# BCDL: A High Speed Balanced DPL for FPGA with Global Precharge and no Early Evaluation.

Maxime NASSAR, Shivam BHASIN, Jean-Luc DANGER, Guillaume DUC, Sylvain GUILLEY
Département COMELEC, Institut TELECOM, TELECOM ParisTech, CNRS LTCI (UMR 5141), PARIS, FRANCE
firstname.lastname@telecom-paristech.fr

***Abstract*—In this paper, we present BCDL (Balanced Cell-based Dual-rail Logic), a new counter-measure against Side Channel Attacks (SCA) on cryptoprocessors implementing symmetrical algorithms on FPGA. BCDL is a DPL (Dual-rail Precharge Logic), which aims at overcoming most of the usual vulnerabilities of such counter-measures, by using specific synchronization schemes, while maintaining a reasonable complexity. We compare our architecture in terms of complexity, performances and easiness to design with other DPLs (WDDL, IWDDL, MDPL, iMDPL, STTL, DRSL, SecLib). It is shown that BCDL can be optimized to achieve higher performances than any other DPLs (more than $1/2$ times the nominal data rate) with an affordable complexity. Finally, we implement a BCDL AES on an FPGA and compare its robustness against DPA by using the number of Measurements To Disclosure (MTD) required to find the key with regards to unprotected AES. It is observed that the SCA on a BCDL implementation failed for 150,000 power consumption traces which represents a gain greater than $20$ w.r.t. the unprotected version. Moreover the fault attack study has pointed out the natural resistance of BCDL against simple faults attacks.**

**Keywords:** Side Channel Attacks, Dual-rail Precharge Logic, Synchronization, Differential Power Analysis, FPGA.

## I. Introduction

Many modern cryptographic algorithms are robust and free from practical cryptanalysis. However, some methods can be deployed to break the security by attacking the physical implementation of an algorithm. These attacks can be mounted by mere observation or perturbation of the targeted system. Observing the activity of the system and its correlation with potential guesses can yield sensible information. Such attacks are better known as Side Channel Attacks (SCAs). When a device is perturbed such that it yields a non-nominal output, this together with expected output can lead to the secret key. Such attacks are called Differential Fault Analyses (DFAs).

The advantages of SCAs are that the system is made to operate in its comfort zone and it is difficult to detect devices which observe the activity of the target. To defeat SCA, the counter-measures have to be introduced at the logic level. Dual-rail Precharge Logic (DPL) is a class of counter-measures which aims at making the device activity constant and independent of the data processed. One DPL computation alternates NULL and VALID tokens, with the remarkable property that exactly one bit toggle occurs in each transition. A pair of gates $(f_{\mathrm{F}}, f_{\mathrm{T}})$ respects the DPL convention if:

- It propagates the NULL $(0,0)$ or $(1,1)$ values, *i.e.*, if all the inputs are NULL, then $(f_{\mathrm{F}}, f_{\mathrm{T}})$ is also NULL.
- It propagates the VALID $(0,1)$ or $(1,0)$ values, *i.e.*, if all the inputs are VALID, then $(f_{\mathrm{F}}, f_{\mathrm{T}})$ is also VALID.

Wave Dynamic Differential Logic (WDDL [1]) is a separable DPL style. In [2], authors introduce a vulnerability in WDDL known as "early evaluation". Owing to this vulnerability, in a WDDL circuit data-dependent leakage can be observed in the side channel and can be further exploited [3], [4].

Masked dual-rail precharge logic (MDPL) [5] is another popular DPL where DPL and masking counter-measures are brought together. The logic interconnect pairs in MDPL are randomly swapped which renders the circuit independent of routing mismatches between each element of a dual-rail pair. The vulnerability of "early evaluation" exists in MDPL also. Practical SCAs against MDPL have been reported in literature [6].

Secure Triple Track Logic (STTL) [7] is a non-masked improvement of WDDL style that is free of early evaluation. Unlike WDDL, STTL is not balanced in structure. It is limited in speed by the validation signal which has to be the slowest signals among the three nets composing a variable. This signal is intentionally delayed with regards to the data signal pairs. This limitation drastically impedes the throughput of STTL.

DRSL [8] combines masking and early evaluation protection. Though DRSL can be used in both ASIC and FPGA, it is optimized to be compact by using one standard ASIC cell (`AOI222`) and all RSL [9] gates. Seclib [10] another DPL counter-measure targeted for ASIC is balanced and free from "early evaluation" & routing imbalance, though its high complexity is a drawback.

