Formalizing and Improving DPA resistance of Quasi Delay Insensitive Asynchronous circuits

G.F. Bouesse¹, M. Renaudin¹, F. Germain², S. Dumont¹
¹ TIMA Laboratory, Concurrent Integrated Systems Group
26 Av. Félix Viallet, 38031 Grenoble Cedex
e-mail: fraidy.bouesse@imag.fr
² SGDN/DCSSI
51 bd. De la Tour Maubourg
75700 Paris Cedex 07
e-mail: fabien.germain@sgdn.pm.gouv.fr

Abstract

The purpose of this paper is to formally specify a flow devoted to the design of Differential Power Analysis (DPA) resistant QDI asynchronous circuits. The paper first proposes a formal modeling of the electrical signature of QDI asynchronous circuits. The DPA is then applied to the formal model in order to identify the source of leakage of this type of circuits. Finally, a complete design flow is specified to minimize the information leakage. The relevancy and efficiency of the approach is demonstrated using the design of an AES crypto-processor.

I. Introduction and motivations

Since the discovery of the Power Analysis attacks such as Single Power Analysis (SPA) and Differential Power Analysis (DPA), asynchronous logic has been presented as a new alternative design solution against side channels attacks.

In fact, Power Analysis attacks with DPA firstly introduced in 1998 by Paul Kocher [1] use the weakness of chip hardware implementations and software running on cryptographic devices (particularly smart card), to reveal the chip’s confidential information. Secret keys are removed from device by observing and monitoring the electrical activity of a device and performing advanced statistical methods.

Additionally to their absence of clock signal which demonstrates the practical way to eliminate a global synchronization signal, asynchronous logic is well-known for its ability to decreasing the consumption and to shape circuits’ current. [2][3][4] demonstrate how 1-of-N encoded speed-independent circuits improve security by eliminating data dependent power consumption. Symmetry in data communication and data processing, persistent storage, timing information leakage and propagating alarm signals (to defend chip against fault induction) are design aspects addressed by those papers for increasing chip resistance.

The countermeasures that used the Self-timed circuit properties are all focused on balancing the operation through special DI Coding scheme. The concrete results of these approaches which illustrates how is Quasi Delay Insensitive asynchronous logic factor effective for resisting against DPA has been presented in [5]. By investigating and analyzing three different QDI DES architectures and design styles, it is demonstrated how the properties of 1-of-N encoded data and four-phase handshake protocol significantly improve the DPA resistance. However this study has also pointed out the limits of this design approach methodology by showing some residual sources of leakage.

This paper is focused on these residual sources of leakage that are still observable when implementing a DPA attack on secured QDI asynchronous circuits as described in [5]. The objective of this paper is to formalize a model of the current dissipated in this type of circuit in order to identify the sources of these remaining leakages and thus propose a new design flow to reduce them.

The paper is organized as follows. Section II presents the asynchronous properties that are used to design a DPA resistant chip, especially N-rail quasi delay insensitive asynchronous logic together with four-phase protocol. Section III investigates the electrical current model of this type of circuit and then section IV recalls the DPA attack skill by applying the attack on the model. Validations are presented in section V. The improvement of the design flow and results are reported in Section VI. The paper is concluded by giving some design perspectives in section VII.

II. Asynchronous logic and DPA

Asynchronous circuits represent a class of circuits which are not controlled by a global clock but by the data themselves. In fact, an asynchronous circuit is composed of individual modules which communicate to each other by means of point-to-point communication channels. Therefore, a given module becomes active when it senses the presence of incoming data. It then computes them and sends the result to the output channels. Communications through channels are governed by a protocol which requires a bi-directional signalling between senders and receivers (request and acknowledge). They are called Handshaking protocols [6] which are the basis of the sequencing rules of asynchronous circuits.

Fig. 1: Handshake based communication between modules. A module can actually be of any complexity.
There are two main classes of handshaking protocols: two-phase protocol and four-phase protocols. In this work, only the four-phase protocol is considered and described.

- Four-phase protocol
  This protocol requires a return to zero phase for both data/requests and acknowledgements.

