applied sciences

MDPI

Article

Gate-Level Hardware Countermeasure Comparison against Power Analysis Attacks

Erica Tena-Sánchez [1,2,*], Francisco Eugenio Potestad-Ordóñez [1,2], Carlos J. Jiménez-Fernández [1,2], Antonio J. Acosta [1,3] and Ricardo Chaves [4]

[1] Instituto de Microelectrónica de Sevilla, IMSE-CNM-CSIC/US, 41092 Seville, Spain; potestad@imse-cnm.csic.es (F.E.P.-O.); cjesus@imse-cnm.csic.es (C.J.J.-F.); acojim@imse-cnm.csic.es (A.J.A.)

[2] Departamento de Tecnología Electrónica, Escuela Politécnica Superior, Universidad de Sevilla, 41011 Sevilla, Spain

[3] Department de Electrónica y Electromagnetismo, Facultad de Física, Universidad de Sevilla, 41012 Sevilla, Spain

[4] INESC-ID, IST, Universidade de Lisboa, 1049-001 Lisboa, Portugal; ricardo.chaves@tecnico.ulisboa.pt

[*] Correspondence: erica@imse-cnm.csic.es

Abstract: The fast settlement of privacy and secure operations in the Internet of Things (IoT) is appealing in the selection of mechanisms to achieve a higher level of security at minimum cost and with reasonable performances. All these aspects have been widely considered by the scientific community, but more effort is needed to allow the crypto-designer the selection of the best style for a specific application. In recent years, dozens of proposals have been presented to design circuits resistant to power analysis attacks. In this paper, a deep review of the state of the art of gate-level countermeasures against power analysis attacks has been carried out, performing a comparison between hiding approaches (the power consumption is intended to be the same for all the data processed) and the ones considering a masking procedure (the data are masked and behave as random). The most relevant proposals in the literature, 35 for hiding and 6 for masking, have been analyzed, not only by using data provided by proposers, but also those included in other references for comparison. Advantages and drawbacks of the proposals are analyzed, showing quantified data for cost, performance (delay and power), and security when available. One of the main conclusions is that the RSL proposal is the best in masking, while TSPL, HDRL, SDMLp, 3sDDL, TDPL, and SABL are those with the best security performance figures. Nevertheless, a wise combination of hiding and masking as masked_SABL presents promising results.

Keywords: hardware countermeasures; gate level; VLSI design of cryptographic circuits; side-channel attacks (SCAs); information security; logic design; Internet of Things (IoT)

Citation: Tena-Sánchez, E.; Potestad-Ordóñez, F.E.; Jiménez-Fernández C.J.; Acosta A.J.; Chaves R. Gate-Level Hardware Countermeasure Comparison against Power Analysis Attacks. *Appl. Sci.* **2022**, *12*, 2390. https://doi.org/10.3390/app12052390

Academic Editors: Guy Gogniat, Vianney Lapotre and Maria Mushtaq

Received: 8 January 2022
Accepted: 22 February 2022
Published: 25 February 2022

1. Introduction

The high growth that the Internet of Things (IoT) is experiencing has brought with it an increase in the exchange of sensitive information from interconnected users. This has meant that the devices need to incorporate cryptosystems capable of protecting the information to maintain the integrity and confidentiality of the data [1,2]. Traditionally, the mathematical algorithm and the length of the key defined the security of cryptosystems. However, the physical implementation of cryptographic algorithms leads to information leakages that can be exploited by third parties to reveal critical data. These cryptocircuits must meet all the necessary requirements to minimize vulnerabilities to malicious attacks by third parties. Here, the development of new attack techniques makes security standards insufficient to protect information [3–6].

Among the different types of attacks, the so-called side-channel attacks (SCAs) belong to the group of passive noninvasive attacks and are those where the cryptographic device is not manipulated, e.g., there is no trace that a malicious agent has had access to the device and there is no damage to the circuit [3–6]. Among SCAs, those based on analysis of the power consumption (power analysis, PA) produced by the circuit have attracted significant attention from the research community [5].

Since the emergence of power analysis attacks in the late 1990s, numerous countermeasures have been proposed by the scientific community to search for alternatives to minimize the weak points of cryptocircuits [7–11]. There are several countermeasure strategies at the hardware level, depending on the abstraction level and the mechanism, to uncorrelate the power consumption from the key and data being processed. These countermeasures range from the layout up to algorithm level and go from attack detection to adding redundant blocks to obfuscate possible information leakage. In this sense, the existing hardware countermeasures can be classified as illustrated in Figure 1. To start, we can make a first classification depending on the abstraction level, with countermeasures focused on the algorithm/circuit or gate level. In this paper, we focus on hardware countermeasures applied at gate level. Their main advantage is that, once the secure cell library has been designed with the selected secure logic style, and the automatic design flow has been adapted for use with this new library, the same design flow can be applied regardless of the implemented algorithm, considering unprotected algorithm. It is important to note that, if countermeasures at the gate level are to be combined with others at a higher level, or complementary ones at the same level of abstraction, additional studies must be carried out to see if their overall security improvements do not interfere with each other. On the other hand, depending on the technique used to break the data correlation with the power consumption, we can classify the countermeasure into two main groups: the hiding or masking technique.

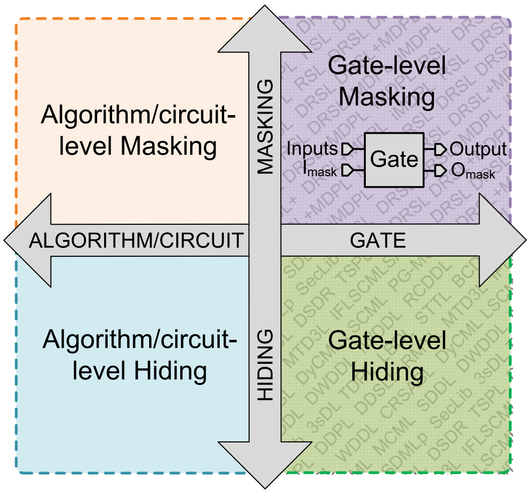


Figure 1. Hardware countermeasures.

The main contribution of this paper is to first make a deep analysis of the state of the art of most relevant hardware PA countermeasures applied at gate level and secondly to analyze their main drawbacks and advantages. This analysis consists of the evaluation of the resource overhead and security improvements of each proposal, and finally compares each of the countermeasures to determine those that best fits the design constraints. In this sense, this work provides the designer with a tool to search and select the most appropriate countermeasure, depending on the target application and the technology considered. For example, we may find ourselves in IoT environments where we cannot exceed a certain area or power budget, or just be in a scenario where the highest levels of security are targeted, regardless of the associated costs of those countermeasures.

The rest of the paper is organized as follows. In Section 2, the basis of hardware countermeasures against power analysis attacks at gate level is presented. In Section 3, the gate-level hiding countermeasures are analyzed, whereas Section 4 focuses on the gate-level masking countermeasures. In Section 5, the comparison of all countermeasures, regarding their performance and security levels, is discussed. Finally, Section 6 concludes this paper.

2. Gate-Level Countermeasures against Power Analysis Attacks

PA attacks exploit the correlation between power consumption and the data that are processed by the cryptographic device during encryption, following several strategies. Single power analysis (SPA) attacks exploit the information from a single trace captured from the power supply.

On the other hand, differential power analysis (DPA) attacks involve the acquisition of a series of power supply traces while the device under attack is operating: varying plain-text inputs for a selected (hidden) key and analyzing the power traces. The simplest analysis is to choose an intermediate value of the encryption process and divide the set of acquired traces according to the expected value for these bits. Following this, we statistically analyze the correlation of these traces point by point, comparing them with a power model, as depicted in Figure 2. A significant difference should then be visible in the points corresponding to where a power consumption difference exists due the key-dependent output. This is typically referred to as first-order analysis, as each point in the output trace is dependent on the same point in time for all traces. If two (or more) points in each trace are combined, we refer to this as a second-order (or higher-order) analysis. In the cyber community, DPA attacks have become the most widespread attack, because of their effectiveness, associated with their reduced cost. For a detailed reading of DPA attacks, power models, and applications, please refer to [3].

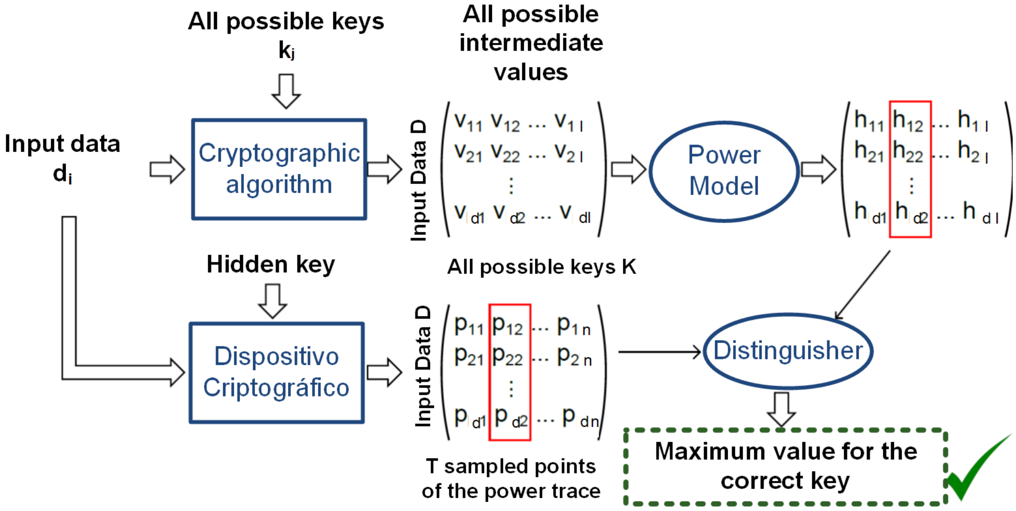
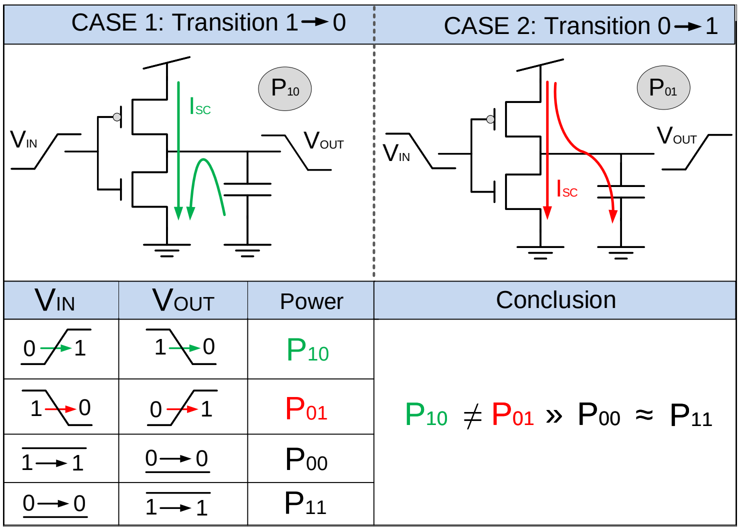


Figure 2. Simplified DPA attack flow.

Hardware countermeasures are oriented towards breaking the relation between data being processed and consumed power. At gate level, this relation is easily visible when working with static complementary metal-oxide semiconductor (CMOS) gates, where the 0→0, 0→1, 1→0 and 1→1 transitions have different power consumption values, as illustrated in Figure 3.

To break this relation at the gate level, two different mechanisms are widely used: hiding and masking techniques. The hiding attempts to have the same power consumption at the gate, circuit, or algorithm level, independently of the data being processed. In masking, the critical data are masked with a random data sequence during encryption such that operations on the masked data are indistinguishable from random data.

Gate-level masking consists of computing both the inputs and the mask inside the gate itself. In these implementations, each masked signal a_m is propagated along with its mask m_a, the unmasked signal being $a = a_m \oplus m_a$. The simplest way to perform masking is through boolean masking, where an input word is masked by being XOR-ed by a random value. Arithmetic masking involves more complex arithmetic operations within specific algorithms. Boolean masking is preferably used at the gate level, while at the algorithm or circuit level, the use of dedicated arithmetic masking techniques that best suit the algorithm are recommended. One of the first techniques for masking complex functions is the use of masked look-up tables or the use of multiplexer trees, as proposed in [12], which can be applied both at the algorithm/circuit level and at the individual gate level.



V_{IN}	V_{OUT}	Power	Conclusion
0→1	1→0	P_{10}	$P_{10} \neq P_{01} \gg P_{00} \approx P_{11}$
1→0	0→1	P_{01}	
1→1	0→0	P_{00}	
0→0	1→1	P_{11}	

Figure 3. Power consumption of an inverter depending on the output transitions.

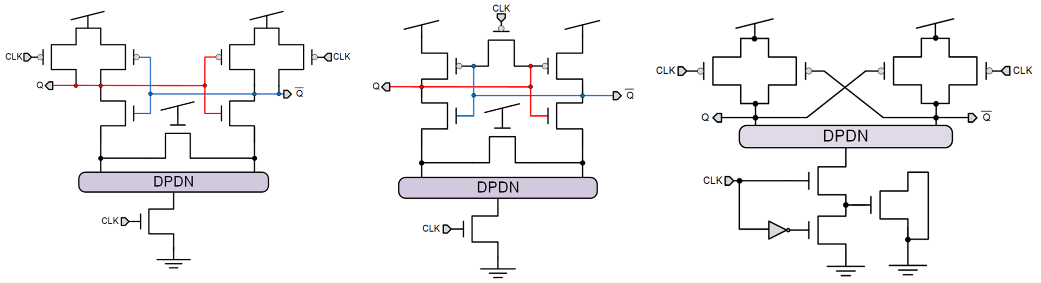
Hiding tries to achieve exactly the same power consumption in operations, regardless of the data being processed. Since the first DPA attacks were presented, there have been numerous logic-style proposals that seek to be resistant to these attacks by having data-independent power consumption. In a first approach, this identical consumption can be achieved using dual-rail signals and differential gates, where the true and complemented outputs are simultaneously generated: in every clock cycle, one of the differential branches performs the gate function, and the other one its complement at the same time.

