Journal of
*Low Power Electronics
and Applications*

MDPI

*Article*

# Time- and Amplitude-Controlled Power Noise Generator against SPA Attacks for FPGA-Based IoT Devices

**Luis Parrilla *, Antonio García, Encarnación Castillo, Salvador Rodríguez-Bolívar and Juan Antonio López-Villanueva**

Departamento de Electrónica y Tecnología de Computadores, Universidad de Granada, 18071 Granada, Spain
* Correspondence: lparrilla@ditec.ugr.es; Tel.: +34-958240482

**Abstract:** Power noise generation for masking power traces is a powerful countermeasure against Simple Power Analysis (SPA), and it has also been used against Differential Power Analysis (DPA) or Correlation Power Analysis (CPA) in the case of cryptographic circuits. This technique makes use of power consumption generators as basic modules, which are usually based on ring oscillators when implemented on FPGAs. These modules can be used to generate power noise and to also extract digital signatures through the power side channel for Intellectual Property (IP) protection purposes. In this paper, a new power consumption generator, named Xored High Consuming Module (XHCM), is proposed. XHCM improves, when compared to others proposals in the literature, the amount of current consumption per LUT when implemented on FPGAs. Experimental results show that these modules can achieve current increments in the range from 2.4 mA (with only 16 LUTs on Artix-7 devices with a power consumption density of 0.75 mW/LUT when using a single HCM) to 11.1 mA (with 67 LUTs when using 8 XHCMs, with a power consumption density of 0.83 mW/LUT). Moreover, a version controlled by Pulse-Width Modulation (PWM) has been developed, named PWM-XHCM, which is, as XHCM, suitable for power watermarking. In order to build countermeasures against SPA attacks, a multi-level XHCM (ML-XHCM) is also presented, which is capable of generating different power consumption levels with minimal area overhead (27 six-input LUTS for generating 16 different amplitude levels on Artix-7 devices). Finally, a randomized version, named RML-XHCM, has also been developed using two True Random Number Generators (TRNGs) to generate current consumption peaks with random amplitudes at random times. RML-XHCM requires less than 150 LUTs on Artix-7 devices. Taking into account these characteristics, two main contributions have been carried out in this article: first, XHCM and PWM-XHCM provide an efficient power consumption generator for extracting digital signatures through the power side channel, and on the other hand, ML-XHCM and RML-XHCM are powerful tools for the protection of processing units against SPA attacks in IoT devices implemented on FPGAs.

**Keywords:** power noise generation; power masking; SPA attacks; power watermarking; IoT

check for updates

**Citation:** Parrilla, L.; García, A.; Castillo, E.; Rodríguez-Bolívar, S.; López-Villanueva, J.A. Time- and Amplitude-Controlled Power Noise Generator against SPA Attacks for FPGA-Based IoT Devices. *J. Low Power Electron. Appl.* **2022**, *12*, 48. https://doi.org/10.3390/jlpea12030048

Academic Editor: Stefania Perri

Received: 28 July 2022
Accepted: 8 September 2022
Published: 10 September 2022

## 1. Introduction

Security of the information generated and managed by Internet of Things (IoT) devices is one of the main current challenges [1] within the IoT context. Several advances have been achieved in the last years, introducing encryption in communication protocols such as MQTT-SN [2], optimizing cryptographic algorithms to be executed in low-cost microcontrollers [3], or using low-cost FPGAs to implement IoT platforms including hardware cryptoprocessors to add public-key cryptography [4]. These security features are focused on the protection of the information when entering or leaving the device, but information being processed by the microcontroller or microprocessor should also be protected against Side Channel Attacks (SCA) [5]. In this sense, power consumption is one of the main side channels widely used to extract sensitive information from IoT devices [5,6], processing systems [7–9] and to attack cryptographic algorithms implemented on hardware [10–12] or software [13]. These attacks are based on acquiring power traces from the target system and analyzing them later. Depending on the type of analysis performed over these power traces, it is possible to distinguish three main power SCAs:

- Simple Power Analysis (SPA) consists of information being extracted directly from power traces [14]. This type of attack does not require specialized instrumentation and allows extracting information from a low number of power traces; thus, it is critical to implement some kind of countermeasure to prevent these attacks. These countermeasures are usually based on masking operations executed in the system [15] by randomizing the order in which instructions are executed [16], or masking the data involved in some operations by means of arithmetic or boolean operations with random values [17]. Other very effective countermeasures consist of generating "power noise" to prevent useful information from being extracted from the power traces [18,19].
- Differential Power Analysis (DPA) is based on analysis of the statistical correlation of acquired power traces [20]. Attacks based on DPA are more difficult to avoid than those based on SPA, especially if they are combined with other techniques such as Artificial Intelligence (AI) [21], because the effects of masking or noise introduction can be overridden. On the other hand, this type of attack requires the acquisition of a high number of power traces, and frequent changes of the secret key can be an effective countermeasure in the particular case of cryptographic circuits. In this case, a Public-Key Cryptosystem (PKC) is usually required, which may be provided by hybrid cryptoprocessors such as in [22] or [4] in order to enable the completion of these frequent changes securely. Other countermeasures in the literature include advanced masking [23] or sophisticated power noise generation [24].
- Correlation Power Analysis (CPA) is based on the same principles as DPA, and the only difference is that CPA makes use of the correlation factor instead of plain correlation [25] to guess the secret key or any other relevant information. It provides similar results and also require a high number or power traces [26,27].

In the case of hardware or software implementations of cryptographic algorithms, the operations to be performed are always the same, they are well known, and it is possible to obtain thousands of power traces generated with the same private key. All these facts make feasible the use of CPA or DPA attacks. In the case of processing units, CPA and DPA attacks are more difficult to perform because the instructions to be executed are not known and can vary due to interrupts, interaction with other devices, communications, data values, etc. On the other hand, with a well-trained classifier, it is possible to perform SPA attacks to disassemble the program being executed [28]. In addition, an SPA is simple to apply [10,29]; thus, it is critical to implement countermeasures against it. Indeed, it can be applied by direct observation of power consumption traces using an oscilloscope or a DC power analyzer. Its simplicity makes it affordable in a variety of situations [9,11], but it is also possible to establish countermeasures [30,31].

In the case of attacks based on monitoring power consumption, one of the most used countermeasures consists of generating additional power consumption by means of dedicated circuits to hide the information included in the power traces of the target circuit or algorithm [32]. These dedicated circuits generating significant power consumption are usually based on Ring Oscillators (RO) [33–36] that can be activated or deactivated by means of a control signal, thus allowing control of the time when the consumption peaks are generated. Other structures such as the High Consuming Module (HCM) presented in [37] may be used for this purpose, but both ROs and HCMs have the drawback that once implemented, it is not possible to control the amplitude of the corresponding power consumption peak. Additionally, HCMs have been proven to be a useful tool to extract digital signatures through the side channel in order to protect the Intellectual Property (IP) of devices.

In this paper, we propose two modifications of the HCM in [37] to both optimize the power consumption generated per LUT in FPGAs and to allow its real-time control.

The first modification, named Xored High Consumption Module (XHCM), can be controlled by means of Pulse-Width Modulation (PWM), taking advantage of the high current consumption range offered by these elements to achieve a fine-grain control of the generated amplitude. This also enables new features for power watermarking [38] and IP core protection. The other modification, named Multi-Level Xored High Consuming Module (ML-XHCM), enables the generation of different power consumption levels with minimal area overhead, thus being suitable to implement countermeasures against SPA attacks. The rest of the article is organized as follows: Section 2 revises power-based side channel attacks and the available countermeasures; Section 3 presents different circuits used to generate additional power consumption in FPGAs and introduces the Xored High Consuming Modules, showing experimental results and comparing them to HCMs; Section 4 is devoted to the description of the two proposals of controlled XHCMs for the protection against SPAs of information processed in FPGA-based IoT platforms; Section 5 summarizes the experimental results and the comparison to other power noise generators; and Section 6 presents some conclusions and future work.

## 2. Previous Work

In this section, we introduce the most common side channel attacks described in the literature for processing and cryptographic circuits, as well as the available countermeasures for SPAs, focusing on those based on the generation of power consumption noise.

