A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards

Kris Tiri, Moonmoon Akmal and Ingrid Verbauwhede
EE Dept. - University of California, Los Angeles
tiri@ee.ucla.edu

Abstract

To protect security devices such as smart cards against power attacks, we propose a dynamic and differential CMOS logic style. The logic operates with a power consumption independent of both the logic values and the sequence of the data. Consequently, it will not reveal the sensitive data in a device. We have built a set of logic gates and flip-flops needed for cryptographic functions and compared those to Static Complementary CMOS implementations.

1. Introduction

Electronic banking, e-commerce, virtual private networks and so on cannot operate without encryption technology and a secure implementation of the encryption technology. To obtain security, many strong encryption algorithms have been developed. While usually strong against mathematical attacks, side channel attacks can reveal the secret key through information leaked by the hardware implementation of the encryption module. Differential Power Analysis (DPA) is based on the fact that logic operations have power characteristics that depend on the input data [1]: statistical analyses of measured power traces link the switching activities of the circuit to the secret key.

Different techniques have been proposed to prevent this information leakage. On the algorithmic level, random process interrupts interleave dummy instructions to avoid sequential execution of the algorithm [1],[2]. Integration techniques, however, are able to resynchronize the power traces [3]. Masking is a technique that prevents intermediate variables to depend on the knowledge of an easily accessible subset of the secret key [4]. DPA has been modified to handle masking [5]. On the architectural level, techniques include adding random power consuming operations [1],[2] and duplicating logic with complementary operations [2]. These procedures merely lower the side-channel information [1],[4] and might easily be disabled through tampering. Active power signal filtering with power consumption compensation, passive filtering, battery on chip and detachable power supply influence the power transfer itself [6]. The first method lags behind the fast power fluctuations and physical dimensions limit the latter three.

The foregoing methods attempt to conceal the supply current variations at the architectural or the algorithmic level, while they originate at the logic level. A simple means to halt DPA is to have the encryption module or at least the sensitive parts of it implemented in a logic, whose power consumption is independent of the signal transitions. As we will see in section 2, differential logic, which is often brought up as a solution, does not possess such power characteristics. The following section develops a dynamic and differential CMOS logic style starting from the problem setting of the DPA. Next, the proposed logic style is compared with Static Complementary CMOS regarding power variation, power consumption and area. Finally a conclusion will be formulated.

2. Imperfections in existing logic and derivation of the dynamic and differential logic

As shown in Figure 1, Static Complementary CMOS logic (scCMOS), which is the default logic style in standard cell libraries used for security IC's, only consumes energy from the power supply when its output has a 0-1 transition [7]. During the 1-0 transition, the energy previously stored in the output capacitance is dissipated. In the two degenerated events of a 0-0 or a 1-1 transition no power is used. This asymmetric power demand provides the information used in DPA to find the secret key.

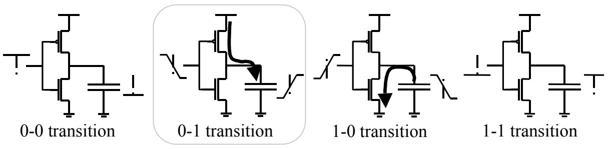
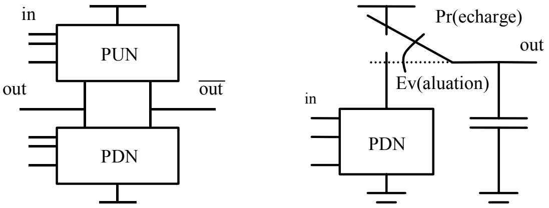


Figure 1. scCMOS: output transitions and their asymmetric power consumption

A logic style, with data-independent power consumption does not reveal this information. When logic values are measured by charging and discharging capacitances, we need to use a fixed amount of energy for every transition. Or in other words even though different capacitances are switched, we need a logic style with the unique property of charging in every cycle a total capacitance with a constant value. The proposed Sense Amplifier Based Logic (SABL) achieves this goal by (1) switching the output independently of the input value, sequence and by (2) having a constant load capacitance equal to all internal nodes combined with one of the balanced output loads.

