A Digital Design Flow for Secure Integrated Circuits

Kris Tiri, *Member, IEEE*, and Ingrid Verbauwhede, *Senior Member, IEEE*

***Abstract*—Small embedded integrated circuits (ICs) such as smart cards are vulnerable to the so-called side-channel attacks (SCAs). The attacker can gain information by monitoring the power consumption, execution time, electromagnetic radiation, and other information leaked by the switching behavior of digital complementary metal–oxide–semiconductor (CMOS) gates. This paper presents a digital very large scale integrated (VLSI) design flow to create secure power-analysis-attack-resistant ICs. The design flow starts from a normal design in a hardware description language such as very-high-speed integrated circuit (VHSIC) hardware description language (VHDL) or Verilog and provides a direct path to an SCA-resistant layout. Instead of a full custom layout or an iterative design process with extensive simulations, a few key modifications are incorporated in a regular synchronous CMOS standard cell design flow. The basis for power analysis attack resistance is discussed. This paper describes how to adjust the library databases such that the regular single-ended static CMOS standard cells implement a dynamic and differential logic style and such that 20 000+ differential nets can be routed in parallel. This paper also explains how to modify the constraints and rules files for the synthesis, place, and differential route procedures. Measurement-based experimental results have demonstrated that the secure digital design flow is a functional technique to thwart side-channel power analysis. It successfully protects a prototype Advanced Encryption Standard (AES) IC fabricated in an 0.18-μm CMOS.**

***Index Terms*—Circuit synthesis, CMOS digital integrated circuits, cryptography, design automation, routing, security, side-channel power analysis.**

I. Introduction

SECURITY is only as strong as its weakest link. The wireless, distributed revolution has put integrated circuits (ICs) and devices in many small embedded and wireless applications, such as smart cards, cellular phones, personal digital assistants (PDAs), and other gadgets. These applications require in almost all cases security and privacy protection. Yet, the security IC, which provides the support for the required algorithms and protocols, emerges as the main vulnerability. Due to physical and electrical effects, it broadcasts information that is related to the secret key. In recent years, several attacks that use information leaked by the so-called "side-channels" to find the secret key have been presented [1]. The attacks analyze information ranging from time delay [2] and power consumption [3] to electromagnetic radiation [4] and often apply advanced statistical techniques to reveal the secret information. In general, side-channel attacks (SCAs) do not require expensive equipment and are rather quick to set up. Even if measures are included to make the devices tamperproof, side-channel information can leak out. SCAs are a real threat for any device of which the security IC is easily observable such as smart cards and embedded devices [5], [6].

At first, SCAs have been fought with ad hoc countermeasures. For instance, the addition of random power-consuming operations obscured the data-dependent variations in the power consumption [7]. The attacks, however, have evolved and become more and more effective. Subsequently, countermeasures have been conceived at the different abstraction levels of the security application. It started at the algorithmic level. One illustration is masking [8]. In this technique, a random "mask" is added to the data prior to the encryption and removed afterwards without changing the result. Algorithmic countermeasures, however, need to be reformulated for each algorithm, and, often, proposed solutions actually appear insecure and/or inefficient afterwards [9], [10]. Only recently, dedicated hardware techniques have been presented [11]–[15]. Instead of concealing or decorrelating the side-channel information, these techniques pursue the effect of not creating any side-channel information. The goal of these countermeasures is to balance the power consumption of the logic gates. The major advantages are that this approach is correct by construction, is independent of the cryptographic algorithm or arithmetic implemented, and is a distributed measure.

The idea is to create digital circuit styles that have a switching behavior independent of the data or sequence of data that they are processing. We propose a logic style called wave dynamic differential logic (WDDL) and a layout technique called differential routing to address this problem.

A third contribution consists of the fact that we integrate these changes in a regular complementary metal–oxide–semiconductor (CMOS) standard cell design flow with minimum changes. In this paper, we will transform a regular synchronous digital design flow into a secure digital design flow [16]. A secure digital design flow is an automated design flow that creates a secure IC or system-on-chip. The design flow starts from the design specifications and results in a secure power-analysis-attack-resistant layout through the subsequent steps of synthesis and place & route. Major smart card vendors and service provides have identified such a design flow as an important open issue related to the general security of cryptographic applications [17], [18]. Recently, several

Manuscript received November 2, 2004; revised April 10, 2005. This work was supported in part by the National Science Foundation (CCR-0098361), UC-Micro 02-079 and 03-088, Panasonic Foundation, Sun Microsystems, and Atmel Corporation. This paper was recommended by Associate Editor A. Raghunathan.

K. Tiri is with Intel Corporation, Hillsboro, OR 97124 USA.

I. Verbauwhede is with the Katholieke Universiteit Leuven, Leuven B-3001, Belgium.

Digital Object Identifier 10.1109/TCAD.2005.855939

research projects have been set up in an attempt to develop a secure digital design flow [19], [20]. To our knowledge, this publication is the first to present a comprehensive top-down automated synchronous very large scale integrated (VLSI) design flow that pursues a constant power dissipation of the security IC.

The modifications and additions are inserted in the back end of the regular automated design flow and have been implemented in a "push-button" approach. They only have a minimal influence on the design flow and a negligible overhead in design time. The additional steps required only a total of 6 min of central processing unit (CPU) time for our prototype IC implementing a high-throughput Advanced Encryption Standard (AES), controller, and fingerprint processor [37].

The remainder of this paper is organized as follows. In Section II, we discuss constant power-consuming logic styles. In Section III, a place & route technique that controls the parasitic effects on differential interconnect wires is described and analyzed. Next, in Section IV, we present the secure digital design flow. Section V compares the secure digital design flow with a regular digital design flow. With the prototype IC, two functionally identical coprocessors have been fabricated in an 0.18-μm CMOS on the same die. The first, "secure," coprocessor is implemented using the secure design flow. The second, "insecure," coprocessor is implemented using a regular design flow. Area and power numbers are given, and the results of a differential power analysis (DPA) are provided. Finally, a conclusion is formulated.

II. Constant Power-Consuming Logic Styles

The power consumption of traditional standard cells and logic is dependent on the signal activity. When the output of the logic gate makes a 0 to 1 transition, a current comes from the power supply and charges the output capacitance. On the other hand, when the output sees a 1 to 0, a 0 to 0, or a 1 to 1 transition, no or only a limited amount of energy (due to short circuit or leakage) is consumed from the power supply. This is the fundamental reason why information is leaked through the power supply and why power attacks are possible. The basis of a secure digital design flow is a logic style with constant power consumption.

Current mode logic (CML), e.g., current steering logic [21], seems the ideal solution. This type of logic continuously draws a current from the supply and measures its state through the path that the current takes. A gate has constant power consumption if it draws a perfectly constant current from the power supply independently of the input and output signals. To build a current source capable of generating a constant current, special circuit techniques that minimize channel length modulation have to be used [22]. The decisive drawback of CML, however, is its static power consumption. When the logic gate is not processing any data, it burns the current, which makes this logic style unacceptable for embedded battery-operated devices.

Voltage mode logic (VML), e.g., static CMOS logic, only draws a current from the supply to change state and measures its state by the amount of charge it stores on a capacitance. Static CMOS is the preferred logic style because of its low power consumption and high noise margins. Yet, two conditions must be satisfied for VML to have constant power consumption, namely: 1) a logic gate must have exactly one switching event per signal transition and 2) the logic gate must charge a constant capacitance in that switching event.

