School of Electrical, Electronic & Computer Engineering

UNIVERSITY OF
NEWCASTLE UPON TYNE

On-line IDDQ testing of security circuits

A. Bystrov, J.P. Murphy

Technical Report Series

NCL-EECE-MSD-TR-2004-103

2004

Contact:
a.bystrov@ncl.ac.uk

j.p.murphy@ncl.ac.uk

Supported by EPSRC grant GR/S81421 (SCREEN)

NCL-EECE-MSD-TR-2004-103

School of Electrical, Electronic & Computer Engineering,
Merz Court,
University of Newcastle upon Tyne,
Newcastle upon Tyne, NE1 7RU, UK

http://async.org.uk/

On-line IDDQ testing of security circuits

A. Bystrov, J.P. Murphy

2004

Abstract

The new at-speed on-line IDDQ testing method is based upon the properties of a special class of security circuits. These circuits implement dual-rail encoding and return-to-spacer protocol, where the spacer is either all-zeroes or all-ones. The alternating spacers of different polarity guarantee that all wires switch once within each clock cycle, thus making energy consumed within a clock cycle independent from data processed. This property earlier used for security purposes is now exploited in order to separate the transient current from the quiescent current, thus making it possible to measure the latter under normal operation. Power signatures are analysed in the frequency domain and the fault signature filtration method is proposed. The proposed method can be used in both production, where it covers all interconnect stuck-at faults in just two clock periods; and on-line testing, where it guarantees the bounded and short period of self-test independent from data. From security point of view, it directly detects a side channel created under a fault injection attack.

Index Terms: On-line testing, IDDQ testing, dual-rail encoding, hardware security, hazard-free design

1 Introduction

Secure applications such as smart cards require measures to resist attacks, e.g. Differential Power Analysis (DPA) [1, 2]. Dual-rail encoding provides a method to enhance the security properties of a system making DPA more difficult. As an example, in the design described in [3] the processor can execute special secure instructions. These instructions are implemented as dual-rail circuits, whose switching activity is meant to be independent from data. Special types of CMOS logic elements have been proposed in [4], but this low-level approach requires changing gate libraries and hence is costly for a standard cell or FPGA user. A method using balanced data encoding together with self-timed design techniques has been proposed in [5, 6]. In recent work [7] a method integrated in a standard design flow was described. Independently, we proposed a different method strongly linked to the industry CAD tools and based upon synchronous dual-rail circuits operating under a special protocol [8, 9, 10].

All these methods improve certain aspects of security, but still suffer from vulnerability to fault injection attacks. The idea behind a fault injection attack is simple: to modify the behaviour of a circuit so that the secret data became “visible” at either the unprotected outputs or at a side-channel such as power waveform or EMI. In this paper we are looking at data exposure in the form of power supply current variations. A particular form of the fault injection attack we are concerned with is the illumination of the die surface by a thin laser beam. Such a beam potentially can be focused at an individual gate, causing its pull-up or pull-down transistors to “leak” the quiescent current, which is strongly related to the data at the output of the gate under attack. Depending upon the intensity of the beam, the fault may or may not change the logic behaviour of the circuit. We analyse the “signature” of the power source current and filter out its quiescent current (IDDQ) component. The switching protocol that makes our circuits secure is also used in the optimal filter for IDDQ signature.

Apart from the security enhancement the method provides *massive controllability* by implementing a special switching protocol for all gates, and *massive observability* by using IDDQ testing. I guarantees that all faults of the given class are detected within a bounded and very short period of time. Similar to traditional IDDQ testing methods [11, 12] it also covers many additional faults.

The rest of this paper is organised as follows: Section 2 describes the class of security circuits we are dealing with (as defined in [8, 9, 10]), Section 3 presents the new method for on-line IDDQ testing of these circuits and Section 4 draws the conclusions.

2 Security circuits

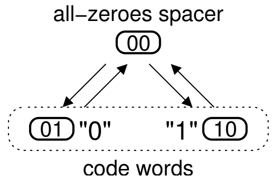
2.1 Return-to-zero dual-rail

Dual-rail code uses two rails with only two valid signal combinations $\{01, 10\}$, which encode values 0 and 1 respectively. Dual-rail code is widely used to represent data in self-timed circuits [13, 14], where a specific protocol of switching helps to avoid hazards. The protocol allows only transitions from all-zeroes $\{00\}$, which is a non-code word, to a *code word* and back to all-zeroes as shown in Figure 1(a); this means the switching is monotonic. The all-zeroes state is used to indicate the absence of data, which separates one code word from another. Such a state is often called a *spacer*.

