Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

An EDA tool for implementation of low power and secure crypto-chips

Behnam Ghavami *, Hossein Pedram, Mehrdad Najibi

Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Available online 11 October 2008

Keywords:
Cryptography
Side channel attack
Asynchronous design
Low power
QDI

ABSTRACT

Regarding the significant mathematical immunity of recent cryptographic algorithms, attacks considering the physical aspects of these algorithms, known as side channel attacks, have received much of interest. Today, it is quite clear that asynchronous circuits possess considerable inherent countermeasure capabilities against side channel attacks, and therefore they are more immune for cryptographic systems compared to synchronous design. However, due to lack of automatic synthesis and optimization tools for these circuits, implementation of secure asynchronous circuits encounters many difficulties. In this paper, a fully automated secure design flow and a set of secure library cells resistant to power analysis and fault injection attacks are introduced for quasi delay insensitive asynchronous circuits. In the proposed flow, a high-level description of the system is received in Verilog format powered by some special macros, and then the corresponding specification will be decomposed into smaller circuits directly mappable to predefined circuit templates. With the use of a special standard-cell library, the final circuit is resistive to differential power analysis on faulty hardware attack. We suggest a restructuring on the conditional statements in the high-level description of the circuit which leads to a considerable optimization in power consumption after the decomposition of the system. To verify the efficiency of our presented design flow, we implemented data encryption standard (DES) and advanced encryption standard (AES) algorithms, and we showed 23% less power consumption compared to the existing data driven decomposition asynchronous synthesis method. Also, these implementations are three times faster than the synchronous implementations on average, in TSMC 0.18 μm technology.

1. Introduction

Cryptographic systems are an integral part of modern digital society providing solutions to secure information from unauthorized access. The degree of complexity of the algorithm, which implies the resistibility of the cryptographic algorithm exposed to the attacks is discussed in the realm of cryptanalysis. The validity of mathematical security models for the cryptographic algorithms is based on the fact that the attackers do not have access to the intermediate computational data. So, any kind of information about the intermediate data will simplify the cryptanalysis dramatically. Since cryptographic algorithms in mathematics perspective have high security, nowadays attackers turn to analyzing the physical aspects of the system to achieve the intermediate computational data. There are some kinds of attacks which take advantage of the implemented physical properties and leaked information from side channels which are known as side channel attacks [1]. Most of the cryptographic algorithms, which are secure in the realm of mathematical model, have been analyzed by these kinds of attacks, and have been broken down easily [1]. Mobile cryptography devices such as smartcards and mobile

* Corresponding author. Tel.: +98 2164542714; fax: +98 2164542700.
E-mail addresses: ghavamib@aut.ac.ir (B. Ghavami), pedram@ce.aut.ac.ir (H. Pedram), najibi@ce.aut.ac.ir (M. Najibi).

0045-7906/$ - see front matter © 2008 Published by Elsevier Ltd.
doi:10.1016/j.compeleceng.2008.06.014

computers are especially vulnerable to these attacks since the physical hardware implementing the algorithms is easily accessible. As a result, introducing these kinds of attacks has caused serious challenges in the smartcard industry and other cryptography equipments.

Several solutions have been proposed to countermeasure against side channel attacks. Today, it is quite clear that asynchronous circuit design methodology is suitable for the implementation of secure cryptographic systems. Over the years, various publications [9,15,21] have claimed that asynchronous circuits are less susceptible to side channel attacks. However, because of the lack of automatic synthesis tools for these circuits, implementation of secure asynchronous circuits faces difficulties.

One of the main issues in the design of cryptographic systems is to reduce the power consumption, especially for portable systems such as smart cards. This fact requires new ways in design and implementation of crypto-chips. However, designing high-performance VLSI is challenged by high power consumption as the clock frequency, die size and number of transistors increase by the technology improvements. Asynchronous design presents potential solutions to some of VLSI design challenges by lower dynamic power consumption and potential better performance [21]. Most of these benefits are due to the elimination of the global clock signal by employing local message passing as the main synchronization method. Unfortunately, raw asynchronous circuits can have some power overhead [2]. Therefore, another obstacle to the wide use of asynchronous circuits in cryptographic applications is the lack of automatic optimization methods. While contemporary asynchronous designs are mainly based on manual optimization, developing automatic optimization methods is an urgent need for these circuits.

The purpose of this paper is to present a new method to eliminate the physical limitations of contemporary cryptographic systems. This methodology prepares a complete design cycle of secure and low power crypto-chips by introducing a practical and high quality electronic design automation tool, which eliminates all sources of leaked information from side channels. The design flow presented here is based on a quasi delay insensitive (QDI) asynchronous circuits in which customized standard cells have been used. This method not only preserves performance and low power consumption but also is a suitable solution for promoting countermeasure against power, timing and fault injection attacks at the same time. Also, circuits implemented by this method are shown to be resistant to the new and effective attack which is made up of fault injection and power analysis [7].

The rest of this paper is organized as follows: the next section describes the side channel attacks and overviews the solutions that have been introduced in the literature and explain their vulnerability. Section 3, introduces the asynchronous design and some benefits of using asynchronous circuits in secure hardware. Section 4presents a synthesis tool for QDI asynchronous circuits. Section 5 draws the power optimization method in outline. Then in Section 6, we explain a standard-cell library which is customized for power balancing and insensitivity to fault injection. Section 7 shows the results of applying this method to data encryption standard (DES) and advanced encryption standard (AES), and finally Section 8 concludes this paper.

2. Side channel attacks and countermeasure solutions

Before 1995, cryptographic algorithms were considered as mathematical functions that get a set of inputs that contain the context and the key and produce outputs corresponding to these inputs. Usually, traditional analysis to breaking the good cryptographic algorithm by mathematical models has not been successful [1]. However, in the real world cryptographic algorithms are hardware–software systems instead of mathematical functions. The implementation of a cryptographic algorithm results in a black box that has several observable physical properties such as power consumption, electromagnetic radiation, surface temperature, time required to complete an operation, or sometimes has several faults. All these properties that can be observed to change while cryptographic operations are processed are information sources which can potentially be used to reveal parts of the secret key. Each of these side effects and transactions with the environment can leak some information from the system, and is a threat to the security of crypto-systems. Such information sources are called side channels, and can be exploited by an attacker in side channel attacks. Up to now, many kinds of important side channels have been explored [1,2,16].

Incresase in the hardware faults and errors occurring during the operation of a cryptographic module in fact have been demonstrated to sometimes even greatly affect the security. These faulty behaviors or outputs may also become important side channels. Differential fault analysis (DFA in the sequel) usually causes some sort of physically erroneous operation to occur in a cryptographic device and then measures resulting phenomena [16,10].

The power consumption of a cryptographic device may provide much information about the operations that take place and the involved parameters. So, power consumed by cryptographic devices is one of the sources of leaked information of side channels. Power analysis attack is actually the current research focus of side channel attacks. Basically, power analysis attack can be divided into simple and differential power analysis [1] (referred to as SPA and DPA, respectively). In SPA attacks, the aim is essentially to guess from the power trace which particular instruction is being executed at a certain time and what values the inputs and outputs have. Therefore, the adversary needs an exact knowledge of the implementation to mount such an attack. On the other hand, DPA attack does not need the knowledge of the implementation details and alternatively exploiting statistical methods in the analysis process. In DPA, measured power traces are compared with a prediction on the power consumption. To make the prediction a guess on the secret key is used. Only if the secret key hypothesis is being correct, then the predicted and the actual power consumption are correlated. Several statistical techniques are available to perform the comparison between the predictions and the measurements.

