Evaluation of Power-Constant Dual-Rail Logic as a Protection of Cryptographic Applications in FPGAs

Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Tarik Graba, Yves Mathieu

▶ **To cite this version:**

Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Tarik Graba, Yves Mathieu. Evaluation of Power-Constant Dual-Rail Logic as a Protection of Cryptographic Applications in FPGAs. Secure System Integration and Reliability Improvement, Jul 2008, Yokohama, Japan. pp.16-23, 10.1109/SSIRI.2008.31 . hal-00259153v5

HAL Id: hal-00259153
https://hal.science/hal-00259153v5

Submitted on 20 Nov 2009

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.

Evaluation of Power-Constant Dual-Rail Logic as a Protection of Cryptographic Applications in FPGAs

Sylvain Guilley*, Laurent Sauvage*, Jean-Luc Danger*, Tarik Graba*, Yves Mathieu*

* Institut TÉLÉCOM / TÉLÉCOM ParisTech (CNRS LTCI, UMR 5141),
Département COMELEC, 46 rue Barrault,
75 634 PARIS Cedex 13, FRANCE.
Email: < sylvain.guilley@telecom-paristech.fr >

***Abstract*—FPGAs are often considered for high-end applications that require embedded cryptography. These devices must thus be protected against physical attacks. However, unlike ASICs, in which custom and backend-level counter-measures can be devised, FPGAs offer less possibilities for a designer to implement counter-measures. We investigate "wave dynamic differential logic" (WDDL), a logic-level counter-measure based on leakage hiding thanks to balanced dual-rail logic.**

First of all, we report a CAD methodology for achieving WDDL in FPGA. An experimental security evaluation of the DES encryption algorithm in WDDL shows that the usage of positive logic is mandatory to resist to straightforward attacks.

Second, we discuss how to reduce the size overhead associated with WDDL. The efficiency of some synthesizers is assessed. In the case of DES, we provide with an original heuristic to obtain substitution boxes smaller than those generated automatically with legacy ASIC synthesizers.

Keywords: FPGA, security, side-channel attacks, attacks mitigation, power-constant logic, positive dual-rail logic.

I. Introduction

Modern cryptographic algorithms have been devised to be implemented efficiently in software. This has been a requirement for the AES [16] candidates, and is still a premium for the SHA-3 [17] contest. However, some applications require a fast execution that software can't always provide. Typically, nomadic mobile devices demand a hardware accelerator to meet the required execution speed with a low power budget. This problematic is now well identified, and has led to a wide field of researches on "lightweight embedded cryptography".

The market for embedded cryptography can be divided into two categories. On the one hand, large volume products, such as commercial identification devices (product/merchandise tracing, credit cards, e-purses or e-money, TPMs, telco SIM cards, pay-TV, transport tickets, customer loyalty programmes) or national infrastructures (passport, identity cards, driving license or healthcare registration), are implemented in smartcards or RFID tags. These devices take advantage of the ASIC technology, which is relevant from an economical point of view. The deployment of such services indeed implies the issuing of millions of electronic devices. Consequently this increases the risk of fraud as product counterfeiting or illegal activities.

On the other hand, low volume secure devices are also needed for specific markets (satellites, critical premises access control, secure login on information systems, watermarking camcorders, IDS, VPN laptop cards, crypto tokens, encrypted hard drives or dongles, IPsec routers), governmental or military (secure terminals, PDAs, network facilities). For these high-end applications, FPGAs are often considered the most suitable choice. Despite a higher unitary cost, FPGAs do not suffer from a high investment (masks preparation and complex manufacturing processes). The FPGAs are readily available COTS and provide a short time-to-market development cycle. In addition, the reconfigurability of FPGAs is an advantage over ASICs: any bug or security vulnerability can, if necessary, be fixed after the products are operationally deployed. This agility is a predominant feature, because these applications are extremely sensitive. They must resist assaults from strong attackers, such as organized mafias or state intelligence services, motivated by large scale fraud or economic/warfare supremacy.

Both cryptographic ASICs and FPGAs must be certified with a high evaluation assurance level, as specified for example by FIPS 140 or the ISO common criteria. These certifications imply that the "target of evaluation" be tamper resistant. An overview of security issues in FPGAs can be found on Saar Drimer's homepage [8]. Concerning bitstream protections FPGA manufacturers already propose solutions based on embedded bitstream deciphering with a local key storage. An overview is given in [29]. For sound trusted computing devices, that are properly architectured with a robust combination of algorithms using a suitable key length, entropic initialization vector (IVs), *etc.*, the most evident threat arises from attacks on the implementation itself. Implementation-level attacks fall into two categories, depending whether they are active or passive. Active attacks consist in either injecting faults so as to gain information from the device corrupted execution results [11]. Usual counter-measures consists in detecting errors, using codes or redundancy. Passive attacks, also called side-channel attacks (SCAs), consist in simply observing the devices' emanations while it is performing a cryptographic operation. Consequently, the device can't know that it's under analysis. The attack consists either in computing a correlation coefficient between the acquired traces

and the expected dissipation according to a key hypothesis (*e.g.* SPA, DPA, CPA, EMA) or in the consultation of a pre-characterized database (*e.g.* template attacks (TA) [1], [3], [5]). Common counter-measures against observation attacks consist in randomizing the execution, using clock jitter, dummy or decoy clock cycles, or in blinding intermediate data words. The goal is to make an statistical treatment irrelevant. Another customary technique consists in balancing the circuit's activity, so as to make any dissipation data-independent. This approach alleviates the need for an high quality randomness source, but in return demands a strong effort in the balancing process.

