A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation

Kris Tiri and Ingrid Verbauwhede

UCLA Electrical Engineering Department,
7440B Boelter Hall, P.O. Box 951594, Los Angeles, CA 90095-1594

Abstract

This paper describes a novel design methodology to implement a secure DPA resistant crypto processor. The methodology is suitable for integration in a common automated standard cell ASIC or FPGA design flow. The technique combines standard building blocks to make 'new' compound standard cells, which have a close to constant power consumption. Experimental results indicate a 50 times reduction in the power consumption fluctuations.

1 Introduction

Encryption algorithms have been designed to be secure against cryptanalysis that has access to plaintext and ciphertext. The physical implementation however, provides the attacker with important information. Numerous attacks have been presented that use 'side channels', such as time delay and power consumption, as an extra source of information to find the secret key [1].

One Side Channel Attack in particular, namely the Differential Power Analysis (DPA) [2], is of great concern. It is very effective in finding the secret key and can be mounted quickly with off-the-shelf devices. The attack is based on the fact that logic operations have power characteristics that depend on the input data. It relies on statistical analysis to extract the information from the power consumption that is correlated to the secret key.

Scores of countermeasures have been presented that try to conceal the supply current variations at the architectural or the algorithmic level. Yet, they are not really effective or practical against DPA and/or its derivatives, as the variations actually originate at the logic level.

On the other hand, implementing the encryption module in a logic style, for which a logic gate has at all times a constant power consumption independently of signal transitions, removes the foundation of DPA and is an effective means to halt DPA [3]. One such logic style available is Sense Amplifier Based Logic (SABL) [4].

SABL however, asks for the design and characterization of a complete new standard cell library. In this paper, we present a design methodology, based on the principles of SABL, that is applicable for designing a secure implementation on (1) any ASIC, using a regular standard cell library with Static Complementary CMOS gates; and on (2) any FPGA, using slices, consisting of a few look up tables (LUT's), multiplexers and registers.

Instead of designing new standard cells as is done in SABL, the technique combines building blocks from an existing standard cell library or from a slice to make 'new' compound standard cells, which mimic the behavior of the SABL gates.

Being able to apply the methodology on an FPGA opens the door to do secure prototyping of a design on a single FPGA, or even to add an FPGA module on a Smart Card, which will extend the lifespan and increase the versatility of a particular Smart Card product.

Section 2 describes SABL. Next in section 3, secure compound logic gates are conceived. Section 4 discusses the variation on the power consumption and section 5 the tradeoff between the increase in security and the increase in area, time and power consumption. Section 6 discusses the extra steps inserted into a typical digital design flow. In section 7, an experiment is setup and results are provided for an ASIC and an FPGA implementation. Finally a conclusion will be formulated.

2 Sense Amplifier Based Logic

Sense Amplifier Based Logic is a logic style that uses a fixed amount of charge for every transition, including the degenerated events in which a gate does not change state. In every cycle, a SABL gate charges a total capacitance with a constant value.

SABL is based on 2 principles [4]. First, it is a Dynamic and Differential Logic (DDL) and therefore has exactly one switching event per cycle and this independently of the input value and sequence. Second, during a switching event, it discharges and charges the sum of all the internal node capacitances together with one of the balanced output capacitances.

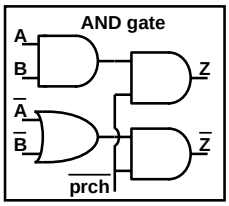
Using standard building blocks, we will now conceive secure compound logic gates, which adhere to these 2 principles.

3 Conception of Dynamic Differential Logic

3.1 Simple Dynamic Differential Logic (SDDL)

Creating a compound standard cell, which has a dynamic and differential behavior, is done with the help of (1) the De-Morgan's Law, which allows expressing the false output of any logic function, using the false inputs of the original logic function and (2) AND-ing the differential output with a precharge signal. Because of the AND-ing with the precharge signal, whenever the precharge signal is 1, the outputs are predischarged to 0 independently of the input-values. On the other hand whenever the precharge signal is 0, exactly one output, which is specified by the inputs, will evaluate to 1.

This methodology can be applied to any given standard cell or within a slice. As an example, Fig. 1 shows the implementation of the SDDL 2-input AND-gate.