### *2.1. Side Channel Attacks on Computing and Cryptographic Circuits*

The first side channel attacks were developed to attack cryptographic circuits, as in [20], where an SPA on a smart card performing Data Encryption Standard (DES) [39] operations is described. This shows the importance of protecting circuits managing sensitive information against power consumption analysis. This type of attack may lead to obtaining the secret key, but on the other hand, there are some countermeasures available in the literature to hinder it. These countermeasures are mainly based on avoiding conditional jumps and carefully studying the timing of the different operations when protecting software implementations [40], or improving dynamic power consumption to mask power variations. Further, in [20], the basis for a DPA on DES is described. This attack is more difficult to avoid because it is based on the statistical correlation of a set of power consumption traces, but at the same time, the need for thousands of power traces enables the frequent change of the secret key as a countermeasure. Nowadays, DES has been abandoned due to its security issues, and the Advanced Encryption Standard (AES) has emerged as the new standard for symmetric encryption. AES construction is different from that of DES, but it presents similar vulnerabilities to SPA and DPA attacks, as reported in [10,29,41,42]. Additionally, the acquisition of power traces can be combined with the setup of a system of algebraic equations in order to obtain the secret key. This technique is known as algebraic crypto-analysis, and some works have reported successful attacks on AES, although the solving of these systems of equations is not trivial [43,44].

Other powerful techniques based on different principles are the one proposed in [45], where fault injection is combined with side channel attacks; the proposal in [21], where AI is combined with correlation power analysis (CPA); the use of collision attacks as in [46]; or the use of AI combined with EM attacks [47]. Although exploitation of side channels was originally intended to attack cryptographic algorithms, their application has been extended to general computing systems [7,8], microcontrollers used in IoT devices [6,28,48] and, more recently, to systems implementing neural networks [9,49]. In these processing systems, a high amount of sensitive information is managed and, even if the exchange and transmission of information is usually protected by means of cryptographic algorithms, it is vulnerable to power analysis of the processing system itself. Moreover, in these processing systems, the countermeasures developed for cryptographic circuits are not always suitable. Indeed, in a cryptographic algorithm the sequence of operations is well-known, and it is possible to use countermeasures such as masking [23], which can be effective against SPA, CPA and DPA attacks. Nevertheless, in general computing systems, the attacks are oriented to find out what instructions are being executed and thus to try to extract associated data. In this case, the attacks are based on previous training of a classifier in order to recognize the power patterns generated by the instructions and a later SPA over the target system [28,48]. Therefore, it is required to implement countermeasures against SPA attacks in both computing and cryptographic circuits.

### *2.2. Countermeasures for SPA Attacks*

There are different proposals of countermeasures for SPA, DPA and CPA attacks. As has been previously commented, DPA and CPA attacks are more effective than SPA attacks when applied to cryptographic circuits, as they always perform the same operations, and it is possible to collect sets containing thousand of power traces generated using the same private key to apply statistical correlation techniques. These countermeasures are also effective against SPA attacks, and there are several approaches to protect software and hardware implementations of cryptographic algorithms, particularly for AES, which currently is the most extended symmetric cipher. In the case of hardware implementations, these countermeasures are mainly based on masking and hiding techniques [50]. Regarding software implementations, masking techniques try to conceal information being processed by applying randomly generated masks to intermediate values using arithmetic [15] or boolean [17] operations. This makes it difficult to identify peaks in power traces with specific intermediate values of AES (or any other cryptographic algorithm) operations. Hiding techniques try to hinder the extraction of information from power traces in two dimensions: time and amplitude. In the case of hiding in the time domain, the idea is to randomize the order in which some operations are performed [23], while hiding in the amplitude domain implies introducing modifications to power consumption [51]. Note that when considering the protection of computing systems or IoT microcontrollers, the instructions and operations may be unknown or may be very different depending on the data or the interaction with external systems. In this context, CPA or DPA techniques are not feasible, and SPAs are the main concern in the protection of the power side channel in such systems. The main proposals for hardware SPA protection in the literature are summarized in the following:

- Use of voltage regulators. In [52], it is shown that Low-Dropout Regulators (LDO) help to de-correlate the input current from the current drawn by the circuit. A more advanced proposal is carried out in [53], where a converter-gating technique including a multi-phase switched-capacitor converter is used for de-correlating currents. These solutions are intended for ASIC implementations of the circuit under protection, where the voltage regulator is included in the same chip.
- Generating power noise. This technique considers power consumption as an output signal and generates "noise" on that output to hide the contents of this signal [54]. These methods make use of Finite State Machines [32] or ring-oscillators to generate such noise [19].
- Masking arithmetic or boolean operations. As has been previously commented, a method to hide the results of cryptographic operations is to mask the operands or the results with boolean or arithmetic operations with random values. In [55], the addition of a randomly generated mask is proposed for protecting AES against SPA and DPA, while [56] proposes the use of multiplication as the masking arithmetic operation. As an example of the use of boolean masking, [57] proposes the use of the $XOR$ operation. These techniques are not suitable for microcrontollers used in IoT because they require additional area and increment the processing time of software programs.
- In the case of processing systems, the combination of the techniques above with the specific design of some modules of the microprocessor can lead to the implementation of designs resistant to power attacks. An example can be found in [58], in which a RISC-V processor is modified in order to be resistant to power side channel attacks, but at the cost of severe area overhead.

Taking into account the advantages and drawbacks of each protection method against SPAs, in this paper we focus on the implementation of an effective power noise generator to hinder SPA attacks on microcontrollers implemented on FPGAs.

### *2.3. Power Watermarking*

Power watermarking [38] enables the use of the power consumption side channel to extract license or intellectual property information from a digital circuit. This information is usually a watermark or a digital signature [59] that unequivocally identifies the owner of the design under protection. The main issue in these protection systems is how to generate the variations on power consumption to enable easy extraction of the watermark from the power signal. In [37], it is shown that HCMs generate suitable signals for this purpose. In principle, HCMs do not significantly increase the power consumption of the system under protection because they are activated only when extraction of the watermark is required. To apply these modules to protect a processing unit against SPA attacks, they have to be modified to generate peaks of power consumption with an amplitude similar to that generated by the execution of a processor instruction, thus generating reasonable power consumption overhead.

## **3. Power Noise Generation**

Power noise generation [19,32] is one of the main countermeasures used against SPA, DPA and CPA attacks. It can be used standalone or in combination with other countermeasures as arithmetic or boolean masking as it acts directly on the power side channel. The idea is to generate power consumption peaks similar to those generated by the computation system or the cryptoprocessor under protection. In order to properly hinder information contained in the power traces, these generated peaks should meet two requirements:

- The amplitude of the peaks should be similar to that generated by the operations of interest.
- The times when the peaks are generated should be random.

With these two requirements, generated peaks should be indistinguishable from those generated by the circuits under protection. In principle, to build a Power Noise Generator (PNG) with these properties, two main elements are required: a Power Consumption Generator (PCG) and a True Random Number Generator (TRNG). These elements are studied in the following sub-sections.

### *3.1. Power Consumption Generators in FPGAs*

To generate controlled power peaks, specific circuits producing instant power consumption are required. In general, the power consumption of a circuit implemented on an FPGA can be expressed as [60]:

$$P = VDD^2 \cdot f \cdot \sum_i \left( C_{eff}(i) \cdot SW(i) \right) \tag{1}$$

where $C_{eff}(i)$ is the effective capacitance, and $SW(i)$ is the switching activity, both of element $i$. Note that in (1), it is assumed that all elements in the circuit are powered at the same voltage, $VDD$, and are operating at the same frequency, $f$. In the case of FPGAs, $C_{eff}(i)$ can be considered equal if all elements under consideration are logic gates, as they are implemented in LUTs. In the case of including flip-flops, $C_{eff}$ will have a different value. From this equation, if we want to generate high power consumption with a small number of LUTs, high values for $f$ should be achieved. This is the idea behind using ring oscillators (ROs) as basic PCG circuits [33]. Indeed, a RO is the most simple structure for generating significant power consumption with the low area requirements in an FPGA, as shown in Figure 1. This scheme corresponds to a Simple Ring Oscillator with one inverting element, $SRO(1)$, which can be enabled or disabled by means of the *enable* input and the $AND$ gate.

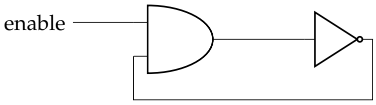
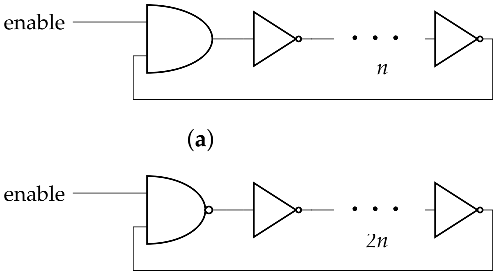


**Figure 1.** $SRO$ with 1 inverting element.

In the case of an SRO with $n$ elements, it is necessary to distinguish two cases depending on $n$ being odd or even, as shown in Figure 2.



