University of Derby

Power Efficient and Power Attacks Resistant System Design and Analysis using Aggressive Scaling with Timing Speculation

Prasanthi Rathnala

Doctor of Philosophy 2017

Abstract

Growing usage of smart and portable electronic devices demands embedded system designers to provide solutions with better performance and reduced power consumption. Due to the new development of IoT and embedded systems usage, not only power and performance of these devices but also security of them is becoming an important design constraint. In this work, a novel aggressive scaling based on timing speculation is proposed to overcome the drawbacks of traditional DVFS and provide security from power analysis attacks at the same time.

Dynamic voltage and frequency scaling (DVFS) is proven to be the most suitable technique for power efficiency in processor designs. Due to its promising benefits, the technique is still getting researchers attention to trade off power and performance of modern processor designs. The issues of traditional DVFS are: 1) Due to its pre-calculated operating points, the system is not able to suit to modern process variations. 2) Since Process Voltage and Temperature (PVT) variations are not considered, large timing margins are added to guarantee a safe operation in the presence of variations.

The research work presented here addresses these issues by employing aggressive scaling mechanisms to achieve more power savings with increased performance. This approach uses in-situ timing error monitoring and recovering mechanisms to reduce extra timing margins and to account for process variations. A novel timing error detection and correction mechanism, to achieve more power savings or high performance, is presented. This novel technique has also been shown to improve security of processors against differential power analysis attacks technique. Differential power analysis attacks can extract secret information from embedded systems without

knowing much details about the internal architecture of the device. Simulated and experimental data show that the novel technique can provide performance improvement of 24% or power savings of 44% while occupying less area and power overhead. Overall, the proposed aggressive scaling technique provides an improvement on power consumption and performance while increasing the security of processors from power analysis attacks.

Table of Contents

List of Figures

List of Tables

List of Symbols

$t_{(su,F)}$: *Setup time of the flip-flop*

$t_{(su,L)}$: *Setup time of the latch*

t_{clk} : *clock period*

t_{ch} : *checking window*

t_{pd} : *propagation delay*

P_{total} : *Total power consumption*

$P_{dynamic}$: *Dynamic power consumption*

P_{static} : *Static power consumption*

V_{dd} : *Supply voltage*

I_{leak} : *Leakage current*

C_{sw} : *Switching capacitance*

f : *Operating frequency*

V_{set} : *New set voltage*

V_{min} : *Minimum voltage*

V_{max} : *Maximum voltage*

f_{min} : *Minimum frequency*

f_{max} : *Maximum frequency*

V_{ref} : *Reference voltage*

V_{o} : *Output voltage*

V_{error} : *Error voltage signal*

e(n) : *Digitized error*

d(n) : *PWM output*

d(t) *: Time domain signal*

T_1, T_0 : *Predicted error signals*

$V_{present}$: *Present value of the voltage*

$F_{present}$ *: Present value of the frequency.*

m *: Slope of the voltage and frequency function*

$C_{(T,P)}$: *Correlation coefficient of two variables T and P*

Var : *Variance*

Cov : *Covariance*

Q : *Ouput of the sequential element*

D : *Input to the sequential element*

En : *Enable signal*

clk : *Clock signal*

d_in : *input to the sequential element*

ff_clk : *flip-flop clock signal*

latch_clk : *latch clock signal*

D_in : *input to the D flip-flop*

f_out : *flip-flop output signal*

l_out *: latch ouput signal*

error : *Error signal*

Del_clk : *Delayed clock signal*

$\rho_{(x.y)}$: *Correlation coefficient of two variables x and y*

Hw : *Hamming weight*

(v_i', f_i') : *new set of voltage and frequency values*

(v_{old}, f_{old}) : *previous set of voltage and frequency values*

S_n : *Number of samples*

$C_{(i, j)}$: *Cipher texts generated*

np : *Number of plain texts*

nk : *Number of keys*

A_i : *Number of plain texts*

K_j : *Number of keys*

$P_{(i, j)}$: *Hypothetical power model*

f_c : *Clock frequency*

C_w : *Interconnect wire capacitance*

α : *Switching activity*

N : Number of encryption rounds

n : Word length

Acronyms

ADC : Analog-to-Digital Converter

AES : Advanced Encryption Standard

AFS : Aggressive Frequency Scaling

APC : Advanced Power Controller

ASIC : Application-Specific Integrated Circuit

AVS : Aggressive Voltage Scaling

CAD : Computer-Aided Design

CMOS : Complementary Metal Oxide Semiconductor

CPM : Critical Path Monitor

CPR : Critical Path Replica

DCM : Digital Clock Manager

DES : Data Encryption Standard

DPA : Differential Power Analysis

DSO : Digital Storage Oscilloscope

DVFS : Dynamic Voltage and Frequency Scaling

FPGA : Field-Programmable Gate Array

HD : Hamming Distance

HW : Hamming Weight

IoT : Internet of Things

LUT : Look-Up Table

NIST : National Institute of Standards and Technology

PIC : Programmable Interface Controllers

PVT : Process Voltage and Temperature

PWM : Pulse Width Modulation

SCA : Side-Channel Attacks

SPA : Simple Power Analysis

VHDL : VHSIC (Very High-Speed Integrated Circuit) Hardware Design Language

WDDL : Wave Dynamic Differential Logic

Acknowledgements

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis.

First and foremost, I would like to thank my supervisor Dr Ahmad Kharaz, for his invaluable guidance, support and encouragement throughout my PhD. His encouragement to explore new ideas and push existing boundaries has helped me to develop this thesis.

I would also like to thank my second supervisor, Mr Tim Wilmshurst, for the constructive feedback and suggestions during the course of the study.

I would like to thank the director of the Institute for Innovation in Sustainable Engineering, Professor Richard Hall for his help. I am also thankful to the Advanced Sensors and Systems lab staff at IISE and College of Engineering and Technology staff for providing me a great atmosphere for carrying out this research. I am grateful to all my past and present research lab members at Advanced Sensors and Systems lab. It was great working with them and I learnt a lot from our discussions.

Finally, I would like to thank my parents, family and friends for their support and motivation during this work.

CHAPTER 1

Introduction

This chapter presents a general introduction to modern embedded system design constraints such as power consumption and security. In order to develop a system with these constraints, all levels of the design process must be addressed subject to constraints on the system performance and the quality of service [1]. This chapter starts by introducing the basics of power consumption and highlighting the system design hierarchical techniques that are used by semiconductor vendors to meet the market growing demands keeping power consumption as a constraint. It also gives a summary of this research work contributions towards power optimization and security of embedded systems.

1.1 Power Efficiency

Power optimization is one of the fundamental necessities of the modern processor designs. Not only it contributes to increasing the battery life of small and smart embedded systems but it also helps to reducing the cost of cooling effects in high processor designs. Essentially, the market for embedded systems is ever growing and the need for power efficiency of the systems is also increasing. Traditionally, dynamic voltage and frequency scaling was the most suitable technique for balancing the performance and power consumption to meet the growing demands. However, during the last decade, it has mostly been associated with the application performance requirements with tight power constraint budgets. On one hand, due to the scaling of process technologies, it is no more suitable for process variations, while on the other hand, it is the industry approved technique due to its simple, efficient and effective design. To account for the variation, semiconductor manufacturers add extra margins to provide reliable and robust operation. However, the extra margins are limiting the processor high performance requirements or power savings [1].

New high performance or multi-core processors need to be designed and developed to meet the performance demands of the modern computing. However, these new processors pose many challenges for the designers, predominantly handling increased power consumption. Figure 1.1 shows the statistics for the power consumption up to 2020 [2]. This roadmap sets aggressive targets for power efficiency of future embedded systems to accommodate the performance requirements. As seen in the figure, the power requirement trend is increasing exponentially while creating more gap between the required and actual levels. This, increases the demand for

embedded systems power consumption, which has to be addressed at different levels of abstraction of the system design.

Figure 1.1: Power requirements and trends through 2020 [2]

1.2 Fundamentals of Power Consumption

Traditionally, digital systems were designed with performance and physical size in mind and the other factors such as testability and power consumption were mainly regarded as side constraints. It was in the early 90's that power consumption became a main design constraint along with performance and physical size. The main driving factor for this trend is the exploding market for small and smart electronic devices that demand high functional density with the limited power supply of batteries.

The total power consumption of a processor can be categorised as static and dynamic power. The amount of power dissipated when there is no circuit activity is called the static power and it mainly encompasses operating voltage, leakage current and fabrication process. The amount of

power dissipated when there is a switching activity is called the dynamic power and the overall power consumed by a CMOS device can be calculated by adding up these two powers.

$$P_{total} = P_{dynamic} + P_{static} \quad (1\text{-}1)$$

$$P_{static} = V_{dd} I_{leak} \quad (1\text{-}2)$$

$$P_{dynamic} = C_{sw} V^2{}_{dd} f \quad (1\text{-}3)$$

As in Equation (1-3), dynamic power depends on load capacitance, operating voltage and frequency. Scaling down frequency helps reducing power but it affects performance. Depending on the computational task, processor must intelligently choose the appropriate frequency to avoid running the system unnecessarily at a high speed clock for the entire period. Reducing operating voltage has a major impact on dynamic power because of the quadratic relation between voltage and power. Moreover, static power can also be minimised by reducing the operating voltage. Hence the voltage scaling is the most effective and commonly used technique. Voltage scaling is accomplished by identifying the critical and non-critical paths, separating and powering them with high and low voltages, and dynamically changing the supply voltage depending on the requirements. Another possible way to reduce dynamic power is reducing the load capacitance. c_{sw}

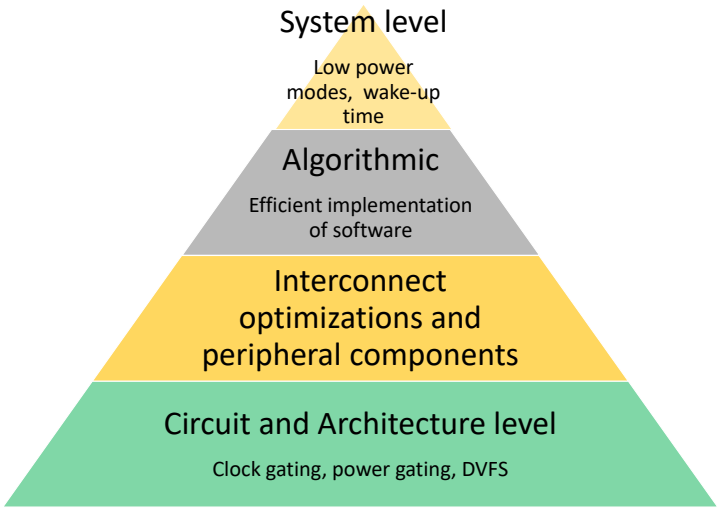


Figure 1.2: The hierarchy of low power embedded system design

The available power reduction techniques are grouped and categorised as shown in Figure 1.2 .The base of the pyramid is the circuit and architectural level techniques which predominantly concentrate on the architecture to optimise power consumption. Popular techniques include Clock gating [3], [4], Power gating [5] and Dynamic Voltage and Frequency scaling [6]. DVFS is known for the ease with which it can be implemented and the effectiveness with which it is possible to achieve power reductions. More details of these techniques are discussed in chapter 2.

The next level in the hierarchy is interconnecting optimisations, is also a well-researched area [7]. One reason for this is, a dominant portion of a processor power consumption is dependent on bus interconnect mechanism. To address this problem, intelligent bus encoding schemes can be used to scale down the toggling activity and efficient bus encoders/decoders can be used to reduce the bus swing and structure [7].

Algorithmic optimisation involves efficient coding and the utilisation of available resources. Also, the effectiveness depends on the programmer knowledge and expertise in the software skills and the application domain.

The highest level in the hierarchy is the system level techniques. The most popular techniques include the usage of low power modes, multiple clock sources, multiple regulators, and wake up time [8].

Power consumption at circuit and architectural level is considered as the main focus of this research work. As illustrated in Figure 1.2, the power consumption needs to be dealt with at each level of the hierarchy mentioned. However, the base of the hierarchy is the most important section to achieve low power as the higher levels of abstraction are built on it. It is identified that switching activity and operating voltage are the two most important factors that influence energy consumption and performance of the modern processors.

1.3 Security

Another focus of this research is the security of embedded systems. With increased use of portable devices and the new developments of IoT [9], the embedded system design metrics have been changed. Security of the systems is also becoming an important design metric to store and access sensitive information. Along with the performance and power constraints, security is also emerging as a design constraint [10].

Mobile phones [11], PDAs [12], smartcards [13] and servers are all the best examples of embedded devices used for secure transactions such as banking, online shopping and payments. This is not limited to these applications but also the new development IoT demands the security of embedded systems as an important metric.

Figure 1.3 Internet of Things: number of connected devices worldwide from 2012 to 2020 (in billions) [14]

A recent survey reveals that the number of devices connected to the internet to transfer or exchange data has been increasing. Figure 1.3 depicts a forecast on the number of connected devices using internet of things (IoT) worldwide from 2012 to 2020 [14]. In 2017, the number of connected devices is 28.4 billion, which is predicted to be doubled by 2020. Similarly, smart device usage for secure transactions has also been increasing exponentially. This situation demands more security than ever before to avoid the attacks on these devices.

In general, side-channel attacks [15] are the most commonly used and pose a major threat to embedded systems security. Since the attacks work on a fundamental principle that the computation is a physical process and all physical characteristics of a computation, leak the secure information used internally. The typical physical characteristics such as time, power consumption, and electromagnetic radiation reveal information about the internal computation. Furthermore, the attacks can be implemented with ease and commonly avilable equipment such as a computer and an oscilloscope. Power analysis is the most powerful and effective technique

to reveal the secret information used during the execution of cryptographic algorithms [17][16][18].

According to Kocher [19], power analysis attacks are two types: simple power analysis attacks (SPA) [20] and Differential power analysis attacks (DPA) [21]. A Simple Power Analysis Attack (SPA) predicts the secret information by analysing the power traces visually. In power analysis, a trace is a collection of power measurements over a specified period. Power consumption of a device typically depends on the instruction being executed and data being processed. In a typical processor context, power varies with the addressing mode and the type instruction such as arithmetic, logic and Boolean. However, the prediction can be made wrong by using simple countermeasures such as random insertion of dummy code and avoiding memory usage [22].

Differential power analysis attacks are a major concern to the embedded systems security. The attacks can be successful even without knowing much information about the system. The attack process is initiated by measuring the power consumption of the device followed by comparing it with the hypothetical power consumption and eventually extracting the secret information contained in a chip or system.

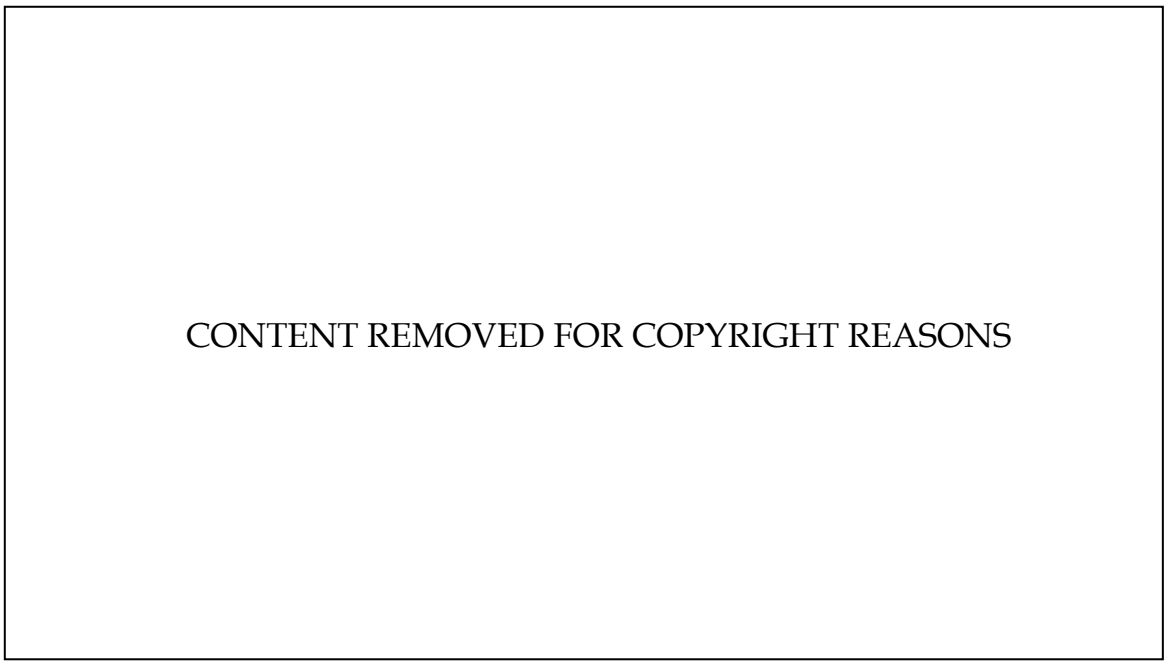


Figure 1.4: The hierarchy of embedded system security [9]

Figure 1.4 shows the hierarchy of embedded system security. The base of the hierarchy needs to be strong to build an effective and secured embedded system. Therefore, this issue is dealt at the circuit and architecture level. Dynamic voltage and frequency scaling technique can be used as a countermeasure to improve the robustness against power analysis attacks. This work addresses the problems of using DVFS as a countermeasure and proposes a new technique, which effectively reduces the correlation between power consumption and the internal data being processed within the device.

1.4 Summary of Contributions

A novel technique based on aggressive voltage scaling has been proposed to address the issues of the main design constraints, power consumption and security. The proposed technique has an error detection capability without error correction timing penalty. As soon as the error is detected, the phase delayed clock is selected to capture the late arrival of data. The area overhead involved is small compared to the other latest techniques. In

addition to this, the present technique is quite attractive in all the design aspects. The summary of main contributions is:

- Voltage scaling techniques have already been proposed as a countermeasure for power analysis attacks. The used aggressive voltage scaling involves large area overhead and error correction overhead. To solve the problem of hardware realization of aggressive voltage scaling, a new countermeasure to improve resistance of the circuits from power analysis attacks is proposed.
- It is well known that DVFS technique is the most effective power reduction technique. To achieve more power savings, processors use aggressive scaling technique with error detection and correction circuitry to operate systems beyond the worst case estimates. The error detection and correction circuitry is used to detect any timing speculations in the circuit as frequency and voltage are scaled.
- Practical experiments have been carried out by prototyping the design on to an FPGA to validate the effectiveness of the technique in terms of power consumption and resistance to DPA attacks.

1.5 Thesis Organization

The organization of this thesis is as follows:

Chapter 2 discusses the basic components of power consumption and the techniques at different levels of abstraction of systems design, particularly circuit and architectural level techniques.

Chapter 3 covers the background of power analysis attacks and different countermeasures that can be used to improve resistance of the systems to differential power analysis attacks.

Chapter 4 presents the latest research of aggressive voltage/ frequency scaling techniques and the Matlab simulation of AVS system with a controller.

Chapter 5 discusses the proposed technique covering the principle of operation, simulation results and the practical hardware implementation of the technique using FPGA.

Chapter 6 is about the proposed technique as a countermeasure for differential power analysis attacks.

Chapter 7 presents concluding remarks and future research work.

1.6 Summary

Different power saving and security related techniques are listed in this chapter ranging from architectural to application development. In addition to this, a summary of this research work contributions are discussed. Scientists have been coming up with new techniques and algorithms at different levels of system design but it seems inadequate. The field requires much more work to be done towards power reduction and security improvement and indeed it does require new methods and new ways of thinking to meet the industry demands. In the next chapter, the various low power system design techniques are discussed with an extra emphasis on the simple yet more effective technique, dynamic voltage and frequency scaling.

CHAPTER 2

Literature Review for Power Efficiency

Low power system design requires reducing power consumption in all components and at all levels of the design process subject to constraints on the system performance and the quality of service [1]. In general, processor is the main power hungry device in an embedded system. Hence, this chapter provides a survey of processor based techniques that are used by semiconductor vendors to meet the market growing demands keeping power consumption as a constraint. Further, it presents a brief overview of the state of the art principles of circuit and architectural level techniques and emphasizes mainly on the most effective and efficient technique for power reduction, Dynamic Voltage and Frequency Scaling (DVFS) technique. It highlights the various forms of DVFS architecture such as open loop DVFS and closed loop DVFS and their benefits. Some emerging

techniques on DVFS such as techniques based on software, hardware and controller are also discussed.

To build a low power embedded system, the design issue must be dealt with at each level of abstraction of the system design cycle. The pyramid in chapter 1, Figure 1.2 shows different levels of abstraction. Though, the design issue need to be dealt with each level, but to have a strong pyramid, the base of it must very strong enough to handle the above layers. Hence, this research work has been concentrated on circuit and architectural level.

2.1 Architectural and Circuit Level Design Level Techniques

In this section, an overview of low power architectural and design level techniques are detailed and all these techniques are based on the principle discussed earlier, manipulation of operating voltage, frequency and leakage current.

2.1.1 Clock Gating

Clock signal consumes significant amount of power in digital systems because of the continuous toggling nature [23]. The sources of power consumption in a clock network are switching activity and switching capacitance. The objective of clock gating is disabling the clock whenever there is no activity. Hence switching activity can reproduced and thus power consumption can be optimized. A simple clock gating circuit is shown in Figure 2.1. Sequential clock-gating can provide major power savings by reducing toggling activity by 15-25% in logic block [3]. Combinational clock-gating provides power reduction by about 5-10% [3].

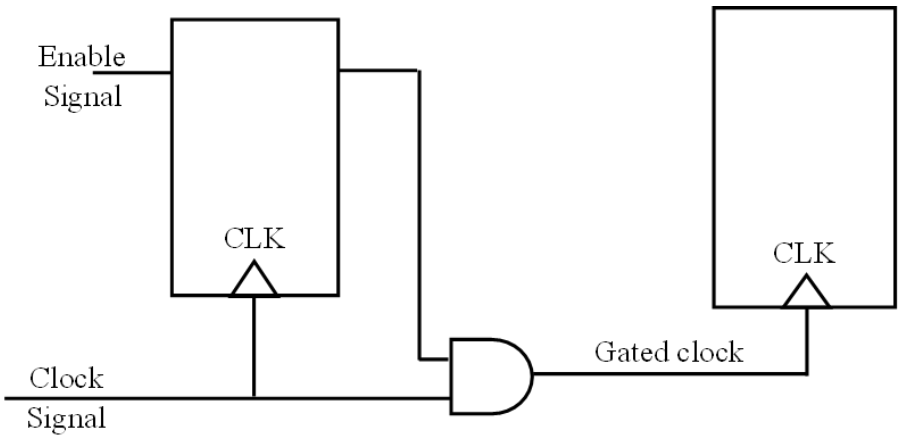


Figure 2.1: A simple clock gating circuit

2.1.2 Power Gating

Similar to clock gating, power gating is also nothing but shutting down the unit when it is not in use. In power gating, the power supply need to be shut off or power down of the functional block when not in use. Sub-threshold leakage and gate leakage both are reduced using this method [5]. Figure 2.2 shows a simple power gating circuit diagram. The functional block is connected to the supply using two switches, header switch and footer switch. When there is no activity to be performed, the sleep signal disconnects the functional block from the power supply and reduces the leakage current significantly. This technique is extensively useful to implement various low power modes in processors.

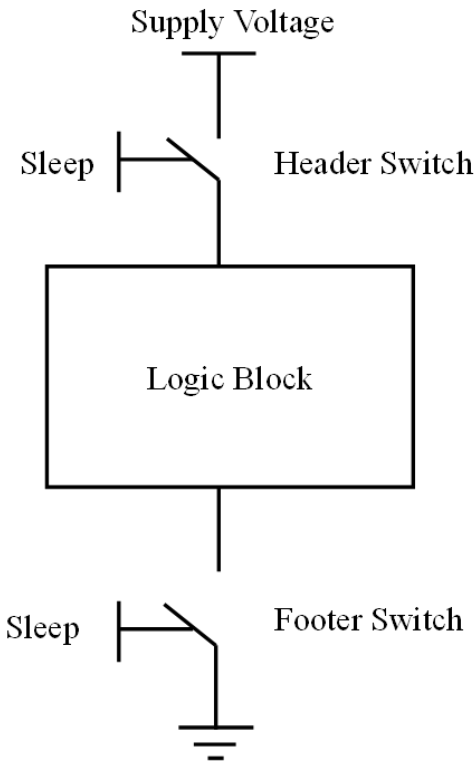


Figure 2.2: Power Gating

2.1.3 Pipelining

Pipelining is the technique used to improve instruction throughput. This method leads to a reduction in the critical path and thus effectively reduces switching capacitance to achieve low power consumption. In some cases the instructions need to be discarded, and it may lead to waste of energy if they are already fetched and partially processed.

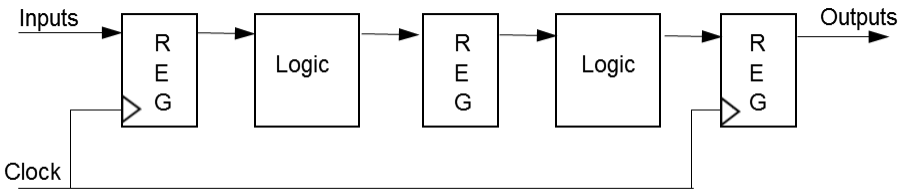


Figure 2.3: Pipelining circuit

As can be seen in Figure 2.3, the main design element of a pipeline is deciding the number of registers to be used to realise the required stage. This information is dependent on the location of the pipeline stage.

Increasing the number of registers provide high throughput but consume more power. To achieve low power the number of registers must be reduced. Shimada et al. [24], proposed another technique called Pipeline Stage Unification (PSU). PSU uses the same principle as dynamic voltage and frequency scaling, modifies frequency and operating voltage on the fly. However, power reduction is attained by reducing the number of stages and consequently reducing the number of registers instead of reducing operating voltage.

2.1.4 On-Chip Communication

On-chip communication significantly contributes to low power consumption. A dominant portion of a chip's power consumption is dependent on bus interconnect mechanism. Low power bus communication can be realised by scaling down the toggling activity through intelligent bus encoding schemes and designing bus encoders/decoders to reduce the bus swing and bus structures. The power consumption in an interconnect mechanism is [25]:

$$P_w = {}^1/_2 \, \alpha f_c C_w \Delta V^2 \qquad (2\text{-}1)$$

Where ΔV the voltage swing, C_w is the interconnect wire capacitance, f_c is the clock frequency and α is the switching activity. From Equation (2-1), one way of reducing power is scaling down the switching activity on interconnects, and hence reducing the capacitance. This can be done by using bus encoding schemes. The encoding methods are applied depending on the kind of bus, architecture and the complexity of the encoder.

