Mitigating EM Side-Channel Attacks with Dynamic Delay Insertion and Data Bus Inversion

Mitigating EM Side-Channel Attacks with Dynamic Delay Insertion and Data Bus Inversion

***Abstract*—Cryptographic circuits are sensitive to electromagnetic (EM) side-channel attacks (SCAs), which aim to detect the EM emissions of these circuits. A novel technique is proposed to mitigate such attacks, by reducing the correlation between the processed data and EM emissions. This objective is achieved by combining energy-efficient data inversion with dynamic delay insertion. The added delay enhances the immunity against EM attacks for the cryptographic circuit without performance degradation and, in specific scenarios, even improves performance. Simulation results on a set of EM traces, captured from an 8-bit interposer-based off-chip memory bus, demonstrate the efficiency of the proposed technique by decreasing SNR below 1 and improving the worst-case bus latency by 9.5%.**

***Index Terms*—Data Bus Inversion (DBI), interposer, switching activity, coupling capacitance, electromagnetic emission, side-channel attack**

I. Introduction

Advanced crytographic algorithms have widely been used in smart devices to encrypt sensitive data. However, these algorithms are all implemented on hardware, where the leaked information from the power supply line, electromagnetic emissions, timing, or acoustic noise can be processed and extract the sensitive information (e.g., the encryption keys). Therefore, the attack that targets the hardware rather than the algorithm is defined as a side-channel attack (SCA). Among these side-channel attacks, correlation power attack (CPA), which exploits the correlation between the sensitive data and power consumption, is generally regarded as a major security vulnerability of the device [1], and more recent CPA techniques show higher effectiveness compared to other SCAs, such as differential power attack (DPA) [2], [3].

To mitigate correlation based SCAs, various techniques have been proposed. Random delay insertion (RDI) is an effective technique to obfuscate the correlation between the data and consumed power, so as to eliminate potential CPAs [4]. However, RDI usually incurs extra latency and degrades circuit performance, which makes this method less appealing.

Recently, a method that introduces a specific delay on the edge lines of an off-chip bus to decrease the correlation coefficient between EM emissions and transmitted data has been proposed [5]. Nonetheless, the static nature of the method is not applicable to diverse buses and types of data. In addition, if the edge lines do not transition, the correlation coefficient is not affected offering no protection in this case.

With the increasing adoption of 2.5-D integration technology, memory and logic components are integrated on the same substrate and communicate through interconnect buses on the interposer [6]. Data Bus Inversion (DBI) technique is typically utilized to decrease the number of transitions on the bus and, hence, reduce the dynamic power [7]. Additionally, DBI can provide benefits on security enhancement for CPAs, and hinder the correlation between transmitted data and related EM emissions [8]. However, as shown in this work, DBI cannot help if the adversary monitors the control bit that indicates data inversion.

Alternatively, security can be enhanced by a dynamic delay insertion technique, where the information leakage is concealed by non-randomly adding delay into bit lines based on the Hamming Distance (HD) of two consecutive pieces of encrypted data. Hence, the delayed lines cannot be identified in order to reverse engineer the method. Furthermore, unlike the previously mentioned RDI technique, dynamic delay insertion does not degrade and, in specific scenarios, improves the bus latency. This advantage is achieved by adding delay into the lines that do not drive the maximum coupling capacitance. Therefore, the new security scheme offers benefits of enhanced immunity against EM attacks for a bus, without degrading the bus performance.

The rest of the paper is organized as follows. The dynamic delay insertion scheme is introduced in Section II and the circuit implementing this delay scheme is presented in Section III. Simulation results are analyzed in Section IV, followed by the conclusions in Section V.

II. Dynamic Delay insertion scheme

A novel dynamic delay insertion scheme to mitigate the EM attacks on off-chip memory buses is proposed in this section. First, the rationale of the delay insertion scheme is explained. The dynamic delay insertion algorithm is demonstrated in four different cases that correspond to all possible transition combinations, encountered when the DBI technique is applied.

