DPA RESISTANCE FOR LIGHT-WEIGHT IMPLEMENTATIONS OF CRYPTOGRAPHIC ALGORITHMS ON FPGAS

Rajesh Velegalati and Jens-Peter Kaps

Volgenau School of IT&E, George Mason University
Fairfax, VA, USA
email: {rvelegal, jkaps}@gmu.edu

ABSTRACT

Recent advances in Field Programmable Gate Array (FPGA) technology are bound to make FPGAs a popular platform for battery powered devices. Many applications of such devices are mission critical and require the use of cryptographic algorithms to provide the desired security. However, Differential Power Analysis (DPA) attacks pose a sever threat against otherwise secure cryptographic implementations. Current techniques to defend against DPA attacks such as Dynamic Differential Logic (DDL) lead to an increase in area consumption of factor five or more. In this paper we show that moderate security against DPA attacks can be achieved for FPGAs using DDL resulting in an area increase of not much more than a factor two over standard FPGA implementations. Our design flow requires only FPGA design tools and some scripts.

1. INTRODUCTION

1.1. Motivation

Field Programmable Gate Arrays (FPGA) are a popular choice for applications where low up-front cost, fast time to market and flexibility are important. Many of those applications are based on battery powered mobile devices which require low power consumption. FPGAs are less energy efficient than ASICs or embedded processors and hence they are not often found in such applications. Recent advances in FPGA technology [1] however, will enable FPGAs to become more popular in this market.

Light weight or low area consuming implementations of cryptographic algorithms facilitate the use of smaller and hence less expensive FPGAs or enable their use in battery powered applications. Unfortunately Differential Power Analysis (DPA) attacks are a threat to otherwise secure cryptosystems on embedded devices.

 Rajesh Velegalati and Jens-Peter Kaps. DPA resistance for light-weight implementations of cryptographic algorithms on FPGAs. In Martin Daněk, Jiří Kadlec, and Brent Nelson, editors, *Field Programmable Logic and Applications, FPL 2009*, pages 385–390. IEEE, Aug 2009. http://dx.doi.org/10.1109/FPL.2009.5272260

It is the goal of this paper to introduce a secure design flow that enables us to achieve resistance to DPA attacks through Dynamic Differential Logic(DDL) for light-weight implementations of cryptographic algorithms on FPGAs.

1.2. Previous Work

In 1999, Kocher introduced DPA [2] which correlates side-channel information of a cryptographic algorithm, such as power consumption, with the secret key. In subsequent years many successful DPA attacks against cryptographic algorithms implemented in software and on ASICs were published. However, four years later the first results on successful DPA attacks against DES and RSA [3] and ECC [4] implementations on FPGAs were reported. The first design methodology to secure ASIC and FPGA implementations using DDL was published by Tiri in 2004 [5]. The goal of DDL is to eliminate the correlation between the data being processed and the power consumption of the circuit, hence making a DPA attack infeasible. This is accomplished through duplication of the original circuit into a direct and a complementary logic which follow two basic principles:

1. *Constant switching activity:* This guarantees a single switching event per clock cycle and gate output. During each clock cycle either a gate output in the direct path switches or the corresponding gate in the complementary path.
2. *Constant load capacitance:* The capacitive loads driven by the gates in the direct path is equal to the load driven by the gates in the complementary path.

Recent implementation results of WDDL on FPGAs show two major drawbacks of this technique [6]. WDDL requires the use of glitch free positive logic and duplication which leads to an increase in area consumption of more than a factor five over a single ended design on FPGAs. Furthermore, balanced load capacitances cannot be guaranteed on an FPGA because the required cross connections between the direct and the complementary logic lead to unbalanced paths. Therefore, Yu proposes in [6] to use Double

Wave Dynamic Differential Logic (DWDDL) which allows for balanced load capacitances on FPGAs. However, it leads to an area increase of more than eleven times.

In [7] Guilley et al. evaluate methods which reduce the size of WDDL implementations on FPGA. They were able to reduce the size of a WDDL implementation of Triple DES by 23% [8] through new synthesis methods. However, this design is still much larger than a single ended design due to the use of only positive logic as required by WDDL. Current FPGA tools can not produce net lists with only positive logic, hence Guilley uses ASIC tools and a special ASIC library containing hundreds of cells.