In SABL, the combination of dynamic and differential logic [7] will charge a capacitance for the four output transitions (0-0,0-1,1-0,1-1). Figure 2 shows the output events for differential logic and for dynamic logic separately. A differential logic masks the input value: independent of the input value, energy is dissipated when exactly one output node is discharged. Therefore there is no difference between a 0-1 and a 1-0 event or between a 0-0 and a 1-1 event. However one can still differentiate between those two main classes: a 0-1/1-0 transition will consume power whereas a 0-0/1-1 not. Note that this is the main reason that addressing the power attack solely by balancing the Hamming weights can not succeed [1], [4]. Whether it is done on algorithmic level (e.g. exclusively handling bytes with Hamming weight 4), architecture level (e.g. duplicating the module with a complementary module) or logic level (e.g. differential logic), this difference will remain present. In none of these proposed methods, a vector with 0-0/1-1 transitions consumes power.

A dynamic logic breaks the input sequence: independent of the input-switching behavior, energy is consumed when the load capacitance is charged. Therefore there is no difference between a 0-0 and a 1-0 event or between a 1-1 and a 0-1 event. Here, only the 0-1/1-1 transition consumes power during the following precharge phase. We clearly need to combine the two into one dynamic differential logic style that switches the output independently of the input value or sequence.



in_i	in_{i+1}	out_i	out_{i+1}	$\overline{\text{out}}_i$	$\overline{\text{out}}_{i+1}$
0	1	1	0	0	1
1	0	0	1	1	0
0	0	1	1	0	0
1	1	0	0	1	1

in_i	in_{i+1}	out_{Ev}	out_{Pr}
0	1	0	1
1	0	1	1
0	0	1	1
1	1	0	1

Figure 2. Power characteristics in form of truth table for differential logic (left) and dynamic logic (right)

Merely making it dynamic and differential is not sufficient however, as it makes the four transitions equal but only to the first order. This will be shown for the dynamic DCVSL style [7]. The DCVSL inverter has a very regular power consumption: simulations show a variation smaller then 1%. But for more complicated logic functions this number will not be accurate. Figure 3 shows the DCVSL AND-NAND gate, for which simulations indicate that the difference can be as large as 50%. This is caused by asymmetry in the gate. Depending on the input, different parasitic capacitances discharge during the evaluation phase. In the succeeding power consuming precharge phase, these capacitances are recharged. In none of the 4 different events, the same combination of capacitances has to be charged.

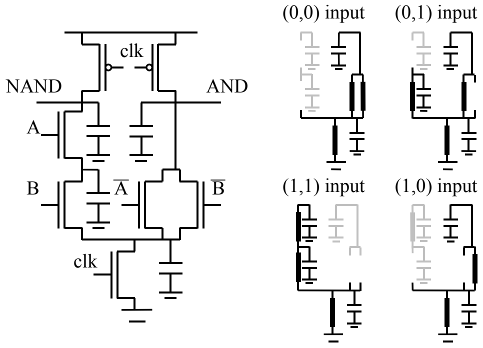


Figure 3. Dynamic DCVSL AND-NAND gate with its parasitic capacitances (left) and the gate's different discharging events during evaluation phase (right)

SABL makes the four output events equal, by charging at every event the same capacitance value: one of the balanced load capacitances and the sum of all the internal node capacitances.

2.1. SABL: basic gate

The SABL gate is based on the StrongArm110 flip-flop (SAFF) [8]. To realize a basic gate, we keep the sense amplifier half of the flip-flop and replace the input differential pair by a differential pull down network (DPDN). Figure 4 depicts the generic n-gate. The DPDN is implemented such that for a stable input combination all nodes that are internal to the DPDN connect to one of the output nodes. During evaluation (clk high) the cross-coupled inverter will toggle to one state and provide a stable output as soon as the DPDN provides a path to ground.