  ![Fig. 2: Four-phase handshaking protocol](image)

  - Phase 1: Data detection (invalid Data to valid Data)
  - Phase 2: Acknowledgement is set to one
  - Phase 3: Data are re-initialized (valid Data to invalid Data, return to zero phase)
  - Phase 4: Acknowledgement is reset (return to zero phase)

Contrary to synchronous circuits where the shape of the current (current peaks) depends on the previous states and data values, QDI asynchronous logic using a four-phase protocol re-initializes all previously activated nodes before processing a new data. This behaviour enables the designer to precisely control the transitions involved in a given computation (fig. 3). Moreover, because it is based on hazard free logic QDI asynchronous circuits eliminate all current variations caused by glitches.

![Fig. 3: Types of transitions](image)

- Signalling
  As presented above, the implementation of a four-phase handshaking protocol requires sensing the presence of data in phase 1 and phase 3. In order to do so, dedicated logic and special encoding are necessary for sensing data validity/invalidity and for generating the acknowledgement signal.

  - Data/ Request encoding
    Considering that one bit has to be transferred through a channel using the four phase protocol, one bit has to encode three different values: invalid, valid at ‘1’, valid at ‘0’. Two wires (A0, A1) are then required to encode the three states. This technique is called dual-rail encoding (table1).

  - Acknowledge / Completion signal generation
    The acknowledgement signal is generated by taking advantage of the data-encoding. As depicted in fig. 4, a Nor gate is usually used to sense the dual-rail encoding output for generating the completion signal.

Table 1: Dual rail encoding of the three states required to communicate 1 bit.

<table>
<thead>
<tr>
<th>Channel data</th>
<th>A0</th>
<th>A1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Invalid</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Unused</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Dual rail encoding is easily extended to N rails. It is called 1-of-N encoding. This encoding data scheme is useful to reduce the number of electrical transitions involved in a given computation which reduces the power consumption. For the sake of DPA resistance, 1-of-N encoding ensures that the same number of transitions is required to encode the values 0 to N-1.

- Balanced data path
  As an example, consider the xor function. Fig. 4 shows a dual-rail xor gate implementation. Every computations of this dual-rail xor gate involve a fixed and constant number of transitions regardless of the data values. Hence, the opportunity to have data independent power consumption i.e. not correlated to the processed data is exactly the goal to achieve for DPA resistant chip.

![Fig. 4: Dual-rail xor gate with four-phase handshake protocol](image)

Dual rail “co” outputs the xor function performed between dual rail inputs “ai” and “bi”.

(Cr is a Muller gate with a reset signal)

The Muller gate (C-element) generates an up-transition when up-transitions occur at all its inputs, and generates a down-transition when down-transitions occur at all its inputs. The Muller C-element’s truth table and symbol are given in Fig. 5.

![Fig. 5: Muller gate or (C-element)](image)

However, the QDI implementation of a function is not always balanced. In such cases, the gate structure is modified to ensure that all data paths and control paths are balanced and do involve a constant number of transitions [7].

Publication [5] demonstrated that secured QDI asynchronous logic permit to achieve a very high level of DPA resistance. However, it also showed that place and route steps are unbalancing some paths, thus creating sources of leakage. Let’s now precisely analyze this point using a formal approach.

### III. Electrical model of secured QDI asynchronous logic

In this section, we propose a current model of a QDI block implementing all the secure techniques described in section II. We assume that all blocks of the circuit are balanced so that a fix number of logical transitions are required regardless of the input data. Let us first estimate the power dissipation of a logical block.

Power dissipation in static CMOS gate has the following components:

1. Static power dissipation due to junction leakage current, $P_s$.
2. Short-circuit power dissipation, $P_c$.
3. Dynamic power dissipation, $P_d$. 


