Security Investigations on an Asynchronous PLA Configuration

Petros Oikonomakos, Simon Moore
Computer Laboratory, University of Cambridge
Petros.Oikonomakos@cl.cam.ac.uk

Abstract

We investigate the potential of using asynchronous PLAs in security applications. For this purpose, we borrow a promising PLA structure from the synchronous domain, modify it for asynchronous operation and balance data-dependent spikes on its power profile. We also work towards protection against malicious fault injection, by means of parity prediction.

1 Introduction

Programmable logic arrays (PLAs) provide an attractive alternative to random logic in VLSI systems [1, 2]. The regularity of a PLA layout makes its timing predictable and controllable. This has recently prompted researchers to adopt PLA structures to achieve "Timing Closure by Design", thus reducing design time [3]. We think that the same argument can equally apply to power. Using a PLA one can more easily predict, control and *balance* power consumption, since parasitics are more likely to be "balanced by construction" in a regular structure rather than in a randomly placed and routed standard-cell based design. From the hardware security point of view, this is particularly interesting, since it is expected to contribute to the production of systems with a degree of security against power analysis attacks "by design".

Previous work both in our group [4] and elsewhere [5] has focused on using dual-rail logic for security purposes, since by nature it offers balanced power consumption *and* fault detection capabilities. As an alternative, in this present work, we are seeking balanced consumption in a *single rail* configuration by exploiting the regularity of a PLA structure, while performing *parity prediction* to protect against optical fault injection.

2 The proposed PLA

2.1 Basic structure

Figure 1 shows the basic PLA AND and OR plane cells used in this work. The synchronous version was first presented and analysed in [2]. The only modification we apply to it at this point is using the "Request" signal from the previous asynchronous stage to trigger the "precharge" and "evaluate" phases of the PLA, exactly as the synchronous version would use the global clock. The C-elements implementing 4-phase single-rail handshaking are also shown, together with the required delay element (four inverters). Evidently, the PLA is treated as combinational hardware handling bundled-data coming from an asynchronous latch (not shown). The PLA output is also considered to feed the asynchronous latch of the next logic stage.

In Figure 1, the usual elements of a NOR-NOR PLA can be identified. Indeed, when the PLA "Req_internal" signal is low, the PMOS transistors MP1 and MP3 precharge the AND and OR planes respectively. The parallel NMOS transistors MN2_1 – MN2_n are controlled by the input data lines and implement the AND plane logic function. Similarly, transistors MN4_1 – MN4_m are controlled by the AND plane outputs and