Transistor M_1, which is always on, prevents a floating node by serving as a path for subthreshold currents, as it does in case of the original SAFF. In addition, in case of the SABL gate, M_1 guarantees that all internal nodes discharge. Either which branch is on, all internal nodes and their respective capacitances are connected through M_1 and will eventually be discharged together with one of the output nodes. The differential output nodes connect differential signals to a differential input. Under the assumption that the differential signals travel in the same environment, the interconnect capacitances are equivalent. Therefore the total output capacitance -consisting of one of the equal intrinsic output capacitances, one of the interconnect capacitances and one of the symmetrical input capacitances- is a constant. During precharge (clk low) all the discharged nodes and capacitances will be precharged. As such every cycle the same capacitances are discharged and charged what makes the power consumption of the gate independent of the input statistics.

A generic p-gate is implemented as a gate that precharges to GND when clk is high and evaluates one node to VDD through a DPUN when clk is low. Figure 4 illustrates the implementation of a AND-NAND n-gate. Figure 5 shows the discharging and charging events of the AND-NAND n-gate for different inputs.

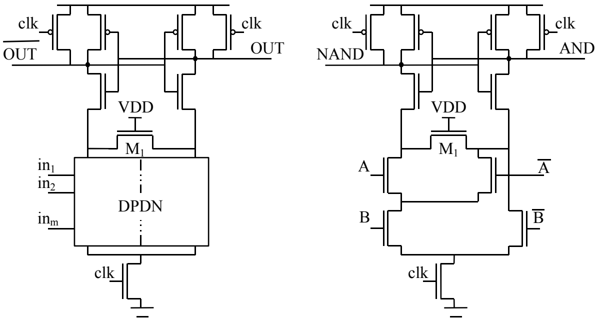


Figure 4. SABL logic: generic n-gate (left) and AND-NAND gate (right)

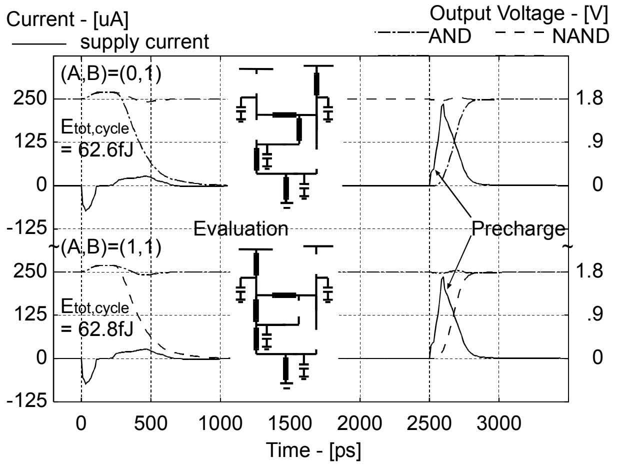


Figure 5: Simulated transient response of AND-NAND gate for (0,1)-input (top) and (1,1)-input (bottom)

2.2. SABL: cascading gates

As it is a dynamic logic, SABL has to be connected using either Domino or np-logic [7]. In case of Domino logic, the use of static inverters between gates does not harm: every cycle exactly one inverter will have a 0-1 event.

2.3. SABL: storage

The Set-Reset latch of the SAFF is static to prevent that the output value would be lost. However if the input to the flip-flop does not change, the output of the latch will not change. Consequently there will be no power consumption in this static gate.

A combination of a p-SA and a n-SA, as shown in Figure 6, acts as a master-slave flip-flop. The p-SA evaluates at the falling edge of clk and keeps this value till the rising edge, while the n-SA evaluates at the rising edge of clk and keeps this value till the falling edge. As a result, the value is stored during one clock cycle. For correct operation, the actual precharge time has to be large enough to evaluate the correct differential input. This delay is implemented with static inverters. Note that each SA can as well be replaced with a module of cascaded gates.

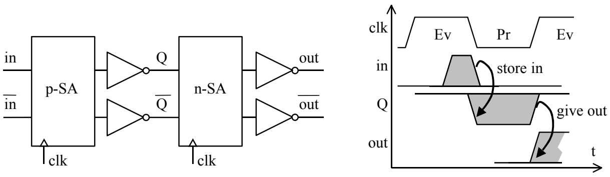


Figure 6. SABL logic: flip-flop implementation (left) and operation (right)

3. Experimental results

Encryption algorithms use arithmetic that is different from the twos complement arithmetic on integers or real numbers. Instead they use Galois field arithmetic and operations such as substitutions and permutations [9]. The operations can be implemented with the following set of basic cells: inverter, NAND, XOR and flip-flop (FF). We have built and compared this set with scCMOS implementations. Besides, we have also tested a typical cryptographic block, namely a substitution box. The substitution box (S9-box) is a main component of the Kasumi algorithm, the encryption algorithm in 3G cellular standard [10]. The block consists of 5 inverter's, 86 NAND's and 92 XOR's and replaces 9 input bits with 9 altered bits.