Many protections against active and passive attacks have been proposed for ASICs. The logical-level countermeasures can be ported as such on FPGAs, since FPGAs are just a reconfigurable ASICs. On the other hand, backend-level countermeasures are more difficult to adapt to FPGAs, because the cells layout and the routing resources are hardwired. Indeed, some FPGA families have been strengthened for remaining functional in harsh environments, such as space, nuclear plants, or radiative medical apparati. They are particularly robust against Single Even Upset (SEU) and/or signal integrity at the I/O interfaces. Protections against active attacks can take advantages of robust FPGAs strategies at RTL level like register triplication or by using sensors to detect an abnormal event. But concerning passive attacks FPGAs aren't intrinsically immune. It seems on the contrary that FPGAs have a propension of leaking much information [23]. As compared to regular ASICs, the interconnection network is extremely dissipative, because it consists of active switches and long distances between logic cells.

We study in this paper a power consumption balancing strategy called "wave dynamic differential logic" (WDDL [26]) that is well suited for FPGAs. Its principle is to duplicate the netlist into a true and a false part, that share the same topology (interconnection graph). The graph is devised such that if any gate of one network switches, then the sibling gate of the dual network does not, and *vice-versa*. This way, from a macroscopic standpoint, the activity of the circuit is constant. This is at least true at the logical level, *i.e.* at the first order.

The rest of the paper is organized as follows: Section II details the methodology used to achieve a WDDL netlist, and gives some indications on the overhead caused by switching from insecure to secure netlists. Then we report in section III experimental security improvements reached by two types of WDDL netlists (non-positive and positive) over an unprotected reference. The section IV analyzes the performance of some synthesizers in mapping into logical gates the most complicated parts of a cryptographic algorithm, namely the substitution boxes. Finally, the section V concludes on the efficiency and the cost of protecting FPGAs against SCAs using a power-constant strategy and provides some suggestions for improvements.

II. Fitting WDDL into FPGAs

A. State-of-the-Art about Dual-Rail Logic in FPGAs

Kris Tiri reports in [26]–[28] implementation methods for WDDL in FPGAs. Other types of logic use also differential logic like MDPL which is a masked logic introduced in [19]. Despite the great advantages provided by differential logic like WDDL or MDPL, it has been proved in [20], [25] that this logic type has still little imbalance due to early evaluation or technological bias. However no successful attacks based on differential logic on FPGAs has been reported so far. Secured designs in FPGAs based on masked logic are also described by François-Xavier Standaert in [24].

The seminal publication [26] suffers a large area overhead due to the restriction to the minimal library consisting of only {`INV`, `AND`, `OR`} (3 gates). In [28], a clustering method allows to use all `AND-OR` combinations (166 gates in LuT4 FPGAs). The method is shown to be automatable thanks to an ASIC synthesizer in [27].

WDDL in FPGAs has already been studied by Pengyuan Yu. Separated Dynamic Dual-Rail Logic (SDDL), described in [30], is shown experimentally to fail because of glitches caused by a race between a global signal (precharge) and local signals (differential data pairs). Double WDDL (DWDDL) introduced in [30] is definitely secure. It is the first design-agnostic method to obtain secure implementations in FPGA from both a logical and physical point of views. The only "industrial acceptance" issue about DWDDL is that it quadruples at least the implementation area.

The DWDDL remains nevertheless weak because the functional design and the complementary one are far one from each other. One previous paper [21] shows that an integrated antenna of about 40 μm extension can measure EM emanations selectively. Such an accuracy probably allows to separate the radiations of the functional from the fake complementary *alter ego* WDDL module.

In the rest of this section, we present a case-study on the DES [15] cryptographic algorithm. As such, DES is no longer suitable for block encryption, because its keylength of 56 bit is too short [10]. Massively parallel or networked machines can indeed exhaust the 2^{56} keys in a few days. This certainly compromises DES ciphertexts, and definitely ruins any hope of forward secrecy. AES [16] is the successor of DES with a key length at least equal to 128 bit. However, when DES is used as DESX (the standard DES is sandwitched between an input and an output Vernam masking of the plain- and ciphertext), or as triple DES (as described in appendix 2 of [15]), it is perfectly secure. It has been selected, amongst others, for the international passport and is still used in banking applications for instance. The main appeal of DES lays in its compactness when implemented in hardware. We have therefore used a fully-fledged DES (achieving simple and triple DES, with all specified modes of operations), whose architecture is described in [12]. As detailed later on in Sec. III, we managed to fit several DES instances in a single FPGA.

Our goal is to evaluate WDDL (positive or not) on a

real embedded cryptographic application. We emphasize that the results presented in this paper are the first experimental implementations and attacks on a full-featured cryptographic system-on-chip equipped with a DES processor protected by WDDL. To be exact, we present WDDL and WDDL+ experimental results at the logical level only. The backend has been delegated without constraints automatic "partition, place-and-route" tools. However, given the symmetry of WDDL and WDDL+ netlist, we assume that backend implementation does not drastically deteriorates the logical symmetry between the dual networks. Albeit intuitive, this hypothesis is nonetheless to be verified on more accurate setups. Our experience, detailed in the sequel, is that the most secure power-constant logic (WDDL+) is already fairly strong against straightforward attacks, without any supplementary backend-level intervention.

B. Design Secure Partitioning

In DES, the control is independent of the data. Whatever the key or the plaintext, the algorithm consists in sixteen consecutive rounds. It is thus only required to secure the datapath (made up of the message and the key paths). The control part is never attacked because it conveys no useful information.

Therefore, the datapath is implemented in WDDL, whereas the control remain regular. The source code of the cryptographic engine can be written in an HDL language in behavioral (*aka* RTL) style. Only the ad hoc converters, that ensure the transcoding between single and dual-rail logic blocks, are described in a structural style.

C. Synthesis in WDDL

On the one hand, the non-secure blocks, typically the control of DES, can be synthesized with the toolchain that comes with FPGAs (Altera `quartus` or Xilinx `ise`). Alternatively, generic synthesizers, such as `precision` by Mentor Graphics or `Synplify Pro` by Synplicity, can be fair FPGA vendor independent substitutes.