Some variants of WDDL (such as Double WDDL (DWDDL) [11], divided backend duplication [12], BDD-style [13] and Isolated WDDL (IWDDL) [14]) have also been devised to ease the balance of WDDL networks. However, it is known that the first two do not solve the early evaluation inherent to this logic. The latter two require manual processing during the synthesis that entails a significant slow-down.

In this paper, we propose a DPL style counter-measure called Balanced Cell-based Dual-rail Logic (BCDL). Aside from providing implementation secured from various vulnerabilities, this logic yields design with better performance than previous DPL styles. The rest of the article is organized as follows. Section II presents the major vulnerabilities in DPL style counter-measures. In Section III, we explain the rationale behind BCDL logic style. The Section IV describes our implementation techniques into FPGAs. Experimental results

in terms of cost, performance and robustness are given in Section IV-C. A comparison with other DPL style is then presented in Section V. Finally, conclusions and perspectives are discussed in Section VI.

## II. DPL Vulnerabilities and Their Solutions

### A. Early Propagation Effect

When a DPL gate switches between phases, input signals acquire their respective values. Switching input signals are likely to acquire respective logic value at different times due to difference in logical path [2], [15]. This stays true even if the gates are balanced. If the transition probability of the gate is unity, the gate will evaluate without waiting for all the signal to acquire the right value, which causes the operation to start at different time in subsequent acquisition. This timing difference can generate a DPA peak on the power traces. This phenomena can occur either during precharge or evaluation and is therefore known as early precharge and early evaluation respectively. Further, authors in [4] demonstrate a successful attack on a DES coprocessor secured with WDDL due to this imbalance. IWDDL [14], another DPL counter-measure separates the positive logic and inverters by introducing registers in between. In this case, the chain of positive gates between two registers will be prone to early evaluation. One way to avoid early evaluation could be by breaking long chain of positive logic by introducing inverters. To counter this flaw, a synchronization is required before the circuit switches between phases.

Isochronous logic can be used to obtain synchronization where the circuit is designed to spend exactly the same time during each of the phases. This can be done by customization of the cells used in the design. Therefore, such techniques are limited to designs targeted for ASIC as in SecLib [10].

The other kind of synchronization is called parallel synchronization. An extra cell of synchronization is added to every cell to overcome early propagation effect. This method is applicable to ASIC and FPGA. Let us consider logics using all $n$-input functions. To ensure minimum leakage from the circuit and avoid glitches, the synchronization should be done according to the following rules:

- **Rule 1:** Evaluation should always be late *i.e.*, evaluation phase starts after all the input signals are valid.
- **Rule 2:** Precharge phase starts :
  1) only after all the inputs becomes NULL and the evaluation outputs are memorized.
  2) before the first input becomes NULL (which do not need any memorization).

DRSL [8] ensures synchronization before evaluation but not before precharge, therefore it has glitches when precharge phase starts as shown in Figure 1. Notice that DRSL+, as we would name DRSL with positive gates, would not glitch. STTL [7] performs synchronization before evaluation and it uses the first method to synchronize before precharge. The latter method is faster and less complex (as it does not require any memorization) and forms the principle of synchronization in BCDL.

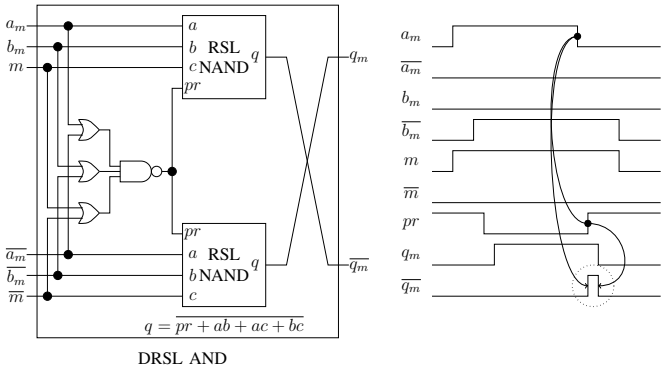


Figure 1. Possible glitch in DRSL due to lack of synchronization before the precharge.