A. Rationale of Delay Insertion Scheme

S-box is the only non-linear component in the AES, which typically consists of complex logic units or a Look-up table (LUT). In order to reduce the dynamic power of the cryptographic circuit in 2.5-D integrated systems, an AES module can be implemented by combining a **memory die** (the purple block, a customized LUT based ROM to perform substitution) with a **logic die** (the yellow block), that includes the other AES components except for S-box [9], and the encrypted application as depicted in Fig. 1. When the ROM receives the address to retrieve the corresponding substitution data, the EM leakage ($\mathcal{M}$) generated by the memory bus is

$$\mathcal{M} = \psi\left(PT, k\right) = \psi\left(HD\left(PT, k\right)\right), \tag{1}$$

where PT is the plaintext, k is the sub-key byte used in the first round of AES encryption, and HD means hamming distance. ψ is the function that links the theoretical value with the measured leakage. The assumed EM leakage $\mathcal{H}$ is computed as the HD between PT and k, while the EM traces generated by the bus are captured at the probe terminal as sampled coupled voltages $\mathcal{V}$. The correlation between the measured EM traces and the assumed EM leakage is calculated by,

$$\rho\left(\mathcal{V},\mathcal{H}\right)=\frac{cov\left(\mathcal{V},\mathcal{H}\right)}{\sigma_{\mathcal{V}}\sigma_{\mathcal{H}}}, \tag{2}$$

where σ and cov are the data set standard deviation and covariance between data sets, respectively.

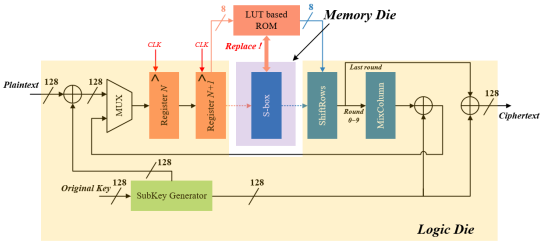


Fig. 1. AES encryption with S-box implemented by an off-chip ROM [9].

Assuming the bus-invert coding technique, proposed in [7] (effectively the origin of DBI), is utilized in the 8-bit interposer-based memory bus in Fig. 1, if the HD between two consecutive pieces of data is higher than four, the data to be transmitted is inverted. The function ψ in (1) thereby no longer maintains the linear relationship with $HD\left(PT,k\right)$. Additionally, as stated in [5], when Δt is added into the bus lines, (2) can be written as,

$$\rho\left(\mathcal{V}_{total},\mathcal{H}\right)\approx\frac{\rho\left(\mathcal{V}_{\max},\mathcal{H}\right)}{\sqrt{1+\frac{1}{SNR}}}\frac{1}{\sqrt{\frac{\delta(\mathcal{V}_{\Delta})}{\delta(\mathcal{V}_{\max})}}}\propto\frac{1}{\sqrt{\delta\left(\mathcal{V}_{\Delta}\right)}}, \tag{3}$$

where $\mathcal{V}_{total}$, $\mathcal{V}_{\max}$, and $\mathcal{V}_{\Delta}$ are the total captured voltage (noise included), maximum coupled voltage, and voltage difference due to Δt shifting, respectively. δ denotes the variance function. SNR (signal-to-noise ratio) is regarded as a security figure of merit in this work, defined as $\rho_r/\rho_{\max,w}$, where ρ_r is the correlation of the right key and $\rho_{\max,w}$ is the maximum correlation of the wrong key. When SNR drops below 1 in the simulation environment, a system is considered to mitigate SCAs in real-world scenarios with a high probability [10].

Next, the effect of added delay on bus latency is analyzed. As described in [11], the bit lines are modeled as *RC* interconnects, where the bus latency (T) is proportional to the wire resistance (R_t) and capacitance (C_t) (comprises the ground capacitance (C_g) and coupling capacitance (C_c)), as shown in Fig. 2. C_t varies when bit lines transition, according to,

$$C_t=C_g+C_c\left|\frac{\Delta V_1}{V}\right|+C_c\left|\frac{\Delta V_2}{V}\right|=C_g+nC_c, \tag{4}$$

where $n\in[0,4]$, C_L, R_{drive} are, respectively, the load capacitance and on-resistance of the driver, V is the supply voltage, and ΔV_1, ΔV_2 are the voltage difference between the observed line with its two neighbouring lines, respectively. The bus performance is determined by the line that drives the maximum capacitance, e.g., as shown in Fig. 2, when line I_1 transitions in opposite direction than I_0, I_2, exhibits a coupling capacitance equal to $4C_c$. When Δt is added into lines I_0 and I_2, the coupling capacitance of I_1 is less than $4C_c$. Therefore, the latency of I_1 can be reduced.