(**a**)

(**b**)

**Figure 2.** $SRO(n)$ with odd (**a**) and even (**b**) numbers of inverting elements.

From this figure, the increase in dynamic power consumption generated by $SRO(n)$ can be estimated as [60]:

$$P = {V_{DD}}^2 \cdot f_{ro} \cdot \Big[(C_{AND} \cdot SW_{AND}) \\ +(n) \cdot (C_{NOT} \cdot SW_{NOT})\Big] \quad (2)$$

where $C_{AND}$ and $C_{NOT}$ are the effective capacitance, and $SW_{AND}$ and $SW_{NOT}$ are the switching activity of the $AND$ and $NOT$ gates, respectively. Note that in the case of $n$ being an even number, there is a $NAND$ gate ($AND$ gate with inverted output), but it can be considered that in an FPGA its effective capacitance is the same as that of an $AND$ gate.

In principle, power consumption generated by an SRO may be increased by adding more inverters. Indeed, according to Equation (1), there will be more elements in the sum, and power consumption should increase. However, there is a side effect: the delay of the signal from the beginning to the end of the ring is increased because there are more elements, and consequently, $f_{ro}$ decreases. Table 1 shows experimental results corresponding to $SRO(n)$ for different values of $n$ in a CMOD-A7 Digilent board including an Artix-7 XC7A35T-1CPG236C device from Xilinx/AMD. The designs have been implemented using the Vivado 2020.2 design suite. The board is powered at 5 V and includes voltage regulators to generate the different voltage levels required by the FPGA. This device has been selected because it is a low-cost, small-size FPGA suitable for IoT designs, and it is included on a board with a reduced number of peripherals that can easily be powered externally.

**Table 1.** Simple Ring Oscillators power consumption results.

| $n$ | LUTs | $\Delta I$ | $\Delta W$ | $\Delta W$/LUT |
|---|---|---|---|---|
| 1 | 2 | 0.4 mA | 2 mW | 1.00 mW |
| 2 | 3 | 0.4 mA | 2 mW | 0.66 mW |
| 5 | 5 | 0.4 mA | 2 mW | 0.40 mW |
| 9 | 9 | 0.4 mA | 2 mW | 0.22 mW |
| 63 | 48 | 0.7 mA | 3.5 mW | 0.07 mW |

In this table, column $\Delta I$ presents the increment of current when enabling the corresponding SRO, and $\Delta W$ is the corresponding increment in power. Measurements were obtained using a Keysight N6705C DC Power analyzer. In [61], a detailed study of ring oscillator implementations in Ultrascale devices from Xilinx is presented, with power consumption densities from 1.7 mW/LUT to 2.2 mW/LUT, which are consistent with results in Table 1, taking into account the technology step between Artix-7 and UltraScale devices.

From Table 1, there is no significant increase in power when adding a reduced number of inverting elements. In order to avoid the effect of $f_{ro}$ decreasing when $n$ increases, one solution is to arrange $n$ $SRO(1)$ elements in parallel rather than using the $SRO(n)$. In that case, the results in Table 2 were obtained, where $m$ is the number of parallel $SRO(1)$.

**Table 2.** Several $SRO(1)$ in parallel, power consumption results.

| $m$ | LUTs | $\Delta I$ | $\Delta W$ | $\Delta W$/LUT |
|---|---|---|---|---|
| 1 | 2 | 0.4 mA | 2.0 mW | 1.00 mW |
| 2 | 4 | 0.5 mA | 2.5 mW | 0.63 mW |
| 4 | 8 | 0.7 mA | 3.5 mW | 0.44 mW |
| 8 | 16 | 1.0 mA | 5.0 mW | 0.31 mW |

These results can be improved with the High Consuming Module (HCM) proposed in [37] and displayed in Figure 3. In this case, eight $SRO(1)$ are operating in parallel, and the corresponding eight outputs are combined by means of $AND$ and $OR$ gates to generate additional switching activity without increasing the number of LUTs. Indeed, one HCM generates $\Delta I$ = 1.2 mA ($\Delta W$ = 6 mW) requiring only 8 LUTs, while two HCMs operating in parallel generate $\Delta I$ = 2.1 mA ($\Delta W$ = 10.5 mW) with 16 LUTs in an Artix-7 XC7A35T-1CPG236C device. It must also be noted that these modules do not affect the maximum operating frequency of the overall system, since they do not interact with the system clock, as shown in Figure 3.

### 3.2. Xored High Consuming Modules (XHCMs)

The HCM presented in the previous section can be improved by replacing the $AND/OR$ gates in Figure 3 with $XOR$ gates, obtaining the so-called Xored High Consuming Module (XHCM). Introduction of $XOR$ gates generates some more switching activity, since in the last level, the $AND$ gate, with 25% switching probability, is replaced by an $XOR$ function with 50% switching probability (in the rest of the levels, $AND$ and $OR$ gates are alternated and, on average, should have the same switching activity). At the same time, this requires the same number of LUTs when implemented on FPGAs. Experimental results show that one HCM generates 6.0 mW of power overhead with one instance and 10.5 mW when using two independent instances, while a single XHCM generates 6.5 mW, and two generate 11.5 mW, thus providing better results than HCMs. Note that the switching activity can be increased if the instances are gated with $OR/AND$ functions in HCMs and $XOR$ gates in XHCMs instead of being independent. Table 3 shows complete experimental results and comparisons of HCMs with XHCMs in gated and non-gated versions, where $m$ is the number of HCM/XHCM instances. When comparing independent instances of HCMs and XHCMs, the latter provide a 7% improvement in power generated per LUT. In the gated versions, this improvement rises up to 10–15%.

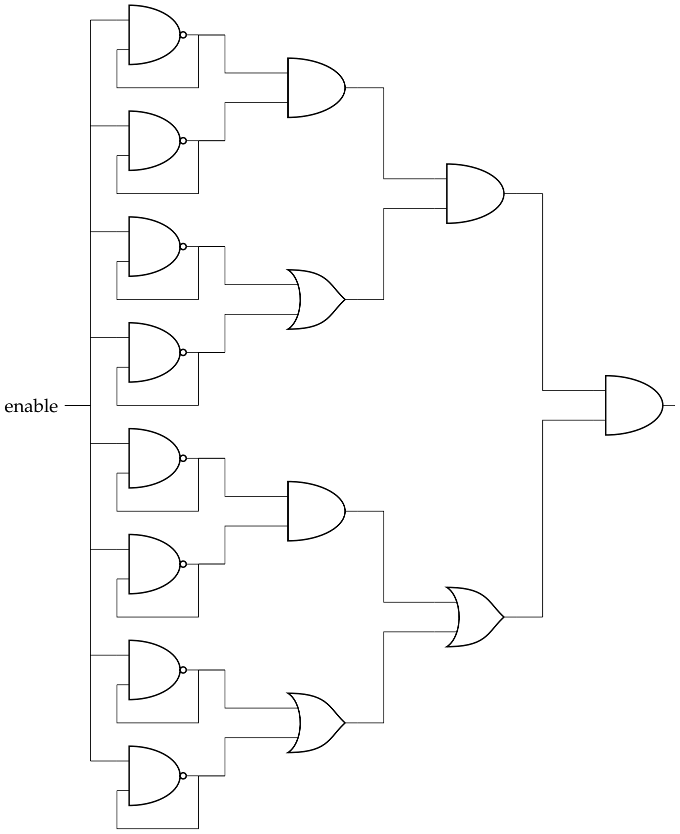
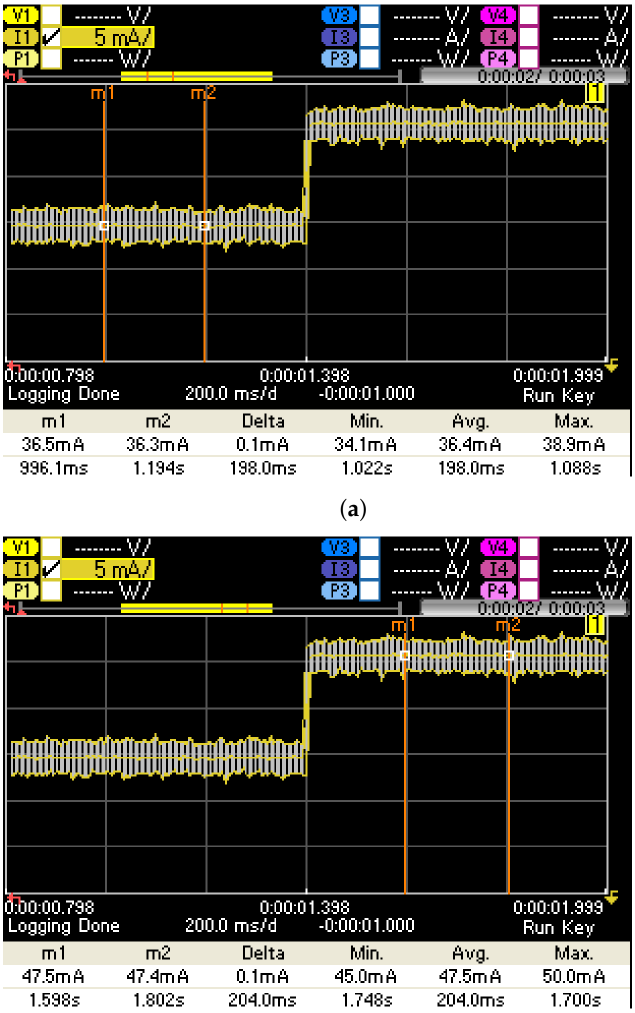