As can be seen in the equation (2-1), interconnect power is directly proportional to the signal voltage swing. Hence reducing voltage swing

causes power reduction. One way of achieving this is using a low voltage differential signalling circuit, which acts as a level converter between the transmitter and receiver. The input signal voltage is reduced by splitting into two opposite polarities at the transmitter. The receiver sees the difference between the two transmitted signals and amplifies it back to normal voltage. A simple schematic representing this scheme is shown in Figure 2.4.

Figure 2.4: Low voltage differential signalling circuit [25]

Traditionally, the bus is designed in such a way that the complete bus charges and discharges for every request. This method divides the bus into various blocks and these blocks are joined by a link that controls the traffic between adjacent blocks. To keep the bus in power down mode, the critical paths that connect links are initiated separately. The other best way is keeping the most often communicating blocks close to avoid extra infrastructure and to achieve power efficiency.

The techniques discussed so far consider an architecture in which different blocks use one or more buses commonly. Sharing buses this way has impact on performance and power. The intrinsic bus bandwidth confines the capacity and speed of data transfers and not suitable for the varying requirements of various functional blocks. The main constraint of bus communication is accessing the bus, several can request for bus access but only one can access at any time. Hence the bus access always needs bus arbitration methods but they are power expensive. Unlike simple data

transfers, every bus transaction comprises multiple clock cycles of handshaking protocols that increase power and delay [26]. The only solution for this problem is a new technology that should substitute buses. Compared to buses, networks offer higher bandwidth and support concurrent connections [26]. The network based architecture helps to concentrate on power sensitive areas. Many networks have sophisticated mechanisms for adapting to varying traffic patterns and quality-of-service requirements [26], [27].

2.1.5 Voltage Scaling Techniques

Scaling down the supply voltage provides significant power reduction because of the square factor in equation (1-3). Static voltage scaling and dynamic voltage scaling are the two techniques that are widely used to reduce the total power consumption of a device.

2.1.5.1 Static voltage scaling

Figure 2.5: Multiple supply voltage system [28]

Multi-supply voltage approach is nothing but the static voltage scaling technique that operates different blocks at different voltages based on their

performance needs. The processor internal blocks such as ALU, memory and I/O are partitioned and provided supply voltages as shown in Figure 2.5. Different voltage islands are identified at the design time and they are fixed for the entire chip operation. They have been used in most of the high-performance commercial chips, including the Intel processors with XScale technology and the Crusoe processor from Transmeta. For example, the Intel Pentium Processors use three power islands to connect to the processor core, memory and I/O units. Multi-voltage domains require enormous amounts of internal development effort in libraries, tools and methodologies. However, this method significantly reduces dynamic power as well as static power since the operating voltage is getting lower. A level shifter is required in this design because there is a level difference between the two blocks.

2.1.6 Dynamic voltage scaling

Dynamic voltage scaling is the most commonly used technique, it sets the supply voltage according to the performance need. The technique is combined with dynamic frequency scaling to achieve significant power reduction. The main goal of the combined technique is to operate the processor at the lowest possible operating conditions while achieving the required performance levels. In other words, the fundamental idea of DVFS is tuning the supply voltage and frequency depending on the current task requirements. Reduction of supply voltage has a major impact on power consumption because power is linearly related to the square of the supply voltage.

Figure 2.6: Block diagram of open loop DVFS [16]

Based on the architecture used, dynamic voltage and frequency scaling approach is divided into two categories: open-loop and closed-loop. Open-loop DVFS is shown in Figure 2.6 and it is the most commonly used form of DVFS. Here, the operating voltage or nominal voltage is pre-determined for the target application and for the desired operating frequency based on a table of frequency/voltage pairs. The values in the table are pre-determined by considering all operating conditions and process corners.

Figure 2.7: DVFS operational sequence [30]

Figure 2.7 shows the sequence of operations performed to switchover between different operating conditions of DVFS. Depending on the application requirements, the processor decides the appropriate frequency and requests the voltage controller a new voltage by looking at the pre characterised frequency/voltage pair look-up tables. Next, the processor commands the power supply to switch to the selected voltage and waits for the new voltage to settle. The processor changes to the new frequency after voltage is stabilised. The Change can be done immediately when changing from a higher frequency to a lower frequency but not when switching from a lower frequency to a higher frequency. The reason for this requirement is the power supply voltage must be high enough to support the new frequency prior to changing the clock [30].

Figure 2.8: Block diagram of closed loop voltage scaling system [16]

A typical block diagram of closed-loop voltage scaling is shown in Figure 2.8 and this closed loop in nature provides better results over open loop DVS. The circuit monitors device temperature, process profile and report the information to an embedded controller called advance power controller (APC). Based on the profile information, the APC decides whether voltage optimisation is required or not and then sends a command to the external power management unit to change the voltage. The system continuously monitors system changes and adjusts the operating voltage accordingly. Better performance can be guaranteed with closed loop voltage scaling system as compared with the look up tables in open loop DVS. The closed loop system can respond much faster since it is limited only by the companion power supply.

Figure 2.9: Energy consumption of different voltage scaling schemes [29]

As can be seen in Figure 2.9, closed loop system also known as adaptive voltage scaling provides more energy savings compared to dynamic voltage scaling and fixed voltage (no scaling techniques) schemes.

D. Ernst et al [31] proposed the first ever AVS system known as Razor. The AVS system adjusts the supply voltage adaptively and detects if the voltage reduction exceeds the limits. This error detection is notified by the special flip flops equipped with the error detection logic. Once detected, the error must be corrected by computing the failed operation at a lower clock rate.

2.2 Problems of DVFS Technique

There are many techniques and approaches at this level of system design hierarchy. Dynamic voltage and frequency scaling is the most promising technique to achieve optimal performance while providing energy savings. This technique is the industry approved and the most dominant research area. However, the technique needs to be extended or upgraded further to suit the modern processor performance and power demands. In order to meet the present requirements, the technique needs to be modified without losing its intended features. Using this technique, the commercial

processors offer several operating modes. In low power mode, the processor performs simple operations and consumes less power. Similarly, in high performance mode, the processor computes the task at high speeds to meet the deadline. The DVFS software switches between different modes depending on the task requirements. All the operating modes are pre calculated by considering the application and PVT variations.

As discussed earlier, the hardware realization of traditional approach uses open loop system. The system pre-determines different voltage and frequency pairs and stores them in a look-up table. Since the values are pre-determined, the system is not able to adapt to the changing needs of the modern processor needs. Another approach is adaptive or closed loop system. The controller monitors the application requirements and decides which level is appropriate. However, the technique is not able to fully realise the benefit to provide ultra-power savings due to the excess timing margins. To summarize, the issues of the traditional DVFS are: 1) due to its pre-calculated operating points, the system is not able to suit modern process variations; 2) since PVT variations are not considered, large timing margins are added to guarantee a safe operation in the presence of variations.

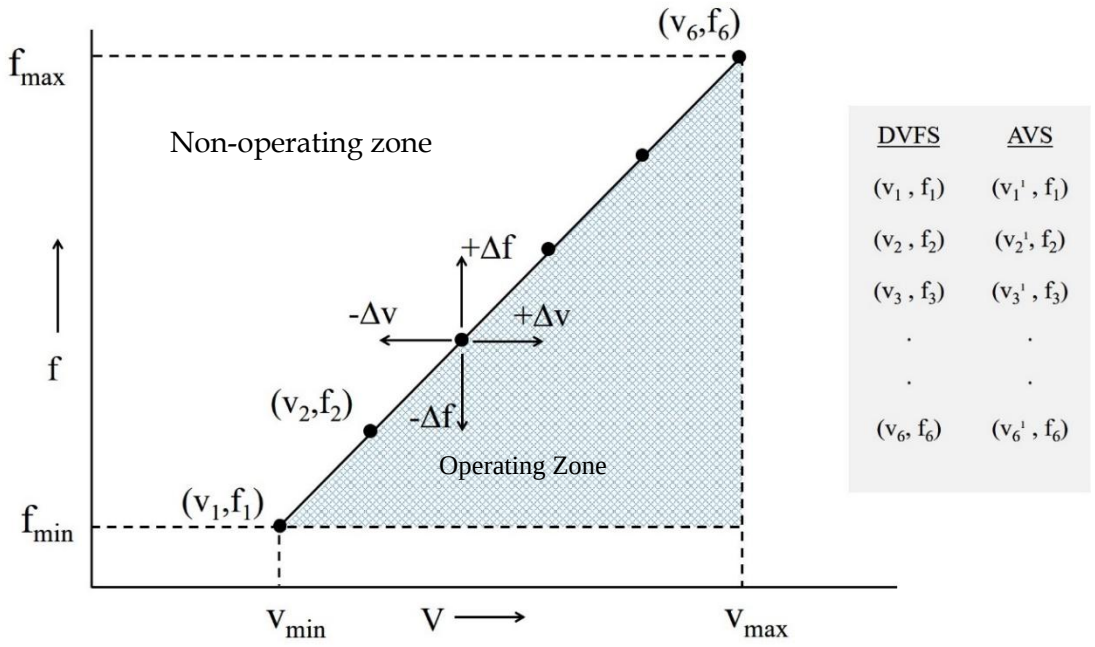


Figure 2.10: Voltage- frequency graph for DVFS and AVS or AFS

Latest research addresses the issues of DVFS using Aggressive scaling technique (AVS), which is the most recent emerging extension to DVFS. It addresses the needs of power optimization very effectively. This technique pushes the walls of the operating region, to the limit, to achieve the required power savings and the main concept of this technique is to go beyond the worst case estimates of voltage and frequency. To have a better understanding, different pairs of (v, f) under DVFS and AVS approach are illustrated in Figure 2.10. As a result of DVFS, all the pairs lie on the linear curve and falls in the operating zone of a typical processor. The pairs can deviate from the linear curve using AVS or AFS. For example, at each operating point, in the case of aggressive voltage scaling, the regular set of values (v_1, f_1) changes to a new set of values ($v_1^1 = v_1 \pm \Delta v$, f) and if voltage changes to (v+Δv), it enters into the operating zone aiming high performance. If voltage changes to (v-Δv), it enters into the non-operating zone aiming to reduce power consumption.

As a result of AVS entering into non-operating zone, it can cause problems *with device handing and functional correctness of the processor. Generally to address these issues, semiconductor manufacturers fabricate devices (ICs) by considering worst case scenario and provide some tolerance. As the occurrence of the worst case is very rare, the devices can relax the extra margins to attain maximum performance or power savings.

To ensure the correct functionality of the device, critical path must be monitored in aggressive scaling techniques. In general, the critical path of the design decides the minimum time period or maximum clock frequency. In order to provide functional correctness, the behaviour of the critical path need to be monitored as voltage or frequency is scaled. The variation of voltage and frequency causes either clock frequency faster or slower with respect to the input data arrival. The misalignment of data with respect to the clock edge causes timing error and causes malfunction. To address this issue, the timing error detection and correction circuits are employed to successfully detect and correct the timing errors.

2.3 Latest Research on DVFS Technique

In general, the latest research trends can be identified as having three main paths;

- Techniques for optimising the overheads involved to implement hardware DVFS
- Techniques for improving the decision making on task scheduling to provide optimised power or performance
- Techniques for providing fast response time on power supplies and controllers to switch over between different modes.

For all paths, the common goal is to trade off performance and power consumption by manipulating voltage and frequency. In other words, all the techniques rely on the fundamental principle of CMOS devices that the dynamic power consumption is directly related to the square of the supply voltage and operating frequency. Using DVFS technology, the industry standards such as Intel SpeedStep [32] and AMD PowerNow [33], achieve the trade-off between power and performance.

Various disparate parameters related to maximum performance, area overhead, power overhead and response time of DVFS technique have been analysed and optimised. The research on hardware includes the possible number of states and on all the individual components of the DVFS technique. The research on software describes different techniques to integrate the end user requirements to the available hardware. This includes task scheduling, power management algorithms, etc. The research on control techniques includes fast response time of power supply units, adaptive power supply and an effective controller.

2.3.1 Techniques based on controller

As a mechanism to support voltage scaling in all modern processors or systems, several power supply designs have been proposed to provide the required levels of voltages. As briefly introduced, many power supplies are available; fixed voltage power supply, on-chip multiple or adaptive power supplies. Essentially, the power supply must be able to provide different functional blocks with different voltage levels simultaneously or different voltage levels required to be operated the system in different operating modes.

In [34], [35], different adaptive power supply designs are presented to provide fast transient response to reduce the losses and latency incurred to switch between different operating modes. A slightly different approach, with complete digital-controller architecture in a closed-loop with a DC-DC switching converter is presented in [36]. Another important challenge for the controllers is to predict the workload variation to optimally choose the corresponding parameters. The efficiency of the controllers depends on the accuracy of the workload prediction or estimation. An efficient controller based on fuzzy logic is proposed in [37] to accurately estimate the workload to tune the supply voltage and frequency to match the task requirements. An on-chip critical path emulator architecture is presented in [38], which tracks the changing critical path in the design. The authors claim that the architecture is up to 43% and 23% more energy efficient compared to open loop and closed loop systems respectively. Another DVFS control mechanism is proposed in [39], where a voltage over-shoot/undershoot is introduced in order to accelerate the transition from one voltage level to another. The authors claim a gain of 30% in the transition time of the applied voltage. A different approach is presented in [40] as standalone chip with a DC-DC regulator and voltage controlled oscillator (VCO). The whole mechanism works on the reference clock frequency, an error signal is generated that controls the regulator through a digital loop filter.

Recently [41],[42], proposed a joint control of voltage and frequency actuators which are designed independently. The approach controls the transition from one operating point to another, optimizing the intermediate path between the states.

2.3.2 Techniques based on software

Scheduling is the most important step in DVFS to trade-off power and performance in multi-processor systems. The combination of DVFS and scheduling of the tasks provide an effective resource utilization and power optimization. During the process, scheduling involves assigning the tasks in the task set on a multi-processor or high performance processor system by attempting to maximize objective function while meeting the deadline constraint [43]. The next phase in the process involves the output of the earlier phase is processed in such a way that the task performs with in the tight deadlines while achieving energy reduction using DVFS technique. In [44], proposed an effective scheduling technique, which maps each task to one of the voltage and frequency pair (v,f) from the available set of discrete pairs using DVFS. Another similar technique with slight modification to the DVFS architecture is proposed in [43]. This technique works on multiple voltage–frequency selection (MVFS) instead of single value to achieve more energy reductions. The author also claims that with their MVFS-DVFS, energy consumption is pushed near the optimal point of using (v, f) pairs supported by tradition DVFS schemes.

2.3.3 Techniques based on hardware

The techniques aim at hardware level mainly comprises of pushing the optimal operating points further to have more number of (v, f) pairs and to reduce power consumption. Hence the techniques employ a special hardware circuit to detect any timing violations. Various techniques have been proposed under this approach [43], [46]. These techniques are discussed in detail further in the coming chapters to provide the background for this research with details of the research aims and objectives. On the other hand, as the technology is shrinking, the supply

voltage scaling requires a reduction in the threshold voltage, as a result of this, the leakage current increases exponentially with each new technology generation. As in [21], the combination of adaptive body bias and DVFS can be used to reduce the leakage current and thus the total power consumption of modern processor systems.

2.4 Summary

The literature survey is conducted in three major areas in the scope of this research. They are a) circuit and architectural techniques of using manipulation of supply voltage, operating frequency and leakage current for power optimization; b) the low overhead and industry approved technique: dynamic voltage and frequency scaling methods suitable for modern processor designs c) latest research on DVFS software, hardware and controller. The survey has shown that the research theme of power optimization of embedded systems has taken different areas and optimization approaches. Some methods were proposed on adaptive power supply to provide fast transient response during the switchover of the operating modes. Only a few studies used the aggressive voltage scaling in their power optimization techniques as trigger, but not necessarily with the intention of improving security of the systems from DPA attacks, which is discussed later in subsequent chapters. Due to primitive and non-optimal operating points, traditional DVFS could not reap the full benefits of the technique in terms of power optimization. This called for more studies on DVFS technique to further exploit the excessive timing margins used to determine the optimal operating points. In order to develop a power efficient and secure system and to perform a comparative analysis on timing error detection and control, a complete aggressive scaling based system simulation model and the evaluation of the proposed

techniques on hardware platform is also required. The rest of the thesis attempts at providing some solutions to the problems identified here.

CHAPTER 3

Commercial Low Power Systems and Techniques

Considering complex and low power commercial processors give better idea of how power and performance has been taken care by the industry and what techniques have been employed for the purpose. This chapter highlights the features of the most powerful and widely used processors for mobile applications; Texas Instruments Application processors - OMAP35x and Samsung low power 32 bit RISC microprocessor - S3C2410X. Further, an overview of system level power reduction techniques is presented by considering commercial microcontrollers. Also, some emerging technologies along with the power management technologies used by industry are provided.

3.1 Texas Instruments Application Processors - OMAP35x

The OMAP35X device includes state of the art power-management techniques required for high-performance mobile products. As seen in Figure 3.1, this device contains Microprocessor unit (MPU) based on the ARM Cortex A8 microprocessor, digital signal processor (DSP) core, integrated Camera image signal processor, Display subsystem and level 3 (L3) and level 4 (L4) interconnects that provide high-bandwidth data transfers to the internal and external memory controllers and to on-chip peripherals [45].

As discussed earlier, the dynamic and static power together represent the total energy consumption of a device. Texas Instrument's OMAP35x processor optimises system power consumption by employing the active power consumption techniques such as dynamic voltage and frequency scaling, adaptive voltage scaling, dynamic power switching and the static power consumption techniques such as power gating.

The OMAP35x contains different voltage, clock and power domains. The partition of these domains in the chip is achieved by employing the most commonly used techniques such as clock gating and power gating. Each voltage domain has several power domains, and voltage distribution is subdivided further in these power domains. The power domains enable/disable power or switch into a low leakage retention mode and this can be done using software.

DVFS helps to achieve the performance requirements of the application by reducing voltage and frequency. These hardware capabilities are combined with software to define Operating Performance Points (OPPs), which are typical combinations of clock frequencies and voltage levels. Predefined operating performance points for the OMAP35 can be seen in Figure 3.2.

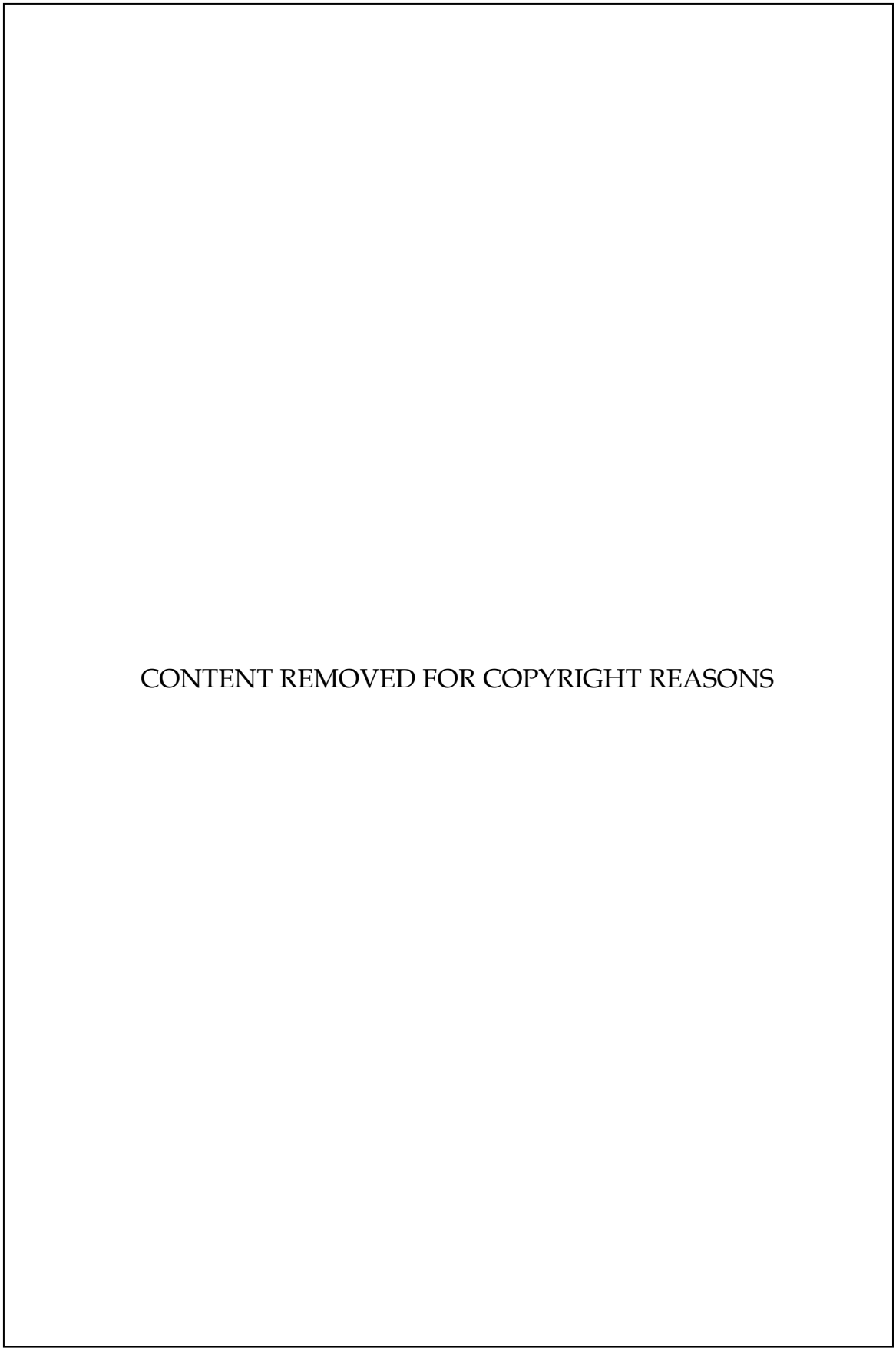


Figure 3.1: OMAP35X functional block diagram [45]

When system needs high performance, clock the ARM at 650 MHz, DSP at 400 MHz, and voltage to 1.35V. In the case of low performance, define an OPP with ARM at 125, DSP at 90 MHz clocks and 0.95V. These settings can be varied by using software controlled registers.

Figure 3.2: Predefined OPPs for the OMAP35x [47]

Another active power management technique is adaptive voltage scaling, which monitors temperature, process performance and changes supply voltage dynamically. In OMAP, this technique has been implemented with a dedicated on-chip Smart Reflex technology based hardware. This circuit has a feedback loop and runs without processor involvement and changes voltage levels adaptively by considering temperature, process variations and silicon degradation. Once the operating performance point for the application has been decided, the algorithm sends signal to the external regulator over I2C bus for the new voltage and requests clock generator for the new frequency. The combination of both dynamic voltage and frequency scaling and adaptive voltage scaling techniques provide better results to achieve power efficiency in OMAP devices.

DPS is the other method of active power management used in OMAP35x, which determines idle parts of a device and keeps them into a power down mode. However this technique keeps only a section of an OMAP35x in a low power state, but in some cases it is more effective to keep the entire device in low power mode either automatically or by request. The

mechanism is developed in OMAP using a mechanism called static leakage management (SLM). Generally, Processors contains several low power modes from standby to deep sleep mode to consume very little power but wake up time always consume more power and time. Using this method makes OMAP device wake up time faster because the content is loaded in external memory and no need to wait for the full operating system restart.

To reduce power, it is imperative to make the best use of low power modes and consider the best wake up logic. Because wake up domain and I/O leakage are the main sources of power consumption. In OMAP devices, the lowest power mode is the device off mode where the device can wake up independently. All the other domains are off except the wake up logic. This domain can use the slowest 32 kHz clock in case the main clock is switched off. In order to provide more power savings, all the modules are internally switched off as there is no need to retain any information. However, The OMAP35x can communicate to external regulators if the situation requests, even in this low power mode. The current state of the system is loaded in external memory before the device enters this mode. As soon as the wake up request is received, the processor jumps to a user defined function.

3.2 Samsung Low Power 32bit RISC Microprocessor - S3C2410X

The S3C2410X was developed using an ARM920T core, 0.18um CMOS standard cells and a memory compiler [48]. The block diagram of this device is shown in Figure 3.3. Its low-power, simple, elegant and fully static design is particularly suitable for cost- and power-sensitive applications. It adopts a new bus architecture called Advanced Microcontroller Bus Architecture (AMBA) [48]. Peripherals like Universal

Asynchronous Receiver/Transmitter (UART) interface and USB are included by way of the on chip AMBA Peripherals Bus (APB).

