Sequence-Aware Watermark Design for Soft IP Embedded Processors

Jedrzej Kufel, Peter Wilson, Stephen Hill, Bashir M. Al-Hashimi, and Paul N. Whatmough

***Abstract*—This paper describes a design approach for incorporating sequence-aware watermarks in soft IP Embedded Processors. The influence of watermark sequence parameters on detection, area and power overheads is examined, and consequently a sequence-aware method for incorporating sequence-aware watermarks in soft IP Embedded Processors is proposed. The intrinsic parameters of sequences, such as the *activity factor* and the *overlapping factor* are introduced, and their impact on correlation results is demonstrated. Measurement experimental results from FPGA and ASIC validate the design approach and demonstrate the resulting IP protection and subsequent costs for constrained embedded processors. Results presented in this paper show that the tradeoff occurs between the watermark robustness against third party IP attacks and hardware implementation costs. The analysis of this tradeoff is provided and an application specific watermark implementation is proposed.**

***Index Terms*—Watermarking, IP Protection, Embedded Processors, Correlation Power Analysis.**

I. Introduction

TECHNOLOGY scaling and innovations in modern processes are allowing increasingly complex systems to be implemented on a single die [1]. To support this design complexity it is desirable to source sub-systems, such as CPUs, from external Intellectual Property (IP) suppliers. IP blocks are usually delivered as either hard-macros, full circuit layouts, or soft-macros, typically register-transfer level (RTL) descriptions. The Virtual Socket Interface (VSI) Alliance [2] proposes three approaches to the problem of securing an IP, namely deterrent, protection and detection. The deterrent approach may deter the infringement from occurring through patents, copyrights, contracts or lawsuits [2]. However, it does not provide physical protection. The protection approach prevents unauthorized use of IP through encryption. Nonetheless, encryption and rights managements support in EDA tools is far from universal and pain-free [3]. Therefore, IP blocks are often supplied as unprotected design files that System-on-Chip (SoC) integrators can use without any complication of their design flow. As a result, auditing the presence of IP in finished products is an important challenge for IP providers. De-encapsulation and die-level reverse engineering can be used to prove the presence of IP, but the process is slow and costly [3], [4]. It is therefore desirable to identify IP candidates to be short-listed for more thorough investigation.

The VSI Alliance proposes digital watermarking as one of detection methods for physical IP protection, at various design levels [2]. In a hard IP, a digital watermark is represented as physical modifications to the IC layout. Techniques alter the placement of technology library cells through the parity modification [5] or scattering [6], [7], modify interconnects in digital or analog devices [8], [9] or utilize intrinsic features of physical IC layout, obtained with EDA tools [10]. In a firm IP, a digital watermark is embedded through application of additional constraints during the optimization steps, such as partitioning [11], [12], graph coloring [13], [14], template matching or operation scheduling [15]. The hard and firm IP protection techniques generate highly tamper-resistant watermarks with negligible area overheads. However, an access to a watermarked design, such as a micro photograph, GDSII file, fully placed and routed or a partial netlist are required. Hence, such techniques are not in the scope of this paper. The use of the soft IP is more desirable as it offers the end user the highest level of flexibility [2]. Therefore, this paper focuses on digital watermarks embedded in a soft IP.

Embedded processors are constrained in terms of circuit area and power consumption. Therefore, area and power overhead minimization of the watermark circuit must be addressed for IP protection of embedded processors. The primary motivation of this work is the analysis of the current power watermark circuit design, which enables the non-invasive detection of a watermark in a fabricated device. Furthermore, this work investigates the reduction of the area and power overheads, necessary for highly constrained embedded processors. Such investigation is performed through the analysis of intrinsic parameters of watermarking sequences and their impact on hardware implementation costs and detection performance. None of the previous power watermark publications [16]–[18] have compared watermark sequences in such way. The commonly used sequence for power watermarks, found in the literature [16]–[18], is a 32-bit maximum length sequence (m-sequence). In this work, sequences not previously discussed in the area of IP power watermarking, such as Barker codes, are compared with m-sequence [18]. Although Barker codes are not new and can be found in the field of communication technology [19] and radar technology [20], their use in the context of IP power watermarking is novel. The sequences have been chosen to demonstrate various combinations of intrinsic parameters. Nevertheless, the provided theoretical analysis is valid for any other sequence. The theoretical analysis is validated with measurements using FPGA and two

J. Kufel is with ARM Ltd., Cambridge, CB1 9NJ, U.K. (e-mail: Andrew.Kufel@arm.com).

P. Wilson and B. M. Al-Hashimi are with the School of Electronics and Computer Science, University of Southampton, SO17 1BJ, U.K. (e-mail: prw@ecs.soton.ac.uk; bmah@ecs.soton.ac.uk).

S. Hill is with ARM Ltd., Austin, TX 78746, U.S.A. (e-mail: Stephen.Hill@arm.com).

P. N. Whatmough is with ARM Ltd., Cambridge, CB1 9NJ, U.K. (e-mail: Paul.Whatmough@arm.com).

ASIC designs. Furthermore, the strong relationship between the choice of a watermark sequence and hardware implementation costs and detection performance is shown and the best sequence for digital power watermarks is determined.

The paper is organized as follows. In Section II, previous work in the area of soft IP protection is analyzed. Section III, demonstrates the architecture of a power watermark circuit. Section IV, provides the in-depth analysis of the Pearson correlation coefficient and introduces the intrinsic parameters of watermark sequences. The sequences for power watermarks are discussed in Section V. The simulation results of sequences are given in Section VI. Section VIII validates simulation results on FPGA and ASICs, discusses the detection performance and hardware implementation costs. In Section IX, the secure digital signature generation methodology is discussed for short length sequences. Section X analyzes the application specific watermark implementation and Section XI compares non-triggering and trigger-based watermark implementations with regards to third party IP attacks. Section XII, concludes the paper.

II. Related Work

The methods for the protection of soft IP can be divided into two groups; invasive and non-invasive. The invasive detection techniques require an access to device internals, such as input and output (I/O) ports, memories and a full or partial knowledge of a system. The Finite State Machines (FSM) have been successfully used through an addition of extra states [21], [22], partial [23], [24] or complete [25] reuse of existing states, and allow a significant reduction of hardware implementation costs. To extract the embedded digital watermarks the output ports of a device are observed, while the dedicated activation vectors, integrated as a part of a test kernel [26], are applied to the input ports. Other techniques embed digital watermarks in look-up tables (LUT) on FPGA [27], but require a dedicated co-processor, to scan the data being fetched from the memory or execute special watermark instructions [28]. Upon detection of a unique input sequence or instruction, a watermark is copied to the specific memory location. In a typical invasive watermarking approach, such as FSM [21]–[25], the watermark is interwoven with the original IP and the removal of the watermarked logic compromises the design. To perform a post-fabrication detection, an IP provider requires an architectural knowledge, such as I/O ports or memory. In case of soft IP, an IP provider knows the architecture of the IP, however, he/she may lack the in-depth knowledge of how it will be integrated as part of a system.

In the non-invasive detection techniques, the system's knowledge is significantly reduced and an access to device internals is not required. The sources of information, also known as side-channel parameters, such as electromagnetic (EM) field radiation and power consumption can be used to detect an embedded watermarks. Techniques based on analysis of EM field [29], [30] offer a high degree of detectability by placing an EM sensor close to a device and performing an EM field characterization with a spectrum analyzer. Nevertheless, in this paper we focus on non-invasive power analysis techniques. The watermark detection is achieved by placing a current sensor and measuring a device power consumption. Since the embedded power watermark causes a deterministic power overhead, the threshold-based [16], [17] or statistical-based [18] detection techniques can be applied. In the threshold-based techniques, a device must be held in a reset state during the power measurement. Moreover, due to the nature of the algorithm deeply embedded watermark power signals cannot be detected. Therefore, statistical-based power analysis techniques, such as Correlation Power Analysis (CPA) [31], are used to detect deeply embedded watermark power signals, using the dynamic or static current variations on the supply voltage rail. The architecture of a power watermark consist of two circuits: a watermark generation circuit (WGC) and a watermark power pattern generator (WPPG) [16]–[18]. The architecture of the WGC depends on the watermark sequence. Nonetheless, it is kept relatively small, and 32 registers have been reported in [16]–[18]. The WPPG consists of shift registers and determines the power consumed by the watermark circuit. Its size is closely related to the system size. In [16], 92 out of 1332 lookup tables (LUT) on FPGA, a 6.9% of system area, were used with each LUT configured as 16-bit shift register, for simple arithmetic coder core. Similarly in [18], 16 LUTs were used to watermark the Advanced Encryption Standard (AES) cryptographic core. As can be seen, the majority of area overhead in the current state-of-the-art power watermark architecture is caused by the significant size WPPG circuit. Although, the WPPG circuit area overhead can be minimized through reuse of existing LUTs [16], [17], such approach is specific to FPGA architecture. Moreover, the device must be held in a reset state to successfully perform a watermark detection. However, this paper focuses on detecting an embedded watermark during the active processor mode. Furthermore, many techniques have been demonstrated which allow detection of a negligible sized circuits, through integration of ring oscillator networks [32], power measurements of multiple supply pads [33] or the combination of numerous side-channel parameters [34]. Nevertheles, the design knowledge is too fine and destructive IC tests are often necessary. Therefore, the CPA [18] remains the current state-of-the-art for power watermark detection and is used in this paper.

