A Secure Asynchronous Hardware Implementation Of DES Cryptography Algorithm

Atabak Mahram **Behnam Ghavami** **Hosein Pedram**

{ mahram,ghavami,pedram}@ce.aut.ac.ir

Computer Engineering Department, Amirkabir University of Technology (Tehran Polytechnic) 424 Hafez Ave, Tehran 15785, Iran

Keywords

Asynchronous Circuit, Cryptography, QDI, Side-channel attack, DES algorithm.

ABSTRACT

QDI Dual-rail asynchronous circuits, if implemented carefully balanced, have natural and efficient resistance to side-channel attacks in cryptography applications. Due to hardware redundancy in previous balanced gate designs, there are many faults which can make them imbalanced without causing logical errors. Therefore, traditional logical testing methods are unable to test and verify if a gate is completely fault-free and hence balanced. This vulnerability opens the possibility of new methods of attacks, based on a combination of fault and power attacks in cryptographic applications. In this paper we present an asynchronous approach to hardware implementation of DES[1] cryptography algorithm that countermeasures against this new multiple side-channel attack.

1. Introduction

The wide spreading use of secure hardware systems in recent years has increased the research interest in cryptanalysis and countermeasure solutions. During the past decade, there has been extensive research to enhance the security of cryptosystems [1][3]. Although cryptographic algorithms have been designed to withstand rigorous cryptanalysis attacks, but they are vulnerable to attacks which analyze the physical characteristics of the hardware. Many attacks have been developed that exploit physical properties of implementations and information leaked through side channels[4][5][6][7].When a cryptographic module performs encryption or decryption, side channel parameters such as power dissipation, electromagnetic radiation or operating times correlate to the data being processed. This information can be used to find the secret key of the system.

Efficient methods for performing power analysis and fault analysis attacks have been developed which can analyze the side-channels and extract useful information [4][8].

[1] Data Encryption Standard

Differential Power Analysis (DPA), one of the well-known type of side-channel attack, is based on the fact that logic operation in standard static CMOS have power characteristics that depend on the input data. Using this attack, the key of an unprotected ASIC AES implementation can be found in less than three minutes [4]. This shows the weakness of an unprotected security hardware module.

Another side-channel attack is Differential Fault Attack (DFA) that uses information obtained from an incorrectly functioning hardware to find the secret key. DFA attacks were first proposed in [8] against hardware implementation of DES algorithm. Faults can be inject into a device even in the presence of tamper resistance packaging by introducing the device to elevated levels of radiation or temperature, atypical clock rate or incorrect voltage[9].

Several solutions have been proposed to countermeasure against side-channel attacks. One of the most effective countermeasures against DPA attacks is based on some specially designed balanced gates for which the power consumption behavior is independent of the input data. There has been several proposals of such balanced gates (e.g SABL [10], BSDT [11], WDDL [12]).previously proposed fault attack countermeasures have been based on the addition of redundancy to the hardware, usually in the form of error-detecting codes, to detect errors in the logical level (i.e. [13]).

Recently, in addition to asynchronous design advancements, it became clear that this methodology if implemented correctly has several features that are convenient for a secure hardware design [14][15].They demonstrate how dual-rail encoded asynchronous circuits improve security by eliminating data dependent power consumption. Symmetry in data processing circuits to assure independence of the power consumption from input values, and exploiting the propagating alarm signals to defend chip against fault injection are design aspects mentioned to increase the resistance to attacks. The asynchronous approach to increase the security is based on balancing the operation through special QDI logic cells. the analysis of three different QDI DES architectures and design styles demonstrate how the properties of a dual-rail encoding and four-phase handshake protocol significantly improve the DPA resistance [14].

Balanced asynchronous circuits are effective countermeasures for their respective attacks if the side-channels are considered separately. Joint consideration of both power and fault side channels attack can raises several practical security limitations of the approaches. In other words, balanced QDI circuits are vulnerable to a new hybrid attack, when power and fault attacks are considered together[2],

All the currently known balanced QDI circuits require hardware redundancy to ensure balanced computations. Weaknesses in the present balanced cells exist due to the redundancy of the cell. There exist many internal transistor level faults which will not affect the Boolean function of the cell, can not detect by traditional testing method, but will affect the balanced behavior of the cell.