The power dissipation of an Xilinx Virtex™-II FPGA consists to more than 60% of power consumed by the routing resources [9]. This illustrates that it is important to balance the paths of the direct and complementary logic. As this is not a trivial problem on FPGAs, masking schemes have been proposed specifically to overcome the routing problem [10]. However, it has been shown in 2005 that circuits protected only by masking are not secure [11]. Later publications demonstrated that masking does not remove the need for balanced routing in WDDL designs [12, 13].

The impact of current place-and-route methods on the security of a WDDL design was explored in [14] with respect to balancing the timing delays. In [15] the authors propose a new switch box design for FPGAs that enables balanced routing and is secure against power as well as EM attacks.

2. DYNAMIC DIFFERENTIAL LOGIC

The dynamic power consumption in a CMOS circuit depends on the output transitions of the logic gates. This dependency is not symmetric i.e a $0 \rightarrow 1$ or $1 \rightarrow 0$ transition will consume power whereas $0 \rightarrow 0$ or $1 \rightarrow 1$ will not. This makes the power consumption dependent on the hamming distance of the data. The information thus obtained can be exploited to reveal the secret key. The Dynamic Differential logic style introduced by Tiri [5] attempts to remove the relation between power consumption and data.

A dynamic logic will have two phases, Pre-charge and Evaluation phase which alternate. During pre-charge phase the output of logic gates is forced to a constant value (0 or 1). The original transition of the logic gates will occur in evaluation phase. A differential logic will have two circuits, direct and complimentary whose outputs will be inverse of each other during evaluation phase. Lets assume that all outputs are pre-charged to 0. If an output of the direct logic evaluates to 1 then the complementary logic will evaluate to 0 leading to a single switching event. Similarly vice versa is also true thus achieving constant switching activity.

Constant load capacitance means that the gates in the direct logic drive the same load as the gates in the complementary logic. This requires that routing in both parts to be same, so called symmetrical routing.

There are two different DDL styles called Wave Dynamic Differential Logic (WDDL) and Simple Dynamic Differential Logic (SDDL).

2.1. Wave Dynamic Differential Logic

WDDL is being used successfully for secure cryptographic implementations in ASICs. It is an all positive logic which guarantees one transition per clock cycle. A pre-charge circuit is added only at the register outputs and system inputs. Inverters are implemented by cross connecting the outputs of direct and complementary circuits. WDDL gate is shown in Fig. 1a). This DDL allows a logic 0 wave to pass through the entire combinational logic hence the name ”wave” is added.

Unfortunately, applying WDDL to FPGAs is not straight forward. Firstly, FPGA CAD tools cannot be restricted to use only positive logic when synthesizing an implementation. Therefore, ASIC synthesizers are commonly used. Yu and Schaumont also show in [6] that the replacement of inverters by cross connection will result in unsymmetrical routing, which is undesirable.

In positive only architectures optimal usage of intrinsic features, for example fast interconnects, dedicated multiplexers, fast carry logics, shift registers etc inside the FPGA fabric cannot be exploited.

2.2. Simple Dynamic and Differential Logic

SDDL allows the usage of negative logic thus making it more flexible than WDDL. However each time negative logic is used a pre-charge circuit should be added as shown in Fig. 1b) because negative logic stops the precharge wave . In ASIC circuit this leads to an increase in the area compared to WDDL. Negative logic can produce glitches therefore SDDL cannot guarantee one switching event per clock cycle.

SDDL can be implemented using only FPGA cad tools. A negative logic style requires no cross connection between direct and complementary logic, hence symmetric routing is possible in FPGAs. SDDL allows optimal usage of the intrinsic features inside the FPGA thereby reducing the slice count. WDDL is considered to be a more secure logic style than SDDL because it is glitch free.

Our motivation was to use only the FPGA CAD toolsto produce a secure low area implementation using DDL. SDDL becomes an obvious choice.