In addition to the mentioned side channel attacks, many other attacks (i.e. EMA attack [2]) are developed by different research groups based on information leaked from their hardware implementation. Due to the vulnerability of good cryptography algorithms against side channel-attacks, it is necessary to develop efficient methods to resist them against side channel attack. To prevent side channel attacks, several countermeasure solutions have been proposed which aim to reduce or eliminate the amount of information which can be inferred about intermediate data in the hardware implementation of a cryptographic algorithm. In the following, we will review the available countermeasures solutions against fault and power attacks and their vulnerabilities.

2.1. Fault attack countermeasures

Most proposed fault attack countermeasures have been based on adding redundancy to the device, usually in the form of error-detecting codes, to detect errors in the logical values of the processed data [8,16]. Karri et al. [25] proposed to add circuitry to perform, in parallel with the encryption, a reverting of the performed operations (with various possible levels of granularity), and to compare them with the input values to ensure that no error has occurred. An asynchronous circuit is characterized by the fact that its execution flow is not controlled by a central clock; instead, various components may operate at their own speed, warning their predecessors when they are ready to process data. Dual-rail encoding is often used to construct and secure these circuits [15].

2.2. Power attack countermeasures

Developing countermeasures against power attacks has been an active research area. The ultimate goal of power attack countermeasures is to increase the number of samples required to reveal the sub-key to a level where it is not feasible to perform such attacks. A simple approach against power attack involves introducing noise into power consumption measurements. Introducing noise increases the number of samples required for an attack, possibly to an unfeasibly large number. In addition, execution timing and order can be randomized to generate a similar effect. Another popular approach is to randomize the execution sequence, i.e. keep operations the same, but permute the order (e.g. in DES, the S boxes are looked up in a random order). However, according to [18], unless this random sequencing is done extensively throughout the computation, which may be impossible since the specification forces a causal ordering, it can be undone and a canonical order recreated by signal processing.

Power attack is based on the fact that logical operations in standard static CMOS have power characteristics that depend on the value of input data. So one of the most effective countermeasures against power analyzing attack is based on the use of specially designed balanced gates for which the power consumption is equal for all data and all transitions of the gate. Several such gates have been previously presented (e.g. DyCML [4], SABL [3] and WDDL [5]). To make the power consumption of a gate independent of the input values, these gates achieve the following goals: the output switching is independent of the input values, and the total load capacitance always sums up to a constant value (Fig. 1).

2.3. Motivation

Most of balanced gates such as those from [3,4,30] have no countermeasures against fault injection attacks, and require additional protection. More importantly, since differential and dynamic (DD) approaches form, they require dynamic (domino) logic cell design. The usage of DD gates is limited to custom or semi-custom design that greatly limits the perceived universality of DD based circuitry. The following are two major reasons of why electronic design automation (EDA) support of dynamic logic based design is very difficult for synchronous methodology [17]. First, each synchronous dynamic gate requires a clock input and uses both levels of clock signal – it means that from the point of view of EDA tools each gate behaves like a flip-flop. Second, due to early/late arrival, charge sharing and clock distribution problems with small clocking

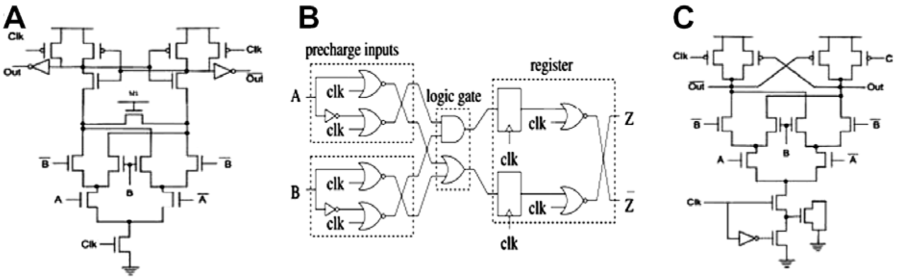


Fig. 1. (A) SABL XOR gate with enhanced special DPDN [3], (B) cascaded WDDL AND gate and flip-flop [7] and (C) DyCML XOR gate [4].

granularity and uncertainty about worst case delay makes static timing analysis (STA) of dynamic circuits very problematic. STA is the core of any synchronous EDA approach. As a result, no EDA tool support is available for synchronous design based on dynamic logic.

Recently, it has been shown that asynchronous circuits possess considerable inherent countermeasure against side channel attacks [6,9,31]. Jacques Fournier et al. presented the security evaluation of an asynchronous smart-card system, and showed that a secure asynchronous processor has interesting tamper-resistance properties [29]. However, the lack of automatic synthesis tools and optimization methods are important obstacles to the wide use of asynchronous circuits in cryptographic applications. Earlier, a secure hardware design flow (SHDF) based on asynchronous micro-pipelines has been proposed [6]. Their methodology allows incorporation of existing synchronous dynamic gate designs and circuit structures that allow automated design resistant to side channel attacks. They proposed a balanced library which designed specifically for the fine-grained asynchronous pre-charged half buffer (PCHB) template that is known as balanced symmetric with discharged tree (BSDT) [6].

It is important to note that the balanced gates are effective countermeasure for power attack if the side channels are considered separately. A joint consideration of both power and fault side channels raises several practical security limitations of the approaches [7]. All the currently known synchronous and asynchronous balanced gate designs (e.g. SABL, BSDT) require considerable hardware redundancy and overhead to ensure balanced computations. Much of this hardware redundancy is not directly associated with the logical or boolean function of the gate; it is present to ensure power balance during computations. Weaknesses of the present balanced gate designs exist due to the redundancy of the gate; there exist many internal transistor level faults which will not affect the function of the gate but will affect the balance of the gate. In the following, we introduce asynchronous circuits and explain the motivation of using asynchronous circuits in secure hardware.

3. Asynchronous circuits

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. An asynchronous circuit is called delay-insensitive (DI) if it preserves its functionality, independent of the delays of gates and wires [12]. It is shown that the range of the circuits that can be implemented completely DI is very limited. Therefore, some timing assumptions exist in different design styles that must hold to ensure the correctness of the circuit. Different techniques distinguish themselves in the choice of the compromises to the delay-insensitivity. Quasi delay insensitive (QDI) circuits are like DI circuits with a weak timing constraint [13].

An asynchronous circuit is composed of individual modules which communicate with each other by means of point-to-point communication channels (Fig. 2). 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 [12].

The encodings of the channels can be in a variety of ways. Return to zero handshaking protocol with dual-rail data encoding that switches the output from data to spacer and back is the most common QDI implementation form. The data channel contains a valid data (token) when exactly one of two wires is high. When the two wires are lowered the channel contains no valid data, and is called to be neutral (Fig. 3). One of the major protocols used in asynchronous circuits is the four-phase protocol. Fig. 4 shows a four phase handshake sequence. Using four-phase handshaking protocol with dual-rail data encoding, caused data independent time and power emissions which is necessary for side channel attacks resistant crypto-chips.

