a dedicated application (MONTAGE [7], GALSA [9], STACC [10]). For examples: PGA-STC was developed to implement two-phase bundled-data systems such as micro-pipelines, GALSA for massively parallel

computing architectures, STACC for reconfigurable computation, PAPA for fine-grain pipelines with a high throughput. From the security point of view, all these FPGAs are vulnerable to Differential Power Analysis attacks and more generally to SCAs attacks. In spite of this situation, very few research works address the FPGA security.

Within a project called SAFE, this work aims at specifying, designing and validating an asynchronous programmable circuit suitable for flexible, high performance and secure-implementations., We propose a novel FPGA architecture, natively robust to DPA attacks and really more flexible than the existing asynchronous programmable circuits. To achieve such a level of robustness, all security problems are addressed at all abstraction layers: architectural, logical, electrical and physical (routing). The paper is organized in three parts. The first part addresses the security constraints that must be fulfilled by the e-FPGA hardware as well as the mapping algorithm. The second part describes the e-FPGA architecture and the mapping algorithm. Finally, the last part gives simulation results obtained on a sensitive crypto-processor module.

2. SECURITY FEATURES OF THE SAFE FPGA

The FPGA reconfigurability offers major advantages for cryptographic applications [23]. However, the physical implementation of FPGAs might provide side-channels which leak unwanted information. The side-channels include in particular power consumption, timing

behavior, electromagnetic emission, surface temperature, etc. All of these side-channels are information sources which can potentially be used by attackers to reveal the secret information. Simple Power Analysis (SPA) and Differential Power Analysis (DPA) have been introduced by Paul Kocher [13]. While performing a ciphering operation, the power consumption of cryptographic devices is analyzed in order to extract the secret cipher keys. These attacks exploit the data power dependency of the cryptographic devices. In [17], the Electromagnetic Analysis (EMA) is presented as a more efficient attack than DPA. It exploits the electromagnetic fields emitted by the switching gates as side-channel information. In addition to SCAs, Fault Attacks (FA) have been presented as an alternative scenario. In [14], faults are injected into the device while a known program is executed. In such situations, the device behavior can reveal the wanted information to the hacker.

Many countermeasures have recently been implemented in ASICs to prevent SPA, DPA, EMA and FAs. One approach — using balanced quasi delay insensitive (QDI) asynchronous circuits [15] — appears to be one of the most promising. The SAFE project aims at transposing this method in an e-FPGA context. The challenge is first to make the asynchronous FPGA natively robust against SPA and DPA while being very flexible. Afterwards, countermeasures against other SCAs and FAs can easily be explored and experimented. The SAFE e-FPGA implements many features for security:

Balanced power consumption — QDI circuits which generally use 1-of-n encoding (for example: dual-rail, triple-rail, etc.) can be balanced to reduce the power consumption dependency with the processed data. Indeed, the bit encoding ensures that the data are transmitted and computations are performed with a constant Hamming weight. This is important since the leakage of the Hamming weight can be exploited by SPA, DPA, and EMA. In addition, the Hamming weight constant technique can be combined with other countermeasures. In [20], the author proposes to use a temporal or spatial jitter to make power consumption more unintelligible.

Absence of a global clock signal — No clock means that FAs based on clock are removed. Moreover, DPA and SPA attacks without global clock signal are expected to be much more difficult. Indeed, the clock absence will make very complicated the synchronization of the DPA and SPA signatures.

Environment variation tolerance — QDI circuits adapt to their environment such as voltage and temperature variations, which means that they tolerate many forms of fault injection (power glitches, thermal gradients, etc.). These QDI circuits can be easily combined with other countermeasure to efficiently counteract FAs [19].

Redundant data encoding — QDI circuits typically use a redundant encoding scheme (1-of-n). For example, the dual-rail encoding (a bit is encoded onto two wires) provides a mean to encode an alarm signal to counteract FAs [15].

3. E-FPGA ARCHITECTURE

This section gives an overview of the proposed asynchronous FPGA architecture. Our programmable Logic bloc architecture consists of 4 major blocks: 2 Logic elements including 2 LUT6-1 (Look-Up-Table with 6 inputs and 1 output), 1 LUT 2-1 and 1 LUT4-1.