**Figure 3.** HCM with eight Ring Oscillators.

**Table 3.** Comparison of HCM and XHCM.

| Type of Module | $m$ | LUTs | $\Delta I$ | $\Delta W$ | $\Delta W$/LUT |
|---|---|---|---|---|---|
| HCM | 1 | 8 | 1.2 mA | 6.0 mW | 0.75 mW |
| HCM | 2 | 16 | 2.1 mA | 10.5 mW | 0.66 mW |
| HCM | 4 | 28 | 3.9 mA | 19.0 mW | 0.68 mW |
| HCM | 8 | 68 | 9.3 mA | 46.5 mW | 0.68 mW |
| XHCM | 1 | 8 | 1.3 mA | 6.5 mW | 0.81 mW |
| XHCM | 2 | 16 | 2.3 mA | 11.5 mW | 0.72 mW |
| XHCM | 4 | 28 | 4.2 mA | 21.0 mW | 0.75 mW |
| XHCM | 8 | 64 | 9.3 mA | 46.5 mW | 0.73 mW |
| HCM-gated | 2 | 16 | 2.1 mA | 10.5 mW | 0.66 mW |
| HCM-gated | 4 | 29 | 4.2 mA | 21.0 mW | 0.72 mW |
| HCM-gated | 8 | 66 | 9.3 mA | 46.5 mW | 0.70 mW |
| XHCM-gated | 2 | 16 | 2.4 mA | 12.0 mW | 0.75 mW |
| XHCM-gated | 4 | 37 | 5.8 mA | 29.0 mW | 0.78 mW |
| XHCM-gated | 8 | 67 | 11.1 mA | 55.5 mW | 0.83 mW |

From the results in Table 3, gated XHCMs with $m$ = 1 and $m$ = 2 are suitable for generating power noise intended to mask power traces of processing systems or cryptographic processors, while those with $m$ = 4 or $m$ = 8 can be useful for generating signals to be transmitted through the power side channel due to the high current increase, which makes it easier to recover the signal, as illustrated in Figure 4. A complete study regarding the generation and recovery of power signals using HCMs for power watermarking (which is immediately applicable to XHCMs) can be found in [37]. In both cases, it would be desirable to have the possibility of not only controlling the time during which the additional power consumption is performed, but also to control the amplitude of the generated consumption peaks. This amplitude control, which is required to generate power noise against SCAs, is approached in the next section. In the following, we use the term XHCM for gated versions, as they are advantageous compared to independent XCHM instances.



0:00:00.798 0:00:01.398 0:00:01.999
Logging Done 200.0 ms/d -0:00:01.000 Run Key

| m1 | m2 | Delta | Min. | Avg. | Max. |
|---|---|---|---|---|---|
| 36.5mA | 36.3mA | 0.1mA | 34.1mA | 36.4mA | 38.9mA |
| 996.1ms | 1.194s | 198.0ms | 1.022s | 198.0ms | 1.088s |

(**a**)

0:00:00.798 0:00:01.398 0:00:01.999
Logging Done 200.0 ms/d -0:00:01.000 Run Key

| m1 | m2 | Delta | Min. | Avg. | Max. |
|---|---|---|---|---|---|
| 47.5mA | 47.4mA | 0.1mA | 45.0mA | 47.5mA | 50.0mA |
| 1.598s | 1.802s | 204.0ms | 1.748s | 204.0ms | 1.700s |

(**b**)

**Figure 4.** Increment of current generated by eight-gated XHCMs on Artix-7 devices. (**a**) Average current with *XHCMs* disabled (*enable* = '0'). (**b**) Average current when 8 XHCMs are enabled (*enable* = '1').

## 4. Controlled XHCMs

A first approach for controlling the amplitude of the power signal generated by XHCMs can be the use of a Pulse-Width Modulation (PWM) module connected to the *enable* signal. This allows generation of different waveforms that can extend the type of signals transmitted by means of XHCMs through the side channel for extracting information. This is covered in the next subsection.

### *4.1. PWM-XHCM*

As commented above, PWM can be used to generate different output levels in an XHCM. Indeed, as the typical delay of an LUT in an Artix-7 device is 0.13 ns, the oscillation frequency of the feedback elements can be estimated to be in the range of GHz. Therefore, if a 100-MHz PWM is introduced, it can be considered that XHCM power generation has continuous behavior, and it can be modulated by PWM. Additionally, it is interesting for the mentioned application to generate a high number of different output levels. Using

10 bits for the period, it is possible to generate 1024 different power levels. On the other hand, this large number of power levels enables the generation of different functions for extracting information from the inside of the chip. Figure 5 shows the block diagram of the proposed PWM-XHCM system, with 10-bit resolution for the PWM period and 10 bits for the duty cycle. It requires 80 LUTs when implemented on an Artix-7 device. As an example, Figure 6 shows the increment of current drain generated by a 40% duty-cycle PWM with a 100-MHz clock and a period of 2.5 µs. In this figure, the white line corresponds to the variations generated by thermal noise, and the solid yellow line is the average of the current. Table 4 shows results for different duty cycles.

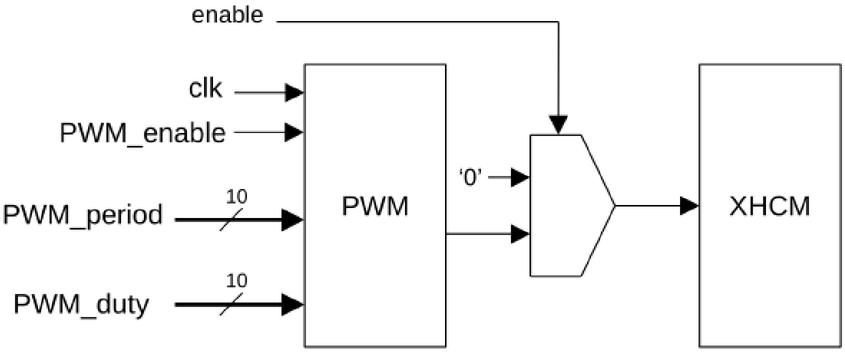
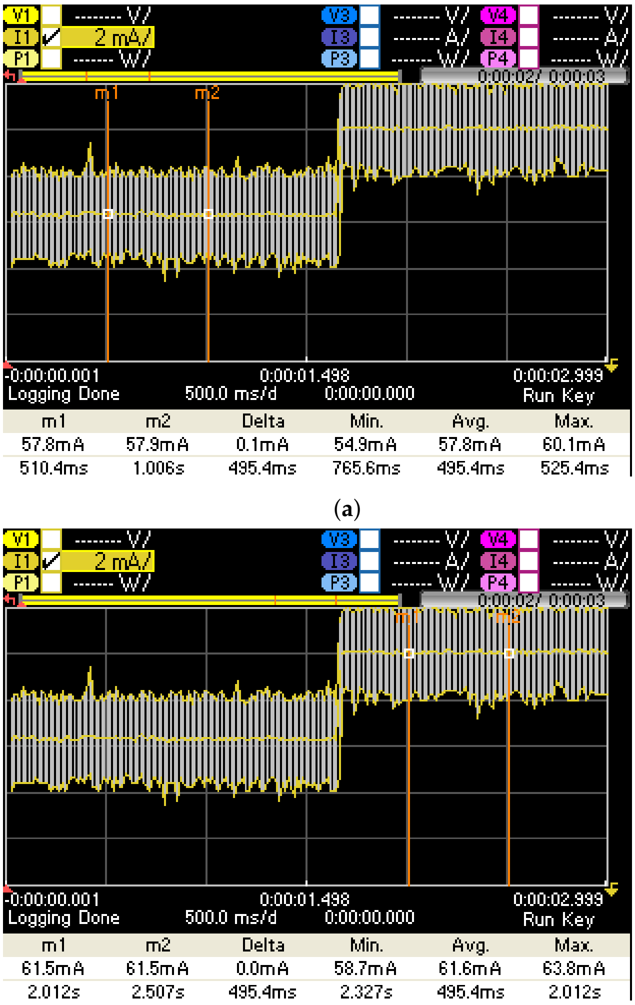