Since hiding means exact power consumption independently of the data processed, it implies full symmetry. However, most of these techniques suffer from the difficulty of tailoring the place and route operation so that the capacitive load of two wires is equal. This is particularly difficult in nanometric technologies, where the transistor sizes and wiring widths continuously shrink. Placing and routing a circuit manually, i.e., creating a full-custom (FC) design, significantly increases the design costs. An additional drawback is the so-called early evaluation, also called data-dependent time-of-evaluation, referring to the cases where a gate evaluates its output at different time instances depending on the value of its input. It becomes more problematic when several of such gates are cascaded to realize a combinational circuit, causing the power consumption pattern of the circuit to have a clear dependency on its input value.

The following two sections detail these two types of gate-level countermeasures, followed by systematic evaluation of them.

3. Gate-Level Hiding Countermeasures

In this section, a detailed description of the structure and functionality for each hiding logic style in the considered literature is provided. In this sense, the reader may skip this detailed description and go to the end of the section for a summary of their main features and comparison. One of the first hiding proposals at gate level was presented by Tiri et al. [7], called sense amplifier-based logic (SABL), based on the StrongArm110 flip-flop (SAFF) [13] structure. SABL is a dual precharge logic (DPL) style and achieves switching the output independently of the input value, always having an output transition due to the computation of the output signal and its complement, charging the same load capacitance in each transition. The SABL structure, depicted in Figure 4a, is composed of a differential pull-down network (DPDN) and a differential pull-up network (DPUN), implementing the logic function and the control between the precharge and evaluation phases. The CLK signal controls the precharge and evaluation phases. The main drawback is its full-custom logic style and the need for differential routing to maintain the same load capacitance in the complemented signal. In [14], authors present an improved SABL structure, called charge recycling SABL (CRSABL), which enables charge recycling and intermediate precharge voltages preserving the same security levels as SABL but saving 20% in power consumption and 63% in peak supply current. The CRSABL structure is depicted in Figure 4b.

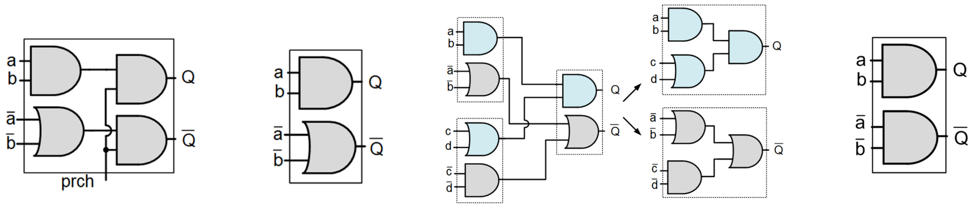


(a) Generic SABL n-gate (b) Generic CRSABL n-gate (c) Generic DyCML n-gate

Figure 4. Gate-level hiding logic styles: (**a**) SABL, (**b**) CRSABL, and (**c**) DyCML.

Dynamic current mode logic (DyCML) [15] was first presented as a low-power high-performance logic style, and was proposed by Macé et al. for DPA-resistant circuit implementations [9]. DyCML is based on current mode logic (CML) gates [16], which have the advantage of high-frequency operation and low commutation noise (as in CML gates), but solving the issue of the static power consumption. A generic DyCML n-gate structure is depicted in Figure 4c, composed of a DPDN structure that implements the gate function and the DPUN that controls the dynamic functionality of the gate. Compared to SABL [9], DyCML achieves better performance characteristics, reducing both the power consumption and delay, while slightly decreasing the security. The performance and security metrics for this logic style vary from paper to paper. For this reason, Table 1 depicts a wide range for area, frequency, and power values, further discussed in Section 5.

The biggest drawback of full-custom logic styles is the design complexity and the impossibility to use a conventional digital design flow, for example, not suitable being for field programmable gate array (FPGA) implementations. To solve this issue, Tiri and Verbauwhede presented, in [8], the simple dynamic differential logic (SDDL), the wave dynamic differential logic (WDDL), and the divided wave dynamic differential logic (DWDDL), based on standard cell libraries. The SDDL implements each function with standard gates, using differential inputs and outputs (a, $\overline{a}$, b, $\overline{b}$ and Q, $\overline{Q}$) with a precharge signal to implement the dynamic functionality (see Figure 5a). Unfortunately, the SDDL is not suitable for secure implementations because it does not ensure one single event per cycle, influencing both the timing and the value of the inputs in the number of switching events, and being impossible to achieve an input signal that is independent of the power consumption. In WDDL, the precharge signal is propagated as a wave along the combinational logic, so the area is halved with respect to SDDL, obtaining the structure depicted in Figure 5b. As in previous full-custom logic styles, WDDL needs both differential routing to balance the wiring capacitances (techniques such as "fat-wire" [17] or "backend duplication" method [18]), and a precharge wave generation stage for the signals; however, it can be applied to both FPGA and application-specific integrated circuit (ASIC) using standard gates. In the case of DWDDL, WDDL is implemented with two different dual parts: first the positive path, with the AND/OR gates, and the complemented part, interchanging the gates for OR/AND, respectively. With this technique, the automatic tool places and routes the positive part, and the designer only needs to duplicate the resulting layout and interchange the standard cell gates, thus achieving the differential implementation, as depicted in Figure 5c. The implementation costs, performance, and security levels are expected to be of the same order as in WDDL.



(a) AND/NAND SDDL (b) AND/NAND WDDL (c) Example DWDDL implementation (d) AND/NAND HDRL

Figure 5. Gate-level hiding logic styles: (**a**) SDDL, (**b**) WDDL, (**c**) DWDDL, and (**d**) HDRL.

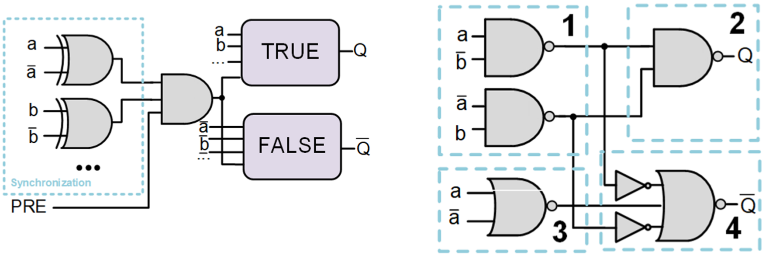
As an improvement of WDDL targeting FPGAs, double-WDDL is proposed in [19,20], which can be considered as a place and routing technique which improves the resistance against DPA attacks presented in [8]. In this case, the authors duplicate the whole WDDL implementation to achieve the same routing path for both WDDL blocks. The main drawback of this logic style is the area and power consumption overhead (2xWDDL). However, the work in [21] demonstrates that this logic style shows a data-dependent time and therefore has information leakage, even when a perfect duplication is achieved. In [22], the authors present a technique to avoid cross-coupling wires between the direct and complementary WDDL circuit paths, called isolated-WDDL (iWDDL), since the direct and complementary paths are isolated from each other. The main drawback of this solution is the area (2xWDDL) and delay overhead (the frequency is halved compared to WDDL), as well as the need to generate additional reset signals to control the precharging of the register's outputs. Both double-WDDL and iWDDL have important area, power consumption, and delay overheads, making them less suitable for low power applications.

Following a similar design to double-WDDL, the authors in [23] present a logic family called homogeneous dual-rail logic (HDLR). To implement an HDLR secure circuit, following the steps proposed in [23], first the original circuit is placed and routed, then, the resulting layout is duplicated and placed next to the original circuit. The inputs to the second circuit will be the complementary inputs of the original circuit. The example of an HDRL AND/NAND gate is shown in Figure 5d. The advantage compared to WDDL is that this logic style does not work with the precharge and evaluation phases. The main disadvantage is the penalty for area and power consumption and the early propagation effect.

The authors of [24] present a new DPL logic style, called dual-spacer dual-rail logic (DSDRL), which tries to eliminate the dependency between data and switching activity in the previously presented dual-rail circuits. This technique uses two spacers, which means that it uses two possible values for circuit signals (complementary signals of a, $\{a_0, a_1\}$, precharge to $\{0, 0\}$ or $\{1, 1\}$) to precharge them to the same voltage value. As an example, for a signal a, composed of $\{a_0, a_1\}$ in dual-rail mode, the precharge spacers will be $\{0, 0\}$ or $\{1, 1\}$ instead of $\{0, 0\}$, such as in the case of SABL or WDDL, alternating in time within the dual-rail logic framework. This technique is based on standard gates and is compatible with conventional digital design flow due to the tool provided by the authors called "Verimap design kit", which successfully interfaces with CADENCE tools. The main drawback is the overhead in area and the complexity added by the use of two spacers.

WDDL and DSDRL suffer from the early evaluation effect [25–27], which the presented secure library (SecLib) overcomes [28]. SecLib is based on standard cell gates, and the designed secure gates are compatible with standard cell libraries. SecLib is a DPL style making changes at protocol, architecture, and backend levels. At the protocol level, it first performs one computation and then reinitializes the nets, placing them in the same electrical state. At the architecture level, they use DPL implementations precharging the nets to "0", and avoiding early propagation by the resynchronization of input arrivals using symmetric Muller C-elements [29]. Finally, they developed a secured routing methodology for differential routing, called shielded design rule checking (DRC)-clean backend duplication. The main drawback of SecLib is the area overhead, which has the advantage of being a logic style compatible with standard libraries.

Another alternative, to avoid the well-known early propagation effect, is balanced cell-based dual-rail Logic (BCDL) [30]. Apart from the dual precharge logic, BCDL includes a synchronization scheme on bundle data that avoids the early propagation effect. A generic BCDL gate is depicted in Figure 6a, where the input *PRE* is the global precharge signal. Among the advantages of using this logic style are the reduced area compared with other DPL logic styles, the elimination of the undesirable early propagation effect, and the capability to detect simple faults when considering fault injection attacks. The drawbacks are the need for a balanced place and route as well as a frequency degradation.



(a) Generic BCDL n-gate

(b) XOR/XNOR RCDDL

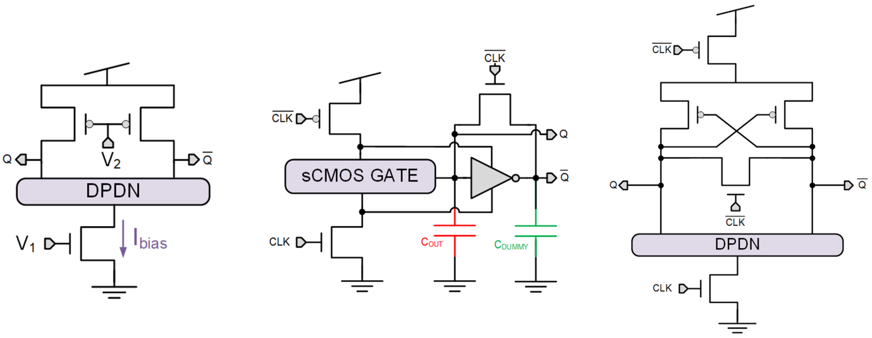
Figure 6. Gate-level hiding logic styles: (**a**) BCDL and (**b**) RCDDL.

Another WDDL-like approach that aims to remove the early propagation effect called DPL-noEE is presented in [26]. DPL-noEE is presented as an alternative to WDDL for FPGA implementations, but, with minor modifications, it can also be adapted for ASICs. DPL-noEE is inspired by BCDL, which also tries to remove the undesirable early evaluation effect. The input signals $a/\overline{a}$ and $b/\overline{b}$ are connected to the same look-up table (LUT) that gives the output $Q/\overline{Q}$ depending on the gate function. This DPL-based logic is obtained with a mask-encoding implementation of the LUT for each combinational gate having two operation phases. In the precharge phase, both inputs are invalid when both ($a/\overline{a}$) or ($a/\overline{a}$) have (0/0) or (1/1) values. In the evaluation phase, the input values are valid when ($a/\overline{a}$) or ($b/\overline{b}$) have (0/1) or (1/0) values. For the DPL-noEE, the authors use a custom tool that converts the single-rail implementation to dual-rail called vDuplicate. In addition to routing issues, the work of [31] showed that DPL-noEE only prevents early propagation in the evaluation phase; however, the transitions at the precharge phase are still data dependent.

To completely remove the early propagation effect on DPL-noEE, another FPGA-based logic style to counteract power analysis attacks without the early propagation effect in both precharge and the evaluation phases is presented in [31], called asynchronous WDDL (AWDDL). For this purpose, an asynchronous design of WDDL is proposed where the FPGA LUTs are used to generate the gate outputs depending on the function of the gate, as in DPL-noEE. On the contrary, the use of emulated S-R latches in AWDDL and the asynchronous design concept guarantees no early propagation effect in both precharge and evaluation phases [31]. As in other DPL families, AWDDL need a balanced place and route process; thus, to mitigate the routing imbalances, a customized place and route tool is used. Although AWDDL improves the early propagation effect, it is noticeable that even when using specific place and route tools to achieve a symmetric implementation in FPGAs, it is impossible to achieve a fully differential routing leading to small leakages, but they can be drastically reduced.

Reduced complementary dynamic and differential logic (RCDDL) [32], was presented as an alternative to WDDL. RCDDL was designed to improve WDDL in terms of security strength and average power consumption, but making it compatible with WDDL gates, so they could be used in conjunction to achieve improved logic functionality. In RCDDL, the complemented output $\overline{Q}$ is implemented by reusing part of the non-complemented output logic tree Q. As in WDDL, to ensure dynamic operation, the precharge phase is propagated as a wave along the circuit. RCDDL has two operation phases: (i) precharge phase, where the logical "0" is propagated as a wave through all differential inputs of the circuit; and (ii) evaluation phase, which produces the output Q and its complement $\overline{Q}$. The structure of an RCDDL gate is depicted in Figure 6b, where four parts can be identified. Segments 1 and 2 constitute the uncomplemented logic designed as a sum of products of the expression; segments 1, 3, and 4 constitute the complementary logic, where the outputs from segment 1 are inverted and provided to segment 4, then segment 3 generates the precharge signals and segment 4 is the force gate generating the complement of the function in segment 2. The main drawback of RCDDL is that when connecting in serial several RCDDL gates, there is a differential delay introduced between the inputs to the force gate (segment 1) during evaluation state, which, for certain inputs, causes glitches in the output. Moreover, the design complexity (gate design is not straightforward and force gates need resizing) as well as the area penalty make RCDDL not a good logic style option for low-power secure applications.