### B. Technological Bias

Another vulnerability in DPL circuits is the technological bias. It comes from two factors. Firstly, the power consumption of a gate and its complement are not always the same when implemented on FPGA. Secondly, non identical routing of the true and the false part can also induce some bias which are reflected in the power curves. The second difference can be removed by "fat wire" [16] and "backend duplication" [17]. Masked DPL logic styles like DRSL [8] and MDPL [5] can fix the two biases by random switching between the complementary parts.

### C. Complexity

Modern cryptographic algorithms rely on complex operations to resist cryptanalysis. These operations have huge implementation cost on hardware in terms of area and performance. Since counter-measures add to this cost, the complexity should be as less as possible, specially for FPGAs.

Symmetrical algorithms possess a non-linear area-consuming part likely to be implemented in the FPGA's embedded RAM. A basic solution is to implement the $n$-input S-Box (RAM of size $2^n$) into two T and F $2n$-input S-Boxes (of size $2^{2n}$ each) because every S-Box needs both T and F inputs. This RAM size can be reduced if there is a way to recognize the precharge state. For instance in WDDL, the AES S-Box can be implemented in two $2^9$-byte ROMs with a ninth bit being used to force the precharge state. However a special care has to be taken to avoid glitches if the precharge state occurs without synchronization with the T and F data.

Although STTL is resistant against early propagation effect, its complexity is an issue. The method used to synchronize before precharge in STTL needs memorization. It also routes three wires per signal which increases the cost. Its throughput is also lower than other DPL style logic. Masked DPL (MDPL and DRSL), which solve the problem of technological bias are expensive in terms of area because its uses twice the area of unmasked DPL.

IWDDL adds registers before the existing inverters to enable flow of precharge. This seriously affects the throughput of the circuit as a complex cryptographic algorithms uses a lot of

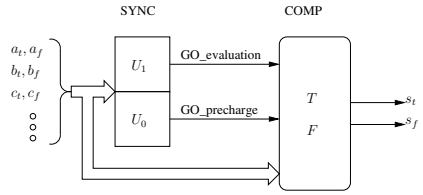


Figure 2. Synchronization and "bundled" data in BCDL.

inverters. For instance, a low cost AES S-Box [18] implemented with finite field arithmetic operations in $GF(2^8)$ with co-efficients in $GF(2^4)$ contains 21 inverters when synthesised with positive logic. This will also add a lot to the cost of the circuit. Early evaluation protection by adding extra inverters will further add to the cost.

## III. BCDL Principle

The main goal of BCDL [19] is to avoid most of the vulnerabilities in current DPL. Therefore it is based on two principles:

1) A specific **synchronization** scheme (to meet the rule explained in section II-A) is added to all logic gates, before the actual precharge or evaluation.
2) The synchronization is performed on **Bundle Data** (which is well adapted to FPGA LUTs).

### A. Synchronization

*1) Basic Principle:* Synchronization in asynchronous logic is usually performed between 2 signals with a rendezvous cell "RV", also called "C-element". RV is a memory that only changes its state when there is unanimity (to 0 or 1) on its inputs. With BCDL, the synchronization is done on Bundle Data, without any memory element, using specific cells: $U_0$ and $U_1$ (Unanimity to 0 and 1).

- $U_1$ is the signal authorizing the evaluation. It raises up to 1 when all signals have left the precharge state. More precisely $U_1(x, y, \dots)$ is defined by equation (1):

$$U_1(x, y, \dots) \doteq \begin{cases} 1 & \text{if } x \neq (0,0) \text{ and } y \neq (0,0), \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

- $U_0 = 1$ when all the inputs are in the precharge state, as shown in equation (2):

$$U_0(x, y, \dots) \doteq \begin{cases} 1 & \text{if } x = y = \dots = (0,0), \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Actual calculation only starts if there is unanimity ($U_1$ or $U_0$ valid) and is frozen otherwise.

Figure 2 shows a schematic diagram for synchronization of bundle data.

*2) Optimized Synchronization Principle with Global Precharge:* Calculation of the precharge is quite a simple operation (compared with evaluation), as it only requires that all cell outputs be forced to 0, while the latter is an actual computation of signals carrying information. Based on this property, we can optimize our model by using a simplified rendezvous scheme, coupled with a **global precharge signal**, $PRE$. It is used to induce the precharge state globally, in a very short amount of time. Thus, it allows the designer to reduce the complexity (and increase the performances) of the BCDL rendezvous cell, by replacing the "unanimity to 0" (of Figure 2) by a logical "AND" between the $PRE$ signal and the output of the "unanimity to 1" (see Figure 3).