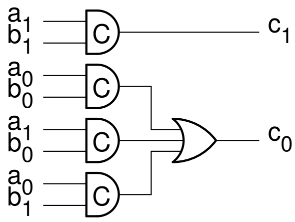
An approach for automatic converting single-rail circuits to dual-rail, using the above signalling protocol, that is easy to incorporate in the standard RTL-based design flow has been described in [15]. Within this approach, called Null-Convention Logic [16] one can follow one of two major implementation strategies for logic: one is with full completion detection through the dual-rail signals (NCL-D) and the other with separate completion detection (NCL-X). The former one is more conservative with respect to delay dependence while the latter one is less delay-insensitive but more area and speed efficient. For example, an AND gate is implemented in NCL-D and NCL-X as shown in Figure 1(b,c) respectively. NCL methods of circuit construction exploit the fact that the negation operation in dual-rail corresponds to swapping the rails. Such dual-rail circuits do not have negative gates (internal negative gates, for example in XOR elements, are also converted into positive gates), hence they are race-free under any single transition.

If the design objective is only power balancing (as in our case), one can abandon the completion detection channels, relying on timing assumptions as in standard synchronous designs; thus saving a considerable amount of area and power. This approach was followed in [8], considering the circuit in a clocked environment, where such timing assumptions were deemed quite reasonable to avoid any hazards in the combinational logic. Hence, in the clocked environment the dual-rail logic for an AND gate is simply a pair of AND and OR gates as shown in Figure 1(d).

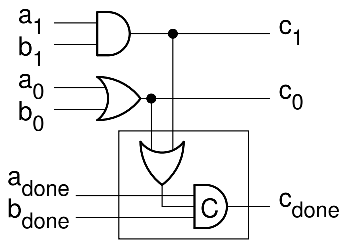
The above implementation techniques certainly help to balance switching activity at the level of dual-rail nodes. Assuming that the power consumed by one rail in a pair is the same as in the other rail, the overall power consumption is invariant to the data bits propagating through the dual-rail circuit. However, the physical realisation of the rails at the gate level is not symmetric, and experiments with these dual-rail



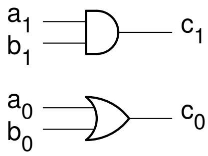
(a) Dual-rail protocol



(b) NCL-D AND gate



(c) NCL-X AND gate



(d) NCL-X AND gate in clocked environment

Figure 1: Single spacer dual-rail

implementations show that power source current leaks the data values.

An example of dual-rail flip-flop design can be found in [9, 10].

2.2 Alternating spacer dual-rail protocol

In order to balance the power signature we propose to use two spacers [9] (i.e. two spacer states, $\{00\}$ for *all-zeroes spacer* and $\{11\}$ for *all-ones spacer*), resulting in a dual spacer protocol as shown in Figure 2. It defines the switching as follows: *spacer→code word→spacer→code word.* A possible refinement for this protocol is the *alternating spacer protocol* shown in Figure 2. The advantage of the latter is that all bits are switched in each cycle of operation, thus opening a possibility for perfect energy balancing between cycles of operation.

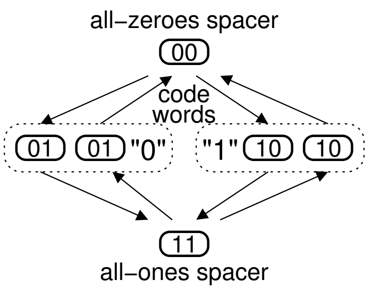


Figure 2: Alternating spacer dual-rail protocol

As opposed to single spacer dual-rail, where in each cycle a particular rail is switched up and down (i.e. the same gate always switches), in the alternating spacer protocol both rails are switched from *all-zeroes spacer* to *all-ones spacer* and back. The intermediate states in this switching are *code words*. In the scope of the entire logic circuit, this means that for every computation cycle we always fire all gates forming the dual-rail pairs.

This protocol is enforced by non-standard flip-flops as described in [9, 10].

2.3 Alternating spacer dual-rail circuits

In CMOS a positive gate is usually constructed out of a negative gate and an inverter. Use of positive gates is not only a disadvantage for the size of dual-rail circuit, but also for the length of the critical path. Negative gate optimisation of our circuits [8] improves both the speed and area metrics.

The alternating spacer dual-rail circuits are identical in their combinational part to the return-to-zero logic, e.g. Figure 1(d). The difference is only in the flip-flops [9], which generate all-zeroes spacers in all odd clock cycles and all-ones spacers in all even clock cycles. The operation of the AND gate as in Figure 1(d) under such a protocol is shown in Figure 3.