In this paper we present a customized template for secure asynchronous hardware implementation. We implement a DES algorithm that countermeasure against DPA on faulty hardware attack by means of this new template. the rest of the paper is as follows. In section 2, we present asynchronous circuit design methodology, especially synthesis of QDI circuits with Persia synthesis tool. Vulnerability of existing library cells to DPA attacks is shows in section 3.The proposed library cell design approach and the DES implementation results based on secure cells is reported in section 4.finally we conclude in section 5.

2. Asynchronous Circuit Design

Asynchronous circuits represent a class of circuits not controlled by a global clock but rely on exchanging local request and acknowledge signaling for the purpose of synchronization. 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 an incoming data. It then performs the computation and sends the result via output channels. Communications through channels are controlled by handshake protocols [16].

An asynchronous circuit is called delay-insensitive if it preserves its functionality independent of the delays of gates and wires[20]. It is shown that the range of the circuits that can be implemented completely delay-insensitive is very limited[19].Therefore some timing assumptions exist in different design styles that must be hold to ensure the correctness of the circuit. Different asynchronous techniques distinguish themselves in the choice of the compromises to the delay-insensitivity.

Quasi delay-insensitive (QDI) circuits are like delay-insensitive circuits with a week timing constraint: isochronic forks. In and isochronic fork the difference between the delay through the branches must be less than minimum gate delay. QDI implementations appear to be the most appropriate – class of asynchronous circuits that can be synthesized automatically from large high-level behavior specifications. This is because of the week timing constraint that can be easily managed in this design style. Return to zero handshaking protocol with dual-rail data encoding that switch the output from data to spacer and back is the most common QDI implementation form. The most efficient QDI implementations are based on per-charge logic. That makes it easy to incorporate existing dynamic domino style power balanced structures in the QDI templates.

The encodings of the channels can be in a variety of ways. We use a dual rail encoding here the data channel contains a valid data (token) when exactly one of 2 wires are high. When the two wires are lowered the channel contains no valid data and is called to be neutral .

	d.t	d.f
Neutral("E")	0	0
Valid '0'	0	1
Valid '1'	1	0
Not used	1	1

Figure 1: Dual rail coding

One of the major protocols used in asynchronous circuits is four phase protocol. In a four phase protocol's sequence a receive action consists of four steps. (1) Wait for input to become valid:[L]. (2) Acknowledge the sender after the computation performed L^{ack}. (3) Wait for inputs to become neutral [~L]. (4) And lower the acknowledgement signal. A send action consists of four phases: (1) send a valid output R.(2) wait for acknowledge [R^{ack}]. (3) Make the output neutral ~R.(4) wait for acknowledge to lower [$\sim R^{ack}$].figure 2 shows a four phase handshake sequence.

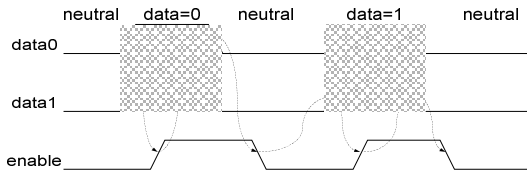

Figure 2: Four-phase protocol

2.1 Persia: A QDI asynchronous synthesis tool

Persia is an asynchronous synthesis tool developed for automatic synthesis of QDI asynchronous circuits[17]. The structure of Persia is based on the design flow shown in figure 1 which can be considered as the following three individual portions: QDI synthesis, layout synthesis, and simulation at various levels. The simulation flow is intended to verify the correctness of the synthesized circuit in all levels of abstraction.