Dynamic differential logic, sometimes also referred to as dual rail with precharge logic, fulfills the first condition [23]. A differential logic family uses the true and the false representation of the input and output signals and a dynamic logic family alternates precharge and evaluation phases. As a result, since both outputs (true and false) are precharged to 1, exactly one of the two output nodes evaluates to 0 to have a differential output signal in the evaluation phase. The discharged output node is charged to 1 in the following precharge phase to precharge both outputs to 1. In other words, every signal transition, including the events in which the input signals remain constant, is represented with an actual switching event, in which the logic gate charges a capacitance. All the logic families that have been introduced to thwart the DPA [asynchronous logic [12]–[14], sense amplifier based logic (SABL) [11], [23], and WDDL [15]], employ some form of dynamic differential logic.

In self-timed asynchronous logic [12]–[14], the terminology refers to dual rail encoded data, in which codewords are interleaved with spacers. The codewords can be seen as differential data in the evaluation phase, while the spacers as the precharge values in the precharge phase. The major disadvantage of the asynchronous approach is that it is extremely difficult to make reasonable sized designs. The methodology for the design of large asynchronous logic systems lags substantially behind that of synchronous circuits. Compared to electronic design automation (EDA) support for synchronous designs, which is very mature, there is still a shortage of computer-aided design (CAD) tools to support asynchronous circuit designs as is acknowledged by the asynchronous research community.

SABL [11], [23] has been conceived to thwart the DPA. It uses advanced circuit techniques to guarantee that the load capacitance has a constant value. SABL completely controls the portion of the load capacitance that is due to the logic gate. The intrinsic capacitances at the differential input and output signals are symmetric, and, additionally, it discharges and charges the sum of all the internal node capacitances. A major disadvantage is the nonrecurrent engineering costs of a custom-designed standard cell library development. SABL also suffers from a large clock load. The clock signal is distributed to all standard cells, as is common to all clocked dynamic logic styles.

In this paper, we propose to use the WDDL [15], because it can be implemented with static CMOS logic. Static CMOS standard cells are combined to form secure compound standard cells, which have a reduced power signature. WDDL has many advantages. It can be readily implemented from an existing standard cell library. The design flow is fully supported with accurate EDA library files that come directly from the vendor. WDDL also results in a dynamic differential logic with only a small load capacitance on the precharge control signal and with the low power consumption and the high noise margins of static CMOS. Furthermore, since the gates do not precharge in parallel, it also benefits from a low supply current derivative di/dt and peak supply current.

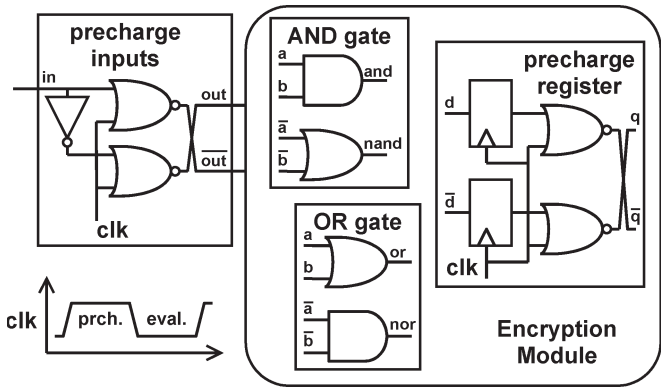

Fig. 1. WDDL: Precharge wave generation.

A. Wave Dynamic Differential Logic

A WDDL gate consists of a parallel combination of two positive complementary gates, one calculating the true output using the true inputs, the other the false output using the false inputs. A positive gate produces a zero output for an all zero input. The AND gate and the OR gate are examples of positive gates. A complementary gate, sometimes also referred to as a dual gate, expresses the false output of the original logic gate using the false inputs of the original gate. The AND gate fed with true input signals and the OR gate fed with false input signals are two dual gates. Fig. 1 shows the WDDL AND gate and the WDDL OR gate. In the evaluation phase, each input signal is differential and the WDDL gate calculates its differential output. In the precharge phase, the inputs to the WDDL gate are set at 0. This puts the output of the gate at 0.

1) WDDL Wave Generation and Propagation: A module in WDDL precharges without distributing the precharge signal to each individual gate. During the precharge phase, the input vector of the combinatorial logic is set at all 0s. Each individual gate will eventually have all its inputs at 0, evaluate its output to 0, and pass this 0 value to the next gate. One could say that the precharge signal travels over the combinatorial logic as a 0-wave, hence, WDDL. There are several ways to launch to precharge wave. In Fig. 1, a precharge operator is inserted at the start of every combinatorial logic tree, i.e., the inputs of the encryption module and the outputs of the registers. They produce an all-zero output in the precharge phase (clk-signal high) but let the differential signal through during the evaluation phase (clk-signal low).

In [15], the library size has been restricted to assure that every gate has a switching factor of exactly 100%. The set of secure compound logic gates is restricted to the WDDL AND and OR gates. Since any logic function in Boolean algebra can be implemented with the AND, OR, and INVERT operators, and given that the compound gates have differential outputs, this library is sufficient to implement any digital design. Special design rules, like NP-rules or domino logic rules, used to cascade conventional dynamic gates are unnecessary. WDDL gates can be freely interconnected. By way of illustration, Fig. 2. shows a measured output voltage transient for ten clock cycles of a test circuit implemented on a field programmable gate array (FPGA). The nonsecure single-ended design suffers from glitches and irregular switching behavior. The WDDL implementation, on the other hand, has, as expected, only one transition. Whenever the output out does not switch, the differential output $\overline{\text{out}}$ switches.

2) WDDL Library Construction: The library can be expanded to include all functions in which the AND and OR operators are combined. Additionally, since all signals will eventually be differential, the input signals may be inverted and the output signals may be inverted. We selected 37 from the 53 basic logic functions of the original standard cell library [24] for our WDDL library.

Any combination of AND and OR operators and its dual, which is constructed with the help of the De Morgan's law (the AND and OR operators are interchanged and the input signals are inverted), will behave as a WDDL gate. The resulting compound gate: 1) is differential as it is constructed to be; 2) propagates the precharge wave as only positive operators are used; and 3) has a 100% switching factor as it is a dual gate consisting of only AND and OR operators. AND–OR–INVERT (AOI), XOR, MUX, etc. can all be implemented. By way of example, Fig. 3 (middle) and (left) shows the WDDL AOI32 gate with drive strength 2 and the original static CMOS gate. Compared with the use of negative differential logic, proposed in [25], the introduction of the inverters does not result in an area overhead. They act as buffers: while for a negative function, the transistors implementing the complex function must be made large, the drive strength now is provided by the inverters. A negative differential AOI32X2 gate, shown in Fig. 3 (right), is 10% larger than a WDDL AOI32X2 gate.

3) WDDL Load Capacitances: The condition that each compound standard cell has exactly one transition is a necessary condition, but it is not sufficient. The next condition is situated at the circuit level. Indeed, it is essential in order to achieve constant power consumption that a fixed amount of charge is used per transition. This means that the load capacitances at the differential output should be matched. The load capacitance has three components, namely: 1) the intrinsic output capacitance; 2) the interconnect capacitance; and 3) the intrinsic input capacitance of the load. The design of the individual WDDL gates only controls the intrinsic capacitances. Additional capacitances can be incorporated in the compound gates to balance the intrinsic capacitances. Or even custom-designed WDDL gates can be manufactured. Yet, with shrinking channel length of the transistors, the share of the interconnect capacitance in the total load capacitance increases and the interconnect capacitances will become the dominant capacitance [26]. Hence, the issue of matching the interconnect capacitances of the signal wires is crucial for the countermeasures to succeed [11], [12].