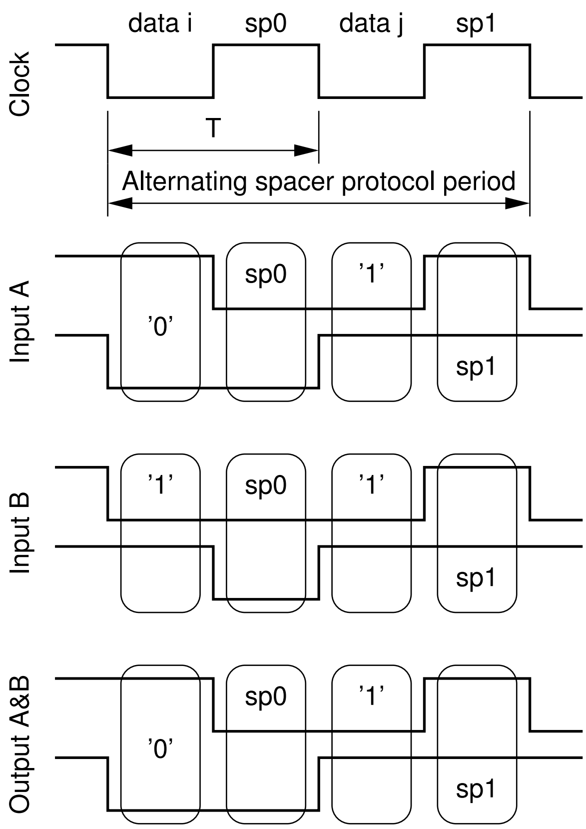
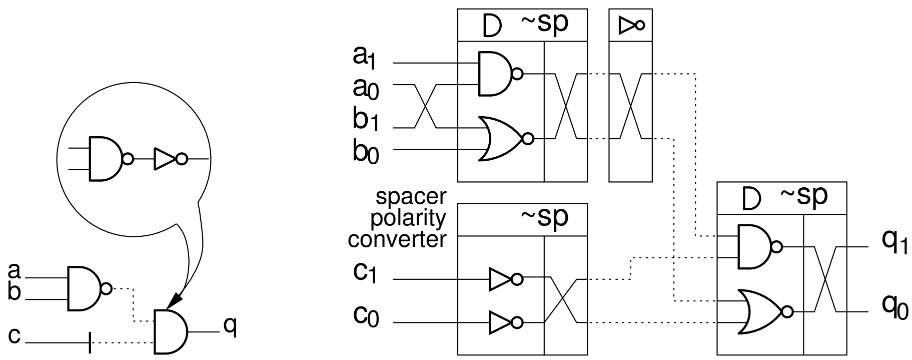


Figure 3: Alternating spacer dual-rail AND-gate in the clocked environment

Our security circuits are automatically generated by replacing all gates in a single-rail netlist by their dual-rail counterparts; this is done for all flip-flops and logic gates. The combinational logic produced by such a replacement comprises positive gates only (inverters are represented as crossovers between the rails). A useful property of the positive logic (regardless of data encoding) is that if all-ones are applied to the primary inputs, then they propagate to all wires within the circuit. The same is true for the all-zeroes input pattern. Thus, no provisions are needed to ensure spacer propagation. Then the combinational logic is optimised w.r.t. negative gates as illustrated in Figure 4 and described in detail in [9].



(a) Single-rail circuit

(b) Negative gate dual-rail circuit

Figure 4: Constructing negative gate dual-rail circuit

Dotted lines in the single-rail circuit, Figure 4(a), indicate the future position of dual-rail signals with inverted spacers (dual-rail data is not inverted as this effect is corrected by rail crossover). The bar on the wire is the location of a spacer polarity converter. The circuit in Figure 4(b) is the result of the conversion.

2.4 Energy balancing

In [9] we introduce two important security characteristics of *energy imbalance* and *exposure time* which in this paper serve as a baseline for the testing method.

Energy imbalance (further referred to as *imbalance*) can be measured as the variation in energy consumed by a circuit processing different data. If e_1 and e_2 are the energy consumption under two input patterns, then the numerical value of imbalance is calculated as:

