PLACE AND ROUTE FOR SECURE STANDARD CELL DESIGN

Kris Tiri and Ingrid Verbauwhede
UC Los Angeles

Abstract: Side channel attacks can be effectively addressed at the circuit level by using dynamic differential logic styles. A key problem is to guarantee a balanced capacitive load at the differential outputs of the logic gates. The main contribution to this load is the capacitance associated with the routing between cells. This paper describes a novel design methodology to route a design in which multiple differential pairs are present. The methodology is able to route 20K+ differential routes. The differential routes are always routed in adjacent tracks and the parasitic effects between the two wires of each differential pair are balanced. The methodology is developed on top of a commercially available EDA tool. It has been developed as part of a secure digital design flow to protect security applications against Differential Power Analysis attacks. Experimental results indicate that a perfect protection is attainable with the aid of the proposed differential routing strategy.

Key words: Place & Route, Differential Pair, Differential Power Analysis, Side Channel Attacks

1. INTRODUCTION

Much design effort is spent in developing secure protocols and selecting strong encryption algorithms to achieve the security level envisioned in the specifications of the smart card application. Any security application however, is only as safe as its weakest link. Information related with the physical implementation of the device, such as variations in time delay and power consumption, has been used repeatedly to find the secret key in so-called Side Channel Attacks [1]. Especially the Differential Power Analysis (DPA)

[2] is of great concern as 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 retrieve the information from the power consumption that is correlated to the secret key.

At first, DPA has been thwarted with ad hoc countermeasures, which essentially concealed the supply current variations. Examples are for instance the addition of random power consuming operations or a current sink. Yet, the attacks have evolved and become more and more sophisticated. To address the problem, countermeasures need to be provided at different design abstraction levels. At the algorithmic level, an example is masking [3]. This technique prevents intermediate variables to depend on an easily accessible subset of the secret key.

Only recently, at the circuit level, dedicated hardware techniques have been proposed [4],[5]. Instead of concealing or decorrelating the side channel information, these techniques aim at not *creating* any side channel information. Goal of these countermeasures is to balance the power consumption of the logic gates. When the power consumption of the smallest building block is a constant and independent of the signal activity, no information is leaked through the power supply and a DPA is impossible.

Both in the synchronous [4] and in the asynchronous [5] approach, dynamic differential logic is employed. In this logic, every signal transition, also e.g. a degenerated 0 to 0 transition, is represented with an actual switching event, in which the logic gate charges a capacitance. Besides a 100% switching factor, 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 3 components: the intrinsic output capacitance, the interconnect capacitance and the intrinsic input capacitance of the load. Through a careful layout of the standard cells, the intrinsic input capacitances of a gate can be matched, as well as the intrinsic output capacitances. 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. Hence, the issue of matching the interconnect capacitances of the signal wires is crucial for the countermeasures to succeed [4],[5]. To our knowledge, this publication is the first to address this problem.

As we will derive in section 3, the best strategy to achieve matched interconnect capacitances is to route the output signals differentially. It is important to note that in this manuscript, differential routing denotes that the 2 output signals are at all times routed in adjacent tracks. Yet, this is the very opposite of current commercial cross-talk aware routers, which precisely

avoid running wires in parallel. As a response, we have elaborated a technique to force commercial EDA tools to route multiple differential pairs. In this technique, each output pair is routed as 1 'fat' wire, which has among other characteristics the width of 2 parallel wires. Afterwards, the fat wires are split into the 2 differential lines.

Differential pair and shielded routing has been available through shape-based routers whose antecedents are in the PCB domain, where electrical constraints are historically more dominant. However, router performance and completion rate degrade rapidly with increasing number of such constraints. Gridded routers, with both routing resource representation and heuristic search optimized for speed and capacity, are very difficult to adapt for connection of 'wide wires' or 'co-constrainted' wires [6]. Recently, shielded routing capability has been migrated to gridded routers. We cannot directly use this approach for differential pair routing because (1) the two parallel wires are not VDD or VSS line (that are typically used for shielding); (2) the spacing between two differential wires will be larger due to the shielded signal wire in the middle; and (3) the two differential wires cannot be guaranteed to have similar length. Therefore, we present a way to work around tool limitations in section 3.