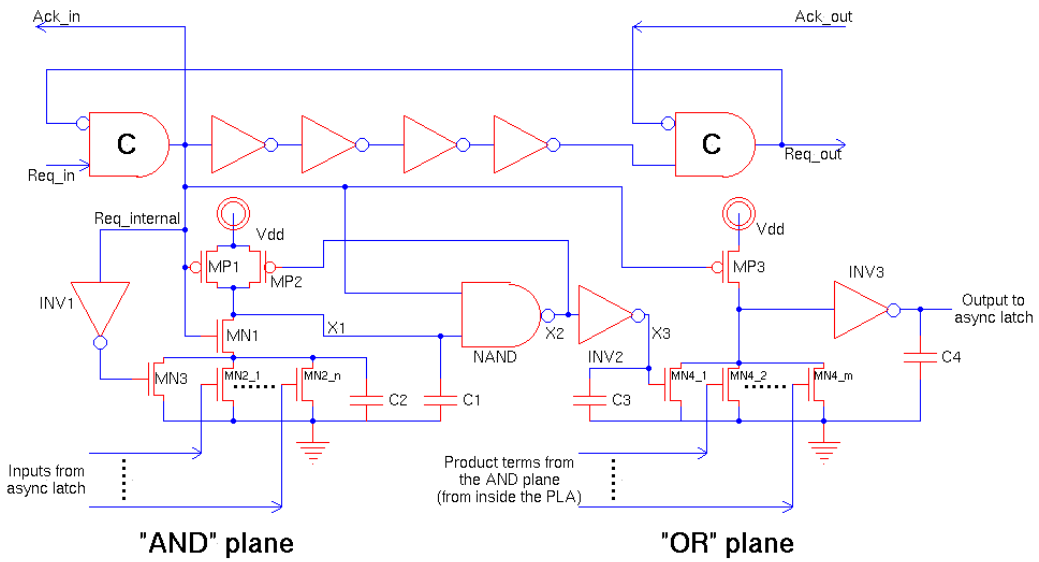

Figure 1: Basic PLA structure based on [2]

perform the OR plane logic function. Capacitors C1 – C4 model parasitics. Notably, the parasitic effects are expected to be very uniform throughout the PLA (e.g. C1 will have the same value in all AND cells), due to regularity.

The design also includes some non-standard elements. First of all, observe that the first inverter of the interplane buffer has been substituted by a NAND gate, ensuring that the voltage in point X2 is the logic inverse of X1 only in the evaluation phase. During precharge, the voltage at X2 is kept high, therefore point X3 is kept low and the need for a ground switch in the OR plane is eliminated. This mechanism both speeds up the OR plane, and saves power, since it minimizes the switching activity in the interplane buffer. It was first proposed in [1]. The second non-standard technique is the charge sharing phenomenon exploited in the AND plane. Notice transistor MN1. It is effectively the ground switch of the AND plane, but it has been moved between the precharge PMOS and the NMOS implementing the function. As soon as Req_internal goes high, capacitor C1 transfers some of its charge to C2 through MN1, no matter what the input pattern is. If any of the MN2_i NMOS is on, then the rest of the charge in C1 will be transfered to ground and X1 will be driven low. The charge sharing effect thus speeds up the discharge process and the overall PLA evaluation phase. If all MN2_i transistors are off, then C1 loses some charge to C2; this charge is replenished when transistor MP2 is turned on, since X2 is driven low. Thus, the design continues operating properly. In the subsequent precharge phase, transistor MN3 turns on and discharges C2.

The PLA prototype of Figure 1 performs very well and, on average, consumes little power. Our simulations, however, have identified one condition under which increased power is consumed and the whole configuration may even not work. In the beginning of the evaluation phase, the NAND gate is fed by two 1s and starts pulling down. If all MN2_i are off, then it will keep on pulling down normally; if one of the MN2_i is on, then X1 should be discharged and the NAND gate pulled up again. This creates a voltage spike and consumes power needlessly. Moreover, if the NAND gate is fast enough for transistor MP2 to be turned on *before* transistors MN2_i have the chance to discharge C1, the result will be a situation where point X1 can never be pulled down. Therefore, in order for the circuit to operate properly, the NAND gate pull-down

transistors should be “slower” than MN1, MN2_i. Alternatively, the minor modification of Figure 2 can solve the problem. The variant of Figure 2 will be referred to as the *delayed structure*, since its only difference from the previous basic structure is that the OR plane is not ordered to evaluate until after two inverter delays. This time is enough for the AND plane to evaluate correctly, and thus the NAND gate can now safely pull down, if needed, at full speed. No power consuming spike will appear at X2 and the circuit will operate properly.

2.2 Security refinements

The PLA paradigm presented in the previous subsection appears to be a very good choice. In order to study it better and, most importantly, evaluate its security characteristics, we conducted a number of HSPICE simulations, on a small example PLA targetting a CMOS VLSI 0.18μm technology. We implemented the following three logic functions on the PLA :

$$\begin{aligned} s &= \overline{a}\overline{b}c + \overline{a}b\overline{c} + a\overline{b}\overline{c} + abc \\ d &= ab + ac + bc \\ p &= \overline{a}c + b\overline{c} + a\overline{b} \end{aligned}$$

Notice that function *p* is effectively the even parity prediction function of *s* and *d*. That is, the overall 3-bit output vector of the PLA will always maintain even parity. Thus, any fault injection attempt corrupting any one of the PLA lines will be detectable at the PLA output, because it will reverse the overall parity. Overall, the considered small PLA has three symmetrical inputs, ten product terms and three outputs ($3 \times 10 \times 3$).