B. Delay Insertion Algorithm

For an 8-bit bus where DBI technique is applied as considered in this work, there can be a maximum of four transitions for any piece of data. Indeed, if there were more than four ($n/2$, where n is the bus width), for example, $x>n/2$ transitions, according to the DBI method [7], the bus is inverted leading to $(n-x)<n/2$ transitions. Based on this observation, the following four cases list all possible transitions on the bus. To better describe these transitions, the bus model shown in Fig. 2 is used, with the bus lines annotated with indices ($I_0\sim I_7$).

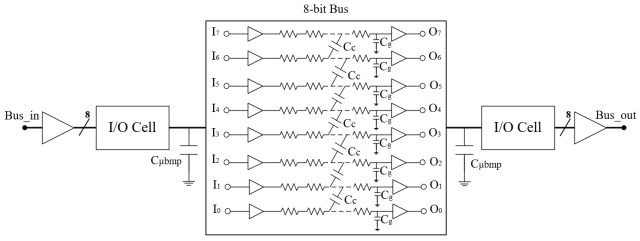


Fig. 2. 8-bit interposer-based interconnect model.

- ***Case 1***: up to four individual transitions (which appear alternately on the 8-bit bus), e.g., lines I_1, I_3, I_5 and I_7 transition. All but lines I_0 and I_7 drive a capacitance of C_g+2C_c. Lines I_0 and I_7 drive a capacitance of C_g+C_c if they transition.
- ***Case 2***: up to two pairs of transitions, e.g., lines I_2, I_3, I_5 and I_6. If the transition is in the same direction, then the lines drive a capacitance of C_g+C_c while for transitions in the opposite direction, the lines drive a capacitance of C_g+3C_c.
- ***Case 3***: three consecutive transitions and one individual transition, e.g., lines I_4, I_5, I_6 and I_2. If lines I_4, I_5, I_6 switch to opposite directions, the worst-case latency (d_{max}) of the bus is produced due to the maximum capacitance C_g+4C_c driven by line I_5.
- ***Case 4***: four consecutive transitions, e.g., lines I_3, I_4, I_5 and I_6. If they switch to opposite directions, lines I_4 and I_5 exhibit the worst-case latency (d_{max}) of the bus proportional to the maximum capacitance C_g+4C_c as also encountered in the previous case.

The pseudocode of the delay insertion algorithm, listed as Algorithm 1, is explained through the specific examples mentioned in cases 1 to 4. Note that since DBI is applied to the bus, there is a XOR operation between the current and upcoming value of bit line x, denoted as I_{x_t} and $I_{x_{t+1}}$, respectively, where the XOR output is denoted as XOR_x. If line x transitions, $XOR_x=1$, otherwise $XOR_x=0$.

First, bit lines I_1~I_6 are considered. In case 1, $XOR_{I_1} = 1$ and $XOR_{I_2} = 0$ (steps 3 and 4), hence Δt is added to line I_1 and to lines I_3, I_5, I_7 as the driven capacitance is $C_g + 2C_c$. In case 2, $XOR_{I_2} = 1$, $XOR_{I_3} = 1$ and $XOR_{I_1} = 0$ (steps 5 and 6), Δt is added to line I_2 and similarly to line I_5 as the inserted delay reduces the driven capacitance to less than C_g+3C_c. In case 3, $XOR_{I_4} = 1$, $XOR_{I_5} = 1$ and $XOR_{I_3} = 0$, Δt is added to line I_4 and similarly to line I_6. For line I_5, $XOR_{I_5} = 1$, $XOR_{I_6} = 1$ and $XOR_{I_4} = 1$, hence no delay is added to line I_5 (steps 7 and 8). Thus, the worst-case latency of the bus d_{max} (latency of line I_5) decreases due to the reduction of the coupling capacitance. Lastly, in case 4, Δt is also selectively added to lines I_3 and I_6 (steps 5 and 6) but not to lines I_4 and I_5 (steps 7 and 8), where the worst-case latency of the bus d_{max} (latency of lines I_4, I_5) drops due to the reduction of the coupling capacitance.

For edge lines I_0 and I_7 (which drive at most a capacitance of $C_g + 2C_c$) whenever a transition occurs, Δt is inserted into these two lines. Note that the original latency of the lines increased by Δt, remains lower than the reduced d_{max}.