The authors in [33] present the low-power variant of the metal-oxide semiconductor (MOS) current mode logic (MCML) for DPA-resistant applications (initially designed for low-power and high-speed applications) achieving low switching noise due to its reduced output voltage swing and differential operation. The structure of an MCML gate, depicted in Figure 7a, is composed of three different blocks: (i) the current source, implemented by the bottom n-channel metal-oxide semiconductor (NMOS) transistor providing a constant bias current; (ii) the DPDN implemented with NMOS transistors and realizing the functionality of the gate; and (iii) the load resistors, implemented with two p-channel metal-oxide semiconductor (PMOS) transistors serving as active resistors. The main drawback is the high static power consumption of the gate, which is not a good option for portable devices and medium- and low-frequency applications. To solve this problem, the authors in [34] present the logic style called power-gating MCML (PG-MCML). This logic style is based on the power-gating technique in which sleep transistors are inserted into the power supply. The main drawback of this logic style is the design complexity, but it has the advantage of having reduced power consumption values.



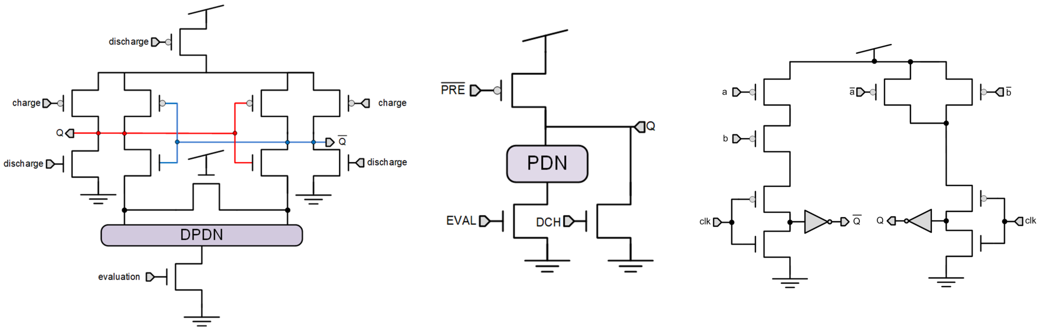
(a) Generic MCML n-gate (b) Generic 3sDL n-gate (c) Generic 3sDDL n-gate

Figure 7. Gate-level hiding logic styles: (**a**) MCML, (**b**) 3sDL, and (**c**) 3sDDL.

In [35] the authors present a dynamic logic style called 3-state dynamic logic (3sDL) based on signals with three possible states: logical "1" with the value of V_{dd}, logical "0" with the value of GND, and finally a third state with the value of $V_{dd}/2$. The structure of this logic style is depicted in Figure 7b, where the differential outputs Q and $\overline{Q}$ go from $\{1,0\}$ or $\{0,1\}$ for the function value "1" or "0", respectively, in the evaluation phase with $CLK = 1$, and then to $V_{dd}/2$ in the precharge phase with $CLK = 0$. The main advantages of this logic style are that the operation frequency is faster than other DPL styles because the swing required for the transition is halved, starting from $V_{dd}/2$, and the direct cascability of 3sDL gates. The main drawbacks are the need for differential routing and the design of the capacitance C_{DUMMY} exactly equal to the value of C_{OUT}.

The same feature of using three-state logic was also used in [36], presenting the three-state differential dynamic logic (3sDDL). The structure of 3sDDL is depicted in Figure 7c. With $CLK = 0$, the gate is in the precharge phase with Q and $\overline{Q}$ to $V_{dd}/2$ value, in the evaluation phase with $CLK = 1$, the outputs Q and $\overline{Q}$ go to values $\{1, 0\}$ or $\{0, 1\}$, respectively, depending on the logic function implemented in the DPDN. It has the same advantages as 3sDL, but in this case it is not necessary to design the capacitance C_{DUMMY}. As main drawbacks, it has the need for differential place and route as well as a good symmetry in the DPDN, and the fact that it requires a full custom design.

In [37], the authors present the three-phase dual-rail precharge logic (TDPL), as an enhancement of the SABL logic style, to avoid the need of differential routing due to imbalances on output capacitances. The TDPL structure is depicted in Figure 8a, and it works in three different phases, namely precharge, evaluation, and discharge phases. In the precharge phase, the output nodes are precharged to V_{dd}; in the evaluation phase, one of the differential branches of the DPDN is discharged by implementing the functionality of the gate and having in the output the values of Q and $\overline{Q}$ as $\{1, 0\}$ or $\{0, 1\}$; and, finally, in the discharge phase, the output Q or $\overline{Q}$ with value V_{dd} is also discharged. As the output wires are precharged to V_{dd} and discharged to GND, the power consumption of the gate remains constant even in the presence of output capacitance mismatches, which is the main advantage of this logic style. The main drawbacks are the increment in the power consumption and the need of generating $charge$, $evaluation$, and $discharge$ signals following a specific timing diagram.



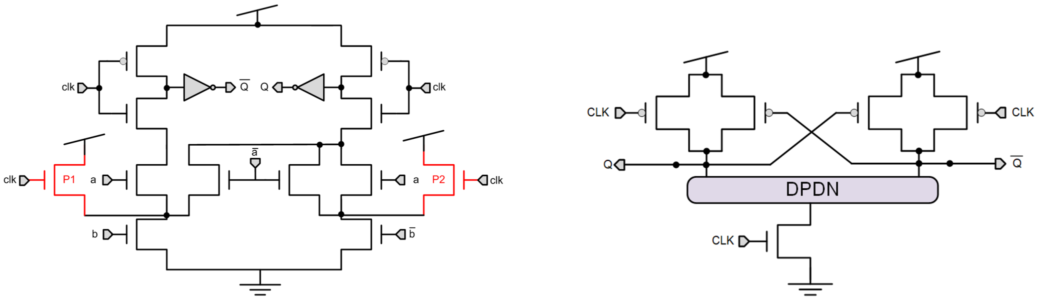
(a) Generic TDPL n-gate (b) Generic TSPL n-gate (c) DDPL AND/NAND gate

Figure 8. Gate-level hiding logic styles: (**a**) TDPL, (**b**) TSPL, and (**c**) DDPL.

In [38], the authors present the three-phase single-rail precharge logic (TSPL), based on TDPL but which eliminates the need for complementary outputs. TSPL is a dynamic logic family that operates with three different phases, namely, precharge (PRE, the output Q is charged to V_{DD}), evaluation (EVAL, the output Q is discharged depending on the inputs and the implemented gate functionality), and discharge (DCH, the output is always discharged). In Figure 8b, the schematic of the generic TSPL n-gate is depicted. The main advantages are the single rail implementation scheme and the resilience against process variations as it does not require special place and routing to balance output loads. The main disadvantages are the security level, being more vulnerable than TDPL or WDDL, and the overhead area.

As in the case of TSPL, the delay-based dual-rail precharge logic (DDPL) is insensitive to unbalanced load conditions, so it allows for the use of a semi-custom design flow with automatic place and routing without any constraints in the routing of the complementary wires [39]. However, in contrast to TDPL, it operates in two phases, namely precharge (output lines precharged to V_{DD} value) and evaluation phases (both output lines discharged to V_{SS}), but maintaining the insensitivity to unbalanced load conditions. DDPL is based on time-enclosed logic (TEL) encoding [40], where the information is represented in the time domain, the logical value 1 being represented by a positive relative delay in the output lines Q and $\overline{Q}$, and the logical value 0 by a negative relative delay in Q and $\overline{Q}$. This means that to represent logical 1, as both differential branches are discharged to V_{SS} in the evaluation phase, the positive branch Q is discharged at a specific time before the negative branch $\overline{Q}$. In the other case, to represent 0, the negative branch $\overline{Q}$ is discharged before the positive branch Q. Therefore, as in TDPL, both outputs are precharged and discharged within the operating cycles. Due to the chosen data encoding, a single control signal is sufficient as in standard dual-rail logic. The structure of DDPL is shown in Figure 8c. The main advantage of DDPL is that it is insensitive to unbalanced outputs; thus, a standard place and routing process can be applied. As main drawbacks, it needs a standard CMOS to DDPL converter, where the delay of the output signal transitions must be specified, and that it suffers from the well-known early propagation effect [41].

To avoid this early propagation effect, the authors in [41] present an optimization of the logic-style DDPL based on TEL encoding plus the DPDN optimization methodologies presented in [42,43], called improved DDPL (iDDPL). The aim of iDDPL is to balance not only the energy in a clock cycle, but also the instantaneous power consumption, reducing also the area and power consumption compared to other logic styles [41]. The operation of the iDDPL gate is the same as in DDPL and its scheme is shown in Figure 9a. Its structure is a combination of the DDPL [39] gate, and the DPDN optimization presented in [42] plus the $dual - switch$ optimization method presented in [43]. First, in [42], the authors present a design methodology to achieve fully symmetrical DPDNs and avoid the early propagation effect, where it must be ensured that the same number of NMOS transistors in series is maintained, creating a discharge path for each internal node of the DPDN, and trying to have the same number of transistors connected to output nodes of the DPDN structure. Thus, the gate will operate with a constant delay (RC value), regardless of the specific input values. Second, as shown in Figure 9a, in the iDDPL gate there are two NMOS transistors (P1 and P2) connected to the internal nodes of the DPDN of the gate to avoid the memory effect due to internal capacitance of the pull-down network. This modification was first presented in [43], where a design methodology for the optimization of DPDN networks of generic DPL gates was presented. Two different methods are presented to eliminate the memory effect of the internal nodes: the $single - switch$ and the $dual - switch$ solution. In the first case, one PMOS transistor is added to equalize the voltage at both internal nodes of the differential DPDN branches. In the second case, two PMOS transistors connected to V_{DD} are placed in the internal nodes to fix their voltage to V_{DD} in the precharge phase, which is the selection of the iDDPL gate structure. The combination of DDPL structure along with DPDN modifications makes iDDPL superior to DDPL in terms of security, avoiding the early propagation effect. The main drawback of this logic style is the need of a full custom library and the need of a CMOS to iDDPL converter, as in the case of DDPL.



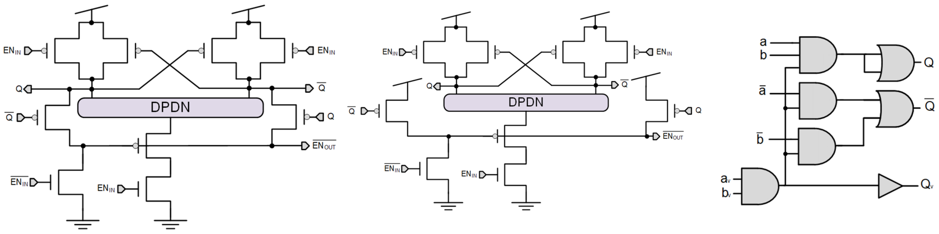
(a) iDDPL AND/NAND gate

(b) Generic DDCVSL n-gate

Figure 9. Gate-level hiding logic styles: (**a**) iDDPL and (**b**) DDCVSL.

Modified domino differential cascode voltage switch logic (DDCVSL) was also first presented as an alternative to the standard static CMOS logic family, which tends to be faster and requires fewer transistors [44], but not for security applications as in the case of DyCML. DDCVSL is based on the basic differential cascode voltage switch logic (DCVSL) [45] and its first evaluation for security application was in [46]. The structure of the DDCVSL, depicted in Figure 9b, is composed of DPUN and DPDN structures, with a clocked NMOS transistor connecting the DPDN with GND. DDCVSL has first a precharge phase where both outputs are set at the same voltage value; then, in the evaluation phase, both the output Q and $\overline{Q}$ are generated, having then always one, and only one, output transition. Although being faster and smaller than most of the gate-level logic styles evaluated in this section, the security level depends to a great extent on the symmetry of the layout and the place and route process. Its main drawbacks are the need for a full custom library design, as well as the impossibility of using a standard design flow.

Low swing current mode logic (LSCML) is presented in [46] as a self-timed differential logic style. It consists of a dynamic current source, a precharge circuit, a DPDN that implements the logic function, a latch to maintain the logic output value after evaluation, and a feedback circuit onto the dynamic current source, implemented by two PMOS transistors, as shown in Figure 10a. It obtains better results in terms of security compared with the DyCML logic style, but at the cost of degrading the maximum operating frequency, area, and power consumption. Again, as a full custom logic style, we need a special design process, since it is impossible to apply a standard design flow. A negative aspect of LSCML is that the source capacitances of the PMOS transistors in the feedback circuit increase the parasitic capacitances at the output nodes, and this negatively affects both the delay and power consumption. To improve this aspect, the authors in [47] present the improved feedback low swing current mode logic (IFLSCML), where the sources of the PMOS transistors in the feedback circuit will be connected to V_{DD}, as depicted in Figure 10b.



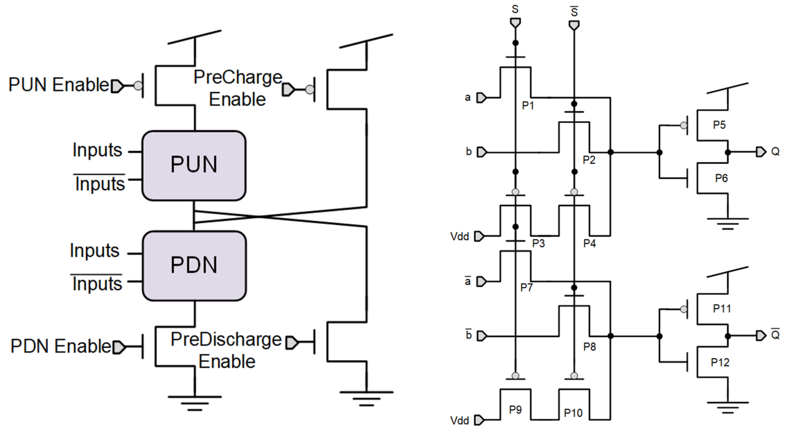
(a) Generic LSCML n-gate (b) Generic IFLSCML n-gate (c) AND/NAND STTL

Figure 10. Gate-level hiding logic styles: (**a**) LSCML, (**b**) IFLSCML, and (**c**) STTL.