$(A.B).\overline{prch} \leftrightarrow (\bar{A}+\bar{B}).\overline{prch}$

A	B	Ā	B̄	prch	Z	Z̄
0	0	1	1	0	0	1
0	1	1	0	0	0	1
1	0	0	1	0	0	1
1	1	0	0	0	1	0
X	X	X	X	1	0	0

Fig. 1. SDDL: AND-gate (left); truth table (right)

Yet, there is no guarantee that each compound gate has only one switching event per cycle. This is best seen with an example. Fig. 2 shows a switching event of a SDDL 2-input XOR-gate. The timing diagram shows that both signals of the differential output have 1 switching event, even though each differential input has only 1 switching event.

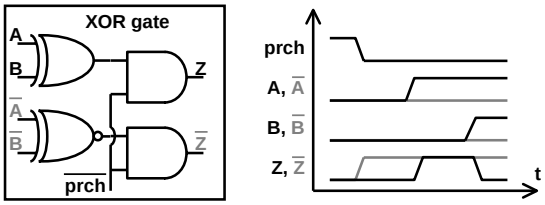


Fig. 2. SDDL: XOR-gate (left); timing diagram (right)

Both timing and value of the inputs influence the number of switching events. SDDL can never achieve an input signal independent power consumption. Restricting the problem to the conception of a secure version of the and- and or-operator resolves this.

Implementing a secure version of only the and- and or-operator is legitimate; any logic function in Boolean algebra can be expressed with only 3 operators, which are the invert-, and- and or-operator. The differential inverter is redundant. Differential Logic has both the true and the false output; the inverter is implemented by exchanging the outputs.

3.2 Wave Dynamic Differential Logic (WDDL)

In section 3.1, we ignored that the input signals, which are the outputs of dynamic gates, precharge to 0. Whenever the inputs of an any-input AND- or OR-gate are precharged to 0, the outputs are automatically at 0. There is no need to force them to 0. Consequently, performing the precharge operation inside the SDDL any-input AND-gate and the SDDL any-input OR-gate can be omitted. The dynamic differential cells are now implemented with half the resources required previously.

Special design rules, like np-rules or domino logic rules, used to cascade conventional dynamic gates are unnecessary. WDDL gates can be freely interconnected. Every compound standard cell has only 1 switching event per cycle.

3.3 Switching factor

The switching factor of WDLL is 100%. The proof is by induction. We assume that DDL registers (FF's) exist that switch only once per cycle. We will present them in section 3.4. Recall that the or- and and-operator are dual: applying the DeMorgan's law on one results in the other.

For compound AND- and OR-gates at logic depth 1 (i.e. compound gates with all inputs connected to FF's):

- At the end of the precharge phase, all input signals are at 0. Next at the onset of the evaluation phase, the FF's switch exactly one of their two output lines to 1 and provide a stable differential output.
- At this point, two scenarios are possible:

1. At least one of the input signals to a single ended OR-gate has made its transition to 1. The output of the single ended OR-gate will switch to 1 and remain at 1. On the other hand, the corresponding single ended AND-gate of the compound standard cell, which is consequently fed by at least one 0, will remain at 0.
2. All input signals to a single ended AND-gate have made their transition to 1. The output of the single ended AND-gate will switch to 1 and remain at 1. On the other hand, the corresponding single ended OR-gate of the compound standard cell, which is consequently fed by all 0's, will remain at 0.

- As a result, the differential gates at logic depth 1 switch once per cycle. Differences in input arrival time are not of any influence and do not cause glitching. For scenario 1, the single ended OR-gate will switch as soon as 1 input is 1 For scenario 2, the single ended AND-gate will switch once all inputs are 1.

Suppose now that the compound gates at logic depth n (i.e.

gates with all inputs connected to gates from logic depth smaller then n) have exactly one output transition.
Then, we can show that gates at logic depth n+1 have exactly one output transition:

- We can repeat the previous discussion. But instead that the signals come from compound registers they come from compound gates at a logic depth equal or less then n, which just like the compound register make exactly one transition. As a result the compound gates at logic depth n+1 have exactly one output transition.

This proves that each compound gate has exactly one output transition.

3.4 Precharge wave generation

Contrary to SDDL gates, WDDL gates do not precharge simultaneously. The precharged 0's ripple through the combinatorial logic. Instead of a precharge signal that resets the logic, there is a precharge wave: hence Wave Dynamic Differential Logic (WDDL).

We just created a Dynamic Logic without a big load on the precharge control signal. The gates are precharged without distributing the signal to each individual gate. Another advantage is that during the precharge phase, WDDL has a lower peak supply current. As a result, the supply bounce, often a problem for signal integrity, is lowered.