3.1. General description of the FPGA architecture

The global architecture of the e-FPGA is an island-style architecture composed by Programmable Logic Blocks (PLBs). As a classical FPGA, it also contains a programmable interconnect network whose the building blocks are Connection Boxes (CBs) and Switch Boxes (SBs) [1][2]. Finally, the e-FPGA architecture is the repetition in 2-dimension of the pattern made by a PLB, 2 connection boxes, and a switch box as described in Figure 1.

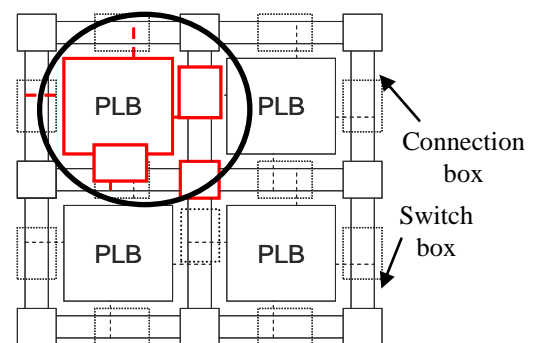


Figure 1: e-FPGA architecture.

3.1.1. The Programmable Logic Block (PLB)

The PLB architecture has been designed to be a good trade-off between the high flexibility required to be style independent (asynchronous logic styles) and to optimally use the PLB resources. Figure 2 shows the details of the PLB architecture, which has 12 inputs and 7 outputs. It consists in two Logic Element (LE), one LUT2-1, and one LUT4-1. Its outputs are directly connected to the connection boxes.

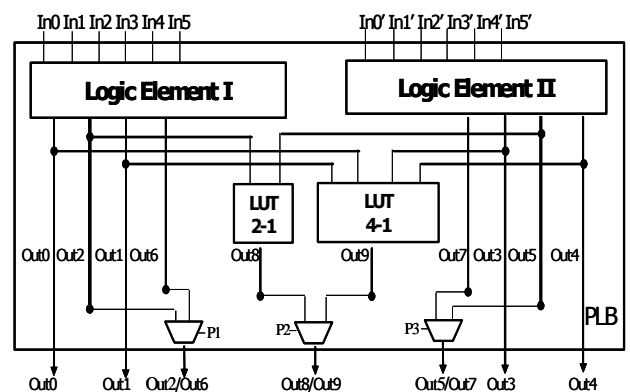


Figure 2: PLB architecture.

3.1.2. The Logic Element (LE)

The Logic Element is a programmable logic component which hosts the function generators. It has 6 inputs, and 4 outputs, and consists of two LUT6-1 [1][2], followed by a multiplexer and one single-output LUT2-1, connected together as shown in Figure 3.

For more flexibility, inside the LE, each input is connected to a multiplexer $M_k In_i Out_m^*$ that allows a programmable choice between 2 types of inputs:

- The first is a primary input ($In_j, j = 0,1,2,3,4,5$) connected to external signals.
- The second is a feedback, where internal signals (in fact some of PLB outputs $Out_i, i = 0,1,3,4,5$) are looped back to the PLB inputs (see Figure 3).

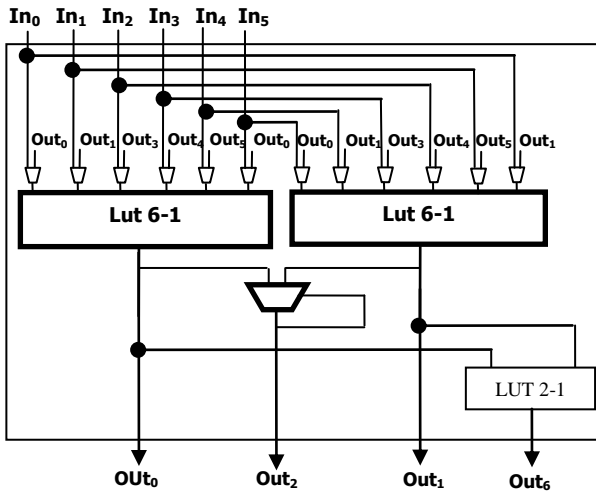


Figure 3: Logic Element architecture.

As shown on Figure 3, the LE outputs are fed back to the LE inputs through multiplexers in order to ensure that implementation of memory elements which are commonly used in asynchronous logic such as Muller gates [16]. Moreover, they give much more flexibility to the LE to implement complex functions.

To ensure such a behavior, each multiplexer on the LUT inputs is controlled by a programmable selection bit S_k : if S_k is set to 1, memorization is enabled, and thus one of the PLB outputs will become one of the PLB input. Otherwise (S_k set to 0), primary inputs are used, and so external signals are connected to the LE inputs.