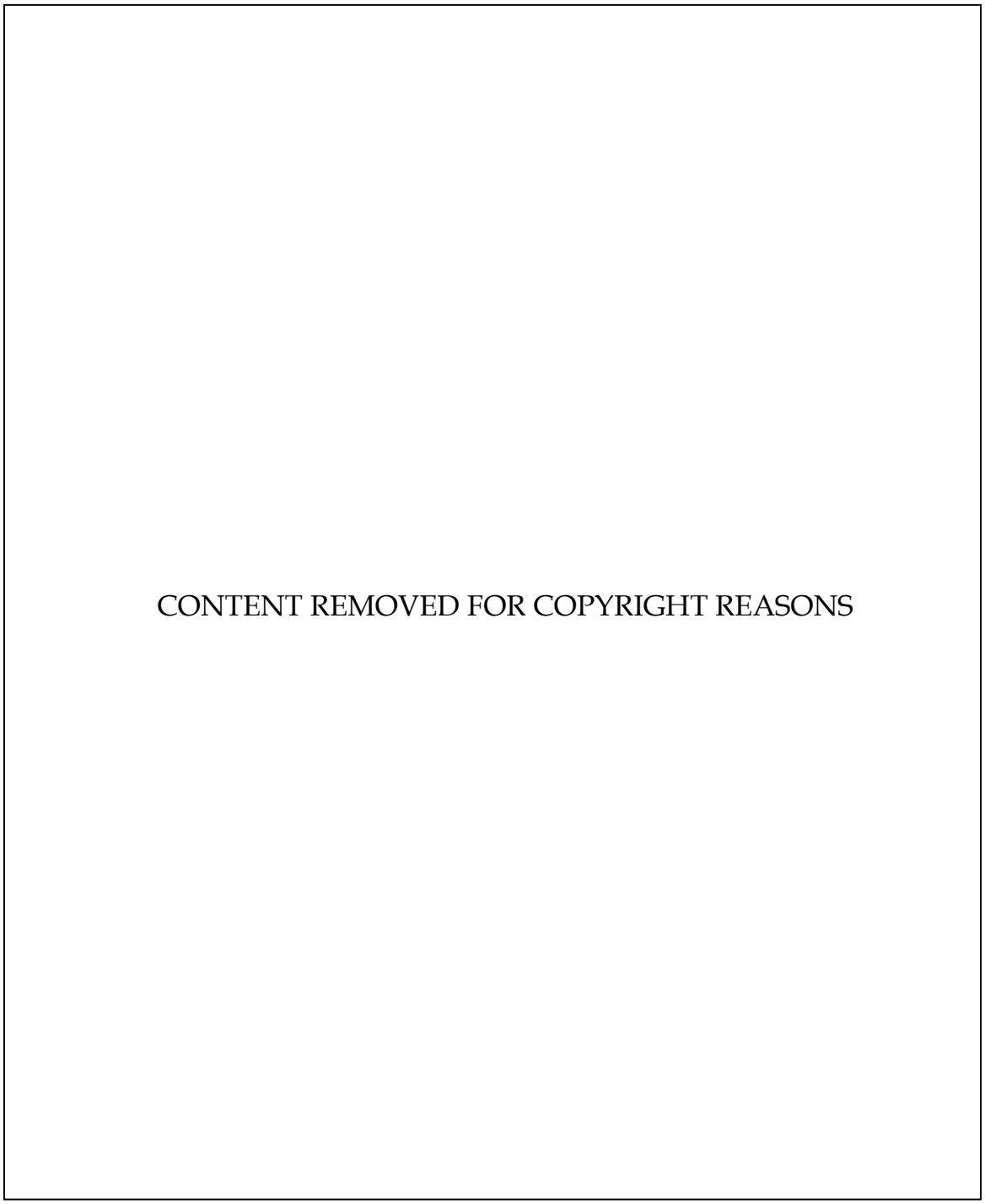


Figure 3.3: S3C2410X Device Functional Block diagram [48]

The processor supports four low power modes: Normal, Slow, Idle, and Power-off mode. Different power domains are designed with power gating technique in this chip to power off certain modules when they are not in use. The processor uses power management controller (PMC), to manage the chip's clock, reset and power supply and to control the processor to shift among different power modes. In addition to this, clock gating might be useful to reduce the dynamic power consumption of the digital unit by providing the clock signal to the digital unit whenever required. To reduce static power, power gating technique can also be used. The non-used blocks are powered off through power gates.

3.3 System Level Design Techniques

The microcontroller vendors use almost all the above mentioned techniques to provide the user a low-cost, low-power and high-performance device. The techniques to build low power embedded systems using these devices are discussed in this section. As with any engineering-oriented discipline, domain expertise stands as an essential prerequisite for leveraging low power technology. Knowing your application domain is a key to knowing which power reduction technique will be effective. With the domain knowledge for the application and having reviewed VLSI techniques, its time move on to the stage of discussing power optimisation techniques available in commercial microcontroller units.

3.3.1 Low power modes

Most microcontrollers feature various low power modes ranging from standby to deep sleep to provide better power optimisations. There is no point in selecting the most powerful and low power device if it is not used effectively. The key thing here is considering all available modes by

keeping the application in mind. Consequently, to develop a low power system, the lowest operating voltage and frequency must be selected to meet the real time goals.

For example, the lowest power mode is sleep mode in PIC18 microcontrollers. In this mode the device stops functioning and disables all clocks except for the wake up logic. In addition to this mode, the device contains some more low power modes as shown in Table 1.

Table 1: Selected operative modes of PIC18F2420

Mode	Frequency	Minimum VDO (V)	Operating Current (μA)
High Speed	20 MHz	5.0	2500
Primary Run	1 MHz	2.0	125
Low Power	32 kHz	2.0	10
Sleep	-	2.0	1.5

Depending on the application requirements, the appropriate mode and the extra logic for switching from low power mode to normal mode of operation need to be considered. Task scheduling algorithms provide better power savings along with switching overhead and application deadlines.

3.3.2 Multiple clock sources and switching

Most modern microcontrollers are coming up with a new feature of having multiple clock sources and changing them on the fly to reduce power consumption. The advantage of this feature is that the user has flexibility over deciding the clock frequency and the clock source by keeping the application demands and power in mind.

Kristjansson [49] stated different ways of reducing power consumption using this feature in ARM7 core. When the processor in standby mode, Real time clock can be used to wake the device to active mode. Instead of using internal PCLK, an external 32 kHz clock to keep RTC active in stanby mode provides better savings. Another advantage mentioned in [49] is optimising code execution and power consumption by changing the clock switching on the fly. The author also hypothesized that "Mixing clock sources, a PLL, a divider and clock enable/disable provides a very flexible architecture for power management."

3.3.3 Multiple regulators

Power optimisation in low power modes is possible by using multiple dedicated on-chip low-power regulators as add-ons in its architecture [49]. The low power regulator can be used to maintain real time clock and 32 kHz oscillator in standby mode. Using the low power regulator instead of the main regulator allows power consumption to be reduced in standby mode.

3.3.4 Wake-up energy

The wake-up time and wake-up source are important parameters that contribute to the overall MCU power consumption. Wake-up time is the time processor takes to come out of a sleep mode and it is very important for systems with short active times because the wake up process consumes

the same amount of current as the normal process does. The main component of wake-up time is the Oscillator start up time.

The proper selection of wake-up sources is another important factor to reduce power consumption during wake-up process. Modern devices have extensive variety of wake-up sources and can wake up from almost any peripheral module on the device.

Almost all microcontrollers have come up with a new feature, which allows a system to use a slow start-up source with a fast wake-up time. For Example, In PIC MCUs, Two-Speed Start-up initially wakes up from the internal RC oscillator, which typically starts up in a few milliseconds. To take advantage of this feature, the processor uses this oscillator until the primary clock source stabilises, reducing total time by executing any code that is not timing critical.

3.3.5 Flash vs. RAM

Running a program or moving a data from external memory will always consume more power than using on chip memory. However, it may not be possible to fit everything on chip, in that case, the program should ideally be created in such a way that many intensive routines should use on chip memory as much as possible. Furthermore, when using on chip memory, accessing Flash consumes more power than SRAMs [50]. To benefit from this feature, many devices offer an option to execute program from RAM to reduce power consumption.

To execute from RAM, the code must be copied from Flash to RAM and this task requires major time and power. To make this task worthwhile, computationally intensive routines must be copied into RAM. The main drawback of this method is, some devices can reduce performance due to the restrictions of using a single memory bus for execution and operation.

For example, the Harvard architecture used in PIC32 devices, generally uses one memory bus to read from Flash and another to read from RAM. This consents the device to run at high speed without processor intervention. However, when operating from RAM, only a single memory bus is used, which can force the processor to stall during some instructions when writing to RAM.

3.4 Available Technologies

3.4.1 Advanced microcontroller bus architecture

The Advanced Microcontroller Bus Architecture was introduced by ARM to use as on-chip bus in SoC designs. The first AMBA buses were Advanced System Bus (ASB) and Advanced Peripheral Bus (APB). The APB comes with a low power peripheral and is mainly proposed for connecting simple peripherals. The APB can be sometimes optimised for reduced interface complexity and minimal power consumption for supporting peripheral functions [50]. The bus can be used with any version of the system bus. Two or more peripheral buses may be used so as to optimise the clock on each bus based on what kind of tasks the peripherals are performing. The typical Bus architecture is shown in Figure 3.4, where the computational intensive peripherals are one side of the bus and less intensive peripherals such as timers are on the others side of the bus. This arrangement allows to disable the clock on timer side and wait for a serial communication to wake up the processor. In addition to this, multiple peripherals can be disabled using a single instruction by enabling and disabling the clock independently. A function of APB divider is to allow power savings when an application does not require any peripherals to run at the full processor rate [50].

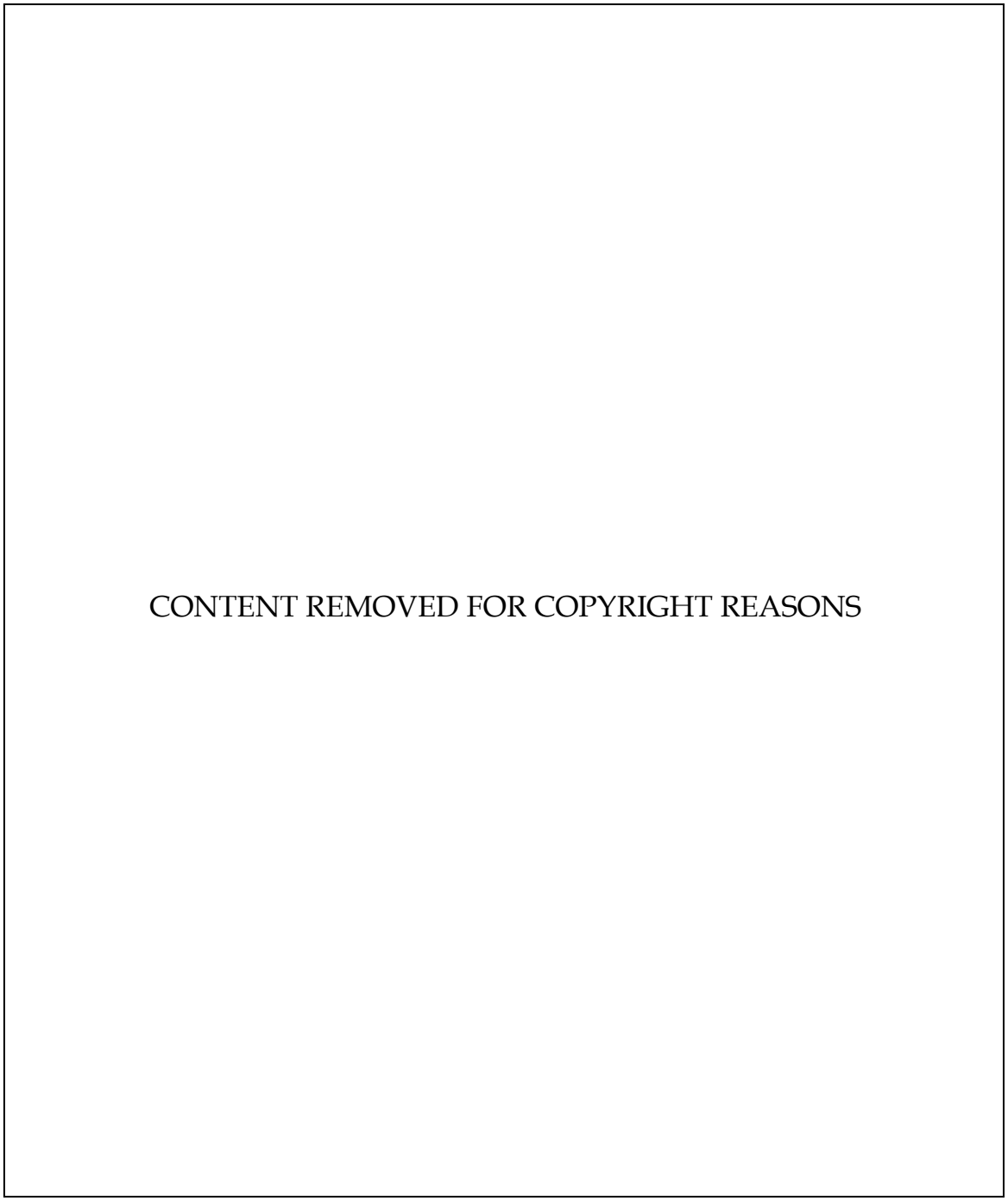


Figure 3.4: ARM Bus Architecture [50]

The AMBA High-performance Bus (AHB) was added in AMBA version 2. ARM architectures with AMBA are profited immensely in terms of power management. The AHB is a single clock edge protocol and uses a full duplex parallel communication whereas the APB uses massive memory-I/O accesses. The AHB in combination with advanced low power bus

encoding techniques largely reduces the power requirement for memory accesses to the caches in ARM11 and highly advanced ARM Cortex cores [50].

3.4.2 Intelligent Energy Management (IEM)

The concept of IEM technology uses the most effective power management technique called dynamic voltage and frequency scaling to achieve more power savings. The block diagram of IEM technology is shown in Figure 3.5. The IEM software monitors the system workload continuously and generates a performance request based on the application requirements. Then IEM hardware adjusts the operating voltage and frequency of the processor accordingly. To use IEM, processor must be implemented with appropriate register slices and an Intelligent Energy Controller (IEC).

Figure 3.5: IEM Technology [51]

The effects of IEM and DVFS are evaluated by David et.al [52]. The study was done by considering three tasks and no activity in between the tasks. They considered two scenarios to evaluate the performance of the IEM. In the first case, they run the tasks by the ARM core at maximum frequency without using IEM. As soon as the task is completed, the core enters sleep mode. The main overhead in this is switching in and out of idle mode

because it consumes considerable power. The other scenario is using IEM and running the task with application requirements and avoiding idle time. They have been practically verified that IEM provides significant power reductions. Thus IEM intelligently manages performance and power by varying the frequency and operating voltage on the fly.

ARM and National Semiconductor have come up with new solutions to extend the battery life of mobile devices. This technology can be used to match the varying performance needs of the application by using dynamic voltage scaling and adaptive voltage scaling along with IEM.

The system architecture of this design contains IEM with Adaptive Power Control and AMBA peripheral bus. The AMBA bus connects to the processor through the intelligent energy manager. Based on the system workload and requirements, the IEM software generates a request to modify the operating voltage and frequency. Using either open loop or closed loop, the adaptive power control block changes the voltage without disturbing the processor. The APC block provides a quick response but also assures the system operates with minimum safe operating voltage.

3.5 Emerging Technologies

3.5.1 Cloud computing

As embedded systems are widely used in various fields, the demand for cloud computing in embedded systems may increase exponentially. This is another approach to reduce the energy consumption of embedded systems. These systems move computation tasks to remote computing facilities. This can be achieved by renting virtual machines from cloud providers or data centres. The energy constrained embedded system simply works as a terminal, and virtual machines in the remote cloud provider are rented to actually execute tasks. In this case, the embedded system, as a terminal,

does not require a significant amount of energy. And a number of virtual machines can be rented based on the computational demand of tasks.

3.5.2 Resonant clock meshes

For digital circuits, clock signal is the main essential signal to perform operations and synchronises all events. Because the clock is the main heart of any embedded system, the signal must be created in such a way that it must not contain any jitter and skew. The clock signal is distributed around the chip using the most traditional approach called clock tree. The disadvantage of this method is the presence of clock skew and it is solved by using clock mesh approach. This method minimises skew but consumes more power than a clock tree. The power consumption of a processor due to its clock network vary depending on the distribution scheme used. To overcome this problem, Cyclos have come up with a new solution to reduce CPU power consumption significantly [53].

The Cyclos developed a tank circuit, which is a combination of an inductor and capacitor connected in parallel to a power source as shown in Figure 3.6. The operation of the circuit is similar to a mechanical pendulum. As can be seen in the Figure 3.6, current flows from a charged capacitor to an inductor and creates a magnetic field around the inductor. This process continues till the voltage across the capacitor reaches zero. The current continues to flow due to the energy provided by the magnetic field around the inductor and charges the capacitor with opposite polarity. This begins the cycle with opposite flow of current as when the cycle started. Thus the circuit provides continuous swinging with minimal energy compared to the tradition methods. Cyclos states that this design reportedly cuts clock distribution power by up to 24% while maintaining the low clock-skew target required by high-performance processors. Cyclos claims that using

its technology can cut total IC power by up to 10%. AMD used this technology in their Trinity processor to push above 4GHz speed [53].

Figure 3.6: Parallel LC Tank Circuit [53]

3.6 Summary

In this chapter, Different system level power saving techniques have been discussed. In addition to this, industry power management technologies and low power processors are presented. The power management technologies such as advanced microcontroller bus architecture and intelligent energy management are covered in this chapter. This provides a basis to understand the concepts of low power and to address the issues of power efficiency. In the next chapter, the concepts and theory behind the latest power saving technique, aggressive scaling, is discussed in detail.

CHAPTER 4

Simulation of Aggressive Scaling System for Power Efficiency

In this chapter, the idea and the principle behind the aggressive scaling mechanisms is presented. It also provides the latest available research, which is categorised into three types based on their hardware realization. It covers the details and principle of operation for each category and their drawbacks which provide the background for this research work. In order to investigate the power savings, a simulation study of the aggressive scaling system is carried out as a whole system with integrated components such as controller, buck converter and critical path delay monitor. The procedure to find the optimal operating conditions based on the feedback information received from processor is demonstrated. The maximum frequency selection is primarily done by the application requirements, mainly on the critical path of the design. From this, a set of

worst-case operating conditions are declared, under which the system has to operate correctly. The design specifications used for the simulation are stated and the results obtained are presented in this chapter.

The operating voltage and frequency of processor or electronic systems work at a lower rate than the typical operating conditions. The supply voltage and frequency are padded with extra margins to operate the system correctly at the worst- case condition. This is to provide a safe and reliable operation, although the system and its components are verified for the worst case conditions such as operating and manufacturing variations. In general, typical worst-case manufacturing scenarios such as doping concentration variation, cross coupling noise etc. and sudden variations in operating conditions such as unexpected voltage drops, temperature fluctuations can have an impact on the design performance. Moreover, the data being processed and applied to the system creates variation in delay. To account for these variations and to ensure correct operation, systems are padded with extra margins. Occurrence of these worst-case situations may be very rare or impossible with the advanced process scaling technologies [54]. These margins result in extra overheads such as power or performance loss. To overcome the problem of excessive margins recent research works propose aggressive scaling mechanisms. They work beyond the worst case estimates by reducing excess margins to achieve maximum power savings.

4.1 Aggressive Scaling

Aggressive scaling mechanisms work on the principle that exploits the excess timing margin by placing an in-situ error detection and correction circuitry in the critical path of the design. The error detection and correction circuits monitor the critical path behaviour as the variations occur. As a mechanism to support voltage or frequency scaling beyond

sub-threshold region in all types of embedded systems, several studies have been published by enlisting various aggressive scaling technologies to perform stable and reliable operation [31], [55][60]. Based on their hardware realization, the aggressive scaling techniques are divided into three categories, direct monitoring [31], 54, [59]-[61], indirect monitoring [55][56] and time borrowing [57], [62].

Figure 4.1: Razor flip-flop functional block diagram [31]

4.1.1 Direct monitoring

One of the earliest studies on direct monitoring approach to obtain maximum power savings was published by Ernest et al. in 2003 [31]. This study was conducted to scale the supply voltage as low as possible while providing reliable operation. In order to provide a stable operation, an in-situ timing monitor was used to monitor the actual circuit delay variations in order to scale the supply voltage. The Razor technique functional block diagram is shown in Figure 4.1. With this technique, the supply voltage is tuned beyond the typical voltage level to gain better power savings. The circuit double samples the input data with main clock and delayed clock. The outputs of these two sequential elements are compared to detect any

timing violations. The detected error is fixed by capturing the correct data from the latch instead of the flip-flop. In this way, the circuit self-tunes supply voltage by monitoring timing errors. The simulations and the detail understanding of timing errors with respect to the supply voltage can be found in [31]. Original flip-flops are replaced with these special flip-flops of the critical path to detect any timing violations

The major problem with flip-flop based design is meta-stability. The input data of the flip-flop can vary any time before set-up time, and then the flip-flop enters into meta-stable state. This can cause unpredicted behaviour of the circuit. To overcome this situation, razor used meta-stability detector in the flip-flop circuit. The power overhead incurred by using error detection circuit is minimal during error–free operation. Further, the power gain overcomes the power overhead incurred. Razor technique also included to address all different kinds of variations.

Various updates and extensions to the original razor technique have been proposed. Razor II [54] proposed to overcome the drawbacks of razor correction mechanism. In [59], [60], Bubble razor uses two latch based architecture to overcome the drawbacks of Razor technique. On the other hand, the practical implementation of Razor technique has been studied to evaluate the ease of the architecture to integrate with the application. The integration of Razor technique with ARM processor architecture is studied in [61]. Similarly the integration of razor technique into an FPGA based design is proposed and studied in [63].

4.1.2 Indirect monitoring

The Timing-Error-Avoidance prototype, TEAtime [55] indirectly monitors the critical path behaviour as the voltage and frequency are scaled until the timing violations occur. The system predicts the timing errors using critical

path replica, which is made up of delay chains, instead of the original critical path. The conceptual diagram of TEAtime is shown in Figure 4.2. The replica co-exists along with the original design. Therefore, the advantage of this method is that the original design need not be changed to implement this circuit. Only the critical path in the design is replicated to predict the error before it actually happens. The tracking logic in the architecture is the critical path replica with added safety margin. Input to this block is a toggle flip-flop, which changes from 0 to 1 and 1 to 0 on alternate cycles. This ensures that the signal tests both types of transmission [56]. The output passed through an XOR gate to the timing checker unit, which decides whether clock frequency need to be increased or not. The timing checker output signal controls the counting direction of the up-down counter and DAC coverts the counter output to an analogue voltage signal. This signal sets the clock frequency by controlling the voltage-controlled oscillator, and the VCO output becomes the system clock, completing the feedback loop as shown in Figure 4.2. The process makes sure that the clock period will never be less than the delay through the tracking logic plus the safety margin delay. This situation ensures that no timing error will occur in the real logic, which ensures correct system operation. The most common digital system issues are meta-stability and multiple worst case paths. Both the issues are addressed with slight changes to the original design. To incorporate TEAtime into the design, the architecture of the design need to be modified. However, since the critical path replica coexists in the design, there is no need to alter the critical path.

Figure 4.2: TEAtime: A canary circuits based approach [55]

4.1.3 Time–borrowing

This technique masks the timing error by borrowing time from the next following stages. Timber [57] is the best example for this category. Timber proposed two types of elements in the architecture; timber flip-flop and latch. Flip-flops generally capture data with respect to the positive or negative edge of the clock. Time borrowing in flip-flop is achieved by further extending the sampling edge to capture the late arrival of data caused by dynamic variations. The checking period is the period used to detect timing errors. In this case, the entire checking period is divided into two types: time-borrowing (TB) and error detection (ED) as shown in Figure 4.4. If single stage error occurs, the period is extended further to 1 TB unit. If two stage error occurs, the period extends to 2 units. Thus, flip-flop borrows time from next stages to recover from error. The TIMBER flip-flop clock control logic is shown in Figure 4.3. EN is the enable signal, which indicates time-borrowing is turned on or off. In case of time-borrowing, the signal must be set to zero. CK is the clock signal used to capture the data of all the sequential elements in the circuit. The three

intervals in the checking period are encoded using select input signals S0 and S1. If the two signals are 00, that indicates TB interval, and 01, 10 specifies ED intervals. DCK is the delayed clock signal, which is used to late arrival data. To avoid the meta-stability problem of flip-flop, in TIMBER, proposed latch configuration. The positive duration of the clock can be used as checking period because it is a level-sensitive sequential element.

Figure 4.3: TIMBER flip-flop clock control [57]

Figure 4.4: TIMBER based error detection and masking [57]

4.2 A Complete Aggressive Voltage Scaling System: Simulation Study

The simulation of the complete system with aggressive scaling is shown in this section using Matlab and Simulink. With the available resources in Matlab, the simulations are carried out to investigate the power saving offered by the aggressive scaling schemes.

Figure 4.5 shows the block diagram of the entire AVS system with timing speculation. The individual blocks are created using the libraries available in Matlab/Simulink software. The controller gets the information about timing margin from the delay monitoring unit (DMU) and decides whether voltage should be increased or decreased. In general, several critical path replicas and monitoring units are inserted around the real critical paths to detect timing predictions. The monitoring unit outputs potential error before the real timing error occurs. The controller makes the decision by analyzing error rate with the pre-defined minimum timing margin. If the error rate is less than the margin, the controller raises the supply voltage but makes sure that there exists plenty of margin to tolerate noise fluctuations. When the timing margin increases the threshold value, the controller scales down the supply voltage to maintain minimum power dissipation. Only one delay monitor unit is considered in this work to make the simulation of the whole AVS mechanism effective. Therefore the controller decides one of the actions mentioned above and gives the command to the voltage actuator to adjust the supply voltage of the processor core based on the timing margin signals T1 and T0.

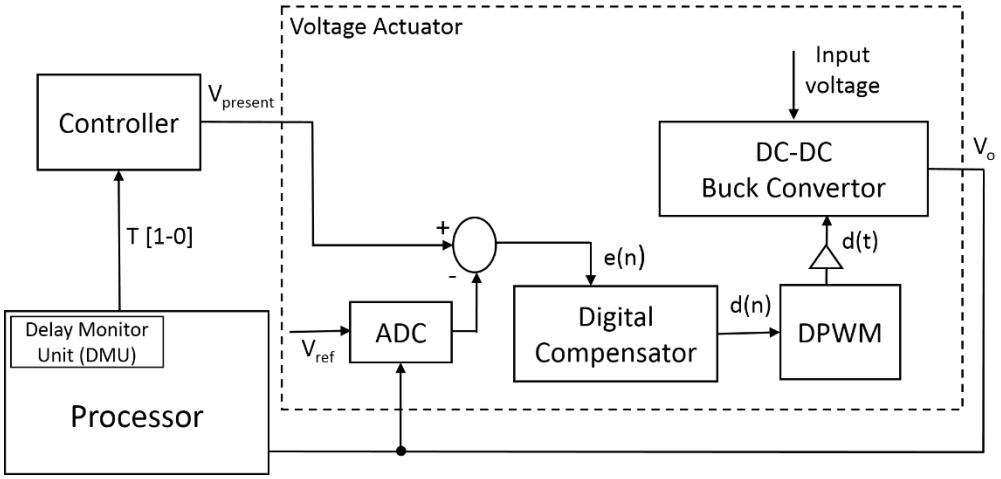


Figure 4.5: Block diagram of the whole aggressive scaling system

4.2.1 Voltage actuator

DC/DC converters are found to be the best solution in AVS enabled systems. Recently digital controlled DC/DC converter has drawn wide attention due to its fast response, low power consumption and compatibility with CMOS processes [64]. Many existing digital controllers use high resolution and high speed analogue to digital (A/D) converter to sample and convert the regulated output voltage into digital signal in feedback loop to determine the duty ratio of the DC/DC converter. Its bandwidth limits the overall dynamic response of the converter.

In general, switching power supplies employ pulse width modulation (PWM) and use either voltage or current-mode control to regulate the output voltage level. The current-mode control is usually used in modern switching regulator designs to overcome the disadvantages of the voltage-mode control [42]. There are two main advantages of current-mode, one is voltage error is directly reflected in the switching current and the other one is achieving fast response time. Considering these benefits, switching regulator with a DC-DC buck converter along with a digital closed loop system is used in this work. This is the most general setup used for

microprocessor systems. The Simulink model of the voltage actuator with feedback loop is presented in Appendix D.