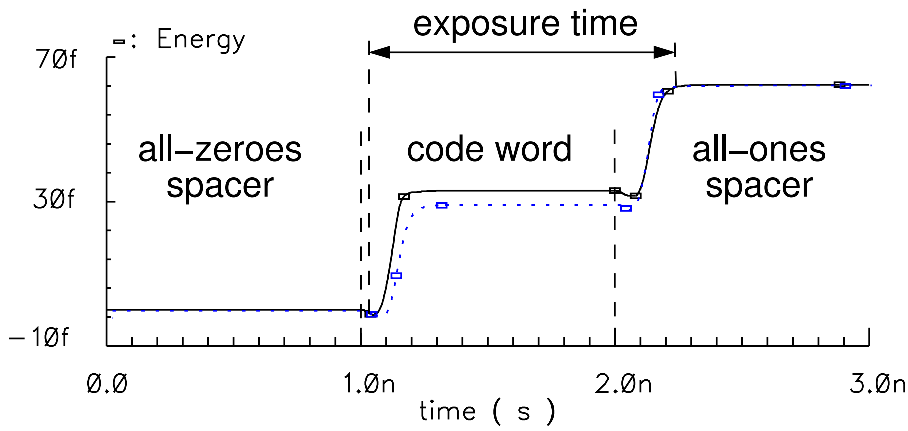
$$d = \frac{|e_1 - e_2|}{e_1 + e_2} \cdot 100\%$$

The imbalance values can be as large as 11% for some standard gates under certain operational conditions.

The *exposure time* is the time during which the imbalance caused by data is exhibited.

The alternating spacer dual-rail circuits have a nice property of the bounded exposure time, which is illustrated in Figure 5. In this experiment we use a 2-input AND element converted to the alternating spacer dual-rail circuit, optimised w.r.t. negative gates and implemented in AMS-0.35μ technology. The figure shows the energy imbalance, which occurs for about one half of the clock period (2ns). In slower circuits, such as those used in smart cards, the upper bound becomes exactly one half of the clock period.

In this paper we use this property to separate the random effects of data from the effects of a fault.



(a) Alternating spacer protocol

Figure 5: Exposure time for the alternating spacer AND2 gate

3 At-speed IDDQ testing

3.1 Benchmark

The four-byte multiplier which is a part of AES [17, 18] is chosen as a benchmark for our experiments. It is a combinational circuit comprising 294 logic gates. In this paper we restrict ourselves to study of combinational circuits only, and the main reason for this is that we use in our library under optimised flip-flops based upon standard cells. These flip-flops will be optimised at the transistor level and this is a subject of future work.

The multiplier implements the MixColumn operation in AES, which maps one column of the input state (4 by 4 matrix) to a new column in the output state. The multiplier computes one output byte of the MixColumns operation based on a four byte input. This is done by considering each column as a polynomial over $GF\left(2^8\right)$ and multiplying it with the constant polynomial $a(x)$:

$$a(x) = \left(\{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}\right) \text{ modulo } \left(x^4+1\right)$$

This means that four multipliers are needed to generate a whole column or reuse of the same multiplier four times. The multiplier also implements the InvMixColumns transformation where the constant polynomial changes to $d(x)$:

$$d(x) = \left(\{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\}\right) \text{ modulo } \left(x^4+1\right)$$

The multiplier circuit was synthesised from the Open Core specifications by using Synopsis tools and then converted into dual-rail by our custom tool described in [9].

3.2 IDD signature of a fault

As our circuits exhibit no long-term imbalance, the power consumption current (IDD), more precisely the mean IDD value, is constant from one protocol cycle to another in absence of faults. The power consumption is defined by the transient currents (IDDT) of the gates. If a fault causing increased quiescent current (IDDQ) occurs, then this will be added to the IDDT. So, if the IDD increases, this is the indication of a fault.

Unfortunately, such an intuitive idea has its flaws. First, the IDDT in a large circuit can be so high that it may be difficult to detect the relative IDD increase due to a single fault. Second, a stuck-at fault changes the logic behaviour of a circuit. In a positive logic circuit, and hence in an alternating spacer circuit, it reduces switching activity. As the switching activity is strongly related to IDDT, this effect may compensate for the increase in IDDQ. In our approach these issues are resolved by using optimal filters "tuned" to pick up the IDDQ waveform. The filtration is possible as the alternating spacer protocol modulates the IDDQ of each single fault, and the modulation law is known.

The fault model we use includes all single stuck-at faults at interconnect between logic gates, i.e. this is the output single stuck-at model. It will be shown that many multiple faults are also covered.