**Figure 5.** Block diagram of the PWM-XHCM-controlled power consumption generator.

**Table 4.** Power consumption levels generated by PWM on $XHCM(8)$ for different duty cycles.

| Duty Cycle | $\Delta I$ | $\Delta W$ |
|---|---|---|
| 20% | 1.9 mA | 9.0 mW |
| 40% | 3.8 mA | 19.0 mW |
| 60% | 5.8 mA | 29.0 mW |
| 80% | 8.6 mA | 43.0 mW |

Since *PWM_duty* can be modified on-the-fly, it is possible to generate periodic functions, such as $sin(\omega t + \phi)$, and to change amplitude, frequency or phase of those functions, thus enabling generation of amplitude-, frequency- or phase-modulated signals to be transmitted from the inside of the chip. This adds new features and flexibility for power watermarking applications [37,38]. As a drawback, the introduction of PWM increases the general power consumption of the entire system, as can be observed in Figure 4 and Figure 6. Indeed, the minimal current has an average of 36.5 mA without PWM, and around 58.0 mA when introducing PWM. This fact, along with the correlation of the signal with the clock feeding the PWM block, makes it not a recommended solution to mask power consumption peaks. Moreover, this structure does not generate instantaneous power consumption peaks, since it generates an average of power consumption over a complete PWM period.



-0:00:00.001 0:00:01.498 0:00:02.999
Logging Done 500.0 ms/d 0:00:00.000 Run Key

| m1 | m2 | Delta | Min. | Avg. | Max. |
|---|---|---|---|---|---|
| 57.8mA | 57.9mA | 0.1mA | 54.9mA | 57.8mA | 60.1mA |
| 510.4ms | 1.006s | 495.4ms | 765.6ms | 495.4ms | 525.4ms |

(**a**)

-0:00:00.001 0:00:01.498 0:00:02.999
Logging Done 500.0 ms/d 0:00:00.000 Run Key

| m1 | m2 | Delta | Min. | Avg. | Max. |
|---|---|---|---|---|---|
| 61.5mA | 61.5mA | 0.0mA | 58.7mA | 61.6mA | 63.8mA |
| 2.012s | 2.507s | 495.4ms | 2.327s | 495.4ms | 2.012s |

(**b**)

**Figure 6.** Increment of current generated by eight-gated $XHCMs$ on Artix-7 devices. (**a**) Average current with 40% duty cycle PWM enabled and $XHCM(8)$ disabled (*enable* = '0'). (**b**) Average current with 40% duty cycle PWM enabled and $XHCM(8)$ enabled (*enable* = '1').

### *4.2. Multi-Level XHCMs*

In order to overcome the drawbacks of a PWM-controlled XHCM, a modification of XHCM allowing individual enabling or disabling every oscillator is proposed. Figure 7 shows the corresponding Multi-Level XHCM (ML-XHCM), which can generate eight different levels of power consumption. Comparing the circuit in Figure 7 to the one presented in Figure 3, two main differences can be observed: First, the $AND/OR$ gates have been replaced by $XOR$ gates in order to take advantage of the 50% switching probability of these gates, as detailed in Section 3.2. Second, the *enable* input, which simultaneously enables or disables the eight ring oscillators in Figure 3, has been replaced in Figure 7 by eight *enable*($i$) inputs that can individually enable or disable each ring oscillator, thus providing different levels of power consumption depending on the number of inputs enabled at a given time. In the case of $m$ = 1, it implies a resolution of $\delta I$ = 0.16 mA, thus generating peaks of power consumption from 0.16 mA to 1.3 mA. If these amplitudes are randomly generated, these peaks can be used to mask power consumption variations generated by different processing operations in computing systems or crypto-processors.

Figure 8 shows the block diagram of the controlled ML-XHCM, where the global *pow_enable* signal enables the power consumption unit (*pow_enable* = '0' resets the latch; *pow_enable* = '1' enables it), *pow_level* allows introduction of the desired power level, and *load* loads the value from the decoder into the latch. The decoder enables as many ROs as indicated by the *pow_level* input. It requires 17 LUTs when implemented on Artix-7 devices.

In the case of $m = 2$, 16 different amplitude levels with a resolution of $\delta I = 0.13$ mA can be generated, requiring 27 LUTs on Artix-7 devices. Note that through the randomization of the values passed to the *pow_level* input of the controlled ML-XHCM and the times when *pow_enable* is enabled, our proposal allows the introduction of random values in both amplitude and time domains, thus making it suitable for generating power noise to mask power traces in SPA attacks.

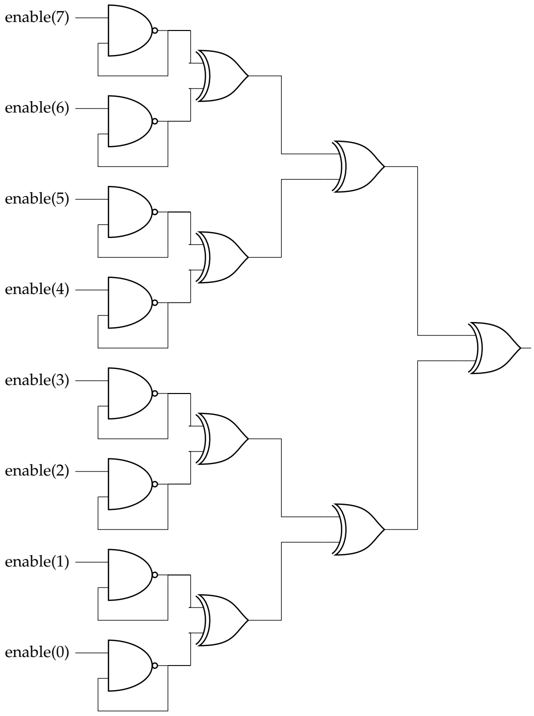


**Figure 7.** ML-XHCM with eight power consumption levels.

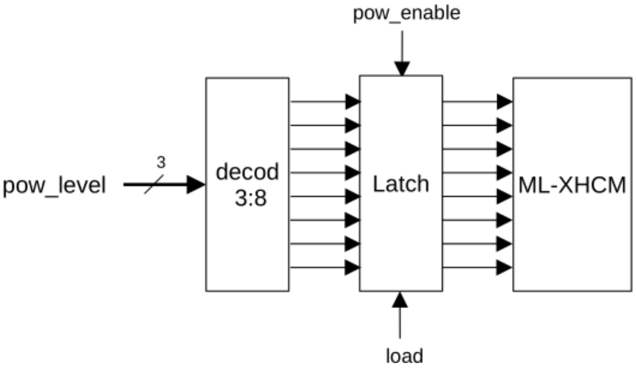


**Figure 8.** Block diagram of the controlled ML-XHCM power consumption generator.

### *4.3. Random Number Generation*

In order to generate random values for the randomization of the amplitude and the enable time of ML-XHCM modules, we propose the use of a TRNG based on ROs and specifically designed for FPGAs [62]. The TRNG has been implemented using 50 ROs and a register to stabilize the values generated by the ROs. Finally, the content of the 50 bits of the register are *XOR*ed to obtain one random bit. Figure 9 shows the scheme of the TRNG, which uses *clk_trng* for syncronization, the *enable_trng* input for enabling or disabling the random generator, and the *rnd_bit* output is the resulting random bit. This TRNG requires only 51 six-input LUTS on an Artix-7 device.

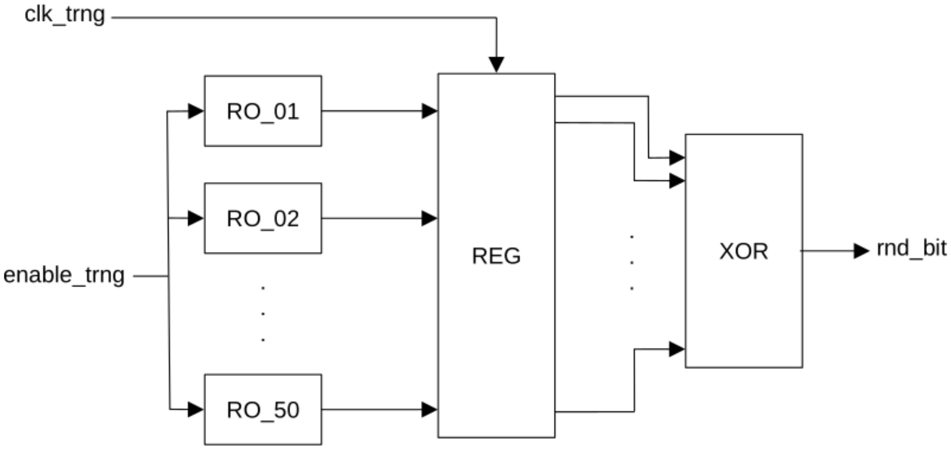


**Figure 9.** Block diagram of TRNG.

### 4.4. Randomized ML-XHCM

In this section, we add two uncorrelated TRNGs to the ML-XHCM, thus enabling the generation of power consumption peaks of random amplitudes at random times. Indeed, if a TRNG feeds the shift register and the output of this register is connected to the ML-XHCM described in Figure 7, different amplitudes may be randomly generated. This is shown in Figure 10, where TRNG_A and SR_A are the TRNG and the shift register, respectively, for randomizing amplitudes. A similar structure can be used to randomize time, but in this case, the consumption peaks will be activated 50% of the time. In order to make the percentage of activation time configurable, a decoder can be added, as shown in Figure 10, thus being active $1/k$ of the time, where $k$ is the number of outputs of the decoder. As will be shown in the next section, the addition of the decoder and shift registers has no significant effect on the required area resources.