Asynchronous circuits are expected to offer a number of advantages over their synchronous counterparts when designing secure crypto-chips in the realm of VLSI design [12,13]. Since there is no clock used in any part of the asynchronous circuit, synchronization problems between clock domains do not exist. This means that these circuits do not have problems associated with clock distribution. The clock in a synchronous system has to be chosen to enable'worst case' operation. Asynchronous circuits use completion detection. While the 'worst case' operation would require the same time in both design approaches, asynchronous circuits would be able to work faster for the remaining cases. For circuits whose average case and worst case performance differ, the average operation speed of an asynchronous design over multiple operations would be higher than a synchronous design. A synchronous circuit continues to'operate' even if it has nothing to do, and it consumes dynamic power during such idle states. An asynchronous circuit would not be triggered in such a case and it would simply wait. In other words, asynchronous circuits do not consume idle power.

Asynchronous circuits could become a trustworthy platform for implementation of secure crypto-systems against side channel attacks [9,15]. Asynchronous circuits adapt to their environments which means that they should tolerate many

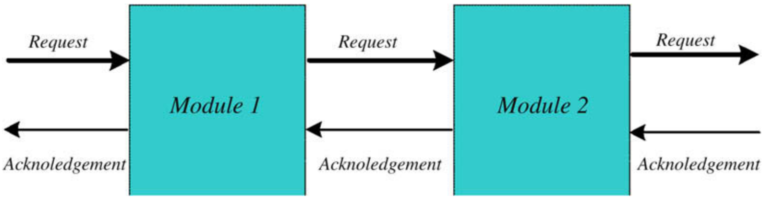


Fig. 2. Handshake based communication between asynchronous modules.

	d.t	d.f
Neutral(“E”)	0	0
Valid ‘0’	0	1
Valid ’1’	1	0
Not used	1	1

Fig. 3. Dual-rail coding.

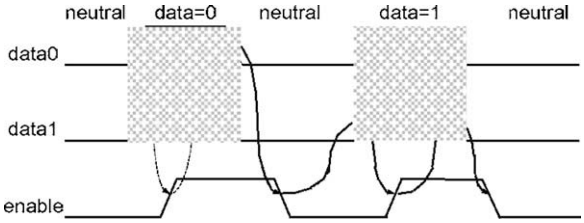


Fig. 4. Four-phase hand-shaking protocol.

forms of fault injection (power glitches, thermal gradients, etc.). This makes fault sensing easier since just major faults need to be detected and reacted to. This is desirable since minor fluctuations in environmental conditions are normal during real-world operation. The elimination of the clock signal in these circuits, clock glitch attacks are removed, and triggering data detection at specific points of the data processing flow is very difficult. By replacing a synchronous processor with an asynchronous one (no clock harmonics), electromagnetic signature is strongly reduced. Removing clock results in significantly flatter noise and electromagnetic interference (EMI) spectrum across the frequency domain (10 dB drop according to [21]). Asynchronous multi-dimensional pipelined array architectures [23] can eliminate data dependent timing, and thereby secure implementations against differential timing analysis. Asynchronous circuits typically use a redundant encoding scheme (e.g. dual-rail). This mechanism provides a means to encode an alarm signal (e.g. use 11 = alarm in a dual-rail scheme [15]). QDI asynchronous circuits comprising dual-rail codes can be balanced to eliminate data dependent power consumption. Return to zero (RTZ) signaling is also required to ensure data independent power emissions.

Despite all its perceived advantages for implementation of crypto-chips, asynchronous design methodology has not seen a widespread acceptance. The lack of automatic synthesis tools for these circuits is a major obstacle for using this methodology. In the following sections we introduce a new methodology for design and implementation of asynchronous circuit. Overhead that usually incorporate with automatic synthesis of asynchronous circuit can be eliminated by using our automatic optimization methods in the duration of synthesis process. Furthermore, our proposed balanced standard cells are customized for detecting faults occurring in hardware redundancy that is required for power balancing.

4. AsyncTool: synthesis of QDI asynchronous circuits

QDI circuits appear to be the most appropriate implementation for the class of asynchronous circuits that can be synthesized automatically from large high-level behavior specifications. This is because of the weak timing constraint that can be easily managed in this design style. As one of the main benefits of asynchronous design is the relaxation of timing constraints, the correct functionality of QDI asynchronous circuit still requires some weakened timing constraints. Most of the asynchronous methodologies including QDI assume unbounded finite delays for both functional elements and wires which can be considered as a great timing relaxation. In QDI methodology for a special class of branched wires known as isochronic forks, while the unbounded finite delay assumption is still valid for both the root and all the branches, correct functionality requires that the delay of the branches is nearly equal. It has been shown that practical asynchronous circuits cannot be made without isochronic forks [12].

The most efficient QDI implementations are based on per-charged logic. This makes it easy to incorporate existing dynamic domino style power balanced structures in the QDI templates. Most importantly, few of QDI-asynchronous EDA tools address fine-grained asynchronous dynamic logic pipelining which is of major importance for security. Another drawback about these EDA tools is that library cells which they used are not compatible with differential dynamic cells such as SABL. At present, most QDI circuits are designed using PCHB (pre-charge logic half-buffer) and PCFB (pre-charge logic full-buffer) templates [19].

AsyncTool (also known as Persia) [20] is an asynchronous synthesis tool developed for automatic synthesis of QDI asynchronous circuits. AsyncTool uses PCFBs for its predefined templates. A PCFB template is an asynchronous buffer circuit that in each cycle of its operation it reads some inputs, performs a particular calculation, and then writes the results to one or more of its output ports. Fig. 5 shows the internal implementation of the simple buffer described using the traditional custom transistor netlist. The following sub-circuits can be enumerated for the circuit: (1) output generation circuit, (2) input

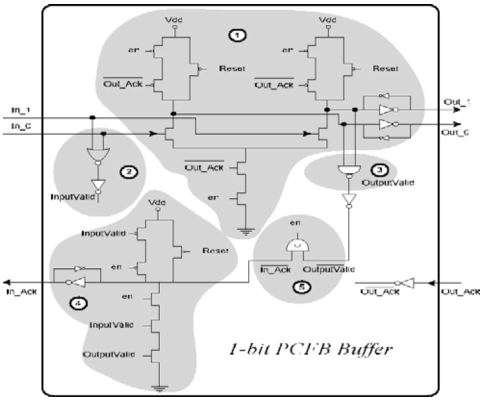


Fig. 5. Internal structure of a 1-bit PCFB buffer.

validity check circuit, (3) output validity check circuit, (4) a sub-circuit that generates the acknowledgement of inputs and (5) a sub-circuit that generates a signal. While traditionally asynchronous pipeline templates are implemented as full-custom transistor netlists, using nonstandard layout tools, standard-cell approach is gaining popularity [6]. Standard-cell implementation of these templates can eliminate isochronic fork timing constraint. Each pipeline template in our synthesis tool is composed of a number of previously laid out standard cells that can be connected to each other with nearly no special timing constraints. Our cells are designed to encapsulate all isochronic forks inside to simplify the task of layout generation [27].

The structure of AsyncTool is based on the design flow shown in Fig. 6 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 at all levels of abstraction.