On the other hand, secure blocks must pass through a more specific synthesis process. Power-constant dual-rail logic, as the DES data- and key-path, must be synthesized carefully. We base our WDDL design on the following dualization of a look-up-table (LuT) f: the dual gate g of f satisfies:

$$g(x) \doteq \overline{f(\overline{x})}\,. \tag{1}$$

This corresponds to a `0x0` spacer for the netlist precharge. For the spacer to propagate, LuTs must satisfy the wave condition:

$$\text{LuT}(000\cdots0) = 0 \qquad \text{and} \qquad \text{LuT}(111\cdots1) = 1\,. \tag{2}$$

For the sake of example, we illustrate how Eq. (1) applies on Altera low-end FPGAs. The encoding for the Altera Stratix $4 \rightarrow 1$ LuTs is given in Tab. I for some representative gates.

Therefore, the dual of a LuT4 gate, such as the one described in Fig. 1, is obtained by replacing the look-up-table configuration mask:

- `lut_mask="FFFE"; /* [OR4] Direct */` by
- `lut_mask="8000"; /* [AND4] Dual */`.

TABLE I
LUT MASKS ENCODING FOR ALTERA STRATIX FPGAS.

	OR4 $a+b+c+d$	AND4 $a\cdot b\cdot c\cdot d$	MUX2 $a\cdot\overline{d}+b\cdot d$	MUX2N $a\cdot d+b\cdot\overline{d}$
$d\,c\,b\,a$	FFFE	8000	CCAA	AACC
0 0 0 0	0	0	0	0
0 0 0 1	1	0	1	0
0 0 1 0	1	0	0	1
0 0 1 1	1	0	1	1
0 1 0 0	1	0	0	0
0 1 0 1	1	0	1	0
0 1 1 0	1	0	0	1
0 1 1 1	1	0	1	1
1 0 0 0	1	0	0	0
1 0 0 1	1	0	0	1
1 0 1 0	1	0	1	0
1 0 1 1	1	0	1	1
1 1 0 0	1	0	0	0
1 1 0 1	1	0	0	1
1 1 1 0	1	0	1	0
1 1 1 1	1	1	1	1

```
// One OR4 gate programmed in a stratix logic cell:
stratix_lcell \y~2_I (
.dataa(d),
.datab(b),
.datac(c),
.datad(a),
.combout(\y~2));
defparam \y~2_I.operation_mode        = "normal";
defparam \y~2_I.synch_mode            = "off";
defparam \y~2_I.register_cascade_mode = "off";
defparam \y~2_I.sum_lutc_input        = "datac";
defparam \y~2_I.lut_mask              = "FFFE";
defparam \y~2_I.output_mode           = "comb_only";
```

Fig. 1. Example of an Altera VQM (Verilog [IEEE 1364] Quartus Mapping) Stratix LuT.

Now, the wave propagation constraint (2) further implies that not all gates are suitable for WDDL implementations. Indeed, amongst the 2^{2^n} gates (n bit $\rightarrow$ 1 bit), only one quarter (*i.e.* 2^{2^n-2}) is suitable for WDDL. The synthesis is thus less efficient than with a complete library. This issue is further discussed in Sec. IV. Now, to reduce the available gates, we need to find a way to constraint the synthesizer to use only some specified cells. This possibility exists only for ASIC synthesizers. Due to their internal heuristics, these tools require at least:

- one flip-flop (DFF — named `fpga_fdr`),
- one invertor (IV — named `fpga_iv`) and
- one two-input gate (say AN2 — named `fpga_an2`).

A typical duplication for a WDDL DES datapath netlist is illustrated in Tab. II. Regular $4 \rightarrow 1$ LuTs are named `fpga_lut4`.

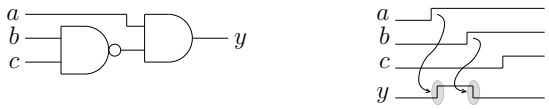
*D. Synthesis in WDDL or in "Positive" WDDL (*aka *WDDL+)*

The property (2) is not enough to ensure the confidentiality of the data. The design must also be free of any glitch. As a consequence, every tabulated function must be positive. Otherwise, glitches can show up. As glitches are data-dependent,

TABLE II
DUPLICATION OF A SENSITIVE WDDL NETLIST (DES DATAPATH).

Resource	Single-ended	Duplication (see (1))	WDDL
fpga_fdr	184	→ ×4 →	736
fpga_iv	240	→ ×0 →	0
fpga_an2	202	→ ×2 →	404
fpga_lut4	1499	→ ×2 →	2998
fpga_⋆	2125		4138

LuT3 #1 (not WDDL-compliant, since (2) is violated):



LuT3 #2 (WDDL-compliant, since (2) is satisfied):

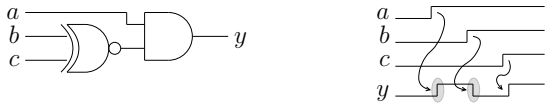


Fig. 2. Glitches in two non-positive gates.

they are indeed a security weakness. For instance, in the Fig. 2, the functions $f(a,b,c) = a \cdot \overline{(b+c)}$ or $a \cdot \overline{(b \oplus c)}$ are shown to glitch. The non-functional transitions are depicted with an ellipsis symbol: ⬮. The first function is a counter-example taken by Kris Tiri in the figure 8 of [28]. This gate propagates the $(0,0,0)$ spacer but not $(1,1,1)$ and therefore does not respect condition (2). The second gate is WDDL-compliant since it does propagate the two spacers, but it still glitches. If the function f is the true part of a WDDL pair of gates, then Fig. 2 represents the transition from the precharge state to the value $(a,b,c) = (1,1,1)$. The functions f glitch iff their input a toggles before their input b or c, which introduces a data-dependent syndrome. Put differently, a DPL is no longer "power-constant" in the presence of glitches.

E. Overhead Incurred by WDDL Netlist Styles

We studied the performance of the WDDL and WDDL+ DES modules. All the DES modules share the same 8-bit VCI interface with an addressing range of 8 bits. They all embed 2,048 memory bits (a 256 bytes RAM); their area and maximal frequency is given in Tab. III. The throughput is computed in simple DES-ECB with 56-bit key and in triple-DES-OCB with 112-bit key modes of operation. To simplify the estimation, we assume that the memory is infinite, and thus neglect the initial latency caused by the loading of the key and the final latency associated with the last block saving in RAM. The encryption of one block lasts 16 clock cycles for the single-ended DES module, but 2×16 for the dual-rail modules.