3. PROPOSED SDDL MODEL

In our proposed SDDL model, FPGA CAD tools are given the maximum flexibility to optimize a given design for the

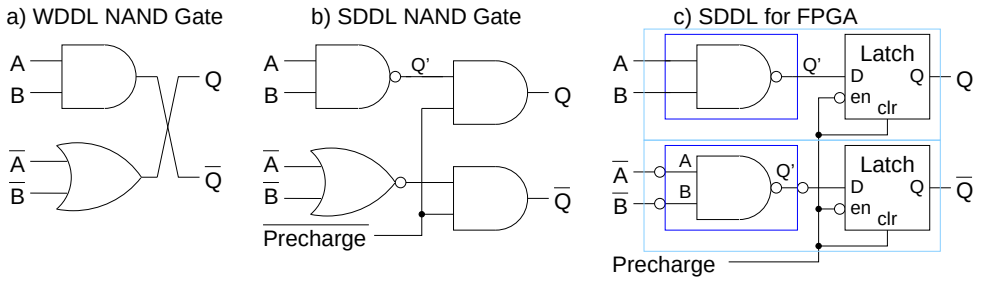

Fig. 1. Mapping WDDL to SDDL for FPGAs

target FPGA. Such an optimized design will allow logic packing in LUTs and also make use of all the intrinsic features present in the FPGA. In this paper we are exploring the usage of Wide Dedicated Multiplexer (WDM). Using WDMs reduces the area consumed by our design however, their effect on DPA resistance has not been explored yet.

3.1. Pre-Charge

Pre-charge insertion is done using the technique introduced by Yu and Schaumont in [6]. An FPGA consists of programmable elements, so called Configurable Logic Block (CLB), and a network of programmable interconnects. In Xilinx Spartan 3 FPGAs a CLB is comprised of four slices, each containing two look-up tables (LUT) and two storage elements that can be used as either flip-flops or latches. It also contains two wide function multiplexers, fast carry logic and other miscellaneous elements. A flip-flop/latch is following every LUT. The pre-charge circuit which is implemented using an asynchronously cleared latch forces the output of the LUT to logic 0 during the pre-charge phase as shown in Fig. 1c). If a flip-flop is already used in the design then the pre-charge circuit should be inserted in a slice as near as possible to the flip-flop so that the routing between the two slices is kept at minimum.

3.2. Duplication

The first step in creating the complementary path is duplication of the original path. Before this can be done, appropriate CLB locations for the duplicate design must be chosen such that they have the same routing resources as the original design. Then the original design is copied (including routing), the components and nets are renamed and moved to the chosen locations.

3.3. Complementing the Logic

The complemented path should produce outputs inverse to that of the direct path. If $f(x)$ is the equation which defines a LUT in the direct path, then it's complimentary equation $g(\bar{x})$ is given by

$$g(\bar{x}) = \overline{f(\bar{\bar{x}})} = \overline{f(x)} \tag{1}$$

This principle is indicated in Fig. 1c). WDMs use LUTs and slice internal multiplexers. Equation 1 holds for LUTs however, for the slice internal multiplexers only the select lines should be inverted.

3.4. Secure Design Flow

Our design flow for implementing SDDL on FPGAs uses Xilinx ISE Design suite 10.1 and Perl scripts. It consists of three phases as show in Fig 2.

In the first phase, the single ended design is synthesized and implemented. Area constraints are applied which limit the design to one section of the FPGA fabric. It also specifies that the locations near registers should be left empty as they will be needed to insert pre-charge in the next phase.

In the second phase, the circuit description file from the first phase is converted into ASCII representation format with help of XDL (Xilinx Design language) tool . Perl scripts interpret the XDL file and insert pre-charge. Subsequently only Place and Route is executed.

In the third phase, the I/O connections are removed and the design is converted into XDL format. Perl scripts duplicate and complement the original circuit resulting in an SDDL implementation. However, as the I/O pins are still disconnected, and all the routing has to be preserved, we use re-entrant routing only.