Succinctly, the target is to add delays to specific lines in cases 3 and 4 such that bus latency decreases and the correlation coefficient drops. For cases 1 and 2, minimum latency is not the target, rather the aim is to respect the reduced worst-case latency d_{max}.

Algorithm 1. Dynamic Delay Insertion Algorithm

Input: the XOR result XOR_x between the current data I_{x_t} and upcoming data $I_{x_{t+1}}$ for bit line x from the DBI circuit;
for $x \in [1, 6]$ **do**
  if XOR_x = 1 and XOR_{x+1} = 0 **then**
    Insert Δt to line x
  else if XOR_x, XOR_{x+1} = 1, and XOR_{x-1} = 0 **then**
    Insert Δt to line x
  else
    Do not insert any delay
  end if
end for
if $x = 0$ and XOR_0 = 1 **then**
  Insert Δt to line 0
else if $x = 7$ and XOR_7 = 1 **then**
  Insert Δt to line 7
else
  Do not insert any delay
end if

This algorithm can be extended to mitigate EM attacks on wider buses. For the case of 8-bit bus, the algorithm leads to improvement of the worst-case latency, determined by a bit line driving the maximum capacitance of $C_g + 4C_c$. For wider buses, the algorithm can be adapted to insert delays only whenever bit lines switch according to cases 1 to 4. In other words, no more than four consecutive bit lines transition. With this approach, the bus latency does not degrade.

III. Circuit Implementation of Delay Scheme

In this section, the low-overhead circuit that generates the required delay is described, followed by the performance evaluation of the delay insertion mechanism.

The data to be sent (off-chip) to the S-box (the LUT based ROM in Fig. 1) are assumed to be launched from a register and are driven by a chain of buffers, as shown in Fig. 3, where the number and size of the buffers can be chosen to satisfy the timing constraints of the bus. The delay is generated by modifying the design of BUF1 (red circle), as depicted in the inset of Fig. 3, where the devices of INV0, and M_4~M_7 generate Δt [12]. The delay control signal $\bar{S}_x$ for line x is determined by XOR_x, XOR_{x-1}, and XOR_{x+1}, already available from DBI and the three gates shown in the figure.

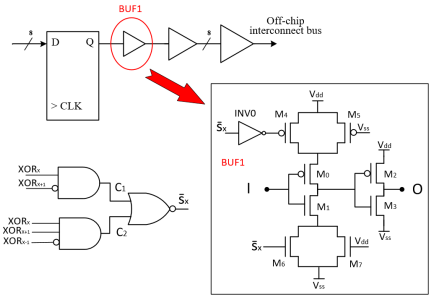


Fig. 3. Circuit implementation of the proposed delay mechanism.

The first inverter in BUF1 is, respectively, pulled high through M_4, M_5, and M_0 (M_5 and M_0) and low through M_6, M_7, and M_1 (M_7 and M_1) if $\bar{S}_x$ is high (low). Thus, the delay of the inverter increases by Δt, as determined by the size of M_4~M_7, when $\bar{S}_x$ = '0'.

At each clock edge, data I_{x_t} is propagated and $\bar{S}_x$ for $I_{x_{t+1}}$ is evaluated. Thus, the correct operation of the circuit is guaranteed only if the delay for generating signal $\bar{S}_x$ is sufficiently higher than the delay required for data I_{x_t} to propagate through the register (t_{CtoQ}) and BUF1. The circuit that generates $\bar{S}_x$ includes a delay path with XOR, AND, NOR, and t_{CtoQ}. The simulated arrival time of $\bar{S}_x$ for three corners, the typical (TT), fast-fast (FF), slow-slow (SS), is, respectively, 110 ps, 94 ps, and 158 ps longer than the data propagating through the register and BUF1.

The delay circuit is simulated for a UMC 65 nm technology [14], and induces a low hardware area cost of about 160 MOS transistors (INV0, M_4~M_7, 2-input AND, 3-input AND, and 2-bit NOR for each bit). Meanwhile, as the calculation of XOR is included in the DBI circuit, the power does not considerably increase either, where the consumed power is $\sim$104 μW, compared with a total power of 138 mW for the CMOS AES circuit [15].