The system mainly comprises of an output filter, a switching circuit, a pulse width modulation and a controller. PWM modulates the duty cycle d, duration of the ON and OFF pulses that are applied to the switching transistor in a switching converter. As shown in Figure 4.5, the converter's output voltage (V_o) is compared with a reference voltage (V_{ref}) via compensation circuit to generate the error voltage signal (V_{error}). The ADC in the digital controller computes the digitized error, e(n) after comparing with the reference input. The digital compensator block uses discrete time integral unit to compute digital control command using d(n) from digital error, e(n) and PWM converts d(n) to time domain signal, d (t). The PWM unit converts the error voltage into the clocking signal with duty cycle to control the converter. The duty cycle is adjusted based on the error signal to make the output voltage follow the reference value and vary the output voltage to a fixed voltage level.

The simulation of the proposed system is performed using Matlab/Simulink and achieved a power savings of 21% compared to a fixed voltage system. To demonstrate the effectiveness of the system, different voltage and frequency transitions are considered. The different configurations of the control signals can be seen in Figure 4.6. Here T_1 and T_0 are the predicted error signals before the actual timing error occurs and depending on this value, an appropriate action is performed. For instance T_1 and T_0 are {00} the processor will increase its clock frequency in order to exploit the timing margins. If T_1 and T_0 are {01} the processor will decrease its clock frequency to prevent the failure of the actual critical paths. F_{out} signal indicates the frequency increase or decrease and $V_{present}$ indicates the reference voltage supplied to the buck converter from the controller.

The controller's next job after frequency selection is to provide reference voltage to the converter. Here the condition is, frequency and voltage values can be set to any value within the predefined upper and lower bounds. The following linear equation describes the relationship between voltage and frequency.

$$m = \frac{f_{max} - f_{min}}{V_{max} - V_{min}} \tag{4-1}$$

$$V_{present} = \frac{f_{present} - f_{min}}{m} + V_{min} \tag{4-2}$$

Where m is the slope of the voltage and frequency function, f_{max} and f_{min} are the maximum and minimum values of the frequency, V_{max} and V_{min} are the maximum and minimum values of voltage, $V_{present}$ is the present value of the voltage and $F_{present}$ is the present value of the frequency.

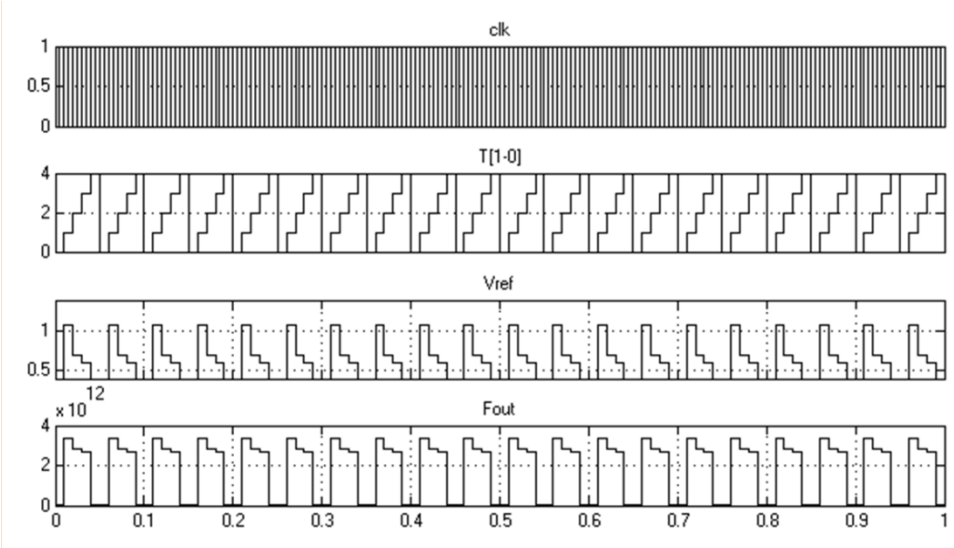


Figure 4.6: Simulation results of the AVS controller

The converter provides regulated output voltage between 0.6 and 1.2 volts for an input voltage of 12V. The output capacitor and inductor values used are 376 µF and 4.1µH respectively. Design performance summary of the

proposed digital controller in closed loop operation is listed in Table 2. This study results provides motivation to propose an effective timing monitor to achieve maximum power savings or performance improvement

Table 2: Parameters and design specifications of voltage actuator

Parameter	**Value**
Frequency	2.7 – 3.6 GHz
Regulated output voltage	0.6-1.2 V
ADC Quantization	15 mV
DPWM switching frequency	1 MHZ
Power efficiency	84-94 %

4.3 Summary

In general, the excess timing margins are incorporated for a typical system operating conditions to include unpredicted process and environmental changes. The characteristic aggressive scaling based system employs monitors and it usually operates the system outside of the desired operating range to eliminate the excess timing margins. This situation is generally advantageous to the emerging modern processor designs but the challenge is to provide a stable and reliable operation.

The latest research addresses the above issue at different levels of the system design hierarchy. Among these, the more preferable are the circuit level techniques which directly track the implications of reducing the margins and thus allowing the system to be operated beyond the worst case estimates. In this chapter, the available research is categorised into

three categories based on their hardware design. In brief, the direct approach samples the data at two different instants of time to detect any timing violations, compares them to detect an error and corrects them using a special circuitry. The main drawback of this approach is stalling the pipeline for an entire cycle in the event of an error [54]. As a result, the approach has a major performance impact in the event of an error.

On the other hand, the indirect approach predicts the error by replicating the critical path in the design before it actually happens. This approach also makes the system work beyond the operating range while incurring errors. The main limitation of this technique is the real and duplicated critical paths differ in the presence of manufacturing and process variations; therefore, timing margins are required to account for these variations. Thus it is not possible to remove timing margins completely with this approach.

Contrary to the above two approaches, the time borrowing approach borrows time from the next successive pipeline stages. The limitation of this approach is the small checking window for the timing speculation.

Also, an aggressive voltage scaling (AVS) based system using delay monitoring unit along with a closed loop control buck converter is presented. The proposed technique scales down the operating voltage based on the timing error provided by the delay monitor unit. The controller makes decisions and gives command to the closed loop buck converter. The buck converter consists of an Analogue to Digital Converter (ADC), a Pulse Width Modulation (PWM) and a Digital compensator. A complete model of the system is developed using Matlab/Simulink. This system is simulated over a frequency range of 2.7 to 3.6 GHz

corresponding to a regulated voltage range of 0.6 to 1.2V and achieved 21% power savings compared to the system without the AVS technique. The next chapter details an overview and experimental analysis of Power Analysis Attacks and Countermeasures.

CHAPTER 5

Power Analysis Attacks and Countermeasures: Overview and Experimental Study

The Internet of Things (IoT) is the emerging technology of embedded systems. In brief, IoT is the network of objects connected tighter to perform tasks without any human intervention. Since any device connected to the network can have access to all devices connected to the same network and also end users poses a significant security problem, a key focus area of the new development of IoT is the security of the IoT devices. Similarly the other applications of embedded systems such as online payments, banking transactions and encrypted storage require resistance from attacks. To improve the security of embedded systems, software level approaches, complex mathematical functions are generally employed to encrypt and decrypt data. However, software level security alone proven to be inadequate to protect the devices from attacks.

Therefore, today's embedded systems include security also as one of the design constraints. That extends the security scheme all the way down to circuit level. Therefore, to address the needs of modern systems, a technique based on the circuit and architectural level is proposed here. This chapter presents a brief overview of differential power analysis attacks, a major threat to the hardware security of systems and then introduces the most Advanced Encryption Algorithm, AES to protect the system from the power analysis attacks. In order to improve the security of embedded systems, the first step must be implementing countermeasure to the Differential Power Analysis (DPA) attacks. The procedure involved in the practical realization of DPA attacks is provided in this chapter along with the experimental setup and the methodology used to compare real and hypothetical values to detect the secure information used.

Research in cryptography to provide secure systems results in more secured cryptographic algorithms that provide data security and authenticity. However, the most secured cryptographic algorithms are mathematically secured but not their physical realization. The hardware implementation of the encryption and decryption process leaks the secret information in the system unintentionally; such information is called side-channel information leakage. In general, there are different potential information leakage side-channels such as power consumption [65], [66] , electromagnetic radiation [67], [68] and execution time [69] in a cryptographic system. The most common method is obtaining the system information through the power consumption called power analysis attacks. The power analysis attacks extract the secret information by measuring the power consumption while the system is performing the encryption or decryption process.

According to Kocher [19], power analysis attacks are two types: simple power analysis attacks (SPA) [20] and Differential power analysis attacks (DPA) [21]. SPA, as the name suggests, analyses the power trace to identify the data and instruction used while encryption or decryption process. Generally, the attacker identifies the number of iterations used to encrypt or decrypt the data. For example Data Encryption Standard (DES) algorithm contains 8 encryption rounds; AES algorithm uses 16 rounds of encryption as seen in Figure 5.1. The power trace captured using our experimental setup is shown in Figure 5.2. With the clear observation of the trace, the repetition of patterns in a trace can be identified. Processors generally contain different set instructions, which exhibit different power profiles. For instance, XOR instruction usually display the highest peak when compared with AND instruction, which display the lowest peak in the power trace. Data identification can be done using Hamming weight, which reveal the number of transitions during instruction execution. Simple power analysis attacks are very easy to implement but the prediction depends on the attacker's knowledge and expertise. In addition, SPA becomes more complex even in simple countermeasures are used.

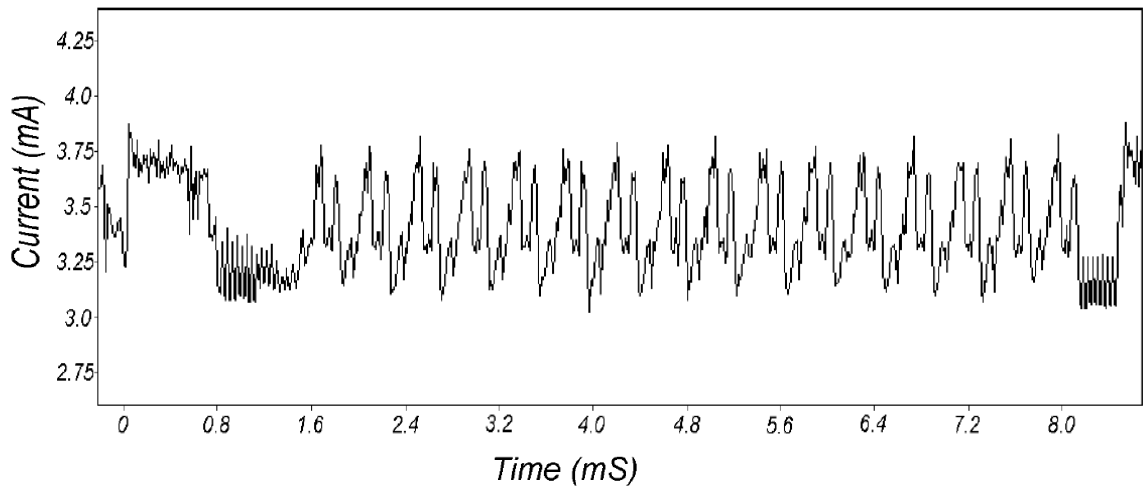


Figure 5.1: Power trace of AES algorithm [6]

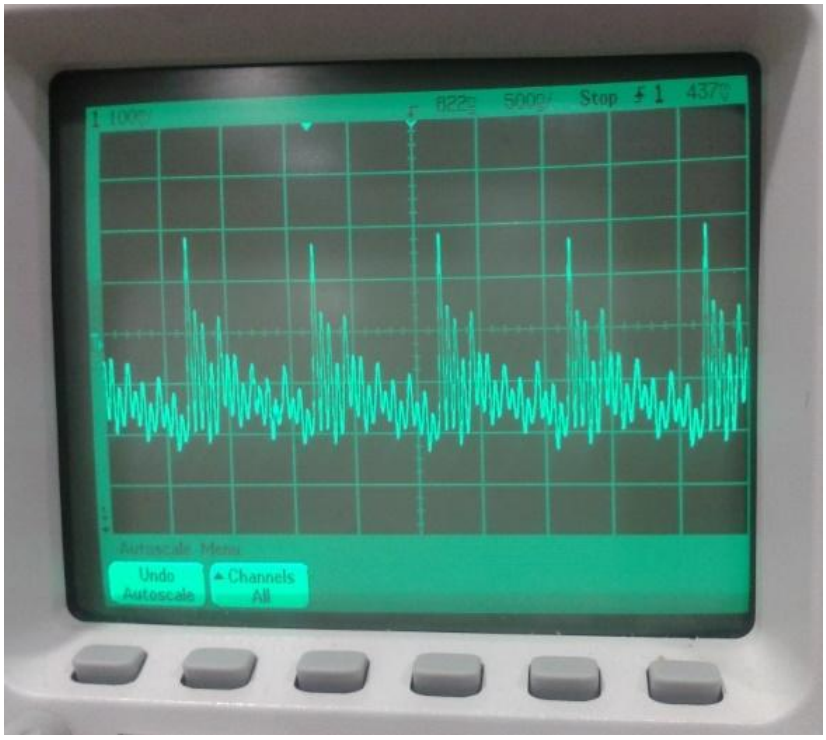


Figure 5.2: Captured Power trace of encryption algorithm using PIC microcontroller

In contrast, a Differential Power analysis Attack (DPA) [65] [70] is the most powerful and sophisticated method to identify the secret key information when countermeasures are used. DPA was first introduced by Kocher in [65], where statistical analysis is used to determine the secret key information in the system. The statistical analysis of DPA eases the job of revealing the key even if the measurements contain large amounts of noise. The direct analysis of SPA in the presence of noise is impossible to reveal the key.

5.1 AES principle

The Advanced Encryption Standard (AES) is a means of encrypting and decrypting data adapted by the National Institution of Standards and

Technology (NIST) on October 2, 2000 [71]. The standard DES algorithm was replaced by this advanced algorithm, which was introduced by Rijandel [71], [72]. This algorithm encrypts symmetric block cipher and decrypts data using a secret key. As seen in Figure 5.3, the flowchart provides the sequence of operations grouped as AddRoundKey, SubBytes, Shiftrows and Mixcolumns. N is the number of encryption rounds. The algorithm fragments the input to be encrypted into sub blocks and repeats the same procedure for N-1 encryption rounds. The AddRoundKey operation combines the input data with a secret key. It performs a simple bit-wise XOR operation of plain text with the key. This function does need an explicit inverse function for decryption. In SubBytes, each byte in the state is replaced by its corresponding byte in a table called Sbox. Sbox contains multiplicative inverse of all possible bytes over $GF(2^8)$ followed by an offline transformation [73]. This function is implemented as a look up table in this case but can be computed dynamically. In the ShiftRows transformation, each row of the state is considered separately and the bytes in that row are cyclically shifted to the left based upon the key-size of the algorithm. For the 128-bit key, the first row is unchanged; however, the second, third and fourth rows are shifted by one, two, and three bytes respectively. The MixColumns transformation is a bricklayer permutation operating on each column of the state [73]. In MixColumns, columns of the State are considered as a four-term polynomial over $GF(2^8)$, then are multiplied with a fixed polynomial $c(x) = 03.x^3 + 01.x^2 + 01.x + 02$. The algorithm for the decryption has the same structure but uses mathematical inverses of the encryption steps, i.e. InvSubBytes, InvShiftRows, and InvMixColumns. The round keys are the same as those in encryption but are used in reverse order. The Sbox is the most important and power consuming element for power analysis attacks. Therefore, for most of the research experiments, the Sbox design is used as system being attacked.

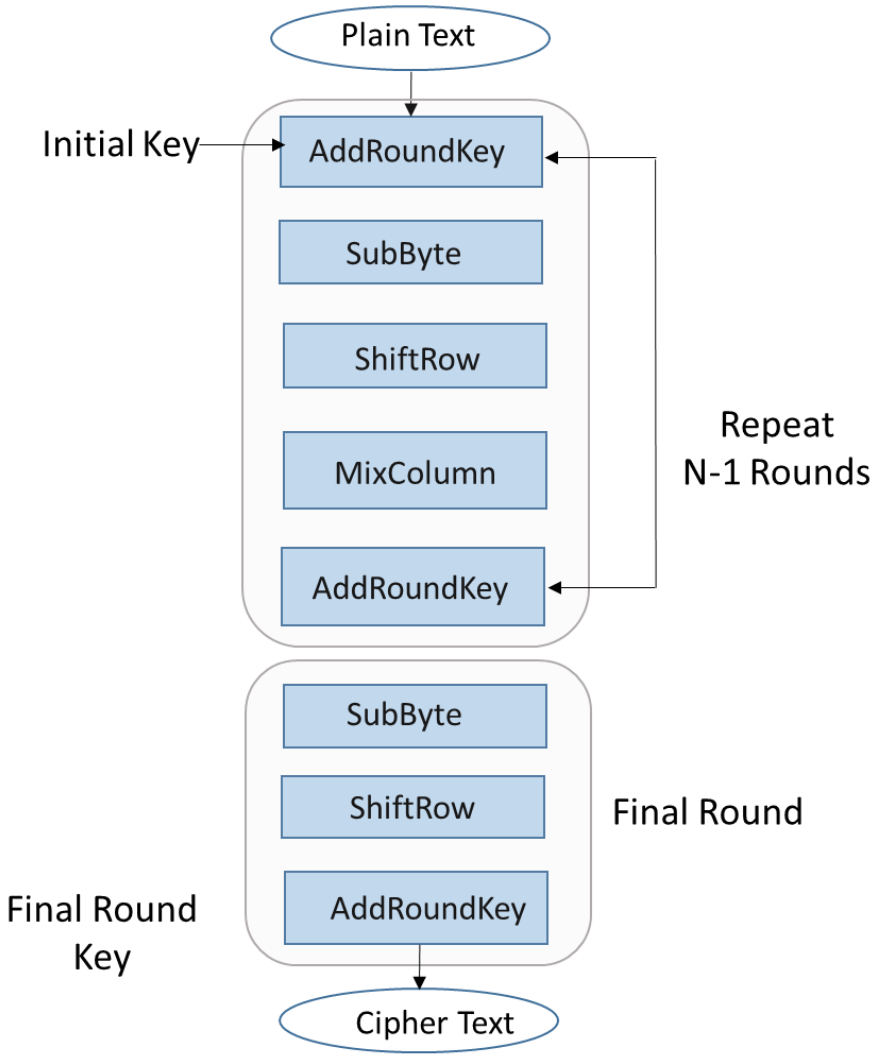


Figure 5.3: Flowchart of AES encryption algorithm

5.2 Related Research

Due to the nature of IoT devices connected over the network, the security of the devices has received considerable attention and there is a growing interest in efficient and secure realization of the hardware devices. As a result, various countermeasures have been proposed to DPA attacks ranging from architectural to system level. All the techniques aim to reduce the direct relation between power consumption and its internal processing of data. The countermeasures available are grouped into two methods; masking and hiding.

In general, masking methods [74]-[76] modify the input data being processed with a random mask in the encryption or decryption process. In general, the power consumption depends on the input data and its computation. Using random mask provides randomness in the power consumption and makes it difficult to reveal the input data being used. At the end, the corresponding mask must be used to recover the original input data. Masking methods can be applied at either algorithmic level or gate level. Most of the research available is based on algorithmic level because of its ease of implementation.

5.2.1 Masking

Data masking uses a random arbitrary value or function to mask the input data or intermediate result. In [65], Kocher proposed an early countermeasure based on this principle to prevent timing attacks. Another technique proposed in [2] to randomize the intermediate values using a mask. As a result, the intermediate value during the encryption process becomes unpredictable by transforming the actual computation into execution involving random values [18].

In general, the intermediate function is done as follows: c = A XOR K. Usage of the data mask modifies the intermediate computation m below:

$$m = rand\ (\ );\ A_m = m \oplus A \tag{5-1}$$

$$c = A_m \oplus K \tag{5-2}$$

Figure 5.4 illustrates the difference between normal operation and data masking of Sbox. In the case of normal operation, the Sbox function is implemented as XORing the input with the secret key. A new addition here is the random mask calculation. The masked input and the secret key

are XORed together to generate the random or masked intermediate data. When the attacker tries to attack such system with the random information instead of the actual one, it does not make the attacks impossible but increases the number of attempts to reveal the secret key.

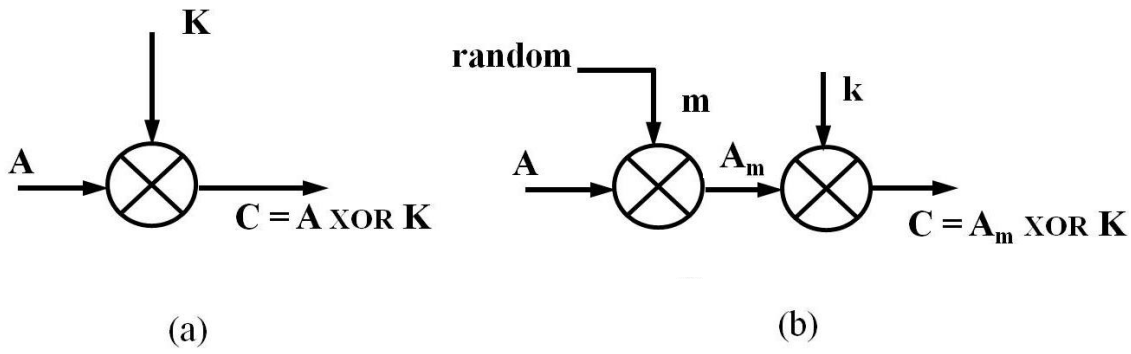


Figure 5.4: Illustration of data masking (a) Normal Sbox operation (b) Sbox with data masking

In [65], a simple mask is added to the plain text instead of adding the masked value to the Sbox function. A similar technique proposed in [77], same mask is used before and after Sbox computation. However, a new random mask is generated for every encryption round.

A technique proposed in [23] claims that it provides an implementation of DES and AES encryption algorithms secure against attacks. The technique masks all the computations throughout the AES algorithm with random values. Usage of additional multiplicative and addictive masks provides randomness but costs 3x runtime and 2x in memory space. To add more randomness into the data, Messerges presented in [78] two complex masks; one based on Boolean logic and the other based on arithmetic logic. The Boolean mask is calculated using bit wise XOR operation of input x and random mask $r_{x,}$ as in Equation (5-3), to generate a masked value x'. In

Equation (5-4), arithmetic mask uses addition or subtraction modulo 2^n, where n in the word length.

$$Boolean\ mask:\ x' = x \oplus r_x \qquad (5\text{-}3)$$

$$Arithmetic\ mask:\ x' = (x + r_x) \bmod 2^n \quad [78] \qquad (5\text{-}4)$$

In [78], all operations except addition and multiplication are masked using Boolean mask, whereas the Arithmetic mask is used for addition and multiplication. Both the masks are used for various encryption algorithms such as Mars, RC6, Rijandel, Serpent and Two Fish. The authors claim that Rijandel algorithm is best suited for masking and overall, data masking is the best solution to prevent DPA attacks.

Another variation to the data masking is the masking of intermediate function, as shown in Figure 5.5. The earlier one uses a random mask with the input data to randomize power consumption. The later one uses a mask to the output of the Sbox function.

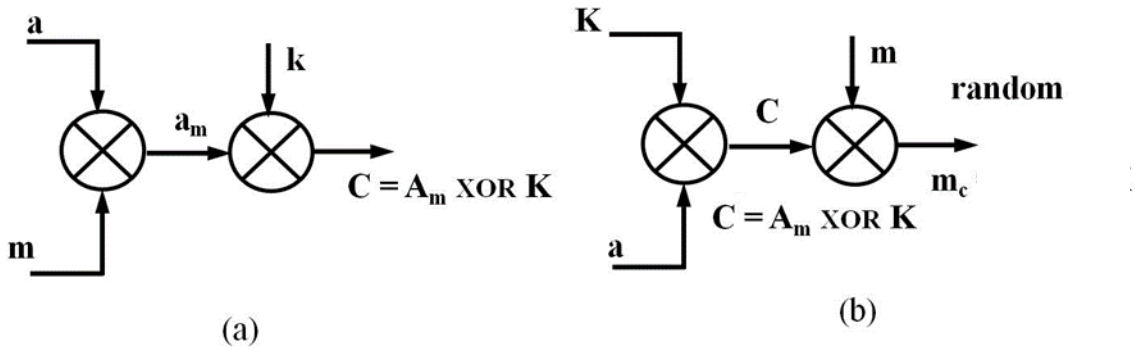


Figure 5.5: Random data masking (a) Before Sbox function (b) After Sbox function

Masking the Sbox provides more unpredictability since its output is used for power analysis attacks. In AES, the Sbox operation can be realized by using look up tables or direct calculation. The look up table approach

receives data and sends the output based on look table value in the computation. Direct calculation involves complex implementation of the function. In [78] proposed another countermeasure based on Sbox masking, which uses an input mask r_{in} and an output mask r_{out}. The input x is masked using r_{in} and fed into the table T, to map the masked value into a different value. The resultant output is masked again using r_{out}, and stored in a new masked table T'. The entire operation of the Sbox masking is shown by Equation (5-5).

$$T'[x] = T\,[x \oplus r_{in}\,] \oplus r_{out} \tag{5-5}$$

However, the drawback of their approach is the memory space requires storing the pre-computed tables and significant amount of time for the computation.

On the other hand, the hiding methods use techniques to make power consumption of the device the same for different operations. In [79], [80], a new logic cell called Wave Dynamical Differential Logic (WDDL) and Sense Amplify based Logic are proposed to make power consumption same for different operations. In order to perform this, the normal regular cell needs to be replaced by this new cell technology to consume the same power for different operations. In [81], another technique is proposed based on switching capacitors. The switching capacitors isolate the power supply and the device by providing power to the device during capacitor discharge time instead of power supply.

However, these countermeasures provide security improvement but involve extra hardware cost and throughput degradation. For example, the WDDL method can increase the security by using 3 times larger silicon area and 75% throughput degradation [2]. The switching capacitor method

can reduce the area overhead by 27%, but the performance is still degraded by 50% [82].

5.3 DPA attacks: An Experimental Study

In this section, the experimental procedure used to realize power analysis attacks on a PIC microcontroller based systems is presented in detail. Before that, the possible setups, which can be used for the attacks are discussed. The results of these experiments provide the foundation for our design concept, which is described later in the coming chapters.