Circuits are designed in a 0.18-µm, 1.8V CMOS technology. Layout is created with LayoutPlus of CADENCE. Figure 7 shows the layouts of the S9-box. Layout to netlist is done with Analog Artist of CADENCE. Simulations are done in Hspice, using the LEVEL 49 transistor model. Basic cells are simulated with a fan out of 4 inverters. The variation on the power consumption is measured for a random input sequence of length 300 and 500 clock cycles for basic cells and S9-box respectively.

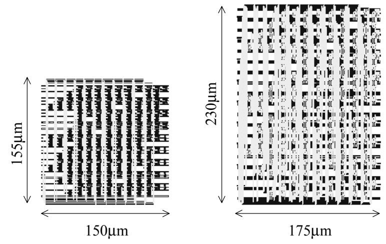


Figure 7. S9-box in scCMOS (left), SABL (right)

The power consumption is represented by the energy per cycle and as a measure for the variation on the power consumption we define the normalized energy deviation:

$$NED = \frac{Max(energy/cycle) - \min(energy/cycle)}{Max(energy/cycle)} \qquad (1)$$

NED ranges from 0 to 1. The smaller the variation, or in other words NED, the more measurements are necessary and the more accurate the measuring devices have to be in order to extract the side-channel information. In case of the S9-box, where an event is rather the sum of large number of independent identical distributions, the distribution will tend towards the normal distribution. Here the normalized standard deviation (NSD), the standard deviation divided by the mean, is also given.

Table 1 demonstrates that SABL behaves as expected. NED is reduced from more then 80% for scCMOS to below 3% for SABL. NSD decreases from 29% to 0.6%. For the S9-box, SABL uses less than two times the area and the energy of scCMOS. In comparison, the ineffective technique of doubling the logic with complementary logic [2] requires twice the area and energy. Figure 8, a histogram of the observed energies per cycle for the S9-box, and Figure 9, a superposition of the power supply current for successive cycles of the simulated transient response, show that while the observed energies are spread out in a broad range for scCMOS, they remain in a narrow band for SABL. SABL is subject to only minor delay variations as the same amount of charge has to go through very similar paths during both precharge and evaluation. Furthermore the histogram shows that the observed energies of SABL are situated near the maximum energy of scCMOS. In comparison, the active power filtering technique [6] will at least require this maximum energy.

Table 1. scCMOS vs. SABL

	INV	NAND	XOR	FF	S9-box			
	NED	NED	NED	NED	NED	NSD	E/cycle (pJ)	Area (μm^2)
scCMOS	1.000	1.000	1.000	0.821	1.000	0.293	5.92	21,446
SABL	0.009	0.032	0.016	0.020	0.032	0.006	11.32	38,541

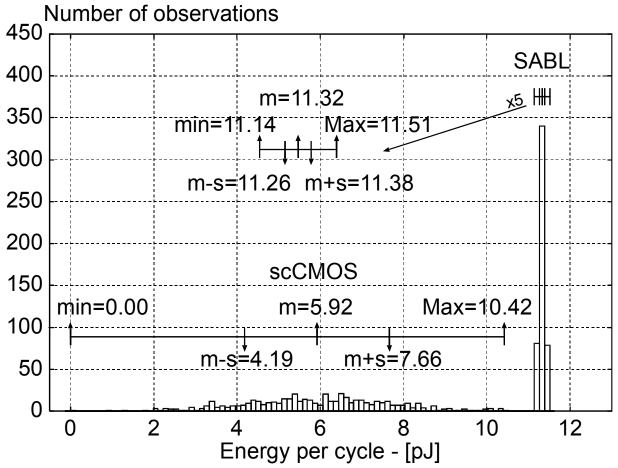


Figure 8. S9-box: histogram of encountered energy consumption per cycle

Scheduling operations, which have different power characteristics at different instances of time, does not influence the power consumption of a SABL design. Indeed whether or not actual data is processed, every gate evaluates at the clock edge. Furthermore, clock gating can be applied if the operations scheduled at a certain instance of time are independent of the internal data. This means that in the controller finite state machine, conditional if-then-else branches should be masked by activating the combination of submodules used in each branch and by adding idle states if one branch is longer than the other.