As both the LSCML and IFLSCML gates have the feedback circuit that limits the performance of the gates slowing the completion signal, the authors in [47] present the dynamic differential swing limited logic (DDSLL). The aim of the proposal of this logic style is to achieve similar security levels to SABL or DyCML but with less degradation in performance. Similarly to the DyCML logic style, DDSLL operates in a self-timing scheme, featuring a precharge phase where all outputs are charged to V_{DD} and an evaluation phase giving as output the Q and its complemented $\overline{Q}$. The operating frequency is better than DyCML, but the overhead in area and power consumption and security degradation do not place this logic as a clear choice over other logics.

As seen above, the dual-rail logic styles appear as an interesting option to counteract DPA attacks [48] by striving to have a power consumption independent of the data being processed. However, the achieved security level strongly depends not only on the logic gates itself but also on the place and routing process. The expected security level is achieved if and only if both the power consumption and the propagation delays of dual-rail gates are data-independent according to the following assumptions: (i) all the inputs of the gates are controlled by identical drivers; (ii) the switching process starts always at the same time; and (iii) both the non-complemented Q and complemented output $\overline{Q}$ nodes are loaded by capacitances of identical value [48]. To eliminate this weakness of dual-rail logic, authors in [49] present the secure triple-track logic (STTL) as an enhancement of WDDL. The main characteristics of STTL are the quasi-data-independent computation time and power consumption, achieved thanks to the introduction of a third rail indicating whenever the output data are stable and valid or not. In the structure of STTL, it is depicted in Figure 10c, where the signals a_v, b_v, and Q_v are the validity signals. The main drawback of this logic style is the power consumption increment as well as the complexity and impact on the frequency given the addition of a third rail to control the availability of data (this signal is intentionally delayed with regard to the data signal pairs). The main advantage is that it avoids the early propagation effect and can be implemented using both full-custom logic or standard cells.

As a complement to existing higher-level DPA countermeasures, authors in [50] present the randomized multitopology logic (RMTL), with which they try to solve the problems resulting from the physical implementation of other logic styles that are still vulnerable given the process variations that can never be perfectly symmetric. The RMTL logic style focuses on gate-level randomization as a hardware-implementation-level solution, where each gate of a circuit can be configured during run time to have different power profiles. The random selection of the used topology is made with a control signal that can be generated using a random number generator (RNG). The RMTL gate is based on standard CMOS logic and is composed of a pull-up network (PUN) and a pull-down network (PDN), with the addition of four transistors. The schematic of a generic RMTL n-gate is depicted in Figure 11a. The main advantage of this logic style is that it is not vulnerable to process variations as it does not need a perfect symmetry in its physical implementation. The main disadvantage is that it is based on the power consumption randomization and is widely known to be vulnerable to DPA attacks, if not implemented along with other complementary countermeasures, and there is need of an RNG.



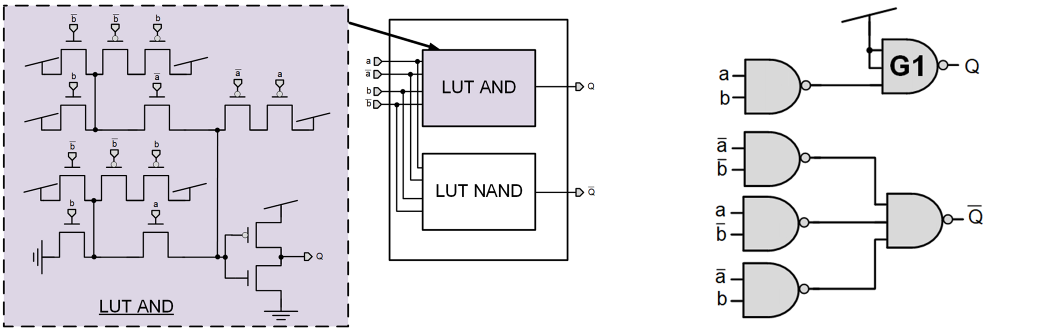
(a) Generic RMTL n-gate (b) SDML$_P$ gate

Figure 11. Gate-level hiding logic styles: (**a**) RMTL and (**b**) SDML$_P$.

The authors of [51] present a DPL scheme called glitch-free duplication (GliFreD) that has been exclusively designed for FPGA platforms. The proposal is a combination of the approach presented in [52] which considers a register at the output of each LUT and the work presented in [53], where each LUT is enabled by at least one global signal. In this sense, GliFreD uses two basic components, the LUTs and flip-flops (FFs), where at least one FF is placed right after the LUTs to avoid direct propagation from one LUT to the other. The whole circuit is duplicated to obtain the complemented circuitry. GliFreD overcomes the well-known early propagation issue, prevents glitches, uses an isolated dual-rail concept, and mitigates imbalanced routing. It is also possible to implement GliFreD in a pipeline fashion, called GliFreD-P [51]. The main drawback of GliFreD is the need to generate two additional control signals, $active1$ and $active2$, apart from clk in a specific configuration with respect to clk. This translates into an increment in design complexity and an area increment due to the required placement of at least one FF between two connected LUTs.

As an alternative to existing logic styles, secure differential multiplexer logic using pass transistors (SDML$_P$) was presented [54]. For the pass-transistors logic styles, although extensively analyzed from the perspective of area and power consumption, their use for security applications was first introduced in [54]. More concretely, the complementary pass-transistor logics (CPL) would be a potential candidate due to their symmetrical structure and differential logic for secure applications. However, they are unable to fulfill the requirement to have a single switching event per clock cycle. To solve this issue, SDML$_P$ is introduced based on CPL, as depicted in Figure 11b. As in DPL styles, SDML$_P$ has two transition networks, one controlling the evaluation (NMOS network) and the other one the predischarge phases (PMOS network). In Figure 11b, transistors P3, P4, P9, and P10 are used to propagate the predischarge signal when both S and $\overline{S}$ are forced to 0 in the predischarge phase, and transistors P1, P2, P7, and P8 control the evaluation phase when S and $\overline{S}$ are valid. Unfortunately, SDML$_P$ also suffers from the early propagation effect, being vulnerable to DPAs.

Following a similar design strategy to SDML$_P$, the authors of [55] presented the look-up-table-based differential logic (LBDL), combining the idea of LUT and differential logic. Instead of storing the gate functionality in bit cells and being the address of the input signal, the authors replace directly the bit cells with V_{DD} and GND to compose the LUT-based logic gates. As an example, an AND/NAND LBDL structure is shown in Figure 12a. Note that the gate functionality configuration is achieved with the V_{DD} and GND inputs on the left, the NMOS transistors compose the evaluation tree, and the PMOS transistors the precharge logic. Although the operating frequency is decreased and the average power consumption is greater than WDDL, for the same area overhead, LBDL shows less power consumption variations. However, it requires a full custom library design, as well as a balanced place and routing process.



(a) LBDL AND/NAND gate

(b) Generic SC-DDPL AND/NAND gate

Figure 12. Gate-level hiding logic styles: (**a**) LBDL and (**b**) SC-DDPL.

Standard cell delay-based dual-rail precharge logic (SC-DDPL) [56] is presented as an alternative logic style implemented with standard gates, and is secure even in the presence of capacitive mismatch at the output signals and is suitable to be implemented both in FPGA or ASIC. SC-DDPL is based on standard cell DPL logic styles using the TEL encoding scheme. Remember that other logic styles based on return-to-zero (RTZ) protocol encode the logic values in the voltage domain, whereas in TEL, the logic value is encoded in the time domain as a difference in time between the two dual-rail signals. The structure of a SC-DDPL AND/NAND gate is depicted in Figure 12b. Notice that a three-input NAND gate (G1) is placed with two of its inputs connected to V_{DD} to preserve gate symmetry and load capacitances. Obviously, one needs a CMOS to TEL standard cell converter to adapt the single rail circuitry outputs provided to the SC-DDPL circuitry.

To protect cryptographic implementations not only from DPA attacks but also from timing attacks, the dual-spacer dual-rail delay-insensitive logic (D^{3}L) was proposed [57]. It is based on threshold gates and decouples power consumption from the processed data by using dual spacers, separating timing–data correlation by inserting random delays. Instead of using a precharge value as in RTZ (0 value), in D^{3}L two spacers are used: the (0/0) and (1/1). The data are valid when inputs $a/\overline{a}$ are 0/1 or 1/0, and the signals are precharged alternatively in each clock cycle between 0/0 and 1/1. Although D^{3}L prevents power and timing attacks, it is important to notice that D^{3}L counteracts only timing attacks if randomized delays are inserted into the gate design. This implies adding extra circuitry to the gates composed of two PMOS and four NMOS transistors, and control signals to change the randomized delay values [58]. This implies extra design effort and performance degradation.

It is clear from the above-presented approaches that a large amount of logic styles and gate-level hiding countermeasures have been presented in the last years. To summarize the most relevant solutions, Table 1 presents an overview of the most important characteristics. To better compare them, the performance and security figures presented are normalized compared to the static CMOS counterpart. This standardization has been carried out as follows. In the case of parameters such as area, maximum operating frequency, and power consumption, we have relied on the data presented by the reference sources included in columns 1 and 2, where the countermeasure overhead is calculated directly with the reference CMOS implementation also provided by the authors. Note that this is important in order to be able to compare the countermeasures with each other, as they can be FPGA or ASIC implementations, or just simulation-based gate-level results. For example, related to area values, some related works present their results as number of transistors, while others as occupied area on an ASIC, or even the number of LUTs on an FPGA. In the cases where authors do not directly compare their implementations with CMOS counterparts, the overhead values are depicted with regard to another countermeasure, which are directly compared in the same referenced work. When the authors provide results relative to different implementations, using the same countermeasure and design flow, a range of maximum and minimum values are presented.

It is important to keep in mind that there are numerous ways of evaluating the security levels of countermeasures against DPA. Some present "indirect" measurements of security levels, in which direct DPA attacks metrics are not present, but offer a value that estimates the complexity of the attack to break our system. Others present "direct" metrics that give as a result the number of traces needed to break the system but do not characterize the acquisition system. Although the same metric may be used to determine the security levels, the measurement setups differ greatly, where we can find FPGA or ASIC implementations, different algorithms of a different nature, or simply measurement environments where the equipment is not the same or simply the results are simulation-based. For this reason, herein a factor of security is presented, with respect to the implementation of the CMOS standard, based on the same attack scenario performed by the same authors.

Qualitative data for the security level are expressed as > or < in reference to a specific proposal, indicating if a proposal is more or less secure than the related work, when the authors do not provide specific data. The lack of a standard procedure to measure the security of a given proposal makes it unfeasible to provide a full quantitative comparison.

Table 1 also includes a mention of the used design methodology: dedicated full-custom *vs* automatic back-end standard CMOS design process (STD). In the same way, also explained when the proposals were presented, it is mentioned if they require special place and route for additional DPA protection. In this sense, semi-custom design flows have been proposed to achieve a more accurate balance of differential parasitic capacitance lines. For example, in [17], the "fat-wire" method is presented to achieve balanced interconnect loads. Their approach can be applied on top of commercial electronic design automation (EDA) tools and consists of routing each output pair as one fat wire, the width of the fat wire being one of two parallel wires, and then splitting the fat wire into the two differential lines. Another approach to differential place and routing, also compatible with commercial EDA tools, is the "backend duplication" method presented in [18]. In this approach, the design is first placed and routed using single-ended gates, and then duplicating the resulting design adding the complementary gates. This implies using standard gates or splitting the differential full custom gates into their complemented and non-complemented functions to apply this method, having the consequent increment in area and power consumption in the case of full-custom designs. A more detailed comparison is presented in Section 5.

Table 1. Gate-level hiding performance and security-normalized values with respect to CMOS logic style.

Logic Style	Data Reference	Based on	Area	Frequency	Power	Security	Special P&R
CMOS	NA [1]	STD [2]	1	1	1	1	NO
3sDDL [36]	[36]	FC [3]	2.64	1.13	3.14	47.5–85.25	YES
3sDL [35]	[35]	FC	3–4	0.67–0.71	2.95–3.11	7.69–50.78	YES
AWDDL [31]	[31,51]	STD	4.36	<WDDL	>>CMOS	>DPL-noEE	YES
BCDL [30]	[30]	STD	1.71	0.70	NA	>20	YES
CRSABL [14]	[14,59]	FC	2.12	0.55	4.56	0.38–0.45 × SABL	YES
D^3L [57]	[57,58]	STD	>>CMOS	<<CMOS	>>CMOS	>>CMOS	NO
DDCVSL [44]	[46]	FC	~0.8 × DyCML	~1.68 × DyCML	~2.70 × DyCML	~1.16 × DyCML	YES
DDPL [39]	[39]	FC	~0.75 × SABL	NA	~1.86 × SABL	~7.97 × SABL	NO
DDSLL [60]	[47,60,61]	FC	1.13–1.45	0.37	1.37	>LSCML	YES
Double-WDDL [20]	[19,20]	STD	11.69	0.19	12	>WDDL	YES
DPL-noEE [26]	[26]	STD	>WDDL	NA	>WDDL	>WDDL	YES
DSDRL [24]	[24,25]	STD	2.08–2.27	~WDDL	~2.11–2.27	>CMOS	NO
DWDDL [8]	[8]	STD	~WDDL	~WDDL	~WDDL	~WDDL	YES
DyCML [15]	[9,15,36,59]	FC	0.81–2.66	0.58–3.7	0.47–4.99	9.6–27.29	YES
GliFred [51]	[51]	STD	1.36	5.06	1.5	>CMOS	NO
GliFred-P [51]	[51]	FC	4.29	5.42	1.7	>>CMOS	NO
HDRL [23]	[23]	STD	2	1	2	>WDDL	NO
iDDPL [40]	[40,41]	FC	3.50	1.76 × SABL	4.40	>SABL	NO
IFLSCML [47]	[47]	FC	~1.10 × DyCML	~DyCML	~0.77 × DyCML	~DyCML	YES
iWDDL [22]	[22]	STD	2 × WDDL	0.5 × WDDL	~2 × WDD	>Double-WDDL	YES
LBDL [55]	[55]	STD	0.98 × WDDL	1.16 × WDDL	1.44 × WDDL	>WDDL	YES
LSCML [46]	[46]	FC	~1.10 × DyCML	~0.64 × DyCML	~1.09 × DyCML	>DyCML	YES
MCML [33]	[33,34]	FC	2.53	0.90	2360	>CMOS	YES
PG-MCML [34]	[34]	FC	2.57	0.88	0.20	>CMOS	YES
RCDDL [32]	[32]	STD	3.94	0.17	3.70	>WDDL	NO
RMTL [50]	[50]	STD	1.50	0.25	3.50	>CMOS	NO
SABL [7]	[7,9,36,40,62]	FC	1.8–3.27	0.51 × DyCML	1.91–7.98	14–166	YES
SC-DDPL [56]	[56]	STD	7.18	NA	4.2	>2.42 × WDDL	NO
SDDL [8]	[8,20]	STD	2 × WDDL	0.35	2 × WDDL	>CMOS	YES
SDMLp [54]	[54]	STD	1.41	0.67	1.07	>WDDL	YES
SecLib [28]	[28,30,63–65]	STD	15.09–28.00	0.25–0.5	4.05	~10 × WDDL	NO
STTL [49]	[49]	FC/STD	5.68	0.22	NA	>CMOS	NO
TDPL [37]	[37,38]	FC	>SABL	0.71–0.96	1.47–2.83	>SABL	NO
TSPL [38]	[38]	FC	1.3	1.20	1.14	>WDDL	NO
WDDL [8]	[8,19,23,30,32,38,54,56,63,64,66,67]	STD	2.47–11.82	0.20–0.94	1.70–13.50	6.02–119.73	YES

[1] NA = not available. [2] STD: standard cells. [3] FC: full-custom.