III. Watermark Architecture

A power watermark is a redundant circuit added to an existing IP block, with the aim of superimposing a weak but deterministic signal on a supply voltage rail. In Fig. 1(a), a typical embedded system is shown, with multiple IP blocks sub-sourced from various IP vendors. The watermark is embedded in one of the IP blocks and consists of two circuits: a watermark generation circuit (WGC), and a watermark power pattern generator (WPPG) [18]. The WGC generates the watermark sequence ($'WMARK'$) which controls the WPPG load circuit. Hence, the WPPG consumes power in clock cycles where $WMARK$ is '1'. Simulation results in Fig. 1(b), demonstrate the effect of an additional watermark circuit on the device total power (in relative terms). The watermark power signal (middle) is added to the power consumed by the

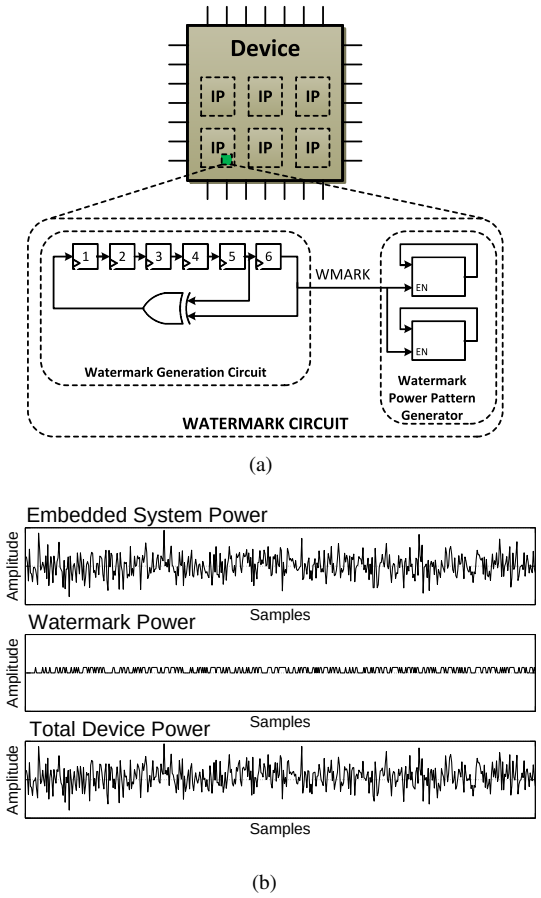


Fig. 1. (a) Architecture of a power watermark circuit. (b) Simulation results of the effect of a watermark power signal on device total power.

embedded system (top), and generates the device total power (bottom). Since the watermark power signal is a much lower amplitude, it is deeply embedded in the overall device power signal. An analytical technique is therefore required, which determines the possibility and the accuracy of a watermark existence. Such a technique is the Correlation Power Analysis (CPA) and has been used in this work as the fundamental power watermark detection technique.

IV. Pearson Correlation Coefficient Analysis

The CPA uses the statistical correlation technique to detect a deeply embedded power watermark signal. The Pearson correlation coefficient is computed as in Eq. (1). In this section, the modifications to Eq. (1) are proposed, to include the intrinsic parameters of watermark sequences.

The CPA [31] requires an information extracted from the measured power consumption of a device, recorded using an oscilloscope. The sampling frequency of an oscilloscope, f_s, is much greater than the frequency of a system clock, f_{clk}, i.e. $f_s >> f_{clk}$. The power vector, Y, is found by averaging all the samples within a single clock cycle, as in [18]. The watermark model vector, X, represents a watermark sequence. As both vectors must be of equal length, a watermark sequence is repeated many times within the X vector. Therefore, the X vector consists of multiple periods of a watermark sequence, to find a single Pearson correlation coefficient, ρ, given by

$$\rho = \frac{N\sum_{i=1}^{N} X_i Y_i - \sum_{i=1}^{N} X_i \sum_{i=1}^{N} Y_i}{\sqrt{N\sum_{i=1}^{N} X_i^2 - (\sum_{i=1}^{N} X_i)^2}\sqrt{N\sum_{i=1}^{N} Y_i^2 - (\sum_{i=1}^{N} Y_i)^2}} \quad (1)$$

Where X can be represented by a binary sequence and Y consists of the sampled power signal. N is the length of both vectors and contains only full periods (M) of a watermark sequence ($N \equiv 0 \ (mod \ M)$). Since both model and power vectors may be out of phase, X is repeatedly rotated by a single clock cycle and the correlation is recomputed [18]. The number of rotations is M. Once all M correlation values have been found, they can be represented by a spread spectrum graph (see Fig. 2) [18]. The watermark is only regarded as detected, if a single significant correlation coefficient can be resolved, as demonstrated in Fig. 2.

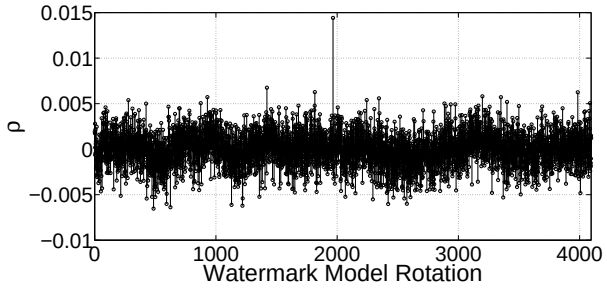


Fig. 2. Spread spectrum of correlation coefficients.

The dynamic power of a canonical static CMOS gate is linearly proportional to the switching activity, α [35]. In digital power watermarking, the activity factor is intrinsic to a watermark sequence, and the dynamic power is consumed in clock cycles when watermark sequence is '1'. Therefore, Eq. (1) can be modified to incorporate the *activity factor*, α, of a watermark sequence into the Pearson correlation. In Section VI, Table II, various watermark sequences are considered and it is demonstrated that α differs between sequences. To compare the detection performance of potential watermark sequences, it is crucial to consider the α parameter. Since vectors X and Y can be out of phase, the watermark model, X, is rotated M times and correlation computation is repeated [18]. In Eq. (1), vector X is substituted with X', which represents the rotated vector X. If both vectors are in phase, then $X' = X$. Additionally, vector Y can be represented as $X + \beta$, where X is the original vector of the watermark model and β is the noise present in the system, such as global switching noise of digital IP blocks, environmental and measurement noise. Therefore, ρ becomes

$$\rho = \frac{N\sum_{i=1}^{N} X'_i(X_i+\beta_i) - \sum_{i=1}^{N} X'_i \sum_{i=1}^{N} X_i + \beta_i}{\sqrt{N\sum_{i=1}^{N} X'^2_i - (\sum_{i=1}^{N} X'_i)^2}\sqrt{N\sum_{i=1}^{N} (X_i+\beta_i)^2 - (\sum_{i=1}^{N} X_i + \beta_i)^2}} \quad (2)$$

Since both vectors X and X' represent a binary sequence, the sum of all terms in a vector ($\sum X_i$ and $\sum X'_i$) is the Hamming Weight, H, of a sequence. Moreover, as both X and X' represent the same but rotated binary sequence, the Hamming Weight is the same. Furthermore, if both vectors X' and X are in phase we have

$$\sum_{i=1}^{N} X'_i X_i = \sum_{i=1}^{N} X_i = H \tag{3}$$

However, since vectors X' and X can be out of phase, the *overlapping factor*, θ, is introduced, such that

$$\sum_{i=1}^{N} X'_i X_i = \theta H \tag{4}$$

To illustrate this point, consider 2 watermark model vectors, where one is the cyclically rotated version of the other.

$$X: \quad 1111000000\ 1111000000 \tag{5}$$
$$X': \quad 0111100000\ 0111100000 \tag{6}$$

$$\sum_{i=1}^{N} X_i X'_i = 6 = \theta H \quad \theta = 6/8 = 0.75 \tag{7}$$

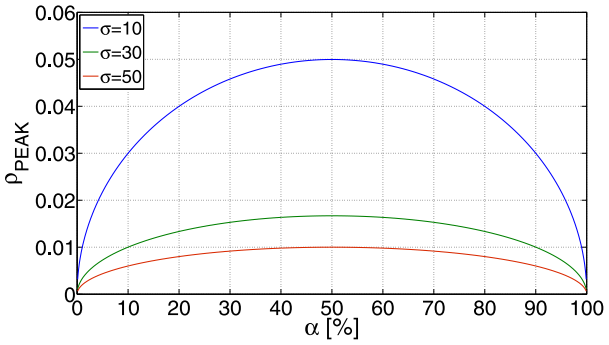
The *overlapping factor*, θ, is 1 when both vectors are in phase, and $\theta < 1$ when both vectors are out of phase, including other rotations of vector X'. In Section VI, Table II, θ_{MAX}, is shown and describes the highest *overlapping factor*, θ, under the assumption that X and X' are not in phase. As shown in Table II, θ_{MAX} varies significantly between sequences. Furthermore, the Hamming Weight, H, can be substituted as the product of the *activity factor*, α, and the length N of vectors, since $\alpha = \frac{H}{N}$. Finally, the Pearson's correlation coefficient, ρ, can be described as a function of *activity*, *overlapping factor*, and both X and X' vectors as follows

$$\rho(\alpha, \theta, X_i, X'_i) = \frac{N\alpha(\theta-\alpha) + \sum_{i=1}^{N}(X'_i\beta_i) - \alpha\sum_{i=1}^{N}\beta_i}{\sqrt{\alpha(1-\alpha)}\sqrt{N^2\alpha(1-\alpha) + N(2\sum_{i=1}^{N}(X_i\beta_i) + \sum_{i=1}^{N}\beta_i^2) - \sum_{i=1}^{N}\beta_i(2\alpha N + \sum_{i=1}^{N}\beta_i)}} \tag{8}$$

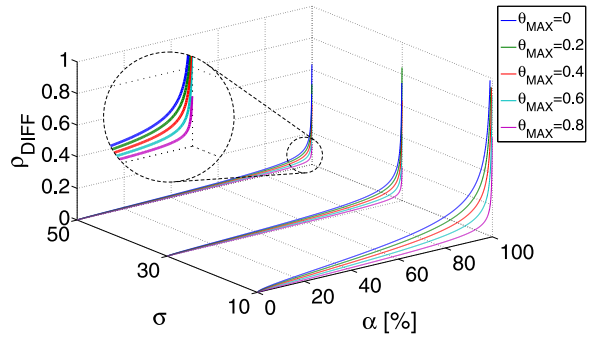
The terms $\sum_{i=1}^{N}(X'_i\beta_i)$ and $\sum_{i=1}^{N}(X_i\beta_i)$ in Eq. (8) depend on the position of '1' in a watermark sequence. However, since $N >> 1$, Eq. (8) can be simplified to

$$\rho(\alpha, \theta) = \frac{N\alpha(\theta-\alpha)}{\sqrt{\alpha(1-\alpha)}\sqrt{N^2\alpha(1-\alpha) + N\sum_{i=1}^{N}\beta_i^2}}, N >> 1 \tag{9}$$

By definition, in a spread spectrum a single correlation peak should be distinguishable, to consider a watermark detected [31]. The maximum correlation coefficient, ρ_{PEAK}, is