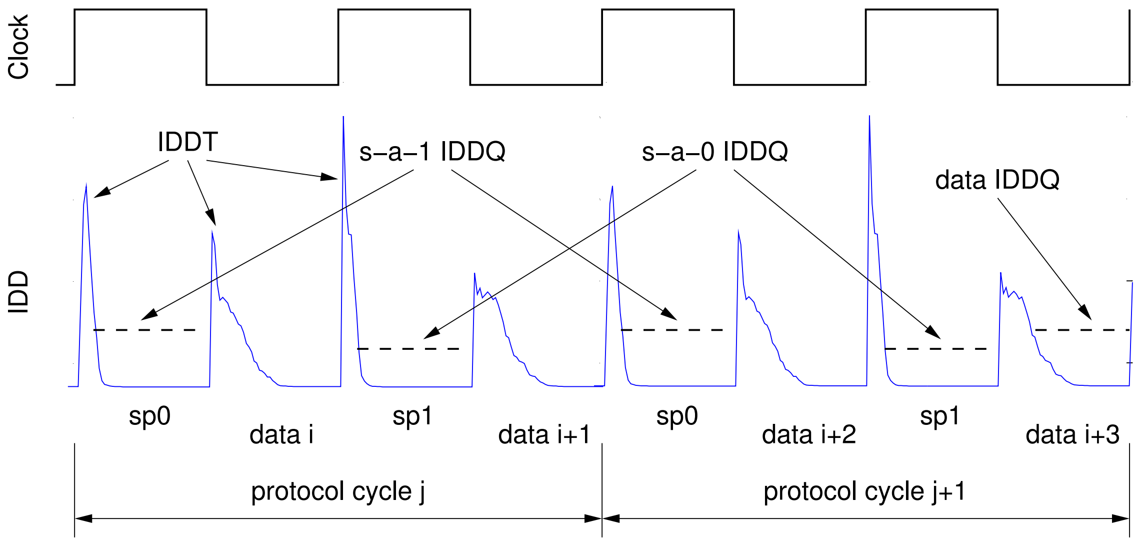


Figure 6: Generalised IDD

Figure 6 shows the power current of the dual-rail circuit under the alternating spacer protocol. One protocol cycle includes two clock periods. In the first period the spacer is all-zeroes (denoted as sp0) and in the second period it is all-ones (sp1). This is indeed only true for the spacer polarities at the primary inputs/outputs. The internal signals, due to negative gate optimisation may have the opposite spacer polarities. For simplicity, in the further discussion we will assume that all gates are positive, although the actual experiments were conducted using optimised circuits. Data in each clock period is different and it is denoted as "data i" to "data i+3". The IDD plot indicates four current peaks (transient current IDDT) per each protocol cycle. All peaks are different in shape and the area under the graph, they correspond to four transitions forming the full protocol cycle (see Figure 2). One can observe that the transitions from data into a spacer are short. This is due to early propagation effects, which *always* take place in these transitions. The transitions from a spacer into data are wider and have distinct shapes. This shape is formed by the computations performed in the logic gates. There is less room for the early propagation here, however it still may take place depending upon the function of the block and the data values. Each pair of subsequent spacer-to-data and data-to-spacer IDDT peaks have the same area under graph if compared to the corresponding pair within any other protocol cycle. This is due to the properties of the alternating spacer protocol. The energies of the first and the second halves of each protocol cycle are also almost identical, at least in absence of faults.

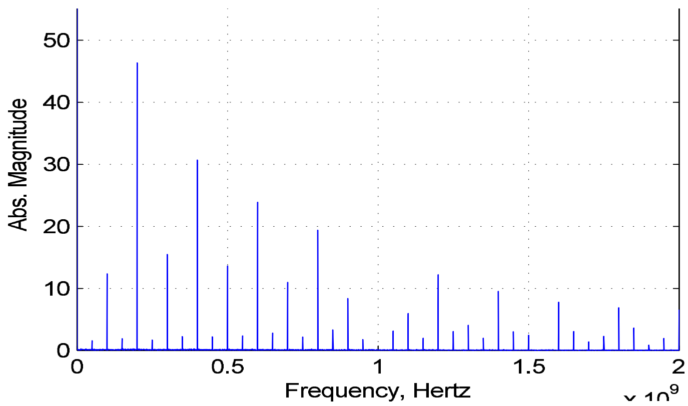
If, however, a single fault occurs, its IDDQ contributes differently to the different halves of protocol cycles. As our fault model includes only stuck-at faults on interconnect, they are guaranteed to be activated by either sp0 or sp1 spacer. Furthermore, if sp0 activates the fault, then under sp1 it will not be activated. The opposite is also true. Thus, the signature of a fault is one spacer activation within each protocol period. These circuits have the property of *massive controllability*, which provides activation of all faults within two subsequent clock periods regardless of data. Each s-a-1 fault is activated by sp0 and each s-a-0 fault is activated by sp1. The IDDQ currents corresponding to the faults of different polarities are shown in Figure 6 as dashed lines. The dashed lines denoted as "s-a-1 IDDQ" and "s-a-0 IDDQ" are the fault currents activated by the corresponding spacers. The label "data IDDQ" corresponds to the fault current activated by a data state. Indeed, under a data state half of wires in a dual-rail circuit have 1 and the other half have 0 values, so they activate the corresponding faults. Fault activation by data, however, cannot be guaranteed in bounded time, because these values are data-dependent and in general case random.