We simulate for four different input combinations, namely (*a*,*b*,*c*) = {(1,1,1),(0,1,1),(0,0,1),(0,0,0)}. Since the inputs are symmetrical, these combinations are enough to illustrate the variations in the power consumption profile of the PLA. We configure and simulate both the basic (Figure 1) and the delayed (Figure 2) design options. Respective power graphs are shown in Figures 3 and 4. The top graph of Figure 3 shows how the Req_in signal changes. All switching activity in the circuit happens at the request input edges, so naturally every experiment produces a pair of power spikes corresponding to the power consumed at the evaluation and precharging phases; these are plotted in the bottom graph of Figure 3 and in the graph of Figure 4. The two leftmost spikes in both cases correspond to the all 1s input, followed by two 1s and a 0, two 0s and a 1 and finally the rightmost pair of power spikes is produced by the all 0s input vector.

The bottom graph of Figure 3 demonstrates very unbalanced power consumption. Indeed, the positive edge spike tops range from 4.43mW (all 1s) to 5.14mW (all 0s), giving a variation of 13.8%. The situation is even worse with the negative edge spikes: they range from 3.39mW (all 1s) down to 1.7mW (all 0s), for a variation of 49.9%. These imbalances originate in the different behaviour of both the AND and the OR planes of the PLA when evaluating a “0” or a “1”. As regards the AND plane, if all inputs of a product term are at “0”, then point X1 in Figure 1 will not be discharged and no power will be consumed at this point. However, the interplane buffer will consume power by pulling X2 low and X3 high. Further, the corresponding OR plane function will discharge and evaluate to a “0”; power will be consumed both during this discharge and at the following inverter INV3 pulling high. At the subsequent precharge phase, all of NAND, INV2, INV3 will switch their values, once more consuming power. Evidently, the situation when a product term produces a “1” is particularly power hungry. In contrast, if at least one of the inputs to an AND plane element is at “1”, then X1 will be discharged and the state of the interplane buffers and the OR plane *should* be unaffected; however, the spikes at point X2 appear and increase power consumption.

Referring back to the logic equations of the implemented functions, one can observe that the number of product terms evaluating to “1” for (*a*,*b*,*c*) = (1,1,1) is maximum (four). When one or two primary inputs are at “1”, then only two product terms give a “1”, while if (*a*,*b*,*c*) = (1,1,1), then *no* AND plane element finds itself