5.4 Practical Implementations of DPA Attacks

In this section, the different methodologies used to realize practical implementations of DPA attacks are demonstrated. Generally, power analysis attacks can be realized in two ways: simulation based and real measurements based. Both the environments can be used to attack any electronic system and to evaluate various countermeasures. The purpose of this section is to prove that the DPA attack involves ease of implementation and it poses a serious problem to the security of the embedded systems.

5.4.1 Simulation based DPA

The procedure for both the setups simulation and real measurements is the same except for the arrangement of the power consumption of the target device.

In simulation based, the power consumption is obtained from a simulation tool instead of real power measurements captured using an oscilloscope. Power estimation tools are used to estimate the power consumption of the device, before even it is fabricated. The countermeasures can also be tested

and evaluated before the actual device is fabricated itself. This is the main advantage of simulation based compared to that of real measurements based. Other advantages include the ease of implementation and noise free measurements from the simulation tools [83]. The simulation based DPA attack methodology is depicted in Figure 5.6. In order to estimate the accurate power measurements, the target design along with a set of plain texts is commonly used. A testbench needs to be created using a secret key, which is same for all the inputs in the set. Power traces are generated and partitioned with respect to its corresponding inputs. To compare the generated power traces with the estimated values, a hypothetical power model generation with the same inputs is essential. In order to estimate the secret key used in the RTL simulation, statistical techniques must be used on both simulated and hypothetical power values. In [83] , explored the simulation based measurements and compared with the real measurements. Modelsim tool is used for RTL and gate level simulation and for the testbench. To estimate accurate power simulations, Nanosim is used.

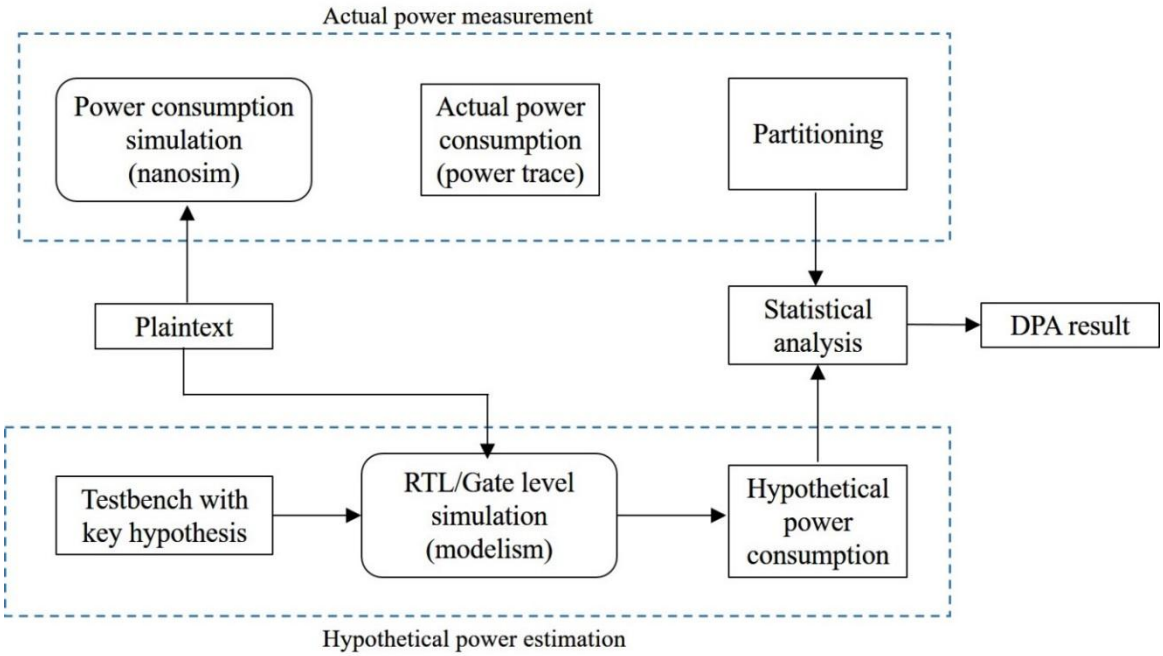


Figure 5.6: Simulation based DPA Attack implementation strategy

5.4.2 Power simulation tools

Generally speaking, there are many power simulation tools available to estimate power consumption of a digital circuit. The selection of tools mainly depends on the accuracy, which must be as close to the actual measurements as possible and the simulation time. In the case of power analysis attacks, the simulation time is the deciding factor because it involves many number of encryption rounds thus consume more time. The available power simulation tools are Hspice[84] Spectre[85], Nanosim[86], Primepower[88], Ultraism[87].

Hspice is an accurate circuit simulator from Synopsys. Also, the tool precisely predicts the timing, functionality and power consumption. Sprectre is a power estimation tool from Cadence. As mentioned in [83] both tools computation time depends on the circuit design size. As the circuit size increases, the simulation time increases.

Primetime, another tool from Synopsys, generates power estimation using gate level simulation. The work presented in [18] used this tool to generate the power simulation traces. The efficiency of the tool depends on the available libraries. Other tools such as Nanosim and ultraism can be used for low level power estimation. In [83] , Nanosim is used to generate accurate power simulations and claimed that the tool provides results very close to the real measurements. However, majority of these tools need accurate details of the design at circuit level and gate level to provide accurate power traces. Considering this fact and the limitations of the available tools, the power analysis attacks are realized in this work using the second method, real measurements based.

5.4.3 Real measurements based methodology

In this section, the real measurements based methodology is presented along with hardware and software design flow, as depicted in Figure 5.7. The software design flow, which is necessary for the statistical analysis to reveal the secret information, is implemented using Matlab software. On the other hand, the hardware flow comprises of coding of AES algorithm [89], downloading it on to a microcontroller and capturing of power traces while executing encryption or decryption

The DPA attack methodology followed in this work is similar to the approach presented in [90], where mixed environment of hardware and software used to replicate the actual realistic attacks. In order to make the attacks successful, a sequence of operations need to be performed.

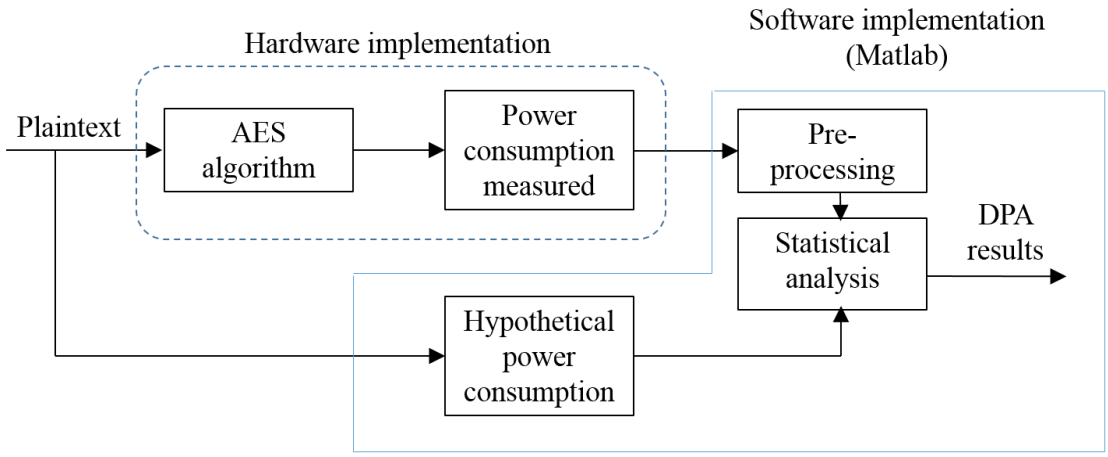


Figure 5.7: Real- measurements based DPA Attack implementation strategy

The first step is identifying the intermediate value of the AES algorithm, which is the AddRoundKey operation. This is the XOR operation of the first byte of the plain text and the first byte of the key. The result of this function is substituted with a non-linear byte value with the use of Sbox. The power analysis was focused on first byte of the secret key. Due to this reason it is necessary to measure power consumption corresponding to the nonlinear operation of SubBytes. The reason for this is simple linear

operations such as AddRoundKey does not leak enough information. In general, AES algorithm encrypts 128 bit input data using 128, 192, 256 key bits. The algorithm is implemented on PIC microcontroller using C language, with 128 bit input and 128 bit key. The detailed implementation of AES algorithm is presented in Appendix B.

The second step is measuring power consumption while this process is executed. Once the setup is established, the oscilloscope was triggered to record the power consumption data and the number of samples recorded depends on the Digital Storage Oscilloscope (DSO) settings. For instance, if the frequency of the microcontroller is 1MHz and the signal is sampled using DSO with a sampling rate of 50Msa/s provides 50 samples of data per cycle. For each input, power consumption is recorded and the same procedure is repeated for different plain texts and each plain text conversion is called an encryption round. Hence the data collected is represented as a two dimension matrix, P [0...99][500], where the first index represents encryption rounds and the second index represents the number of samples.

The third step is estimating the hypothetical power consumption. As shown in Figure 5.7, this phase is implemented in Matlab, whereas the above two steps are related to the hardware. The model used in this case is Hamming Weight model, which is used to predict the power model of the attacked system. Here the quality depends on the knowledge of the attacker about the targeted device. The better the simulation of the attacker matches the actual power consumption characteristics of the device, the more effective the DPA attack is [71]. The attacker expects that the power consumption is directly proportional to the number of non-zero bits in the processed data. The data values that are processed before or after this value are ignored. From this reason, HW model is not suitable for

simulation of the CMOS circuits but the experimental results from practice show that Hamming weight of currently processed data is dependent on power consumption of CMOS circuits and it can be used. The second basic power simulation model is Hamming distance model (HD), in this the attacker expects that the power consumption is directly proportional to the number of changed data values in the processed data. The attacker can map the data which are transmitted via data bus to the value of power consumption without the knowledge of the device netlist. Power consumption, which is caused by a change of the data bus value from v_0 to v_1 is proportional to HD (v_0, v_1) = HW ($v_0 \oplus v_1$). Similarly, it can be applied to other buses such as the address buses and to the power description of registers.

In fourth step, the task is to apply statistical analysis on the measured and hypothetical power traces. The most commonly used methods are correlation analysis [71], difference of means [90] and in this experiment, correlation analysis method is used in for detecting the secret key.

5.4.3.1 Correlation Analysis

The correlation coefficient is one of the best known methods to determine the linear relationship between two random variables. Hence, it is also a suitable method for performing DPA attacks. A very well defined theory exists for the correlation coefficient which can be used to model the static properties of DPA attacks. The correlation coefficient is defined by the covariance as follows:

$$C(T,P) = \frac{E(T,P) - E(T).E(P)}{\sqrt{Var(T).Var(P)}} \quad (5\text{-}6)$$

Here T and P are two variables to be compared and the correlation coefficient reveals the relation between them. It is a dimensionless quantity

and it can only accept values between plus and minus one $-1 \leq \rho \leq 1$. Value '−1' of correlation coefficient denotes indirect dependence that is the change in one group is accompanied by an opposite change in the second group. Value '0' indicates that detectable statistic dependence between values of the two groups does not exist. Whereas, if the correlation coefficient is equal to 1, it indicates a direct dependence or a perfect correlation between the values of the two groups. In correlation analysis, the correlation coefficient is the main important attribute, which reveals the relationship between the two variables.

5.4.4 Experimental Setup for DPA attacks

This section provides the experimental setup used for a successful DPA attack along with the obtained experimental results. It includes the detailed information about the devices used, their typical features and the settings used for the attack.

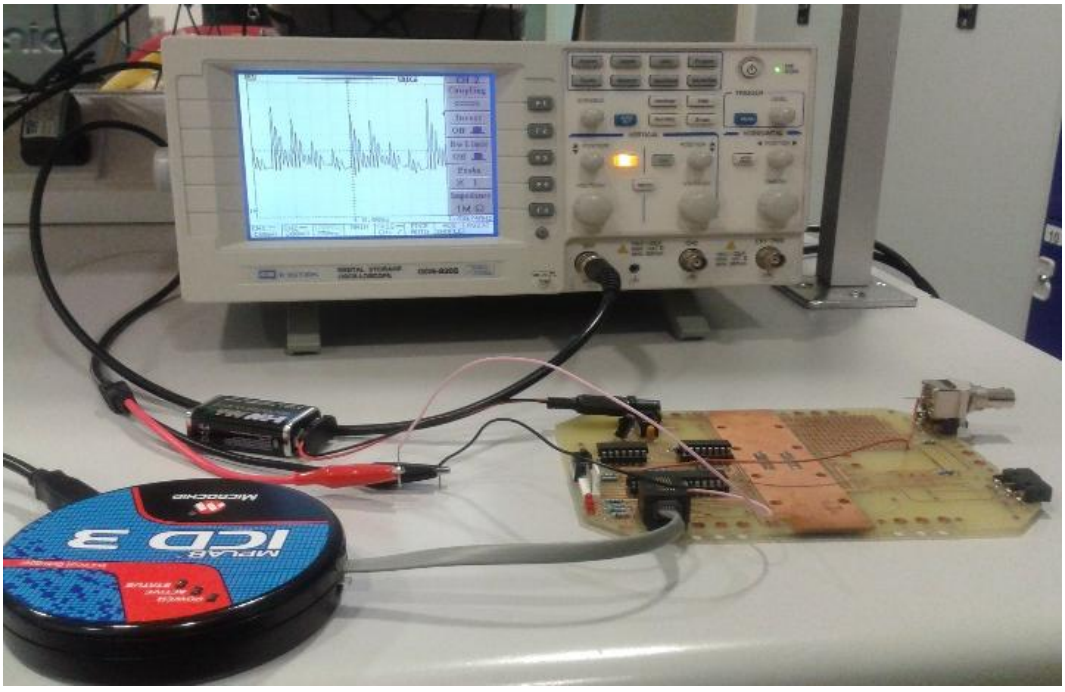


Figure 5.8: Experimental setup

The main components of this setup are a PIC microcontroller [91] based board and Digital Storage Oscilloscope GDS-820S, as shown in Figure 5.8. The typical features of the devices used for the measurement are given in

Table 3. The testbed used for the attacks is a PIC microcontroller based board, which represents a cryptographic module. The details and schematics of the testbed and the basics of microcontroller programming can be found in [92]- [93]. The cryptographic module, AES was developed and downloaded to the microcontroller and verified with test vectors. DPA attack was realized after the verification of the algorithm. The input plain text data block was sent to the microcontroller from the computer. There it was encrypted and sent back to the computer. The data interface between microcontroller and computer was achieved using Microchip In-Circuit Debugger, MPLAB ICD3.

The selection of the measurement procedure depends on the available resources and requirements. In this case, a resistor method, with a small resistor of 10 Ohms is connected between the power supply and the board to measure power consumption across the device. Due to the insertion of the resistor, the instantaneous current consumed by the system can be measured and this current is a measure of the power consumed. The next communication interface is from the board to the oscilloscope.

Table 3: Details of equipment used

Devices Used	Typical Features		
Microcontroller PIC18F2420	Program Memory	Flash	16K Bytes
		Single word instructions	8192
	Data Memory	SRAM	768
		EPROM	256
	I/O	25	
	In-circuit Debugger	MPLAB ICD3	
	CPU Frequency	40 MHz	
Oscilloscope	Bandwidth	150 MHz	
	Memory	125K points long memory	
	Sampling Rate	100 Msa/s	
	Interfaces	USB, GPIB, RS-232, Printer port	
Computer	Processor	Intel core - i53210M	
	Speed	2.5 GHz	
	Memory	8 GB	
RS 232 Interface	Baudrate	9600	
	Terminal	DTE - DTE	
	Connector	DB9	

GDS-820s has different interfaces such as RS-232, USB and Parallel Interface; however, USB interface is not suitable for transferring the captured data to the computer. Out of the other two available interfaces,

RS-232 interface is used in this work because speed is not the paramount importance in these attacks. The FreeCapture [94] software captures data from the oscilloscope to the PC using RS- 232 Interface. The configuration in the software and the oscilloscope must be the same for a successful communication. But the software used may vary with the scope. The data is stored in the computer in csv file format and this file is used in Matlab for further analysis.

The attack is realized using the general procedure described earlier and the results of the Correlation analysis are shown in Figure 5.9 and Figure 5.10. The highest peak in the figure 5.10 shows the correlation coefficient for that key value. This information provides the attacker the first byte of the secret key and the same process must be repeated to reveal the entire key.

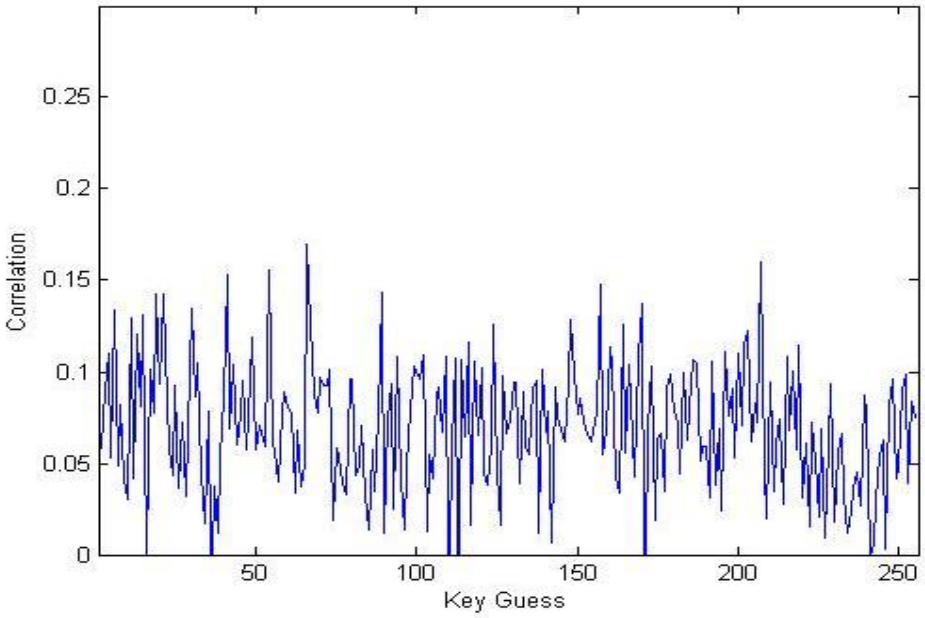


Figure 5.9: DPA Attack with a wrong key guess

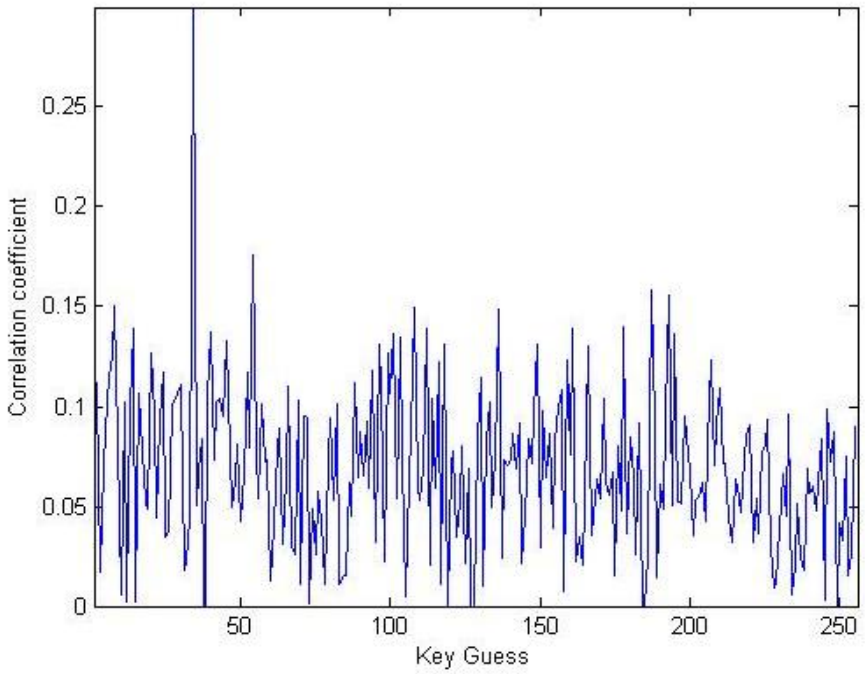


Figure 5.10: DPA Attack using Correlation Analysis for key = 35

5.5 Summary

This chapter provides an overview of Differential Power Analysis attack and the practical realization of the attack. DPA attacks are most powerful in the presence of noise and countermeasures. Understanding the concepts and subsequent realization of a successful attack is challenging. In this chapter, the DPA attack with a detailed description of steps along with the full information about the experimental setup, device features and their communication for better understanding of the whole process is presented. The AES algorithm is implemented on microcontroller and the attack was focused only on the SubBytes. The power traces are captured using the setup and the analysis is carried out using Matlab. In the next chapter, a novel technique based on aggressive mechanisms to overcome the drawbacks of the available research is presented.

CHAPTER 6

A Novel Time-Borrowing Technique for Aggressive Voltage/Frequency Scaling

Low power consumption along with better performance has become an important aspect of processor and system design. Many techniques ranging from architectural to system level are available. Among all methods, voltage/frequency scaling methods are the most effective to achieve low power consumption as dynamic power is proportional to frequency and square of supply voltage. The key idea of aggressive voltage scaling is to adjust the supply voltage to the lowest level possible to achieve maximum power benefit, while ensuring reliable operation. Similarly, aggressive frequency scaling is to alter the operating frequency to achieve maximum performance improvement. In this chapter, an aggressive voltage/frequency scaling technique using time-borrowing feature is presented. The proposed technique allows the system to go

beyond the worst case estimates of the operating voltage/frequency based on timing error to improve performance or to reduce power consumption of the systems. The technique uses the flip-flop and latch combination to double sample the input data to detect any timing violations as frequency/voltage is scaled. The detected violations are masked by phase delaying the flip-flop clock to capture the late arrival data because of dynamic variations. This makes the system timing error tolerant without error correction timing penalty. The technique is validated by prototyping on to an FPGA in Spartan 3E FPGA development board using two stage arithmetic pipeline. In addition to this, the technique is implemented in various benchmarks to evaluate the performance improvement.

Due to the increased use of mobile and portable devices, power consumption has become a major constraint in the design of modern processors. Many techniques ranging from architectural to system level have been developed. Of all these, dynamic voltage and frequency scaling (DVFS) has received more attention over the last two decades. This technique dynamically adjusts voltage and frequency depending on the present task requirements, thereby reduces the power consumption of the system. The technique has been proven ideal to trade off performance and power consumption of a processor because supply voltage reduction is the ideal way of achieving low power consumption. Due to its benefits, DVFS has been used not only to reduce power and temperature but also to improve security of the systems from power analysis attacks. The traditional DVFS approach scales the supply voltage based on either workload or timing margins available [95] . To guarantee safe operation, voltage and frequency pairs are pre-determined along with workload conditions by considering all process corners. The hardware realization of this open loop system uses Look Up Tables (LUTs) to store voltage and

frequency pairs. Since the LUT is pre-loaded with values, the system is not able to adapt to process variations or environmental conditions. The basic DVFS technique has lost its demand as process technology advances due to the increasing Process, Voltage and Temperature (PVT) variations.

On the other hand, timing margin based DVFS is a closed loop system, which adaptively changes supply voltages to a minimum level to meet performance requirements by considering process, voltage and temperature variations. The typical closed loop system employs a monitoring algorithm to provide application requirements in order to configure the programmable DC-DC converter and the programmable clock generator. In recent years, this technique uses critical path monitors to monitor critical path delay in a process and adjusts the frequency/voltage so as to minimize timing margins during run time.

The direct monitors track the actual critical path delay during run time for detecting and correcting timing errors. This technique is more effective in reducing all the timing margins. However, this approach detects errors after their occurrence and requires large timing penalty for error correction.

The indirect monitors use critical path replicas (CPR) that reflect the delay behaviour of the actual critical paths in the design. The design of these CPRs is most critical and it must be as close to the actual delay of the circuit as possible because these are used to generate the feedback signal controlling voltage or frequency of the processor. This technique requires large timing margin compared to the direct technique because of the difference between the actual and predicted paths.

The time-borrowing monitors mask timing error by borrowing time, either delaying the arrival time of the correct data to the next pipeline stage or

delaying the clock to capture the late arrival data. This technique needs large checking window to be more effective. The checking window is referred as the period of time after the clock edge reserved for error detection and masking.

To summarize, the main challenges posed by aggressive voltage/frequency scaling are :- i) reducing large timing penalty for error correction and ii) reducing large timing margins and iii) having large checking window for error detection and correction. All these issues can be dealt by using a suitable critical path monitor. Hence, for the AVS system to have better power savings, it must have a suitable critical path monitor.

6.1 Difference between Latch and Flip-flop

Before diving into the discussion of the proposed technique based on aggressive scaling, it is appropriate to first review the fundamentals of sequential elements and the approach used to develop digital ICs. The two basic sequential elements used in digital logic to store information are flip-flop and latch. Both these elements are used to store single bit of information. The fundamental difference between these two is; latch is a level sensitive device and flip-flop is an edge sensitive device. In other words, latch changes its output based on the input changes as long as the enable signal is asserted. On the other hand, flip-flops change its output as the input changes either at the rising or falling edge of the enable signal.

Latch

```
process (en, d)
begin
{
if  (en = '1') then
            Q <= d
end if
}

end process;
```

FF

```
process (en)
Begin
{
if (en' event) and (en = '1') then
            Q <= d
end if
}

end process;
```

Figure 6.1: VHDL code snippets for latch and flip-flop

Figure 6.2 shows the difference between latch and flip-flop using timing waveforms. The corresponding VHDL code for both the elements is shown in Figure 6.1. Here 'Q', 'd' are the input and output of the elements and 'en' is the enable signal and decides the operation of the blocks. In the case of latch, the output Q continuously checks its input d and correspondingly changes when the enable signal is high. The same operation can be completed when the enable signal is low. Therefore, Latch is called level sensitive device.

In the case of flip-flop, the output Q captures the input data with respect to the positive edge of the enable signal. Using negative edge also, the same operation can be achieved. Hence, flip-flop is called edge sensitive device. The fundamental difference between these two elements is used to capture the late arrival of data in the proposed architecture.