Spectral representation of the IDD of our multiplier running at clock speed of 100MHz and random data is shown in Figure 7(a). The tallest peak is 200MHz, the frequency of transitions irrespective to their designation. The frequency of 100MHz is the signature of gate imbalance. The zero frequency is the mean value of IDD. Finally, 50MHz is the frequency of spacers of a particular polarity, this is where the fault IDDQ is activated. Figure 7(b) shows the difference between the spectrums with and without a fault. The fault shows up at 0 and 50MHz. The last diagram in Figure 7(c) shows the same difference in relation to the absolute value of the spectrum without fault. At zero frequency the relative difference is 12% (which is not clearly seen in this diagram). This indicates a potential problem with fault detection by IDD mean value. At the frequency 50MHz, however, the increase is about 180%, which is good enough for driving a coarse comparator.

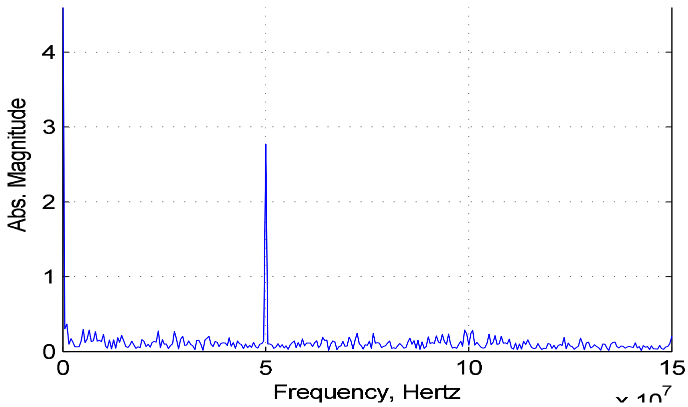
We use the above observations for a quick evaluation of our idea only. So far we were looking at the amplitude spectrum only, which does not represent any information about the phase of signal components. As Figure 6 shows, the fault IDDQ and the IDDT take place in different phases of the protocol sequence, and the experiments described below utilise this additional information.

The principle of cross correlation is used in our approach in order to filter out the fault IDDQ signature. The top diagram in Figure 8 is a fragment of the IDD waveform obtained by Spice simulation of the multiplier benchmark, it has the mean value subtracted in order to simplify the following processing. There were 1000 clock cycles simulated in total and the first protocol cycle was discarded. The reference signal is defined as a sine wave of half clock frequency. The initial phase is chosen so, that the maximum of the wave approximately corresponds to sp0 spacer and the minimum to sp1 spacer (the phase of the reference signal in Figure 8 corresponds to the time in the IDD plot). Then the cross correlation between the IDD wave and the reference signal is calculated (see the Cross Correlation plot, no fault, random data), and the initial phase of the reference signal is adjusted to produce 0 cross correlation under 0 lag.

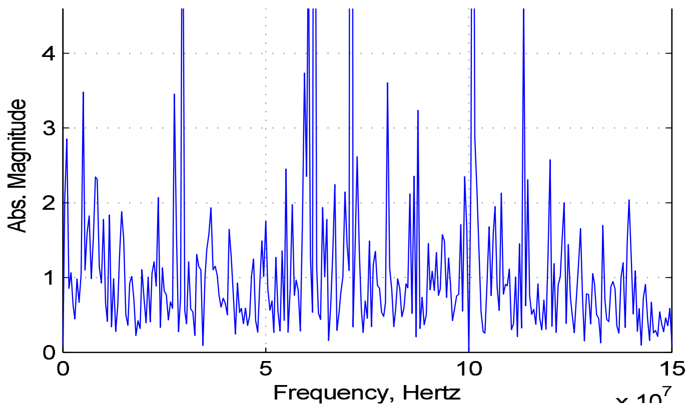
The effect of data on the cross correlation diagram was investigated. The maximum deviation from