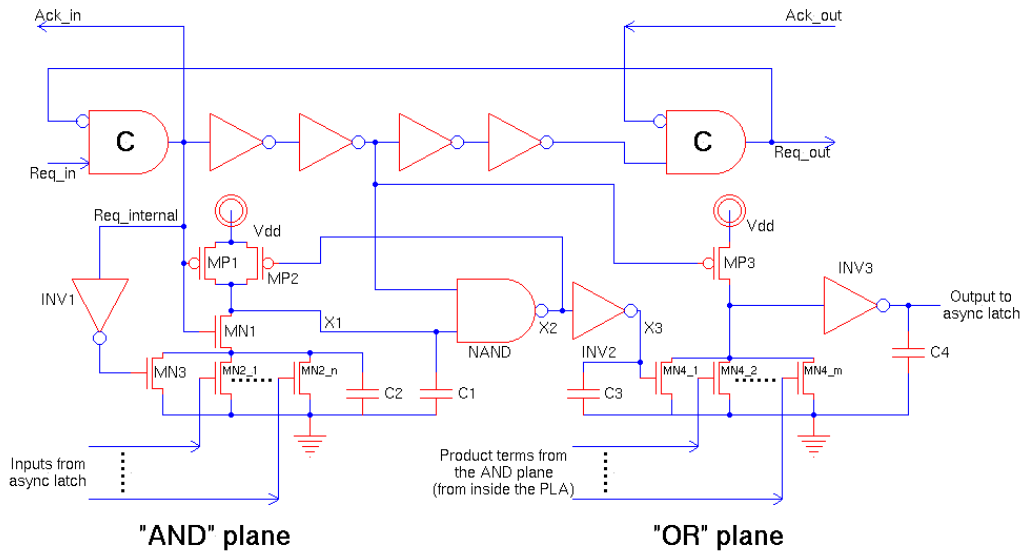

Figure 2: PLA configuration with delayed activation of the OR plane

in a power hungry situation. This is reflected in the graph in the negative edge spikes. In the positive power spikes, the dominating effect is the power consumed at X2 when it is pulled low and immediately high again.

Let us now focus on Figure 4. Both the positive and the negative edge power spikes are more balanced. The former range from 3.32mW to 3.58mW (7.8% variation), while the latter reach as high as 2.65mW and as low as 1.77mW (33.2%). The overall reduced power consumption can be attributed to the absence of unnecessary deep voltage drops, due to better "scheduling" of switching events. Of interest is also the emergence of small "secondary" power spikes immediately after the positive edge primary ones. These correspond to activity in the OR plane. In principle, any spreading and smoothing of the power spectrum should be considered favourable for anti-power analysis defence.

Even in the relatively improved state of Figure 4 the power variations in different input scenarios in the precharge power spike is unacceptably high. The obvious way to reduce it is to fight its cause. Recall that the increased power consumption in the precharge phase of the power-hungry $(a,b,c) = (1,1,1)$ case, is very much due to interplane buffer switching activity: the NAND gate pulling up and inverter INV2 pulling down. We therefore modified the NAND pull-up and INV2 pull-down transistor sizes, increasing their "on" resistance by a factor of 3. We expect this to slow down the precharge process. Note that slowing down the precharge phase does not degrade the actual PLA performance. Indeed, as soon as Req_in falls, the PLA does not need to be fully precharged until the next set of input data is valid and Req_in has been asserted again. This will only happen after a period of time that will be equal to *at least* the delay through two C-elements and four inverters. On the other hand, the above modifications do not affect the time-critical evaluation phase, since during evaluation the NAND gate may only pull *up* and INV2 only *down*. Notice that such fine tuning would not have been possible in a synchronous environment, since in that case the buffer would need to finish quickly, in time for the next clock edge.

Figure 5 shows the simulation results obtained thus. Significant improvement can be noticed. The negative edge spikes now only range between 2.66mW and 2.79mW (4.9%), while the positive edge ones range from 3.55mW to 3.78mW (6.1%).