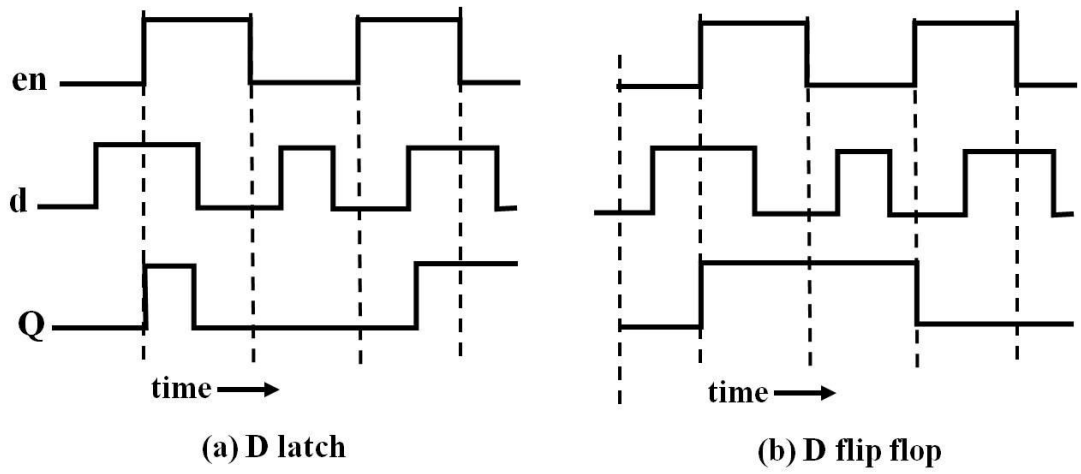


Figure 6.2: Timing waveforms depicting behaviour of latch and flip-flop

6.2 Digital Design Flow

A digital design goes through a top-down approach to transform the original set of specifications to an end product. During its development, each step corresponds to a detailed description of the system, which has its own specific semantics and set of primitives to develop the end product, digital design flow can be used as application ASIC or SOC or FPGAS. ASIC or SOC as the name itself, the products are designed for a specific purpose. Hence the design flow for these devices varies keeping the application requirements through all the stages. In contrast to this, FPGA are field programmable, they work as intended by the designer. In other words ASIC's are designed and manufactured keeping the application in mind throughout the process where FPGA's are manufactured in advance and design is implemented later based on the application. In this section, a high level overview of the general digital design is discussed.

Figure 6.3 presents the typical stages involved in the design flow. It represents a top-down approach from specifications to final product. The development of a product with the most typical design requirements

power, area, cost and functionality involve many iterations through various stages of the design flow. In general, the initial requirements are contained in a document with details of the final product functionality and a set of constraints that it must satisfy. In their scenario, functional design is the process of converting the specifications and constraints into a feasible solution. This involves tasks such as developing micro-architectural model for the entire design solution and software-hardware co-design.

Due to its hierarchical nature of the design flow, each stage depends not only on the present state but also on the previous states. Hence, care must be taken at all stages of the design process. However, the functional stage is the primary stage of the flow and it also involves the basic model development for the entire problem, the design must spend quality time and

The design architecture consists of different distinct modules, each of which contributes a specific functionality to the overall design. Each module input and output posts and communication protocol required to establish a link between the modules also defined here. The outcomes of this phase is a constructive model of the design with integration of the later modules developed in the following stages.

Further to the function design is the register transfer level stage. The architecture of the design developed in the earlier stage is more defined using hardware description languages (HDL). The individual design elements such as memory and functional blocks are designed during this phase. In addition to this, the decision the clocking system to meet the design contracts such as performance and power is done.

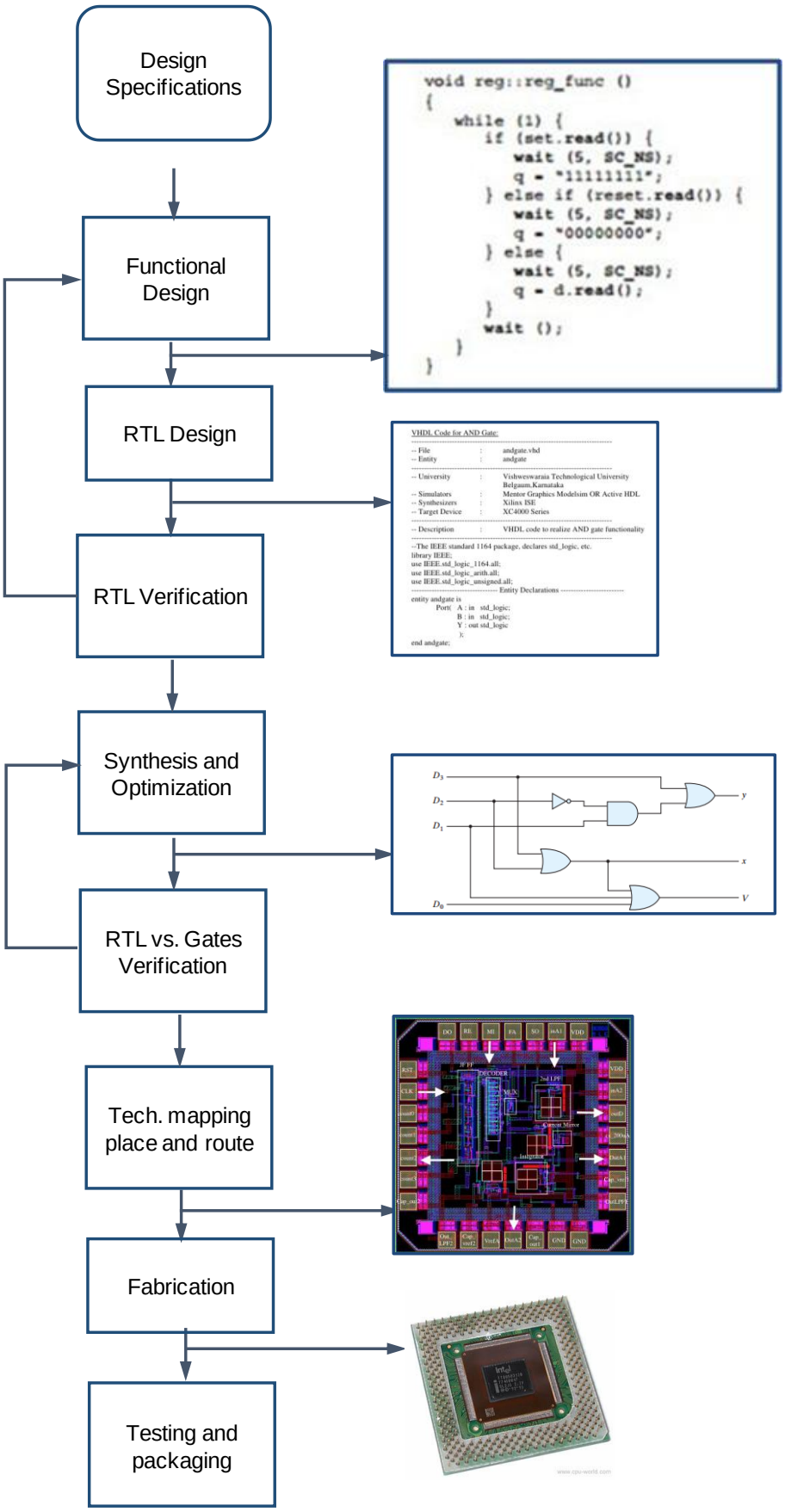


Figure 6.3: Flowchart for digital system design

The RTL verification stage is involved as soon as the design phase completed. This stage verifies the functionality of the developed circuit design under the assumption that no manufacturing errors are present. The basic idea here is that the design must be verified functionally, virtually error free before proceeding further to the manufacturing. In case of errors, the iteration process must be initiated till the design becomes error free. During this phase, various techniques and test benches are developed to verify that the developed model meets its initial functional requirements. When this is not the case, the design flow has to be moved to the initial function design stage to modify the model and accordingly RTL design needs to be updated.

The synthesis and optimization of the RTL design phase involves generation of a detailed model of circuit and gates, which is optimised based on the design constraints. In the case of power optimised design, the model needs to be considered as a constraint at their stage. Similarly, area and performance can also be applied. The outcome of this stage provides a model, which is a combination of different gates, interconnect logic and memories. The resultant model is optimized for the design constraints and their procedure involves number of iterations before it provides a suitable solution to meet the requirements.

Up to this stage, the procedure is same for all digital devices. In the case of FPGA, the target FPGA needs to be selected and then place and route places all the internal cells and connectivity between them. Generally, the design flow for FPGA is done using FPGA vendor tools such as Xilinx IDE and Altera Quartus. The tools provide a back annotate file, which can be used in timing analysis and a bit stream file, which can be used to program the FPGA. The next stage in the flow is the RTL and gates verification. This stage is to verify the earlier developed RTL model functionality and the

later developed gate-level model. The pre-synthesis and post-synthesis models are verified for functional correctness. The final three stages of the design flow involve mapping the design into a geometrical layout for fabrication. In the end, the fabricated chip is tested and packaged.

6.3 Related Work

The first classification of the available critical path monitors is direct monitors. The best example for this category is Razor technique [31]. This technique detects timing error by double sampling the critical path output at different points in time and compares them with each other. Some of the benefits of this technique are highlighted in [31], however, it poses significant challenges. The technique monitors critical path signals for late arrival data, detects error after the system is corrupted and stalls the current execution to recover from the error. Furthermore, Razor circuits are susceptible to datapath metastability, requiring substantial design overhead [96].

TEAtime [55] is the best example for the next category, indirect monitors. This method uses critical path replica to estimate the timing error before it actually happens. This is achieved by replicating the most critical paths to estimate the behaviour of the path with respect to voltage/frequency scaling. The advantages of this method are 1) errors are detected before they actually happen; 2) no need for architecture modification of processor designs. But the major drawback of indirect monitoring is how well the replica resembles the actual critical path.

The third classification is based on time borrowing feature. Many techniques have been proposed related to this, including Transition Detector with Time Borrowing (TDTB) [97], Double Sampling with Time

Borrowing (DSTB) [97] and Timber [57]. In DSTB, the position of flip-flop and latch are interchanged to eliminate datapath metastability but the checking window for timing speculation is small compared to Razor flip-flop. Timber proposed a new time borrowing scheme, which detects late arrival of data and masks by reducing the arrival time of the correct data to the next pipeline stage [96]. The drawback of Timber is the small checking window for timing speculation. In conventional systems, closed loop is used to change the delay of the flip-flop clock based on timing error on the fly. This will consume more power as the phase of the clock is changing on the fly. Moreover, clock signal and clock tree are the most power consuming elements of a digital processor.

Table 4 Comparison of present technique to other latest techniques

	Indirect (CPR)	Razor	TDTB	Timber	DSTB	Present Technique
Architecture Modification	No	Yes	Yes	Yes	Yes	Less, no need to replace all registers in the pipeline
Datapath Metastability	No	Yes	No	Yes	No	No
Error Correction	No	Yes	No	No	No	No
Area Overhead	High	CPM + Error Correction Circuitry	Medium	High	Medium	Small
Design Complexity	High	Low	High	High	Low	Low
Timing Margin	Large	Small	Small	Small	Small	Small
Power Overhead	CPM	CPM + Additional Pipes + Recomputing + Buffers	CPM	CPM	CPM	CPM

6.4 Principle of Operation

The aim of the proposed technique is to operate systems beyond worst case estimates by avoiding the excessive error correction overhead and timing margins. To reduce excess timing penalty of error correction overhead, the following timing monitor unit using time borrowing approach is presented.

Figure 6.4 shows a gate-level circuit diagram of the proposed unit. The fundamental idea of the present technique is to delay the main clock in such a way that the late arrival data can be captured with time borrowing. The proposed circuit mainly consists of the following elements: flip-flop, latch, multiplexed clock signal and enable signal. The combination of flip-flop and latch performs double sampling to identify when there is any timing violation as in Razor circuit. However in this circuit, as in DSTB, the position of flip-flop and latch are interchanged to avail the feature of time borrowing and to reduce sequence overhead. But the drawback of DSTB is having small checking window for timing speculation.

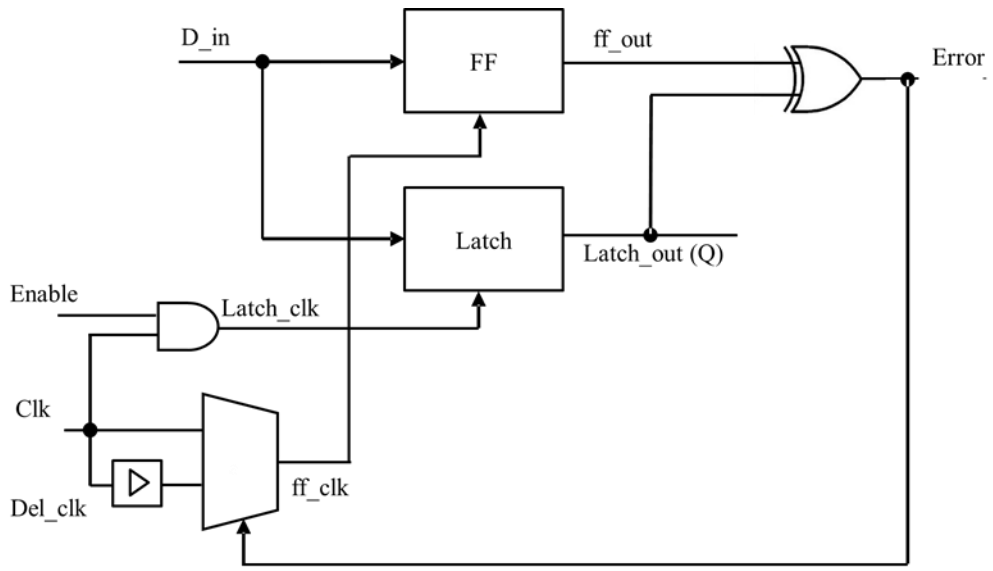


Figure 6.4 Gate-level diagram of proposed technique

The present technique overcomes this drawback by adding a multiplexed clock signal to further extend the checking window. Having larger checking window helps to detect timing error if the data arrives late, further the rising edge of the clock signal [98]. Another added feature is enable signal, which helps to activate/deactivate the time borrowing feature. The proposed circuit detects timing error by double sampling the data using flip-flop and latch combination. When data arrives late, the output of flip-flop and latch differ, and thereby activates the error signal, which in turn changes the phase of the clock signal to the flip-flop. Now flip-flop clock is delayed by 0.5 times the latch clock. Flip-flop can receive the late arrival data and eliminate the timing error. If the error continues beyond this point, further action needs to be taken to change voltage or frequency. To reduce the sequential clock energy overhead, both main and delayed clock signals are generated beforehand and only the selection is done on the fly.

6.5 Timing Waveforms

Figure 6.5 shows the timing waveforms of the present technique. The main clock and the input data signals are clk and d_in respectively. The same input data d_in is fed to both the sequential elements flip-flop and latch.

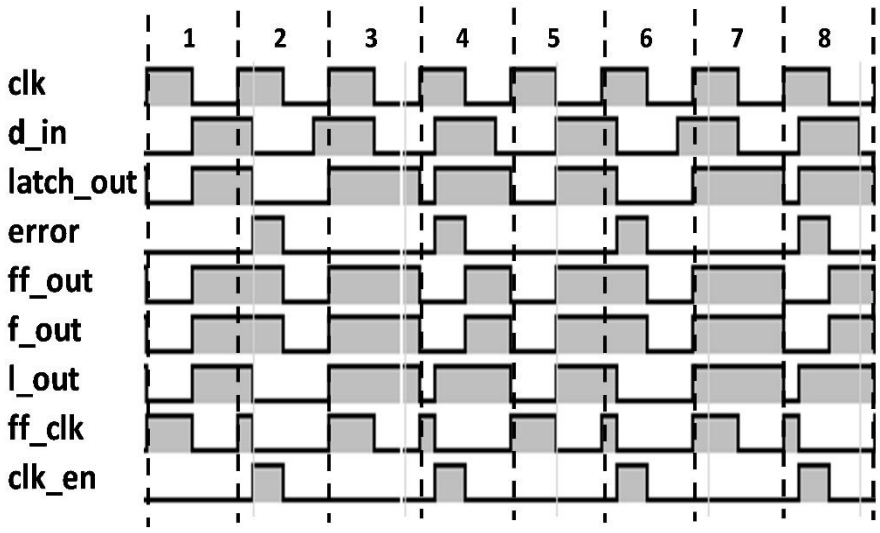


Figure 6.5 Timing Diagram illustrating the behaviour of the present technique in the presence of errors

During cycle 2, d in arrives before setup time and rising edge of the flip-flop. This is the ideal case of operation for any digital circuit. The input data is captured correctly by both flip-flop and latch. Hence, there is no error signal during cycle 3.

During cycle 4, d in arrives after rising edge of the main clock signal, clk. As frequency is scaled beyond the worst case estimate, the input data arrives late [99]. This means that the clock frequency is not sufficient for the critical signal to travel along the path. So flip-flop missed the data, causing the error signal to be activated. When it happens, the multiplexer selects the phase delayed clock and flip-flop uses this to capture the late arrival data. The error signal is masked immediately as soon as both flip-flop and latch capture the same data. Since the error is masked, the present technique resumes the normal pipeline operation.

6.6 Timing Constraints

To ensure correct functionality, the input data must arrive a setup time prior to the rising edge of the clock. Sometimes the data can arrive after the rising edge due to the worst case dynamic variation. The proposed technique can be used to ensure the correct functionality. The two main elements in the technique are flip-flop and latch; both elements are clocked by different clock signals ff clk, latch clk. Both the clocks have the same frequency but ff clk is phase delayed by 180 degrees with respect to the latch clk. Timing constraints must be analysed thoroughly to safeguard any digital circuit.

The main basis for the analysis is taken from [100] for DSTB circuit; modified further to analyse the benefits of the present technique. The timing constraints for the present technique are given below;

$t_{(su, F)}$ = Setup time of the flip-flop; $t_{(su, L)}$ = Setup time of the latch

t_{clk} = clock period; t_{ch} = checking window

t_{pd} = propagation delay

$$t_{ch} = t_{clk} + (t_{(su, F)} - t_{(su, L)}) \tag{6-1}$$

Equation (6-1) ensures that the proposed circuit has a checking window, which detects late arriving data. In general, the input data to be captured properly by the flip-flop, it must stabilize at least $t_{(su, F)}$ before the rising edge of clock. Because of the variations, there is a possibility for the late arrival of data. The circuit must have a large checking window to ensure the late arrival data can be sampled by the latch if flip-flop missed it. The checking window should be as wide as possible to eliminate large guard bands. In our case, the clock to the flip-flop is delayed by 180 degrees to

capture the late arrival data, which is indicated by an error signal. Therefore Equation (6-1) can be modified as mentioned below.

$$t_{ch} = t_{clk} + (t_{(su,F)} - t_{(su,L)}) - t_d$$

$$= t_{clk} + (t_{(su,F)} - t_{(su,L)}) - t_{clk}/2$$

$$= t_{clk}/2 + (t_{(su,F)} - t_{(su,L)}) \quad (6\text{-}2)$$

From Equation (6-2), it is evident that the checking window is almost 50% of the clock period which is wide enough to eliminate the excess timing margins.

Another important metric in digital circuits is the maximum propagation delay, which is the maximum delay of the circuit under worst case conditions as in Equation (6-3). This delay helps to hide the flip-flop set up time from the critical path.

$$t_{pd} \leq t_{clk} + t_{pcql} - t_{(su,F)} - t_d$$

$$\leq t_{clk} + (t_{(su,F)} - t_{(su,L)}) - t_{clk}/2$$

$$\leq t_{clk}/2 - t_{pcql} - (t_{(su,F)} \quad (6\text{-}3)$$

6.7 Extension to Processor Architecture

Figure 6.6 shows the integration of the present technique into the general five stage pipeline architecture. Unlike the techniques in [31] [55], all the pipeline registers need not be replaced with the new circuit. Only the critical paths that get negative slack because of the variations need to be replaced, however, the worst case situation is shown in the Figure 6.6.

Note that the paths that require time borrowing need to be replaced with the present circuit.

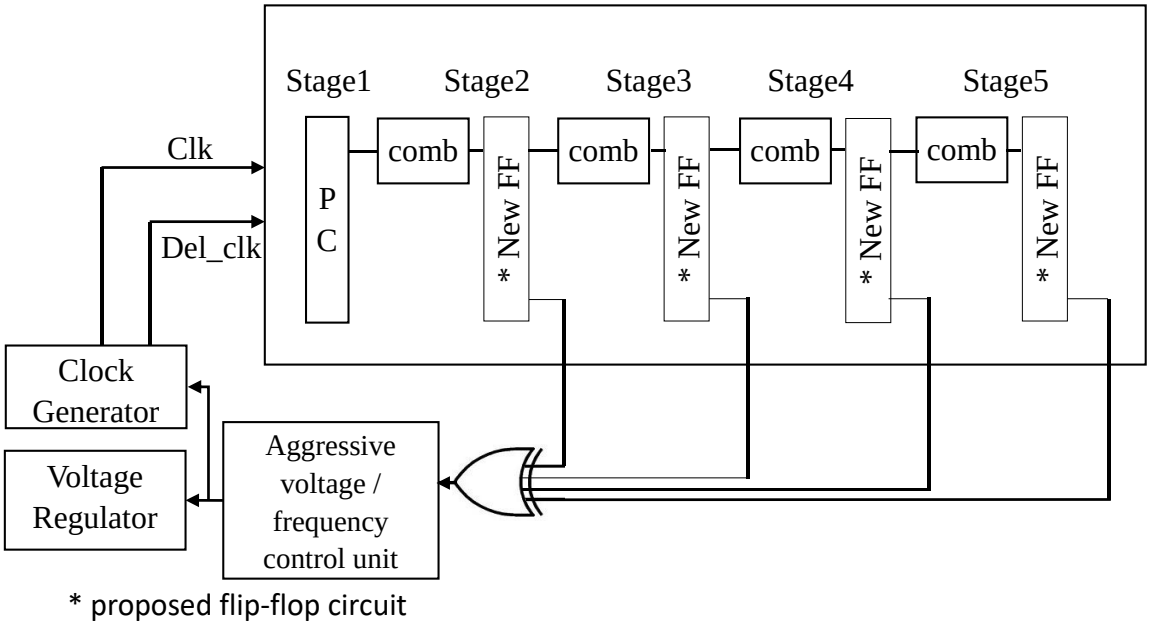


Figure 6.6: Extension to general processor architecture

The pipeline executes instructions normally when there is no error at any stage of the pipeline. If the operating frequency scaled beyond the worst case estimate, timing errors start to occur. Error signal is activated to indicate the occurrence of timing error at each stage of the pipeline. All these error signals are combined to generate a global error signal. Timing errors can be masked by phase delaying the clock to capture the late arrival data. The activation of global error signal indicates the worst case scenario of the circuit. This situation shows that the present operating frequency/voltage need to be altered to perform normal operation. The control unit makes the decision and commands clock/voltage generator to supply a new set of operating frequency/voltage to recover from the error.

6.7.1 Simulation and evaluation

In order to verify the effectiveness of our proposed technique, 90 nm Spartan 3E FPGA development board has been used. The reason for choosing the FPGA environment is the flexibility of FPGA architecture to modify circuit post-placements to insert the present technique to monitor critical paths post-compilation. Such opportunities are not available in ASICs without significant incremental cost [101]. The technique is validated by implementing in a simple application circuit, two stage pipeline counter and targeted to FPGA. The implementation of this design is presented with DCM in Appendix C. The Spartan 3E FPGA development board is preferred because of two reasons 1) ease of frequency tuning and 2) no need to alter the hardware for evaluating the proposed design. Either frequency or voltage can be varied to verify any timing error estimation, detection and correction circuitry. In this case, frequency is varied to verify the effectiveness of the circuit. The whole idea here is to evaluate the circuit behaviour with respect to the timing error and not the source of the error. The timing error can occur by either changing voltage or frequency beyond the worst case estimate. In this experiment, frequency is varied by monitoring the timing error of the critical paths.

The flip-flop clock is delayed by using Digital Clock Manager (DCM) and latch clock is set to the base clock frequency. Since the clocks have the same frequency, the base clock frequency is set as the main clock frequency and the DCM is used to provide the required phase difference to the flip-flop clock.

The present technique in a two stage arithmetic pipeline has been implemented. The first stage of pipeline consists of an adder_subtractor module with 16 bit output. The second stage of pipeline uses 16 input XOR

tree to output the synchronised counter output. The design is simulated, synthesized and targeted to Xilinx Spartan 3E FPGA. The potential critical paths are identified with the static timing analysis. For these critical paths, the original flips-flops in the netlist are replaced with our present technique.

6.7.2 Experimental methodology

In order to evaluate the effectiveness of the proposed unit, a control scheme presented in [58], is used to switch clock frequency between the worst case clock frequency F_{min} and the possible over clock frequency F_{max}. In this case, F_{min} is the estimated worst clock frequency by the timing analysis.

The phase delay between flip-flop and latch clocks are 0.5 times F_{min}. Because of this delay, the late arrival of data detection can be extended from the rising edge to the negative edge of the main clock signal. This allows the circuit to go beyond the maximum clock frequency. The maximum clock frequency can be extended to 1.5 times the original/ estimated worst case clock frequency. To evaluate the performance of the proposed circuit, three scenarios, as mentioned below are monitored.

a) Normal operation: Chosen a clock frequency below the worst case clock frequency. From the static timing analysis, the worst case clock frequency in this case is 120 MHz. The above mentioned two stage arithmetic is run for 10,000 cycles to count the number of errors occurred. No errors were found in this case and the circuit executes normal operation.

b) Over clocking: Chosen a clock frequency between worst case clock frequency F_{min} and maximum frequency F_{max}. The same procedure as in first case is repeated here, to monitor the number of errors that occurred during 10,000 cycles. Now few errors are occurred, which are masked by the time borrowing feature of the circuit.

c) Aggressive over clocking: Chosen a clock frequency just below the maximum frequency F_{max} with the same above procedure repeated for 10,000 cycles. These errors are masked by using this proposed circuit, without even error correction timing penalty. Thus this circuit permits aggressive over clocking beyond worst case clock frequency without any performance penalties.

6.7.3 Experimental results

The above procedure is used to evaluate the performance increase of the present technique as a function of error rate. Figure 6.7 shows the relation between error rate and frequency of the two stage design with and without present technique. In general, the occurrence of timing errors before the worst case clock frequency that is estimated by the CAD timing analysis is very rare. The same can be seen in the figure. The blue line corresponds to the design without the present technique, no errors occurred up to about 124 MHz. As the operating frequency is increased beyond this point, the percentage of error rate started increasing.