There are 2 ways to launch the precharge wave. The first method is to insert a precharge operator at the start of every combinatorial logic tree, i.e. the inputs of the encryption module and the outputs of the registers. This can be done with SDDL registers, as is shown in Fig. 3.

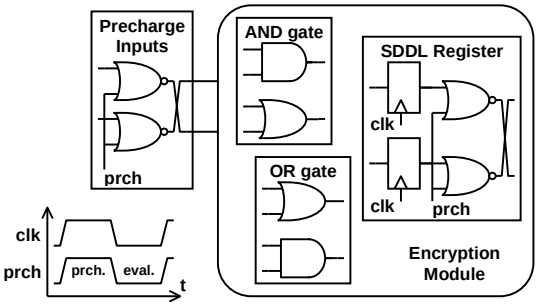


Fig. 3. Precharge wave generation with SDDL FF's

But, it is sufficient to solely precharge the input signals of the encryption module, as is shown in Fig. 4. Prerequisite is that Master-Slave DDL registers are used. Once the precharged signals have propagated, the encryption module is in stable operation mode. From then on, the registers launch the precharge wave. They store the precharged 0's, sampled at the end of the preceding precharge phase, during the evaluation phase. The input signals and all the combinatorial logic concurrently interleave precharge mode and evaluation mode.

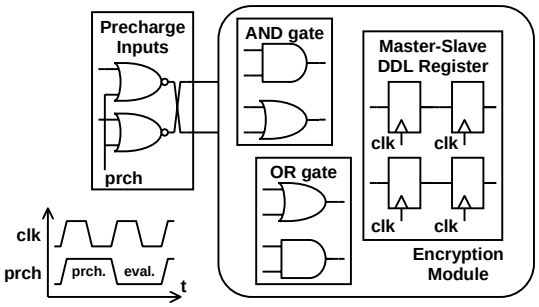


Fig. 4. Precharge wave generation with M-S DDL FF's

The second method is preferred despite the double clock frequency for the same data rate: the entire compound register is reset in every cycle.

3.5 Divided Wave Dynamic Differential Logic

Fig. 5 (left) shows an arbitrary WDDL module. Whenever an inversion is not present in the original single ended function, the WDDL implementation consists out of two distinct parts, as shown in Fig. 5 (middle). The two parts are dual. One can be derived from the other by inverting the inputs and by replacing the single ended AND-gates by single ended OR-gates and vice versa. One generates the true outputs, the other the false outputs.

As a result, it is possible to place and route the original gate level netlist and subsequently take the layout and interchange the AND- and OR-gates, Fig. 5 (right). The combination of the two single ended modules, which we will call Divided Wave Dynamic Differential Logic (DWDDL), has the same behavior as the original WDDL.

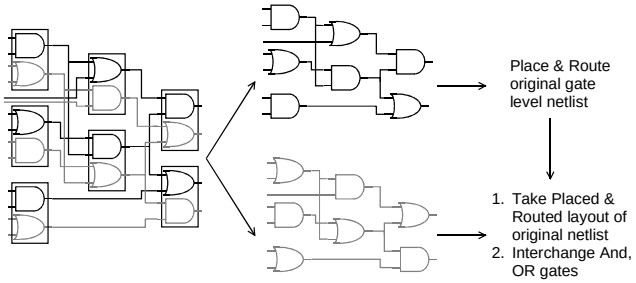


Fig. 5. Derivation of Divided WDDL

This approach avoids that the differential signals have to be matched by the router. This is a requirement derived in section 4 for constant load capacitance. The true and false signal see the 'same' environment even though they are physically separated. Furthermore, it is not necessary anymore to generate the compound standard cells. It is still necessary to match the interconnects of the inputs to the combinatorial logic and to generate compound registers.

An inversion can not exist inside the original gate level netlist. It halts the precharge wave: the 0 at the input of the inverter is propagated as a 1. Providing the true and the

false input at the single ended combinatorial logic, as in Fig. 6, assures that the module can be implemented without any further inversions. The single ended module can be synthesized as multilevel logic or as a Programmable Logic Array (PLA). NOR-NOR PLA's and NAND-NAND PLA's have a switching factor of exactly 100% despite the inversions. PLA's offer the advantage that they can be implemented with tight control over cross coupling and thus the power varation [5].