CSP (communicating sequential processes) is a well-known language for the description of concurrent systems, and is accepted as a suitable description language for asynchronous systems. While CSP can powerfully describe concurrency, synchronization and decision making, it is not yet standardized. There exist only a limited number of supporting tools for pure CSP. On the other hand standard languages such as Verilog cannot be used directly to specify asynchronous circuits due to the lack of synchronization mechanisms. In [32], we showed that it is both possible and easy to use a standard

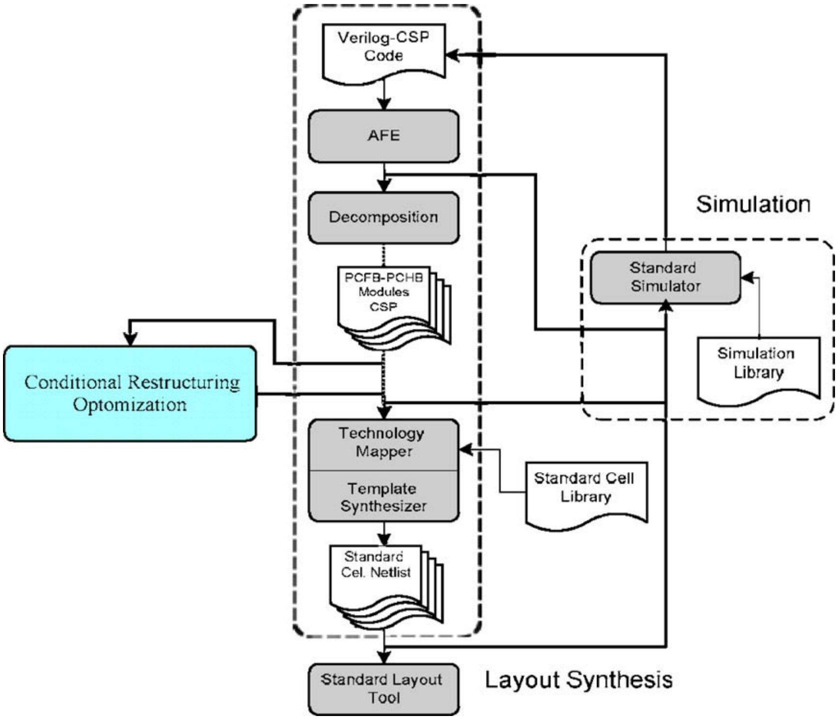


Fig. 6. AsyncTool synthesis flow.

HDL language like Verilog HDL, along with PLI to model asynchronous circuits at all levels of abstraction. Our method allows CSP codes to be simulated on ordinary Verilog simulators. AsyncTool uses Verilog–CSP [11], an extension to standard Verilog for the purpose of expressing the input description. READ and WRITE macro operations are added to the Verilog in order to model the handshake operation on communication channels. The Verilog language lacks this capability, but enhancement is possible by adding macros that are implemented as PLI. Therefore, the input to the synthesis tool is an asynchronous circuit description in Verilog–CSP which includes READ and WRITE macros to facilitate asynchronous communication over the channels. Design description will be converted to a netlist of standard-cell elements through several steps of QDI synthesis flow. In the following subsections, we briefly describe the functionality of these three stages.

4.1. Arithmetic function extractor (AFE)

Technology-mapper, as a part of template synthesizer (TSYN), is only able to synthesize one-bit assignments containing logical operators like AND, OR, and XOR, etc. Arithmetic operations are not synthesizable by TSYN, so AsyncTool extracts these operations from the CSP source code and then implements them with pre-synthesized standard templates. This is the role of the first stage of our asynchronous synthesis flow called AFE. AFE extracts each assignment that contains arithmetic operations like addition, subtraction, and comparison, and generates a tree of standard circuits which implements the extracted assignment. The communication between the main circuit and the arithmetic circuit is made by introducing new channels and added READ/WRITE macros. As a result, the main circuit will contain only logical assignments, and arithmetic computations will be performed in standard unconditional modules that are designed and included in the library.

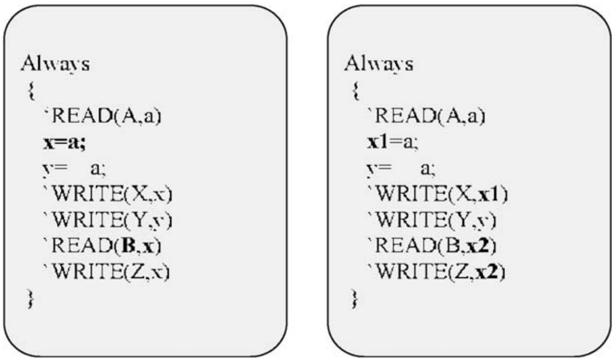
4.2. Decomposition

Our synthesis approach is based on pre-design asynchronous dual-rail templates. Each template can be considered as a simple pipeline stage. The high-level CSP description of even the very simple practical circuits is not directly convertible to PCFB templates. The intention of decomposition stage is to decompose the original description into an equivalent collection of smaller interacting processes that is compatible to these templates, and they are synthesizable in the next stages of QDI synthesis flow. Decomposition also enhances the parallelism between the resultant processes by eliminating unnecessary dependencies and sequences in the original CSP description. The major steps of basic decomposition are dynamic single assignment (DSA) form and projection.

In DSA phase, the sequential program is converted to dynamic single assignment form. This conversion can reduce the number of operations performed on a single variable, and can help create simpler modules which can be fitted into the circuit templates. After DSA, only true data dependencies will remain in the code. Fig. 7 shows the DSA conversion. Once the program is in DSA form, the technique of projection can be applied to break up the program into a concurrent system of smaller modules. This involves forming a projection set for each variable which contains the variable itself and all the dependent variables. When the program is projected into a number of sets, a new module, created for each set, contains only the statements of the original input description which involve the variables and channels of that set. Fig. 8 shows the processes that are built based on the projection sets of the corresponding variables which are shown in Fig. 7.

4.3. Template synthesizer (TSYN)

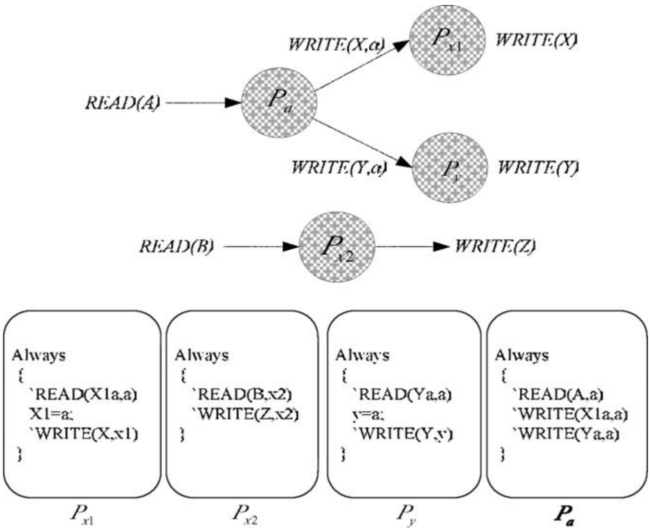
TSYN, as the final stage of QDI synthesis flow, receives a CSP source code containing a number of PCFB-compatible modules and optionally a top-level netlist, and generates a netlist of standard-cell elements with dual-rail ports that can be used for creating final layout. TSYN can synthesize all logical operations including AND, OR, and XOR with conditional or unconditional READ and WRITES. In addition, TSYN adds acknowledgement signals to I/O ports and converts the top-level netlist to



```
Always
 {
   `READ(A,a)
   x=a;
   y=   a;
   `WRITE(X,x)
   `WRITE(Y,y)
   `READ(B,x)
   `WRITE(Z,x)
 }
```

```
Always
 {
   `READ(A,a)
   x1=a;
   y=    a;
   `WRITE(X,x1)
   `WRITE(Y,y)
   `READ(B,x2)
   `WRITE(Z,x2)
 }
```