4. TEST CIRCUIT IMPLEMENTATION AND ATTACK

4.1. Test Circuit

The Advanced Encryption Standard (AES) [16] is one of the most widely used block ciphers. It was designed to be resistant towards linear and differential cryptanalysis. However, unprotected AES hardware implementations are susceptible to DPA attacks. The test circuit for our proposed SDDL model is shown in Fig. 3. It consists of some of the main

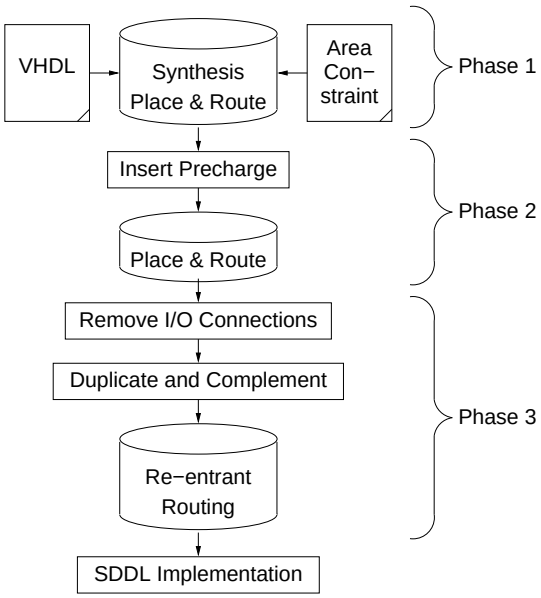

Fig. 2. SDDL Design Flow

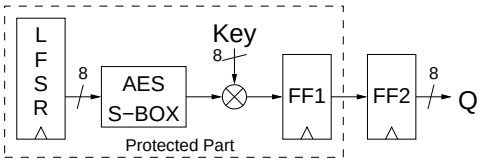

Fig. 3. Block Diagram of Test Circuit

building blocks of AES i.e. SBOX and key XORing. The test circuit allows us to replicate a DPA attack on AES on a smaller scale. An 8-bit LFSR is used to supply inputs to the SBOX. The output of the SBOX is XORed with key and stored in register FF1. The dashed line in Fig. 3 indicates the part of the circuit that we want to protect. The register FF2 drives the outputs of the chip and is implemented in I/O blocks (IOB).

4.2. SBOX implementations

The AES SBOX maps 8 input bits to 8 output bits using a substitution table. The Xilinx tool implements this function by default as a mixture of boolean logic and multiplexers. The usage of WDMs and the maximum size of the multiplexers can be controlled by the Xilinx ISE tool.

Each CLB is associated with one switch box which provides connections to the routing resources. An exception to these are local interconnections, so called fast interconnects, that exist between slices and between CLBs. These interconnects are used to create WDMs with sizes upto 32:1. Multiplexer of size upto 4:1 are supported within a single slice.

In our design we explore two AES SBOX implementations one using only 4:1 multiplexers and the other using 16:1 WDMs. The output of a 4:1 multiplexer can be pre-charged within the same slice. On the other hand, the 16:1 WDM consists of 4 slices and only the output of the last slice can be pre-charged. The input LUTs to the WDMs can contain negative logic and hence might produce glitches and disrupt the pre-charge wave. These signals travel through local interconnects and might make this design susceptible to DPA attacks. This leads to a tradeoff between security vs area.

4.3. Experimental Setup

We implemented our designs on a Xilinx Spartan 3e starter board containing a XC3S500eFG320-4 FPGA. We removed the capacitances of the core voltage net and connected it to an external regulated power supply. Power consumption is measured using a Tektronics CT-1 current probe and an Agilent DSO6054A oscilloscope, which has a bandwidth of 500MHz and samples at 4GSa/sec.

4.4. Attack Methodology

We use correlation attacks to test the effectiveness of our design [17]. The power model for the single ended cases is given by Equation 2. It calculates the Hamming distance between the previous output of the LFSR and the estimated following output. We estimate the following output of the LFSR for all possible key guesses. We use a different power model to mount a DPA attack on SDDL designs, given by Equation 3. The pre-charge phase sets all logic outputs to 0 therefore, the Hamming distance is computed between 0 and the estimated outputs of the LFSR for all possible key guesses. This is equal to their Hamming weight. We use Pearson's product moment correlation to compare the measured power and the power model [17].