The remainder of this paper is organized as follows. In section 2, a place & route approach is developed in order to thwart DPA. Section 3 discusses the differential routing technique. In section 4, an experiment is setup in which the technique is applied to route the DES encryption algorithm and results of a DPA are provided. Finally a conclusion will be formulated.

2. BALANCING INTERCONNECT LOADS

2.1 Given Information

A standard cell of a dynamic differential cell library has 2 differential outputs A and A', which connect to k differential input pairs $\{(I_1, I_1'), \ldots (I_k, I_k')\}$. Each standard cell has a balanced design. This means that (1) the intrinsic output capacitances $C_{o,A}$ and $C_{o,A'}$ seen at outputs A and A' are equal; that (2) the intrinsic input capacitances $C_{i,Ij}$ and $C_{i,Ij'}$ seen at inputs I_j and I_j' are equal; and that (3) the drive strengths at output A and A' are equal. The standard cell has exactly 1 switching event per cycle. This event is always the same and consists of 2 transitions: (1) in the evaluation phase: both outputs are at 1 and 1 output discharges to 0; and (2) in the precharge phase: 1 output is at 1; the other output is at 0 and is charged to 1.

2.2 Place & Route Constraints

2.2.1 Match Load Capacitance

In addition to engaging in 1 charging event per clock cycle, it is mandatory for input independent power consumption that the load capacitance is a constant. The differential standard cell has a load capacitance at each output. Since only 1 output undergoes a transition per switching event, the total load at output A should match the total load at output A': $C_A = C_{A'}$. This, as can be seen in Figure 1, can be restated as: $C_{o,A} + C_{w,A} + C_{i,I1} + \ldots C_{i,Ik} = C_{o,A'} + C_{w,A'} + C_{i,I1'} + \ldots C_{i,Ik'}$, which can be reduced to $C_{w,A} = C_{w,A'}$ because of the balanced standard cell design. This means that the Steiner routing tree over $\{A, I_1, \ldots I_k\}$ must have the same total capacitance as the Steiner routing tree over $\{A', I_1', \ldots I_k'\}$.

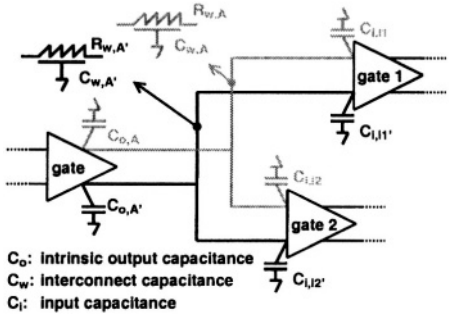

Figure 1. Load capacitance decomposition.

It is not necessary to balance the routing tree over $\{A, I_1, \ldots I_k\}$ with the routing tree over $\{B, I_m, \ldots I_s\}$ (where A and B are the outputs of two different gates). For the total encryption module to have constant power consumption, it is sufficient that the power consumption of each building block is input independent. There is no need for mutually matching the routing trees.

Cross-talk, which is the phenomenon of noise induced on one wire by a signal switching on a neighboring wire, influences not only the delay, but also the power consumption. Therefore the pair of interconnects need to be routed with the same capacitance and with control over any cross-talk.

2.2.2 Match Source-Sink Delays

An attacker will analyze all information available from the power consumption. He/she will not settle for the total charge per switching event, but will trace the instantaneous power consumption. To assure that only minor

instantaneous current variations exist between different switching events, each pair of differential routes should have a constant source-sink delay: the delay from A to I_i must be equal to the delay from A' to I_i' for each i = 1, ... k. It is not necessary that the delay from A to I_i is equal to the delay from A to I_j (where A is connected to the inputs I_i of gate B and I_j of gate C), nor that the delay from A to I_i is equal to the delay from B to I_j, (where the gates A and B are connected to the inputs I_i and I_j of gate C).