III. EXPERIMENTAL EVALUATION OF WDDL SECURITY

A. State-of-the-Art about Attacks on FPGAs

The first attack on an FPGA (a handmade Xilinx Virtex 800 board) is reported in 2003 [18]. The impact of the RTL architecture on the leakage is studied next year [23]. Some improvements, made possible by signal pre-processing (such as filtering and averaging), are presented in [22]. The overall conclusion of these studies is that unprotected implementations of FPGAs are vulnerable to side-channel attacks, even if their dissipation process is different from that of ASICs. Some acquisition improvements have been done in 2007 [14]. On independent acquisition banks, an attack on AES programmed in an Altera Cyclone, is implemented successfully by exploiting EMA signals [4]. The first attack on a complete system-on-chip embedding a cryptoprocessor is reported in [6]. This study shows that even small cryptographic applications are at risk in FPGAs.

TABLE III
PERFORMANCES OF THE REGULAR AND THE TWO DUAL-RAIL DES MODULES SYNTHESIZED BY CADENCE BGX_SHELL.

Implementation	Single-ended DES	WDDL DES	WDDL+ DES
Area	1,248 LEs	4,736 LEs	6,038 LEs
Max. Frequency	74.95 MHz	68.65 MHz	55.85 MHz
DES-ECB speed	300 Mbit/s	137 Mbit/s	111 Mbit/s
3DES-OCB speed	99 Mbit/s	45 Mbit/s	37 Mbit/s

B. Evaluation Methodology

We have embedded a secured DES processor into the SecMat [7] system-on-chip (Soc). This setup represents a realistic usage of FPGAs as security devices. Additionally, the SoC was equipped with an unprotected DES processor, to serve as a reference.

C. Evaluation Board and its Customization

We chose a Parallax board, for its simplicity, and also because it can accommodate the whole SecMat SoC along with the DES co-processors. We illustrate the synthesis on the example of Altera, but the principle could be applied as well to any other tool that can read a structural netlist. The board is shown in Fig. 3. On the bottom left corner we can notice the small power shunt circuitry based on a coil-resistor impedance. The advantage of this device is that no coupling capacitors have to be removed and it allows to grab transient currents with enough sensitivity. This small intrusion allows to perform acquisitions with a differential probe at each side of the coil. The figure 4 shows a zoom on the "power drop" measurement ancillary printed circuit board (PCB).

D. Attacks on Experimental Power Traces

We attack the regular module with the Hamming distance model. This model corresponds to the CMOS power dissipation which is produced by a signal transition. The attack of the WDDL circuit considers the Hamming weight model which indicates that the dissipation corresponds to the signal level and not the transition. This is due to the fact that the Hamming distance between the precharge state (full zeroes) and the evaluation state degenerates into a Hamming weight. Two correlation power attacks, Differential Power Analysis (DPA)

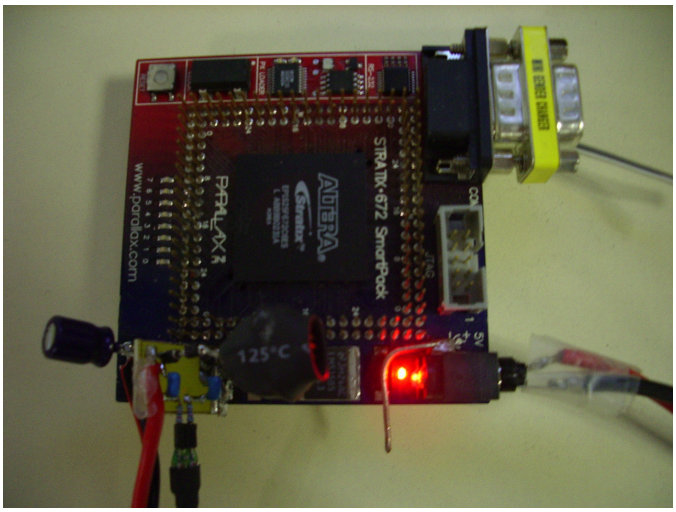


Fig. 3. Board used to acquire side-channel information on the Stratix FPGA.

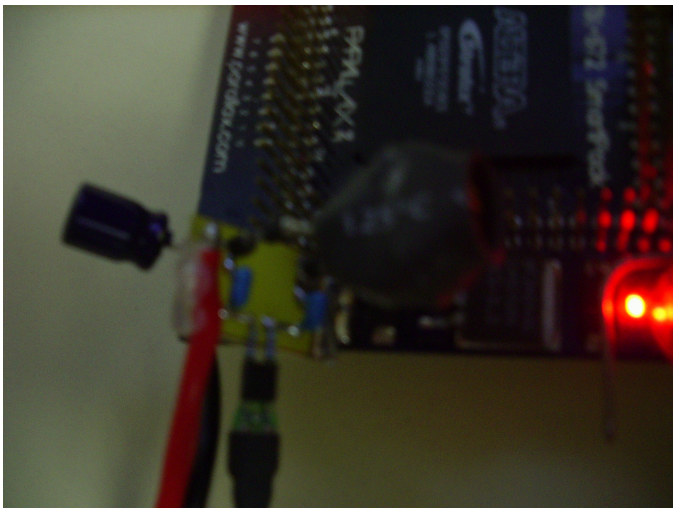


Fig. 4. Customization of the commercial board shown in Fig. 3 for better power measurement.