5. RESULTS AND ANALYSIS

We have implemented four different designs of our test circuit. MUX-4 SE and MUX-16 SE are single ended implementations of our test circuit which use 4:1 multiplexer and 16:1 WDM for the AES SBOX respectively. MUX-4 SDDL and MUX-16 SDDL are two symmetrically routed SDDL designs of the said single ended. Table 1 shows the results of our implementations and measurements to disclosure (MTD) of the key.

The MUX-16 design is much smaller than the MUX-4 design and also has a shorter critical path delay. Both SDDL designs are little bit more than a factor 2 larger than the single ended designs. This is due to the fact that the outputs of the flip-flops need to be pre-charged. This increases the area by one slice per two flip-flops. The delay of the SDDL

$$Power_{guess} = \text{HD}\left(lfsr - output_{(i-1)}, (\text{SBOX}^{-1}(Key_{guess} \oplus Output))_i\right) \quad (2)$$

$$Power_{guess} = \text{HD}\left(0x00, (\text{SBOX}^{-1}(Key_{guess} \oplus Output))_i\right) \quad (3)$$

Table 1. Results of LFSR and SBOX implementations

Design	Slices	Delay (ns)	MTD
MUX-4 SE	134	9.08	1024
MUX-4 SDDL	283	18.16	$> 10,000$
MUX-16 SE	80	7.51	1024
MUX-16 SDDL	166	14.59	$> 10,000$

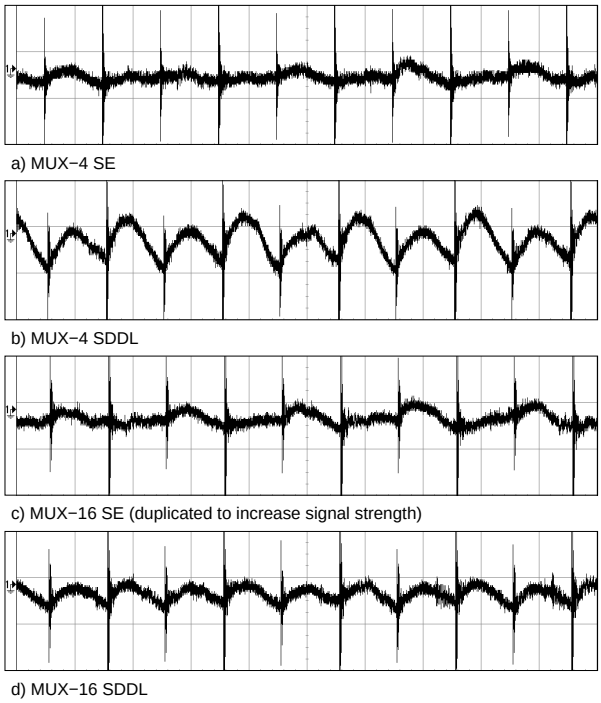

Fig. 4. Power Traces (5 mV/div, 1 μs/div)

designs is roughly 2 times larger than the single ended designs because all computations have to be performed during the evaluation phase which is half a clock cycle in length.

Figure 4 shows the power consumption traces for all four designs. The single ended wave forms have their peak near the rising edge of the clock. Both SDDL designs show lower peaks during pre-charge phase and higher peaks during the evaluation phase. It can also be clearly seen that the peaks of both SDDL designs are more uniform compared to the ones of the single ended. Therefore they are less correlated to the data being processed. This suggests that the SDDL designs are more difficult to attack.

We used a fixed 8-bit key value of 174 for all designs. The correlation plots between power guess and power measured for the MUX-4 SE implementation (Fig. 5) and the MUX-16 SE implementation (Fig. 6) taken over 1024 measurements show a sharp peak at the key guess 174. Therefore both single ended designs were broken.