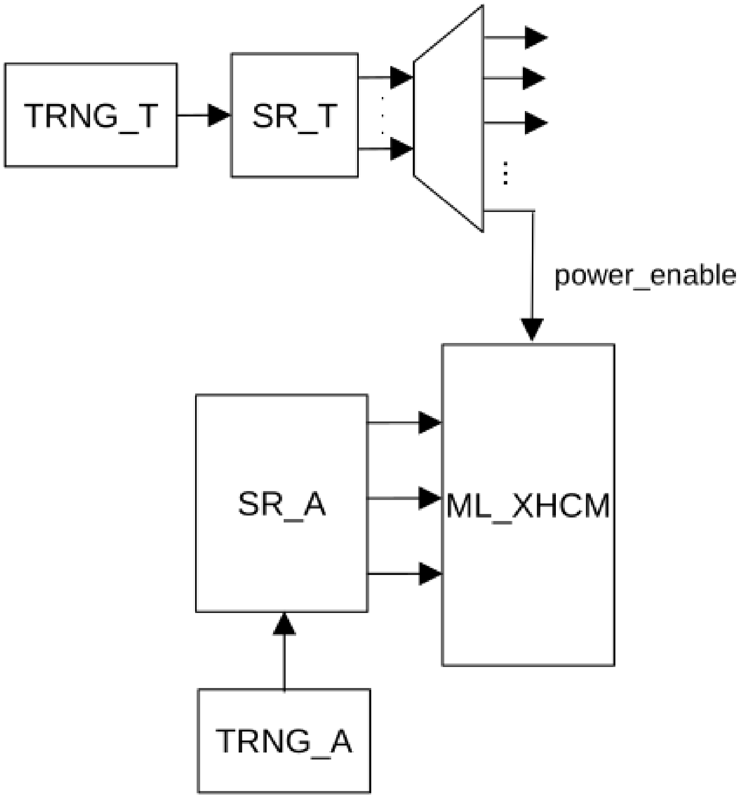


**Figure 10.** Block diagram of randomized ML_XHCM.

## 5. Experimental Results

Table 5 summarizes the experimental results obtained from the different implementations proposed in this paper on an Artix-7 device. In this table, $n_{ro}$ stands for the number of ROs included in the module, *Amp. levels* is the number of different levels of amplitude that the modules can generate, and $\delta I$ is the resolution in the current increase. From this table, XCHM modules allow the generation of high current (or equivalently, high power) increases, thus being useful for power watermarking applications in which easily recoverable square signals are preferred (see [37]). If other types of signals, such as sinusoidals, are required, PWM-XHCM provides 1024 different levels to build any type of waveform at the cost of a few more LUTs. On the other hand, to generate power noise against SPA attacks on processing systems, ML-XHCM and RML-XHCM provide low-area overhead solutions for the protection of processing units used in IoT devices and implemented on low-cost FPGAs. Indeed, ML-XHCM allows the generation of power consumption peaks of different amplitude levels at desired times. With only 17 LUTs, it is possible to have 8 different amplitude levels with a resolution of 0.16 mA. In the case of requiring 16 amplitude levels, ML_XHCM requires 27 LUTs on an Artix-7 device, and 49 LUTs for 32 amplitude levels. In the case of introducing randomization of amplitude levels and times of activation, RML-XHCM provides eight different random levels of amplitude with 0.16 mA of resolution while activating the power peaks at random times in 12.5% of the defined clock period at a cost of 118 LUTs. In the case of generating 16 random levels of amplitude, 129 LUTs are required, and 138 LUTs for 32 levels.

**Table 5.** Implementation results for RML_XHCM.

| Type of Module | $n_{ro}$ | Amp. Levels | %act.time | LUTs | $\delta I$ | $(\Delta W \cdot n_{levels}/LUTs)$ | Application |
|---|---|---|---|---|---|---|---|
| XHCM | 8 | 1 | 100 | 8 | 1.3 mA | 0.81 mW/LUT | power watermarking |
| XHCM | 16 | 1 | 100 | 16 | 2.4 mA | 0.75 mW/LUT | power watermarking |
| XHCM | 32 | 1 | 100 | 37 | 5.8 mA | 0.78 mW/LUT | power watermarking |
| XHCM | 64 | 1 | 100 | 67 | 11.1 mA | 0.83 mW/LUT | power watermarking |
| PWM-XHCM | 32 | 1024 | 100 | 59 | 0.004 mA | 0.35 mW/LUT | power watermarking |
| PWM-XHCM | 64 | 1024 | 100 | 80 | 0.009 mA | 0.58 mW/LUT | power watermarking |
| ML-XHCM | 8 | 8 | 100 | 17 | 0.16 mA | 0.38 mW/LUT | power masking |
| ML-XHCM | 16 | 16 | 100 | 27 | 0.13 mA | 0.39 mW/LUT | power masking |
| ML-XHCM | 32 | 32 | 100 | 49 | 0.13 mA | 0.42 mW/LUT | power masking |
| RML-XHCM | 8 | 8 | 50 | 113 | 0.16 mA | 0.057 mW/LUT | power masking |
| RML-XHCM | 8 | 8 | 25 | 118 | 0.16 mA | 0.054 mW/LUT | power masking |
| RML-XHCM | 8 | 8 | 12.5 | 118 | 0.16 mA | 0.054 mW/LUT | power masking |
| RML-XHCM | 16 | 16 | 50 | 127 | 0.13 mA | 0.082 mW/LUT | power masking |
| RML-XHCM | 16 | 16 | 25 | 129 | 0.13 mA | 0.081 mW/LUT | power masking |
| RML-XHCM | 16 | 16 | 12.5 | 129 | 0.13 mA | 0.081 mW/LUT | power masking |
| RML-XHCM | 32 | 32 | 50 | 147 | 0.13 mA | 0.14 mW/LUT | power masking |
| RML-XHCM | 32 | 32 | 25 | 138 | 0.13 mA | 0.15 mW/LUT | power masking |
| RML-XHCM | 32 | 32 | 12 | 138 | 0.13 mA | 0.15 mW/LUT | power masking |

Table 6 shows the comparison of our proposals for building countermeasures against SPA attacks to other methods in the literature. In [19], a power consumption generator based on the activation of 20-RO sets is proposed, with a total of 32 amplitude levels. Although this work does not provide results of area requirements in terms of LUTs, the need of 620 ROs implies more area resources than ML-XHCM or RML-XHCM. Amplitude levels are controlled by an activity sensor, also based on ROs, to try to follow the levels of activity in an AES implementation. Randomization in time is achieved using an LFSR, which provides worse random properties than a TRNG. Regarding the works in [24,54], they are oriented to the protection of AES implementations, and the control is based on specific operations of this encryption algorithm. Therefore, our proposals present very low-area overhead while providing complete flexibility for applications not only for the protection of cryptographic algorithms, but also for the protection of processing units in low-cost FPGAs, thus improving the features provided by other works in the literature.