Fig. 7. DSA conversion.



```
Always
 {
   `READ(X1a,a)
   X1=a;
   `WRITE(X,x1)
 }
```

```
Always
 {
   `READ(B,x2)
   `WRITE(Z,x2)
 }
```

```
Always
 {
   `READ(Ya,a)
   y=a;
   `WRITE(Y,y)
 }
```

```
Always
 {
   `READ(A,a)
   `WRITE(X1a,a)
   `WRITE(Ya,a)
 }
```

Fig. 8. Projection phase of decomposition.

dual-rail form, and makes appropriate connections between ports and acknowledge signals. Since the template synthesizer is limited to one-bit logical expressions, a utility program (assignment expander) is used to convert multiple-bit expressions into one-bit expressions. The output of TSYN can be simulated in standard Verilog simulators by using the behavioral description of standard-cell library elements. For correct functionality, isochronic fork property must be asserted in the final implementation of the circuit. Violating this property can lead to the introduction of logical hazards, premature firing of signals, and generation of unwanted tokens. Our synthesis tool proposes a new solution for this problem. This method is based on handling isochronic forks inside the manually laid out standard cells. Inter-cell connection between these cells while it is not completely DI has no isochronic forks. Since AsyncTool does not impose any timing constraints on physical design, it can work with every standard back-end layout tool. The major technique that assists AsyncTool in achieving this goal is the innovative selection of the standard cells and some modifications in the TSYN [27].

5. Power optimization

When we use automatic asynchronous circuit design to implement asynchronous circuits, a considerable power overhead can be resulted [13]. While contemporary asynchronous designs are mainly based on manual optimization, developing automatic optimization methods is an urgent need for these circuits. Among the various asynchronous synthesis methods, data driven decomposition [22] shows a higher capability to generate high-performance and lower power systems (AsyncTool uses this methodology).

As mentioned in the previous section, in decomposition stage, the high-level sequential specification of the circuit is broken into a set of communicating modules that are individually synthesized at lower levels. The inter-module communications mapped out during this step consume the bulk of power by extra circuitry as data must not only be sent, but also be validated and acknowledged. Since, decomposing forms the general structure of the system as the main synthesis step, it affects largely on the power consumption of the final circuit.

The decomposition stage causes an increase in power consumption as well as circuit area. In [26], we presented three methods to tackle the problem. Two of the methods are for particular circuits, and the third method is general. In the third method, which utilized the simulated annealing (SA) algorithm, the decomposed circuit is converted to a number of sets where some of the original modules are combined, and accordingly some of intermediate channels are canceled.

Furthermore, we optimize high-level description of the design utilizing conditional structures to reduce handshaking overheads in the final decomposed circuit which consequently leads to the power reductions. As shown in Fig. 9, after decomposition, a conditional variable in the high-level description is distributed over the broken blocks; all the inputs, outputs and the middle blocks, which compute the results between inputs and outputs, must handshake with conditional variable block. Handshaking is the main source of power consumption in QDI asynchronous circuit [24], so with removing extra handshaking a huge optimization in power consumption can be resulted.

To preserve functionality of the generated circuit, it is required to guarantee that the generated circuit for the body statement is only activated whenever the condition is true. It can be done by ensuring that none of the fine grain processes of the body statement will generate an output value when the condition is not true. One way to achieve this goal is that all these processes receive the conditional variable. Consider the if statements; due to the nature of decomposition algorithm, the process

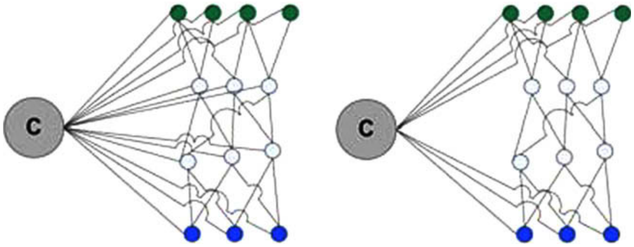


Fig. 9. Handshaking removed after using optimization.

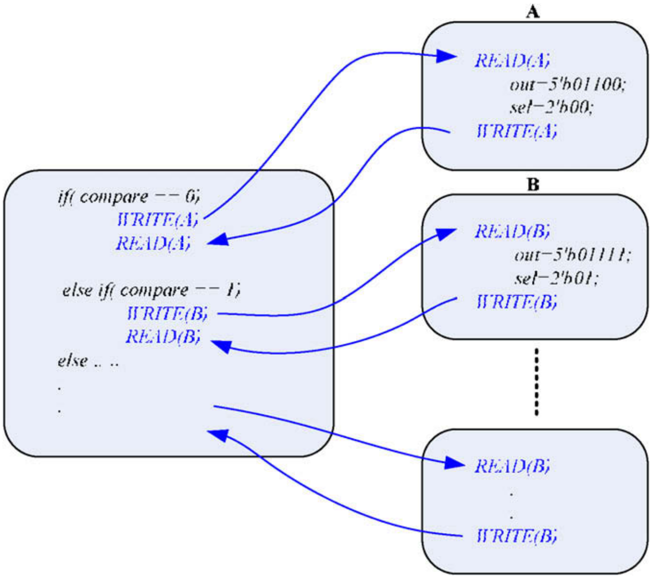
which is corresponding to generate the value of conditional variable must send the value to all the fine-grain processes generated from the decomposition of the body statement.

Being aware of the synchronization mechanism of asynchronous circuits, conditional activation of the body statement can simply be done by conditional writing on and reading from its input and output ports. So it is only necessary to apply condition on input and output channels of the body statement which eliminates the need of distributing the conditional variable to all the processes of the body statement, so the number of extra handshake communications can be reduced considerably which in turn can lead to effective reduction on area and power consumption of the circuit.

The computational portion of the conditional body statement can be moved to a separated block that only performs the computation itself without applying any condition. In this case, conditions are applied by conditionally writing the operands to the separated computational block and conditionally reading the results. When we apply conditional restructuring optimization (CRO), all computations inside the conditional block are removed, and will be added to a new block which computes the results unconditionally. CRO method acts as shown in Fig. 10. As shown in it, some of the extra conditional applications were eliminated from the computation section using the validity check circuitry with the corresponding power overheads. After applying the CRO, the data path synthesize is done without the extra conditions, so some extra handshaking is removed and the conditions remain only in the controller.