Thus each LE can implement a 6-input logical function, which is almost the equivalent of a 2-input – dual rail gate (2 rails for each input + acknowledgement signal + feedback for memorization). Implementing a 3-input dual-rail gate requires two LEs. Each output rail will be mapped on the LE output Out_2 . The Multiplexer output Out_2 connects the outputs of the LUT6-1, Out_0 and Out_1 (see Figure 3). This output is fed back to the multiplexer selection bit. Thus a memorization is always implemented on Out_2 (see Equation 1).

$$Out_2 = f(Out_0, Out_1, Out_2) \quad (1)$$

As multiple outputs are available per LE, dual-rail encoding can easily be implemented. In addition, implementing multi-rail encoding is also quite simple with a reduced number of LEs, thanks to their large number of inputs.

Using dual-rail encoding, which is a balanced data encoding, in association with a symmetric e-FPGA architecture, enhances the security characteristics of the e-FPGA, making DPA more difficult [12]. With this encoding style, both logical ‘0’ and logical ‘1’ are encoded with code words of the same Hamming weight, ‘01’ and ‘10’ respectively. If the electrical symmetry is respected in the whole design, with such an encoding style, power consumption is expected to be data independent.

As we can see in figure 3, the LE, which is the function generator, is fully balanced at the architectural level.

3.1.2.1. The Look Up Table 6-1 (LUT 6-1).

Equation 2 shows that a LUT6-1 is able to implement, a 6-input function.

$$Out_0 = f_1(In_0|Out_0, In_1|Out_1, In_2|Out_3, In_3|Out_4, In_4|Out_5, In_5|Out_0), \quad (2)$$

It is important to notice that the feedback usage allows:

Memorization: In this case, the internal feedback is used. The LUTs are able to implement functions with memorization which are required for asynchronous logic.

Cascading functions: These feedbacks allow the mapping of multi-rail and complex functions (more than 6-inputs). Thanks to a specific algorithm, the four external feedbacks help to split complex functions into smaller ones (≤ 6 inputs) and to cascade them in the same PLB. Thus no resource is taken from the interconnection network.

3.1.2.2. The LUT 2-1 and LUT 4-1.

Asynchronous logic requires implementing protocols between communicating modules, which basically consists in computing an acknowledgment signal. Inside the Logic Element, this protocol is supported by adding a LUT2-1 directly connected to the outputs of both LUT6-1 (out_6) (see Figure 3). This enables to check the data validity on the two wires, out_0 and out_1 , and to provide an acknowledgement.

Outside the Logic Element, the LUT2-1 and the LUT4-1 ensure the same role (out_8 and out_9) (cf. Figure 2). As out_2 and out_6 are never used simultaneously, we connected them to a multiplexer whose selection input is a programming bit (P1) (cf. Figure 2). Same thing is done to (out_7/out_5) and (out_8/out_9). As a result the number of PLB’s outputs is decreased and the interconnection complexity is reduced.

3.2. Technology mapping

The e-FPGA is a reconfigurable integrated circuit that consists of an array of programmable logic blocs (PLBs) with vertical and horizontal programmable routing network. The PLBs are based on the 6-input LUT where a LUT6-1 contains 2^6 truth table configuration bits so it can implement any 6-input function.

On the one hand, the number of PLBs needed to implement a given circuit determines the size and cost of the FPGA. On the other hand, its security depends on the symmetry of the whole PLB network. Therefore one of the most important phases of the FPGA design flow is the technology mapping step which maps the optimized circuit description into a PLB network.

The goal of the technology mapping step is first to balance the architecture of the whole circuit and second to reduce area and delay. Within the SAFE project, the algorithm of technology mapping allows keeping the symmetry when implementing “balanced function”. More precisely, this algorithm is able to implement electrically balanced functions with the following features:

Area-efficient: the implementation minimizes the PLB’s resources.

Secure: the whole circuit will be balanced at the architectural level. In the case of a complex function (K-input, $K > 13$), the algorithm is able to split the latter into many smaller functions (K-input, $K < 13$). Afterwards, the algorithm is able to map each small function on a PLB and implement a fully balanced circuit (Balanced tree topology). More precisely, in a logical cone, each input is propagated from input to output through the same number of blocks.

3.3. The mapping algorithm

This section will briefly describe the mapping algorithm used with the FPGA presented above. It allows us to implement 4-phase Quasi Delay Insensitive (QDI) logic with 1-of-m encoding.