In [25], an alternating spacer protocol is proposed. Instead of always precharging to 0, the idea is to alternate precharging to 0 and precharging to 1. This approach requires the distribution of a separate precharge signal, with stringent timing requirements in relation to the original clock signal and the support of the inputs for the alternating spacer protocol. At first sight, this approach has a constant load capacitance independent of the load capacitance value at each of the differential outputs. During the transition from the 0-spacer to the evaluation phase, one of the outputs is charged. During the subsequent transition

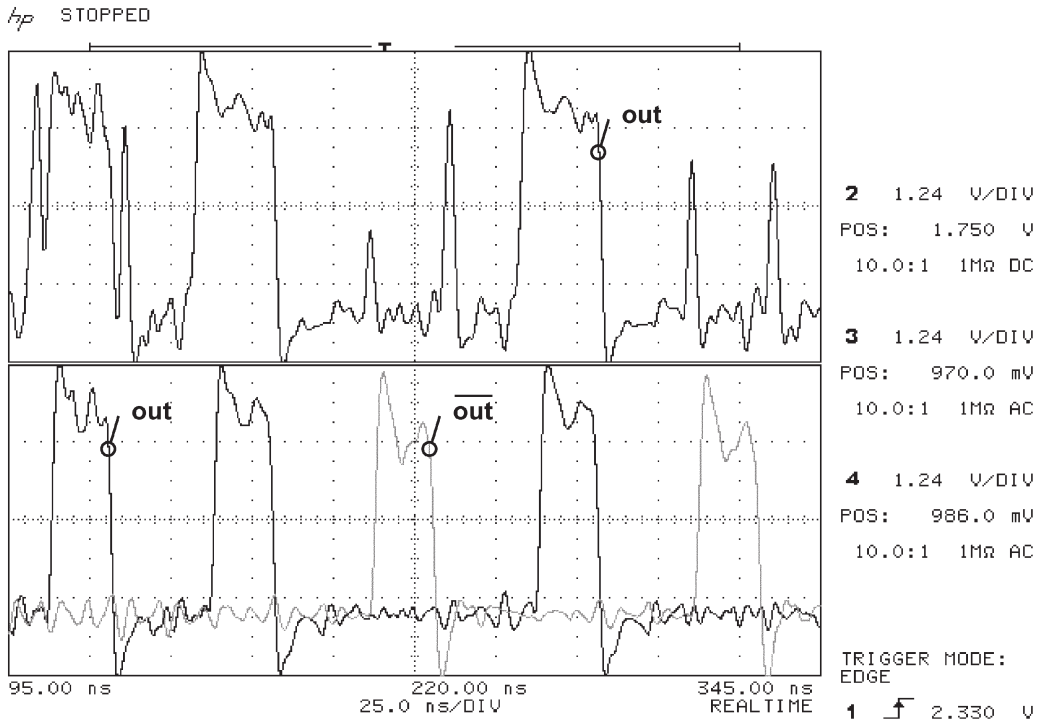

Fig. 2. Measurement of output transient: Single-ended design (top); and WDDL implementation (bottom).

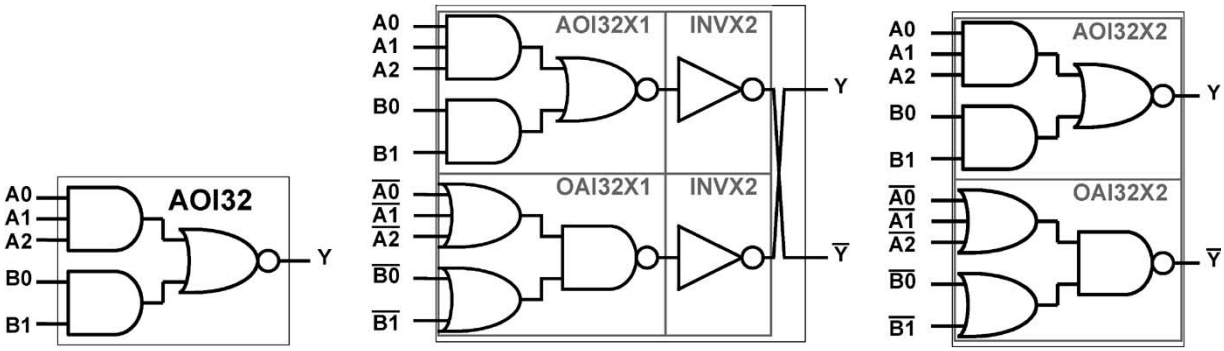

Fig. 3. AOI32 gate with drive strength 2 in: Static CMOS (left); WDDL (middle); and negative differential logic (right).

of the evaluation phase to the 1-spacer, the other output is charged. Note that it is perfectly possible to differentiate between the precharge and the evaluation phase in a measured supply current trace. Thus, it is sufficient for the attacker to only look at the transition from the 0-spacer to the evaluation phase. In order for the logic gate not to have a different power signature for each output event that is possible during this transition, the two output capacitances must be matched and routing differences may not exist between the two differential nets.

III. Matching Interconnect Capacitances of Dual Rail Logic

Matched interconnect capacitances can be obtained by routing the true and false output signals with parallel routes that are, at all times, in adjacent tracks of the routing grid, on the same layers, and of the same length. Then independent of the placement, the two routes have the same first-order parasitic effects.

The parasitic effects of the interconnects are caused by the distributed resistance and by the distributed capacitance to the substrate and to neighboring wires in other metal layers. Though aside from process variationsn, these effects are equal for both nets. The resistance is the same, since both interconnects have the same number of vias and have the same length in each metal layer. The capacitance to the other layers is ideally the same, since, in general, the length of the differential route is orders of magnitude larger then the pitch between the two differential routes and one can, therefore, argue that both nets travel in the same environment. Making every other metal layer a ground plane would completely control the capacitance to other layers. This reduces the solution space and increases the total capacitance.

The pair of interconnects, however, needs also be routed with control over any crosstalk. Crosstalk, which is the phenomenon of noise induced on one wire by a signal switching on a neighboring wire, has an effect on the power consumption. Crosstalk effects are caused by the distributed capacitance to adjacent

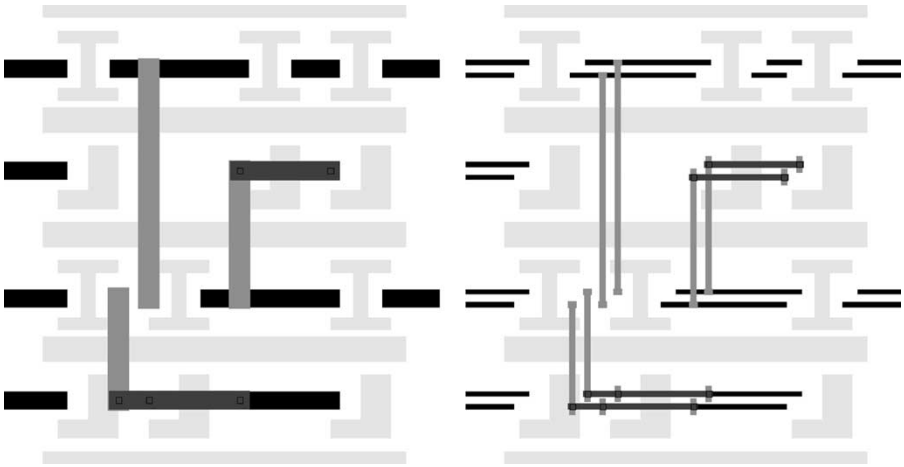

Fig. 4. Placed & routed design: Fat design (left); and differential design (right).

wires in the same metal layer. Routing the two output nets in parallel already removes the uncertainty of one neighbor: During a switching event, only one output line switches, the other output line remains quiet. All uncertainty can be removed by shielding the differential routes on either side with a VDD or VSS line. Reserving one grid line out of three upfront for a power line reduces the problem to routing two differential lines. Note that the approach of alternating signal lines and quiet power lines has been shown to produce predictable interconnect parasitics [28]. Alternatively, the crosstalk effects can be controlled by increasing the distance between different differential routes. As for any security application, there is a tradeoff between increased security and implementation costs, which are loss of routing tracks.