4. Gate-Level Masking

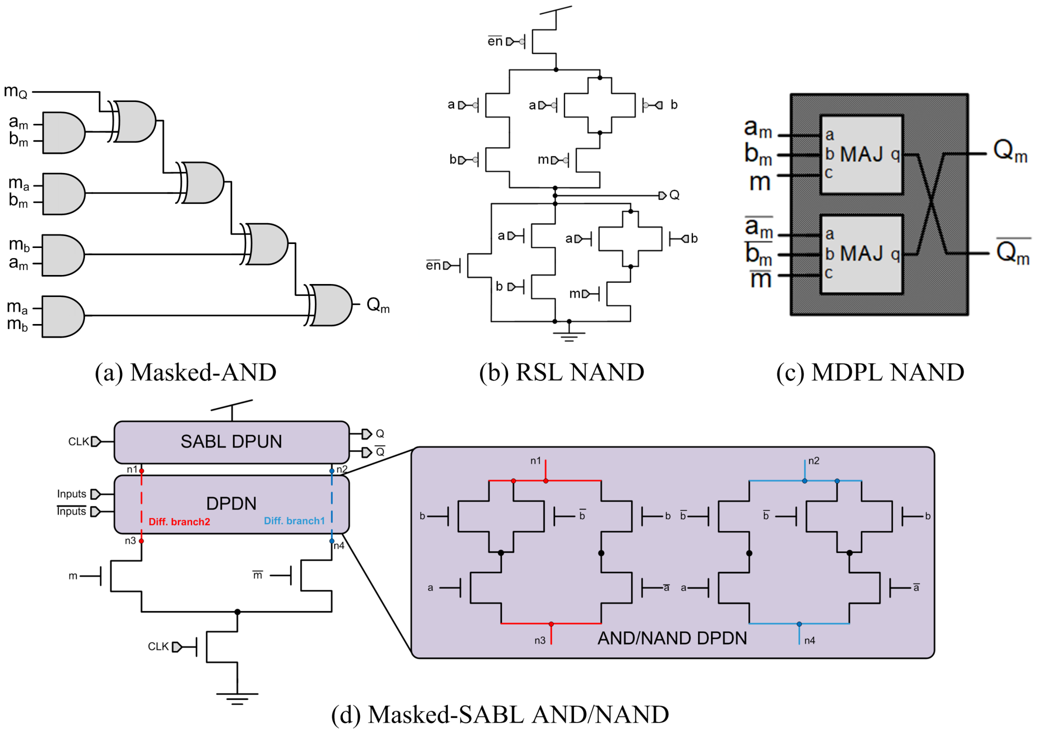
Depending on the complexity of the design or the security level required for a given implementation, there are three different ways in which the mask values associated with a gate-level masking implementation can be generated [3,68]: (i) use one mask for each generated signals; (ii) use a single mask for a group of signals; or (iii) use the same mask for the entire implementation.

These strategies result in greater or lesser complexity in the design phase. For example, using a single mask for the entire implementation allows for a smaller implementation, but is less secure than using one mask for each generated signal. Consequently, this last option implies an increase in the design complexity since a greater number of random masks are needed for the correct operation of the countermeasure.

In this section, a detailed description of the structure and functionality of each masking logic style proposed in the literature is provided. Following the indications of the previous sections, the reader can go to the end of the section for a summary of all the main features and comparison.

One of the first approaches of gate-level masking was presented in [12] in 2001, where functions are masked using one of the two different methods: look-up table masking or the use of multiplexer trees apparatus. Although these techniques are typically applicable on an higher design level, single masked gates can also be implemented. Specifically, the application of this multiplexer method was also studied in [69], where two additional techniques are presented to implement masking at the gate level, namely, the use of the XOR technique to mask the AND and OR gates, and the implementation of the multiplexer technique using AND and OR gates. This study was later extended in [70].

In 2003 [71], the masked-AND logic style was presented (further analyzed and studied in [69,72] in 2004). This masked-AND gate, depicted in Figure 13a, implements the AND operation of a and b, with Q as output. The gate has as inputs the masked values a_m and b_m, and the respective mask values m_a and m_b outputting the mask m_Q and the masked output value Q_m. Given that, this work only focuses on the combinational logic for the advanced encryption standard (AES) substitution-box (Sbox) implementation, using XOR and AND gates, and given that the XOR gates are linear operation, not needing to be unmasked, the masking manipulation is performed on the AND gates. The masked-AND technique requires the generation of a mask for each of the signals, implying the generation of many masks as there are signals in the implementation. The area increase is approximately ×3.86, while the frequency degradation is ×0.58. The power consumption is expected to increase proportionally to the increase in area caused by the masked AND gates [10].



(a) Masked-AND (b) RSL NAND (c) MDPL NAND

(d) Masked-SABL AND/NAND

Figure 13. Gate-level masking hardware countermeasures.

The previous presented implementations [12,69,71], although protected by "secure" schemes, still leak side-channel information in presence of glitches which can be exploited by DPA attacks [73]. Given this, a logical style based on standard cells resistant to DPA attacks in the presence of glitches called FGL was presented [74].

At the same time that gate-level masking countermeasures were presented, gate-level hiding countermeasures were also introduced, with great results in terms of security.

The first approaches using gate-level hiding countermeasures try to equalize the power consumption of the gate using complementary operations, being dependent on wire length or fan-out, which often makes the design very difficult. To solve this problem, the authors in [10] propose the masked logic style called random switching logic (RSL). This logic style does not require complementary operations and processes original signals and a mask simultaneously having the following two properties: (i) RSL uses the same random mask for all the signals and (ii) needs an enable signal that executes operations while $en = 1$, otherwise drives 0. Figure 13b depicts a NAND gate implemented in RSL logic style, where a, b are input signals, $\overline{en}$ the complemented enable signal, m is the random mask, and Q is the output. The authors also present a solution for FPGA implementations, called RSLUT, based on look-up table implementations. The main design drawbacks of RSL are that it needs a full-custom gate design and the generation of the enable signal and the random mask.

In [11], the authors present MDPL (masked dual-rail precharge logic), a masked logic style that prevents glitches by using the dual-rail precharge principle. It can be classified as a hiding or masking countermeasure as it is a mix of both techniques. Hence, for each masked signal a_m, its complementary masked signal $\overline{a_m}$ is also present in the circuit, every signal being masked with the same mask m. The MDPL AND gate, based on CMOS majority gates, is depicted in Figure 13c, having six dual-rail inputs (a_m, $\overline{a_m}$, b_m, $\overline{b_m}$, m, $\overline{m}$) and producing two output values (Q_m, $\overline{Q_m}$). The main design drawback of this implementation is the need to generate a random mask but a single one for the entire design.

Unfortunately, later it was shown that the RSL and MDPL implementations were vulnerable to the well-known early propagation effects [27,75], with successful attacks presented in [76,77]. To solve this problem in RSL, a new logic style was presented, called DRSL (dual-rail random switching logic) [78]. DRSL prevents glitches by using a precharging protocol to reduce early propagation effects while removing routing constraints, introducing a random mask. DRSL is based on RSL (e.g., uses an RSL NAND gate with an evaluation-precharge detection unit (EPDU) to implement an NAND DRSL gate [78]) and MDPL, but as an advantage over MDPL by avoiding side-channel leakage caused by asynchronous inputs. However, according to the analysis in [76], DRSL does not completely avoid the early propagation effect in the precharge phase. The input signals arriving at different moments can still precharge the DRSL gates given that the EPDU still allows for some precharge to happen. The main design drawback is the need to generate a mask for the entire implementation and the differential implementation of the design, although it does not need special place and routing. Another improvement of DRSL is briefly presented in [79], consisting of implementing DRSL in positive logic. This solution has a cost in CMOS logic, since inverting gates are smaller than non-inverting ones.

To solve the early propagation effect in MDPL, a new logic style called iMDPL (improved MDPL) was proposed [76], where the authors include an EPDU, which generates 0 at its output only if all input signals are in a differential state. Evaluation of iMDPL gates over CMOS and MDPL presented in [80] suggests that the main drawback of this design is the need to generate a random mask for the whole design. In [76], MDL was also shown to have lower security levels than DPL logics [7,8,37].

As in the case of MDPL, where both hiding and masking techniques are implemented together in the same logic style to withstand DPA attacks, masked-SABL is also presented [81]. In this case, the authors include in the SABL logic style the use of a mask m to prevent the unbalance produced in the circuit connections due to aging. Modification of the SALB logic style is performed in the DPDN structure, as shown in Figure 13d. In each of the differential branches, an NMOS transistor is placed, one controlled by the mask m and the other one with its complement signal $\overline{m}$. This implies complementing or not the functionality of the gate depending on the mask value, having at the output $Q(\overline{Q})$ the result of the function $f(\overline{f})$ implemented by the gate if the mask is 0 and $\overline{f}(f)$ if the mask is 1. It is important to note that the security level reached by SABL depends directly on the imbalances of the output capacitance, so this enhanced logic style is expected to improve the security level even if the problem of aging is not taken into account. However, it still requires a differential place and routing process to reach the highest security level. The main drawbacks are the area overhead, as the DPDN is almost duplicated compared with standard SABL, and the need for a full-custom design.

Despite the masking logic styles presented above, using a single bit to mask is still susceptible to DPA attacks, as presented in [82]. The authors conclude that the mask bit value can be determined by a first analysis of the power consumption traces. For this reason, logic styles that use both masking and hiding techniques are potentially capable of achieving better security levels, as well as resistance against aging effects, as considered in [81].

Table 2 summarizes the performance and security values for the discussed approaches, normalizing with respect to the CMOS logic style. Data have been obtained from different sources, so Table 2 shows in some cases ranges of values or direct comparisons with other implementations. For more information, please refer to the references indicated in the second column.

Table 2. Gate-level masking performance and security normalized values with respect to CMOS logic style.

Logic Style	Data Reference	Based on	Area	Frequency	Power	Security	Special P&R
CMOS	NA[1]	STD	1	1	1	1	NO
DRSL [78]	[78,79]	FC/STD [2]	2–7.5	0.50	>CMOS	>MDPL [3]	NO
iMDPL [76]	[80]	STD	18–19	0.2–0.3	>MDPL	90–120	NO
MASKED-AND [71]	[10]	STD	3.86	0.56	NA	>CMOS [4]	NO
Masked-SABL [81]	[81]	FC	1.19 × SABL	0.88 × SABL	=SABL	3 × SABL	YES
MDPL [11]	[11,76,83]	STD	4–5	0.5–0.6	17.43	1.69	NO
RSL [10]	[10]	FC/STD [2]	2.02	0.68	NA	>MASKED-AND [4]	NO

[1] NA = not available. [2] Full-custom (FC) for RSL and based on standard gates (STD) in FPGA implementations RSLUT. [3] CMOS < MDPL < WDDL < DRSL. [4] CMOS < MASKED-AND < WDDL < RSL.

5. Comparative Analysis

The comparison between performances, features, and security levels of these proposals is not easy to carry out, given the variety of approaches and considered technologies. However, a comparative analysis is presented here using the normalized values shown in Tables 1 and 2. This analysis starts by first comparing countermeasures in the same category. Comparison between masking and hiding techniques can be unfair due to the different nature of the operations. The presented analysis is based on the figures resulting from the area *vs* delay cost of each solution and from the relation between area delay product *vs* security. The first one gives an idea of the complexity and operation speed, while the second one represents the security with respect to the general performance cost. Overall, the selection of one approach rather than another depends on the required security level and the resulting impact in terms of performance and cost, which depends on the specific application and available technology.

5.1. Gate-Level Hiding

To better analyze the values presented in Table 1, they are depicted in Figures 14 and 15, showing the area vs. delay and the area-delay product (ADP) vs. security metrics. To clarify the clustered values, zoom-in view is also shown.