(a)



(b)

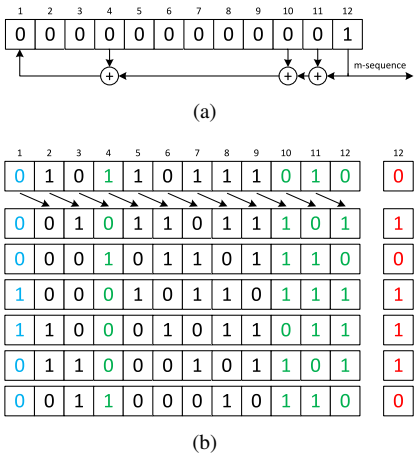
Fig. 3. The influence of the *activity factor*, α, and the maximum *overlapping factor*, θ_{MAX}, on (a) maximum correlation coefficient, ρ_{PEAK}, and (b) correlation coefficient difference, ρ_{DIFF}; MATLAB simulations.

$$\rho_{PEAK} = \rho(\alpha, 1) = \frac{N\alpha(1-\alpha)}{\sqrt{\alpha(1-\alpha)}\sqrt{N^2\alpha(1-\alpha) + N\sum_{i=1}^{N}\beta_i^2}}, N >> 1 \tag{10}$$

It is expected (Fig. 2), that the highest ρ_{PEAK} occurs when both vectors X and X' are in phase, i.e. $\theta = 1$. In the noiseless environment, ρ_{PEAK} is 1 for all sequences, since

$$\rho_{PEAK} = \rho(\alpha, 1) = \frac{N\alpha(1-\alpha)}{\sqrt{\alpha(1-\alpha)}\sqrt{N^2\alpha(1-\alpha)}} = 1 \tag{11}$$

In Fig. 3(a), MATLAB simulation of Eq. (10) is shown with various watermark sequences and noise levels. The noise vector, β, consists of normally distributed random values. The frequency spectrum is shown in Section VI, Fig. 6(a). Therefore, it approximates white noise, to represent the global switching noise of digital IP blocks, environmental and measurement noise [36]. Since the mean value of β is 0, the power follows the variance, σ^2. To increase the power of β (Fig. 3(a)), the standard deviation, σ, of the generated random values is increased. From Eq. (10), ρ_{PEAK} is principally influenced by the *activity factor*, α. Due to the parabolic shape of the graph, watermark sequences with $\alpha \approx 50\%$ produce the highest ρ_{PEAK}. As the noise increases, the term $N\sum_{i=1}^{N}\beta_i^2$ becomes dominant and the graph becomes flatter. The ρ_{PEAK} decreases and watermark sequences produce similar results.



(a)

(b)

Fig. 4. (a) Block diagram of the 12-bit LFSR. (b) States of LFSR in consecutive clock cycles.

In practice, a power supply noise and measurement error give rise to undesirable spurious correlation coefficients. If such spurii are considered as the system noise floor, the correlation coefficient difference, ρ_{DIFF}, can be described as the distance from ρ_{PEAK} to the noise floor, as in Eq. (12).

$$\begin{aligned}\rho_{DIFF} &= \rho_{PEAK} - \rho(\alpha, \theta, X_i, X_i') \\ &= \frac{N\alpha(1-\theta)}{\sqrt{\alpha(1-\alpha)}\sqrt{N^2\alpha(1-\alpha) + N\sum_{i=1}^{N}\beta_i^2}}, \theta < 1, N >> 1\end{aligned} \tag{12}$$

The simulation of Eq. (12) with various watermark sequences is shown in Fig. 3(b). As expected, ρ_{DIFF} is influenced by both α and θ_{MAX} parameters. If the noise standard deviation, σ, is increased, ρ_{DIFF} approaches 0. This means that there is no distinctive correlation peak in a spread spectrum graph, and a watermark cannot be found.

In this section, the intrinsic parameters of sequences, such as the *activity factor*, α, and the *overlapping factor*, θ, have been introduced, and their impact on correlation coefficient values has been demonstrated. The significance of Eq. (10) and Eq. (12) is the ability to design embedded power watermarks with low overheads. In the following sections, the sequences for power watermarks are presented and the discussion, supported by simulation results, of differences between ρ_{DIFF} and ρ_{PEAK} is provided. Furthermore, the detection performance of watermark sequences is established.

V. Sequences for Power Watermarks

In this section, two types of binary sequences are discussed. These are sequences generated with the Linear Feedback Shift Register (LFSR), as demonstrated in the current state-of-the-art power watermark architecture [16]–[18], and Barker codes. Such sequences have been chosen to demonstrate the impact of various intrinsic parameters and lengths on detection performance, hardware implementation costs and robustness against third party IP attacks, analyzed in the following sections.

A. Linear Feedback Shift Register

The binary sequence generated with the LFSR is also known as the maximum length sequence (m-sequence). The block diagram of the 12-bit LFSR is shown in Fig. 4(a) and can be described by the following polynomial [37]:

$$\mathbf{1}x^{12} + \mathbf{1}x^{11} + \mathbf{1}x^{10} + 0x^9 + 0x^8 + 0x^7 + 0x^6 + 0x^5 + \mathbf{1}x^4 + 0x^3 + 0x^2 + 0x \tag{13}$$

The degree of the polynomial is directly related to the length of the LFSR and contains $'0'$ and $'1'$ coefficients, with $'1'$ corresponding to the taps of registers connected to XOR gates (shown in green in Fig. 4(b)), forming a feedback path. The last register in the LFSR is used as an output and generates the m-sequence (shown in red in Fig. 4(b)). For M number of registers, the length of the m-sequence is $2^M - 1$.

B. Barker Codes

The m-sequence is a unipolar sequence represented by $'1'$ and $'0'$. The Barker codes are bipolar sequences represented by $'1'$ and $'-1'$. Therefore, they must be transformed into their unipolar representation, by substituting all $'-1'$ with $'0'$. The commonly used Barker codes [38] are shown in Table I, and are used throughout this paper.

TABLE I
Barker Codes

Length	Bipolar Sequence	Unipolar Sequence
2	+1 -1	1 0
3	+1 +1 -1	1 1 0
4	+1 +1 -1 +1	1 1 0 1
5	+1 +1 +1 -1 +1	1 1 1 0 1
7	+1 +1 +1 -1 -1 +1 -1	1 1 1 0 0 1 0
11	+1 +1 +1 -1 -1 -1 +1 -1 -1 +1 -1	1 1 1 0 0 0 1 0 0 1 0

The generation of Barker codes can be achieved with simple circular shift registers. Since no feedback loop exists, the M-bit Barker code requires M number of registers. For comparison, a 12-bit LFSR generates m-sequence with the length of $4,095$ clock cycles, while using only 12 registers. The 11-bit Barker code generates a sequence with the length of 11 clock cycles and requires 11 registers. Therefore, the m-sequence architecture allows generation of much longer sequences with less number of registers. This has a direct impact on security of an embedded power watermarks, as discussed in Section XI.

VI. Simulation Results

The summary of sequences, discussed in Section V, and their parameters are shown in Table II. As can be seen, α, and θ_{MAX} vary for all Barker codes. However, for m-sequences α decreases and tends to 50%, and θ_{MAX} is constant and equals 0.5, as the length of a sequence increases.

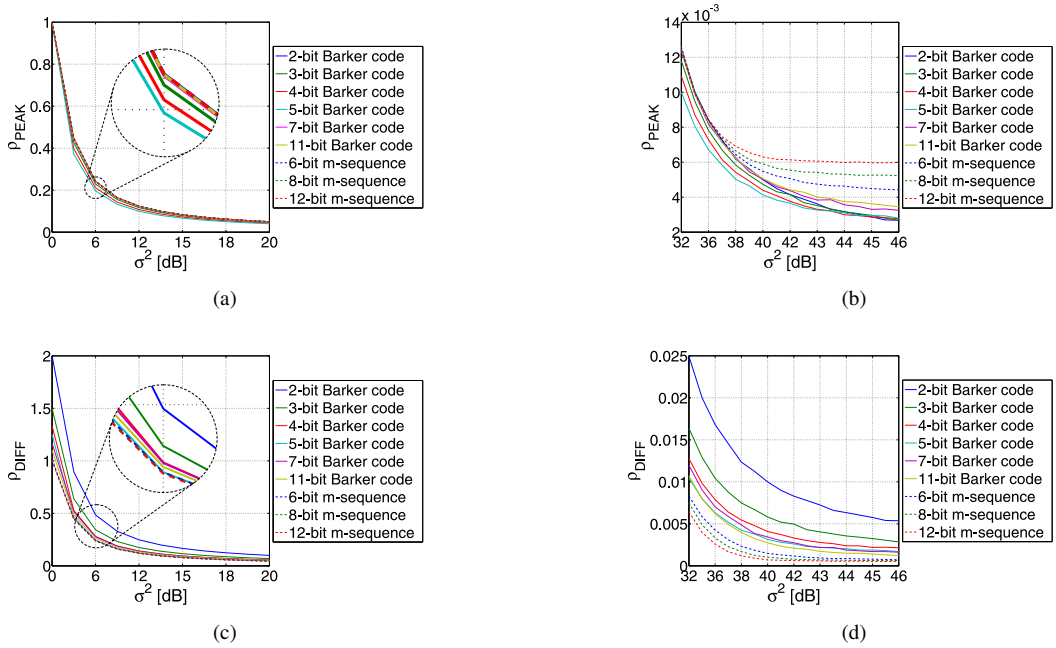


Fig. 5. Maximum correlation coefficient, ρ_{PEAK} (a, b), and correlation coefficient difference, ρ_{DIFF} (c, d), simulation results of watermark sequences; MATLAB simulations.