Differential pair routing has been available through gridless routers. However, their goal is to route a few critical signals, such as the clock or general reset signal. They are not built for crypto applications where all signals need a differential route, and, thus, router performance and completion rate degrade rapidly with increasing number of differential pairs. These tools are unable to route 20 000+ differential pairs as an encryption algorithm requires. An experiment with a mere 221 differential pairs required 7 h 56 min and 33 s in CPU time on a Sun ULTRA 5 for Cadence Chip Assembly Router version 11.0.06 [29] to perform 100 iterations without generating a completely routed result. It still had 972 conflicts and 125 unconnected nets. High-capacity gridded routers, on the other hand, have no or only limited capability to route differential pairs and often even avoid running wires in parallel to prevent crosstalk effects. We have recently presented a way to work around tool limitations [27]. The same experiment only required 3.85 s in CPU time to route the 221 differential pairs without any violations.

A. Differential Pair Routing

The technique is built on top of a commercial place & route tool and forces the tool to route the two output signals at all times in adjacent tracks. In the technique, each differential output pair is abstracted as a single fat wire. The differential design is routed with the fat wire, and at the end, the fat wire is decomposed into the differential wire. Fig. 4 demonstrates the place & route approach. The figure shows a placed & routed design consisting of six differential gates. On the left, the result is shown of the fat routing. On the right, the result after decomposition is shown. Two normal wires replace each fat wire.

The place & route tool cannot handle differential standard cells and fat interconnects at the same time. It is not possible to connect one single fat interconnect wire to two differential pins. The tool needs a fat gate level netlist and a fat gate library database. The fat gate level netlist is obtained from the differential gate level netlist by substituting each differential input and output signal by one single signal. The fat library database contains the routing rules that are applicable for the fat wires and the macro cell definition of the fat gates. A macro cell is a simplified representation of the standard cell [30]. It contains information such as height, width, and pin placement. The macros of the fat cells are obtained from the differential cells by abstracting the pins of the differential signals as one single pin.

In a postprocessing procedure, the fat wire is decomposed into the differential wires. This procedure, depicted in Fig. 5, consists of two translations of the fat wire and a width reduction to the normal width. The translation of the fat wires to the differential wires is done by editing the netlist that comes out of the router. The width reduction of the translated wires is accomplished by importing the edited netlist and a library database that contains the real macros of the differential gates and the routing rules for the normal wires into the router.

B. Matching Precision

The matching precision and optimization of the interconnect capacitances has to be in line with the quality and optimizations of the logic style. The intrinsic capacitances of the logic gates and the interconnect capacitances must have similar matching. There is no need for concentrating on balancing the intrinsic capacitances of the logic gates if the interconnect capacitances are not balanced and vice versa.

Fig. 6 plots the capacitances of the true signal nets versus the capacitances of the false signal nets for three cases, namely: 1) the input capacitances of our WDDL 0.18-μm library; 2) the interconnect capacitances of a DES substitution box

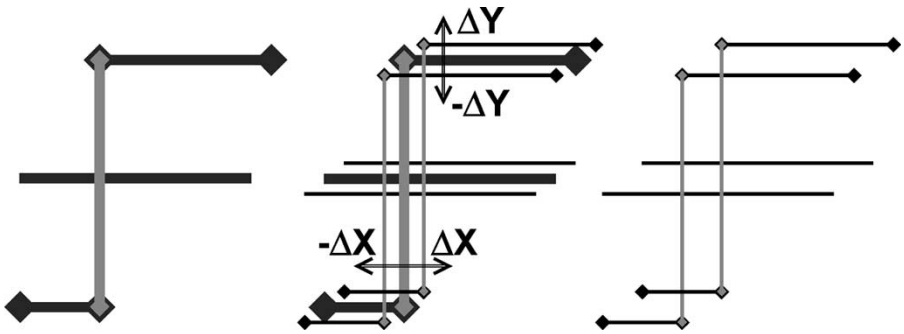

Fig. 5. Fat routes (left); translation operation (middle); and differential routes (right).

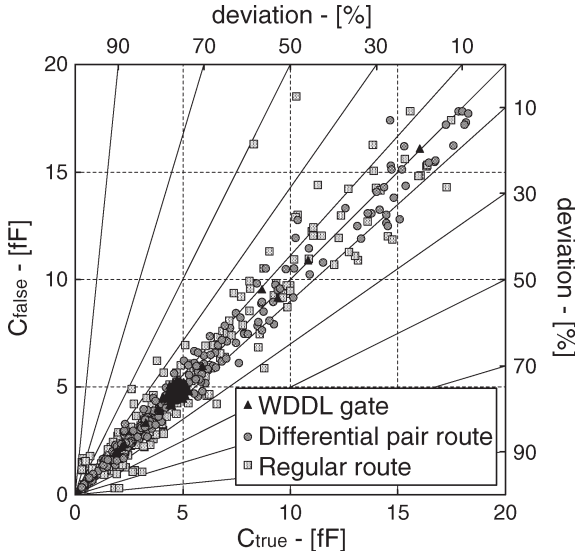

Fig. 6. Capacitances at true signal nets versus capacitances at corresponding false signal nets: input capacitances of WDDL gates in library; interconnect capacitances of DES module with differential pair routes; and interconnect capacitances of DES module with regular routes.

routed with differential pair routes; and 3) the interconnect capacitances of a DES substitution box routed with regular routes, without differential pair constraints. Both implementations in 2) and 3) have been routed with the same WDDL standard cell placement.

The interconnect capacitances have been extracted with the tool HyperExtract in Silicon Ensemble [31]. The lumped capacitance values are used. Ideally, for a 0% deviation between the true and false nets, the data points should be on a straight line. Note that the capacitances at the true and the corresponding false signal nets of the implementation with differential pair routes, directly reported from Silicon Ensemble using Simcap, have exactly the same values. In contrast with HyperExtract, Simcap does not report the second-order parasitics.

The variation on the interconnect capacitances after differential pair routing stays within 20%. The variation on the input capacitances of the WDDL gates is a maximum 10%. For a typical fanout of four logic gates, the absolute variation on the load capacitances due to the logic gates could, in a worst case event, add up to a similar variation as the variation due to the interconnect capacitances. Incorporating intrinsic capacitances or using custom WDDL cells will make the differential interconnects the limiting factor; shielding the differential routes or alternating ground and routing planes will make the WDDL cells the limiting factor that causes the principal share of the power variation. When using full custom logic styles, such as SABL, which have a symmetric design and balanced intrinsic capacitances and which even pay attention to the internal node capacitances, one or more of these techniques are mandatory to reduce the variation on the interconnect capacitances and to take advantage of their unique (dis)charging behavior.

The variation on the interconnect capacitances after genuine routing without differential pair constraints is up to 50%, and this is true for interconnects that have a capacitance value much larger than the input capacitances of the gates. A dual rail logic countermeasure will not succeed without differential pair routing. The failure of a fabricated prototype IC with asynchronous dual rail logic to provide a significant increase in DPA resistance has precisely been attributed to unbalanced signal paths caused by routing differences [12].

IV. Secure Digital Design Flow

The secure digital design flow [16] is depicted in Fig. 7. In addition to the regular steps in an IC design (logic design, logic synthesis, place & route, stream out, and verifications), one can recognize two additional steps, namely: 1) "cell substitution"; and 2) "interconnect decomposition." These operations have been inserted in the back end of the flow and do not interfere with the creative part of a design, indicated by the "logic design" task. We will now present an elaborate description of the secure digital design flow.

In the logic design phase, the design specifications (`design specs`) are translated into a behavioral model (`behavior.v`). A hardware description language, such as Verilog or very-high-speed integrated circuit (VHSIC) hardware description language (VHDL), is used to model the desired functionality.