6. Cell-library customized for security

Our synthesis approach, based on the predefined templates, is thoroughly independent of the detailed implementation of the template cells. Since the basic templates are based on differential dynamic cells, almost all the existing or newly introduced dynamic circuit structures can be incorporated into our standard-cell library. Using this scheme, new circuit structures do not have to be redesigned or invented for a particular application in order to be incorporated into the flow. Reusability of intellectual property (IP), decrease of the design development and time-to-market would be inherent results of such a synthesis scheme.



```
if( compare == 0)
        WRITE(A)
        READ(A)

else if( compare == 1)
        WRITE(B)
        READ(B)
else .. ..
.
.
```

```
READ(A)
    out=5'b01100;
    sel=2'b00;
WRITE(A)
```

```
READ(B)
    out=5'b01111;
    sel=2'b01;
WRITE(B)
```

```
READ(B)
.
.
WRITE(B)
```

Fig. 10. Power optimization method which avoids condition distribution after decomposition.

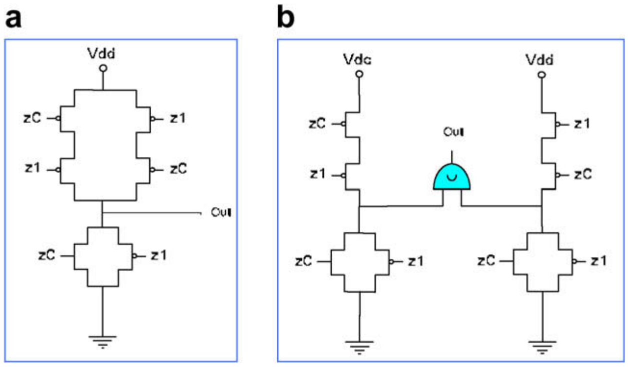


Fig. 11. (a) Balanced NOR gate [5] and (b) proposed enhanced balanced NOR gate for Input-validity checker.

Now we are focused on the PCFB power balancing requirements. Our analysis and simulation results show that the function and operation of handshake (control) part of a PCFB template is completely data independent. The only part which requires some trivial power balancing considerations is the input and output validity checker; this can be met using two additional transistors in NAND/NOR gate (Fig. 11a). However, as the result of hardware redundancy in balanced gate designs, there are many faults that make a balanced gate imbalanced without causing logical errors. This vulnerability opens the possibility of DPA/FI attack [7].

To overcome this vulnerability, a better balancing solution can be obtained by using two identical NOR gates and a C_Element Muller gate [12] in input and output validity checker of a PCFB template (Fig. 11 .b). The C_Element output changes when both its inputs have the same value and their values are opposite of the C_Element current value. By nature, a C-element gate is balanced. If a fault is injected in the proposed balancing circuits as previously mentioned, the circuit creates deadlock (then activates an alarm signal). By using this method, when the output must be charged to one and when the circuit is fault free, both the pull-up network branches will charge the inputs of the C_Element. As a result, the output of the C_Element will be charged to one. But if a fault is injected to one of the pull-up network branches, the C_Element will not be charged to the new value. Consequently, we will be able to discover the attacker's injected fault in the logic level and avoid any DPA/FI attack. Using this method and the proposed computational part, elaborated in the following paragraphs are the main contribution of our work against DPA/FI attack. The computational part is the last module for balancing considerations, and naturally is the main source of power imbalance. By using a SABL [3,6] gate as the computational part of the PCFB template, a QDI balanced gate can be resulted which preserves all the balanced properties while enhancing their fault resistance and robustness. Moreover, asynchronous handshake module eliminates the clocking and timing difficulties that normally are associated with the dynamic gates and enhances the security of applications due to the benefits of asynchronous behavior as mentioned before.

By using a modified SABL gate and employing discharge tree (DT) as the computational part of the PCFB template, a fully QDI balanced gate can be resulted (BSDT style [6] employed this method to balance the PCHB template). Using DT causes the parasitic capacitors placed in intermediate nodes discharge in each evaluation phase (Fig. 12). This will result in a certain amount of capacitor charging and discharging totally independent of input data.

In this balancing method, there is not a complete overlap between the structures necessary for the balanced-power and Boolean functionality of a gate. As a result, there are faults and failures which can easily imbalance the gate without affecting the boolean functionality. Due to the redundancy in the DT, some faults might not create logical errors which in turn would not be detected by traditional voltage level testing and reliability measures. To overcome this vulnerability, we added a circuit to the computational part of the template to detect those faults and generate a stall-signal. The stall-signal becomes active when a fault occurs in one of the transistors in the discharge tree. Since this signal is active permanently low, it turns P1 transistor on and this leads to a high value (Fig. 12). This causes a pipeline stall, which naturally prevents further data processing and creates deadlock within the pipeline (then activates an alarm signal), consequently avoiding DPA/FI attack. An OR function block in this style is straight-forward because it also has asymmetry property of AND function.

To measure the balance of proposed templates, a power balancing measurement should be adapted for the entire evaluation phase, so that any power consumption difference can be observed during chip activity time. We use a transient standard deviation of current traces vector (SDV) [6] to measure power balancing effectiveness. Fig. 13 shows the simulation results. Results show that our proposed cell library is approximately 5.82 times more balanced than the SABL gate. The balance of our proposed cells is approximately reduced to 3% compared to BSDT cell library. This is due to the existence of the stall-signal generator and enhanced balanced input/output validity checker. On the other hand, our proposed cell library is resistant against DPA/FI attacks.