(a) IDD frequency spectrum under fault



(b) spectral difference fault – no fault



(c) relative spectral difference

Figure 7: Power signatures in the frequency domain

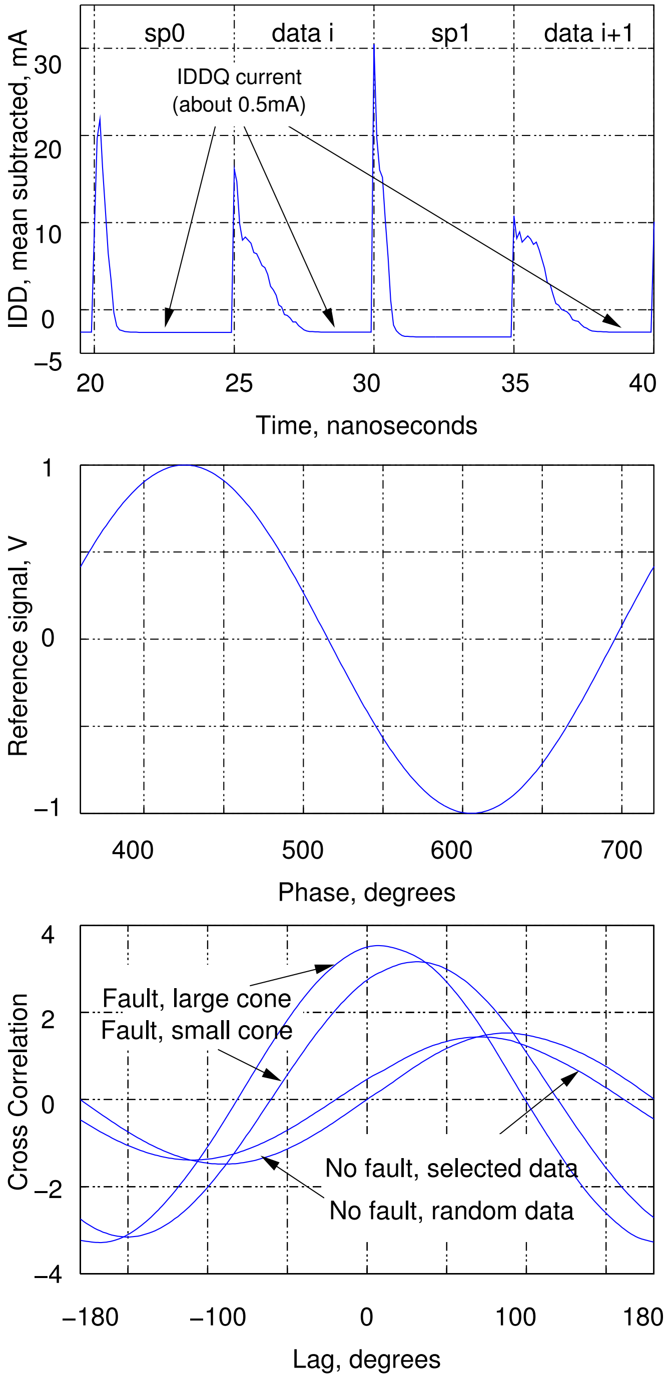


Figure 8: Fault IDDQ detection by cross-correlation

the first plot is obtained under the data producing the most asymmetrical IDD waveform (see "No fault, selected data").

Then a fault was introduced. At first we placed the fault at the output of the circuit in order to have the minimum impact on the switching activity. This produces the cross correlation curve labelled as "Fault, small cone" (the cone is a set of the successor gates of the fault location). At 0 lag the cross correlation value is about 5.5 times greater than the maximum value produced by the "bad" data without faults. This gives a very good margin for fault detection. Then we moved the fault close to the input of the circuit (the third bit in one operand), thus creating the largest cone with reduced switching activity. The result of this experiment is labelled as "Fault, large cone". The result of cross correlation shows even greater value of 7 at lag 0. We explain this effect by the IDDT imbalance due to violation of the alternating spacer protocol in the cone. An interesting observation is that the switching activity reduction in the cone reduces the mean value of IDD (frequency zero), but in the same time introduces more imbalance, which is good for the proposed fault detection method. In this sense our method uses not only the quiescent current, but also to some extent the transient current to detect faults.

3.3 Fault model extension

Although we designed the method for single stuck-at faults, it should also work for multiple unidirectional faults. By unidirectional we understand all faults increasing the IDDQ during either sp0 or sp1 spacer of the protocol. If the faults are not unidirectional, then they can cancel the effect of each other. Furthermore, the analysis of such faults should take into account their location w.r.t. the cones generated by them. This analysis is the subject of the future work.

We have also conducted a preliminary investigation on detection of parametric faults. In our setup we modeled a leakage fault as a small open transistor, e.g. 1μ pull-up transistor at the output of an inverter that uses 2μ pull-up and 1μ pull-down devices. The effect of such a fault could be reliably detected by our method producing the effect of about one-half of the short-circuit fault.

4 Conclusions

The proposed IDDQ method is applicable to a special class of security circuits: synchronous, dual-rail, return-to-spacer protocol with alternating spacer states. The data-independent power consumption of such circuits opens an opportunity for on-line application of IDDQ testing.