As Static and Circuit-power represent about 15% of the CMOS gate power dissipation, we consider for the purpose of this study the Dynamic power dissipation \( P_d \) which is defined as the power required to charge and discharge the capacitive load of the gate. Its expression is given by:

\[
P_d = \eta f V dd^2
\]

(1)

Where \( \eta \) is a switching activity ratio, \( f \) is the switching frequency, \( V dd \) is the supply voltage, and \( C \) is the total charge of the output gate node, defined by:

\[
C = C_l + C_{par} + C_{sc}
\]

in which \( C_l \) represents the load capacitance (gate and routing capacitance), \( C_{par} \) is the parasitic capacitance, and \( C_{sc} \) is the Short-circuit equivalent capacitance. This model of power dissipation can be extended to a CMOS gate working in an asynchronous environment.

\[
P_{da} = \eta f_s V dd^2
\]

(2)

Where \( f_s \) is the switching frequency of the acknowledge signal. Its value can be fixed to 2 when one cycle time is considered: evaluation phase and return to zero phase (section I). The fix number of logical transitions in each QDI asynchronous block offers the possibility to estimate the block’s dynamic power dissipation. Let \( N_i \) be a number of logical transitions \((N_i \) is fixed for each block). The block’s dynamic power dissipation is given by:

\[
P_{di} = \eta f_s \sum_{i=1}^{N_i} C_i V dd^2
\]

(3)

The aim of the asynchronous balanced paths is to always have the same current profile regardless of the data computed. This modeling of the power dissipation shows that in spite of having the same number of transitions, the loading charge of each gate is not necessarily equal. Let then estimate the dynamic current dissipated in a block. The gate dynamic power dissipation is given by:

\[
I(t) = C \frac{dV}{dt}
\]

(4)

Let define \( N_g \) as the number of gates along the critical data-path. \( N_l \) represents the maximum number of gate in series in the block. This number \((N_l)\) can be used to divide a block in \( N_l \) logical levels. \( N_g \) is the number of gate switching at each logical level \( N_l \). The expression of the block dynamic current profile is now expressed by:

\[
P_{di} (t) = \sum_{i=1}^{N_g} \left[ \sum_{j=1}^{N_l} I_{ij}(t_j) \right] + P_{di} (t)
\]

(5)

\( I_{ij}(t_j) \) represents the dynamic current of the \( ij \)th gate of level \( i \) and \( P_{di} \) is a dynamic noise function. The values \( N_g \), \( N_l \) and \( N_d \) are determined by a simple analysis of a graphic representation of the block. For example, let consider the block of fig. 4. Its representation in the form of a directed graph is presented in fig. 5 \((G_{xor}=(V,E))\). The directed graph \( G_{xor}=(V,E) \) is built from the gate Netlist by defining all the gates as the elements of the set \( V \) (vertices) and all the interconnections as the elements of the set \( E \)(directed edges).

The graphic representation adopted in this study is well appropriated for this type of analysis. In the one hand, it offers the opportunity to formally verify the logical symmetry of the data-path and in the other hand it offers the possibility to annotate the graph with information collected at each different phase of the design.

\[\text{Fig. 5: Annotated directed graph } G_{xor}(V,E) \text{ of the Dual-Rail gate } Xor.\]

All dotted lines represent inputs and outputs of the block. We deduce through the graph exploration the values of \( N_i \), \( N_g \), and \( N_d \): \( \quad N_i = 4 \quad ; \quad N_g = N_d = 4 \quad N_d = 4 \quad \)

Therefore,

\[
P_{daxor}(t) = (I_{11}(t_1) + I_{21}(t_2) + I_{31}(t_3) + I_{41}(t_4)) + P_{di}(t)
\]

(6)

Equation (6) represents, in a first approximation, the profile of the dynamic current of the Dual-rail xor gate. This approach can be extended to all secured QDI asynchronous block.

This formal model enables to evaluate the sensitivity of the secured QDI asynchronous circuits to DPA with high accuracy.

\[\text{IV. Applying DPA on the formal model}\]

Before applying DPA attack on the formal model, we first review the basis of the attack. The formalization adopted in this paragraph was introduced by Thomas S. Messerges and Al. in [7].