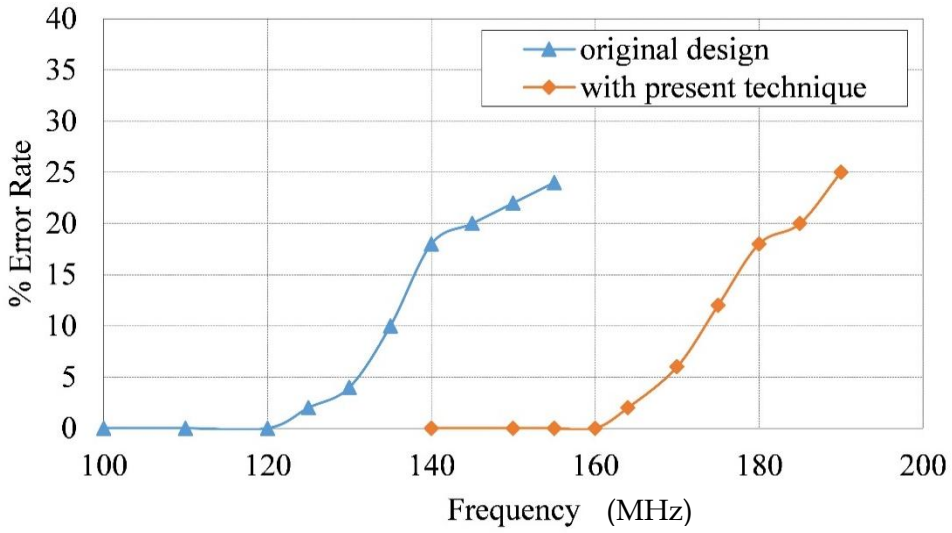


Figure 6.7: Error rate vs. frequency for present technique

The same procedure is repeated using two stage arithmetic with the present technique. No errors occurred until the frequency has reached up to 160 MHz. The timing errors occurred as the frequency tuned beyond worst case frequency 120 MHz. However, these errors are masked by the technique, causing performance improvement. The percentage of error rate started increasing after 160 MHz. This indicates that critical path timing violations exceed the maximum time borrowing allowed with the present technique. The two stage design with Razor technique is simulated to evaluate the performance improvement. Figure 6.8 shows the results of Razor technique and present technique. As seen in the figure, the present technique provides little less performance improvement compared to Razor technique and this is due to the fixed phase delay of the clock signal. This limitation can be improved by using tuneable phase delay element to capture late arrival data and to mask timing errors. However, this has an impact on area overhead and power overhead.

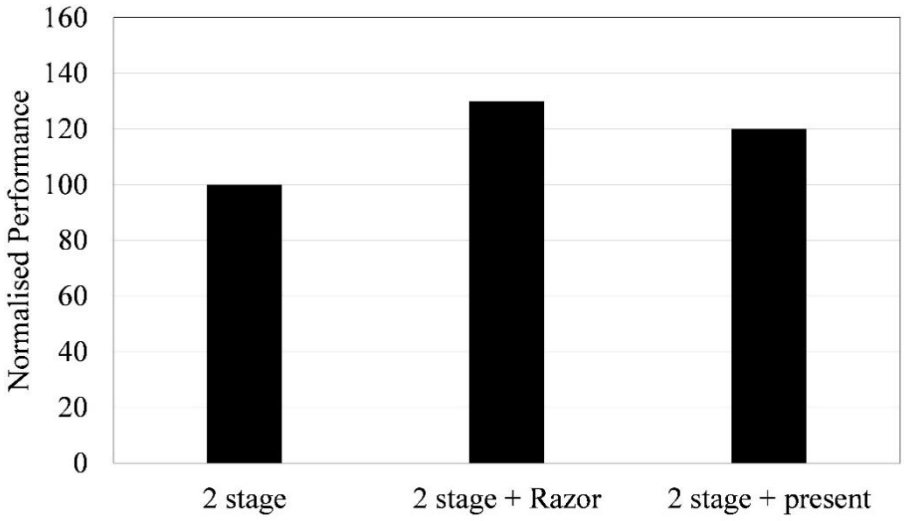


Figure 6.8: Comparison of present technique performance improvement to the original design and Razor technique.

In this work, simulated different benchmarks to evaluate the performance improvement of the present technique. The results of the three benchmarks are presented in Figure 6.9. From the results, it is evident that the present technique provides approximately 20% performance improvement compared with the original design.

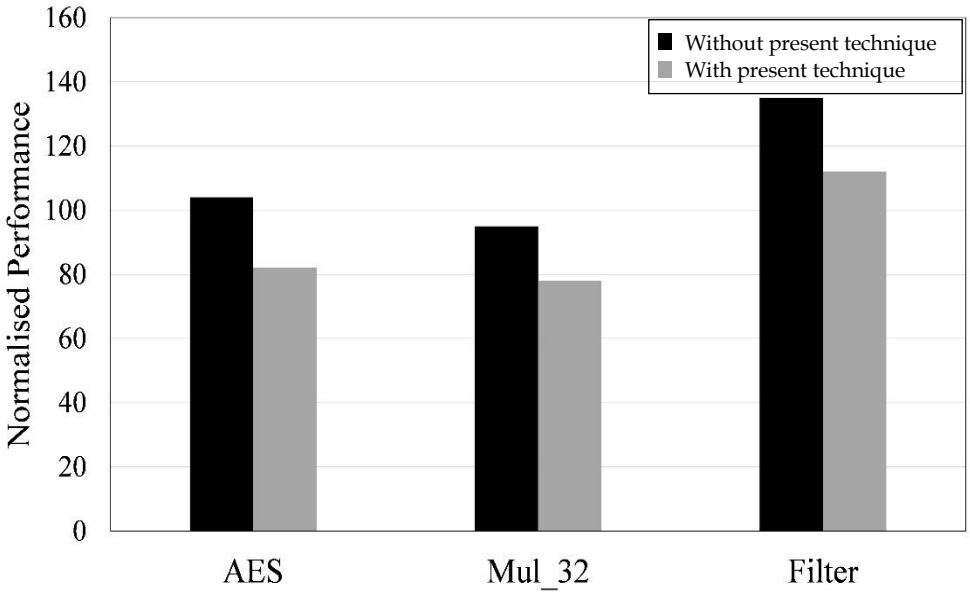


Figure 6.9: Performance improvement of various benchmarks using present technique

Also evaluated the present technique power overhead involved in dealing with error detection and correction. The original two stage design is run with voltage fixed to 1.0v and frequency is varied to measure total power consumption of the design with Razor and present technique. One frequency is chosen below the worst case frequency estimated by the timing analysis and the other frequency is selected beyond the worst case clock frequency. The idea here is to evaluate the power consumption of the present technique when there are no timing errors and in the presence of timing errors. Results of both techniques, power consumption vs. frequency are presented in Figure 6.10. In both cases Razor technique consumes more power compared to the present technique. Though Razor technique provides little better performance improvements compared to the present technique, Razor consumes more power before and after timing error occurrences. The present technique consumes less power, providing approximately 10% more power savings than the original design

in the case of frequency less than the worst-case estimate. The savings are improved by 3 times by using the proposed techniques in the case of frequency after the worst-case estimate.

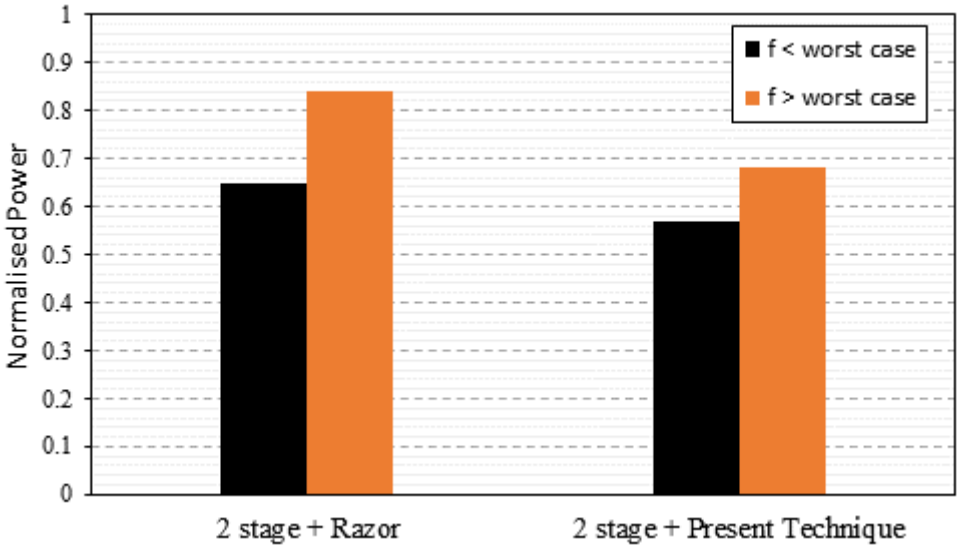


Figure 6.10: Power consumption of Razor and present technique

6.8 Conclusion

In this chapter, a novel technique is presented to reduce the excessive timing margins, thus to reduce the power consumption of embedded systems. The present technique corrects the timing error caused by aggressive voltage/frequency scaling by phase delaying the clock. The main benefits of the technique are less area, power and design overheads compared with other latest techniques. The technique is validated by prototyping on to an FPGA in a Spartan 3E FPGA development board.

Results are presented to indicate the performance improvements and power consumption of the present technique.

CHAPTER 7

Proposed Countermeasure Based on Aggressive Scaling for Security against DPA Attacks

Differential power analysis attacks [19] are the main concern to embedded systems' security. Even without knowing much information about the system, the attacks can be successful. The process is initiated by measuring the power consumption followed by estimating the hypothetical power consumption of the device. The next step involves comparison of the measured and estimated power models to extract the secret information contained in a chip or system. This chapter provides the practical realization of such attacks on the Advanced Encryption Standard (AES) implemented on an FPGA based system. Also includes the experimental setup used, statistical analysis of the captured power traces to reveal the secret key information. The implementation of the proposed

countermeasure along with AES encryption and comparison of results with and without countermeasure are also presented.

The experimental investigation was carried out to perform power analysis attacks and their prototype on a PIC based system was discussed in chapter 4. In general, a set of worst- case operating conditions are outlined for digital ICs during their fabrication, under which the designed IC must work. To avail better power savings, the system must be operated beyond the worst-case estimates. The techniques to address this issue and the findings of chapter 3 are used in the design and development of the proposed countermeasure. Moreover, the selection of the most suitable frequency or voltage for the proposed aggressive scaling countermeasure depends mainly on the worst-case values estimated by the CAD tools. A scheduler is modelled to effectively decide the operating frequency by considering the feedback information from the proposed unit and the worst-case estimates. The complete setup helps to monitor the behaviour of the critical path of the design as the operating conditions vary. In addition, to successfully perform the attacks, the hypothetical power model need to be developed, which must be as close to the actual power traces of the system under test as possible.

7.1 Device Power Consumption vs Attacks

In order to effectively address the issues of DVFS as a countermeasure, it is helpful to analyse and understand the basic concepts of power consumption and the relation between power and leakage of information.

$$P_{dynamic} = CV^2{}_{dd}f \tag{7-1}$$

As shown by Equation (7-1), dynamic power consumption of a CMOS device depends on the load capacitance, operating voltage and frequency. Scaling down frequency helps reducing power but it affects performance. Depending on the computational task, the processor must intelligently choose the appropriate frequency to avoid running the system always at a high speed clock. Reducing operating voltage has a major impact on dynamic power because of the quadratic relation between voltage and power. Moreover, static power can be minimized by reducing the operating voltage. Hence voltage scaling is the most effective and commonly used technique.

In terms of security, the device power consumption is the main source of leakage. From Equation (7-1), the power consumed by a device depends on the supply voltage and operating frequency. The main task of improving security from power attacks can happen by reducing the correlation between the power and internal computation. In order to accomplish this, either voltage or frequency should be scaled during the tasks. The most suitable technique for doing this is DVFS and it is proposed as one of the best countermeasures to DPA attacks.

7.2 Problems with Related Work and Motivation

In [102], an interesting study on using DVFS as a countermeasure for DPA attacks is presented by Yang. The traditional DVFS is used to reduce power consumption of processor designs. But in this study, DVFS is proposed against all kinds of power attacks, which provides randomness in power as voltage and frequency are changed between different operating modes. In addition to this, an advanced control algorithm proposed in the work to introduce more randomness in the design, by creating long random

waiting time between the switchovers of the operating modes. The control algorithm and DVFS was able to reduce the relation between the power traces, while providing energy reduction of approximately 27%. However, the real validation of the technique by prototyping on hardware was not discussed.

Another study proposed by Baddam et al. [103], delivers a random DVS technique to provide an effective countermeasure. Baddam emphasizes that using the same voltage and frequency pairs as in DVFS will not improve the immunity of DPA attacks. The study reports that the linear relation between voltage and frequency would result in easy estimation of (v,f) pairs, thus, results in information leakage. Also, it suggests that the scaling of either voltage or frequency is the best option to improve robustness of the design from attacks.

Naga et al. [10] proposed an aggressive voltage scaling based on the outcomes of the work in [10] . But the drawbacks of the technique are V_{max} and F_{max} depend on t_{cd} (contamination delay) and %error rate. The error detection and correction circuitry increases with t_{cd}, which causes additional area overhead.

On the other hand, the hardware based countermeasures include data masking techniques, re-configurable crypto processors [104]-[106], pseudo random based architecture to generate random sequence [107], ring oscillator approach [110], leakage based analysis [108], 109]. All these techniques improve resistance of the devices from DPA attacks but involve extra design and power overheads.

To overcome the drawbacks of the mentioned techniques, a new timing error detection and correction circuit based on time-borrowing feature is proposed. With this technique, the following are achieved

- the linear relation between V and F is broken;
- increased operating range; V_{max} and F_{max} range is increased, so randomness also increased;
- No error correction overhead; as soon as the error detected, the phase delayed clock is used to capture the late arrival data.

7.3 Hardware Realization of Random Dynamic Voltage and Frequency Scaling

Yang [102] first proposed the usage of DVFS to improve the resistance of embedded systems to DPA attacks. The correlation between power consumption and internal computations can be reduced by varying the supply voltage and frequency randomly.

The hardware realization of DVFS involves an open loop structure, which pre-determines different voltage and frequency pairs by considering all process corners and stores these values as a look-up table. Depending on the application task requirements, the appropriate (v,f) pairs are selected. As mentioned in chapter 2, the drawback of this approach towards power consumptions is that the system is not able to adapt to the changing needs of modern computing world. Similarly, for improving resistance from DPA attacks, the approach is not suitable because once frequency is known to the attacker, it is very easy to calculate the corresponding voltage level.

7.4 Necessity for Aggressive Scaling Techniques

As discussed in chapter 5, further to DVFS, many techniques proposed to improve energy efficiency, the AVS technique is getting more attention. The AVS technique scales the supply voltage beyond the worst case estimates to remove the excess timing margins. In general, processors use

typical operating conditions to guarantee safe operation. But systems are designed to operate in the worst case scenarios, which are very rare to occur. To safeguard the operation, the system designers consider excess timing margins. These excess timing margins limit the systems from having better performance and reduced power consumption.

In [31], proposed Razor technique which works on the principle of aggressive mechanisms with a critical path monitor. The technique monitors critical paths in the design as voltage/ frequency is scaled. The purpose of monitoring critical path helps the system to fix the required operating frequency and voltage for the present task. The CAD tools generally estimate the typical values for any design. Going beyond the estimates of CAD tools remove the excess timing margins and make the systems benefit for maximum performance enhancement or power savings.

The main task involved in AVS systems is not only to make the systems work beyond the typical estimated values but also to make the systems to perform stable and reliable operation. Many techniques have been proposed related to aggressive and stable voltage scaling techniques. TEAtime [55] uses critical path replica to estimate the critical path behaviour as frequency/ voltage is scaled. The critical path estimation techniques are really useful because these techniques estimate the error before it actually happens. As a result, there is no need to halt the instruction execution because of the error. However the major drawback of these techniques is how well the replica resembles the actual critical path.

Similarly, DVFS technique has problems towards security improvement of systems. To address the problems, the above mentioned aggressive scaling

mechanisms are exploited to break the linear relationship between two important variable frequency and voltage.

7.5 Problem Formulation

The goal of the work presented in this chapter is to introduce randomness to power consumption by increasing the available number of operating states and by breaking the linear relation between voltage and frequency.

The Pearson correlation coefficient is the one element that reveals the relation between two variables x and y.

$$\rho(x, y) = \frac{cov(x,y)}{\sqrt{Var(x).Var(y)}} \quad (7\text{-}2)$$

Cov is the co-variance of the two variables and Var is the variance.

Using DVFS as a countermeasure [111], the coefficient can be changed as shown below:

$$\rho(x_{(v_i,f_i)}, y_{(v_i',f_i')}) = \sum_{i=1}^{n} \frac{cov(x_{(v_i,f_i)}, y_{(v_i',f_i')})}{\sqrt{Var(x).Var(y)}} \quad (7\text{-}3)$$

Here $x_{(v_i,f_i)}$= real power measurements when system using DVFS; n = number of possible (v, f) pairs; $y_{(v_i',f_i')}$ = hypothetical power model;

From Equation (7-3), It is clear that the correlation coefficient relies on the number of (v, f) pairs and the information leakage at each particular (v, f) pair.

$$y_{(v_i',f_i')} = HW(x)|_{(v_i', f_i')} \quad (7\text{-}4)$$

The hypothetical model is calculated using Hamming weight, as in Equation (7-4). The coefficient is high only when $x = y$. This condition must not be true for all the cases. To make this work, the hypothetical power

model should not leak the information about the voltage and frequency levels. If the attacker can guess the correct operating frequency of the encryption, the corresponding voltage can be easily calculated from the Equation (7-5), based on their linear relationship. Here (v_i', f_i') are the new set of voltage and frequency values and m is the slope of the linear curve; (v_{old}, f_{old}) are the previous set of voltage and frequency values;

$$v_i' = \frac{(f_i' - f_{old})}{m} + v_{old} \tag{7-5}$$

To make the traditional DVFS as a countermeasure, the relation between voltage and frequency must be broken.

From [90] the number of samples required to estimate in a DPA attack depends on ρ_{max}, the maximum correlation value.

$$S \propto \rho_{max} \tag{7-6}$$

With DVFS, the required number of samples is changed to

$$S_n \propto n \,.\, \rho_{max} \tag{7-7}$$

From Equation (7-7), increasing the number of possible (v, f) pairs, n, increases the number of samples required to attack the system.

Therefore, the goal of the work is to have an increased number of (v, f) pairs, which makes difficult to make DPA successful and to break the relation between voltage and frequency.

7.6 Circuit Architecture and Features

The architecture of our circuit is shown in Figure 7.1.

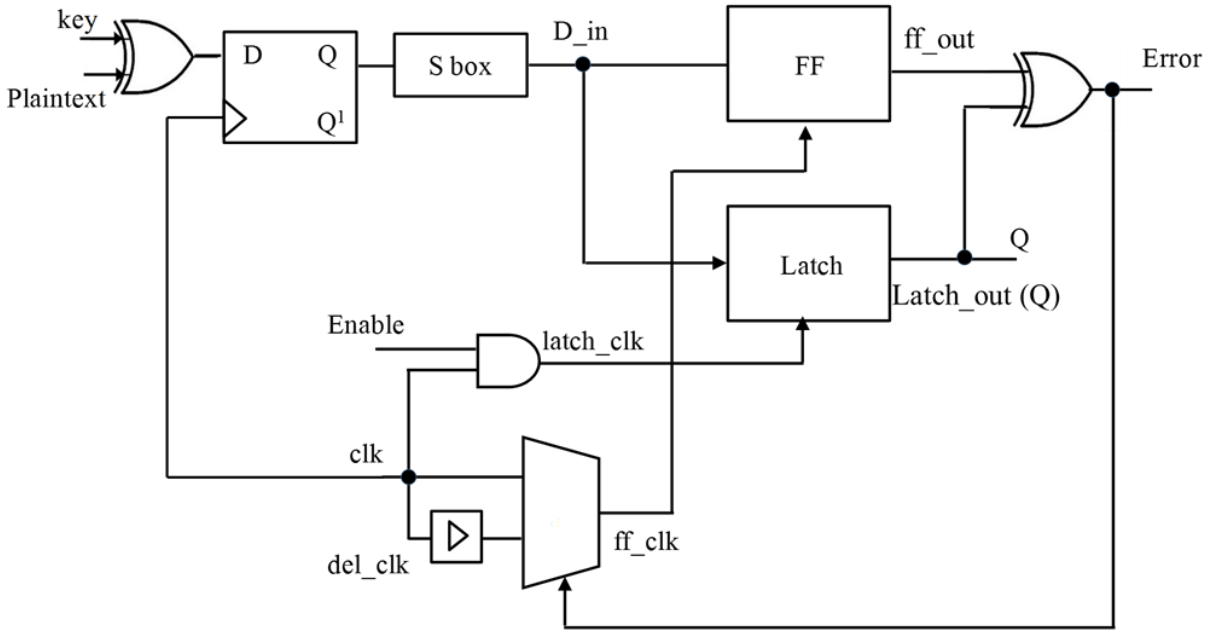


Figure 7.1: The circuit diagram of the proposed technique

- Flip-flop and latch combination used to detect any timing violations.
- Once detected, flip-flop clock is phase delayed to capture the late arrival data.
- Enable is used to enable /disable time borrowing approach
- Flip-flop clock is generated beforehand only selection done on the fly

To break the linear relation between voltage and frequency, aggressive voltage/frequency scaling can be used. AVS systems must operate beyond the worst case estimates yet providing reliable operation. To make this possible, the systems must use timing error detection and correction circuit. The timing errors might occur going beyond the worst case values to reduce timing margins.

Using AFS/AVS, at each (v, f) pair voltage or frequency can be varied to break the linear relationship. At each (v, f) pair, either voltage or frequency

can be varied by keeping the other constant. For example, AFS technique involves frequency variation by keeping voltage constant.

7.7 Benefits of Mixed Random Dynamic Voltage and Frequency Scaling

By considering all the issues and benefits of traditional DVFS approach, randomization countermeasure and AVS/AFS, a new technique called Mixed Random Dynamic Voltage and Frequency Scaling (MRDVFS) is proposed. The technique introduces randomness by four factors:

- the number of states
- voltage or frequency
- Magnitude; voltage or frequency values are selected depending on error rate
- Up or Down

Using DVFS, the possible number of operating modes is shown in Figure 7.2. The procedure for implementation of the proposed technique is illustrated in Figure 7.3. Out of all the possible states, one operating mode is selected randomly. The selection of operating mode is happening randomly to introduce randomness to the countermeasure. However, this is not sufficient to improve immunity. As a result, to increase the randomness of the countermeasure, above 4 factors are considered.

Instead of running at the selected pair (v, f), the next task is deciding to increase either voltage or frequency. The architecture needs to provide provision to vary both. For one encryption round, voltage can be selected and for the next round frequency can be selected. Therefore, the technique has been named as mixed random dynamic voltage and frequency scaling. Tradition approach varies both voltage and frequency at each state. This

approach varies either voltage or frequency at each stage depends on the selection.

After the selection of voltage or frequency, the next job is finding the value of the voltage or frequency. The direction of variation can go up or down. The possible outcomes are V + ΔV, V - ΔV. If V - ΔV is selected, the operating mode still operates in the operating zone; hence the circuit performs normal operation. If V + ΔVis selected, the circuit goes into the non-operating zone. To perform a safe and reliable operation, the circuit needs to employ a timing error detection circuitry. The same can be used for frequency variation. Further, the selection of values is done by considering the timing error rate and the circuit capability to handle the errors.

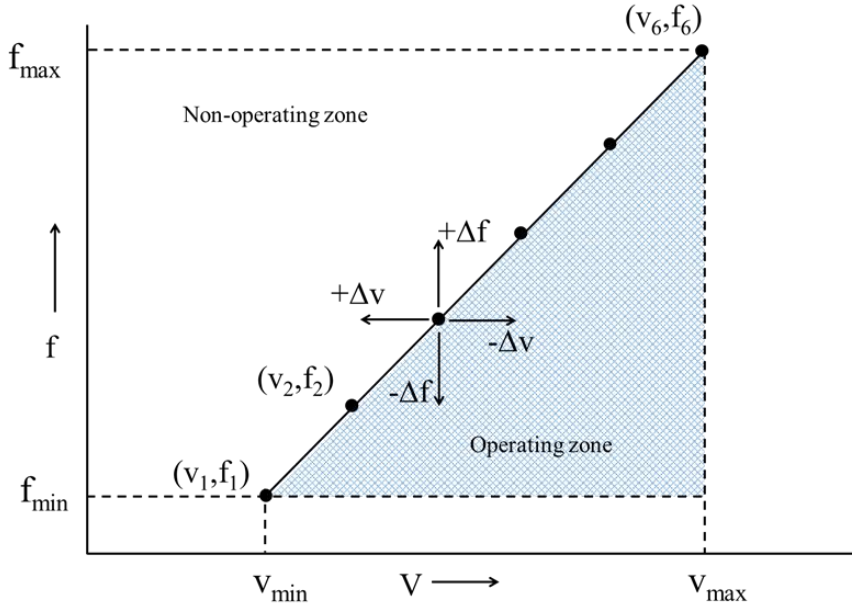


Figure 7.2: Voltage- frequency graph for DVFS and AVS or AFS

Input: V_{min}, V_{max}; F_{min}, F_{max}

Output: (V′, F′) new operating mode

Algorithm:

1: Generate a random number to select one state (v, f) from all possible states (v_1, f_1) to (v_5, f_5)

2: Choose the variable voltage or frequency variation; V or F

3: Select the direction of variation up or down; V + ΔV, V - ΔV

4: Choose the value of the variable;

5: The core unit encrypts/decrypts at new operating mode (v′, f′)

6: Go to step1

Figure 7.3: Implementation of mixed random multi-dynamic voltage and frequency scaling

7.8 Experimental Methodology

The same differential power analysis scheme, shown in Figure 7.4 is reproduced on an experimental platform with AES hardware implementation on Spartan 3E FPGA. The best and fast suitable environment for prototyping is the FPGA and to evaluate the performance of the technique [112][110]-[117]. Real output is the measurements captured from FPGA board while running encryption. Hypothetical output is calculated using Hamming weight method and Matlab software. Correlation power analysis is applied to measure the correlation between the real measured power and hypothetical power values.