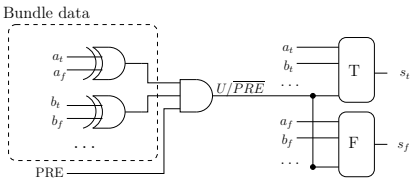


Figure 3. BCDL $n$-input cell.

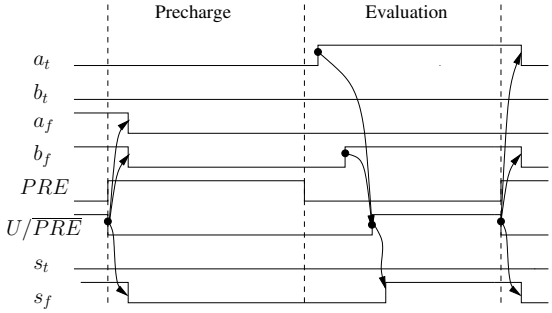


Figure 4. Temporal relationships of a 2-input BCDL OR gate signals.

The actual computation is then synchronized by the $U/\overline{PRE}$ signal as follows:

- When $U/\overline{PRE}$ falls to 0 (just after the signal $PRE$), the precharge is forced, independently from the inputs. Using a global signal in FPGAs ensures us that the precharge signal will always be faster than any input. As a matter of fact $PRE$ takes advantage of the FPGA global lines which are specific, fast and sized to broadcast heavy loaded signals. Moreover frequency of $PRE$ is half that of the clock signal which can be generated from a FPGA PLL without any skew with respect to the clock.
- When $U/\overline{PRE}$ raises to 1, indicating that, on one hand, the signal $PRE$ is valid and, on the other hand, that the "rendezvous" of inputs is over, the evaluation phase begins.

Precise temporal relationships between signals of a 2-input OR gate (where $(a_t, a_f)$, $(b_t, b_f)$ are the inputs and $(s_t, s_f)$ the output) are shown in Figure 4.

*3) Lut-Level Optimization:* Based on the above properties, our logic is able to overcome early evaluation on a global scale (between each cell of the circuit). However, it must also be synchronized at LUT-level, in order to avoid local early evaluation and technological unbalance.

An analysis of FPGA cells shows that their LUT structure is a tree of multiplexers as shown in Figure 5.

Local synchronization is achieved by applying the two following constraints:

- The $U/\overline{PRE}$ signal is assigned to the first column of this tree.

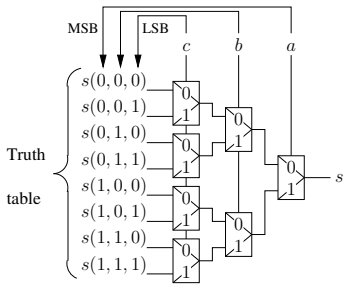


Figure 5. Structure of T and F LUT.

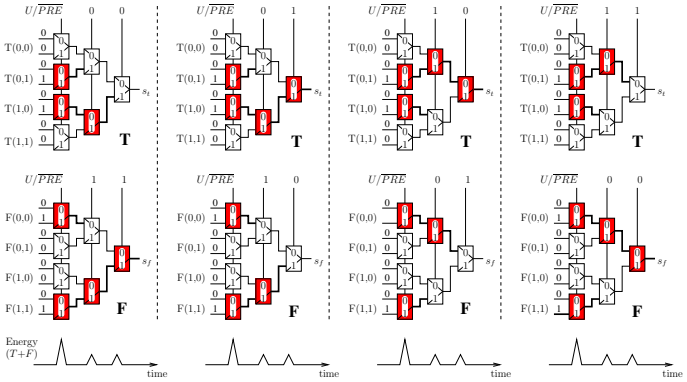


Figure 6. Local switching balance in BCDL: LUT3 example.

- The inputs $e_t$ and $e_f$ are plugged on the same pin respectively on T and F cells.