Logic synthesis is the process of mapping the behavioral model (`behavior.v`) into logic gates of the library file (`lib.v`). It generates a gate level description of the desired circuit (`rtl.v`). The constraints file (`script`) contains area and timing optimization directives. Additionally, it restricts the gates used during synthesis. The gates available for synthesis

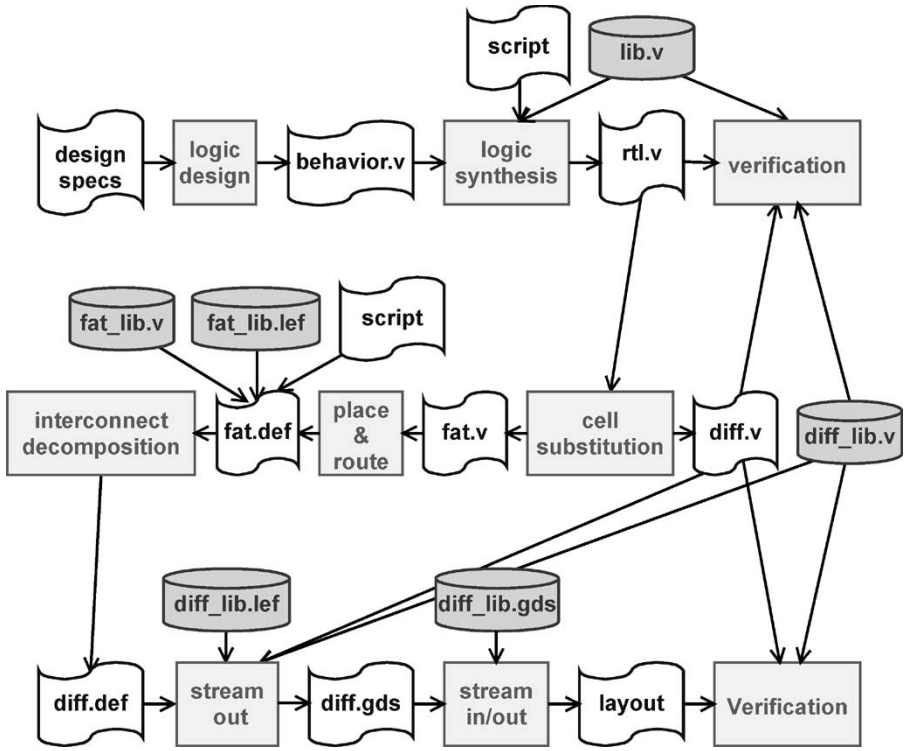

Fig. 7. Secure digital design flow.

depend on the WDDL cells that have been assembled. The minimum set consists of a register, an inverter, an AND gate, and an OR gate. Our WDDL library contains 128 distinct macro cells and implements 37 logic functions. The library file, however, is the original static CMOS standard cell library file.

The functionality and a preliminary timing of the gate level netlist (`rtl.v`) are verified with a gate-level simulation and a static timing analysis. This also requires the library file (`lib.v`) and is done in the verification step.

The cell substitution procedure modifies the gate level description. A script, e.g., in practical extraction report language (PERL) or in Awk, transforms the gate level netlist (`rtl.v`). Two files are generated, namely: 1) a fat gate level netlist (`fat.v`), which will be used to route the design and 2) a differential gate level netlist (`diff.v`), which will be used in the verification steps. The differential netlist is obtained by replacing each gate by its WDDL counterpart. This means that each net is duplicated, made differential, and connected to the differential pin. The inverters are also removed; the inversions are implemented by switching the nets. The fat netlist is equivalent to the differential netlist except that the differential signals have been abstracted as one single signal. This kind of parse procedure is not present in the regular design flow. The run time overhead, however, is negligible. The parser required a little less than 4 min to generate both files for the prototype IC containing 39 000 effective gates on a SunFire v100 [550 MHz CPU, 2 GB random access memory (RAM)].

In order to validate the result of the parser, a logic equivalence checker, such as Formality [32] or Verplex LEC [33], is used to verify the equivalence between the fat gate level netlist and the original netlist. The differential gate level netlist (`diff.v`) is used together with the differential (`diff_lib.v`) and the original (`lib.v`) library to verify that the design goals are met. Since the WDDL gates are compound gates, we have an accurate representation in function of the original gates. The verification step gives an estimate on the critical path delay and the area requirements. The verification step also includes a gate level simulation.

In the place & route step, the fat gate level netlist (`fat.v`) is placed and routed. The place & route tool requires the fat gate library database (`fat_lib.lef`), containing cell macros and routing rules, and a functional description of that library (`fat_lib.v`). The tool executes the commands file (`script`). This file contains the instructions for, among other things, floor planning, power planning, routing, etc. Note that the information from the original library files is used in procedures such as clock-routing- and timing-driven placement. The resulting design file (`fat.def`) specifies the location of the cells in the core and of the wires connecting the cells.

Clock routing changes the fat gate level netlist (`fat.v`). The new netlist contains the buffers from the clock tree and the original fat gate level netlist. The differential gate level netlist (`diff.v`) must also be updated with this information. The fat gate level netlist can be generated by the place & route tool. Parsing this file will result in the new differential gate level netlist. A logic equivalence test between the original and the new differential gate level netlist ensures correctness.

The wires in a .def design file are described as lines between two points and vias are assigned as points. The wire width and via characteristics are defined in the .lef library database. The fat to differential routing transformation consists of two

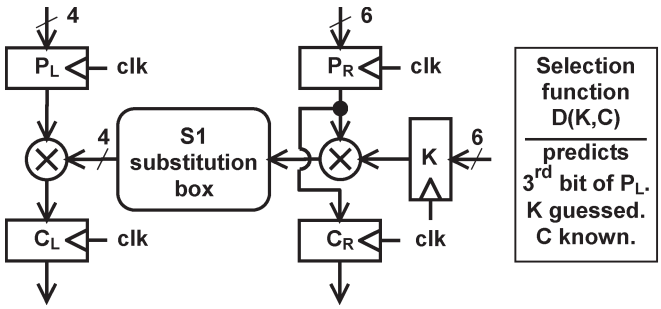

Fig. 8. Design example: DPA on a module of the DES encryption algorithm [11].

separate procedures, namely: 1) a duplication and translation of the fat wires and 2) a width reduction.

The interconnect decomposition procedure accomplishes the duplication and translation. This procedure edits the fat design file (`fat.def`). A parser duplicates and translates the coordinates of the points that represent the wire segments. This procedure is not present in the regular design flow. It has a negligible timing overhead. The parser required 2 min to generate the differential design file (`diff.def`) for our 39 000-gate IC.

The width reduction is accomplished by updating the library database with the differential library database during the stream out of the design. In this step, the differential design file (`diff.def`) and the differential library database (`diff_lib.lef`), which contains the normal wire definition and the differential gate macros, are combined in the place & route tool to generate the layout (`diff.gds`). The differential gate level netlist (`diff.v`) is only needed to verify the connectivity.

The parasitics are also extracted in this stage. They are used for delay annotation in a static and dynamic timing analysis. The verification produces accurate information as the extraction and delay annotation are done with the original library files that have been provided by the vendor.

The layout (`diff.gds`) only contains the macros of the standard cells. In order to update the macros with the actual standard cells (`diff_lib.gds`), an additional stream in/out procedure is required. The final stream file (`layout`) describes the mask layer information of the IC.

Once the final layout passes the final verification, which consists of layout versus schematic (LVS), electric rule check (ERC), design rule check (DRC), and antenna checks, it can be sent for tapeout.

V. Experimental Results

A. Design Example

A test circuit is implemented through the secure digital design flow and through a regular digital design flow using ordinary static CMOS standard cells. The block diagram of the test circuit is depicted in Fig. 8. This circuit has been presented as a sufficient subset of the DES encryption algorithm on which a DPA can be mounted [11]. The algorithm has been reduced to this setup such that the instantaneous supply current transient can be simulated with the transistor level simulator Hspice.