For a given dual rail function f , Out_0 and Out_1 are the outputs representing respectively the bit value “0” and “1”. The algorithm is given below:

For functions Out_0 and Out_1

Begin

Compute the number n of single inputs of both functions.

(It must be the same because both of them belong to the same dual rail function. In fact as 4-phase protocol is used, n is representing the number of inputs including the feedback.

ex: if $Out_0 = Y(a0, a1, b0, b1, c0, c1, out_0^{-1})$, then $n = 7$, where out_0^{-1} is the feedback.)

If $n \leq 6$ then

use 2 LUT6 of the same LE to implement each function.

If $n = 7$ then

use 2 LE of the same PLB to implement each of them.

(It is worth noting that the feedback of each function is in this case ensured by the multiplexer connecting the outputs of the 2 LUT6 of the LE (see Figure 3)).

If $7 < n \leq 13$ then

do method_for_more_than_7.

If $n > 13$ then

split both functions into smaller ones until inputs ≤ 13 and do method_for_more_than_7.

End

Method for more than 7:

Before describing the method_for_more_than_7, it is important to mention the following points:

- 1- The LE does not support more than 7-input functions (Figure 3). Thus, a function with more than 7 inputs should be split into smaller sub-functions (inputs ≤ 7), in order to map each of them in different LEs.
- 2- The result of this decomposition is not secured against side channel attacks, because the final circuit is neither electrically nor logically balanced.
- 3- It is possible to make the above decomposition robust against side channel attacks, by adding countermeasures that balanced the circuit logically and electrically. Balancing the circuit results in an increase of the number of PLBs used to map the circuit.

As a result, the so-called method_for_more_than_7 is developed for securely mapping functions with more than 7 inputs and minimizing as much as possible the number of PLBs. The method is based on separating the communication protocol part from the computing part of the function. It is defined by 3 main steps:

- The first step consists in computing the output without considering the communication protocol.
- The second step adds the communication protocol to the circuit. At this level, the circuit work normally, but it is not balanced.
- In the third step, countermeasures are added to balance logically and electrically the circuit.

This method has been validated, as presented in the example below. The resulting circuit is well-balanced and power is data-independent.

4. EXPERIMENTAL RESULTS

In this section, a sensitive sub-module of the DES [24] (Data Encryption Standard) algorithm is studied (see Figure 4). A brief presentation describes the design and

the architecture used to secure this module and make its consumption data-independent.

4.1. Mapping a sensitive DES sub-module.

Figure 4 shows that the combination of the plaintext with the secret key is done by a 6-input XOR. It is important to ensure a high level of security on this block and on the S-BOX bloc [25]. Otherwise a hacker could easily retrieve, through a power analysis, the secret key used during the encryption.

In terms of resources, 9 PLBs are required to have an implementation of this sub-module that respects the security constraints. The XOR is implemented on three PLBs and six are used to implement the 6-input dual-rail S-BOX2 (6 dual-rail inputs, 4 dual-rail outputs).

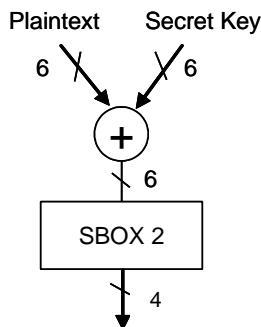


Figure 4 : Experimental setup.

In order to evaluate the area efficiency of the bloc implementation, a filling ratio described below has been calculated. The filling ratio of the LEs is defined as the number of used primary inputs over the total number of primary inputs. Figure 3 shows that a Logic Bloc (LE) has in total 6 primary inputs (In_0 , In_1 , In_2 , In_3 , In_4 , and In_5). Thus the overall filling ratio of this sub-module is 92%.

To meet the security constraints presented in section 2:

- The sub-blocs of this function are implemented using a 1 out of N encoding. This strategy guarantees a constant Hamming weight which is required to make power consumption data-independent.
- The circuit architecture is fully symmetric. This means that all the data paths have the same logical depth.

To validate the e-FPGA native robustness against SPA and DPA attacks, an electrical simulation campaign have been carried out. The analyzed bloc (cf. Figure 4) has been designed in a CMOS 65nm technology. Remind that, to be robust against SPA and DPA, the bloc (cf. Figure 4) should have the same current profiles and a constant running time whatever the manipulated data.