DPA attack is performed by computing \( N \) random values of plain-text-input \((PTI)\). For each of the \( N \) plain-text-input, a discrete time power signal \( S_j \) and cipher-text-output are collected. The index \( i \) of power signal \( S_j \) corresponds to the \( PTI \) that produced the signal and the \( j \) index corresponds to the time of the sample. According to a DPA algorithm, the \( S_j \) are split into two sets by a separating function \( D() \).

\[
\begin{align*}
S_0 & = \{ S_j | D() = 0 \} \\
S_1 & = \{ S_j | D() = 1 \}
\end{align*}
\]

(7)

The average power signal of each set is given by:

\[
\begin{align*}
A_0(J) & = \frac{1}{\lfloor n_0 \rfloor} \sum_{j=1}^{n_0} S_j(J) \\
A_1(J) & = \frac{1}{\lfloor n_1 \rfloor} \sum_{j=1}^{n_1} S_j(J)
\end{align*}
\]

(8)

Where \( |n_0| \) and \( |n_1| \) represent the number of power signals \( S_j \) respectively in set \( S_0 \) and \( S_1 \). The DPA bias signal is obtained by:

\[
T[J] = A_1[J] - A_0[J]
\]

(9)
If the DPA bias signal shows important peaks, it means there is a strong correlation between the $D(t)$ function and the power signal. Selecting an appropriate $D(t)$ function is then essential in order to guess a good secret key. The methods to succeed the attack with a minimum of random values are presented in [8]. An example of such a $D(t)$ function is as follows:

1- DES algorithm:
$$D(C_i,P_b,K_b) = \text{SBOX1}(P_b \oplus K_b)$$

Where $C_i$ = first bit of SBOX1 function.

$P_b$ = 6-bit plain-text-input of the SBOX1 function.

$K_b$ = 6-bit of the first round of SBOX1.

$\text{SBOX1}$ = a substitution function of DES with 4-bit output.

2- AES algorithm:
$$D(C_i,P_b,K_b) = \text{XOR}(P_b,K_b)$$

Where $C_i$ = first bit of XOR function.

$P_b$ = 8-bit plain-text-input of the XOR function.

$K_b$ = 8-bit of the first round of XOR.

$\text{XOR}$ = a xor function of AES with 8-bit output.

The number of bits chosen for $C_i$ in the selection function determines the number of sets to create. If only one bit is chosen (which is the case mostly used), two sets are created as show in equation (7).

Let us apply this technique to a secured QDI asynchronous design. Choosing a XOR for the $D(t)$ function implies to analyse the electrical signature of an XOR gate. This function is chosen because it directly handles the secret key in most of the cryptographic algorithms.

Contrary to synchronous design where the DPA attack reveals path dissymmetry of the attacked bit ($C_i$), DPA on the secured QDI asynchronous design reveals path dissymmetry of all rails that are used to encode the attacked bit. In fact, applying DPA on Dual-rail XOR gate requires dissymmetry of all rails that are used to encode the attacked bit ($C_i$). Then, the average current signal of each bit is chosen (which is the case mostly used), two sets are created as show in equation (7).

The differential current analysis of two symmetric data-paths reveals the effects of each gate’s charge capacitance ($C_i$). It formally illustrates the impact of the gate’s charge, particularly the load capacitance, on the DPA bias signal.

V. Validation using electrical simulations

All electrical simulations are performed with Eldo and used the HCMOS9 design kit (0.13µm) from STmicroelectronic. The electrical simulation offers the possibility to analyze without disturbing signal (noise), the gate’s electrical behaviour with more details. Hence, the number of necessary messages ($N$) is minimal.

Figure 6 illustrates the electrical signature of the dual-rail XOR gate using all load capacitance ($C_d$) equal. Both evaluation and return to zero phases are analyzed. Signal $S(t)$ shows a few peaks due to internal gate capacitance: Short-circuit capacitance ($C_{sc}$) and parasitic capacitance ($C_{par}$).

Let us then vary the value of the load capacitance (precisely the value of the interconnection capacitance) in order to evaluate its effect on the gate electrical signature (Fig. 7). A default ($C_d$) value of net capacitance is fixed to 8fF (femto Farad).

Fig. 7-a represents the profile of signal $S(t)$ when $C_{d1}$ is two times bigger than $C_d$. As $C_{d1}$ is on the third level, we have one important peak at the end of each phase. It corresponds to the last term of the equation (12). When $C_{d1}$ is fixed to 2*$C_d$, two important peaks appear in the signal $S(t)$ (Fig. 7-b). In fact, as this value is inside the data-path, all computing operations after this gate are shifted by the time taken to charge/discharge this node. It is confirmed when $C_{d1}$ and $C_{d2}$ are both two times bigger than $C_d$ (fig. 7-c). This dissymmetry is amplified when the difference of capacitance between both data-paths increases. In fig. 7-d the values of $C_{d12}$ and $C_{d11}$ are four times bigger than $C_d$. As the difference occurs in the beginning of the block, the electrical curve of both sets are completely shifted so that the electrical signature is maximum.