and Correlation Power Attack (CPA), as described in [13], are performed. A first acquisition campaign of 67,753 traces corresponding to the reference DES coprocessor activity, was performed. This module has been completely broken, as shown in Table IV, by attacking during either the first round or the last round of DES. Table IV shows the maximal correlation (for DPA) or covariance (for CPA) levels to get a reference to compared with the levels of the WDDL protected DES. The SNR indicator illustrates the ratio between the level obtained with the right key and those from the strongest wrong key. The CPA traces for every sbox are shown in Fig. 5. Similar attacks are led on protected DES modules. In a view to enhance the attacker's strength, we focused the acquisitions (cadenced at 20 Gsamples/s) around the first round of DES. **Three** sboxes of the non-positive WDDL module are recovered (see Tab. V). The correlation levels and the SNR of the attacks are far

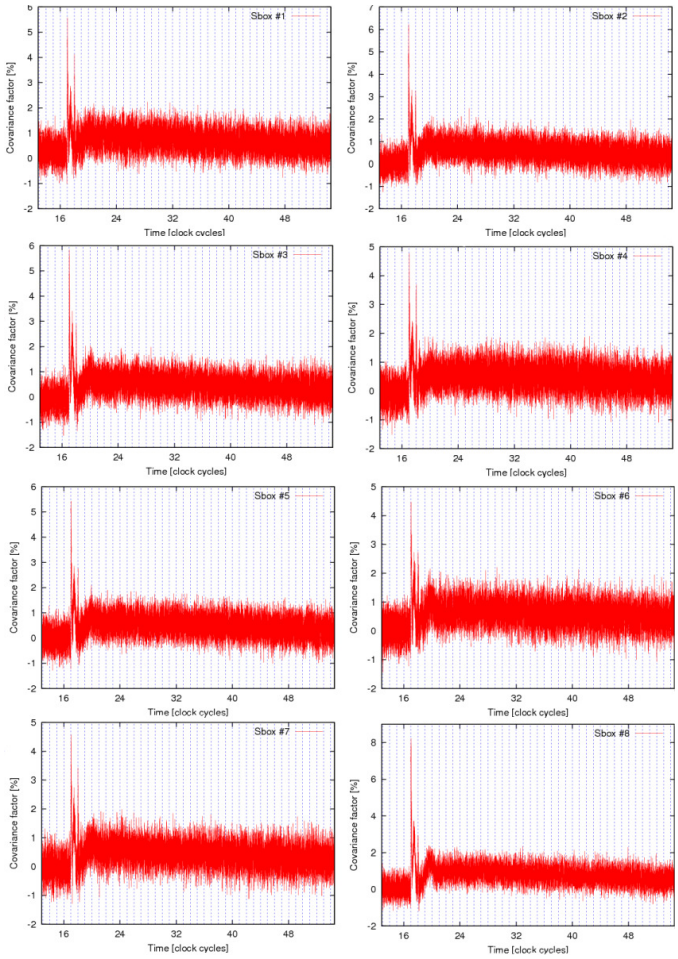


Fig. 5. Covariance factors obtained for the correct key hypothesis when attacking the first round of the reference DES module eight sboxes.

below those obtained with the regular DES. The CPA is more efficient than the DPA because the correlation is normalized and allows to recover the signal in a noisier environment. Only **one** of the WDDL+ sboxes is broken after 123,743 traces (see Tab. VI). The positive logic used in WDDL+ to remove the glitches provides then a greater robustness. However the fact one sbox can be broken shows that the countermeasure is not fully efficient and more sboxes could be broken by enhancing the acquisition platform sensitivity or the attack algorithm.

IV. Synthesis Optimization of WDDL+ Netlists

A. *Synthesis with Legacy Tools*

Some substitution boxes are synthesized with various synthesizers. For DES and Kasumi (Feistel ciphers), the sboxes are given according to the standard. For AES (substitution-permutation network), the sbox and its inverse are studied.

The RTL description is tabulated. We chose this solution to avoid any segregation between the sboxes based on their internal structure. More compact netlist could be obtained with description that takes advantage of the mathematical description of the sboxes of Kasumi or AES.

TABLE IV
ATTACK OF THE REGULAR DES MODULE.

(a) Number of measurements to disclose (MTD) the key								
Analysis \ Sbox #	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
DPA on the first round	9,877	3,065	7,983	10,473	7,216	11,103	7,678	2,257
DPA on the last round	11,540	4,150	4,861	6,031	4,580	2,881	4,539	19,743
CPA on the first round	7,480	3,079	7,252	9,605	6,546	11,079	5,442	2,095
CPA on the last round	11,804	3,408	4,880	6,027	3,433	2,874	3,994	19,988

(b) Maximal correlation in μV (DPA) / covariance in % (CPA) factors								
Analysis \ Sbox #	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
DPA on the first round	46	51	48	39	44	37	37	66
DPA on the last round	36	64	44	60	48	55	59	33
CPA on the first round	5.58	6.23	5.84	4.78	5.42	4.47	4.58	8.23
CPA on the last round	4.44	7.86	5.48	7.42	6.03	6.70	7.37	4.02

(c) Maximal SNR								
Analysis \ Sbox #	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
DPA on the first round	6.23	6.41	6.34	6.76	6.37	5.29	4.56	8.07
DPA on the last round	7.90	6.50	5.02	6.75	5.69	6.18	4.04	6.60
CPA on the first round	6.22	6.41	6.23	7.03	6.37	5.29	4.57	8.08
CPA on the last round	7.90	6.48	5.03	6.75	5.70	6.22	4.04	6.00

TABLE V
ATTACK (ZOOMED ON THE FIRST ENCRYPTION ROUND) OF THE WDDL DES MODULE (NON-POSITIVE).

(a) Number of measurements to disclose (MTD) the key								
Analysis \ Sbox #	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
DPA on the first round	—	—	—	224,018	—	—	—	—
CPA on the first round	99,943	—	193,028	73,524	—	—	—	—

(b) Maximal correlation in μV (DPA) / covariance in % (CPA) factors								
Analysis \ Sbox #	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
DPA on the first round	17	19	12	4	11	18	12	12
CPA on the first round	0.72	1.23	0.56	0.34	1.34	0.79	0.47	1.32

(c) Maximal SNR								
Analysis \ Sbox #	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
DPA on the first round	2.66	2.67	2.49	3.94	3.40	2.56	3.73	2.97
CPA on the first round	3.19	3.83	3.35	4.04	2.74	2.49	3.14	3.81

TABLE VI
ATTACK (ZOOMED ON THE FIRST ENCRYPTION ROUND) OF THE WDDL+ DES MODULE (POSITIVE).