IV. Simulation Results

The effectiveness of this novel delay insertion scheme is explored in this section. Firstly, the EM side-channel attack setting is described. Then, the total bus latency and EM attack results on the 8-bit bus are compared, where no protection, DBI method, and DBI combined with dynamic delay insertion method are, respectively, applied.

A. EM Side-Channel Attack Setup

The targeted off-chip memory bus is assumed to be implemented in the top metal layer of the interposer redistribution layers in a 2.5-D system, where the bus parameters are listed in Table I (bus length is 2 mm). A rectangular coil with a 100 μm length and 50 μm width, which can obtain the maximum

variation in EM emissions [13], is placed vertically above the bus. The bus is modeled in *ANSYS HFSS*. The S-parameters of the model are generated, exported from *HFSS*, and are input into *Spectre* for simulation.

TABLE I. *RLC* PARAMETERS OF THE BUS LINES.

R (Ω/mm)	L (nH/mm)	C_g(fF/mm)	C_c(fF/mm)	C_{total}(fF/mm)
30.56	1.64	41.34	157.64	356.62

B. Simulation results

The latency of each bit line is the 50% delay from the output of the register to the far-end of the bus, as shown in Fig. 3, The bus latency is determined by the worst-case bit line latency.

Diverse Δt can be obtained by adjusting the size of transistors in BUF1. By adjusting the width of M_7 and M_5, Δt can reach up to 211 ps and 177 ps, where a bit line transitions from 0 to 1 and 1 to 0, respectively.

The simulated total bus latency with delay insertion for the four cases mentioned in Section II-B is shown in Fig. 4, where the x-axis is the delay Δt added to specific bus lines and the y-axis is the worst-case bus latency. As shown in Fig. 4, for all four cases, if no delay is added, the worst-case latency of the bus d_{max} is 284 ps, determined by the middle line latency in cases 3 and 4 (e.g., line I_5 in case 3). For the specific setup as illustrated in Fig. 4, adding a delay Δt of up to 50 ps for all possible switching scenarios (see cases 1 to 4 in Section II-B), the bus latency reduces to 263 ps.

Based on the premise that Δt is chosen during the design process, such that the bus performance does not degrade, the effect of the delay on mitigating EM attacks is explored.

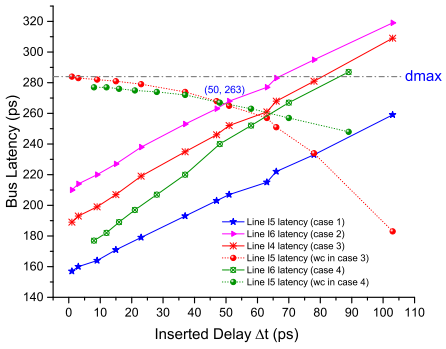
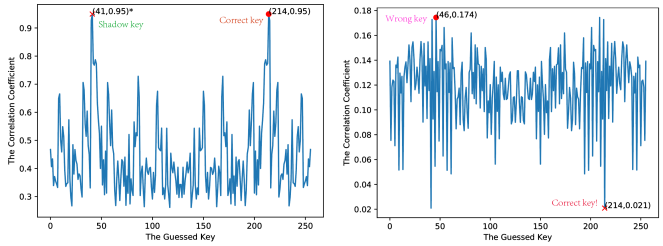


Fig. 4. Bus latency *vs*. delay inserted into specific bus lines.

Firstly, the security benefit from the DBI method is analyzed. 256 EM traces (simulation) are, respectively, collected where the bus is unprotected and with DBI applied. As shown in Fig. 5, for the unprotected bus, the correct key 214 with maximum correlation coefficient 0.95 can be retrieved within 256 traces (50 traces are sufficient to obtain the key and *SNR* is 1.025), while for the bus with DBI applied, the detected key is 46 and *SNR* drops to 0.12, where the correct key 214 is not detected and exhibits a low correlation coefficient.