TABLE II
PARAMETERS OF WATERMARK SEQUENCES

Watermark Sequence	Bit Length	Period (clock cycles)	*activity factor* α [%]	Maximum θ_{MAX}
Barker	2-bit	2	50%	0
	3-bit	3	66.6%	0.5
	4-bit	4	75%	0.667
	5-bit	5	80%	0.75
	7-bit	7	57.15%	0.5
	11-bit	11	45.45%	0.4
m-sequence	6-bit	63	50.8%	0.5
	8-bit	255	50.2%	0.5
	12-bit	4095	50.01%	0.5

A. Maximum Correlation Coefficient

The maximum correlation coefficient, ρ_{PEAK}, of watermark sequences at various noise levels is shown in Fig. 5(a), and Fig. 5(b). For noise signals with relatively low power (Fig. 5(a)), watermark sequences with $\alpha \approx 50\%$ produce the highest ρ_{PEAK}. This is as expected based on results in Section IV, Fig. 3(a). As the system noise level increases, the watermark signal-to-noise ratio (SNR) decreases. Finally, it reaches the point where the watermark power signal is too low to be reliably detected. In the marginal case, results of ρ_{PEAK} are dictated by the robustness of a watermark sequence against the correlation to the noise present in the system. Therefore, the results can be considered as the *noise-to-sequence* correlation, ρ_{NOISE}. In Fig. 5(b), it can be seen that when the noise power approaches $38dB$, ρ_{PEAK} of m-sequences are higher than other sequences. As the noise level is further increased, the length of a sequence determines the *noise-to-sequence* correlation, with longer sequences producing higher ρ_{PEAK}. To understand the reason of such behaviour, consider ρ in Eq. (1) when no watermark is present. Vector Y, which originally represents the measured power signal and contains the watermark model X and system noise β, is replaced with β, since no watermark exists. If the substitution of the Hamming Weight (Section IV) is followed, ρ_{NOISE} can be represented as

$$\rho_{NOISE} = \frac{\sum_{i=1}^{N} X'_i\beta_i - \alpha \sum_{i=1}^{N} \beta_i}{\sqrt{\alpha(1-\alpha)}\sqrt{N\sum_{i=1}^{N}\beta_i^2 - (\sum_{i=1}^{N}\beta_i)^2}} \quad (14)$$

We simulated Eq. (14) with sequences of various lengths and α and found that α had no effect on ρ_{NOISE} for very low SNR. Therefore the diminishing effect of α on correlation coefficients can be observed, as the noise power is increased.

The period of a watermark sequence (M) determines the number of frequency components in the watermark model. Short sequences contain only a few frequency components, Fig. 6(b). As the length of a sequence increases, more frequency components appear in the frequency spectrum of the watermark model, Fig. 6(c). In Fig. 6(a), the frequency spectrum of the noise signal obtained from simulations is shown. If the convolution of the watermark model and noise signal is considered, Fig. 6(d) and Fig. 6(e), the overlapping area between the two signals increases with the length of a sequence. Therefore, more information contained within the noise signal is retained, Fig. 6(f) and Fig. 6(g). At the same time, the correlation between the two signals increases, which causes ρ_{PEAK} to be higher for longer m-sequences.

B. Correlation Coefficient Difference

Equation (12) demonstrates that ρ_{DIFF} is determined by α and θ (θ_{MAX}) parameters. It should be noted that the highest

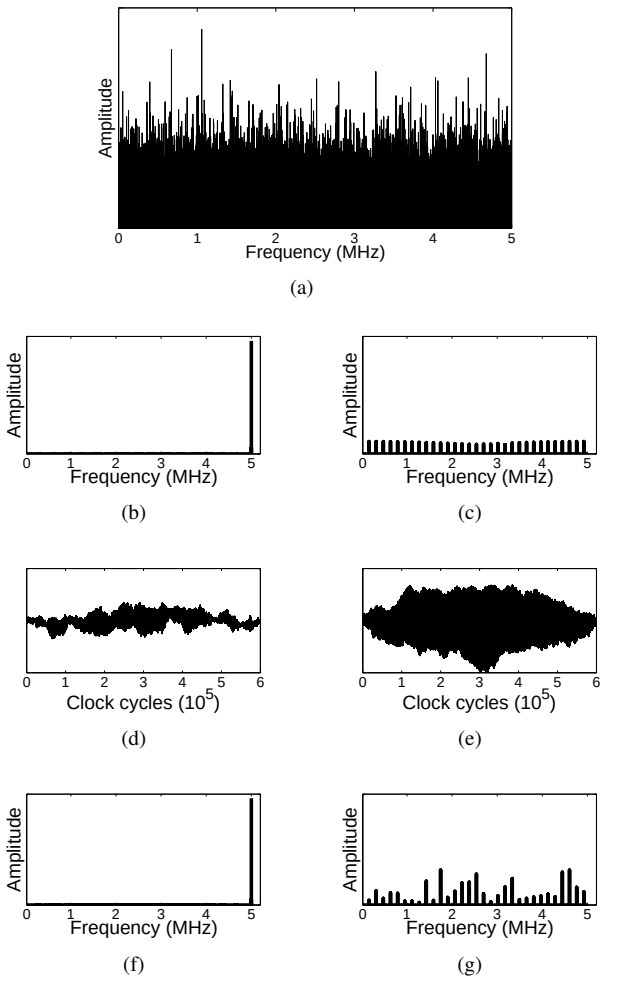


Fig. 6. Frequency spectra of (a) noise β, (b) 2-bit Barker code, (c) 6-bit m-sequence. Convolutions of (d) 2-bit Barker code, (e) 6-bit m-sequence, with noise β. Frequency spectra of convolved (f) 2-bit Barker code, (g) 6-bit m-sequence, with noise β.

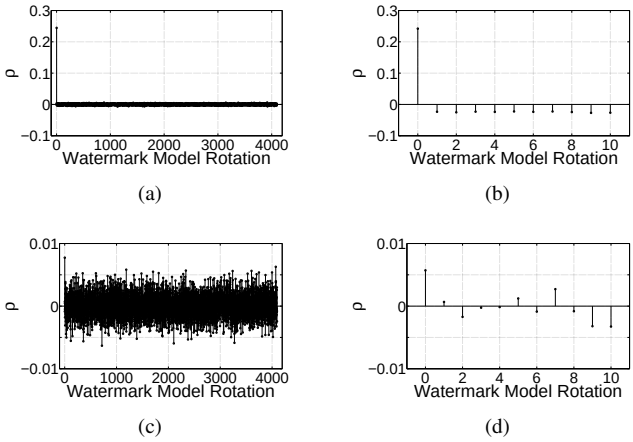


Fig. 7. Spread spectra of (a, c) 12-bit m-sequence and (b, d) 11-bit Barker code for noise with power of $6dB$ and $40dB$, respectively.

ρ_{DIFF} occur for watermark sequences with the highest α and lowest θ_{MAX} (see Fig. 3(b)). In Fig. 5(c), the 2-bit Barker code produces the highest ρ_{DIFF}, since $\alpha = 50\%$, and $\theta_{MAX} = 0$. Maximum length sequences (m-sequences) produce much lower ρ_{DIFF} than most of the Barker codes, since α reduces when M is increased and tends to 50%, while θ_{MAX} remains at 50%. The difference between subsequent m-sequences is minimal due to the same θ_{MAX} and similar α. However, since longer watermark sequences contain more frequency components that correspond with the noise signal, there are multiple correlation coefficients with values close to ρ_{PEAK}. This causes ρ_{DIFF} to be much lower as the watermark sequence length increases (see Fig. 5(d)).

In Fig. 5, the relationship between ρ_{DIFF} and ρ_{PEAK} varies with power of the generated noise signal and watermark sequences. In Fig. 7(a) and Fig. 7(b), the 12-bit m-sequence and 11-bit Barker code are shown, for $6dB$ noise signal. As can be seen for 12-bit m-sequence (Fig. 7(a)), ρ_{DIFF} and ρ_{PEAK} have similar values (0.25), since the noise floor in the spread spectrum is close to 0. However, for 11-bit Barker code (Fig. 7(b)) ρ_{DIFF} (0.27) is higher than ρ_{PEAK} (0.25). Increasing the noise power to $40dB$ (Fig. 7(c) and Fig. 7(d)), causes ρ_{DIFF} to be of much lower amplitude than ρ_{PEAK}, since the noise floor increases and gets closer to ρ_{PEAK}. In Fig. 7(c), the noise floor is of similar amplitude as ρ_{PEAK} in Fig. 7(d). This is as expected, since ρ_{PEAK} is higher for longer sequences, for very low SNR (Fig. 5(b)). However, these are separate test cases and as shown in Section VI-C, the threshold levels differ based on a sequence. Therefore, as shown in Fig. 7(d), at $40dB$ 11-bit Barker code is clearly detectable. However, in Fig. 7(c) the noise floor in the spread spectrum is significant to detect the 12-bit m-sequence.

C. Null Hypothesis Significance Test

In Section VI-A and Section VI-B, the influence of watermark sequence length on *noise-to-sequence* correlations was demonstrated. In Fig. 5(b), results of ρ_{PEAK} are higher for longer m-sequences as the noise level increases. However, based on results of ρ_{DIFF} in Fig. 5(d), other correlation coefficients exist which make the spread spectrum more even, and no significant peaks can be distinguished, at high background noise levels. To compare the detection performance of watermark sequences, the Null Hypothesis Significance Test (NHST) [36] was performed for each sequence. The percentage of rejected null hypotheses was found by applying a 5% threshold to results, where the null hypothesis states that the watermark does not exists. The ρ_{DIFF} was chosen to describe the detection performance of a watermark sequence, since it considers multiple correlation values in a spread spectrum. If ρ_{DIFF} is above the threshold, the null hypothesis can be rejected with 5% possibility of a false alarm. This means that there is a 5% possibility of detecting a watermark which does not exist. To minimize the possibility of false alarms lower percentages, hence higher threshold levels can be used. To determine the threshold for each watermark sequence separately, the simulation was repeated 100 times with no watermark present in the system. The null hypothesis was found by plotting a distribution of ρ_{DIFF} from which a 5% threshold level was determined [36]. The process was repeated 10 times and the average threshold level was found

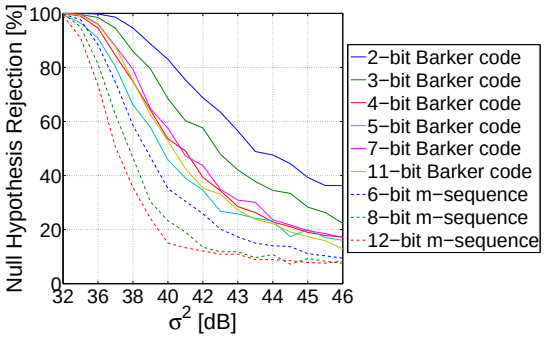
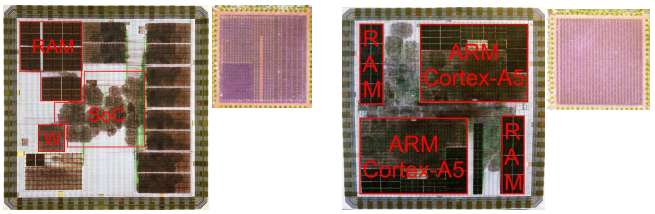


Fig. 8. NHST of simulated watermark sequences with 5% threshold level; MATLAB simulations.