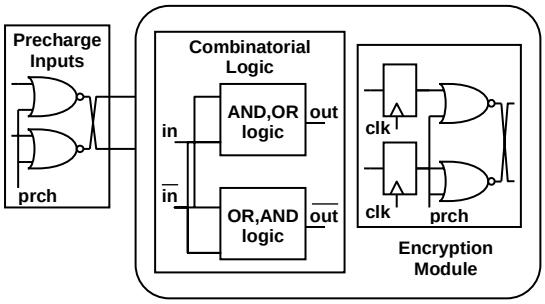


Fig. 6. Divided Wave Dynamic Differential Logic

The rest of this work discusses WDDL.

4 Constant load capacitance

For constant power consumption, the compound standard cell should charge a total capacitance with a constant value. The capacitance has four components: the internal node capacitances, the intrinsic output capacitance, the interconnect capacitance and the intrinsic input capacitance of the load.

In case of an ASIC, the AND-gate and the OR-gate, which make up a compound standard cell, are not identical. The internal and the intrinsic capacitances can not be identical between both single ended gates.

Yet, with shrinking channel-length of the transistors, the interconnect capacitance becomes more and more the dominant capacitance. This makes it appropriate to primarily concentrate on the interconnect capacitances. Under the assumption that the differential signals travel in the same environment, the interconnect capacitances are equivalent.

In case of an FPGA, it all depends on the precise implementation of the LUT. For example, for the Virtex-II platform, the propagation delay is independent of the function implemented [6]. This implicates that the internal and the intrinsic capacitances are identical. Here however, routing the signals in the same environment is harder; only a limited number of routing tracks are available.

5 Tradeoff

Increasing the security is never free. Going from a single ended design (S-E) to a WDDL design has a tradeoff with an increase in area, time and power.

For an ASIC, the lower bound on the area increase is a factor of 2, the number of gates in a WDDL gate. For an actual module the increase is higher. A WDDL FF uses 4 single ended FF's and the single ended design is synthesized with a broad palette of gates. Table 1, which compares 3 different encryption algorithms, indicates an increase between a factor 3.2 and 3.6. The area in equivalent gates and the delay are given for the datapath of a 1 round encryption plus registers in a 3.3V, 0.35µm CMOS library.

Table 1. Area and delay comparison

	Area (Kgates)		Delay (ns)	
	S-E	WDDL	S-E	WDDL
Kasumi	7,304	26,549	33.3	47.3
DES	1,493	4,810	7.9	8.4
AES	13,241	44,827	13.6	14.6

In Table 1, only Kasumi sees a significant increase in the critical path. This algorithm incorporates many xor-operations, which are responsible for the increase.

The data rate however, decreases with a factor of two. The first clock cycle is spent in evaluation phase, the second in precharge phase. Note that the precharge phase is fundamental to any dynamic logic, which is a conditio sine qua non to achieve a switching factor of 100%.

The increase in power consumption -the energy consumption per encrypted bit- is much harder to define. Many factors, such as input statistics of the data and glitching of the single ended logic, have its influence. A factor equal to the area increase may be a good approximation. This factor reflects the increase of total load capacitance. The experiment in section 7.1 indicates a power penalty of a factor 3.5 for a 4 times area increase.

On the Virtex-II FPGA, the area increase for a non-inverting logical function is a factor of 6: a single ended design can place 3 2-input gates into each 4-input 1-output LUT, whereas a WDDL design uses 2 LUT's to generate a compound gate with 2 differential outputs. Practically for an arbitrary logic function the increase is higher.

There is also an increase in critical path delay. The reason has been addressed above; the single ended design has a smaller logic depth since it can place 3 2-input gates into an LUT.

Clustering will reduce the area consumption and the delay. For example, a direct implementation of the xor-operator requires 6 LUT's with a logic depth of 2. Clustering the upper LUT's of the 3 compound gates into one LUT ($\mathbf{A.\bar{B} + \bar{A}.B}$) and the lower into the adjacent ($(\bar{A}+B).(A+\bar{B})$) requires only 2 LUT's with a logic depth of 1. The resulting cell is a WDDL cell: it has differential in- and outputs and since it is a combination of only AND and OR-gates has only one switching event per cycle.

6 Automated design flow

A major advantage of our proposed logic style is that it can be incorporated by the common EDA tool flow. We propose the design flow shown in Fig. 7.