During the electrical simulation campaign and for a given secret key, random plaintext vectors have been processed. The corresponding current profiles are given in Figure 5 and the outputs are given in Figure 6.

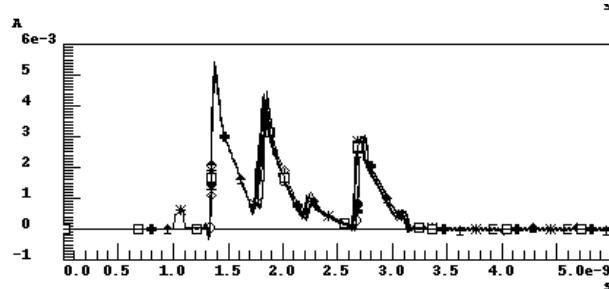


Figure 5: Current profiles of the bloc.

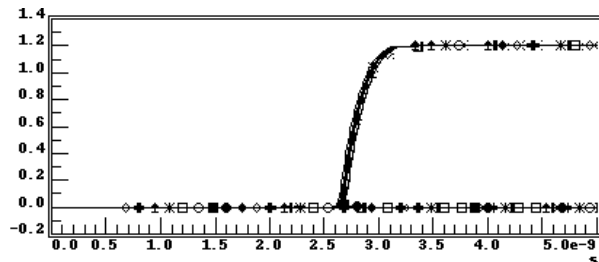


Figure 6 Outputs of the Figure 4 bloc.

Figure 5 shows that, whatever the manipulated data are, the current profiles are very similar. In other words, the power consumption is data independent. In addition, as shown in Figure 6, the outputs are completely superposed. This means that the e-FPGA running time is data independent. This drastically increases the circuit robustness against SCAs exploiting the running time variations. In conclusion, with data independent power consumption and a constant running time, the proposed asynchronous e-FPGA architecture is natively robust against SPA, DPA and timing attacks.

5. CONCLUSION

In this paper, we proposed a novel asynchronous embedded FPGA architecture which is more appropriate than conventional FPGA architectures to support asynchronous designs and mainly dedicated to security applications. This novel architecture has been designed to be natively robust against power-based attacks and enough flexible to allow exploring and experimenting countermeasures against SCAs and FAs. To achieve data independent power consumption, this novel architecture adopts the 1-of-n encoding and four phase protocol communication. In addition, the building blocs have been designed to be logically and electrically balanced. A balanced multi-rail routing technique is also proposed in the paper entitled "Physical Design of FPGA Interconnect to Prevent Information Leakage" [26]. In summary, within the SAFE project, security problems have been addressed at the architectural, logical, electrical and routing levels.

Up to now, these first encouraging results indicate that a high security level is possible for secure applications implemented on a dedicated asynchronous e-FPGA. Such an embedded FPGA is a promising approach to counteract attacks such as SPA, DPA, timing- and fault-

attacks and to be enough flexible for updating secure systems.