(a) (b)

Fig. 9. Layout and die photo of test chips. (a) chip I, (b) chip II.

TABLE III
AREA OF WATERMARK CIRCUIT IMPLEMENTED ON FPGA

Watermark Sequence	WPPG Registers (SRL16)	FPGA Slices	Area Overhead
ARM Cortex-M0 IP core	-	2,696	-
7-bit Barker code	-	4	-
	8	**12**	**0.45%**
	16	20	0.74%
	32	36	1.34%
11-bit Barker code	-	6	-
	8	**14**	**0.52%**
	16	22	0.82%
	32	38	1.41%
6-bit m-sequence	-	5	-
	8	13	0.48%
	16	21	0.78%
	32	**37**	**1.37%**
8-bit m-sequence	-	5	
	8	13	0.48%
	16	21	0.78%
	32	**37**	**1.37%**
12-bit m-sequence	-	8	
	8	16	0.59%
	16	24	0.89%
	32	**40**	**1.48%**

for each watermark sequence. Next, the watermark was added and the simulations were repeated in the same way. The threshold levels were applied to each sequence, to determine the detection performance, as in Fig. 8. The difference between Barker codes is clearly distinguishable, however it is not as strong as in Fig. 5(d). This demonstrates the higher *noise-to-sequence* correlations of shorter sequences. Nevertheless, as the length of sequences increases, the null hypothesis rejection ratio decreases most quickly for longer m-sequences. When noise approaches $46dB$, most sequences reach the 5% threshold.

VII. Design Approach

To verify the theory and simulation results of Section IV and Section VI, the watermark circuit (see Fig. 1(a)), was embedded in an ARM Cortex-M0 microcontroller IP core implemented on a Xilinx Virtex-II Pro XC2VP30 FPGA, along with an on-chip bus and both program and data memories. An FPGA was used for illustration purposes to demonstrate the relationship between the WPPG size and detection performance, as in Section VIII-B.

The architecture of the watermark generation circuit (WGC) depends on the watermark sequence. The LFSRs (Section V-A) were used for m-sequences, and simple circular shift register were used for Barker codes (Section V-B). The output from the last register ($'WMARK'$) serves as the clock enable signal for the watermark power pattern generator (WPPG). The WPPG dissipates power due to shifting data in the flip-flops, when enabled by the $WMARK$ signal. In the Xilinx FPGA, a single LUT can be configured as a 16-bit shift register (SRL16). To increase the SNR between the watermark power signal and the system noise signal, the number of SRL16 blocks is increased. To generate the maximum power in clock cycles when watermark sequence is $'1'$, each SRL16 block is pre-initialized with $'1010...'$ sequence. In Table III, the size of the watermark circuit implemented on FPGA is shown for deterministic sequences, discussed in Section VIII-A and various WPPG sizes.

Additionally, to aid with experimental results and investigate the impact of process variation (PV) on watermark sequence detection results, two ASIC designs were fabricated in TSMC $65nm$ low leakage CMOS technology, with nominal operating voltage of $1.2V$. The designs were completed using industry standard EDA tools. In the first design (chip I), the watermark circuit ($'W'$) was embedded as a hard macro block, on a separate power domain, Fig. 9(a). The SoC consists of the ARM Cortex-M0 microcontroller IP core, along with an on-chip bus and numerous commercial IP blocks. In the second design (chip II), the watermark circuit was embedded from an RTL description, Fig. 9(b). Therefore, the watermark circuit was propagated through the entire design flow, which is closer to the intended usage scenario when embedding watermarked soft IP. The chip consists of dual core ARM Cortex-A5 microprocessor IP core and caches. The SoC, shown as the unmarked circuitry, consists of the ARM Cortex-M0 along with an on-chip bus and numerous commercial IP blocks and the watermark circuit. The watermark circuit architecture is the same in both chips, Fig. 10. To accommodate the possibility of generating various watermark sequences, the watermark circuit contains two sequence generators which can be configured as either 32-bit LFSRs or a simple 32-bit circular shift registers. The WPPG design contains $1,024$ registers, divided into 32 words. Upon watermark sequence bit $'1'$, all 32 words are rotated in a word-wise fashion. Therefore, to generate the maximum power, words are initialized to $0xFFFFFFFF$, and $0x00000000$, consecutively. For ASIC implementation of the watermark circuit, see [39].

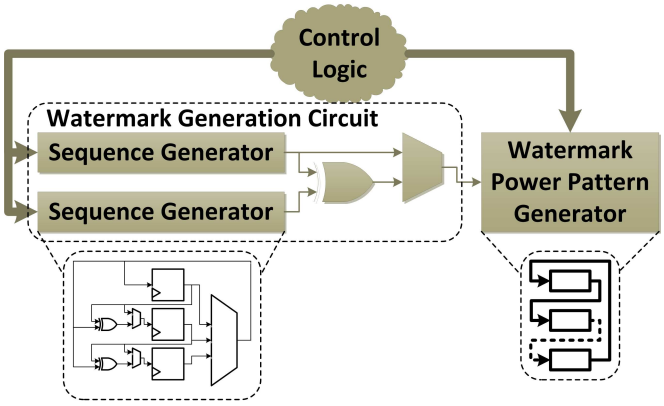


Fig. 10. Schematic diagram of the watermark circuit embedded in test chips.

VIII. Experimental Results

In the FPGA implementation the core voltage was measured very close to the package. In the ASIC implementation, the power domains were connected off-chip and the total current consumed by the chip was measured, using the shunt $270m\Omega$ resistor. The operating frequency of both FPGA and ASICs was $10MHz$. Such frequency was appropriate to demonstrate the effect of an embedded power watermark and compare the detection performance of watermarking sequences. The voltage and current signals were measured using an Agilent MSO6032A oscilloscope with Agilent 1130A active differential probe, at a sampling frequency of $500MHz$. Therefore, 50 samples per single clock cycle were averaged to obtain the power vector, Y. The length of both X and Y vectors was approximately $300,000$ clock cycles. We attempted to detect the watermark while running the Dhrystone benchmark. This is one of the most common benchmarks used in the industry to measure the performance of a processor, and reflects the activities of the integer IP processor core, such as integer arithmetic, string operations, logic decisions and memory accesses [40].

A. Repeatability of Results

In Fig. 11, FPGA measurements of ρ_{PEAK} are shown. Results match the simulation results of Section VI, Fig. 5(b). As the SNR between the watermark power signal and the system noise signal is high (128 to 16, SRL16), all watermark sequences generate similar ρ_{PEAK}. As the SNR decreases, the impact of α diminishes and the length of the watermark sequence becomes the major factor. Therefore, longer sequences produce higher ρ_{PEAK}.

The simulation results in Section VI show a clear differentiation between short and long watermark sequences. Moreover, no significant variations have been found between results when simulated multiple times. However, experimental FPGA results indicate that some watermark sequences are less repeatable than others. This means that when the experiment is repeated multiple times, distributions of ρ_{PEAK} or ρ_{DIFF}

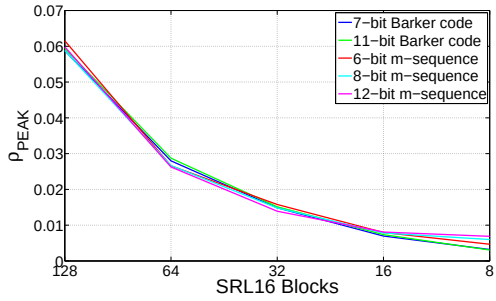


Fig. 11. FPGA experimental results of ρ_{PEAK}.

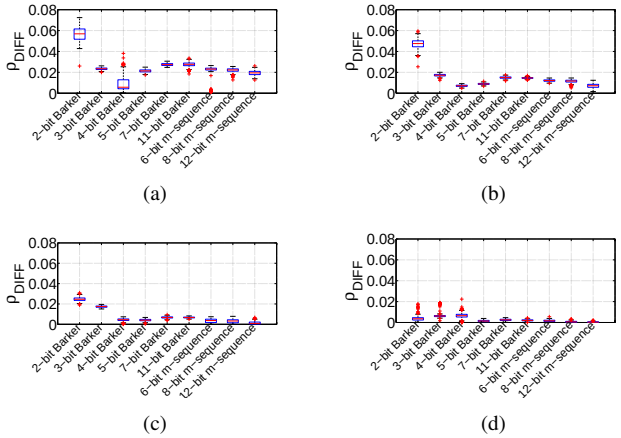
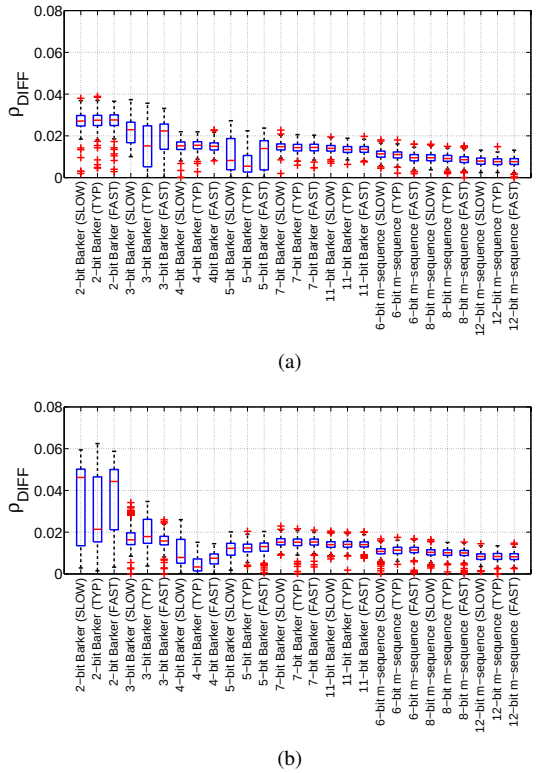


Fig. 12. Box plots of ρ_{DIFF} at various sizes of WPPG circuit on FPGA: (a) 64 SRL16, (b) 32 SRL16, (c) 16 SRL16, (d) 8 SRL16.

vary from test to test. In Fig. 12, the variance of results is shown in terms of box plots, for various sizes of WPPG circuit. Each box represents the combined distributions of ρ_{DIFF}, obtained from multiple tests. As in Section VI-C, the ρ_{DIFF} is used, since it considers other correlation coefficients in the spread spectrum. Nevertheless, the same variance occurs for ρ_{PEAK}. The test was repeated 3 times for each watermark sequence and the 100 point distributions were found for each test. The FPGA was re-configured between each test, and the delay between the start of the program and the start of the watermark circuit was modified. This causes the noise characteristics to vary between consecutive tests and correlate differently for some sequences. As can be seen in Fig. 12, short watermark sequences correlate with much higher variance for most WPPG circuit sizes. Additionally in Fig. 12(d), medians of very short watermark sequences do not match the simulation results discussed in Section VI, Fig. 5. According to the simulation results shorter watermark sequences produce higher ρ_{DIFF}, which are not observed for 2, 3, 4, and 5 bits Barker codes. As the period of a watermark sequence increases, the variance of results is significantly lower. Results become deterministic and the expected behaviour can be predicted.