The single-ended gate level netlist has been obtained through DesignAnalyzer [34]. The place & route step has been done in Silicon Ensemble [31] with an aspect ratio of 1 and a fill factor of 80%. The language Awk is used in the parser to generate the fat and differential netlists. The layouts of the secure implementation and the reference implementation require 12 880 and 3782 μm^2, respectively. The spice netlists, which include the layout parasitics, have been extracted in Virtuoso [35]. In total, 2000 input vectors have been consecutively encrypted with a random input at the plaintext P_L and P_R, and with a fixed secret key K, equal to 46. The clock frequency of the circuit is chosen at 125 MHz. The sampling rate was 100 GHz.

The clock and input signals are driven by cascaded inverters to provide realistic data and clock signals. The power consumption of the additional input circuitry is excluded from the measurements. The mean energy consumption is 27.1 and 4.6 pJ for the secure implementation and the reference implementation, respectively. The normalized energy deviation, which is defined as the difference between the maximum and the minimum energy consumption per encryption divided by the maximum energy consumption per encryption, is 6.6% and 60%. The normalized standard deviation, which is the standard deviation on the energy consumption per encryption divided by the mean energy consumption per encryption, is 0.9% and 12%.

Fig. 9 shows the result from the DPA on the transient simulation. In a DPA [3], the supply current measurements of a large number of encryptions are divided over two sets by means of a selection function and a guess on the secret key. The difference between the typical supply currents of the two sets will approach zero for a wrong key guess, but has noticeable peaks if the correct secret key has been predicted. The selection function calculates a state bit of the encryption module. If the correct secret key has been used, the outcome is equal to the state bit and, hence, correlated with the power consumption of the logic operations that are affected by the state bit. The power consumption of the other logic operations and measurement errors, however, are uncorrelated. As a result, the difference, also referred to as differential trace, will approach the effect of the target bit on the power consumption, and there are noticeable peaks. If, on the other hand, the guess on the secret key was incorrect, the result of the selection function is uncorrelated with the state bit: The difference will approach 0.

The resistance against DPA is quantified with the number of measurements to disclosure (MTDs). This number expresses how many measurements are necessary to correctly distinguish the correct secret key from all the other wrong key guesses.

Fig. 9 (left) shows that for the reference design, 250 measurements are sufficient to disclose the secret key. The secure digital design flow, on the other hand, has been effective in reducing the peaks of the differential trace of the correct secret key: The peak-to-peak value of the secret key is conforming with the peak-to-peak value of the other key guesses. The DPA does not reveal the secret key. Fig. 9 (right) shows the peak-to-peak value of the differential traces of the secret key guesses for 2000 measurements. The secret key clearly stands out for the reference implementation.

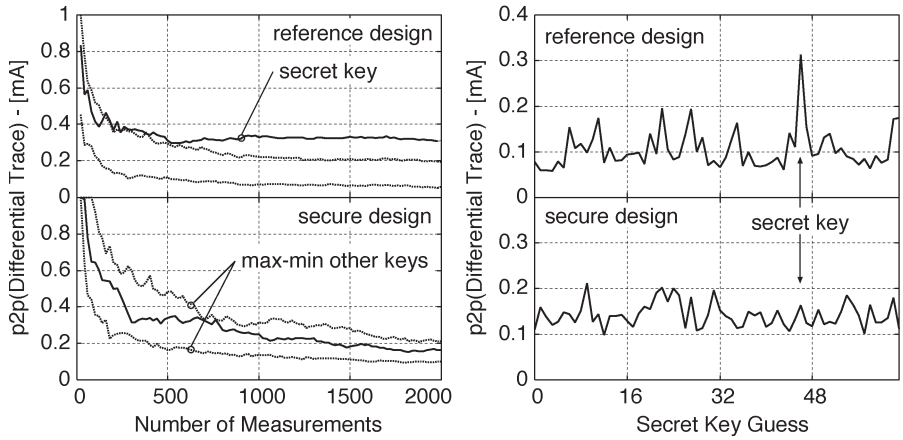

Fig. 9. Number of measurements to disclosure (left); and peak-to-peak value of the differential traces at 2000 measurements (right).

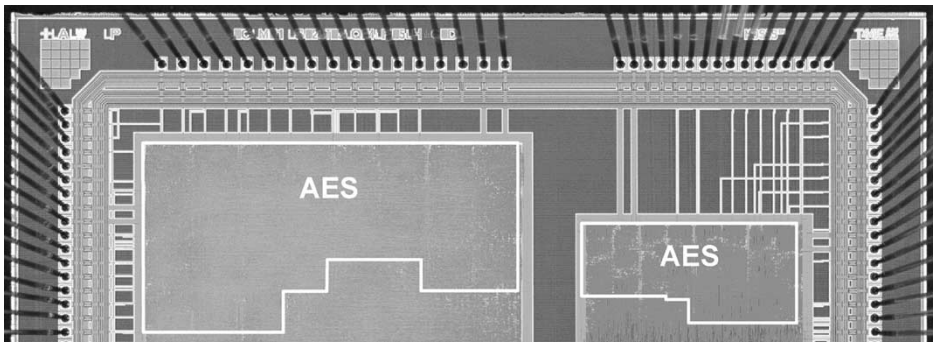

Fig. 10. IC micrograph: secure coprocessor using WDDL and differential routing (left); and insecure coprocessor using standard cells and regular routing (right).

B. Prototype IC

The secure digital design flow described in this paper is applicable to large realistic designs. It is part of a domain specific codesign methodology for secure embedded systems [38]. It implements the secure portion of a system-on-a-chip (SOC) design. The prototype IC, depicted in Fig. 10, consists of two functionally identical coprocessors and is fabricated on the same die using a TSMC 6M 0.18-μm process [37]. An insecure coprocessor serving as benchmark is implemented using standard cells and regular routing techniques. A secure coprocessor is implemented through the secure digital design flow using WDDL and differential routing. Both coprocessors have been implemented starting from the same synthesized gate level netlist. The WDDL compound gates have been derived from the Artisan SAGE-X 0.18-μm 1.8-V static CMOS standard cell library [24] that has been used in the regular insecure design.

The cryptographic engine is an AES core. The data path is based on a single round of the AES-128 algorithm. A full encryption of 128-bit data using a 128-bit key takes precisely 11 cycles. Fig. 11 shows transient measurements of the encryption start signal and the supply current of the coprocessors with the AES cryptographic engine in OFB mode. The supply current of the insecure coprocessor exhibits large variations. It broadcasts the 11 encryption rounds. The high-power peak at the starting point of each new encryption can be used as a synchronization signal. The power consumption profile of the secure implementation, on the other hand, is invariant and does not reveal any information in a simple power analysis. In each clock cycle, the same total load capacitance is charged. To facilitate the synchronization of the measurements, however, we have access to the encryption start signal.

We performed a correlation DPA attack [36] on each coprocessor as it executed AES to find the secret key byte per byte. For the insecure implementation, the correct key bytes are found very easily. On average, 2000 measurements are required to disclose a key byte. In one case, a mere 320 samples were sufficient to mount a successful attack. The secure coprocessor, on the other hand, substantially improves the DPA resistance. Our measurements show that out of 16 key bytes, WDDL effectively protects five key bytes. One and a half million measurements, which is larger than the lifetime of the secret key in most practical systems, are not sufficient to disclose the correct key bytes. The 11 key bytes that are found require, on average, 255 000 measurements, an increase of more than two orders of magnitude when compared with the insecure coprocessor.

Table I summarizes the results. WDDL and differential routing is a functional technique to thwart power attacks. The trade-off is a three times increase in area and a four times increase in power consumption and minimum clock period.