CSP is a well-known language for description of concurrent systems which is accepted as a good description language for asynchronous systems.. Persia uses Verilog-CSP[2][18], an extension of the standard Verilog which supports asynchronous communications as the hardware description language for all levels of abstractions except the netlist which uses standard Verilog. The input of Persia is a Verilog description of a circuit. This description will be converted to a netlist of standard-cell elements through several steps of QDI synthesis flow. For

[2] Communicating Sequential Processes

simpler synthesis first arithmetic operations are extracted from the code and the major steps of synthesis only works on the codes without any arithmetic operations. This is done by the AFE which also replaces the arithmetic functions by standard library modules. The two major steps in Persia synthesis are Decomposition and TSYN. In the following subsections we briefly describe the functionality of these three stages.

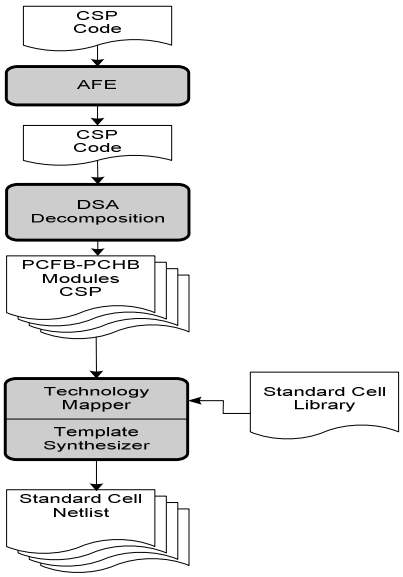

Figure 3: Persia synthesis flow [17]

2.1.1 AFE

Arithmetic operations are not synthesizable by TSYN (part of Synthesizer), so Persia extracts these operations from the CSP source code and then implements them with pre-synthesized standard templates. AFE extracts each assignment that contains arithmetic operations like addition, subtraction, comparison, etc and generates a tree of standard circuits which implement the extracted assignment.

2.1.2 Decomposition

Our synthesis approach is based on pre-design asynchronous four-phased dual rail templates. Each template can be considered as a simple pipeline stage. The most renowned form of these templates is named as pre-charge full buffer (PCFB)[19].A PCFB reads its data from input ports, performs the computations and writes the results on the output ports. A PCFB can have multiple inputs and outputs, have conditional inputs and outputs, and hold states. The circuit is similar to pre-charge domino-logic style circuits in synchronous designs except that instead of a global pre-charge signal local pre-charge signals are generated. The QDI timing constraint (i.e Isochoric fork) is local to each template. Figure 1 represents a PCFB buffer used in Persia synthesis tool.

The high-level Verilog-CSP description of even very simple practical circuits is not directly convertible to PCFB. The intention of Decomposition stage is to decompose the original description into a collection of smaller interacting processes that are compatible to these templates and are synthesizable in next stages of QDI synthesis flow

2.1.3 TSYN

Template Synthesizer, as the final stage of QDI synthesis flow, receives a Verilog-CSP source code containing a number of PCFB-compatible modules and optionally a top-level netlist and generates a netlist of standard-cell elements with dual-rail ports that can be used for creating final layout. The output of TSYN can be simulated in standard Verilog simulators by using the behavioural description of standard-cell library elements.

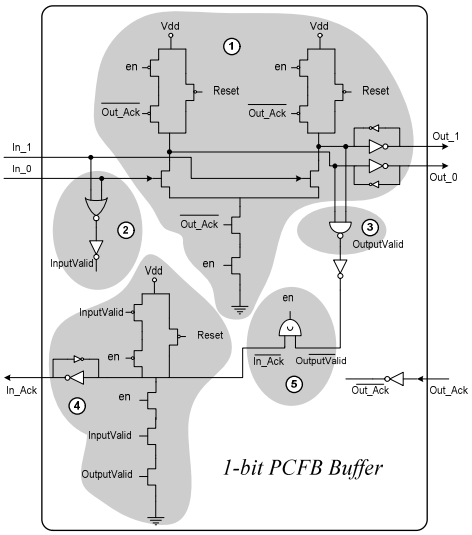

Figure 4: The PCFB 1-bit buffer

3. DPA attacks and vulnerabilities of existing cell libraries

DPA attacks use statistical techniques to determine the secret keys by observing power consumption [4]. In a typical attack, an attacker samples the target device's power consumption and builds a power trace. A high-speed analog-to-digital converter can be used to create these power traces. These measured power traces are compared with predicted power consumptions. To make a prediction a guess on the secret key is used. Several statistical and mathematical techniques are available to correlate the predictions and measurements. Based on these analyses the secret key can be found.