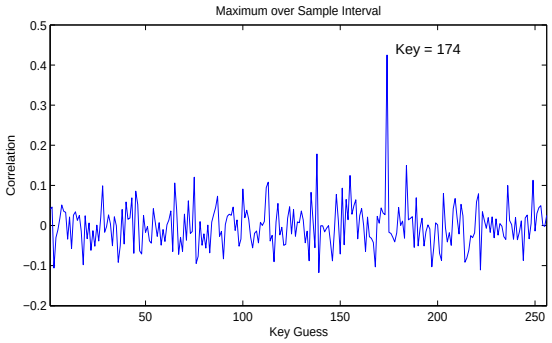

Fig. 5. DPA Attack on MUX-4 SE Implementation

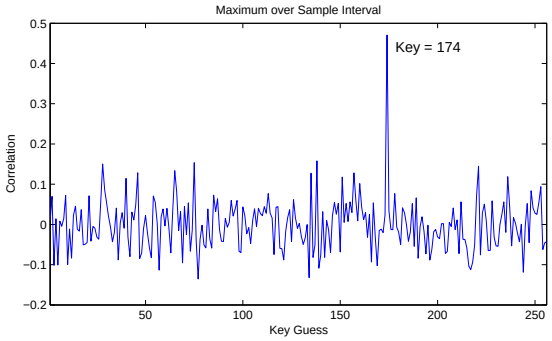

Fig. 6. DPA Attack on MUX-16 SE Implementation

The correlation plots for the SDDL design did not show a definite peak after 1024 measurements. Therefore, we had to take multiple sets of measurements. A set spans 1024 clock cycles (one measurement per clock cycle) each containing 800 samples. We sub-divide each clock cycle into 10 intervals. These intervals are shown as rows in Tables 2 and 3. For each interval we compute the maximum measured value. We correlate these maximum values of each set with the power model in Sect. 4.4. The correlation peaks of 8 sets and each interval are shown in the Table 2 and 3.

Both tables show that the correct key of 174 appears sporadically with no obvious pattern. Only if we look at more than 15 sets it slowly becomes apparent that 174 might be the correct key hence, we estimate the MTD to be larger than 10,000. We obtained the correct key for MUX-16 SDDL design using fewer measurements as compared to MUX-4 SDDL design because of non pre-charged signals passing through fast interconnects.

Table 2. Maximum Correlations for MUX-16 SDDL

Interval	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8
0	164	52	20	23	209	160	94	115
1	119	2	55	88	117	188	194	142
2	169	202	140	212	40	142	58	120
3	12	164	51	141	72	95	160	51
4	223	249	76	177	123	62	168	236
5	150	212	174	216	204	79	46	140
6	82	58	208	247	230	28	174	248
7	192	252	89	72	199	136	230	214
8	93	14	60	57	190	147	26	213
9	155	25	96	30	106	197	69	21

Table 3. Maximum Correlations for MUX-4 SDDL

Interval	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8
0	43	174	238	71	3	174	21	174
1	193	247	203	26	201	175	12	44
2	228	175	203	170	107	182	219	174
3	84	126	184	149	242	247	161	100
4	177	115	74	235	167	143	151	15
5	59	204	185	119	158	94	232	19
6	94	110	199	161	242	154	94	100
7	100	199	180	85	170	161	128	99
8	43	180	94	161	91	100	161	222
9	112	46	172	42	2	56	20	130

6. CONCLUSIONS AND FUTURE WORK

Perfect security does not exist. A high level of security is achievable but at the cost of a large area consumption. Our results show that we were able to achieve a moderate level of security by using our design flow at an area increase by a factor of just greater than 2. Thus showing that it is possible to apply Dynamic and Differential logic styles to low area implementations on FPGAs. Our SDDL can still be broken mainly due to glitches. For future work we plan to reduce the glitches and also examine the DPA resistance of other intrinsic features provided in FPGAs.

7. REFERENCES

[1] T. Tuan, S. Kao, A. Rahman, S. Das, and S. Trimberger, "A 90nm low-power FPGA for battery-powered applications," in *FPGA '06*, ACM/SIGDA. New York, NY, USA: ACM, 2006, pp. 3–11.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *CRYPTO'99*, ser. LNCS, vol. 1666. Berlin: Springer Verlag, Aug 1999, pp. 388–397.