The above process was repeated on both test chips. The number of test repetitions was increased to 5, to test the susceptibility of watermark sequences to run-to-run noise variations. Additionally, 30 chips were characterized and 3 corners were chosen: fast, slow, and typical. We investigate



(a)

(b)

Fig. 13. Box plots of ρ_{DIFF} on test chips: (a) chip I, (b) chip II.

the impact of PV, which occurs in the foundry during the chip fabrication. It should be noted that the current consumption measurement included the noise of the system caused by the SoC and RAM on both chips, and the clock tree of the dual core Cortex-A5 on chip II.

Results of ρ_{DIFF} obtained from both test chips are shown in Fig. 13. First, consider the impact of run-to-run variations (size of box plots), when the test is repeated multiple times, and the chip is re-configured between tests. Results confirm the FPGA conclusions. Short period sequences cause much higher variance in results than longer period sequences. Next, consider the impact of PV on ρ_{DIFF}, which is represented by the variance between the boxes in the box plot for the same chip and the same sequence length. For example in Fig. 13(b), the width of the boxes for 2-bit Barker code varies. Moreover, as can be seen, the median for the same boxes significantly differs between the slow and fast corners and the typical corner. It should be noted that the size of box plots is similar for most sequences. However, medians differ considerably for short period sequences on both test chips.

Experimental results demonstrate that short period sequences are not suitable for embedded power watermarking, due to high variance of results and strong sensitivity to PV. Therefore, results are non-deterministic and the expected detection performance cannot be estimated.

B. Detectability

To determine the detection performance, the Null Hypothesis Significance Test [36] was performed on FPGA measurements. The 5% threshold levels were found for each watermark sequence, when a watermark signal was not present. Furthermore, the thresholds were applied to the results in Fig. 12 and the average null hypothesis rejection ratio was found, Fig. 14. We focused on deterministic watermark sequences, as discussed in Section VIII-A. Results in Fig. 14, match the simulation results of Section VI, Fig. 8. Longer period watermark sequences such as 12-bit m-sequence approach the threshold level much faster than shorter sequences, such as 7 and 11 bits Barker codes. Therefore, it is possible to reduce the area and power overheads with shorter watermark sequences, through reduction of WPPG registers.

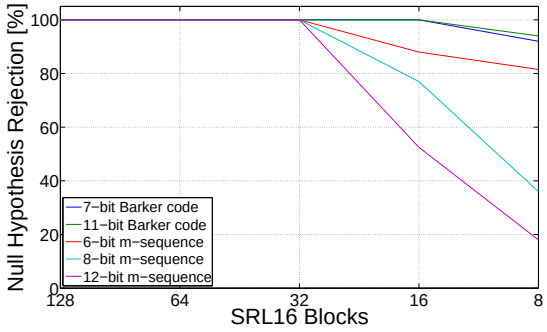


Fig. 14. Null Hypothesis Significance Test of watermark sequences implemented on FPGA, with 5% threshold level.

C. Area and Power Overheads

Minimization of area and power overheads is one of the major factors of all power watermarks implemented on embedded processors. In Section VI-C, various watermark sequences were simulated and it was shown that shorter sequences produce higher null hypothesis rejection ratio than longer sequences, for the same noise power. The theory in Section IV and simulation results of Section VI have been validated on FPGA and test chips. Experimental results from the FPGA in Section VIII-B, demonstrated that shorter period watermark sequences, such as 7 and 11 bits Barker codes, achieve the null hypothesis rejection ratio close to 95%, when the number of SRL16 blocks for WPPG is 8. To achieve the similar detectability with the 12-bit m-sequence, 32 SRL16 blocks must be used. Therefore, shorter Barker codes enable area overhead reduction of approximately 75%, by reducing the number of WPPG registers. To estimate the power reduction, the watermark circuits were synthesized using $65nm$[1] technology library. The fully placed and routed watermark circuit netlist, embedded in chip I, was simulated using Synopsys VCS, and a value change dump (VCD) file was created from the switching activity of the circuit. The estimate of the power consumption was obtained with Synopsys Primetime-PX, using the VCD file obtained from simulations. Results are shown in Table IV. The size of the WPPG circuit was varied, while keeping the 75% ratio between sequences. As the size of the WPPG is reduced, the 7-bit Barker code enables greater area and static power minimization, when compared to the 11-bit Barker

[1]TSMC $65nm$ low leakage technology library.

TABLE IV
AREA AND POWER REDUCTION IN ASIC

Watermark Sequence	WPPG Registers	Area Reduction	65nm		
			P_{DYN} Reduction	P_{STATIC} Reduction	P_{TOTAL} Reduction
7-bit Barker code	512	74.9%	73.2%	74.8%	73.3%
11-bit Barker code	512	74.8%	75.3%	74.8%	75.2%
12-bit m-sequence	2048	-	-	-	-
7-bit Barker code	256	74.8%	74.5%	75.7%	74.5%
11-bit Barker code	256	74.5%	75.5%	73.9%	75.5%
12-bit m-sequence	1024	-	-	-	-
7-bit Barker code	128	74.7%	75.8%	75.7%	75.8%
11-bit Barker code	128	74.3%	77.7%	75.3%	77.6%
12-bit m-sequence	512	-	-	-	-
7-bit Barker code	64	74.3%	76%	75.8%	75.9%
11-bit Barker code	64	73.4%	77.4%	75%	77.3%
12-bit m-sequence	256	-	-	-	-
7-bit Barker code	32	73.4%	72.7%	73.6%	72.7%
11-bit Barker code	32	71.7%	73.1%	71.8%	73.1%
12-bit m-sequence	128	-	-	-	-

code. The 7-bit Barker code requires 7 registers, while 11-bit Barker code requires 11 registers. However, since the *activity factor*, α, of the 11-bit Barker code is lower by 12% (Table II), it consumes less total power for all WPPG sizes. Furthermore, as can be seen the total power reduction of at least 73% is achieved when using short watermark sequences, such as 7 or 11 bits Barker codes, instead of longer m-sequences, due to lower implementation requirements of the WPPG circuit. The reason for this is Eq. (10) and Eq. (12) shown in Section IV.

IX. SECURE DIGITAL SIGNATURE

The watermarks discussed in previous sections transmit a single bit of information, to determine the presence of an IP. The watermark implementation followed [16]–[18], to establish the influence of sequence parameters on hardware implementation costs and detection performance. However, as the watermark can only be regarded as found or not, the IP candidates must be short listed for more thorough investigation. Therefore, the digital signature, such as author of a core, serial number or license agreement is not conveyed and anyone can claim an ownership, once he detects a watermark in a system [41]. Furthermore, as discussed in Section XI, short period sequences are more vulnerable to various types of attacks. This includes both Barker codes and short period m-sequences.

To overcome the limitations of such short sequences and generate a digital signature, the private/public key encryption and the cryptographic hash functions, such as $MD5$ [42], can be used as in [5], [13], [21]–[23], [27]. However, as the encryption and the cryptographic hash functions are used, the encoded signatures vary with conveyed messages. Hence, the power pattern parameters, such as α and θ_{MAX}, change along with the implementation costs, to provide a high detection performance, Fig. 8. To ensure the most cost-efficient parameters are utilized, the encoded signature can be generated as in [41]. In Fig. 15(a), the implementation algorithm is shown. The digital signature ("Cortex-M0") is encrypted with a private key, known only to the IP vendor. To reduce the length of the output bitstream, the encrypted message is later encoded using the cryptographic hash function (MD5). Furthermore, the hash encrypted bit sequence is used to modulate the cost-efficient sequence. In Fig. 15(a), the 7-bit Barker code is used for illustration purposes. To generate bit $'1'$, a full period of a 7-bit Barker code is used. To generate bit $'0'$, the inverse

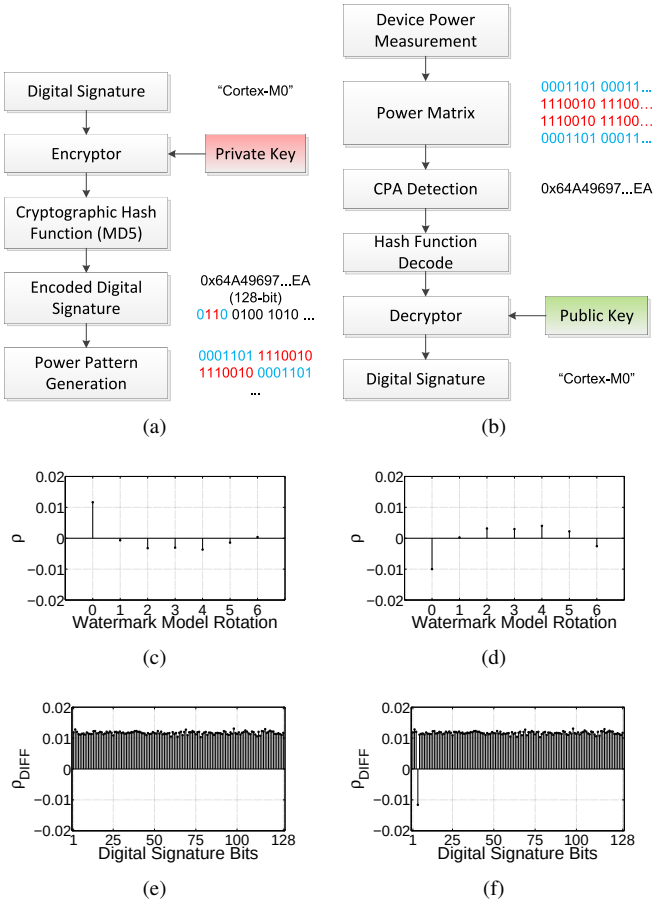


Fig. 15. Implementation (a) and detection (b) diagrams of secure digital signature. Detection of a correct (c) and an incorrect (d) signature bit. Correct (e) and an incorrect (f) detection of a digital signature.