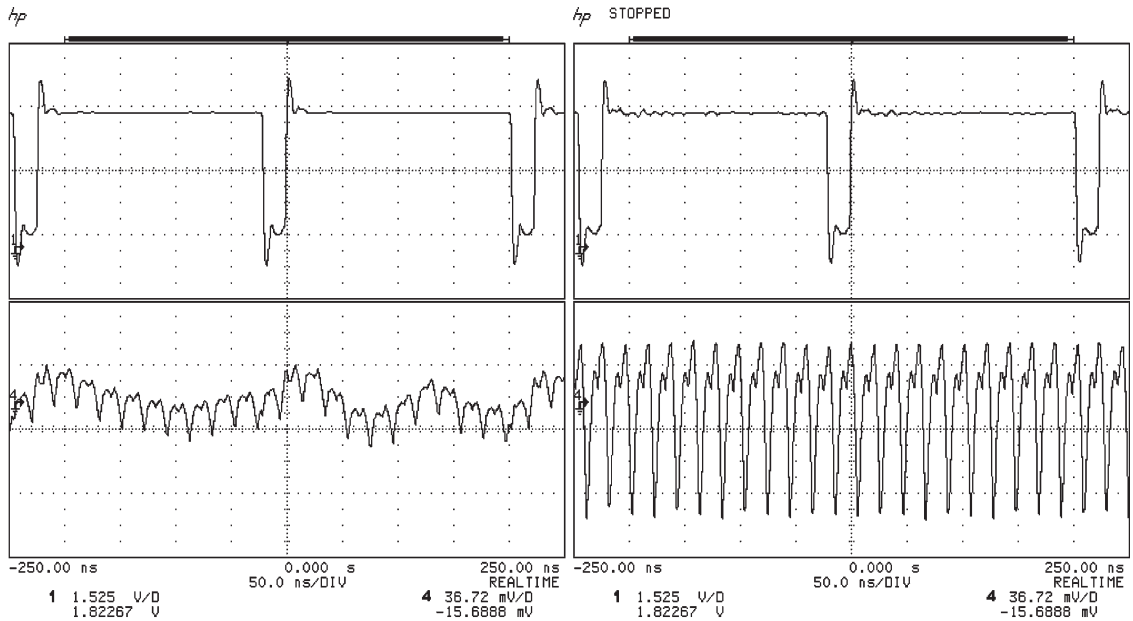

Fig. 11. Transient measurement (2 encryptions and 22 clock cycles) of encryption start signal (top) and core supply current (bottom): Insecure coprocessor (left); and secure coprocessor (right).

TABLE I
AES RESULTS SUMMARY

Parameter	Unprotected AES	Protected AES
Gate Count (eq. gates) [K]	79	245
Area [mm^2]	0.79	2.45
Maximum Frequency (@1.8V) [MHz]	330.0	85.5*
Maximum Throughput (@1.8V) [Gb/s]	3.84	0.99
Power Consumption (@1.8V, 50 MHz) [mW]	54	200†
Measurements to Disclosure‡		
min	320	21,185
mean	2,133	255,391
max	8,168	1,276,186
Key bytes not found (@1.5M Meas.)	n/a	5

*Duty factor of clock > 50% to guarantee precharge of all gates
†Estimation based on area ratio AES vs. Entire System
‡Based on correctly guessed key bytes

To our knowledge, the secure digital design flow is the first to deliver a working practical DPA countermeasure implemented and tested in actual silicon. All other published techniques have never been implemented in silicon, have never been measured and attacked, or did not offer any significant DPA resistance.

A dual rail asynchronous chip has been presented previously [12]. The implementation, however, did not provide a significant increase in DPA resistance. This failure has been attributed to unbalanced signal paths caused by routing differences. Note that if asynchronous logic is used to increase the DPA resistance, dual rail encoded asynchronous logic must be used. Because of the dual rail logic, there is also a factor three area increase compared with a single-ended synchronous benchmark [13].

Algorithmic countermeasures are mathematically DPA resistant. In practice, however, proposed solutions actually have been insecure [9], [10]. We are aware of one silicon implementation of an algorithmic countermeasure [39]. Measurements and assessment of the DPA resistance, however, have not yet been performed.

VI. CONCLUSION

We have presented a secure digital design flow. The design flow provides an accessible means to fabricate a security IC that is SCA resistant regardless of the implementation details. The approach is independent of the cryptographic algorithm implemented. It relies on a logic style that has constant power consumption and a place & route approach that controls the parasitic effects: WDDL has exactly one charging event per cycle and differential pair routing matches the interconnect capacitances of the true and false output signals. The design flow is completely supported by mainstream EDA tools and uses a commercially available static CMOS standard cell library. The differences with a regular synchronous CMOS standard cell design flow are minor. The secure digital design flow starts

from a normal design in a hardware description language and only a few key modifications with a minimal influence on the design flow are incorporated in the back end of the design flow. The additional steps required only a total of 6 min of CPU time for the prototype IC. A cell substitution phase and an interconnect decomposition phase parse intermediate design files. The former procedure modifies the gate level description, the latter duplicates and translates the interconnect wires. Measurement-based experimental results have demonstrated that it is a working practical technique to thwart power analysis attacks. It successfully protects AES on a prototype IC fabricated in an 0.18-μm CMOS. A DPA attack does not disclose the entire secret key at 1 500 000 measurements, which is larger than the lifetime of the secret key in most practical systems.

ACKNOWLEDGMENT

The authors acknowledge D. Ching, A. Hodjat, D. Hwang, B.-C. Lai, Y. Matsuoka, P. Schaumont, and S. Yang for their effort in the design of the ThumbPodII chip. This work was performed while the authors were at UCLA.

REFERENCES

[1] E. Hess, N. Janssen, B. Meyer, and T. Schuetze, "Information leakage attacks against smart card implementations of cryptographic algorithms and countermeasures—A survey," in *Proc. Eurosmart Security Conf.*, Marseille, France, 2000, pp. 55–64.

[2] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Proc. Advances Cryptology (CRYPTO)*, Santa Barbara, CA, 1996, vol. 1109, pp. 104–113.

[3] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Advances Cryptology (CRYPTO)*, Santa Barbara, CA, 1999, vol. 1666, pp. 388–397.

[4] J. Quisquater and D. Samyde, "ElectroMagnetic analysis (EMA): Measures and counter-measures for smart cards," in *Proc. Smart Card Programming and Security (E-smart)*, Cannes, France, 2001, vol. 2140, pp. 200–210.

[5] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, and S. Ravi, "Security as a new dimension in embedded system design," in *Proc. 41st Design Automation Conf. (DAC)*, San Diego, CA, 2004, pp. 753–760.

[6] S. Ravi, A. Raghunathan, and S. Chakradhar, "Tamper resistance mechanisms for secure, embedded systems," in *Proc. 17th Int. Conf. Very Large Scale Integration Design (VLSID)*, Mumbai, India, 2004, pp. 605–610.

[7] J. Daemen and V. Rijmen, "Resistance against implementation attacks: A comparative study of the AES proposals," in *Proc. 2nd Advanced Encryption Standard (AES) Candidate Conf.*, Rome, Italy, 1999, pp. 122–132. [Online]. Available: http://csrc.nist.gov/encryption/aes/round1/conf2/aes2conf.htm

[8] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *Proc. Advances Cryptology (CRYPTO)*, Santa Barbara, CA, 1999, vol. 1666, pp. 398–412.

[9] E. Oswald, S. Mangard, and N. Pramstaller, "Secure and efficient masking of AES—A mission impossible?" IACR Cryptology ePrint Archive, Santa Barbara, CA, Rep. 2004/134, Jun. 2004

[10] S. Mangard, T. Popp, and B. Gammel, "Side-channel leakage of masked CMOS gates," in *Cryptographers' Track—RSA Conf. (CT-RSA)*, San Francisco, CA, Feb. 2005, pp. 351–365.