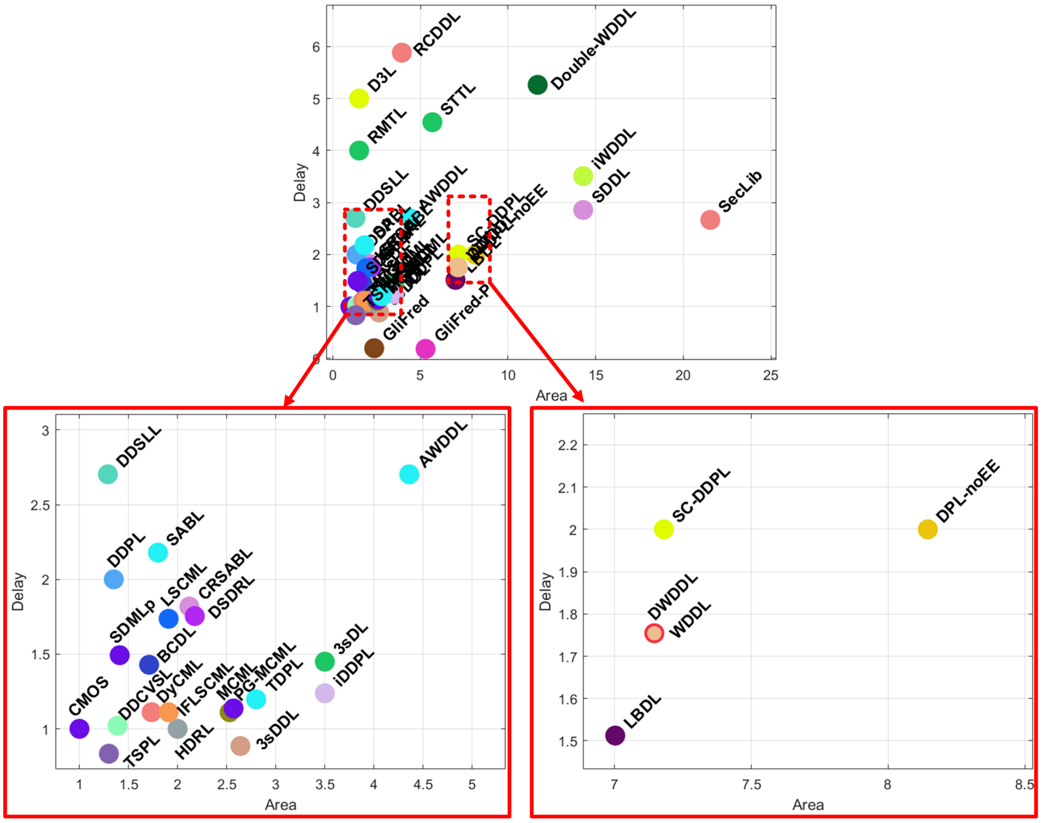


Figure 14. Area vs. delay of gate-level hiding countermeasures.

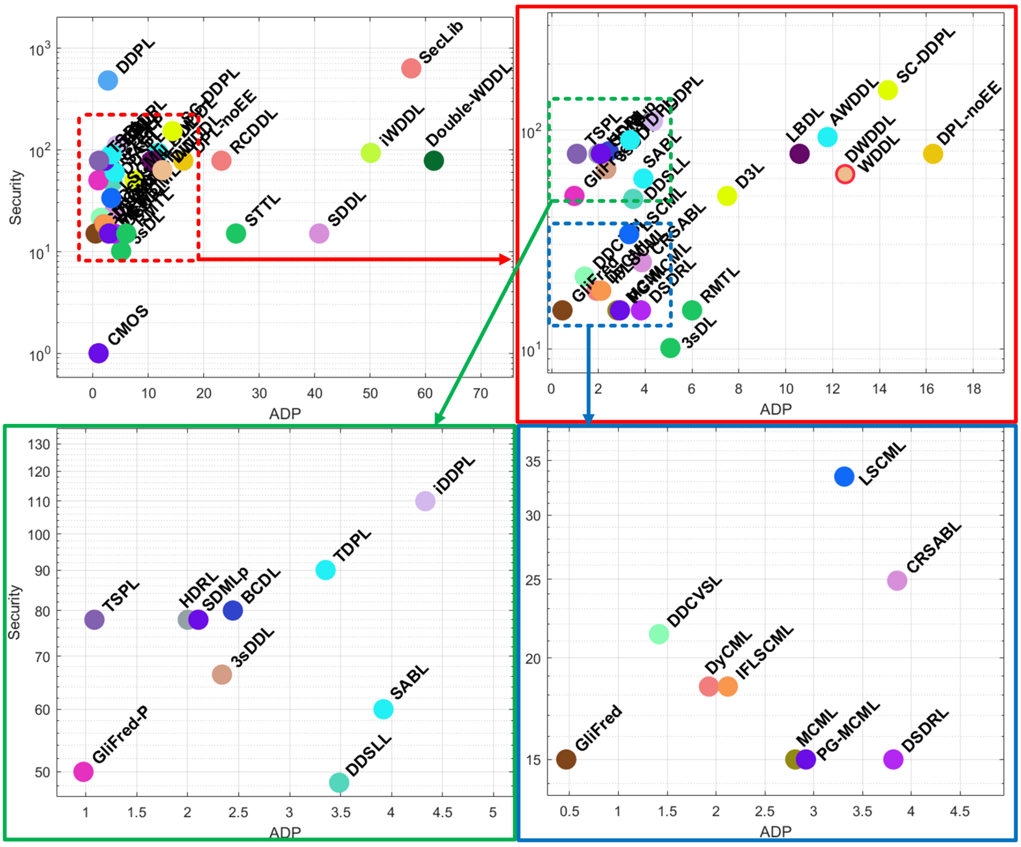


Figure 15. ADP vs. security of gate-level hiding countermeasures.

From Figure 14, it can be clearly seen that the 26 most cost-effective proposals are located in the bottom left corner. The potential selection for low area and with a low performance impact should be kept within these zoomed squares. These values show that all the hiding techniques require extra hardware compared to CMOS, with GliFred and GliFred-P being the fastest ones. It can also be seen that CML-based and SABL-based solutions have a linear growth between area and speed, with TSPL being the one with the best area–performance ratio. As seen below, this solution also presents an interesting security level.

Regarding the security level, Figure 15 depicts the distribution of these solutions according to their security levels *vs* ADP values, with the upper-left corner ones suggesting the best security cost trade-off.

It is clear that the lack of quantified values for the security level does not facilitate a fair comparison. Often the only information provided by the authors is "this proposal is more secure than that proposal". To fairly assess each solution in terms of their relative security, the following relation was used, based on the comparisons presented on each state-of-the-art paper:

- In [36], DyCML < SABL < 3sDDL.
- In [31], WDDL < DPL_noEE < AWDDL.
- In [14], CRSABL < SABL.
- In [46], DyCML < DDCVSL < LSCML.
- In [39], DyCML < SABL < DDPL.
- In [47], DyCML < IFLSCML < LSCML < DDSLL.
- In [9], DDCVSL < DyCML < SABL.
- In [51], DPL_noEE < BCDL < AWDDL.
- In [38], WDDL < TSPL < TDPL.

Notice that in [9] and [46], the DyCML and DDCVSL proposals have contradictory results, but it is important to notice that both have security levels within the same order of magnitude.

The zoomed areas contain specific values for the security level in terms of order of magnitude above CMOS. These specific solutions can be considered acceptable. The best security results are clearly achieved by the SecLib and DDPL solutions. However, these require further validation, other than their authors. While SecLib presents a partially higher security level, it does so with a significantly higher area-performance cost. With identical results between each other, BCDL, SDMLp, TSPL, HDRL, and SABL show consistent security-performance figures.

Combining all the figures, BCDL, SDMLp, TSPL, HDRL, and SABL are the most promising proposals, but it should be kept in mind that all of them have inherent difficulties. For example, TSPL requires the operation under a three-phase clock.

5.2. Gate-Level Masking

To better analyze the gate-level masking solutions, depicted in Table 2, Figures 16 and 17 present the respective area vs. delay metric and the ADP vs. security metrics.

From these figures, the iMDPL implementation is clearly seen as the outlier with a significantly higher area cost in relation to the other gate-level masking solutions. Its predecessor, MDPL, has a much better area and delay trade-off; however, its security level is low, since it is vulnerable to DPA attacks [82]. By itself, the masked-AND shows a lower security level with higher costs than RSL. However, if a different mask is used in the masked-AND solution for each circuit signal, it may be more secure than RSL, which is shown in [77] to be potentially compromised if a first filtering is performed to obtain the mask value and then perform the DPA attack.

The DRSL has a good trade-off between security and performance while improving the security level of the RSL by removing the early propagation effect; however, in [82] the authors point out that if a first analysis is made to retrieve the secret mask, the dual-rail characteristic does not increase the security, given the imbalances in the differential wires when no special routing is performed.

Finally, the masked_SABL solution shows that by combining gate-level masking with full-custom hiding techniques, higher security levels can be achieved with good performance trade-offs.

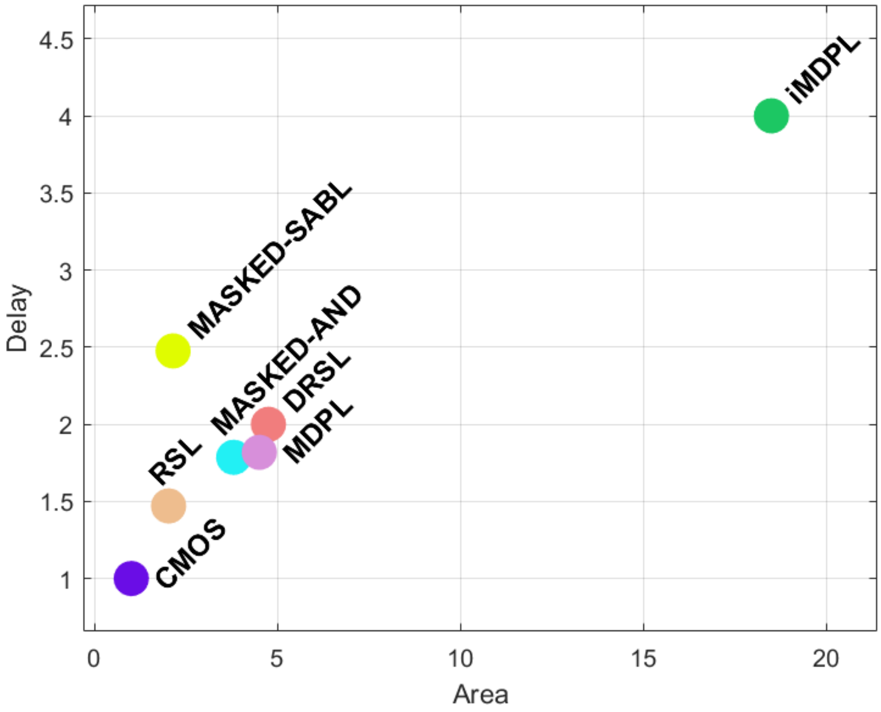


Figure 16. Area vs. delay of gate-level masking countermeasures.

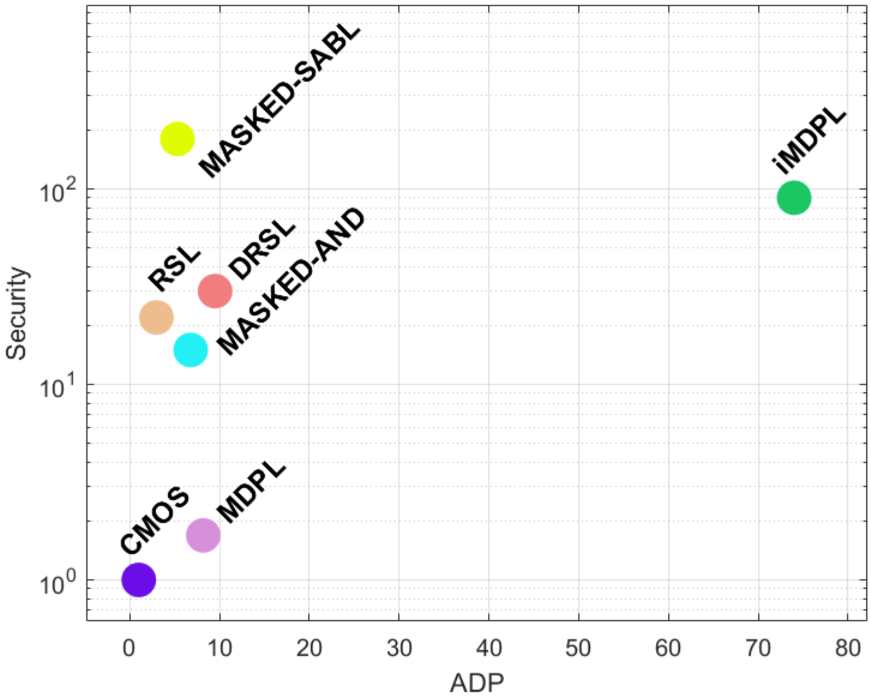
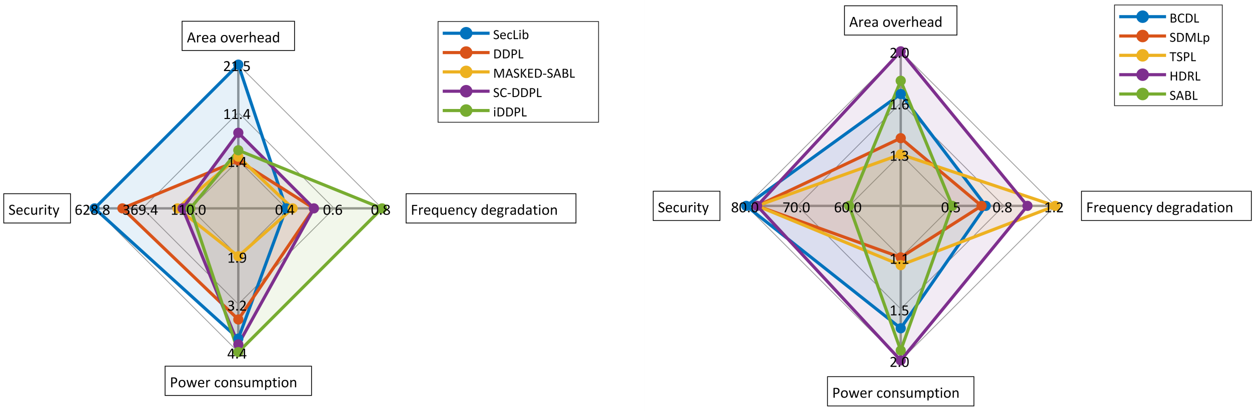


Figure 17. ADP vs. security of gate-level masking countermeasures.

To provide a visual overview of the top five best state-of-the-art countermeasures, Figure 18a,b depict the metrics for area overhead, frequency degradation, power consumption, and resulting security for both masking and hiding.



(a) (b)

Figure 18. Top five countermeasures in security levels (**a**) and top five countermeasures with best trade-off between performance and security levels (**b**).