2.2.3 Miscellaneous Constraints

Implementations of encryption algorithm generally require 20K+ gates. This means that 20K+ differential routes must be balanced. It might be possible to only implement the most sensitive parts of the encryption module and reduce this number. Yet, the size of the problem will not be reduced with an order of magnitude.

The differential routing procedure must complete the missing part of a secure digital design flow [7]. This design flow does not restrict the fanout, which is the number of gates a gate connects to.

2.3 Place & Route Approach

If two gates are connected with parallel routes that are at all times in adjacent tracks and on the same layers, then since independent of the placement, the geometric distances are balanced, the two routes have to a first order the same capacitances and the same delays. Yet, this is not completely true because both nets may have different parasitics and cross-talk effects.

Parasitic effects are caused by the distributed resistance of the interconnect and by the distributed capacitance of the interconnect to the substrate and to neighboring wires in other metal layers. Though aside from process variations, 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 2 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. Yet this would not only reduce the solution space but also increase the total capacitance.

Cross-talk effects are caused by the distributed capacitance to adjacent wires in the same metal layer. Routing the two output nets in parallel 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. Besides the loss in routing tracks, it will also be hard to get multiple sets of 4 routes into and out a standard cell. Yet, reserving 1 grid line out of 3 upfront for a power line reduces the problem again to routing 2 differential lines. Note that the approach of alternating signal lines and quiet power lines has been shown to produce predictable interconnect parasitics [8]. Alternatively, the cross-talk effects can be controlled by merely increasing the distance between different differential routes. This can easily be done with the differential routing methodology we will present now.

3. DIFFERENTIAL ROUTING

The methodology that we propose is to abstract the differential pair 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.

3.1 Basic Ideas

3.1.1 Fat Wire Definition

The fat wire covers the two differential wires. The centerline of the fat wire is the centerline between the two differential wires. The width of the fat wire W_f is set by the summation of the pitch P_n of the normal wires and 2 times half the width of the normal wire W_n: $W_f = P_n + 2W_n/2$. The pitch, which is the distance between the centerline of two adjacent wires, P_f of the fat wires is set by the summation of 2 times half the width of the fat wire and the desirable distance Δ between the fat wires: $P_f = 2W_f/2 + \Delta$. The distance Δ can be made large to reduce cross-talk effects. The minimum spacing rules do not change.

3.1.2 Transformation

After place & route with the fat wire, the resulting design must be transformed into the final differential design. The transformation consists of 2 translations of the fat wire and a width reduction to the normal width.

Since the centerline between two normal wires is the centerline of the fat wire, a translation of the fat wire in the positive direction will result in one differential line and a negative translation in the in the other line. The translation must occur both in the horizontal and the vertical direction. As shown in Figure 2, a consistent shift of all segments of the fat wire with a ΔX in the X direction and a ΔY in the Y direction will result in one wire; a shift with a

$-\Delta X$ and a $-\Delta Y$ in the other wire. The shifts ΔX and ΔY are half the pitch lengths of the normal wires in the X and Y direction.

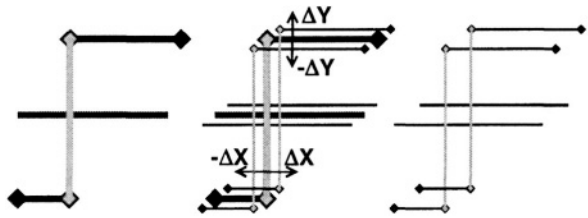

Figure 2. Fat routes (left); translation operation (middle); and differential routes (right).

The resulting differential wires have the same number of vias and segments. Each segment has the same length in both wires and is routed over the same number of wires in the other metal layers. As a result, both lines have the same distributed resistances and parasitic capacitances to the substrate and to the routes in the other metal layers.

3.2 Practical Aspects

This section discusses some of the practical issues we have come across. While these issues are independent of the tool, the guidelines presented are for Silicon Ensemble [9], which we have used as the place & route tool.