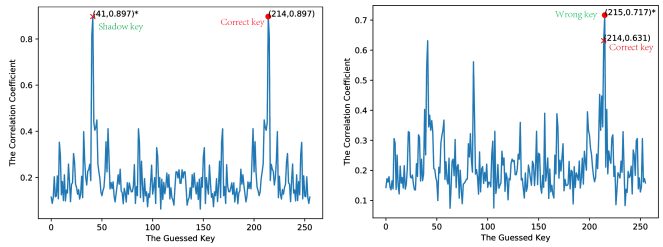
Indeed, DBI can secure the bus against EM attacks. However, the DBI method requires one additional control bit, which determines whether the data to be transmitted should



(a) (b)

Fig. 5. EM attack results after 256 traces for (a) unprotected bus and (b) bus where DBI is applied.

be inverted. If this control bit line is monitored by the attacker (e.g., through another high-resolution probe), the correct key can be obtained within 256 traces (80 traces are sufficient to obtain the correct key 214 and SNR is 1.103), as shown in Fig. 6(a), where the security protection only from DBI does not suffice. To avoid this risk, the dynamic delay insertion technique enhances DBI to provide proper protection against such attacks.



(a) (b)

Fig. 6. EM attack results after 256 traces for (a) bus with DBI applied and (b) bus with dynamic delay insertion plus DBI applied. The control bit is monitored by the attacker in both (a) and (b).

Applying delay insertion combined with DBI to the bus lines, where the control bit line for DBI is assumed to be monitored by the attacker, the EM attack result is illustrated in Fig. 6(b). The correct key is indistinguishable within 256 traces and SNR is 0.88. As depicted in Fig. 4, when Δt equals 50 ps, d_{max} decreases by 9.5% (compared to the scenario with no added delay), offering higher performance in addition to greater security.

V. CONCLUSIONS

A new delay insertion methodology applied to off-chip interconnect buses to mitigate EM attacks without degrading or even improving bus latency is proposed. The core idea is to dynamically insert delay to specific lines to reduce the coupling capacitance of the bit lines for the worst-case switching patterns, combined with the DBI method. As a result, the correlation between the data and EM emissions is significantly reduced. For an off-chip 8-bit interconnect bus scenario, when 50 ps are inserted into appropriate bit lines, the SNR drops below 1 and the worst-case bus latency decreases by 9.5%, demonstrating the usefulness of delay insertion to mitigate EM attacks.

REFERENCES

[1] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, Vol. 51, No. 5, pp. 541–552, Aug. 2002.

[2] M. Tunstall *et al.*, "Correlation Power Analysis of Large Word Sizes," *Proceedings of IET Irish Signals and Systems Conference*, pp. 145–150, 2007.

[3] S. Mangard, E. Oswald, and T. Popp, "Power analysis attacks: Revealing the secrets of smart cards," *Springer-Verlag*, Vol. 31, Jan. 2008.

[4] I. Levi *et al.*, "Data-dependent delays as a barrier against power attacks," *IEEE Transactions on Circuits and Systems I*, Vol. 62, No. 8, pp. 2069–2078, Aug. 2015.

[5] Omitted for blind review.

[6] V. F. Pavlidis, I. Savidis, and E. G. Friedman, *Three-Dimensional integrated circuit design*, 2^{nd} Ed. Morgan Kaufmann Publishers, 2017.

[7] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol 3, No. 1, pp. 49–58, March 1995.

[8] M. A. Vosoughi, L. F. Wang, and S. Köse, "Bus-invert coding as a low-power countermeasure against correlation power analysis attack," *ACM/IEEE International Workshop on System Level Interconnect Prediction*, June 2019.

[9] C. Teegarden, M. Bhargava, and K. Mai, "Side-channel attack resistant ROM-based AES S-Box," *Proceedings of IEEE International Symposium on Hardware-Oriented Security and Trust*, pp. 124–129, Jun. 2010.

[10] I. Levi, A. Fish, and O. Keren, "CPA secured data-dependent delay-assignment methodology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 25, No. 2, pp. 608–620, Feb. 2017.

[11] K. Hirose and H. Yasuura, "A bus delay reduction technique considering crosstalk," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, Vol. 85, No. 1, pp. 24–31, Jan. 2002.

[12] E. Maragkoudaki and V. F. Pavlidis, "Energy-efficient time-based adaptive encoding for off-chip communication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 28, No. 12, pp. 2551–2562, Aug. 2020.

[13] M. Jiang and V. F. Pavlidis, "A Probe Placement Method for Efficient Electromagnetic Attacks," *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, April 2021.

[14] United Microelectronics Corporation UMC, http://umc.com/.

[15] S. A. Lu, Z. Y. Zhang, and M. Papaefthymiou, "1.32 GHz high-throughput charge-recovery AES core with resistance to DPA attacks," *Symposium on VLSI Circuits*, pp. C246–C247, June 2015.