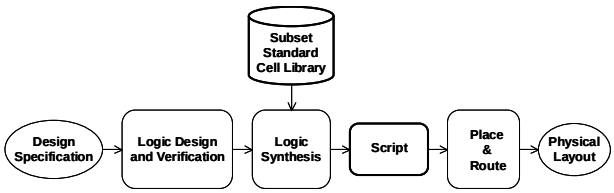


Fig. 7. Secure digital design flow

In synchronous logic, the logic design of a module can be done with a standard hardware description language, such as VHDL. Next, synthesis is done with a subset of the standard cell library. The subset consists of the inverter, all AND- and OR-gates and a register. Subsequently, a script, e.g. in PERL, transforms the resulting synthesized code at gate level to a code that reflects the differential gates. The script replaces the single ended gates with WDDL gates, removes the inverters and establishes the right connections. Next once the single ended gates are put together to form the compound standard cells, these cells can be placed by the placement-tool. At the end, the router-tool should match the two output lines of each compound gate.

The automated design flow generates a secure design from the VHDL netlist. The digital designer does not need specialized understanding of the methodology, contrary to other DPA blocking techniques. He can write the code for a crypto processor like for every other design. In the resulting encryption module, each gate has constant power consumption independently of the input signals and thus independently of which and how the operation has been coded.

7 Experimental results

The methodology is illustrated on ASIC and on FPGA. The test circuit is depicted in Fig. 8. The minimum and maximum logic depth of the combination of the balanced XOR-trees is 2 and 9 respectively. Two versions are implemented: a single ended design and a WDDL design.

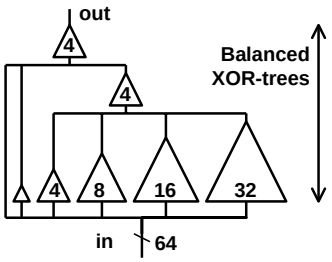


Fig. 8. Test circuit

7.1 ASIC

DPA uses statistical methods that detect small power variations. Cycle accurate simulators, which have been used to show DPA blocking techniques in [8] and [9], neglect these small power variations. Furthermore, an attacker will sample several times per clock cycle in order to capture the instantaneous current. We have simulated accurate instantaneous supply current traces at the transistor level using HSPICE. In order to capture all current variation, one sample has been taken every 10ps. The modules have been implemented in a 1.8V, 0.18µm library. The layout parasitics have been neglected. In total, 200 random input vectors have been simulated.

Table 2 indicates a reduction of a factor 37 in the NED and a factor 52 in the NSD, which are the normalized absolute variation and the normalized standard deviation of the energy per cycle respectively. The reduction comes with an increase of a factor 3.5 in the power consumption.

Table 2. Characterisation of energy consumption

	NED	NSD	E/cycle (pJ)
S-E	0.4231	0.1152	2.26
WDDL	0.0112	0.0022	7.95

Fig. 9 shows the statistical properties of the instantaneous supply current. The WDDL mean current is a representative switching event. The point wise absolute variation and standard deviation are small throughout the entire event. This is not the case for the single ended design.

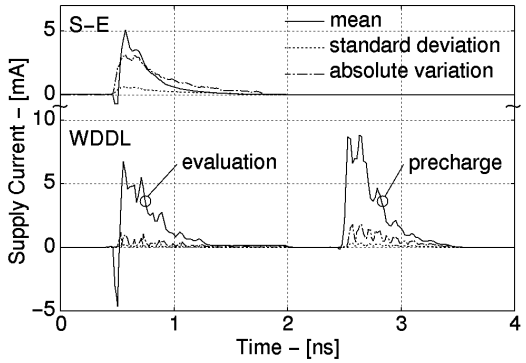


Fig. 9. Supply current characteristics

7.2 FPGA

The results in this section are actual measurement results. The prototyping board is the Xilinx Virtex-II Development Kit by Avnet Design Services [7]. Measurements are performed with a HP 54542C oscilloscope.

An indirect method to prove that the power consumption is independent of the input statistics is to show that the delay is a constant. The time constant **R.C** is proportional

to the time required to charge a capacitance **C** through a resistor **R**. Since the propagation delay of an LUT is independent of the logic function implemented, **R** is a constant. As a result, since the charge **Q**, and thus the power consumption **P,** is proportional to the capacitance **C**, showing that the time delay **R.C** is a constant, shows that **C**, and consequently **Q**, is a constant. In our test circuit, the delay is set by a summation of the individual **R.C** time constants. Showing that the total delay is independent of the input vector shows that the summation of the individual capacitances **C**, and consequently **Q**, is a constant.

Fig. 10 shows a measured output voltage transient for 8 clock cycles. The figure shows that the single ended design suffers from glitches. The WDDL implementation on the other hand, has as expected only one transition.