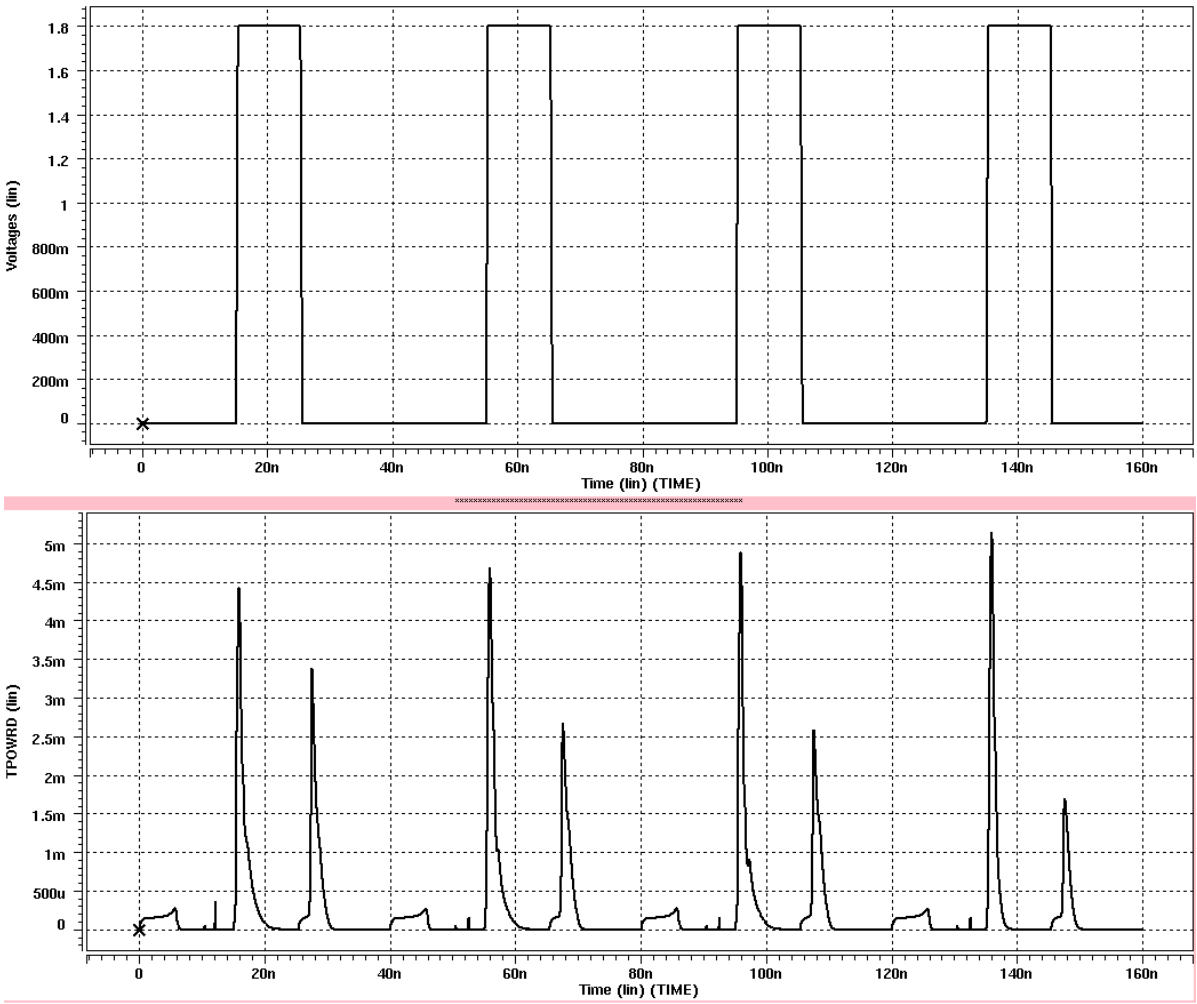

Figure 3: Req_in signal waveform (top) and Power profile of the basic structure (bottom)