of a sequence is used. The inverted sequence demonstrates different α and θ_{MAX} parameters. However, the parameters complement each other (Fig. 3(b)) and similar ρ_{PEAK} and ρ_{DIFF} results are expected, when compared with the non-inverted sequence. In such way, the highly robust digital signature is generated. The detection algorithm is shown in Fig. 15(b). The device power signal (trace) is measured with an oscilloscope. The power matrix is created by dividing a power trace, such that each signature bit corresponds to a specific trace. The Correlation Power Analysis is applied to each trace separately and the correlation spectra, ρ_{PEAK} and ρ_{DIFF} are found. To demonstrate the use of such algorithm, we simulated the digital signature of Fig. 15(a), and introduced the normally distributed noise of $32dB$, as in Section IV. The size of the obtained power matrix was $128\ x\ 300,000$ clock cycles. When a watermark model for a particular signature bit is correct, a high positive correlation peak can be noticed, Fig. 15(c). Otherwise, when a model is incorrect and represents the inverted sequence, a high negative correlation peak is seen, Fig. 15(d). Furthermore, if a model of another sequence was used or the data was not properly arranged, the correlation value would be close to 0. Finally, ρ_{DIFF} corresponding to all signature bits are plotted, Fig. 15(e). In case ρ_{DIFF} have similar positive values, the correct hash encoded sequence is

considered as found and can be further decoded and decrypted using the public key. In Fig. 15(f), an error bit was introduced at the 4^{th} bit of a hashed sequence. As can be seen, the negative ρ_{DIFF} peak occurs and the error bit is detected. This means that the detected signature does not match the expected signature and the IP differs from the expected IP.

The use of the private/public key and the cryptographic hash functions ensures the embedded power signature is highly robust. Although the attacker may not know the generation scheme of the digital signature, due to the limited number of signature combinations when short sequences, such as Barker codes are considered, it is feasible for an attacker to record all power data and use the brute force attack, to reverse engineer the hash encrypted message. Nevertheless, if such occurs the IP vendor's private key remains uncompromised [5], and it can still be used to prove the IP infringement. Additionally, if an attacker obtains the RTL a digital simulation of the design can be performed to check if any registers follow a Barker code. Since, the Barker code is public and there are only a few codes that can be used, this approach is also feasible. As all Barker codes are short and require only few clock cycles for the generation, the number of false alarms caused by other registers switching in the similar pattern increases with the system size. Moreover, the attacker would have to inspect most of such occurrences which may become time consuming for large designs. The methodology, Fig. 15, demonstrates that it is computationally infeasible for an attacker to forge an owner's signature and provide stronger evidence of an IP ownership. However, as discussed in Section XI-A, an attacker can tamper with an owner's signature and change the sign of one or more correlation results. To prevent such an attack a new trigger-based watermark generation methodology is proposed in Section X.

The proposed methodology, Fig. 15, is however impractical in case of the longer m-sequences. Although it is possible to hash encode the signature with the longer m-sequence as in [18], the α and θ_{MAX} parameters would change and larger WPPG circuit would be required. In Fig. 15, due to the modulation of the watermark sequence with the hash encoded bit sequence the original parameters of a watermark sequence are retained. Nevertheless, the proposed approach requires an additional circuitry to implement the key encrypted and hash reduced sequence (Fig. 15). In a typical implementation, an extra 128-bit shift register would be required to hold the hash value and a state machine would have to be implemented to achieve the desired modulation. In an FPGA, such a shift register requires 8 LUTs, configured as 16-bit shift registers (SRL16). Although the basic state machine with only few states would be sufficient, the final hardware implementation would approximate the m-sequence approach (Table III). Furthermore, an ASIC implementation would require the entire 128 registers to be implemented.

In this section, the secure digital signature approach was proposed for short period watermark sequences. As it will be discussed further in Section XI longer period m-sequences are robust against various types of attacks but require bigger area to implement significant size WPPG circuit. Shorter sequences offer a reduced area and power overheads but are not as robust as longer period m-sequences. The robustness of shorter sequences can be improved with the approach demonstrated in Fig. 15 but the area overhead gains vanish. The tradeoff between longer m-sequences and shorter sequences occurs and the watermark implementation must be reconsidered for various types and sizes of systems.

X. Application Specific Watermark Implementation

In small processors, such as microcontrollers (e.g. ARM Cortex-M0), the area overhead of the secure short sequence implementation (Fig. 15) may be excessive. Since, a small WPPG is sufficient to generate a strong enough watermark power consumption, the longer m-sequence approach [18] may be a better solution. In bigger processors, such as application processors (i.e. ARM Cortex-A9), the WGC circuit has negligible impact and the WPPG size is the main factor. Since the WPPG size increases relatively linearly with the system size, the area overhead of the WPPG circuit for longer m-sequence would certainly be larger than the area overhead of the secure short sequence implementation. Therefore, the use of encoded short sequences is expected to be more suitable, since it allows both area and power overhead minimization through a reduction of the WPPG circuit implementation.

Furthermore, in embedded systems the area and power overheads are often prioritized and it is not viable to generate the watermark power signal at all times. In such systems, the watermark is required to be active non-deterministically and for a short period of time. To ensure such operation, the watermark circuit activation time can be modulated with a specific system instruction, to increase the attacker's effort and computational time of the simulation. Since an attacker must know when a watermark sequence is active, finding the activation time without a full knowledge of a system architecture is infeasible. If an attacker obtains a power signal, the watermark signal may be to weak to be found or an erroneous correlation peaks may be generated, due to incorrect assumptions of the implemented architecture. When an IP owner tries to extract the embedded watermark pattern, it uses a special trigger to combine multiple power acquisitions into a single trace, where a watermark pattern is continuous. Such a trigger is however not known to an attacker and will significantly increase the effort required for a successful attack. Additionally, the secure implementation of short sequences can be performed as in [39], to significantly reduce the area and power overheads. The visibility of an overridden clock enable signal due to watermark circuit can be kept to minimum since a simple XOR gate would ensure the clock gate is modulated according to a watermark sequence. This also ensures that if an attacker embeds his own "always-ON" watermark they may violate the original area and power specification, which is easily detectable. Furthermore, the watermark embedded by an attacker can be of much lower amplitude, since they would use the WPPG circuit to achieve the desired watermark power consumption. The IP owner instead would use the original processor to emulate the WPPG circuit. This minimizes the occurrence of error bits with implementation in Fig. 15. If an

attacker wishes to understand the watermark implementation they would need to re-simulate the entire RTL to understand the watermark activation scheme, which is not a trivial task.

XI. IP Attacks and Robustness

In this section, the security of watermarks in the non-triggered and trigger-based implementations are compared against various types of third party IP attacks. The security, commonly known as the robustness, is determined by the attacks a watermark is able to withstand and the effort of an attacker. As it is difficult to quantify the robustness of side-channel watermarks [41], this section discusses it in relative terms [18]. In the classical cryptographic scenario an attacker aims to retrieve a secret key. If such occurs, the security of a system is breached and allows an attacker to extract sensitive information. In case of IP watermarks, the system's security is not the aim of an attack, but the legal rights to an IP. This section discusses attacks against watermarks and focuses on the most prominent approaches, such as tampering, finding ghosts and forging. These are illustrated using the commonly used "Alice and Bob" scenario, often used in Cryptography, where "Alice" and "Bob" denote two individuals at either end of a communications channel, with cryptographic techniques applied to ensure their conversation is secret.

A. Tampering

Bob (attacker) can tamper with Alice's (IP owner) solution, by removing Alice's signature (watermark) and adding own signature. Due to the transparent nature of the RTL description and unprotected design files provided to the SoC integrators, Bob has virtually unlimited access to a design. Therefore, it is not possible to prevent Bob from adding own watermark. However, it is crucial that Alice's watermark circuit is hidden, such that Bob cannot easily find it.

In the non-triggered implementation the use of longer m-sequences is a more appropriate solution. Shorter sequences, such as Barker codes and short m-sequences, require an additional circuitry to increase the security capabilities. Nevertheless, the algorithm (Fig. 15) does not provide complete protection against tampering attacks, since an attacker only needs to change the sign of the correlation and not remove the entire correlation. Also, such an approach is only feasible in bigger application processors due to the vanishing effect of the area overhead reduction offered by shorter sequences.

If one however considers a secure short sequence approach in the trigger-based implementation (Section X), the robustness against tampering attacks can be significantly improved. The shortfalls of the secure algorithm identified earlier are overcome by the trigger-based watermark generation. Furthermore, in bigger application processors the area overhead is negligible due to removal of the WPPG circuit.

B. Finding Ghosts and Forging

Bob can attempt to find a ghost signature, such as specific power pattern, and claim that an IP contains his own watermark. Furthermore, Bob can forge Alice's implementation and watermark other solutions, which do not belong to Alice. In such case, Bob demonstrates that Alice's signature is not genuine since it can be found in another IP.

In the non-triggered implementation, the robustness of sequences against such attacks is limited to their intrinsic characteristics. In the case of the commonly used m-sequence, the robustness increases with the number of registers used for WGC. For example, a 32-bit m-sequence is more robust than 12-bit m-sequence, since it contains more frequency components (Fig. 6(c)). Therefore, it increases the number of watermark combinations and the amount of transmitted information. Nevertheless, the detection performance of longer m-sequences must be complemented by increasing the number of WPPG registers, which has the direct impact on the watermark robustness against tampering attacks.