Due to these constraints, we can obtain significant properties regarding local robustness which are:

- **No glitches:** As the $U/\overline{PRE}$ signal is the first to switch before the precharge state, the internal nets are all forced to '0' without any glitch, regardless of the inputs. Likewise, it is the last one to switch prior to the evaluation after the other inputs are already positioned.
- **Reduction of the technological bias:** The total number of commutations for T and F equipotentials does not change according to the inputs. It is a constant $= (2^n - 1)$ for a $n$-input LUT. It is therefore difficult to discriminate the activity of T from that of F as the consumption profile is identical regarding the couple T , F . This is illustrated in Figure 6, which describes all the combinations of a 2-input XOR, when $U/\overline{PRE}$ switches. Bold nets correspond to multiplexers outputs that are switching. There is, thereby, an overall balance in terms of switching time as well as energy consumption (number of simultaneous switching).
- **No local early evaluation or precharge:** Indeed, $U/\overline{PRE}$ is always delaying the evaluation (switching to '1' last) and forcing the precharge (falling to '0' before any other signal). In other words, the evaluation is always **delayed** and the precharge always **anticipated**, regardless of the data.

### B. Reduced Area

Area-consumption is a limitation for most DPL counter-measures on FPGA. Even if the main goal is to achieve the best robustness, it could prove useless if it is actually too complex to implement on a real device. Thereby, one of the main objectives of BCDL is to keep a reasonable complexity. Thanks to our synchronization schemes we obtain three significant properties:

- **Reduced S-Box area:** As stated before (see Section II-C), in various DPL style logic, one basic 8-input S-Box ($2^8$ byte) could be merely dualized into two T and F 16-input S-Boxes ($512 \times 2^8$ byte). This huge size can be reduced by building a local precharge signal but it might induce glitches due to the lack of synchronization. BCDL takes advantage of its global precharge signal to reduce the RAM size to only **4 times** the basic one without any glitch risk. Indeed there will be T and F S-Boxes, which will only have one more input than the basic implementation, that is the $U/\overline{PRE}$ signal.
- **Reduced complexity:** Due to the absence of glitches within LUTs (see section III-A3), BCDL is not limited to positive functions (unlike WDDL), and can use all $2^{2^n}$ existing functions (for a $n$-input LUT), which provides many optimization opportunities.
- **Integrated rendezvous:** We can exploit the recent FPGA technologies to make this optimization, in fact, for a 2-input function, if we use a FPGA with 5-input LUTs ($LUT5$) or more, we can directly integrate the synchronization scheme in the T and F cells. Indeed, if the $true$ and $false$ signals as well as $U/\overline{PRE}$ are inputs of the same LUT, the rendezvous can be computed without additional logic cells. It is then possible to implement a 2-input BCDL function with only 2 $LUT5$.

### C. High Performances

As of now, all DPL-based counter-measures have about the same performances, and that is a speed at least two times slower than the unprotected architecture. This is mainly the result of the typical 2-phase functioning ($precharge$, $evaluation$) which is common to all DPLs. Most of the times, the precharge must have roughly the same duration as the evaluation, in WDDL for example, it must last long enough for the '0' to go through all the logic. On the other hand, due to the global precharge signal, BCDL can be optimized to be faster than any other DPL. As a matter of fact, the global signal being extremely fast and homogeneously distributed throughout the device, the duration of the precharge state can be quite short. More time is then given to the evaluation, which dictates the speed of the design. This can be achieved by using a non-regular clock, as shown in Figure 7. Using this scheme, we can rise the speed up to $\sim 1.3 - 1.5$ times the basic one.

### D. In-Built Robustness against Fault Attacks

Another interesting characteristic of BCDL is that it automatically detects simple faults ($1 \rightarrow 0$ and $0 \rightarrow 1$) without adding any hardware. As a matter of fact, the truth table of our cells is such that if a single net is stuck to 0 or 1 (i.e. $true$ and $false$ nets of the same variable have the same value during the evaluation phase), both true and false cells will switch to

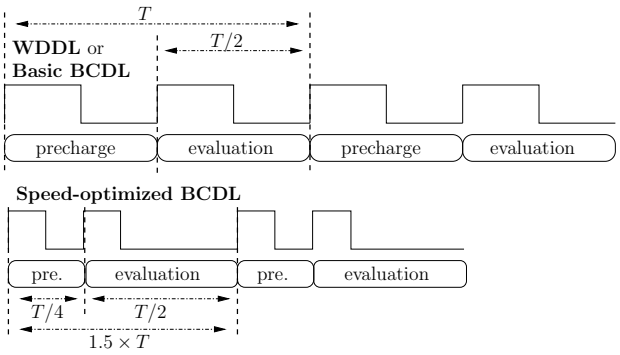


Figure 7. Basic BCDL *versus* speed-optimized BCDL timings.

an "error state" (both will output a 0) which is propagated till the output.