[3] F.-X. Standaert, L. van Oldeneel tot Oldenzeel, D. Samyde, and J.-J. Quisquater, "Power analysis of FPGAs: How practical is the attack?" in *FPL 2003*, ser. LNCS, P. Y. K. Cheung, G. A. Constantinides, and J. T. de Sousa, Eds., vol. 2778. Berlin / Heidelberg: Springer, 2003, pp. 701–711.

[4] S. B. Örs, E. Oswald, and B. Preneel, "Power-analysis attacks on an FPGA – first experimental results," in *CHES 2003*, ser. LNCS, C. D. Walter, Çetin K. Koç, and C. Paar, Eds., vol. 2779. Berlin: Springer-Verlag, Sep 2003, pp. 35–50.

[5] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Proc. Design, Automation and Test in Europe (DATE'04)*. IEEE Computer Society, Feb 2004, pp. 246–251.

[6] P. Yu and P. Schaumont, "Secure FPGA circuits using controlled placement and routing," in *CODES+ISSS '07*. New York, NY, USA: ACM, 2007, pp. 45–50.

[7] S. Guilley, L. Sauvage, J. Danger, T. Graba, and Y. Mathieu, "Evaluation of power-constant dual-rail logic as a protection of cryptographic applications in FPGAs," in *SSIRI '08*. IEEE, Jul 2008, pp. 16–23.

[8] S. Guilley, L. Sauvage, J.-L. Danger, and P. Hoogvorst, "Area optimization of cryptographic co-processors implemented in dual-rail with precharge positive logic," in *FPL 2008*, U. Kebschull, M. Platzner, and J. Teich, Eds. IEEE, Sep 2008, pp. 161–166.

[9] L. Shang, A. S. Kaviani, and K. Bathala, "Dynamic power consumption in Virtex™-II FPGA family," in *FPGA '02*, ACM/SIGDA. New York, NY, USA: ACM, 2002, pp. 157–164.

[10] T. Popp and S. Mangard, "Masked dual-rail pre-charge logic: DPA-resistance without routing constraints," in *CHES 2005*, ser. LNCS, J. R. Rao and B. Sunar, Eds., vol. 3659. Heidelberg: Springer, 2005, pp. 172–186.

[11] D. Suzuki, M. Saeki, and T. Ichikawa, "DPA leakage models for CMOS logic circuits," in *CHES 2005*, ser. LNCS, J. R. Rao and B. Sunar, Eds., vol. 3659. Heidelberg: Springer, 2005, pp. 366–382.

[12] D. Suzuki and M. Saeki, "Security evaluation of DPA countermeasures using dual-rail pre-charge logic style," in *CHES 2006*, ser. LNCS, L. Goubin and M. Matsui, Eds., vol. 4249. Heidelberg: Springer, 2006, pp. 255–269.

[13] P. Schaumont and K. Tiri, "Masking and dual-rail logic don't add up," in *CHES 2007*, ser. LNCS, P. Paillier and I. Verbauwhede, Eds., vol. 4727. Heidelberg: Springer, 2007, pp. 95–106.

[14] S. Guilley, S. Chaudhuri, L. Sauvage, T. Graba, J.-L. Danger, P. Hoogvorst, Vinh-Nga, and M. Nassar, "Place-and-route impact on the security of DPL designs in FPGAs," in *HOST 2008*. IEEE, 2008, pp. 26–32.

[15] S. Chaudhuri, S. Guilley, P. Hoogvorst, J.-L. Danger, T. Beyrouthy, A. Razafindraibe, L. Fesquet, and M. Renaudin, "Physical design of FPGA interconnect to prevent information leakage," in *ARC 2008*, ser. LNCS, R. Woods, K. Compton, C. Bouganis, and P. C. Diniz, Eds., vol. 4943. Berlin / Heidelberg: Springer, 2008, pp. 87–98.

[16] *Advanced Encryption Standard (AES)*, National Institute of Standards and Technology (NIST), FIPS Publication 197, Nov 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[17] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems – CHES 2004*, ser. Lecture Notes in Computer Science, vol. 31. Berlin / Heidelberg: Springer, Aug 2004, pp. 135–152.