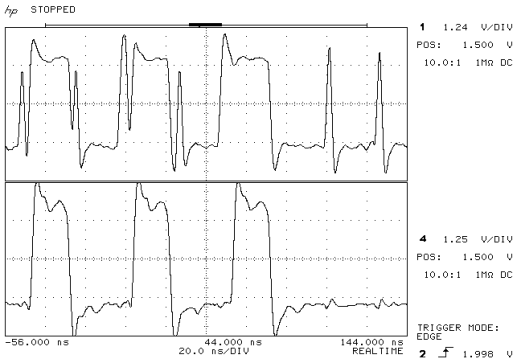


Fig. 10. Output transient: S-E (top); WDDL (bottom)

Fig. 11 depicts an eye diagram of the output voltage. The measurements are triggered by the positive edge of the precharge signal. The figure shows a much larger variation for the S-E design than for the WDDL implementation. The jitter on the output of the former, which has a minimum logic depth of 1 LUT and a maximum of 5 LUT's, is 7ns. The jitter of the latter, which has a minimum logic depth of 2 LUT's and a maximum of 9 LUT's, is 2 ns.

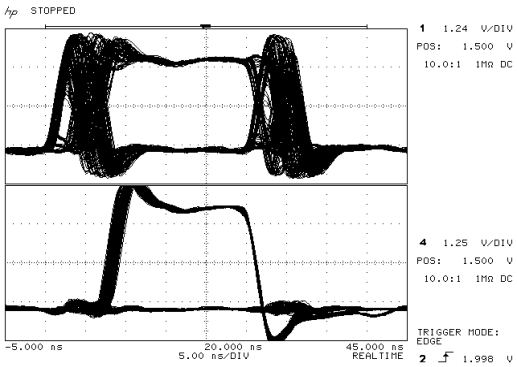


Fig. 11. Output eye-diagram: S-E (top); WDDL (bottom)

We have not measured the instantaneous supply current. The prototyping board has its own power supply and there is no space to insert a sensing resistor. Yet even with our own power supply or with a sensing resistor, accurate instantaneous current measurements are impossible on a commercial board. These boards have large decoupling capacitances, placed adjacent to the FPGA power pins and dimensioned to prevent power supply bounce. These capacitances also seize the current variations. In order to make accurate instantaneous supply current measurements, we are working on the design of a custom board.

8 Conclusions

We have presented a design methodology to implement a DPA-resistant crypto processor. The technique combines standard building blocks into secure compound gates, which mimic the behavior of SABL gates. The technique can be readily applied using both regular standard cell based ASIC design flow and FPGA design flow. An important advantage of our design methodology is that the implementation details of how to create a secure encryption module are hidden from the designer. Experimental results have demonstrated that WDDL is an effective technique to achieve an important reduction in the power variation for both ASIC and FPGA. The tradeoff is in an increase in area, time and power consumption.

Acknowledgement

The authors would like to acknowledge the support of the National Science Foundation (CCR-0098361).

9 References

1 E. Hess, N. Janssen, B. Meyer, T. Schuetze. Information Leakage Attacks Against Smart Card Implementations of Cryptographic Algorithms and Countermeasures – a Survey. *EUROSMART Security Conference (2000)* pp. 55–64

2 P. Kocher, J. Jaffe, B. Jun. Differential Power Analysis. *Proc. of Advances in Cryptology (1999)* pp. 388-397

3 K. Tiri, I. Verbauwhede. Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology. *CHES 2003* pp. 125–136

4 K. Tiri, M. Akmal, I. Verbauwhede. A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards. *ESSCIRC 2002* pp. 403-406

5 S. Khatri, R. Brayton, A. Sangiovanni-Vincentelli. Cross-talk Immune VLSI Design using a Network of PLAs Embedded in a Regular Layout Fabric. *ICCAD 2000* pp. 412-418.

6 Xilinx: Virtex-II Platform FPGAs: Detailed Description. www.xilinx.com/bvdocs/publications/ds031-2.pdf (2002)

7 Avnet Design Services: Xilinx Virtex-II Development Kit. http://www.ads.avnet.com PN: ADS-XLX-V2-DEV1500

8 Saputra, H., Vijaykrishnan, N., Kandemir, M., M.J. Irwin, R. Brooks, S. Kim, and W. Zhang. Masking the Energy Behavior of DES Encryption. *DATE 2003* pp. 84-89

9 L. Benini, A. Macii, E. Macii, E. Omerbegovic, M. Pancino, and F. Pro. Energy-Aware Design Techniques for Differential Power Analysis Protection. *DAC 2003* pp. 36-41