In the trigger-based implementation, the use of the secure short sequences is more suitable due to the watermark generation algorithm. It is very hard for an attacker to detect a watermark without knowing the architecture of the processor core (Section X) and the modulation of the entire (or most) processor core prevents an attacker from generating a stronger signature.

XII. Conclusions

The goal of this work was to achieve a design method for incorporating sequence-aware watermarks in soft IP Embedded Processors. Using a new theoretical definition, the relationship between the watermark sequence parameters and detection performance has been illustrated and validated with simulations and experimental results of FPGA and ASIC designs of embedded processors. It has been shown that the tradeoffs occur between shorter and longer sequences, in terms of hardware implementation costs and robustness against third party attacks. The tradeoffs have been analyzed and it has been concluded that for smaller systems the commonly used long m-sequence approach is a better solution due to its robustness against third party attacks. However, in bigger systems the trigger-based secure short sequences achieve better hardware implementation costs without sacrificing the robustness performance.

Acknowledgment

The authors would like to thank EuroPractice Mini-ASIC program for silicon fabrication and packaging, Prof. Steve Gunn for his insightful comments, Dr. Jatin Mistry, James Myers, Prof. David Flynn and Anand Savanth for their help in fabricating and testing the silicon test chips, and Dr. Sheng Yang for constructive discussions.

References

[1] G.E. Moore. Cramming More Components onto Integrated Circuits. *Electronics*, 38(8):114–117, April 1965.
[2] VSI Alliance. VSI Alliance Architecture Document: Version 1.0, 1997.
[3] VSI Alliance. Intellectual Property Protection: Schemes, Alternatives and Discussion, Aug 2001.
[4] R. Torrance et al. The State-of-the-Art in IC Reverse Engineering. In *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 363–381. 2009.

[5] A.B. Kahng et al. Constraint-Based Watermarking Techniques for Design IP Protection. *TCAD*, 20(10):1236–1252, Oct 2001.

[6] M. Ni et al. Constraint-Based Watermarking Technique for Hard IP Core Protection in Physical Layout Design Level. In *ICSICT*, pages 1360–1363, 2004.

[7] X. Cai et al. A Watermarking Technique for Hard IP Protection in Post-Layout Design Level. In *ASICON*, pages 1317–1320, Oct 2007.

[8] N. Narayan et al. IP Protection for VLSI Designs via Watermarking of Routes. In *ASIC/SOC*, pages 406–410, Sep 2001.

[9] T. Nie et al. A Post Layout Watermarking Method for IP Protection. In *ISCAS*, pages 6206–6209, 2005.

[10] Y. Du et al. IP protection platform based on watermarking technique. In *ISQED*, pages 287–290, Mar 2009.

[11] G. Qu. Publicly Detectable Techniques for the Protection of Virtual Components. In *DAC*, pages 474–479, 2001.

[12] A.E. Caldwell et al. Effective Iterative Techniques for Fingerprinting Design IP. *TCAD*, 23(2):208–215, Feb 2004.

[13] G. Qu. Publicly Detectable Watermarking for Intellectual Property Authentication in VLSI Design. *TCAD*, 21(11):1363–1368, Nov 2002.

[14] F. Koushanfar et al. Behavioral Synthesis Techniques for Intellectual Property Protection. *TODAES*, 10(3):523–545, July 2005.

[15] D. Kirovski et al. Local Watermarks: Methodology and Application to Behavioral Synthesis. *TCAD*, 22(9):1277–1283, Nov 2003.

[16] D. Ziener et al. FPGA Core Watermarking Based on Power Signature Analysis. In *FPT*, pages 205–212, Dec 2006.

[17] D. Ziener et al. Power Signature Watermarking of IP Cores for FPGAs. *Journal of Signal Processing Systems*, 51:123–136, Apr 2008.

[18] G. Becker et al. Side-Channel Based Watermarks for Integrated Circuits. In *HOST*, pages 30–35, Jun 2010.

[19] J. Mikulka et al. CCK and Barker Coding Implementation in IEEE 802.11b Standard. In *Radioelektronika*, pages 1–4, 2007.

[20] X. Chen et al. A New Algorithm to Optimize Barker Code Sidelobe Suppression Filters. *TAES*, 26(4):673–677, 1990.

[21] I. Torunoglu et al. Watermarking-Based Copyright Protection of Sequential Functions. *JSSC*, 35(3):434–440, Mar 2000.

[22] E. Charbon et al. Watermarking Techniques for Electronic Circuit Design. volume 2613 of *Lecture Notes in Computer Science*, pages 147–169. 2003.

[23] A. Abdel-Hamid et al. A Public-Key Watermarking Technique for IP Designs. In *DATE*, volume 1, pages 330–335, Mar 2005.

[24] A. Cui et al. A Hybrid Watermarking Scheme for Sequential Functions. In *ISCAS*, pages 2333–2336, May 2011.

[25] A. Abdel-Hamid et al. Fragile IP Watermarking Techniques. In *AHS*, pages 513–519, Jun 2008.

[26] A. Cui et al. A Robust FSM Watermarking Scheme for IP Protection of Sequential Circuit Design. *TCAD*, 30(5):678–690, May 2011.

[27] E. Castillo et al. IPP@HDL: Efficient Intellectual Property Protection Scheme for IP Cores. *TVLSI*, 15(5):578–591, May 2007.

[28] L. Parrilla et al. Protection of Microprocessor-Based Cores for FPL Devices. In *SPL*, pages 15–20, Mar 2010.

[29] J.J. Quisquater et al. ElectroMagnetic Analysis (EMA): Measures and Counter-measures for Smart Cards. In *E-smart*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. 2001.

[30] L. Sauvage et al. Electromagnetic Radiations of FPGAs: High Spatial Resolution Cartography and Attack on a Cryptographic Module. *TRETS*, 2(1):1–24, Mar 2009.

[31] E. Brier et al. Correlation Power Analysis With a Leakage Model. In *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 135–152. 2004.

[32] X. Zhang et al. RON: An On-Chip Ring Oscillator Network For Hardware Trojan Detection. In *DATE*, pages 1–6, Mar 2011.

[33] J. Aarestad et al. Detecting Trojans Through Leakage Current Analysis Using Multiple Supply Pad I_{DDQs}. *TIFS*, 5(4):893–904, Dec 2010.

[34] S. Narasimhan et al. Hardware Trojan Detection by Multiple-Parameter Side-Channel Analysis. *IEEE Transactions on Computers*, 62(11):2183–2195, Aug 2012.

[35] M. Keating et al. *Low Power Methodology Manual: For System-on-Chip Design*. Springer, 2007.

[36] J. Goodwin et al. Power analysis detectable watermarks for protecting intellectual property. In *ISCAS*, pages 2342–2345, Mar 2010.

[37] P. Alfke. Efficient Shift Registers, LFSR Counters, and Long Pseudo-Random Sequence Generators. Tech. rep., Xilinx, July 1996. xAPP052.

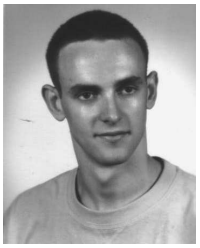
[38] R.H. Barker. Group Synchronizing of Binary Digital Sequences. In *Communication Theory*, pages 273–287. 1953.

[39] J. Kufel, P. Wilson, S. Hill, B.M. Al-Hashimi, P.N. Whatmough, and J. Myers. Clock-modulation based watermark for protection of embedded processors. In *DATE*, pages 1–6, March 2014.

[40] R. York. Benchmarking in Context: Dhrystone. ARM, White Paper, Mar 2002.

[41] G.T. Becker et al. Detecting Software Theft in Embedded Systems: A Side-Channel Approach. *TIFS*, 7(4):1144–1154, 2012.

[42] R.L. Rivest. RFC 1321: The MD5 Message-Digest Algorithm. Internet Activities Board, April 1992.



Jedrzej Kufel received the M.Eng. degree (first class Hons.) in Mechatronics and Robotic Systems from the University of Liverpool, in 2010. He is currently pursuing the Ph.D. degree with the School of Electronics and Computer Science, University of Southampton. In 2014, he joined the IoTBU department at ARM Ltd., Cambridge, U.K..

Peter R. Wilson (M'99, SM'06) was born in Edinburgh, Scotland, and received the B.Eng. (Hons.) in Electrical and Electronic Engineering from Heriot-Watt University, Edinburgh, Scotland, in 1988; an M.B.A from the Edinburgh Business School, Scotland in 1999, and Ph.D. from the University of Southampton, England in 2002.

Dr Wilson is currently an Associate Professor in Electronic and Electrical Engineering at the School of Electronics and Computer Science, University of Southampton, UK. His current research interests include modeling of magnetic components in electric circuits, power electronics, renewable energy systems, integrated circuit design, VHDL-AMS modeling and simulation, and the development of electronic design tools.

Stephen Hill is currently ARMs Director of CPU Engineering in the US. Previously he lead ARM CPU Core R&D and before that he was a microarchitect and logic designers working multiple processor generations. He studied Physics at the University of Bristol, UK and microelectronics at the University of Southampton, UK.

Bashir M. Al-Hashimi (M'99-SM'01-F'09) is a Professor of Computer Engineering and Director of the Pervasive Systems Center in University of Southampton, UK. He is ARM Professor of Computer Engineering, and Co-Director of the ARM-ECS research center. His research interests include methods, algorithms and design automation tools for low-power design and test of embedded computing systems.

Paul N. Whatmough received the B.Eng. degree (first class Hons.) in Electronic Communications Engineering from the University of Lancaster, in 2003, the M.Sc. degree (with distinction) in Communications Systems and Signal Processing from the University of Bristol, in 2004, and the Doctorate degree from University College London, in 2012, all in the U.K.

From 2005 to 2008, he held the position of Research Scientist at Philips Research Labs, Redhill, U.K. (which became NXP Semiconductors Research in 2006), focussing on digital radio approaches for multi-standard cellular systems. In 2008, he joined the R&D department at ARM Ltd., Cambridge, U.K., where he is currently Staff Research Engineer. His research interests are in low-power circuits, algorithms and architectures relating to wireless, DSP and embedded computing.