This analysis confirmed the strong dependence between the net capacitance and the DPA bias signal of the balanced QDI asynchronous design. This directly points out the necessity to precisely control the place and route step.
VI. Improving place and route steps

We have demonstrated in paragraph V the effects of the net capacitances on the DPA bias signal. We now define a criterion for evaluating the net capacitance difference between the two rails of a dual-rail channel. If $C_{x}$ represents the net capacitance of rail $i$ ($i \in \{0,1\}$ for a dual rail channel) of channel $\mathcal{A}$, then the dissymmetry of a channel "$\mathcal{A}$" is defined by the following expression:

$$d_{\mathcal{A}} = \frac{|C_{10} - C_{01}|}{\text{Min}(C_{10}, C_{01})}$$

As illustrated in paragraph V, the lower the value of $d_{\mathcal{A}}$, the more resistant to DPA the chip is. By using this criterion, we can then estimate the channels sensitivity to DPA.
Ciphering data-path and Sub-key data-path are 32-bit wide. Iterations as specified in the Rijndael algorithm [10]. Both which control both data-paths so that they execute $N_r$ path. The controller (finite state machine) generates signals the ciphering data-path with the sub-key computation data-path. Channel Sub-key synchronises communicating channels. Channel Sub-key synchronises based on three self-timed loops synchronized through is described in fig. 8. It is basically an iterative structure, data paths [9]. The architecture of the AES crypto-processor is described in fig. 8. It is basically an iterative structure, based on three self-timed loops synchronized through communicating channels. Channel Sub-key synchronises the deciphering data-path with the sub-key computation data-path. The controller (finite state machine) generates signals which control both data-paths so that they execute $N_r$ iterations as specified in the Rijndael algorithm [10]. Both deciphering data-path and Sub-key data-path are 32-bit wide. The details of the architecture choices are discussed in [9].

Fig. 9 represents the constrained floorplan of the AES chip (fig. 8) realized with Soc Encounter. To be able to quantify the benefit brought by the hierarchical place and route methodology, a flat AES Netlist is also routed to be used as a reference. Let us name AES_v1, the circuit designed with the proposed methodology, and AES_v2 the reference one (flat place and route). The hierarchical approach has of course a cost in terms of silicon area. The core area of the first version (AES_v1 - hierarchical) is about 20% larger than the second version (AES_v2 - flatten).

To evaluate the benefits in terms of the DPA sensitivity, we computed for both versions the dissymmetry criterion for all the channels. Table 2 reports the most critical bits, i.e. the ones presenting the highest dissymmetry. For the second version (AES_v2 - flatten), the criterion value can reach up to 1.25. It represents an important source of leakage as demonstrated in equation 12 and illustrated in fig. 7. Note that, even though most of the channels present a low criterion value, the existence of some channels having a high criterion value greatly degrades the DPA resistance level of the circuit. Moreover, we observed that the most sensitive channels are never the same from one place and route to another, confirming that the place and route process is not under the designer’s control.

On the contrary, the hierarchical version (AES_v1) does not present any channel having a criterion value higher than 0.13 (Table 2). The channel net capacitance differences are therefore drastically reduced with the proposed approach.

### VII. Conclusion

This paper presented a logical and electrical formal analysis of the Differential Power Attack of secured Quasi Delay Insensitive asynchronous circuits. The definition of a formal model of the current dissipated in such circuits allowed us to apply DPA on this model, and thus identify sources of leakage. From this formal analysis, a design methodology is derived to limit the source of leakage by using hierarchical place and route. The gain obtained is illustrated with the design of a secure AES crypto-processor. It is shown that a significant improvement is achieved in theory, with the formal model.

The AES chip was sent for fabrication in August and will be back in October. DPA will be performed on the chip to characterize its resistance and verify the powerfulness of the method. The results will be presented at the conference.

### References