Figure 18a provides a visual comparison of the top five countermeasures with the best security levels. Figure 18b depicts the top five countermeasures with the best trade-off between security values and ADP performance and area overhead.

From these figures, it can be seen that, typically and as expected, the higher the security, the higher the cost. However, this is not always the case. For example, in Figure 18b it can be seen that the SABL approach has approximately the same power and area costs as BCDL but provides significantly less protection against SCA. Nevertheless, in addition to performance degradation and security levels, it is also important to consider the inherent design difficulties of each proposal, as well as the feasibility of including the countermeasure in the design.

6. Conclusions

In this paper, a deep review of the state of the art of gate-level countermeasures against power analysis attacks has been carried out. The importance of developing new secure logic styles and techniques to be used in secure IoT applications where the security is a crucial issue has generated dozens of proposals claiming high protection against power analysis attacks with reduced area-delay costs. This work splits the existing gate-level solution into two large groups: those following a hiding approach (the power consumption is intended to be the same for all the data processed) and the ones considering a masking procedure (the data are masked and behave as random). The presented analysis considers the most relevant solutions in the literature, 35 hiding proposals, and 6 based on masking, not only by using the data provided by proposing authors, but also those included in the other references for comparison. Advantages and drawbacks of the proposals are analyzed, showing quantified data for cost, performance (delay and power), and estimated security level, when available. This work also visually depicts the performance, cost, and security level relation of the several solutions to better assist cryptodesigners in the selection of the best solution, style according to their constraints. Overall, these results suggest that RSL and DRSL solutions are the best approaches when considering masking, while BCDL, SDMLp, TSPL, HDRL, and SABL are those with best security-performance figures. It can also be concluded that hiding proposals reach higher security levels, but with more difficult design constraints, which, if not met, can result in security weaknesses. Finally, this review also suggests that the combination of masking and hiding, as in masked_SABL, can provide the most secure solution, but at the cost of more complexity.

Author Contributions: Conceptualization, E.T.-S., F.E.P.-O., A.J.A., C.J.J.-F. and R.C.; methodology, E.T.-S., F.E.P.-O. and R.C.; formal analysis, E.T.-S., F.E.P.-O., A.J.A., C.J.J.-F. and R.C.; investigation, E.T.-S., F.E.P.-O. and A.J.A.; resources, A.J.A., C.J.J.-F. and R.C.; data curation, E.T.-S., F.E.P.-O. and A.J.A.; writing—original draft preparation, E.T.-S., F.E.P.-O., A.J.A. and R.C.; writing—review and editing, E.T.-S., F.E.P.-O., A.J.A., C.J.J.-F. and R.C.; visualization, E.T.-S., F.E.P.-O., A.J.A. and R.C.; supervision, A.J.A., C.J.J.-F. and R.C.; project administration, A.J.A., C.J.J.-F. and R.C.; funding acquisition A.J.A. and C.J.J.-F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Grant PID2020-116664RB-I00 funded by MCIN/AEI/10.13039/501100011033.

Acknowledgments: Authors want to acknowledge to projects Programa Operativo FEDER 2014-2020 and Consejería de Economía, Conocimiento, Empresas y Universidad de la Junta de Andalucía under Project US-1380823, to the SPIRS (Secure Platform for ICT Systems Rooted at the Silicon Manfacturing Process) Project with Grant Agreement No. 952622 under the European Union's Horizon 2020 research and innovation programme, to Erasmus+, to COST Action CRYPTACUS, to FCT (Fundação para a Ciência e a Tecnologia, Portugal), to the ERDF (European Regional Development Fund, EU) through the projects FCT: UIDB/50021/2020 and LISBOA-01-0145-FEDER-031901 (PTDC/CCI-COM/31901/2017, HiPErBio) and to Disruptive Sdn seCuRE communicaTIons for eurOpean defeNse, EDIDP-CSAMN-SDN-202-74-DISCRETION.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

3sDDL	3-state Differential Dynamic Logic
3sDL	3-state Dynamic Logic
ADP	Area-Delay Product
AES	Advanced Encryption Standard
ASIC	Application Specific Integrated Circuit
AWDDL	Asynchronous Wave Dynamic Differential Logic
BCDL	Balanced Cell-based Dual-rail Logic
CML	Current Mode Logic
CMOS	Complementary Metal-Oxide Semiconductor
CPL	Complementary Pass-transistor Logics
CRSABL	Charge Recycling Sense Amplifier Based Logic
D^3L	Dual-spacer Dual-rail Delay-insensitive Logic
DCVSL	Differential Cascode Voltage Switch Logic
DDCVSL	Domino Differential Cascode Voltage Switch Logic
DDPL	Delay-based Dual-rail Precharge Logic
DDSLL	Dynamic Differential Swing Limited Logic
DPA	Differential Power analysis
DPDN	Differential Pull Down Network
DPL	Dual Precharge Logic
DPL-noEE	Dual Precharge Logic with no Early propagation Effect
DPUN	Differential Pull Up Network
DRC	Design Rule Checking
DRSL	Dual-rail Random Switching Logic
DSDRL	Dual Spacer Dual-Rail Logic
DWDDL	Divided Wave Dynamic Differential Logic
DyCML	Dynamic Current Mode Logic
EDA	Electronic Design Automation
EPDU	Evaluation-Precharge Detection Unit
FC	Full-custom
FF	Flip Flops
FPGA	Field Programmable Gate Array
GliFreD	Glitch-Free Duplication
GliFred-P	Pipelined fashion Glitch-Free Duplication

HDRL	Homogeneous Dual-Rail Logic
iDDPL	Improved Delay-based Dual-rail Precharge Logic
IFLSCML	Improved Feedback Low Swing Current Mode Logic
iMDPL	Improved Masked Dual-rail Precharge Logic
IoT	Internet of Things
iWDDL	Isolated Wave Dynamic Differential Logic
LBDL	Look-up-table Based Differential Logic
LSCML	Low Swing Current Mode Logic
LUT	Look-Up Table
MCML	Metal-Oxide Semiconductor Current Mode Logic
MDPL	Masked Dual-rail Precharge Logic
MOS	Metal Oxide Semiconductor
NMOS	n-channel Metal-Oxide Semiconductor
PA	Power Analysis
PDN	Pull-Down Network
PG-MCML	Power-Gating Metal-Oxide Semiconductor Current Mode Logic
PMOS	p-channel Metal-Oxide Semiconductor
PUN	Pull-Up Network
RCDDL	Reduced Complementary Dynamic and Differential Logic
RMTL	Randomized MultiTopology Logic
RNG	Random Number Generator
RSL	Random Switching Logic
RTZ	Return to Zero
SABL	Sense Amplifier Based Logic
SAFF	StrongArm110 Flip-Flop
Sbox	Substitution-box
SCA	Side-Channel Attack
SC-DDPL	Standard Cell Delay-based Dual-rail Precharge Logic
SDMLp	Secure Differential Multiplexer Logic using Pass transistors
SecLib	Secure Library
SPA	Single Power analysis
SDDL	Simple Dynamic Differential Logic
STD	Standard
STTL	Secure Triple Track Logic
TDPL	Three-phase Dual-rail Precharge Logic
TEL	Time Enclosed Logic
TSPL	Three-phase Single-rail Precharge Logic
WDDL	Wave Dynamic Differential Logic

References

1. Mahmoud, R.; Yousuf, T.; Aloul, F.; Zualkernan, I. Internet of things (IoT) security: Current status, challenges and prospective measures. In Proceedings of the 10th International Conference for Internet Technology and Secured Transactions (ICITST'15), London, UK, 14–16 December 2015; pp. 336–341.
2. Shahverdi, A.; Taha, M.; Eisenbarth, T. Lightweight Side Channel Resistance: Threshold Implementations of Simon. *IEEE Trans. Comput.* **2017**, *66*, 661–671. [CrossRef]
3. Mangard, S.; Oswald, E.; Popp, T. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*; Springer: Berlin/Heidelberg, Germany, 2007.
4. Kocher, P. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, Other Systems. In Proceedings of the International Cryptology Conference (CRYPTO'96), Barbara, CA, USA, 18–22 August 1996; pp. 104–113.
5. Kocher, P.; Jaffe, J.; Jun, B. Differential Power Analysis. In Proceedings of the International Cryptology Conference (CRYPTO'99), Barbara, CA, USA, 15–19 August 1999; pp. 388–397.
6. Hayashi, Y.; Homma, N.; Mizuki, T.; Aoki, T.; Sone, H.; Sauvage, L.; Danger, J.L. Analysis of Electromagnetic Information Leakage From Cryptographic Devices With Different Physical Structures. *IEEE Trans. Electromagn. Compat.* **2013**, *55*, 571–580. [CrossRef]
7. Tiri, K.; Akmal, M.; Verbauwhede, I. A Dynamic and Differential CMOS Logic With Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards. In Proceedings of the 28th European Solid-State Circuits Conference (ESSCIRC'02), Firenze, Italy, 24–26 September 2002; pp. 403–406.

8. Tiri, K.; Verbauwhede, I.; Hall, B.; Box, P.O.; Angeles, L. A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In Proceedings of the International Conference on Design, Automation and Test in Europe (DATE'04), Paris, France, 16–20 February 2004; pp. 246–251.
9. Macé, F.; Standaert, F.-X.; Hassoune, I.; Quisquater, J.-J.; Legat, J.-D. A Dynamic Current Mode Logic to Counteract Power Analysis Attacks. In Proceedings of the 19th International Conference on Design of Circuits and Integrated Systems (DCIS'04), Bordeaux, France, 24–26 November 2004; pp. 186–191.
10. Suzuki, D.; Saeki, M.; Ichikawa, T. Random Switching Logic: A Countermeasure against DPA based on Transition Probability. IACR Cryptology ePrint Archive. 2004; p. 346. Available online: https://eprint.iacr.org/2004/346.pdf (accessed on 1 February 2022).
11. Popp, T.; Mangard, S. Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'05), Edinburgh, UK, 29 August–1 September 2005; pp. 172–186.
12. Messerges, T.S.; Dabbish, E.A.; Puhl, L. Method and Apparatus for Preventing Information Leakage Attacks on a Microelectronic Assembly. U.S. Patent 6,295,606 B1, 25 September 2001.
13. Nikolic, B.; Oklobdzija, V.G.; Stojanovic, V.; Jia, W.; Chiu, J.K.S.; Leung, M.M.T. Improved Sense-Amplifier-Based Flip-Flop: Design and Measurements. *IEEE J. Solid-State Circuits* **2000**, *35*, 876–884. [CrossRef]
14. Tiri, K.; Verbauwhede, I. Charge Recycling Sense Amplifier Based Logic: Securing Low Power Security IC's against DPA. In Proceedings of the 30th European Conference on Solid-State Circuits (ESSCIR'04), Leuven, Belgium, 21–23 September 2004; pp. 179–182.
15. Allam, M.W.; Elmasry, M.I. Dynamic Current Mode Logic (DyCML): A New Low-Power High-Performance Logic Style. *J. Solid-State Circuits* **2001**, *36*, 550–558. [CrossRef]
16. Alioto, M.; Gaetano, P. *Model and Design of Bipolar and MOS Current-Mode Logic: CML, ECL and SCL Digital Circuits*; Springer: Berlin/Heidelberg, Germany, 2006.
17. Tiri, K.; Verbauwhede, I. Place and route for secure standard cell design. In Proceedings of the Sixth International Conference on Smart Card Research and Advanced Applications (CARDIS'04), Toulouse, France, 22–27 August 2004; pp. 143–158.
18. Guilley, S.; Hoogvorst, P.; Mathieu, Y.; Pacalet, R. The Backend Duplication Method A Leakage-Proof Place-and-Route Strategy for ASICs. In Proceedings of the In International Workshop on Cryptographic Hardware and Embedded Systems (CHES'05), Edinburgh, UK, 29 August–1 September 2005; pp. 383–397.
19. Yu, P.; Schaumont, P. Secure FPGA circuits using controlled placement and routing. In Proceedings of the 5th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS'07), Salzburg, Austria, 30 September 2007; pp. 45–50.
20. Yu, P. Implementation of DPA-Resistant Circuit for FPGA. Ph.D. Thesis, Virginia Tech, Blacksburg, VA, USA, 2007.
21. Wild, A.; Moradi, A.; Güneysu, T. Evaluating the duplication of dual-rail precharge logics on FPGAs. In Proceedings of the International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE'15), Berlin, Germany, 13–14 April 2015; pp. 81–94.
22. Mcevoy, R.P.; Murphy, C.C.; Marnane, W.P. Isolated WDDL: A Hiding Countermeasure for Differential Power Analysis on FPGAs. *ACM Trans. Reconfigurable Technol. Syst.* **2009**, *2*, 3. [CrossRef]
23. Tanimura, K.; Dutt, N.D. HDRL: Homogeneous dual-rail logic for DPA attack resistive secure circuit design. *IEEE Embed. Syst. Lett.* **2012**, *4*, 57–60. [CrossRef]
24. Sokolov, D.; Murphy, J.; Bystrov, A.; Yakovlev, A. Improving the Security of Dual-Rail Circuits. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'04), Cambridge, MA, USA, 11–13 August 2004; pp. 282–297.
25. Sokolov, D.; Murphy, J.; Bystrov, A.; Yakovlev, A. Design and Analysis of Dual-Rail Circuits for Security Applications. *IEEE Trans. Comput.* **2005**, *54*, 449–460. [CrossRef]
26. Bhasin, S.; Guilley, S.; Flament, F.; Selmane, N.; Danger, J.-L. Countering Early Evaluation: An Approach Towards Robust Dual-Rail Precharge Logic. In Proceedings of the Workshop on Embedded Systems Security (WESS'10), Scottsdale, AZ, USA, 24 October 2010; pp. 1–8.
27. Suzuki, D.; Saeki, M. Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'06), Yokohama, Japan, 10–13 October 2006; pp. 255–269.
28. Guilley, S.; Hoogvorst, P.; Mathieu, Y.; Pacalet, R.; Provost, J. CMOS structures suitable for secured hardware. In Proceedings of the International Conference on Design, Automation and Test in Europe (DATE'04), Paris, France, 16–20 February 2004; pp. 1414–1415.
29. Shams, M.; Ebergen, J.C.; Elmasry, M.I. Modeling and comparing CMOS implementations of the C-element. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **1998**, *6*, 563–567. [CrossRef]
30. Nassar, M.; Bhasin, S.; Danger, J.; Duc, G.; Guilley, S.; Effect, A.E.P. BCDL: A High Speed Balanced DPL for FPGA with Global Precharge and no Early Evaluation. In Proceedings of the Conference on Design, Automation and Test in Europe (DATE'10), Dresden, Germany, 8–12 March 2010; pp. 849–854.