3.2.1 Restrictions on Differential Standard Cell

As can be seen in Figure 2, the vias are all aligned on a positive tilted diagonal. The in- and output pins of the standard cells must also be aligned likewise and with the same offsets. The upper pin is the pin associated with the true net, the lower with the false net. Only then the translation can be done in a consistent way.

3.2.2 Fat Wire Definition

Silicon Ensemble extends routes with at least half the width at their endpoints. Because of the increased width, the fat wire extends too far at its endpoints and covers an area where there is no actual normal route. As a result, spacing errors are generated for certain patterns of wires, which are only virtual errors. To address this problem, the original normal wire is routed on a large grid that has been defined such that there will be no spacing violations after splitting. Doubling the original grid pitches results in such a grid. Now, the .lef library database [10], which contains all the infor-

mation that is relevant for the router tool, such as routing layers, via rules, grid definition, spacing rules and abstract views of the standard cells, can be left unchanged except for a new grid definition and the abstract views of the cells with fat pin information. Note that from now on, we will route a normal wire, but we will continue to refer to this wire as the fat wire.

3.2.3 Grid Definition

To facilitate the placement, the height and width of a standard cell should be a multiple of the horizontal and vertical pitch respectively. In addition, since we can only route on the grid the pins should be situated on the grid crossings. The most straightforward is to take the pitch of the fat design a multiple of the original grid. The minimum pitch between the fat wires is 2 times the pitch between the normal wires.

We defined the grid and the standard cells as follows: (1) the horizontal and vertical pitches of the fat grid are double the ones of the normal grid; and (2) the normal and fat grids have an offset of half their pitch length in both the horizontal and vertical direction. With this definition all requirements previously derived are fulfilled: (1) the standard cell dimensions are multiples of the horizontal and vertical pitch of the fat and the normal grid; (2) the fat pins are situated on the crossings of the fat grid, the differential pins on the crossings of the normal one; and (3) the differential pins can obtained by shifting the fat pin with half a pitch length of the normal grid in both the horizontal and vertical direction.

3.2.4 Non-preferred Routing Direction

If the fat wire takes a turn in a metal layer, the wires of a differential route may cross in the same metal layer and result in an electric short between both wires. This however, can only happen if a metal layer is used in the vertical and the horizontal direction. Even though each metal layer has a preferred routing direction, this does not guarantee that the routing layer is only used in that direction. This required us to force Silicon Ensemble to only route in the preferred direction.

3.2.5 Transformation Procedure

The transformation procedure consists of two parts: (1) parsing the placed and routed fat design to reflect the differential design and (2) reading in the differential library database. The differential 'diff.lef' library database contains the normal grid definition, normal wire definition, normal via definition and the differential gates with differential pin information.

The wires in the routed 'fat.def' design file are described as lines between 2 points and vias are assigned as points. The wire width and via characteristics are defined in the .lef library database. As a result the parser only needs to translate the (X,Y) coordinates of the end points without to worry about the wire characteristics. The translation is done by (1) repeating each statement that defines a net; (2) attaching the first statement to the positive pins and translating it in a positive $(\Delta X,\Delta Y)$ direction; and (3) attaching the second statement to the negative pins and translate it in a negative $(\Delta X,\Delta Y)$ direction. Recall that ΔX and ΔY are half the pitch lengths of the normal wires in the X and Y direction. Besides the translation of the nets, each fat gate in the 'fat.def' file is substituted by its corresponding differential gate.

Figure 3 summarizes the differential routing methodology.

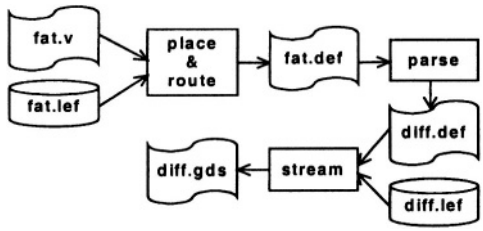

Figure 3. Differential routing methodology.