[11] K. Tiri and I. Verbauwhede, "Securing encryption algorithms against DPA at the logic level: Next generation smart card technology," in *Proc. Cryptographic Hardware and Embedded Systems (CHES)*, Cologne, Germany, 2003, vol. 2779, pp. 125–136.

[12] J. Fournier, S. Moore, H. Li, R. Mullins, and G. Taylor, "Security evaluation of asynchronous circuits," in *Proc. Cryptographic Hardware and Embedded Systems (CHES)*, Cologne, Germany, 2003, vol. 2779, pp. 137–151.

[13] S. Moore, R. Anderson, R. Mullins, and G. Taylor, "Balanced self-checking asynchronous logic for smart card applications," *J. Microprocess. Microsyst.*, vol. 27, no. 9, pp. 421–430, Oct. 2003.

[14] L. Plana, P. Riocreux, W. Bainbridge, A. Bardsley, S. Temple, J. Garside, and Z. Yu, "SPA—A secure amulet core for smartcard applications," *J. Microprocess. Microsyst.*, vol. 27, no. 9, pp. 431–446, Oct. 2003.

[15] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Proc. Design, Automation and Test Eur. Conf. (DATE)*, Paris, France, 2004, pp. 246–251.

[16] ——, "A VLSI design flow for secure side-channel attack resistant ICs," in *Proc. Design, Automation and Test Eur. Conf. (DATE)*, Munich, Germany, 2005, pp. 58–63.

[17] M. Renaudin, F. Bouesse, P. Proust, J. P. Tual, L. Sourgen, and F. Germain, "High security smartcards," in *Proc. Design, Automation and Test Eur. Conf. (DATE)*, Paris, France, 2004, pp. 228–232.

[18] J. Yoshida. (2004). "Smart card designers need security tools," *EEtimes*. [Online]. Available: http://www.eedesign.com/showArticle.jhtml?articleID=17701143

[19] SCA Resistant Design (SCARD), *6th Framework Program of the European Comission Sponsored Research Project*. (2004). [Online]. Available: http://www.scard-project.org/

[20] *Opensmartcard*. (2004). [Online]. Available: http://www.comelec.enst.fr/recherche/opensmartcard/

[21] H. Ng and D. Allstot, "CMOS current steering logic for low-voltage mixed-signal integrated circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 5, no. 3, pp. 301–308, Sep. 1997.

[22] K. Tiri and I. Verbauwhede, "A dynamic and differential CMOS logic style to resist power and timing attacks on security IC's," IACR Cryptology ePrint Archive, Santa Barbara, CA, Rep. 2004/066, Feb. 2004.

[23] K. Tiri, M. Akmal, and I. Verbauwhede, "A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards," in *Proc. Eur. Solid-State Circuits Conf. (ESSCIRC)*, Florence, Italy, 2002, pp. 403–406.

[24] Artisan SAGE-X Standard Cell Library. [Online]. Available: http://www.artisan.com

[25] D. Sokolov, J. Murphy, A. Bystrov, and A Yakovlev. (2004). *Improving the Security of Dual-Rail Circuits*. [Online]. Available: http://www.staff.ncl.ac.uk/i.g.clark/async/tech-reports/NCL-EECE-MSD-TR-2004-101.pdf 2004

[26] ITRS. (2003). "Interconnect," *The International Technology Roadmap for Semiconductors*. [Online]. Available: http://public.itrs.net/Files/2003ITRS/Interconnect2003.pdf

[27] K. Tiri and I. Verbauwhede, "Place and route for secure standard cell design," in *Proc. Smart Card Research and Advanced Application IFIP Conf. (CARDIS)*, Toulouse, France, 2004, pp. 143–158.

[28] S. Khatri, A. Mehrotra, R. Brayton, A. Sangiovanni-Vincentelli, and R. Otten, "A novel VLSI layout fabric for deep sub-micron applications," in *Proc. Design Automation Conf. (DAC)*, New Orleans, LA, 1999, pp. 491–496.

[29] *Cadence Chip Assembly Router*. [Online]. Available: http://www.cadence.com/products/custom_ic/chip_assembly

[30] *LEF/DEF Language Reference 5.5*. (2003, Jan.). [Online]. Available: http://www.openeda.org

[31] *Silicon Ensemble*. [Online]. Available: http://www.cadence.com/products/digital_ic/sepks

[32] *Formality*. [Online]. Available: http://www.synopsys.com/products/verification/

[33] *Verplex LEC*. [Online]. Available: http://www.cadence.com/products/functional_ver/index.aspx

[34] *Design Analyzer*. [Online]. Available: http://www.synopsys.com/products/logic/deanalyzer_ds.html

[35] *Virtuoso*. [Online]. Available: http://www.cadence.com/products/custom_ic/index.aspx?lid=custom_ic_design

[36] J. Coron, P. Kocher, and D. Naccache, "Statistics and secret leakage," in *Financial Cryptography (FC)*, Anguilla, British West Indies, Feb. 2000, vol. 1962, Lecture Notes in Computer Science, pp. 157–173.

[37] K. Tiri, D. Hwang, A. Hodjat, B. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "AES-based cryptographic and biometric security coprocessor IC in 0.18-μm CMOS resistant to side-channel power analysis attacks," in *Symp. Very Large System Integration (VLSI) Technology and Circuits*, Kyoto, Japan, Jun. 2005, pp. 216–219.

[38] P. Schaumont and I. Verbauwhede, "Domain-specific codesign for embedded security," *IEEE Computer*, vol. 36, no. 4, pp. 68–74, Apr. 2003.

[39] N. Pramstaller, F. Gürkaynak, S. Häne, H. Kaeslin, N. Felber, and W. Fichtner, "Towards an AES crypto-chip resistant to differential power analysis," in *30th Eur. Solid-State Circuits Conf. (ESSCIRC)*, Leuven, Belgium, Sep. 2004, pp. 307–310.

Kris Tiri (S'99–M'06) was born in Bree, Belgium, in 1976. He received the M.S. degree in electrical engineering from the Katholieke Universiteit Leuven, Leuven, Belgium, in 1999, and the Ph.D. degree in electrical engineering from the University of California, Los Angeles, in 2005. His doctoral research focused on the design for side-channel attack resistant security integrated circuits (ICs).

He is currently with the Trusted Platform Laboratory of Intel Corporation, Hillsboro, OR. From 1999 to 2005, he was a Research Assistant with the Electrical Engineering Department of the University of California, Los Angeles. During the spring of 1999, he was with the COMELEC of Ecole Nationale Supérieure des Télécommunications, Paris, France. During 2001 and 2002, he was with IMEC, Heverlee, Belgium.

Dr. Tiri was awarded a Francqui Foundation Fellowship by the Belgian American Educational Foundation, in 1999, and he received the 2005 EDAA Outstanding Dissertation Award.

Ingrid Verbauwhede (M'92–SM'00) received the M.S. degree in electrical engineering, in 1984, and the Ph.D. degree in applied sciences, in 1991, both from the Katholieke Universiteit Leuven, Leuven, Belgium.

She was a Lecturer and Visiting Research Engineer at the University of California, Berkeley, from 1992 to 1994. From 1994 to 1998, she was a Principal Engineer first with TCSI and then with Atmel, Berkeley, CA. She joined UCLA in 1998 as an Associate Professor and the Katholieke Universiteit Leuven, in 2003. Her interests include circuits, processor architectures, and design methodologies for real-time embedded systems in application domains such as cryptography, security, digital signal processing, wireless, and high-speed communications.

Dr. Verbauwhede is or was a member of several program committees, including DAC, ISSCC, DATE, CHES, ICASSP. She is the design community chair on the 42nd and 43rd DAC executive community. More details of her embedded security research group can be found at www.emsec.ee.ucla.edu and www.esat.kuleuven.be/cosic.