(a) Number of measurements to disclose (MTD) the key								
Analysis \ Sbox #	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
DPA on the first round	—	—	—	—	123,743	—	—	—
CPA on the first round	—	—	—	—	—	—	—	—

(b) Maximal correlation in μV (DPA) / covariance in % (CPA) factors								
Analysis \ Sbox #	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
DPA on the first round	13	12	11	13	11	12	12	13
CPA on the first round	1.43	1.44	1.27	1.40	1.40	0.59	0.74	1.50

(c) Maximal SNR								
Analysis \ Sbox #	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
DPA on the first round	3.46	4.97	3.62	3.85	3.59	2.53	3.44	3.32
CPA on the first round	3.16	4.22	3.20	4.72	3.07	3.42	3.53	3.32

TABLE VII
VARIOUS SUBSTITUTION BOXES AREA IN STRATIX ALTERA FPGA.

Single	DES								Kasumi		AES	
-ended	S1	S2	S3	S4	S5	S6	S7	S8	S7	S9	S	S^{-1}
LuT4	24	21	24	24	24	21	21	24	69	29	206	205
LuT3	0	1	0	0	0	2	2	0	9	11	2	3
LuT2	0	0	0	0	0	0	0	0	1	2	0	0
LuT1	0	0	0	0	0	0	0	0	0	0	0	0
LuT0	0	0	0	0	0	0	0	0	0	0	0	0
Total	24	22	24	24	24	23	23	24	79	42	208	208

TABLE VIII
VARIOUS SUBSTITUTION BOXES AREA IN VIRTEX-II XILINX FPGA.

Single	DES								Kasumi		AES	
-ended	S1	S2	S3	S4	S5	S6	S7	S8	S7	S9	S	S^{-1}
LuT4	15	12	18	10	15	15	12	15	71	132	131	131
Slices	8	6	9	5	8	8	6	8	35	71	66	66
Tiri [27]	14	15	14	8	13	15	15	13	30	32	174	—

The results are listed in Tab. VII for the Stratix using `quartus` version 7.1. Similar results for the Virtex-II using `ise 9.2i` are reported in Tab. VIII. The performance of `bgx_shell` (64-bit version v05.15-s095+1) is assessed in Tab. IX. The ASIC synthesizer `rc` (64-bit version v06.10-s017_1) is not as good as `bgx_shell`, as shown in Tab. X. The results obtained with ASIC synthesizers read as follows:

- The library "plain" contains all the 2^{2^n} cells: it is meant to provide a lower-bound for the area.
- The library "WDDL" contains all the 2^{2^n-2} cells that satisfy (2). After synthesis, the inverters are removed, and the number of remaining gates is doubled.
- The library "WDDL positive" (WDDL+) contains the positive cells that, in addition, propagate 0 and 1, as per (2). The number of these functions is equal to the Dedekind numbers $M(n)$ [9] minus two (the two constant functions zero and one). Although Dedekind first considered this question in 1897, there is still no concise closed-form expression for $M(n)$. There are 4 (resp. 18, 166, 7 579, 7 828 352) WDDL+ gates with two (resp. 3, 4, 5, 6) inputs.

TABLE XI
SYNTHESIS SCRIPT FOR THE ASIC SYNTHESIZER `BGX_SHELL`.

```
do_optimize -flatten on -effort high \
            -priority area -max_area 0;
```

TABLE XII
SYNTHESIS SCRIPT FOR THE ASIC SYNTHESIZER `RC`.

```
synthesize -to_mapped -csa_effort high -eff high;
remap -start 9 -end 11 $top_level;
synthesize -to_mapped -eff high -incr;
```

The ASIC synthesizers are tuned to spend the maximal effort on the area optimization. The synthesis TCL command for `bgx_shell` is given in Tab. XI and for `rc` in Tab. XII.

B. A Novel Heuristic to Compact DES WDDL+ Sboxes

We present a heuristic for achieving a better synthesis for $6 \rightarrow 4$ substitution boxes, such as that of DES. The goal is to reach a compact netlist in LuT4, restricted to WDDL+ gates. The heuristic consists in:

1) using a decoder for the two most significant bits (MSB) $(a[5], a[4])$ and in
2) keeping a multiplexor-tree architecture for the remaining least significant bits (LSB) $(a[3], a[2], a[1], a[0])$.

The head part consists in all the possible two-input functions. They are synthesized in LuT4, fed by $(a_t[5], a_t[4])$ and $(a_f[5], a_f[4])$. There are 16 of them, amongst which 6 are trivial: the constants 0 and 1, the identities $a[5] = a_t[5]$ and $a[4] = a_t[4]$ and the inverses $\overline{a[5]} = a_f[5]$ and $\overline{a[4]} = a_f[4]$. Hence only 10 2-input gates lead to non-trivial LuT4s instances. All these 10 LuTs are instantiated, and shared between the 4 output bits tail logic.

The tail part consists in 2 multiplexor trees (one true and one false) of 16 inputs and 4 inputs. This tree can be synthesized with $2 \times 4 \times (8 + 4 + 2 + 1) = 120$ two-input multiplexors. Now, multiplexors as such are not positive. However, a two-input multiplexor can be implemented in positive logic if the selection signal is available both plain and inverted. This is actually the case, because the selection s is $a[i]$, $i \in [3:0]$, and that both $a_t[i]$ and $_f[i]$ are available. The multiplexing of inputs a and b with s, resulting in $y \doteq a \cdot \overline{s} + b \cdot s$, fits into a single positive LuT4 as: $y_t = a_t \cdot s_f + b_t \cdot s_t$ (the four inputs are s_f, s_t, a_t and a_f).