**Table 6.** Comparison to other power noise generators on FPGAs.

| Module | $n_{ro}$ | Amp. Levels | %act.time | LUTs | $(\Delta W \cdot n_{levels}/LUTs)$ | Amplitude Control | Time Control |
|---|---|---|---|---|---|---|---|
| [19] | 620 | 32 | 100 | - | - | Controlled by sensor | Random based on LFSR |
| [24] | - | - | 100 | 104 | - | Fixed by algorithm | Fixed by algorithm |
| [54] | - | - | 100 | 456 | - | Fixed by algorithm | Fixed by algorithm |
| ML-XHCM16 (this work) | 16 | 16 | 100 | 27 | 0.39 mW/LUT | Customizable | Customizable |
| ML-XHCM32 (this work) | 32 | 32 | 100 | 49 | 0.42 mW/LUT | Customizable | Customizable |
| RML-XHCM16 (this work) | 16 | 16 | 12.5 | 129 | 0.081 mW/LUT | TRNG | TRNG |
| RML-XHCM32 (this work) | 32 | 32 | 12.5 | 138 | 0.15 mW/LUT | TRNG | TRNG |

## 6. Conclusions and Future Work

Power consumption noise generation is an effective countermeasure against SPA attacks against cryptographic circuits and processing units. In this context, power consumption generators are essential modules to build security applications such as power noise generation for masking power traces in side channel attacks against computing systems or cryptographic processors. Moreover, these modules can also be applied for power watermarking for protecting intellectual property of IP cores. In this paper, a modification of the HCM from [37], named XHCM, has been proposed. This module improves the amount of current consumption per LUT on FPGAs when compared to other proposals in the literature. Indeed, experimental results show it is possible to achieve current increments in the range from 2.4 mA, with only 16 LUTs when using a single XHCM on Artix-7 devices with a power consumption density of 0.75mW/LUT, to 11.1 mA with 67 LUTs when using 8 XHCMs, with a power consumption density of 0.83 mW/LUT. Additionally, two controlled versions have been developed. The first one, PWM-XHCM, can be controlled by PWM and allows the implementation of different types of waveforms and modulation procedures. The characteristics of XHCM and PWM-XHCM are specially suitable for power watermarking applications. On the other hand, the multi-level controlled ML-XHCM is able to provide different power consumption levels with minimal area overhead, thus enabling the generation of randomized power noise in the amplitude and time domains. This design is suitable for hiding processing information in power traces as countermeasure to SPA and DPA attacks. As a proof-of-concept of the possibilities for randomizing the time and amplitude domains of ML-XHCM, the so-called RML-XHCM has been developed, including two de-correlated TRNGs to generate random amplitude power consumption peaks at random times. All these features can be implemented requiring less than 150 LUTs in a low-cost Artix-7 device from Xilinx/AMD. Therefore, the main contributions of this article can be summarized as:

- An improved high-consuming power generator, XHCM, and a PWM-controlled variant, PWM-XHCM, have been developed, thus providing an efficient solution to extract digital signatures through the power side channel.
- A modification of XCHM enabling the control of both amplitude and time domains, named ML-XHCM, has been proposed, providing a powerful tool for the protection of processing units against SPA attacks in IoT devices implemented on FPGAs. Moreover, a time-randomized version, called RML-XHCM, has also been developed as a proof-of-concept.

As future work, we plan to build a secure IoT device with power noise masking for the processing unit based on the open-hardware platform for IoT devices on FPGAs presented in [63].

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

**Funding:** This work was partially supported by the Consejería de Economía y Conocimiento de la Junta de Andalucía (Spain) and by the European Regional Development Funds (ERDF) under Project B-TIC-588-UGR20.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet Things J.* **2019**, *6*, 8182–8201. [CrossRef]
2. Park, C.S.; Nam, H.M. Security architecture and protocols for secure MQTT-SN. *IEEE Access* **2020**, *8*, 226422–226436. [CrossRef]
3. Kim, Y.; Seo, S.C. Efficient Implementation of AES and CTR_DRBG on 8-bit AVR-based Sensor Nodes. *IEEE Access* **2021**, *9*, 30496–30510. [CrossRef]
4. Parrilla, L.; Castillo, E.; López-Ramos, J.A.; Álvarez-Bermejo, J.A.; García, A.; Morales, D.P. Unified compact ECC-AES co-processor with group-key support for IoT devices in wireless sensor networks. *Sensors* **2018**, *18*, 251. [CrossRef]
5. Devi, M.; Majumder, A. Side-channel attack in Internet of Things: A survey. In *Applications of Internet of Things*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 213–222.
6. Park, J.; Tyagi, A. Using Power Clues to Hack IoT Devices: The power side channel provides for instruction-level disassembly. *IEEE Consum. Electron. Mag.* **2017**, *6*, 92–102. [CrossRef]
7. Wei, L.; Luo, B.; Li, Y.; Liu, Y.; Xu, Q. I know what you see: Power side-channel attack on convolutional neural network accelerators. In Proceedings of the 34th Annual Computer Security Applications Conference, San Juan, PR, USA, 3–7 December 2018; pp. 393–406.
8. Naghibijouybari, H.; Neupane, A.; Qian, Z.; Abu-Ghazaleh, N. Rendered insecure: Gpu side channel attacks are practical. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 2139–2153.
9. Maji, S.; Banerjee, U.; Chandrakasan, A.P. Leaky nets: Recovering embedded neural network models and inputs through simple power and timing side-channels—Attacks and defenses. *IEEE Internet Things J.* **2021**, *8*, 12079–12092. [CrossRef]
10. Mangard, S. A simple power-analysis (SPA) attack on implementations of the AES key expansion. In *International Conference on Information Security and Cryptology*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 343–358.
11. Kadir, S.A.; Sasongko, A.; Zulkifli, M. Simple power analysis attack against elliptic curve cryptography processor on FPGA implementation. In Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, 17–19 July 2011; pp. 1–4.
12. Ghandali, S.; Ghandali, S.; Tehranipoor, S. Deep K-TSVM: A Novel Profiled Power Side-Channel Attack on AES-128. *IEEE Access* **2021**, *9*, 136448–136458. [CrossRef]
13. Lyu, Y.; Mishra, P. A survey of side-channel attacks on caches and countermeasures. *J. Hardw. Syst. Secur.* **2018**, *2*, 33–50. [CrossRef]
14. Oswald, E. Enhancing simple power-analysis attacks on elliptic curve cryptosystems. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Redwood Shores, CA, USA, 13–15 August 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 82–97.
15. Akkar, M.L.; Giraud, C. An implementation of DES and AES, secure against some attacks. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Paris, France, 14–16 May 2001; Springer: Berlin/Heidelberg, Germany, 2001; pp. 309–318.
16. Bayrak, A.G.; Velickovic, N.; Ienne, P.; Burleson, W. An architecture-independent instruction shuffler to protect against side-channel attacks. *ACM Trans. Archit. Code Optim. (TACO)* **2012**, *8*, 1–19. [CrossRef]
17. Herbst, C.; Oswald, E.; Mangard, S. An AES smart card implementation resistant to power analysis attacks. In Proceedings of the International Conference on Applied Cryptography and Network Security, Singapore, 6–9 June 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 239–252.