3.2.6 Differential and Single Ended Routing

Up to now, we only presented a methodology to route a design of which all wires are differential. It is however possible to combine single ended routing and differential routing. There are 3 options. The design can be routed in 2 stages. First the differential lines are routed, and subsequently with a new library database the single ended lines are routed, or visa versa. The design can also be routed concurrently by defining the fat routes or the single ended routes as nondefault routing rules. Or, one could route every wire as a fat wire and subsequently transform the single ended signals into a single line and the differential signals into 2 lines. The last option is not preferred if the single ended routes are in the majority and area constraints are tight.

3.3 Design Example

Figure 4 demonstrates the differential routing technique. The figure shows an arbitrary placed and routed design consisting of 7 differential

gates. At the left, the result is shown of the fat routing. At the right, the result after transformation is shown. Each fat wire is replaced by 2 normal wires.

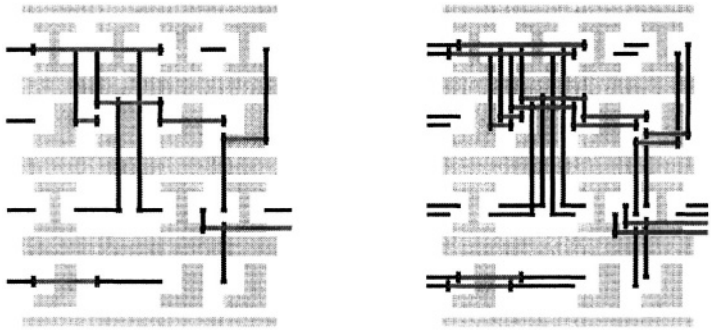

Figure 4. Example: fat (left); and differential design (right).

4. EXPERIMENTAL RESULTS

In this section, we compare two dynamic differential implementations of the DES algorithm [11]. One implementation uses the differential routing technique described above. For the other implementation, we use the default standard cell route as generated by the EDA router. We mounted a DPA on both differential implementations. We have chosen DES, because much research has focused on both the implementation and prevention of DPA on DES. The first subsection presents the basic procedures of the DPA. Then the experimental setup is discussed and implementation details are given. The last subsection covers the experimental results. Please note that a DPA attack on a regular (non differential) standard cell implementation is described in [4].

4.1 Differential Power Analysis

A DPA attack is carried out in several stages. First, the power consumption is recorded for a large number of encryptions. This is done by measuring the instantaneous supply current. Next, the measurements are partitioned over 2 sets based on a so-called selection function, which uses a guess on a subset of the secret key to make its decision. At the end, the difference is calculated between the typical supply currents of the 2 sets. The difference is referred to as Differential Trace (DT). If the DT has noticeable peaks, the guess on the secret key was correct.

The selection function is a behavioral model of the encryption module and predicts 1 state bit of the module. If the secret key has been guessed correctly, the outcome of the selection function is always equal to the actual state bit and is therefore correlated with the power consumption of the logic operations that are affected by the state bit. Measurement errors and the power consumption of the other logic operations are uncorrelated. As a result, the DT will approach the effect of the state bit on the power consumption. 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 and the DT will approach 0.

4.2 Experimental Setup

The experimental setup is depicted in Figure 5 [4]. The module forms part of the DES encryption algorithm. Based on 4 bits of the left plaintext $\mathbf{P_L}$, 6 bits of the right plaintext $\mathbf{P_R}$ and 6 bits of the secret key K, it calculates 4 bits of the left ciphertext $\mathbf{C_L}$. The selection function D(K,C), which is used in the DPA, predicts the first bit of the register that stores the left plaintext $\mathbf{P_L}$. In the selection function, only K is guessed, $\mathbf{C_L}$ and $\mathbf{C_R}$ are known.

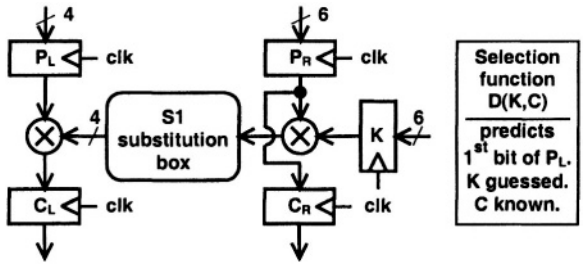

Figure 5. Experimental setup: DPA on submodule of the last round in the DES-algorithm [4].