BCDL, as any DPL style w/o early evaluation, is also fully immune against setup violation attacks [20] and mostly immune against multiple faults [21].

## IV. Security Experimental Results

In order to evaluate the robustness, area cost and performance of BCDL, we implemented a an unprotected AES-128, a BCDL AES-128 and a WDDL AES-128 on an Altera StratixII FPGA. As discussed in Section II-C it would hardly be possible to put WDDL S-Boxes in RAM so we implemented them using logic elements.

### A. Implementation

To implement BCDL, we used a "Bottom-up" approach, which consists in building primitives for the given algorithm, and use them to reconstruct the entire design. In our case, we implemented BCDL $XOR$, $MUX$ and ROM to create a VHDL description of our BCDL AES-128. We used solely 2-input XOR gates (exploiting the last optimization described in Section III-B), in order to directly integrate the synchronization scheme in the gates, and so provide a low-cost implementation in terms of area. For the same purpose, we implemented the S-Boxes in the RAM of the FPGA (using BCDL first property of Section III-B). This netlist was then given to Quartus (Altera design software) for synthesizing and generating the bitstream.

### B. Complexity

We noted down the area consumed, in terms of ALM (Adaptive Logic Modules are StratixII basic logic elements), RAM, and registers, for both WDDL and BCDL, and compared them with those of the unprotected AES. We also estimated the maximal clock frequency and the throughput for each design. Results are displayed in Table I.

As expected, we observe that BCDL is faster, and a lot less area-consuming than WDDL.

### C. Robustness Against DPA

We performed DPA attacks on the unprotected AES (as reference) and both WDDL and BCDL AES, implemented on the same StratixII chip. None of these DPL has been

Table I
Cost and performance of the AES datapath in single-rail (unprotected reference), WDDL and BCDL styles.

| | ALM | Reg | RAM | Max. freq. | Max. throughput |
|---|---|---|---|---|---|
| **AES** | 1078 | 256 | 40 Kb | 71.88 MHz | 287.52 Mbps |
| **WDDL** | 4885 | 1024 | — | 37.07 MHz | 74.14 Mbps |
| **BCDL** | 1841 | 1024 | 160 Kb | 50.64 MHz | 151.92 Mbps |

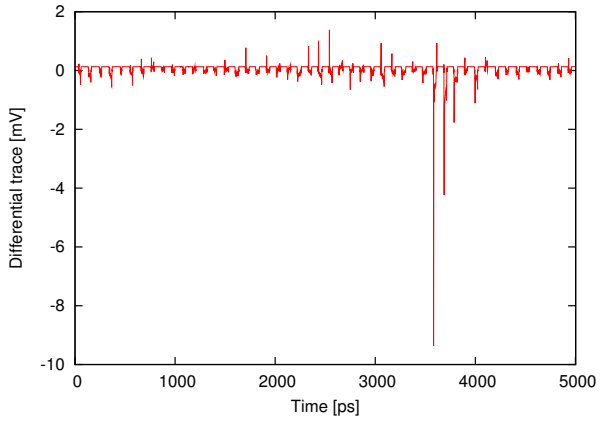


Figure 8. Differential curve for DPA on S-Box1 of the basic AES.

constrained at the Place and Route (P/R) stage to balance the T and F networks. We took power consumption measurements ($traces$), using a *differential probe* plugged to the positive rail of the FPGA core power supply through a 1 Ω shunt resistor, coupled with a *54855 Infiniium oscilloscope* from Agilent Technologies.In order to reduce acquisition noise, each trace was averaged 256 times. We were able to find most of AES sub-keys with about $8,000$ traces (one of the resulting differential curve is displayed in Figure 8), but were unable to find even one sub-key of either WDDL or BCDL AES with $150,000$ traces (which makes a robustness gain $\sim 20$).

Precise results are exposed in Table II.

## V. Further Comparisons with Other DPL Styles

Table III draws up a comparison of the main DPL styles, in terms of principle, design constraints and performance, highlighting most of the known advantages (masking, synchronization) and drawbacks (primitives and back-end constraints, and technological bias) of such counter-measures.