18. Duan, S.; Wang, W.; Luo, Y.; Xu, X. A survey of recent attacks and mitigation on FPGA systems. In Proceedings of the 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Tampa, FL, USA, 7–9 July 2021; pp. 284–289.
19. Krautter, J.; Gnad, D.R.; Schellenberg, F.; Moradi, A.; Tahoori, M.B. Active fences against voltage-based side channels in multi-tenant FPGAs. In Proceedings of the 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Westminster, CO, USA, 4–7 November 2019; pp. 1–8.
20. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397.
21. Wang, H.; Dubrova, E. Tandem deep learning side-channel attack against FPGA implementation of AES. In Proceedings of the 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), Chennai, India, 14–16 December 2020; pp. 147–150.
22. De La Piedra, A.; Braeken, A.; Touhafi, A. Sensor systems based on FPGAs and their applications: A survey. *Sensors* **2012**, *12*, 12235–12264. [CrossRef]
23. Lee, J.; Han, D.G. Security analysis on dummy based side-channel countermeasures—Case study: AES with dummy and shuffling. *Appl. Soft Comput.* **2020**, *93*, 106352. [CrossRef]
24. Kamoun, N.; Bossuet, L.; Ghazel, A. Correlated power noise generator as a low cost DPA countermeasures to secure hardware AES cipher. In Proceedings of the 2009 3rd International Conference on Signals, Circuits and Systems (SCS), Medenine, Tunisia, 6–8 November 2009; pp. 1–6.
25. Brier, E.; Clavier, C.; Olivier, F. Correlation power analysis with a leakage model. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Cambridge, MA, USA, 11–13 August 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 16–29.
26. Alioto, M.; Poli, M.; Rocchi, S. Power analysis attacks to cryptographic circuits: A comparative analysis of DPA and CPA. In Proceedings of the 2008 International Conference on Microelectronics, Medenine, Tunisia, 6–8 November 2008; pp. 333–336.
27. Fei, Y.; Ding, A.A.; Lao, J.; Zhang, L. A statistics-based success rate model for DPA and CPA. *J. Cryptogr. Eng.* **2015**, *5*, 227–243. [CrossRef]
28. van Geest, J.; Buhan, I. A Side-Channel Based Disassembler for the ARM-Cortex M0. Cryptology ePrint Archive. 2022. Available online: https://eprint.iacr.org/2022/523 (accessed on 23 May 2022).
29. Banciu, V.; Oswald, E. Pragmatism vs. elegance: Comparing two approaches to simple power attacks on AES. In Proceedings of the International Workshop on Constructive Side-Channel Analysis and Secure Design, Leuven, Belgium, 11–12 April 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 29–40.
30. Bertoni, G.; Zaccaria, V.; Breveglieri, L.; Monchiero, M.; Palermo, G. AES power attack based on induced cache miss and countermeasure. In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II, Las Vegas, NV, 4–6 April 2005; Volume 1, pp. 586–591.
31. Regazzoni, F.; Wang, Y.; Standaert, F.X. FPGA implementations of the AES masked against power analysis attacks. *Proc. COSADE* **2011**, *2011*, 56–66.
32. Wang, X.; Yueh, W.; Roy, D.B.; Narasimhan, S.; Zheng, Y.; Mukhopadhyay, S.; Mukhopadhyay, D.; Bhunia, S. Role of power grid in side channel attack and power-grid-aware secure design. In Proceedings of the 50th Annual Design Automation Conference, Austin, TX, USA, 29 May–7 June 2013; pp. 1–9.
33. Liu, P.C.; Chang, H.C.; Lee, C.Y. A low overhead DPA countermeasure circuit based on ring oscillators. *IEEE Trans. Circuits Syst. II Express Briefs* **2010**, *57*, 546–550.
34. Fu, H.P.; Hsiao, J.H.; Liu, P.C.; Chang, H.C.; Lee, C.Y. A low cost DPA-resistant 8-bit AES core based on ring oscillators. In Proceedings of the Technical Program of 2012 VLSI Design, Automation and Test, Hsinchu, Taiwan, 23–25 April 2012; pp. 1–4.
35. Liu, P.C.; Chang, H.C.; Lee, C.Y. A true random-based differential power analysis countermeasure circuit for an AES engine. *IEEE Trans. Circuits Syst. II Express Briefs* **2012**, *59*, 103–107. [CrossRef]
36. Chung, S.C.; Yu, C.Y.; Lee, S.S.; Chang, H.C.; Lee, C.Y. An improved DPA countermeasure based on uniform distribution random power generator for IoT applications. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2017**, *64*, 2522–2531. [CrossRef]
37. Parrilla, L.; Castillo, E.; Todorovich, E.; García, A.; Morales, D.P.; Botella, G. Improvements for the applicability of power-watermarking to embedded IP cores protection: E-coreIPP. *Digit. Signal Process.* **2015**, *44*, 110–122. [CrossRef]
38. Ziener, D.; Teich, J. Power signature watermarking of IP cores for FPGAs. *J. Signal Process. Syst.* **2008**, *51*, 123–136. [CrossRef]
39. Pub, F. Data Encryption Standard (des). FIPS PUB. 1999. pp. 46–53. Available online: https://www.techtarget.com/searchsecurity/definition/Data-Encryption-Standard (accessed on 24 May 2022).
40. Kocher, P.C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 104–113.
41. Bertoni, G.; Breveglieri, L.; Fragneto, P.; Macchetti, M.; Marchesin, S. Efficient software implementation of AES on 32-bit platforms. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Redwood Shores, CA, USA, 13–15 August 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 159–171.
42. Ors, S.B.; Gurkaynak, F.; Oswald, E.; Preneel, B. Power-analysis attack on an ASIC AES implementation. In Proceedings of the International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004, Las Vegas, NV, USA, 5–7 April 2004; Volume 2, pp. 546–552.

43. Renauld, M.; Standaert, F.X.; Veyrat-Charvillon, N. Algebraic side-channel attacks on the AES: Why time also matters in DPA. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Lausanne, Switzerland, 6–9 September 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 97–111.
44. Mohamed, M.S.E.; Bulygin, S.; Zohner, M.; Heuser, A.; Walter, M.; Buchmann, J. Improved algebraic side-channel attack on AES. *J. Cryptogr. Eng.* **2013**, *3*, 139–156. [CrossRef]
45. Roche, T.; Lomné, V.; Khalfallah, K. Combined fault and side-channel attack on protected implementations of AES. In Proceedings of the International Conference on Smart Card Research and Advanced Applications, Leuven, Belgium, 14–16 September 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 65–83.
46. Niu, Y.; Zhang, J.; Wang, A.; Chen, C. An efficient collision power attack on AES encryption in edge computing. *IEEE Access* **2019**, *7*, 18734–18748. [CrossRef]
47. Wang, R.; Wang, H.; Dubrova, E.; Brisfors, M. Advanced Far Field EM Side-Channel Attack on AES. In Proceedings of the 7th ACM on Cyber-Physical System Security Workshop, Hong Kong, China, 7 June 2021; pp. 29–39.
48. Cristiani, V.; Lecomte, M.; Hiscock, T. A bit-level approach to side channel based disassembling. In Proceedings of the International Conference on Smart Card Research and Advanced Applications, Prague, Czech Republic, 11–13 November 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 143–158.
49. Méndez Real, M.; Salvador, R. Physical side-channel attacks on embedded neural networks: A survey. *Appl. Sci.* **2021**, *11*, 6790. [CrossRef]
50. Tillich, S.; Herbst, C.; Mangard, S. Protecting AES software implementations on 32-bit processors against power analysis. In Proceedings of the International Conference on Applied Cryptography and Network Security, Zhuhai, China, 5–8 June 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 141–157.
51. Krieg, A.; Grinschgl, J.; Steger, C.; Weiss, R.; Haid, J. A side channel attack countermeasure using system-on-chip power profile scrambling. In Proceedings of the 2011 IEEE 17th International On-Line Testing Symposium, Athens, Greece, 13–15 July 2011; pp. 222–227.
52. Singh, A.; Kar, M.; Ko, J.H.; Mukhopadhyay, S. Exploring power attack protection of resource constrained encryption engines using integrated low-drop-out regulators. In Proceedings of the 2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Rome, Italy, 22–24 July 2015; pp. 134–139.
53. Yu, W.; Köse, S. A voltage regulator-assisted lightweight AES implementation against DPA attacks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2016**, *63*, 1152–1163. [CrossRef]
54. Kamoun, N.; Bossuet, L.; Ghazel, A. A masked correlated power noise generator use as a second order DPA countermeasure to secure hardware AES cipher. In Proceedings of the ICM 2011 Proceeding, Hammamet, Tunisia, 19–22 December 2011; pp. 1–5.
55. Messerges, T.S. Securing the AES finalists against power analysis attacks. In Proceedings of the International Workshop on Fast Software Encryption, New York, NY, USA, 10–12 April 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 150–164.
56. Golić, J.D.; Tymen, C. Multiplicative masking and power analysis of AES. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Hammamet, Tunisia, 19–22 December 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 198–212.
57. Trichina, E. Combinational Logic Design for AES Subbyte Transformation on Masked Data. Cryptology EPrint Archive. 2003. Available online: https://eprint.iacr.org/2003/236 (accessed on 23 May 2022).
58. KF, M.A.; Ganesan, V.; Bodduna, R.; Rebeiro, C. PARAM: A microprocessor hardened for power side-channel attack resistance. In Proceedings of the 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), San Jose, CA, USA, 7–11 December 2020; pp. 23–34.
59. Castillo, E.; Meyer-Baese, U.; García, A.; Parrilla, L.; Lloris, A. IPP@ HDL: Efficient intellectual property protection scheme for IP cores. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2007**, *15*, 578–591. [CrossRef]
60. Degalahal, V.; Tuan, T. Methodology for high level estimation of FPGA power consumption. In Proceedings of the 2005 Asia and South Pacific Design Automation Conference, Shanghai, China, 18–21 January 2005; pp. 657–660.
61. La, T.M.; Matas, K.; Grunchevski, N.; Pham, K.D.; Koch, D. Fpgadefender: Malicious self-oscillator scanning for xilinx ultrascale+ fpgas. *ACM Trans. Reconfigurable Technol. Syst. (TRETS)* **2020**, *13*, 1–31. [CrossRef]
62. Wold, K.; Tan, C.H. Analysis and enhancement of random number generator in FPGA based on oscillator rings. In Proceedings of the 2008 International Conference on Reconfigurable Computing and FPGAs, Cancun, Mexico, 3–5 December 2008; pp. 385–390.
63. Parrilla, L.; García, A.; Castillo, E.; Álvarez-Bermejo, J.A.; López-Villanueva, J.A.; Meyer-Baese, U. Dracon: An Open-Hardware Based Platform for Single-Chip Low-Cost Reconfigurable IoT Devices. *Electronics* **2022**, *11*, 2080. [CrossRef]