The experimental setup is a necessary and sufficient subset of the DES encryption algorithm on which a DPA can be mounted [4]. The algorithm has been reduced to this setup such that it becomes computationally feasible to simulate it with Hspice. High-level power estimators, such as cycle accurate simulators, cannot be used. A DPA attack uses statistical methods that can detect very small power variations, which are plainly neglected by high-level power estimators. Furthermore, in order to retrieve as much hidden information as possible, an attacker will sample several times per clock cycle.

4.3 Implementation Details

We have implemented the module in Wave Dynamic Differential Logic (WDDL) [7]. WDDL is a logic style that has dynamic differential behavior. Yet, it is implemented with static complementary CMOS logic, which is the default logic style used in standard cell libraries. In WDDL, static CMOS standard cells are combined to form secure compound standard cells, which have a reduced power signature. It has the advantage that it can readily be implemented without the investment in a custom designed standard cell library. We used a commercially available standard cell library developed for a 0.18µm, 1.8V CMOS technology. A WDDL gate actually is a double-height cell that consists of 2 abutted static CMOS standard cells, which are extended with filler cells in which the pin placement requirements are fulfilled. A WDDL gate does not have balanced input capacitances, nor balanced output capacitances. The filler cells could also incorporate additional capacitances such that the intrinsic capacitances become balanced. The implementation consists of 194 WDDL gates, which are 440 actual static CMOS standard cells.

Two different layouts have been created. Both designs have exact the same floor plan and cell placement. The difference is in the routing procedure. One implementation, which we will call the 'regular route'-design, has been routed without any constraints or special techniques. The other, which we will call the 'differential pair route'-design, has been routed with the differential routing technique described in this manuscript.

Place & route has been done in Silicon Ensemble version 5.3. Row utilization and aspect ratio are set at 0.80 and 1 respectively. Layout-to-netlist, in which transistors and layout parasitics are extracted, is done with Virtuoso. Simulations are done in Hspice, with the transient increment set at l0ps. The clock frequency of the circuit is chosen at 125MHz, and therefore 800 'measurement' samples are made per clock cycle. The clock and input signals are driven by cascaded inverters in order to provide realistic data and clock signals. The power consumption of the additional input circuitry is excluded from the measurements. In total, 2000 clock cycles have been simulated with a random input at the plaintext P_L and P_R, and with a fixed secret key K, equal to 46.

4.4 Experimental Results

Silicon Ensemble required 8 and 3 CPU seconds on a SUN ULTRA 5 to route without any violations the regular route and the differential pair route respectively. It took 0.85 CPU seconds to parse and create the final differential netlist. Note that many floorplan settings can be evaluated; only the final

design should be parsed. To have a comparison, we have also made an attempt to use Cadence Chip Assembly Router version 11.0.06 [12] to route the placed design. This is one of the commercially available tools that has the capability of routing differential pairs. Only the 442 internal nets have been defined as 221 differential pairs. In total, 100 iterations have been performed. This required 7 hrs 56 min and 33 sec in CPU time without generating a completely routed result. It still had 972 conflicts and 125 unconnected nets.

Figure 6 and Figure 7 show 2 histograms in which the internal interconnect capacitances of the regular route and the differential pair route are compared. The capacitance per net was reported directly from Silicon Ensemble using Simcap. Figure 6 depicts the distribution of the ratio between the capacitance at the true signal net and the capacitance at the corresponding false signal net. The variation between the capacitances at the differential nets is up to a factor 4 for the regular route procedure. On the other hand, the differential pair route procedure does not show any variation. In fact, the tool always returned exact the same values.