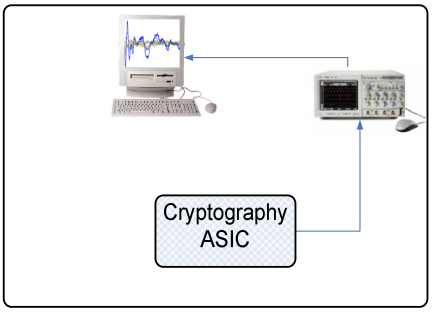

Figure 6: DPA attack platform

Initially our cell library was designed to get the highest performance of the system. This cell library has the potential of being attacked by power analysis but it is not possible to

perform a fault attack in this cell because any injected faults to this logic cell will cause the system to go to deadlock..
Figure 4 shows the transistor level circuit of a PCFB buffer. This circuit is venerable to power attacks yet it is better than a synchronous counterpart. The HSPICE simulation of a imbalanced PCFB buffer's consumed power is shown in figure 8.

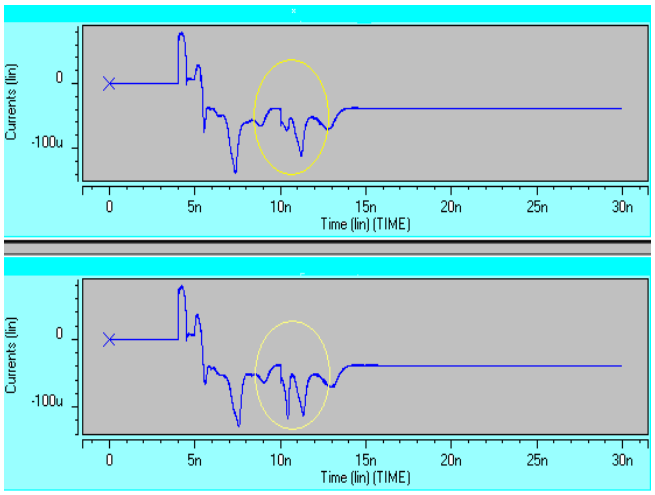

Figure 8: Power consumption of imbalanced PCFB Buffer

It is clear that the power consumption behavior of the system is dependant on the input data and hence it is power attackable. Kulikowski and etc. showed how one can incorporate a balanced system design in QDI asynchronous pipelines [2]. The authors also mentioned the possibility of a hybrid Fault and power attack to their system. in this kind of attack a fault is injected to the system that changes the balanced behavior of the system while the functionality is preserved. So the circuit becomes an unbalanced functional circuit that can be attacked. In next section we will change our logic cell to make them invulnerable to this Hybrid attack.

4. Cell customization and DES implementation

We balanced the buffers by adding redundancy to the circuit's unbalanced parts -The input validity checking circuit, the output validity checker and the pull-down computation logic(parts 1 to 3 in figure 9 respectively). Figure9 shows how we made these balancing in a PCFB Buffer gate. Parts 2 and 3 were not balanced originally. We balanced them with the addition of a duplicate gate and a C_Element. The C_Element's output changes when both of its inputs have the same value and their values are opposite to the C_Element's current value. If someone injects a fault in the balancing circuits of previously mentioned parts the circuit will go to deadlock.