The overall construction requires $120 + 10 = 130$ positive LuT4s. This figure, albeit close to that obtained by the `bgx_shell` and `rc` synthesizers, is better for all the eight DES sboxes (refer to the last line of Tab. IX and X). Moreover, it is likely that using some peculiarities of the $6 \rightarrow 4$ sboxes, the 130 LuT4s score can be improved. For instance, it might happen that some inputs of the tail 4-input multiplexor-tree be

TABLE IX
VARIOUS SUBSTITUTION BOXES AREA IN LUT{2,3,4} WITH CADENCE BGX_SHELL SYNTHESIS.

Library	DES								Kasumi		AES	
type	S1	S2	S3	S4	S5	S6	S7	S8	S7	S9	S	S^{-1}
LuT2 plain	149	154	140	149	140	164	141	150	393	1462	856	854
LuT2 WDDL	304	312	290	300	286	332	288	300	788	2934	1716	1708
LuT2 WDDL+	304	312	290	300	286	332	288	300	788	2934	1716	1708
LuT3 plain	87	94	80	85	83	100	85	88	235	865	506	513
LuT3 WDDL	174	188	160	174	170	202	170	176	470	1730	1012	1026
LuT3 WDDL+	186	194	174	178	172	204	184	178	478	1766	1024	1028
LuT4 plain	60	66	57	57	56	75	58	65	171	608	371	371
LuT4 WDDL	118	132	114	116	116	148	116	126	342	1214	742	742
LuT4 WDDL+	138	144	134	134	136	150	134	134	360	1258	758	748

TABLE X
VARIOUS SUBSTITUTION BOXES AREA IN LUT{2,3,4} WITH CADENCE RC SYNTHESIS.

Library	DES								Kasumi		AES	
type	S1	S2	S3	S4	S5	S6	S7	S8	S7	S9	S	S^{-1}
LuT2 plain	156	145	150	155	155	157	157	158	370	1343	815	834
LuT2 WDDL	312	294	300	310	310	314	314	316	740	2690	1630	1668
LuT2 WDDL+	312	294	300	310	310	314	314	316	740	2690	1630	1668
LuT3 plain	104	93	96	98	98	100	102	102	236	844	513	522
LuT3 WDDL	208	186	192	196	196	200	204	204	472	1688	1026	1044
LuT3 WDDL+	210	194	196	200	196	204	206	204	474	1700	1028	1054
LuT4 plain	79	72	76	76	78	75	77	74	182	640	394	399
LuT4 WDDL	158	144	152	152	156	150	154	148	364	1280	788	798
LuT4 WDDL+	160	146	152	154	156	152	156	152	364	1286	792	802

constant, or that some resources can be shared between the true or false dual networks.

In any case, we conclude that using ASIC synthesizers for generating positive logic is relevant, but that some optimizations are possible. As a perspective, we emphasize that there is a room for custom WDDL+ synthesizers or for enriching legacy synthesis tools with new heuristics when the mapping library is recognized as positive.

Finally, we note that the AES direct substitution box (`SubBytes`) is synthesized in respectively 758 and 792 LuT4s by respectively `bgx_shell` and `rc`. Once again, those figures are worse than hand-made syntheses, such as the ones presented in the dual-rail BDD optimized netlist [2].

C. Comparison between `bgx_shell` and `rc`

As already shown in Tab. X and IX, `bgx_shell` is better than `rc` to synthesize substitution boxes. However, when it comes to simple blocks (everything but the sboxes), `rc` appears to produce more compact netlists than `bgx_shell`. For instance, a two-input `XOR` or a two-input multiplexor is synthesized in 4 LuT4 by `bgx_shell`, but in 2 LuT4 by `rc`. The solution found by `rc` is optimal. The `XOR` is mapped as the positive function $f(a_t, a_f, b_t, b_f) \doteq a_t \cdot b_f + a_f \cdot b_t$, while the multiplexor is inferred as $f(s_t, s_f, a_t, b_t) \doteq a_t \cdot s_f + b_t \cdot s_t$. Therefore, the area reported in Tab. III could be decreased, by using `bgx_shell` for the sboxes an `rc` for the rest.

V. CONCLUSION

The usage of power-constant logic styles to impede the power attacks in FPGA has been shown experimentally. Incidentally, we report the first attack against a non-positive WDDL DES co-processor implemented in an FPGA. The attack takes place by means of very little intrusive acquisition circuitry and by using DPA or CPA strategies. The positive WDDL version proves to be more secure than its non-positive counterpart. However the slight imbalance of this robust logic type should be detectable with a more sensitive acquisition platform. A custom tool based on both FPGAs and ASICs logic synthesizers has been build to get non-positive and positive WDDL netlists. The estimation of overhead is about a factor 2. But by using home-made heuristics, the area bloat can be reduced. We report a constructive method to generate $6 \rightarrow 4$ sboxes in a more compact way than ASIC synthesizers. We therefore expect a new market for secured synthesizers to appear.

REFERENCES

[1] Dakshi Agrawal, Josyula R. Rao, Pankaj Rohatgi, and Kai Schramm. Templates as Master Keys. In *CHES*, volume 3659 of *LNCS*, pages 15–29. Springer, 2005.

[2] Toru Akishita, Masanobu Katagi, Yoshikazu Miyato, Asami Mizuno, and Kyoji Shibutani. A Practical DPA Countermeasure with BDD Architecture. In *CARDIS*, volume 5189 of *Lecture Notes in Computer Science*, pages 206–217. Springer, Sept 2008. London, UK.

[3] Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template Attacks in Principal Subspaces. In *CHES*, volume 4249 of *LNCS*, pages 1–14. Springer, 2006.

[4] Vincent Carlier, Hervé Chabanne, Emmanuelle Dottax, and Hervé Pelletier. Generalizing Square Attack using Side-Channels of an AES Implementation on an FPGA. In Tero Rissa, Steven J. E. Wilton, and Philip Heng Wai Leong, editors, *FPL*, pages 433–437. IEEE, 2005.