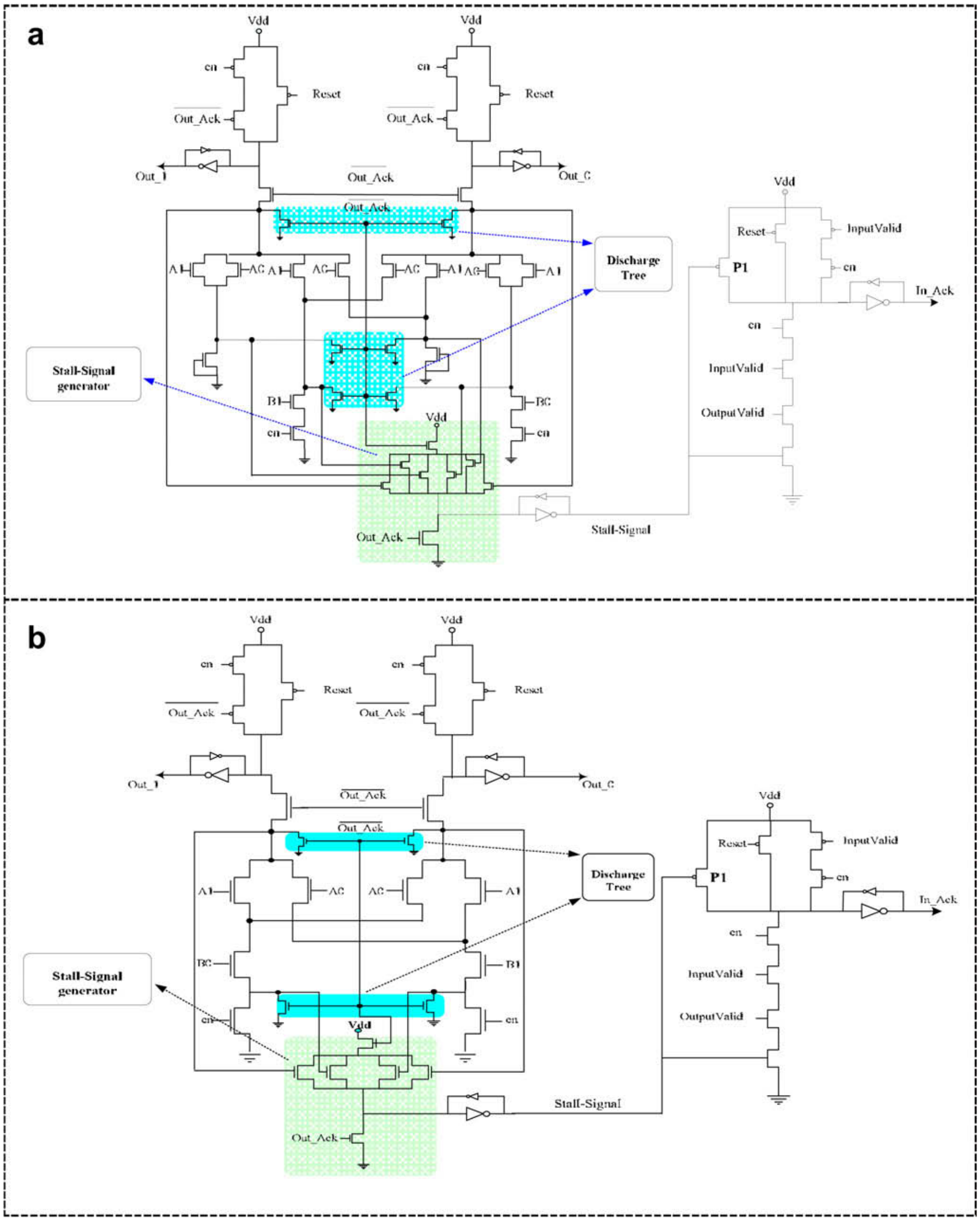


Fig. 12. The proposed balanced (a) "AND" and (b)"XOR" computational and InputAck parts.

Based on the balanced dynamic functional blocks, the current synchronous versions of the balanced library cells still require balanced routing considerations. However, due to gate level asynchronous QDI nature of the method the resulting implementations are very tolerant of process/voltage variations. The natural tolerance of the template can allow more aggressive dynamic balancing techniques which can allow for routing independent gate design. We are currently using a balanced library of cells which does not require balanced routing considerations.

In addition, to have more robust designs for balanced dynamic functional blocks, natural fault resistance is added to the design making use of the sequence in asynchronous handshake protocol. Our transistor level simulation shows almost 80% single stuck-at faults, the inside and outside of the complete balanced gate result in a pipeline stall which naturally prevents further data processing and creates deadlock within the pipeline. That is, the faults prevent or stop the necessary four-phase handshake protocol between each gate, which leads to stalling the communication between dependent downstream gates and prevents any further data processing. To resolve the deadlock, the pipeline requires an explicit reset which will clear all intermediate faulty data values inside the pipeline, removing the possible source of fault attack information. Synchronously

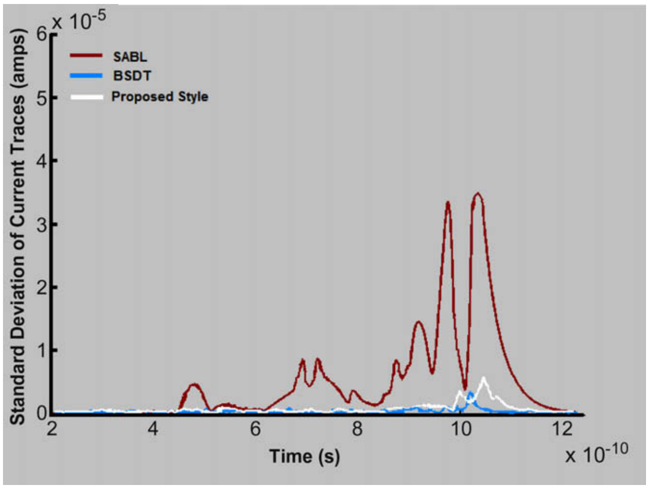


Fig. 13. The standard deviation of the evaluation phase of SABL, BSDT [6] and the proposed method.

balanced dynamic logic gates have no comparable property. Additional error detection based on other high level fault-tolerant methods can be added easily due to the specification of the circuit.

7. Experimental result

To estimate the efficiency of the proposed methodology, we compare the performance and power consumption of automatically synthesized synchronous and asynchronous (with the use of proposed balanced cell library) implementations of DES and AES algorithms.

The same specification of the special DES algorithm with 64 key/inputs [28] has been used for both implementations. Synchronous implementation was synthesized with the Artisan Sage-XTM [14] standard cell library using the TSMC 0.18 μm technology. By using automatically pipelined synchronous implementation (with Synopsys Design Compiler R "pipeline design – period 0" command – maximum performance setting), on the average, every encryption lasts for 693 ns while every encryption in our asynchronous approach lasts for 224 ns. Furthermore, our new method has the lowest power consumption of the asynchronous implementation compared to the existing synchronous ones. Table 1 shows the power consumption of two encryption algorithm implementations, namely DES and AES. The asynchronous DES implementation shows a 1.38 times power consumption reduction over the synchronous implementation using the same technology. The asynchronous implementation after applying conditional restructuring power optimization method demonstrates up to 21% power reduction in comparison to conventional asynchronous design method.

The RTL electronic code book mode (unfolded 10-round) HDL specification of the AES (128 bit where 128 bits is the input and cipher text word length) was used for synchronous implementations, and it was synthesized with the Artisan Sage-XTM [14] standard cell library using TSMC 0.18 μm technology. The automatically synthesized pipelined synchronous implementation (with Synopsys Design Compiler) performed at 45 MHz. Our asynchronous implementations exceed 130.5 MHz. Moreover, this asynchronous implementation shows a 1.17 times power consumption reduction over the synchronous implementation. The asynchronous AES implementation after applying conditional restructuring method shows power reduction of up to 25% compared to the conventional design. Table 1 shows the power consumption of three AES implementations.

Note that in the synchronous case there is no side channel attacks protection. The cost of high-performance and the achieved protection level is the significant area overhead. Thanks to the inherent resistance of asynchronous circuits to operating conditions, they can operate at lower voltage with higher speed and lower power consumption. We performed various

Table 1
The power consumption of AES and DES implementations

Circuit	Power		
	Synchronous (mW)	Secure asynchronous (mW)	Low power/secure asynchronous (mW)
DES 64 bit	8.05	5.73	4.81
AES 128 bit	14.04	11.92	9.46

analyses of the side channel information leakage from this implementation. Simulations of power and timing analysis attacks on our implementations indicate the benefits of the balanced dynamic gates and QDI asynchronous circuits, which are the main goals of our proposed implementations. Furthermore, the DFA was applied to attack our algorithms implementations, and simulation results proved their relative immunity to the attacks. Finally, we would like to note that design the characteristics can be improved with better optimizations and richer micropipeline library.

8. Conclusions

In this paper, a fully automated secure design flow and a secure library cell that is resistant to power analysis and fault injection attacks were introduced for the implementation of QDI asynchronous circuits. In the proposed flow, a high-level description of the system is received in Verilog–CSP format, and then the corresponding specification is decomposed into smaller circuits directly mappable to predefined circuit templates. Furthermore, a power optimization method, which employs the conditional restructuring at the high-level description of asynchronous circuits, was presented. Finally, a self-testable template to resolve the faults that make the balance template imbalanced was presented that makes the DPA/FI attacks to the circuits ineffective. We showed that our cell library is approximately 5.82 times more balanced when compared to the best differential dynamic cells designed using synchronous methods. Also, our approach proved the low power consumption and high performance of the fine-grained pipelined asynchronous circuits.