Figure 9: Symmetric And gate

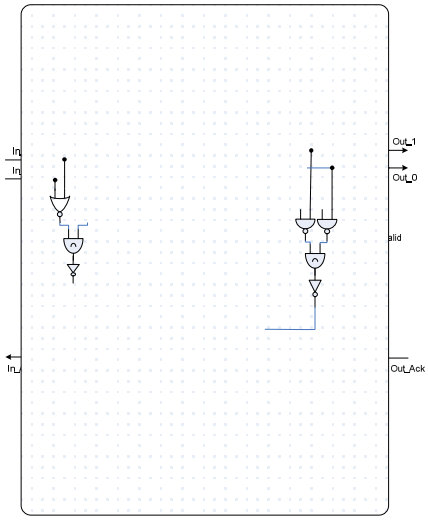

Figure 10: The Balanced PCFB 1-bit buffer

Figure 10 shows a balanced PCFB XOR circuit's computational network. The circuit is completely symmetric with respect to the input signals so it is obvious that the power consumption behavior of the circuit is independent of the input data.The circuit goes to a deadlock state in case of any faults .here again a C_Element helps us to find out the intended unbalancing of the circuit. If a fault is injected to the balancing circuit the C_Element Muller gate does not change its state because its inputs are not the same. So the normal operating cycle of the circuit will not happen and the circuit will go to a deadlock state. The power consumption behavior of the circuit is completely input independent and hence it is resistant to power attacks.

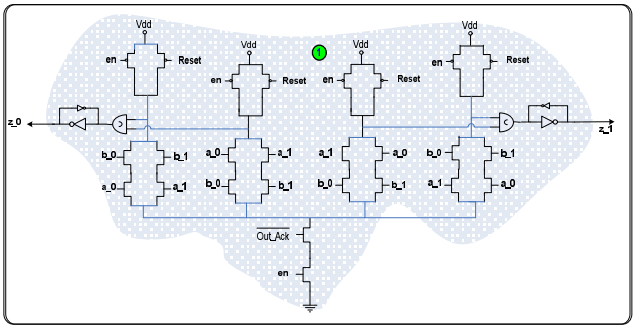

Figure 11: The balanced pull-down computation logic network of a 2-bit PCFB-XOR

We implemented a DES algorithm in HSPICE and our simulations of the power consumption are totally input independent. Figure11 shows the power simulations for two different inputs.

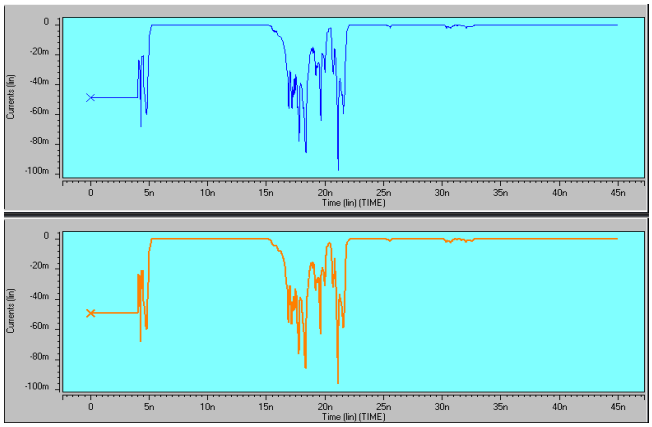

Figure 12: Power consumption of a DES stage hardware implemented with secure library cells

5. Conclusions and future works

This paper presented an asynchronous approach to design and implementation of a DES cryptography algorithm. To do this, some of the cells of the Persia asynchronous synthesis tool‘s library has been customized so that the power consumption of the cells has became input independent. We also presented a logic level method to test the faults that make our templates imbalanced. In case of fault attacks the system implemented with these templates will go to deadlock and hence the hybrid DPA – FA attacks to our cells are not possible. Result shows that this implementation has more resistance against multiple side-channel attacks compared with previous both synchronous and asynchronous solutions.

6. References

[1] P. Kocher, R. Lee, G. McGraw, A. Raghunathan and S. Ravi, “Security as a New Dimension in Embedded System Design,” *DAC*, pp. 753-760, 2004.