[5] S. Chari, J.R. Rao, and P. Rohatgi. Template Attacks. In *CHES*, volume 2523 of *LNCS*. Springer, August 2002. ISBN: 3-540-00409-2.

[6] Sumanta Chaudhuri, Sylvain Guilley, Philippe Hoogvorst, Jean-Luc Danger, Taha Beyrouthy, Alin Razafindraibe, Laurent Fesquet, and Marc Renaudin. Physical Design of FPGA Interconnect to Prevent Information Leakage. In *ARC (Applied Reconfigurable Computing), in LNCS*, volume 4943, pages 87–98, London, UK, mar 2008.

[7] "Circuits Multi-Projets" (alias CMP, $<$cmp@imag.fr$>$) Annual Report 2005. .

[8] Saar Drimer. Personal web page, entitled "FPGA design security bibliography". http://www.cl.cam.ac.uk/~sd410/fpgasec/.

[9] Neil J. A. Sloane (Ed.). The On-Line Encyclopedia of Integer Sequences, published electronically at http://www.research.att.com/~njas/sequences/, Sequence A000372: Dedekind numbers: number of monotone Boolean functions of n variables or number of antichains of subsets of an n-set, 2009.

[10] G. Rouvroy and F.-X. Standaert and J.-J. Quisquater and J.-D. Legat. Efficient Use of FPGAs for Implementations of DES and Its Experimental Linear Cryptanalysis. *IEEE Transactions on Computers*, 52(4), April 2003.

[11] Christophe Giraud and Hugues Thiebeauld. A Survey on Fault Attacks. In *WCC/CARDIS*, pages 159–176, August 2004. Toulouse, France.

[12] Sylvain Guilley, Philippe Hoogvorst, and Renaud Pacalet. A Fast Pipelined Multi-Mode DES Architecture Operating in IP Representation. *Integration, The VLSI Journal (Elsevier)*, 40(4):479–489, July 2007. DOI: 10.1016/j.vlsi.2006.06.004.

[13] Sylvain Guilley, Philippe Hoogvorst, Renaud Pacalet, and Johannes Schmidt. Improving Side-Channel Attacks by Exploiting Substitution Boxes Properties. In *BFCA*, pages 1–25, 2007. May 02–04, Paris, France.

[14] E. De Mulder, S. B. Örs, B. Preneel, and I. Verbauwhede. Differential power and electromagnetic attacks on a FPGA implementation of elliptic curve cryptosystems. *Computers & Electrical Engineering, Elsevier*, 33(5–6):367–382, September–November 2007. doi:10.1016/j.compeleceng.2007.05.009.

[15] NIST/ITL/CSD. Data Encryption Standard. FIPS PUB 46-3, October 1999.

[16] NIST/ITL/CSD. FIPS PUB 197: Advanced Encryption Standard (AES), November 2001.

[17] NIST/ITL/CSD. Cryptographic hash project, November 2007. (http://www.csrc.nist.gov/groups/ST/hash/).

[18] Sıddıka Berna Örs, Elisabeth Oswald, and Bart Preneel. Power-Analysis Attacks on an FPGA: First Experimental Results. In *CHES*, volume 2779 of *LNCS*, pages 35–50. Springer-Verlag, 2003. Cologne, Germany.

[19] T. Popp and S. Mangard. Masked Dual Rail Pre-Charge Logic: DPA Resistance without Routing Constraints. In *CHES*, volume 3659 of *LNCS*, pages 172–186. Springer, September 2005.

[20] T. Popp, T. Zefferer, and S. Mangard. Evaluation of the Masked Logic Style MDPL on a Prototype Chip. In *Proceedings of CHES'07*, volume 4727 of *LNCS*, pages 81–94. Springer, September 2007.

[21] Jean-Jacques Quisquater and David Samyde. ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smard Cards. In I. Attali and T. P. Jensen, editors, *Smart Card Programming and Security (E-smart)*, volume 1240 of *LNCS*, pages 200–210. Springer-Verlag, 2001.

[22] F.-X. Standaert, E. Peeters, F. Macé, and J.-J. Quisquater. Updates on the Security of FPGAs Against Power Analysis Attacks. In *ARC*, volume 3985 of *LNCS*, pages 335–346. Springer-Verlag, March 2006. Delft, The Netherlands.

[23] François-Xavier Standaert, Sıddıka Berna Örs, and Bart Preneel. Power Analysis of an FPGA Implementation of Rijndael: Is Pipelining a DPA Countermeasure? In *CHES*, volume 3156 of *LNCS*, pages 30–44. Springer-Verlag, 2004. Cambridge, MA, USA.

[24] François-Xavier Standaert, Gaël Rouvroy, and Jean-Jacques Quisquater. FPGA Implementations of the DES and Triple-DES Masked Against Power Analysis Attacks. In *FPL*, August 2006. Madrid, Spain.

[25] D. Suzuki and M. Saeki. Security Evaluation of DPA Countermeasures Using Dual-Rail Precharge Logic Style. In *Proceedings of CHES*, volume 4249 of *LNCS*, pages 131–138. Springer, October 2006.

[26] K. Tiri and I. Verbauwhede. A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In *Proceedings of DATE'04*, pages 246–251, February 2004. Paris, France.

[27] Kris Tiri and Ingrid Verbauwhede. Secure Logic Synthesis. In *FPL*, number 3203 in LNCS, pages 1052–1056, aug 2004. Antwerpen, Belgium.

[28] Kris Tiri and Ingrid Verbauwhede. Synthesis of Secure FPGA Implementations. In *International Workshop on Logic and Synthesis (IWLS)*, pages 224–231, june 2004. Temecula, California, USA.

[29] Steve Trimberger. Trusted design in FPGAs. In *Design Automation Conference (DAC)*, pages 5–8, June 2007. San Diego, California, USA.

[30] Pengyuan Yu. Implementation of DPA-Resistant Circuit for FPGA. Master's thesis, Virginia Institute of Technology, april 2007.