6. REFERENCES

- [1] L. Fesquet, M. Renaudin. "Programmable logic architecture for prototyping clockless circuits" In FPL 2005, Tampere, Finland, August 24-26, 2005, pp 293-298.
- [2] N. Huot, H. Dubreuil, L. Fesquet, M. Renaudin, "FPGA architecture for multi-style asynchronous logic", In DATE 2005, Munich Germany, March 7-11, 2005, pp. 32-33.
- [3] M. Renaudin, "Asynchronous circuits and systems: a promising design alternative", *Microelectronic Engineering* 54(2000), p. 133-149
- [4] A.J. Martin, and Andrew Lines et al. "The Design of an Asynchronous MIPS R3000 Microprocessor". In *ARVLSI'97*, p.164-181, Ann Arbor, MI, Sep. 15-16 1997.
- [5] A.J.Martin and M. Nystrom et al. The Lutonium: "A sub-nanjoule asynchronous 8051 microcontroller". In *ASYNC 2003*, p.14-23, Vancouver, Canada, May 12-15 1997.
- [6] J.D. Garside et al.: "AMULET3i – an Asynchronous System-on-Chip", In *ASYNC 2000*, pages 162-175, Apr. 2-6 2000.
- [7] Scott Hauck, Gaetano Boriello, Carl Ebeling: "Montage: An FPGA for Synchronous and Asynchronous Circuits", 2nd International Workshop on Field-Programmable Logic and Applications, Vienna, August 1992.
- [8] Kapilan Makeswaran and Venkatesh Akella: "PGA-STC : Programmable Gate Array for Implementating Self-Timed Circuits", *International Journal of Electronics*, Volume 84, Number 3/March 1, 1998.
- [9] B. Gao: "A globally asynchronous locally synchronous configurable array architecture for algorithm embeddings", PhD thesis, University of Edinburgh, December 1996.
- [10] Robert Payne : "Self Timed Field Programmable Gate Array Architectures", PhD thesis, University of Edinburgh, 1997.
- [11] John Teifel and Rajit Manohar: "Highly Pipelined Asynchronous FPGAs", In 2th ACM International Symposium on Field-Programmable Gate Arrays, Monterey, CA, February 2004.
- [12] D.Solokov, J Murphy, A.Bystrov, A.Yakovlev. : "Improving the Security of dual rail circuits", Workshop on Cryptographic Hardware and Embedded Systems, Cambridge (Boston), Ma., USA August 10-13, 2004.
- [13] Paul C. Kocher, Joshua Jaeger, and Benjamin Jun: "Differential power analysis, *Advances in Cryptology* ", *CRYPTO '99* (M. Wiener, ed.), *Lecture Notes in Computer Science*, vol. 1666, Springer-Verlag, 1999, pp. 388-397.
- [14] Ross J. Anderson and Markus G. Kuhn: "Low cost attacks on tamper resistant devices, *Security Protocols*", 5th International Workshop, Paris, France, April 7-9, 1997 (M. Lomas et al., ed.), *Lecture Notes in Computer Science*, vol. 1361, Springer-Verlag, 1997, pp. 125-136.
- [15] S. Moore, R. Anderson, P. Cunningham, R. Mullins and G. Taylor: "Improving Smart Card Security using Self-timed Circuits", in *Proc. 8th IEEE International Symposium on Asynchronous Circuits and Systems – ASYNC '02*, pp. 23–58, IEEE 2002.
- [16] L. Fesquet, B. Folco, M. Steiner, M. Renaudin: "State-holding in Look-Up Tables: application to asynchronous logic", in *VLSI-SoC*, November 2006, Nice, France, pp. 12-17.
- [17] Quisquater et al, *ElectroMagnetic Analysis (EMA): Measures and Counter-measures for Smart Cards*, *E-smart'01*, LNCS 2140, p. 200.
- [18] Renaudin M., Bouesse G.F., Proust P., Tual J.-P., Sourgen L., Germain F., "High security smartcards", in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, Paris, France, 2004
- [19] Yannick Monnet, Marc Renaudin, Régis Leveugle, "Designing Resistant Circuits against Malicious Faults Injection Using Asynchronous Logic," *IEEE Transactions on Computers*, vol. 55, no. 9, Sept., 2006, pp. 1104-1115.
- [20] F. Bouesse, G. Sicard, M. Renaudin, "Path Swapping Method to Improve DPA Resistance of QDI Asynchronous Circuits" 8th International Workshop on Cryptographic Hardware and Embedded Systems—CHES2006, , Yokohama, Japan, October 2006, LNCS 4249 Springer 2006, pp. 384-398.
- [21] Quoc Thai Ho, Jean-Baptiste Rigaud, Laurent Fesquet, Marc Renaudin, Robin Rolland: 'Implementing Asynchronous Circuits on LUT Based FPGAs', *FPL 2002*, pp 36-46.
- [22] P. Chow, S. Ong, J. Rose, K. Chung, G. Paez-Monzon, I. Rahardja, "The design of an SRAM-based Field-Programmable Gate Array, Part II: Circuit Design and Layout", *IEEE Transactions On VLSI Systems*, Vol. , 1999.
- [23] Thomas Wollinger and Christof Paar, "How Secure Are FPGAs in Cryptographic Applications?", In 13th International Conference on Field Programmable Logic and Applications - FPL 2003, Lisbon, Portugal, September 1-3, 2003
- [24] FIPS 46-3 Published October 1999, Data Encryption Standard (DES); specifies the use of Triple DES.
- [25] <http://www.achronix.com/>
- [26] Sumanta chaudhuri, Philippe Hoogvorst, Sylvain Guillely, Jean-Luc Danger, Taha Beyrouthy, Alin Razafindraibe, Laurent Fesquet, Marc Renaudin, "Physical Design of FPGA Interconnect to Prevent Information Leakage", *International conference on Applied Reconfigurable Computing ARC'08*, Imperial College, London, UK, March 26-28, 2008.