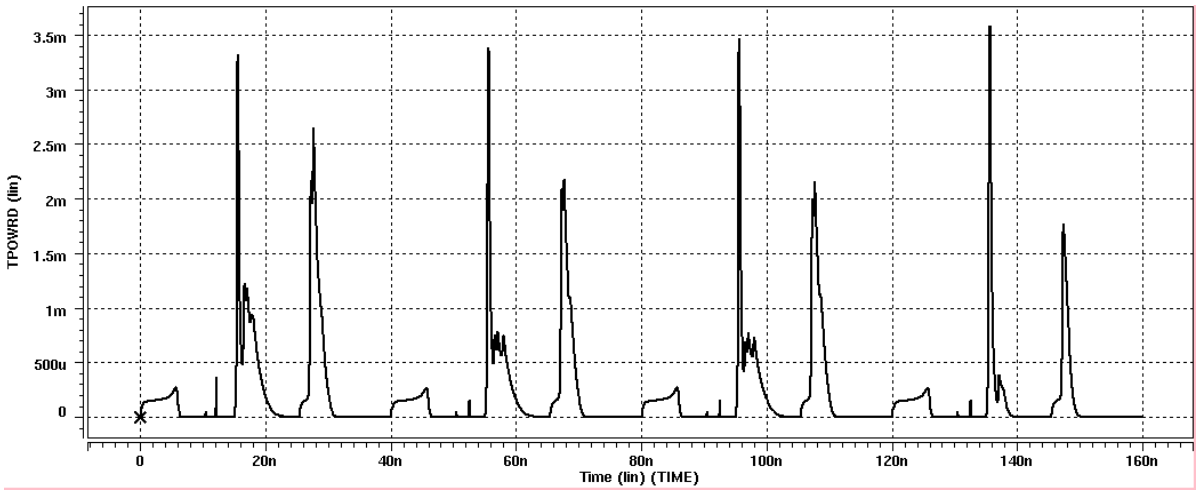

Figure 4: Power profile of the delayed structure

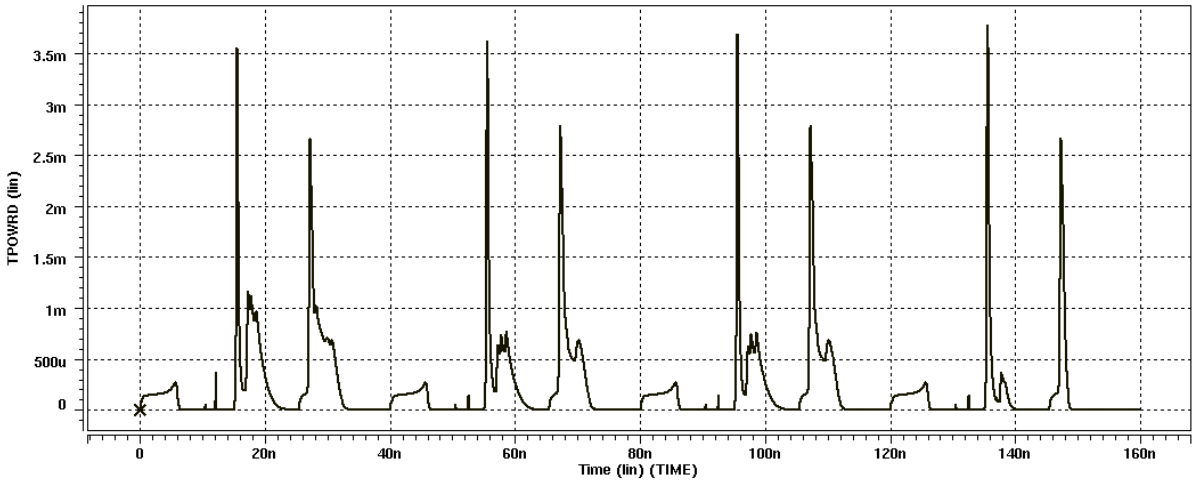

Figure 5: Security improvements through power spike balancing

3 Conclusion and on-going work

We have presented our first simulation experiments towards a general methodology for the design of side-channel attack defiant and fault-indicating asynchronous PLAs. Our on-going and future work in this direction involves simulations of larger scales as well as actual fabrication of large PLA structures.

References

[1] C.C. Wang et al, ”A Low-Power and High-Speed Dynamic PLA Circuit Configuration for Single-Clock CMOS”, IEEE Transactions on Circuits and Systems I, Vol 46, No 7, July 1999, pp 857-861.

[2] J.S. Wang et al, ”Analysis and Design of High-Speed and Low-Power CMOS PLAs”, IEEE Journal of Solid-State Circuits, Vol 36, No 8, August 2001, pp 1250-1262.

[3] S. Posluszny et al, ”Timing Closure by Design, A High Frequency Microprocessor Design Methodology”, DAC 2000, pp 712-717.

[4] S. Moore et al, ”Improving Smart Card Security Using Self-Timed Circuits”, ASYNC 2002, pp 211-218.

[5] D. Sokolov et al, ”Improving the Security of Dual-Rail Circuits”, University of Newcastle-upon-Tyne Technical Report NCL-EECE-MSD-TR-2004-101.