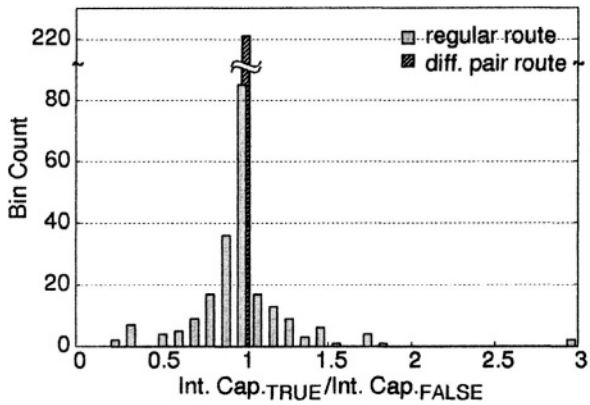

Figure 6. Ratio between interconnect capacitances at true and false nets.

The distributions of the absolute values of the capacitances, which are shown in Figure 7 are very much similar between the 2 routing procedures. This indicates that the mean power consumption and the time delay of the 2 implementations will be alike.

In the transient simulation, the mean energy consumption per clock cycle is 42.72pJ and 44.21pJ for the regular route and the differential pair route respectively. The normalized energy deviation, which specifies the absolute range of the variation on the energy consumption per cycle, is 1% for the

regular route and 0.7% for the differential pair route. The normalized standard deviation is 0.2% and 0.1% respectively.

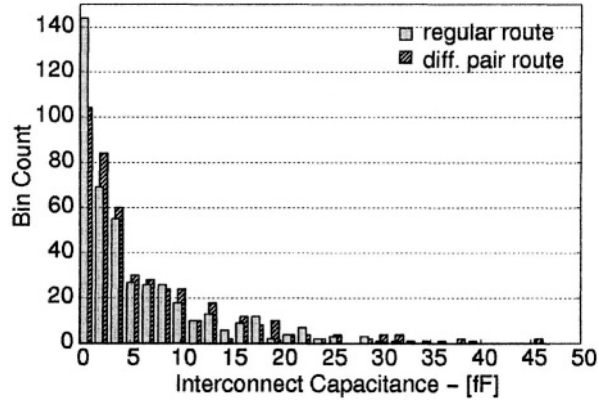

Figure 7. Absolute interconnects capacitances.

Figure 8 shows DTs from the DPA on our transient simulation. For each implementation, 2 DTs are shown: the one from the correct secret key guess and one from an arbitrary incorrect secret key guess. For transparency of the figure, the DTs of the other incorrect key guesses have been omitted. The omitted DTs are in accordance with the one shown. The differential pair routing has been effective in reducing the peaks in the DT of the correct secret key. Compared to the regular routing, the differential pair route achieves a complete reduction.

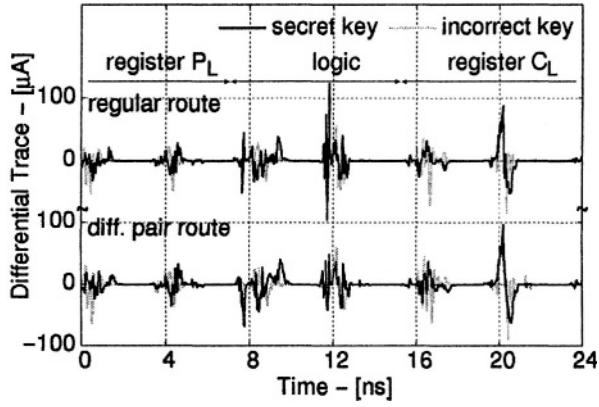

Figure 8. Differential trace for correct and arbitrary incorrect key guess.

Figure 9 shows the peak-to-peak value of the differential traces of the secret key guesses. For the regular routed design, the DT of the secret key stands out. On the other hand, for the differential pair routed design, the DT does not reveal the secret key.

Several approaches, which we have mentioned throughout the paper, such as for instance shielded lines or a larger pitch, balanced intrinsic capacitances and ground planes, are still available to improve the results. These options all have in common that they are not for free: one requires more design time, the other more area, etc. This is typically for security applications, where the higher the level of security is aspired, the more expensive the implementation will be. Yet, note also that we have performed a perfect attack with perfect measurement results. In real life, many factors, such as other circuitry, noise, decoupling, sample frequency, jitter, etc. will influence final result. The more the attacker is willing to invest in his measurement setup, the better his results may be.