**Masking** allows to greatly reduce the technological bias, but also results in a significant increase of area. As a matter of fact, it requires at least a transformation of 2-input operations into 3-input majority function (MDPL) or into a 4-input RSL gate (DRSL).

**Synchronization** on both precharge and evaluation is mandatory to avoid glitches and early propagation effects (as

Table II
DPA results on AES.

| **S-Box index** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **NB traces** | 2821 | 1215 | 2627 | 1372 | 6333 | 3046 | 5194 | 6841 |
| **S-Box index** | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| **NB traces** | 11683 | 9510 | 11743 | 11857 | 8368 | 8822 | 11770 | 3681 |

Table III
DPL PERFORMANCE AND SECURITY FEATURES OVERVIEW.

| Logic | Mask | Synchro | | Constraints | | Tech Bias | Speed |
|---|---|---|---|---|---|---|---|
| | | Pre | Eval | Primitives | Back-end | | |
| **WDDL** | no | ✗ | ✗ | positive gates only | balanced place&route | high | $< 1/2$ |
| **MDPL** | yes | ✗ | ✗ | $MAJ$ † | no | no | $< 1/2$ |
| **STTL** | no | ✓ | ✓ | no | delay on sync signal | very low | $< 1/4$ |
| **DRSL** | yes | ✓ | ✗ | no | no | no | $< 1/2$ |
| **Seclib** | no | ✓ | ✓ | specific lib | back-end duplication | very low | $< 1/2$ |
| **IWDDL** | no | ✓ | ✓ | no | netlist post-processing | low | $< \frac{1}{2 \cdot n_i}$ ‡ |
| **BCDL** | no | ✓ | ✓ | no | balanced place&route | low | $> 1/2$ |

† $MAJ$ stands for the majority gate: $MAJ(a, b, c) \doteq a \cdot b + b \cdot c + c \cdot a$.
‡ $n_i$ is the maximum number of inverters amongst all combinatorial paths.

described in section II-A).

**Primitive constraints** induce a higher complexity, by reducing the panel of usable functions (like in WDDL where only positive gates are allowed), or by binding the designer to use specific functions that can be more area-consuming or slower than basic ones (Seclib, MDPL, DRSL).

**Back-end constraints** generate extra design work as the P/R stage has to properly balanced the T and F networks. It can also cause a loss of performance, like in STTL where the synchronisation signal must be manually made slower than the others, by adding delay elements between each gates, in order to ensure that it always switches last.

**Technological bias** is a significant source of information leakage, and must therefore be as low as possible to ensure a perfectly secure counter-measure (see section II-B).

## VI. Conclusions and Perspectives

In this article, we presented BCDL as a new DPL counter-measure against side-channel attacks. BCDL counters early propagation effects deploying an optimized synchronization scheme with a global precharge signal. Thanks to this global precharge signal, BCDL can be optimized to provide better performances than other DPL-based countermeasures (more than $50\%$ throughput gain). Considering FPGA target, the number of cells increase is roughly a factor 2 w.r.t. unprotected implementation by using a bottom-up synthesis approach. The S-Box implementation can take advantage of the global precharge to easily fit into FPGA embedded RAM.

DPA attacks have been carried out on a full-fledged AES in an StratixII FPGA without any P/R constraint. With the BCDL implementation we were unable to find the right key with $150,000$ traces providing a robustness gain of at least $\sim 20$ w.r.t. the unprotected implementation. As BCDL should theoretically require a special care at the back-end stage of the design to balance the T and F networks, a perspective could be to implement a masked version, called MBCDL.

## References


[1] K. Tiri and I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation," in *DATE'04*. IEEE Computer Society, February 2004, pp. 246–251, Paris, France.

[2] D. Suzuki and M. Saeki, "Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style," in *CHES*, ser. LNCS, vol. 4249. Springer, 2006, pp. 255–269.

[3] S. Guilley, L. Sauvage, J.-L. Danger, T. Graba, and Y. Mathieu, "Evaluation of Power-Constant Dual-Rail Logic as a Protection of Cryptographic Applications in FPGAs," in *SSIRI*. Yokohama, Japan: IEEE Computer Society, jul 2008, pp. 16–23, DOI: 10.1109/SSIRI.2008.31, http://hal.archives-ouvertes.fr/hal-00259153/en/.