References

[1] Kocher P, Jaffe J, Jun B. Differential power analysis. In: Proceedings of the 19th intl advances in cryptology conference – CRYPTO '99; 1999. p. 388–97.

[2] Quisquater JJ, Samyde D. Electromagnetic analysis (EMA): measures and counter-measures forsmart cards. In: Proceedings of the 19th intl. advances in cryptology conference – CRYPTO '99.

[3] Tiri K, Akmal M, Verbauwhede I. A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards. In: 28th European solid-state circuits conference (ESSCIRC 2002); 2002. p. 403–6.

[4] Mace, Standaert FX, Quisquater JJ, Legat JD. A design methodology for secured ICs using dynamic current mode logic. Lect Notes Comput Sci 2005;3728:550–60.

[5] Tiri K, Verbauwhede I. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. design. In: Automation and test in Europe conference (DATE 2004); 2004. p. 246–51.

[6] Kulikowski K, Smirnov A, Taubin A. Automated design of cryptographic devices resistant to multiple side-channel attacks. In: Cryptographic hardware and embedded systems (CHES); 2006.

[7] Kulikowski K, Karpovsky M,Taubin A. DPA on faulty cryptographic hardware and countermeasures. In: Fault diagnosis and tolerance in cryptography, third international workshop; 2006.

[8] Kulikowski K, Karpovsky M, Taubin A. Robust codes for fault attack resistant cryptographic hardware. In: Fault diagnosis and tolerance in cryptography, second international workshop; 2005.

[9] Fraidy Bouesse, Laurent Fesquet, Marc Renaudin. QDI circuit to improve smartcard security. In: Second asynchronous circuit design workshop (ACID2002), Munich, Germany; 28–29 Januray 2002.

[10] Biham E, Shamir A. Differential fault analysis of secret key cryptosystems. In: RYPTO 97, LNCS, vol. 1294. p. 513–25.

[11] Arash Seifhashemi, Hossein Pedram. Verilog HDL, Powered by PLI: a suitable framework for describing and modeling asynchronous circuits at all levels of abstraction. In: Proceedings of 40th DAC, Anneheim, CA, USA; 2003.

[12] Sparso Jens, Furber Steve. Principles of asynchronous circuit design – a system perspective. Kluwer Academic Publishers.; 2002.

[13] Alain J Martin. Synthesis of asynchronous VLSI circuits. Caltech, CS-TR-93-28; 1991.

[14] TSMC 0.18 μm process 1.8-volt Sage-X standard cell library databook; September 2003.

[15] Moore S, Anderson R, Cunningham P, Mullins R, Taylor G. Improving smart card security using self-timed circuits. In: Proceedings of the 8th IEEE international symposium on asynchronous circuits and systems – ASYNC'02, IEEE; 2002. p. 23–58.

[16] Chen CN, Yen SM. Differential fault analysis on AES key schedule and some countermeasures. In: ACISP 2003, LNCS, vol. 2727; 2003. p. 18–129.

[17] Chinnery David, Keutzer Kurt. Closing the gap between ASIC and custom. tools and techniques for high-performance ASIC design. Kluwer Academic Publishers; 2002.

[18] Towards sound approaches to counteract power-analysis attacks. In: Wiener M, editor. Advances in cryptology – CRYPTO'99. Lectures notes in computer science (LNCS), vol. 1666. Springer-Verlag; 1999.

[19] Lines AM. Pipelined asynchronous circuits. MSc Thesis, California Institute of Technology; June 1995, revised 1998.

[20] Persia Site: http://www.async.ir/persia/persia.php.

[21] McCardle J, Chester D. Measuring an asynchronous processor's power and noise. In: SNUG; 2001.

[22] Wong CG, Alian J. Martin. Data-driven process decomposition for the synthesis of asynchronous circuits. In: Proceedings of the ICECS; 2001.

[23] Taubin A, Fant K, McCardle J. Design of delay-insensitive three dimension pipeline array multiplier for image processing. In: ICCD; 2002.

[24] Ghavami B, Niknahad M, Najibi M, Pedram H. A fast and accurate power estimation methodology for QDI asynchronous circuits. In: PATMOS; 2007. p. 463–73.

[25] Karri R, Wu K, Mishra P, Kim Y. Concurrent error detection of fault-based side-channel cryptanalysis of 128-bit symmetric block ciphers. In: DAC 2001, ACM 1-58113-297-2/01/0006; 2001.

[26] Shafiaabadi MH, Najibi M, Pedram H, Naderi M, Saleh K. New methods to reduce energy and area in asynchronous circuits following the decomposition stage. In: Proceedings of the 10th annual computer society of Iran computer conference (CSICC2005); 2005.

[27] Najibi M, Saleh K, Pedram H. Using standard ASIC back-end for QDI asynchronous circuits: dealing with isochronic fork constraint. In: ISVLSI; 2007.

[28] Sadeghian B, Aghaee M. Design a special DES processor. Master of Scenic Thesis. Computer Engineering Department, Amirkabir University; 1999.

[29] Fournier Jacques JA, Moore Simon, Li Huiyun, Mullins Robert, Taylor George. Security evaluation of asynchronous circuits.

[30] Tiri K, Verbauwhede I. Design method for constant power consumption of differential logic circuits. In: Proceedings of design, automation and test in Europe conference (DATE); 2005; p. 628–33.

[31] Bouesse GF, Renaudin M, Dumont S, Germain F. DPA on quasi delay insensitive asynchronous circuits: formalization and improvement. In: Design, Automation and Test in Europe (DATE'05), vol. 1; 2005. p. 424–9.

[32] Saifhashemi A, Naderi M, Pedram H, Farhoodfar A. Using standard HDLs and CAD tools for the design and simulation of asynchronous circuit. CSI J Comput Sci Eng 2003;1(4(b)):1–10.

Behnam Ghavami was born in Esfarayen in North Khorasan of Iran, on April 9, 1982. He received his BS degree in Computer Engineering from Bahonar University in 2005. He graduated from the Tehran Polytechnic University. He is a member of Asynchronous Design Laboratory in the same school.

Hossein Pedram received his BS degree from Sharif University in 1977 and an MS degree from Ohio State University in 1980, both in Electrical Engineering. He received his PhD degree from Washington State University in 1992 in Computer Engineering. He has served as a faculty member in the Computer Engineering Department of Amirkabir University of Technology since 1992. He teaches courses in Computer architecture and distributed systems. His research interests include innovative methods in computer architecture such as asynchronous circuits, management of computer networks, distributed systems, and robotics.

Mehrdad Najibi received the B.Sc. degree in electronics from the Department of Electronics and Computer Science at Shahid Beheshti University, Tehran, Iran, in 2001, an Ms.C. degree in computer architecture from the Department of Computer Engineering and Information Technology of AmirKabir University, Tehran, Iran, in 2003, and is still working on his Ph.D. thesis on performance driven synthesis of asynchronous circuits in the same school.