31. Moradi, A.; Immler, V. Early Propagation and Imbalanced Routing, How to Diminish in FPGAs. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'14), Busan, Korea, 23–26 September 2014; pp. 598–615.
32. Sundaresan, V.; Rammohan, S.; Vemuri, R. Power Invariant Secure IC Design Methodology using Reduced Complementary Dynamic and Differential Logic. In Proceedings of the International Conference on Very Large Scale Integration (VLSI-SoC'07), Atlanta, GA, USA, 15–17 October 2007; pp. 1–6.
33. Toprak, Z.; Leblebici, Y. Low-Power Current Mode Logic for Improved DPA-Resistance in Embedded Systems. In Proceedings of the International Symposium on Circuits and Systems (ISCAS'05), Kobe, Japan, 26 May 2005; pp. 1059–1062.
34. Cevrero, A.; Regazzoni, F.; Schwander, M.; Badel, S.; Ienne, P.; Leblebici, Y. Power-gated MOS Current Mode Logic (PG-MCML): A power aware DPA-resistant standard cell library. In Proceedings of the 48th Design Automation Conference (DAC'11), San Diego, CA, USA, 5–9 June 2011; pp. 1014–1019.
35. Aigner, M.; Mangard, S.; Menicocci, R.; Olivieri, M.; Scotti, G.; Trifiletti, A. A novel CMOS logic style with data independent power consumption. In Proceedings of the International Symposium on Circuits and Systems (ISCAS'05), Kobe, Japan, 26 May 2005; pp. 1066–1069.
36. Giancane, L.; Marietti, P.; Olivieri, M.; Scotti, G.; Trifiletti, A. A New Dynamic Differential Logic Style as a Countermeasure to Power Analysis Attacks. In Proceedings of the IEEE International Conference on Electronics, Circuits and Systems (ICECS'08), Nabeul, Tunisia, 7–9 November 2008; pp. 364–367.
37. Bucci, M.; Giancane, L.; Luzzi, R.; Trifiletti, A. Three-Phase Dual-Rail Pre-charge Logic. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'06), Yokohama, Japan, 10–13 October 2006; pp. 232–241.
38. Menendez, E.; Mai, K. Extended Abstract: A High-Performance, Low-Overhead, Power-Analysis-Resistant, Single-Rail Logic Style. In Proceedings of the International Workshop In Hardware-Oriented Security and Trust (HOST'08), Anaheim, CA, USA, 9 June 2008; pp. 33–36.
39. Bucci, M.; Giancane, L.; Member, S.; Luzzi, R.; Scotti, G.; Trifiletti, A. Delay-Based Dual-Rail Precharge Logic. *IEEE Trans. Very Large Scale Integr. Syst.* **2011**, *19*, 1147–1153. [CrossRef]
40. Bongiovanni, S.; Centurelli, F.; Scotti, G.; Trifiletti, A. Design and validation through a frequency-based metric of a new countermeasure to protect nanometer ICs from side-channel attacks. *J. Cryptogr. Eng.* **2015**, *5*, 269–288. [CrossRef]
41. Bellizia, D.; Scotti, G.; Trifiletti, A. TEL Logic Style as a Countermeasure Against Side-Channel Attacks: Secure Cells Library in 65nm CMOS and Experimental Results. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2018**, *65*, 3874–3884. [CrossRef]
42. Tiri, K.; Verbauwhede, I. Design method for constant power consumption of differential logic circuits. In Proceedings of the Design, Automation and Test in Europe (DATE'05), Munich, Germany, 7–11 March 2005; pp. 628–633.
43. Tena-Sánchez, E.; Castro, J.; Acosta, A.J. A methodology for optimized design of secure differential logic gates for DPA resistant circuits. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2014**, *4*, 203–215. [CrossRef]
44. Ng, P.; Balsara, P.T.; Steiss, D. Performance of CMOS differential circuits. *IEEE J. Solid-State Circuits* **1996**, *31*, 841–846. [CrossRef]
45. Chu, K.M.; Pulfrey, D.L. Design procedures for differential cascode voltage switch circuits. *IEEE J. Solid-State Circuits* **1986**, *21*, 1082–1087. [CrossRef]
46. Hassoune, I.; Mace, F.; Flandre, D.; Legat, J.-D. Low-swing current mode logic (LSCML): A new logic style for secure and robust smart cards against power analysis attacks. *Microelectron. J.* **2006**, *37*, 997–1006. [CrossRef]
47. Hassoune, I.; Mace, F.; Flandre, D.; Legat, J.D. Dynamic differential self-timed logic families for robust and low-power security ICs. *Integr. VLSI J.* **2007**, *40*, 355–364. [CrossRef]
48. Razafindraibe, A.; Robert, M.; Maurine, P. Formal evaluation of the robustness of dual-rail logic against DPA attacks. In Proceedings of the International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS'06), Montpellier, France, 13–15 September 2006; pp. 634–644.
49. Soares, R.; Calazans, N.; Lomné, V.; Maurine, P.; Torres, L.; Robert, M. Evaluating the Robustness of Secure Triple Track Logic through Prototyping. In Proceedings of the 21st Annual Symposium on Integrated Circuits and System Design (SBCCI'08), Gramado, Brazil, 1–4 September 2008; pp. 193–198.
50. Avital, M.; Dagan, H.; Keren, O.; Fish, A. Randomized Multitopology Logic Against Differential Power Analysis. *IEEE Trans. Very Large Scale Integr. Syst.* **2015**, *23*, 702–711. [CrossRef]
51. Wild, A.; Moradi, A.; Tim, G. GliFreD: Glitch-Free Duplication Towards Power-Equalized Circuits on FPGAs. *IEEE Trans. Comput.* 2018, 67, 375–387, . [CrossRef]
52. Moradi, A.; Mischke, O. Glitch-free implementation of masking in modern FPGAs. In Proceedings of the IEEE International Symposium Hardware-Oriented Security and Trust (HOST'12), San Francisco, CA, USA, 3–4 June 2012; pp. 89–95.
53. He, W.; Otero, A.; Torre, E.D.L.; Riesgo, T. Automatic generation of identical routing pairs for FPGA implemented DPL logic. In Proceedings of the International Conference on Reconfigurable Computing and FPGAs (ReConFig'12), Cancun, Mexico, 9–11 December 2012; pp. 1–6.
54. Ramakrishnan, L.N.; Chakkaravarthy, M.; Manchanda, A.S. SDMLp: On the Use of Complementary Pass Transistor Logic for Design of DPA Resistant Circuits. In Proceedings of the International Symposium on Hardware-Oriented Security and Trust (HOST'12), San Francisco, CA, USA, 3–4 June 2012; pp. 31–36.
55. Vue, D. A Look-Up-Table Based Differential Logic to Counteract DPA Attacks. In Proceedings of the 8th International Conference on ASIC (ASICON'09), Changsha, China, 20–23 October 2009; pp. 855–858.

56. Bellizia, D.; Bongiovanni, S.; Olivieri, M.; Scotti, G. SC-DDPL: A Novel Standard-Cell Based Approach for Counteracting Power Analysis Attacks in the Presence of Unbalanced Routing. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 2317–2330. [CrossRef]
57. Cilio, W.; Linder, M.; Porter, C.; Di, J.; Smith, S.; Thompson, D. Side-channel attack mitigation using dual-spacer Dual-rail Delay-insensitive Logic (D3L). In Proceedings of the IEEE SoutheastCon (SoutheastCon'10), Charlotte, NC, USA, 18–21 March 2010; pp. 471–474.
58. Cilio, W.; Linder, M.; Porter, C.; Di, J.; Smith, S.; Thompson, D. Mitigating power-and timing-based side-channel attacks using dual-spacer dual-rail delay-insensitive asynchronous logic. *Microelectron. J.* **2013**, *44*, 258–269. [CrossRef]
59. Sundström, T.; Alvandpour, A. A comparative analysis of logic styles for secure IC's against DPA attacks. In Proceedings of the NORCHIP Conference (NORCHIP'05), Oulu, Finland, 21–22 November 2005; pp. 297–300.
60. Kamel, D.; Renauld, M.; Bol, D.; Standaert, F.-X.; Flandre, D. Analysis of Dynamic Differential Swing Limited Logic for Low-Power Secure Applications. *J. Low Power Electron. Appl.* **2012**, *2*, 98–126. [CrossRef]
61. Renauld, M.; Kamel, D.; Standaert, F.X.; Flandre, D. Information Theoretic and Security Analysis of a 65-Nanometer DDSLL AES S-Box. In *Cryptographic Hardware and Embedded Systems (CHES'11)*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6917.
62. Tiri, K.; Verbauwhede, I. Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology. In *InInternational Workshop on Cryptographic Hardware and Embedded Systems*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2779, pp. 125–136.
63. Guilley, S.; Flament, F.; Hoogvorst, P.; Pacalet, R.; Mathieu, Y. Secured CAD back-end flow for power-analysis-resistant cryptoprocessors. *IEEE Des. Test Comput.* **2007**, *24*, 546–555. [CrossRef]
64. Guilley, S.; Sauvage, L.; Hoogvorst, P.; Pacalet, R.; Bertoni, G.M.; Chaudhuri, S. Security Evaluation of WDDL and SecLib Countermeasures against Power Attacks. *IEEE Trans. Comput.* **2008**, *57*, 1482–1497. [CrossRef]
65. Guilley, S.; Sauvage, L.; Flament, F.; Vong, V.; Hoogvorst, P.; Pacalet, R. Evaluation of Power Constant Dual-Rail Logics Countermeasures against DPA with Design Time Security Metrics. *IEEE Trans. Comput.* **2010**, *59*, 1250–1263. [CrossRef]
66. Wang, H.; Tiri, K.; Hodjat, A. AES-based security coprocessor IC in 0.18 µm CMOS with resistance to differential power analysis side channel attack. *IEEE J. Solid-State Circuits* **2006**, *41*, 781–791. [CrossRef]
67. Rammohan, S.; Sundaresan, V.; Vemuri, R. Reduced complementary dynamic and differential logic: A CMOS logic style for DPA-resistant secure IC design. In Proceedings of the IEEE International Frequency Control Symposium and Exposition, Miami, FL, USA, 4–7 June 2008; pp. 699–705.
68. Chaves, R.; Chmielewski, Ł.; Regazzoni, F.; Batina, L. SCA-Resistance for AES: How Cheap Can We Go? In Proceedings of the International Conference on Cryptology in Africa, Cairo, Egypt, 9–11 July 2018; pp. 107–123.
69. Golic, J.D.; Menicocci, R. Universal masking on logic gate level. *IEEE Electron. Lett.* **2004**, *37*, 898–899. [CrossRef]
70. Golic, J.D. Techniques for random masking in hardware. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2007**, *54*, 291–300. [CrossRef]
71. Trichina, E. Combinational Logic Design for AES Subbyte transformation on Masked Data. IACR Cryptology ePrint Archive. 2003; p. 236. Available online: https://eprint.iacr.org/2003/236.pdf (accessed on 1 February 2022).
72. Trichina, E.; Korkishko, T.; Lee, K.H. Small Size, Low Power, Side Channel-Immune AES Coprocessor: Design and Synthesis Results. In Proceedings of the International Conference on Advanced Encryption Standard, Bonn, Germany, 10–12 May 2004; pp. 113–127.
73. Mangard, S.; Popp, T.; Gammel, B.M. Side-channel leakage of masked CMOS gates. In Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, 14–18 February 2005; pp. 351–365.
74. Fischer, W.; Gammel, B.M. Masking at Gate Level in the Presence of Glitches. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'05), Edinburgh, UK, 29 August–1 September 2005; pp. 187–200.
75. Moradi, A.; Kirschbaum, M.; Eisenbarth, T.; Paar, C. Masked Dual-Rail Precharge Logic Encounters State-of-the-Art Power Analysis Methods. *IEEE Trans. Very Large Scale Integr. Syst.* **2012**, *20*, 578–1589. [CrossRef]
76. Popp, T.; Kirschbaum, M.; Zefferer, T.; Mangard, S. Evaluation of the Masked Logic Style MDPL on a Prototype Chip. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'07), Vienna, Austria, 10–13 September 2007; pp. 81–94.
77. Tiri, K.; Schaumont, P. Changing the odds against Masked Logic. In Proceedings of the International Workshop on Selected Areas in Cryptography, Montreal, QC, Canada, 17–18 August 2006; pp. 134–146.
78. Chen, Z.; Zhou, Y. Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'06), Yokohama, Japan, 10–13 October 2006; pp. 242–254.
79. Danger, J.; Guilley, S.; Bhasin, S.; Nassar, M. Overview of Dual Rail with Precharge Logic Styles to Thwart Implementation-Level Attacks on Hardware Cryptoprocessors. In Proceedings of the International Conference on Signals, Circuits and Systems (SCS'09), Medenine, Tunisia, 6–8 November 2009; pp. 1–8.
80. Kirschbaum, M.; Popp, T. Evaluation of a DPA-Resistant Prototype Chip. In Proceedings of the Annual Computer Security Applications Conference (ACSAC'09), Washington, DC, USA, 7–11 December 2009; pp. 43–50.
81. Fadaeinia, B.; Anik, M.T.H.; Karimi, N.; Moradi, A. Masked SABL: A Long Lasting Side-Channel Protection Design Methodology. *IEEE Access* **2021**, *9*, 90455–90464. [CrossRef]

82. Schaumont, P.; Tiri, K. Masking and Dual-Rail Logic Don't Add Up. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'07), Vienna, Austria, 10–13 September 2007; pp 95–106.
83. Popp, T.; Thomaspoppiaiktugrazat, E.; Mangard, S. Implementation Aspects of the DPA-Resistant Logic Style MDPL D-fitipfloDPs. In Proceedings of the International Symposium on Circuits and Systems (ISCAS'06), Singapore, 4–7 December 2006; pp. 2913–2916.