[4] L. Sauvage, S. Guilley, J.-L. Danger, Y. Mathieu, and M. Nassar, "Successful Attack on an FPGA-based WDDL DES Cryptoprocessor Without Place and Route Constraints," in *DATE*. Nice, France: IEEE Computer Society, apr 2009, pp. 640–645.

[5] T. Popp and S. Mangard, "Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints," in *CHES*, ser. LNCS, vol. 3659. Springer, Sept 2005, pp. 172–186., Edinburgh, Scotland, UK.

[6] E. D. Mulder, B. Gierlichs, B. Preneel, and I. Verbauwhede, "Practical DPA Attacks on MDPL," Cryptology ePrint Archive, Report 2009/231, 2009, http://eprint.iacr.org/.

[7] R. Soares, N. Calazans, V. Lomné, P. Maurine, L. Torres, and M. Robert, "Evaluating the robustness of secure triple track logic through prototyping," in *SBCCI'08*. New York, NY, USA: ACM, 2008, pp. 193–198.

[8] Z. Chen and Y. Zhou, "Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage," in *CHES*, ser. LNCS, vol. 4249. Springer, 2006, pp. 242–254, yokohama, Japan.

[9] D. Suzuki, M. Saeki, and T. Ichikawa, "Random Switching Logic: A New Countermeasure against DPA and Second-Order DPA at the Logic Level," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E90-A, no. 1, pp. 160–168, 2007.

[10] S. Guilley, F. Flament, R. Pacalet, P. Hoogvorst, and Y. Mathieu, "Security Evaluation of a Balanced Quasi-Delay Insensitive Library," in *DCIS*. Grenoble, France: IEEE, nov 2008.

[11] 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.

[12] K. Baddam and M. Zwolinski, "Divided Backend Duplication Methodology for Balanced Dual Rail Routing," in *CHES*, ser. LNCS, vol. 5154. Washington, DC, USA: Springer, aug 2008, pp. 396–410.

[13] T. Akishita, M. Katagi, Y. Miyato, A. Mizuno, and K. Shibutani, "A Practical DPA Countermeasure with BDD Architecture," in *CARDIS*, ser. Lecture Notes in Computer Science, vol. 5189. Springer, Sept 2008, pp. 206–217, London, UK.

[14] R. P. McEvoy, C. C. Murphy, W. P. Marnane, and M. Tunstall, "Isolated WDDL: A Hiding Countermeasure for Differential Power Analysis on FPGAs," *ACM Trans. Reconfigurable Technol. Syst. (TRETS)*, vol. 2, no. 1, pp. 1–23, 2009.

[15] K. J. Kulikowski, M. G. Karpovsky, and A. Taubin, "Power Attacks on Secure Hardware Based on Early Propagation of Data," in *IOLTS*. IEEE Computer Society, 2006, pp. 131–138, Como, Italy.

[16] K. Tiri and I. Verbauwhede, "Place and Route for Secure Standard Cell Design," in *Proceedings of WCC / CARDIS*, Kluwer, Ed., Aug 2004, pp. 143–158, Toulouse, France.

[17] S. Guilley, P. Hoogvorst, Y. Mathieu, and R. Pacalet, "The "Backend Duplication" Method," in *CHES*, ser. LNCS, vol. 3659. Springer, 2005, pp. 383–397, August 29th – September 1st, Edinburgh, Scotland, UK.

[18] J. Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC Implementation of the AES SBoxes," in *CT-RSA*, ser. Lecture Notes in Computer Science, B. Preneel, Ed., vol. 2271. Springer, 2002, pp. 67–78.

[19] J.-L. Danger and S. Guilley, "Circuit de cryptographie programmable – Logique BCDL (Balanced Cell-based Differential Logic)," 25 Mars 2008, Brevet Français FR08/51904, assigné à l'Institut TELECOM; WO/2009/118264.

[20] N. Selmane, S. Bhasin, S. Guilley, T. Graba, and J.-L. Danger, "WDDL is Protected Against Setup Time Violation Attacks," in *FDTC*. IEEE Computer Society, September 6th 2009, pp. 73–83.

[21] S. Bhasin, J.-L. Danger, F. Flament, T. Graba, S. Guilley, Y. Mathieu, M. Nassar, L. Sauvage, and N. Selmane, "Combined SCA and DFA Countermeasures Integrable in a FPGA Design Flow," in *ReConFig*. IEEE Computer Society, December 9–11 2009, p. 6 pages.