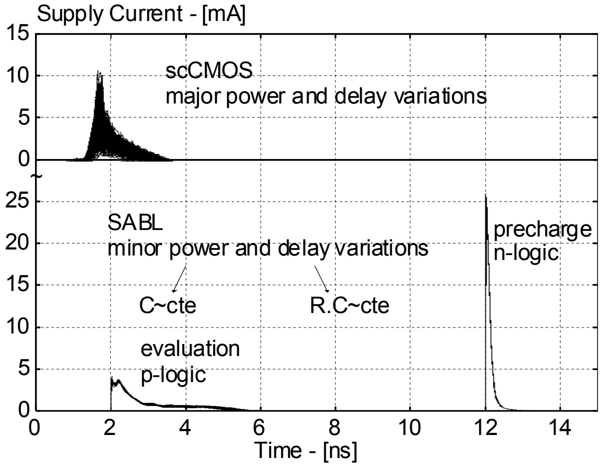


Figure 9. S9-box: superposition of the power supply current for 500 successive cycles of the simulated transient response

4. Conclusions

A dynamic and differential CMOS logic style is presented in which a gate uses a fixed amount of energy per evaluation event. To this purpose, the gate switches its output at every event and loads at that instant a constant capacitance. Experimental results demonstrate a normalized energy variation that is up to 116 times less pronounced when compared to scCMOS implementations while using only two times the area and power.

5. References

[1] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis," *in Proc. of Advances in Cryptology (CRYPTO'99)*, Lecture Notes in Computer Science, vol. 1666, 1999, pp. 388-397.

[2] J. Daemen and V. Rijmen, "Resistance Against Implementation Attacks: A Comparative Study of the AES Proposals," *in Proc. of the Second Advanced Encryption Standard (AES) Candidate Conf.*, http:// csrc.nist.gov/encryption/aes/ round1/conf2/aes2conf.htm, March 1999.

[3] C. Clavier, J.-S. Coron and N. Dabbous, "Differential Power Analysis in the Presence of Hardware Countermeasures," *in Proc. of Cryptographic Hardware and Embedded Systems (CHES 2000)*, Lecture Notes in Computer Science, vol. 1965, 2000, pp. 252-263.

[4] S. Chari, C. S. Jutla, J. R. Rao and P. Rohatgi, "Towards Sound Approaches to Counteract Power-Analysis Attacks," *in Proc. of Advances in Cryptology (CRYPTO'99)*, Lecture Notes in Computer Science, vol. 1666, 1999, pp. 398-412.

[5] T.S. Messerges, "Using Second-Order Power Analysis to Attack DPA Resistant Software," *in Proc. of Cryptographic Hardware and Embedded Systems (CHES 2000)*, Lecture Notes in Computer Science, vol. 1965, 2000, pp. 238-251.

[6] A. Shamir, "Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies," *in Proc. of Cryptographic Hardware and Embedded Systems (CHES 2000)*, Lecture Notes Computer Science, vol. 1965, 2000, pp. 71-77.

[7] J. Rabaey, *Digital Integrated Circuits: A design perspective*, Prentice Hall, 1996.

[8] B. Nikolic, V. G. Oklobzija, V. Stojanovic, W. Jia, J.K. Chiu and M.M. Leung, "Improved Sense-Amplifier-Based Flipflop: Design and Measurements," *IEEE J. Solid-State Circuits*, vol. 35, pp. 876-883, June 2000.

[9] A. J. Menezes, P. C. Van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[10] *Specification of the 3GPP Confidentiality and Integrity Algorithms*, http://www.etsi.org/dvbandca/3GPP/3gppspecshtm , 1999.