[2] K. Kulikowski, M. Karpovsky, and A. Taubin. DPA on faulty cryptographic hardware and countermeasures. In Fault Diagnosis and Tolerance in Cryptography, 3nd International Workshop, 2006.

[3] Hwang, P. Schaumont, K. Tiri and I. Verbauwhede, “Making Embedded Systems Secure,” submitted *IEEE Security & Privacy Magazine*.

[4] P. Kocher, J. Jaffe, and B. Jun, “Differential Power Analysis,” in *Proc.19th Intl. Advances in Cryptology Conference-CRYPTO’99*, Aug. 1999,pp. 388–397.

[5] J. J. Q. adn D. Samyde, “Side-channel Cryptanalysis,” in *Proc. SECI*, Sept. 2002, pp. 179–184.

[6] P. Kocher, “Timing Attacks on Implementations of Diffe-Hellman, RSA, DSS and Other Systems,” in *Proc. 16th Intl. Advances in Cryptology Conference-CRYPTO’96*, Aug. 1996, pp. 104–113.

[7] J. J. Quisquater and D. Samyde, “ElectroMagnetic Analysis (EMA): Measures and Counter-measures for Smart Cards,” in *Proc. 19th Intl.Advances in Cryptology Conference-CRYPTO’99.*

[8] Biham, E. and A. Shamir, "Differential fault analysis of secret key cryptosystems", CRYPTO 97, LNCS 1294, pp.513-525

[9] Bar-El H., H. Choukri, D. Naccache, M. Tunstall and C. Whelan. "The Sorcerer’s Apprentice Guide to Fault Attacks". Cryptology ePrint Archive, Report 2004/100. Available: http://eprint.iacr.org/2004/100.pdf

[10] Tiri, K., M. Akmal, and I. Verbauwhede. A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards. 28th European Solid-State Circuits Conference (ESSCIRC 2002), pp. 403-406, September 2002.

[11] MacDonald, D.J., A Balanced-Power Domino-Style Standard Cell Library for Fine-Grain Asynchronous Pipelined Design to Resist Differential Power Analysis Attacks. Master of Science Thesis. 2005, Boston University

[12] Tiri, K. and I. Verbauwhede, A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. Design, Automation and Test in Europe Conference (DATE 2004), pp. 246-251, February 2004.

[13] Kulikowski, K., M. Karpovsky, and A. Taubin. Robust Codes for Fault Attack Resistant Cryptographic Hardware. in Fault Diagnosis and Tolerance in Cryptography, 2nd International Workshop. 2005. Edinburgh.

[14] Fraidy Bouesse, Laurent Fesquet, Marc Renaudin “QDI circuit to Improve Smartcard Security”, 2nd Asynchronous Circuit Design Worshop (ACID2002), Munich; Germany, 28-29 Januray,2002.

[15] G. F. Bouesse, M. Renaudin, S. Dumont, and F.Germain. DPA on quasi delay insensitive asynchronous circuits: Formalization and improvement. In DATE, 2005.

[16] Marc Renaudin, “Asynchronous circuits and systems: a promising design alternative”, Microelectronic for Telecommunications : managing high complexity and mobility” (MIGAS 2000), special issue of the Microelectronics-Engineering Journal, Elsevier Science, GUEST Editors : P; Senn, M. Renaudin, J, Boussey, Vol. 54, N° 1-2, December 2000, pp. 133-149.

[17] http://www.asynch.ir/persia

[18] Arash Seifhashemi, Hossein Pedram, "Verilog HDL, Powered by PLI: a Suitable F ramework for Describing and Modeling Asynchronous Circuits at All Levels of Abstraction", Proc. Of 40th DAC, Anneheim, CA, USA, June 2003.

[19] A.M.Lines. “Pipelined Asynchronous Circuits”, M.Sc. Thesis, California Institute of Technology, June 1995, revised 1998.

[20] Jens Sparso, Steve Furber, Principles of Asynchronous Circuit Design – A System Perspective, Kluwer Academic Publishers, 2002