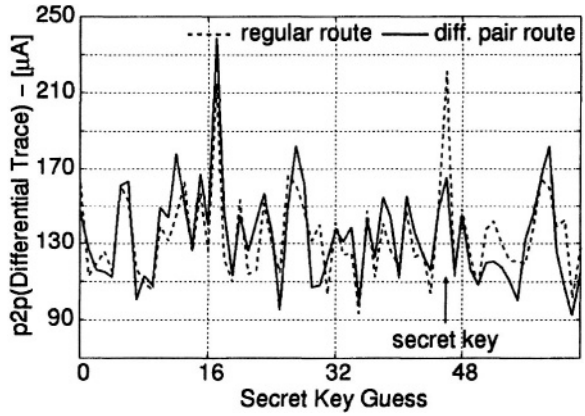

Figure 9. Peak-to-peak value of differential traces.

It does not come as a surprise that the secret key is still detectable in the regular routed design, even though the cycle-to-cycle variation on the power consumption is a mere 1%. DPA is a very powerful attack that, as we mentioned before, will reveal apparent insignificant power differences. Therefore it is also necessary to proof the resistance of a technique against DPA with the results of an actual DPA and not to rely on any form of visual inspection of the power consumption behavior.

5. CONCLUSIONS

We have presented a methodology to route multiple differential wires between secure dynamic differential standard cells. In the methodology, a wire code is assigned that abstracts the differential pair as a single fat wire. Subsequently, the router is run using this fat wire and at the end the fat wire is decomposed back into the 2 wires. Experimental results have shown that a differential design is routed 2.6 times faster with this methodology compared with the case that the same differential design is routed without any constraints. The differential routing effectively helps in controlling the parasitic effects between the two wires of each differential pair. Experimental results show that the differential pair routing is an essential technique to successfully thwart the DPA.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the National Science Foundation, grant NSF – CCR 0098361.

6. 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," Proc. of Eurosmart Security Conference pp. 55–64, June 2000.
2. P. Kocher, J. Jaffe, B. Jun, "Differential Power Analysis," Proc. of CRYPTO'99, LNCS 1666, pp. 388–397, Jan. 1999.
3. S. Chari, C. S. Jutla, J. R. Rao and P. Rohatgi, "Towards Sound Approaches to Counteract Power-Analysis Attacks," Proc. of CRYPTO'99, LNCS 1666, pp. 398-412, Jan. 1999.
4. K. Tiri and I. Verbauwhede, "Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology," Proc. of CHES 2003, LNCS 2779, pp. 125–136, Sept. 2003.
5. J. Fournier, S. Moore, H. Li, R. Mullins and G. Taylor, "Security Evaluation of Asynchronous Circuits," Proc. of CHES 2003, LNCS 2779, pp. 137–151, Sept. 2003.
6. R. Brashears, and A. Kahng, "Advanced routing for deep submicron technologies," vlsicad.ucsd.edu/Presentations/talk/supporting.html, May 1997.
7. K. Tiri, I. Verbauwhede, "A Logic Level design methodology for a secure DPA resistant ASIC or FPGA implementation," Proc. of DATE 2004, pp. 246-251, Feb. 2004.
8. S. Khatri et al. "A novel VLSI layout fabric for deep sub-micron applications," Proc. of DAC 1999, pp. 491-496, June 1999.
9. Silicon Ensemble, www.cadence.com/products/digita_ic/sepks
10. LEF/DEF Language Reference 5.5, Jan. 2003, www.openeda.org
11. NIST FIPS PUB 46-2 "Data Encryption Standard," Dec. 1993. www.itl.nist.gov/fipspubs/fip46-2.htm
12. Cadence Chip Assembly Router, www.cadence.com/products/custom_ic/chip_assembly