The method includes the provision of *massive controllability*, i.e. all potential faults are activated within two clock periods, and *massive observability* (IDDQ measurement). This can be useful for both production testing (minimisation of time on tester and minimisation of the number of test pins), and on-line testing. In the on-line testing application the method guarantees a bounded and very short self-test period, independent from data, which is different from many other on-line testing methods. Furthermore, the method also detects parametric (leakage) faults, which are often used in attacks on security devices. In this sense, the proposed approach directly detects the information leakage through a side channel.

It was shown that the increase of the mean IDD value due to a fault cannot be reliably used for fault detection, because of two reasons. First, a stuck-at fault in the given class of circuits reduces switching activity and, thus, the transient current. This effect may compensate for the quiescent current increase. Second, the transient current in large circuits is so high, that the relative increase in the overall current due to IDDQ is small. In our example it was only 12%.

The method is based upon the cross correlation operation and it is close to the optimal filtration approach. It is different from the optimal filter as it uses a sine wave as a reference signal. In the optimal filter it would be a pattern matching the shape of the IDDQ. We believe that using signals more complex than a sine wave will significantly increase the complexity of the filter, giving little benefits. More experiments are needed to support this statement.

The method was applied to a benchmark circuit, which is an important part of AES cryptographic block. The circuit comprises 294 logic gates. It is important to do more case studies in order to determine the maximum size of a circuit served by a single current sensor. The design of the current sensor itself and the signal processing part are the subject of future work.

Acknowledgements: We are grateful to A. Yakovlev, G. Russell and A. Koelmans for useful comments, and to D. Sokolov for designing the tool for synthesis of the dual-rail security circuits. EPSRC supports this work via GR/S81421 (SCREEN).

References

[1] P.Kocher, J. Jaffe, B. Jun, “Differential Power Analysis”. Proc. Crypto, 1999.

[2] T.Messerges, E.Dabbish, R.Sloan: “Examining smart-card security under the threat of power analysis attacks”. IEEE Trans. on Computers, 2002, 51(5), pp. 541–552.

[3] H.Saputra, N.Vijaykrishnan, M.Kandemir, M.J.Irwin, R.Brooks, S.Kim, W.Zhang: “Masking the energy behaviour of DES encryption”. Proc. DATE, 2003.

[4] K.Tiri, M.Akmal, I.Verbauwhede: “A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards”. Proc. ESSCIRC, 2002.

[5] S.Moore, R.Anderson, P.Cunningham, R.Mullins, G.Taylor: “Improving smart card security using self-timed circuits”. Proc. ASYNC, 2002, pp. 211–218.

[6] Z.Yu, S.Furber, L.Plana: “An investigation into the security of self-timed circuits”. Proc. ASYNC, 2003, pp. 206–215.

[7] K.Tiri, I.Verbauwhede: “A logical level design methodology for a secure DPA resistant ASIC or FPGA implementation”. Proc. DATE, 2004.

[8] A.Bystrov, D.Sokolov, A.Yakovlev, A.Koelmans: “Balancing power signature in secure systems”. 14th UK Asynchronous Forum, 2003.

[9] D.Sokolov, J.Murphy, A.Bystrov, A.Yakovlev: “Improving the security of dual-rail circuits”. Proc. CHES, 2004.

[10] D. Sokolov, J. Murphy, A. Bystrov, A. Yakovlev: “Design and analysis of dual-rail circuits for security applications”. Accepted IEEE Trans. on Comput., 2004

[11] M. W. Levi, "CMOS is most testable," in Int. Test Conf., 1981, pp. 217-220

[12] Rajsuman, R.: “Iddq testing for CMOS VLSI”. Proceedings of the IEEE , Volume: 88 , Issue: 4 , April 2000, pp.544-568

[13] V.Varshavsky (editor): "Self-timed control of concurrent processes" Kluwer, 1990 (Russian edition 1986).

[14] I.David, R.Ginosar, M.Yoeli: "An efficient implementation of boolean functions as self-timed circuits". IEEE Trans. on Computers, 1992, 41(1), pp. 2–11.

[15] A.Kondratyev, K.Lwin: "Design of asynchronous circuits using synchronous CAD tools". Proc. DAC, 2002, pp. 107–117.

[16] K.Fant, S.Brandt: "Null Convention Logic: a complete and consistent logic for asynchronous digital circuit synthesis". Proc. ASAP, IEEE CS Press, 1996, pp. 261–273.

[17] National Institute Of Standards and Technology: "Federal Information Processing Standard 197, The Advanced Encryption Standard (AES)". http://csrc.nist.gov/publications/fips/fips197/fips197.pdf, 2001.

[18] R.Usselmann: "Advanced Encryption Standard / Rijndael IP Core". http://www.asic.ws/.