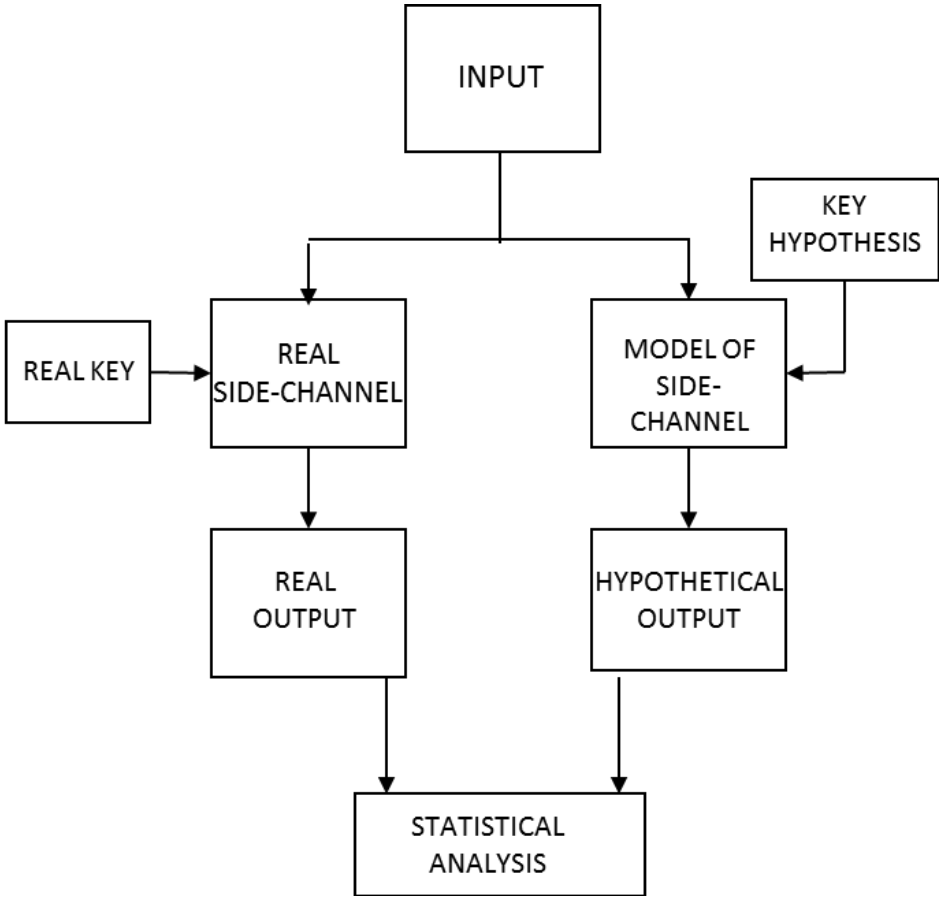


Figure 7.4: Block diagram of the correlation power analysis attacks

The experimental setup used for this work is shown in Figure 7.5. This section provides the methodology used for our experiments to implement DPA attacks along with AES algorithm.

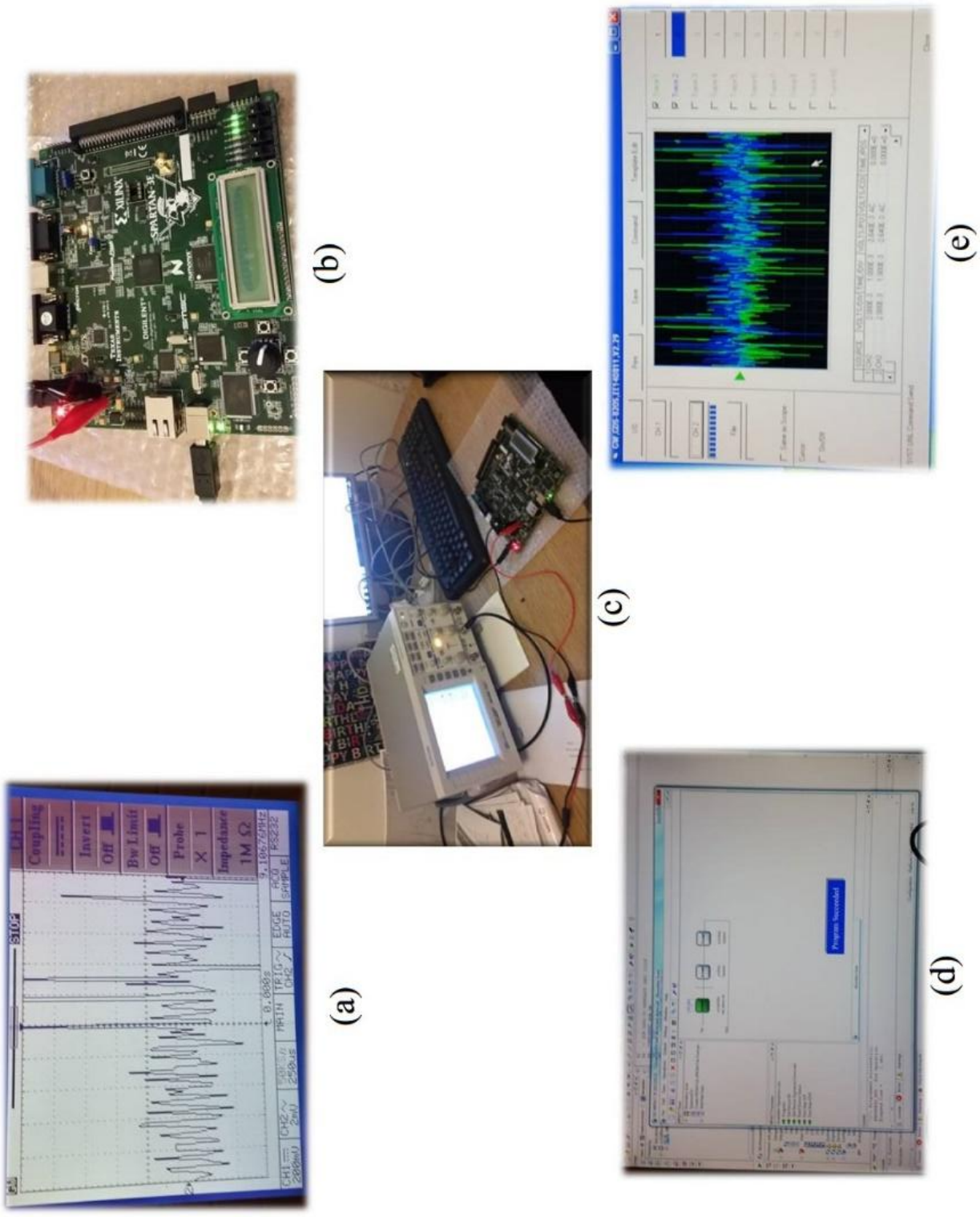


Figure 7.5: Experimental setup used (a) Captured power traces using oscilloscope, (b) Xilinx Spartan 3E FPGA development board, (c) Snapshot of the setup, (d)Successful programming of FPGA, (e) Captured power traces from oscilloscope to PC

7.9 Hardware Configuration

The Spartan 3E FPGA development board is used for implementing the attacks, so Spartan 3E FPGA acts as the device under test. The most advanced encryption algorithm, AES is implemented on FPGA using VHDL. The board contains a 50 MHz on chip crystal oscillator, which acts as a base clock frequency. Using digital clock manager (DCM), different set of frequencies under AFS scheme are generated. To have enough number of samples, the AES is run at 5 MHz, dividing down the base clock frequency by 10. Another advantage of the board is having current sense feature on jumpers jp6 and jp7. In general, FPGA devices have three power supplies; one for I/O blocks; one for the peripherals and the last one for the internal logic. To measure power consumption of the FPGA, the internal logic supply must be used. The power consumption of the device is measured across the resistor and captured using oscilloscope. A trigger signal is created to synchronise the power consumption traces with the corresponding input data.

7.10 AES Sbox Implementation

AES Sbox implemented in VHDL and simulated using Xilinx simulation and verified the functional correctness. Figure 7.6 depicts the block diagram of the proposed technique with Sbox. To implement AFS scheme, three additional blocks are added to the original 8-bit Sbox. 1) Scheduler, 2) random number generation, 3) proposed timing error monitor. The scheduler decides the operating frequency and voltage based on the task requirements and timing monitor feedback. The random generator selects a state from the available number of voltage-frequency pairs. The proposed monitor detects timing error whenever there is a timing violation.

At each clock cycle, the state decided and frequency varied from the pre-calculated one. As a result, the number of states has increased when compared to the traditional DVFS to increase the randomness. For example, the typical operating zone is (1, 8.069 GHz). Using AFS, frequency can be varied with a swing of 0.05 GHz, which results in possible two states of (1, 8.019 GHz) and (1, 8.119). Note that the voltage is kept constant. Using AVS, voltage can be varied with a swing of 0.2v keeping frequency constant. The possible states can be (0.8v, 8.069GHz) and (1.2v, 8.069GHz).

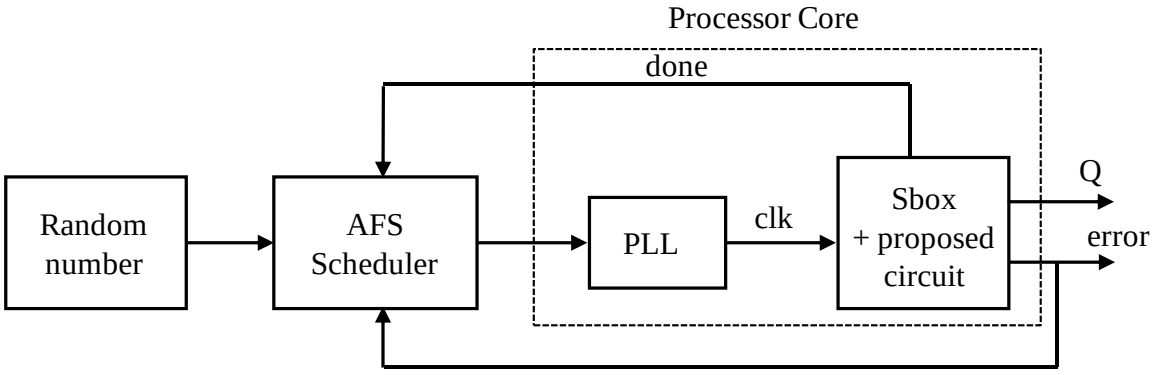


Figure 7.6: Block diagram of Sbox with proposed technique

The proposed technique validated by using AFS mechanism because both either AFS or AVS result in timing errors. The key idea is that the technique needs to detect the timing errors and handle them with safe operation. Another reason for choosing AFS is there is no need to alter the hardware in order to verify the proposed technique. Using PLLs, the required frequency is achieved in each state. The AFS allows the frequency to increase from the worst case estimation. As a result, the critical path in the design executes more delay than the new clock period. Hence, timing errors start to appear. The CAD tool estimated the worst case delay as 2.5 ns. As discussed earlier, no timing errors occurred up to this point. As

frequency scaled further, it causes timing violations and thus timing errors. However, it provides a performance improvement of 20 %. Another consideration here is that the rate of increase in frequency is more that the rate of increase in timing error rate.

The timing monitor unit detects the error as soon as it occurs. The whole system must handle the errors without any performance loss. The idea is that the system must be error tolerant while providing maximum performance. The unit not only detects the error but also self corrects it without timing penalty. With the help of the proposed unit, the system enable to operate beyond the worst case estimates yet providing a reliable and stable operation.

7.11 Procedure

The proposed technique replaces flip-flop at each pipeline stage of a processor. In general, for a 5-stage pipeline processor consists of instruction fetch, instructions decode, execution stage, memory access stage and write back stage. Flip-flops at each stage have been replaced with the proposed technique to monitor errors at each stage. In this case, for AES Sbox design, the flip-flop in the design replaced with the technique to monitor and recover the timing errors as shown in Figure 7.6.

With the number of states of (v, f) pairs, random generator block chooses a state every clock cycle feeds it to the scheduler unit. The scheduler decides the frequency based on complete signal and proposed unit's feedback. The new clock frequency information is fed to the PLL unit to generate the required frequency to run the AES encryption algorithm. The function verification of the whole design is done by using Xilinx simulation tool and downloaded to the FPGA. The power traces while running encryption algorithm are measured using oscilloscope.

7.12 Power Analysis Attacks

To validate the technique as an effective countermeasure, the power analysis attacks have to be performed on the implemented Sbox.

A set of plain texts are encrypted using number of keys to generate cipher texts;

$A = \{a_1, a_2, a_3 \ldots a_{np}\}$ *here* $np =$ *number of plain texts*

$K = \{k_1, k_2, k_3 \ldots k_{nk}\}$ $nk =$ *number of keys*

$$C_{(i,j)} = Sbox(a_i \; XOR \; k_j) \tag{7-8}$$

With the above calculation, all subsets of inputs and keys are used to generate a matrix with the order of np * nk.

$$C_{(np,\, nk)} = \begin{bmatrix} C(1,1) & C(1,2) & \cdots & C(1,nk) \\ C(2,1) & C(2,2) & \ldots & C(2,nk) \\ \vdots & . & \ddots & \vdots \\ . & . & . & . \\ C(np,1) & C(np,2) & \cdots & C(np,nk) \end{bmatrix} \tag{7-9}$$

The above cipher text calculation is done by varying frequency and voltage. Hence it is changed to:

$$C_{(i,j)} = Sbox(a_i \; XOR \; k_j) \Big|_{(V_i,\; F_j)} \tag{7-10}$$

Here (v_i, F_j) corresponds to different voltage and frequency pairs. The entire cipher text matrix calculation was done by changing frequency for each element in the matrix. The hypothetical power model is done by looking at the 1-0s and 0-1s transition of Sbox output.

P = Hamming weight (Sbox(A XOR K)) . The cipher text matrix is formed as power model matrix, the size of both matrices are the same. Real power consumption is measured for each plaintext and formed a matrix with real

power traces with dimensions as np X nt. Here np = number of plain texts and nt = number of samples.

$$P_{(nd,nt)} = \begin{bmatrix} P(1,1) & P(1,2) & \cdots & P(1,nt) \\ P(2,1) & P(2,2) & \dots & P(2,nt) \\ \vdots & . & \ddots & \vdots \\ . & . & . & . \\ P(nd,1) & P(nd,2) & \cdots & P(nd,nt) \end{bmatrix} \quad (7\text{-}11)$$

The correlation coefficient reveals the relation between these two variables T and P. if T and P are matched, the correlation coefficient is high and low indicates that the two elements are not identical.

To measure correlation, first consider first row of the hypothetical power model and then the first column of the real power traces.

$$X_j\ (j \in (1, np));\ X_j = [p_{(1,j)}, p_{(2,j)}, p_{(3,j)} \dots p_{(np,j)}]^T$$

$$Y_t\ (t \in (1, nt))\ ;\ Y_t = [T_{(1,t)}\ T_{(2,t)}\ T_{(3,t)} \dots T_{(nt,t)}] \quad (7\text{-}12)$$

The above concept extended to AFS approach in that case X_j contains set of data operating at different voltage and frequency pairs. The calculation of correlation coefficient must be modified accordingly.

$$Corr_{(x_j,\ y_t)} = \frac{Cov_{(x_j,\ y_t)}}{\sqrt{Var_{(x_j)}.Var_{y_t}}} \quad (7\text{-}13)$$

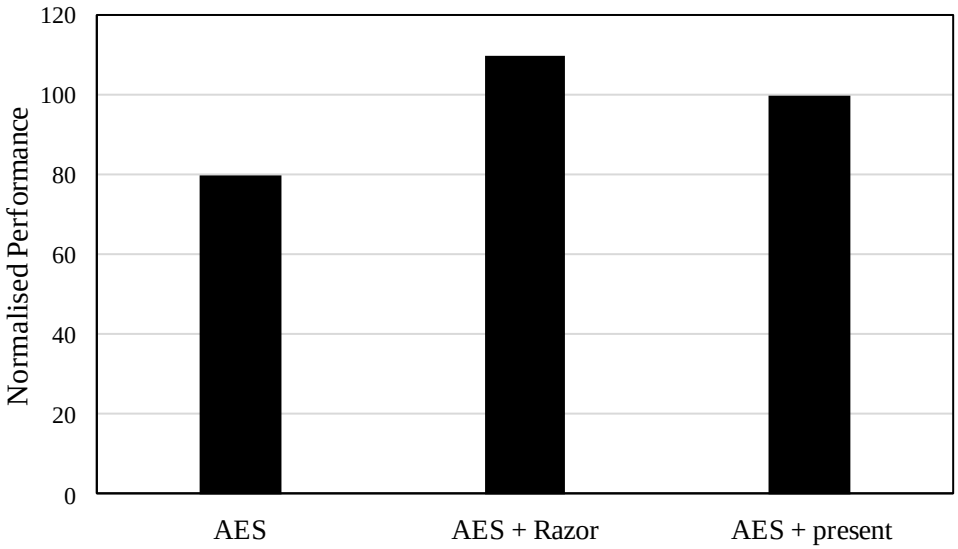
The countermeasure is more effective if the correlation coefficient is small; that means the relation between X_j and Y_t are not related to each other. The attacker needs to predict the exact (v, f) pairs used for input data to have higher correlation coefficients.

7.13 Results

The results presented in this section are dedicated to test the robustness of the proposed countermeasure. The simulation results of correlation analysis with and without countermeasure for all possible key values are listed in Appendix A.

The maximum operating frequency of AES encryption suggested by CAD tools is 400 MHz. Since the countermeasure allows the system to operate beyond the worst case estimates, the operating frequency extended up to 540 MHz. The performance improvements offered by the other techniques presented in Figure 7.7. The proposed countermeasure provides significant improvement compared with other techniques.

To verify the effectiveness of the technique, statistical analysis is performed on the measured power traces and hypothetical power traces. The correlation coefficient reveals the relation between the two traces. Figure 7.8 shows the peaks for correct key (94, 112) respectively with the countermeasure. Figure 7.9 shows the peaks for the same keys (94, 112) without any countermeasure. It is clearly evident that the peaks corresponding to the correct keys are high without any countermeasure. Adding countermeasure makes difficult to detect the correct key.



-

Figure 7.7: Performance improvement of present technique

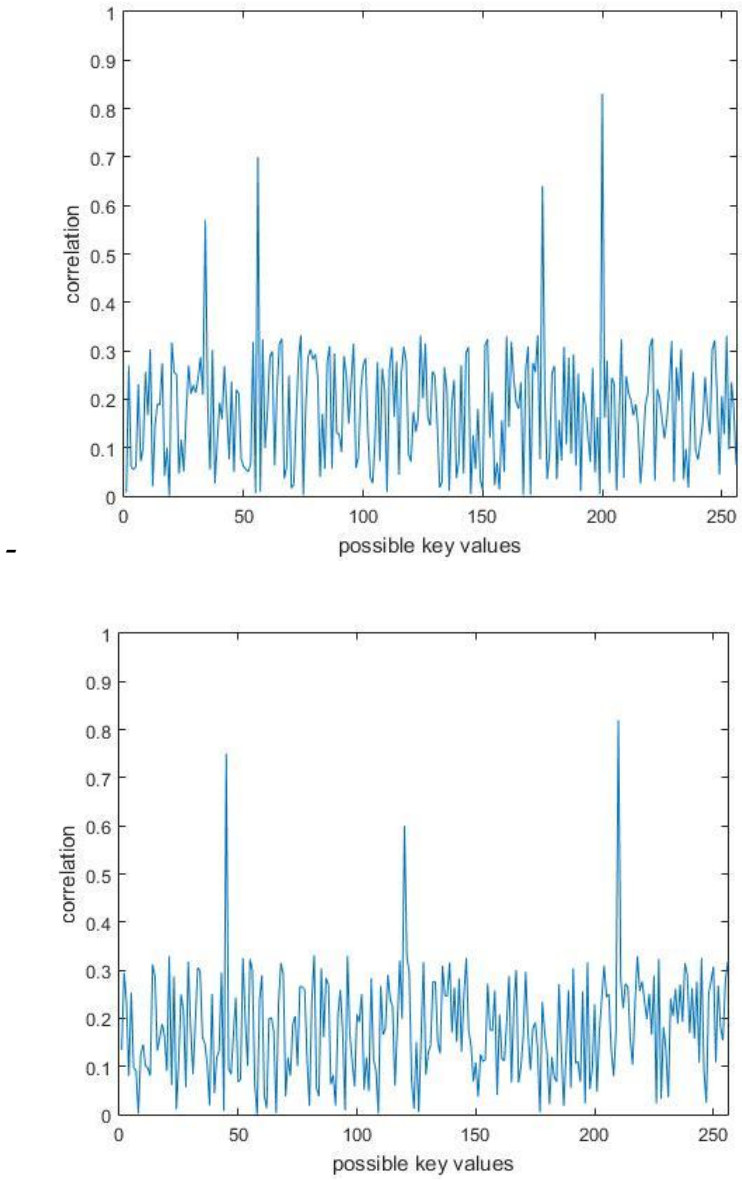


Figure 7.8: Correlation coefficient for correct keys 112d, 94d with countermeasure

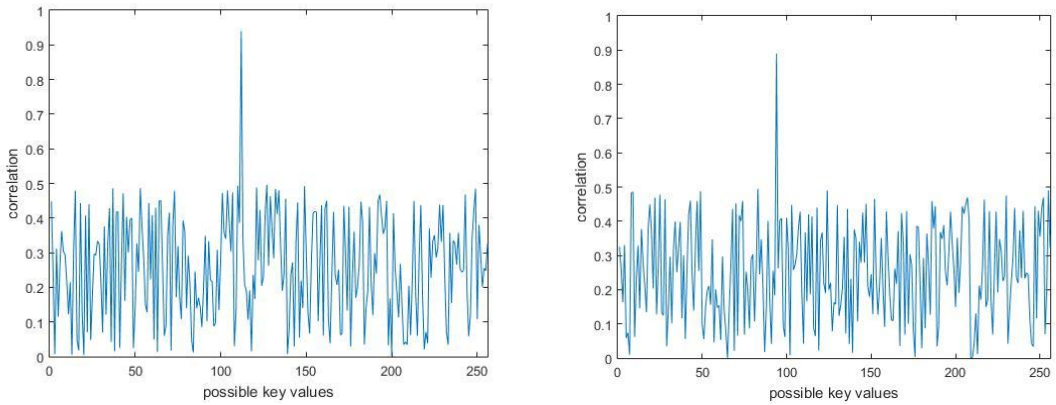


Figure 7.9: Correlation coefficient for correct keys 112d, 94d without any countermeasure

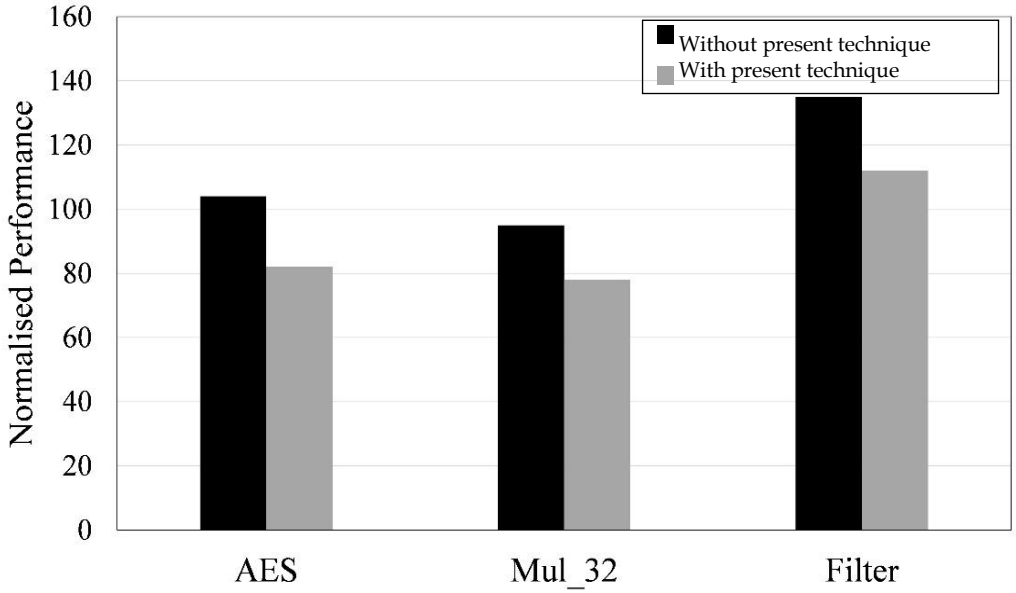


Figure 7.10: Performance improvement of various benchmarks using present technique

Different benchmarks have been simulated to evaluate the performance improvement of the present technique. The results of the three benchmarks are presented in Figure 7.10. From the results, it is evident that the present technique provides approximately 20% performance improvement compared with the original design.

Figure 7.11 shows the comparison of power savings of the present technique and other techniques with frequency below and after worst-case frequency. The present technique consumes less power, providing approximately 12% more power savings than the original design in the case of frequency less than the worst-case estimate. The savings are improved by 3 times by using the proposed techniques in the case of frequency after the worst-case estimate. Figure 7.12 illustrates the power savings of the present technique. The power consumption of AES circuit is 5.3mW with 1V supply voltage at 200 MHz operating frequency. With the countermeasure, the power is reduced to 4.9mW with 1V supply voltage and varying frequency. Note that the countermeasure provides more savings with the reduction of supply voltage.

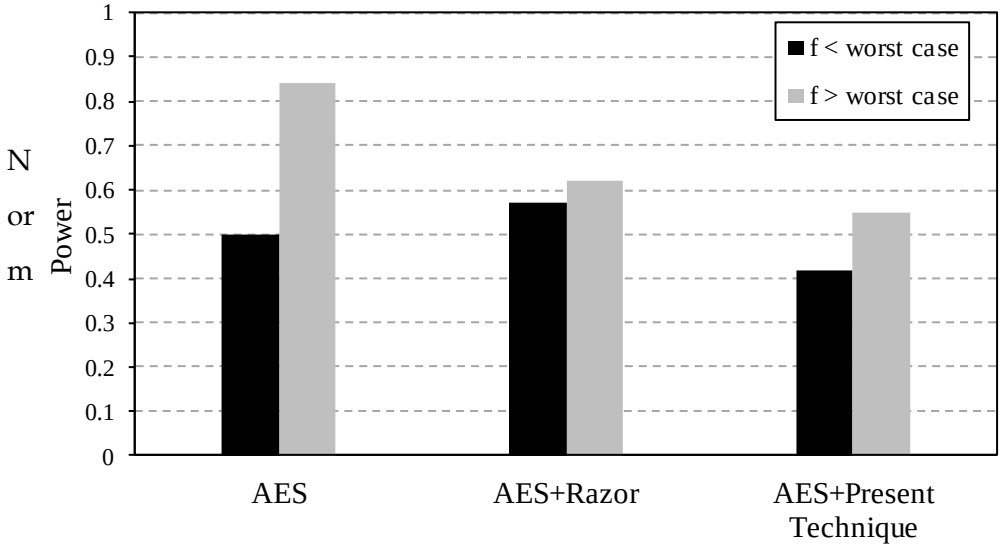


Figure 7.11: Power consumption of Razor and present technique

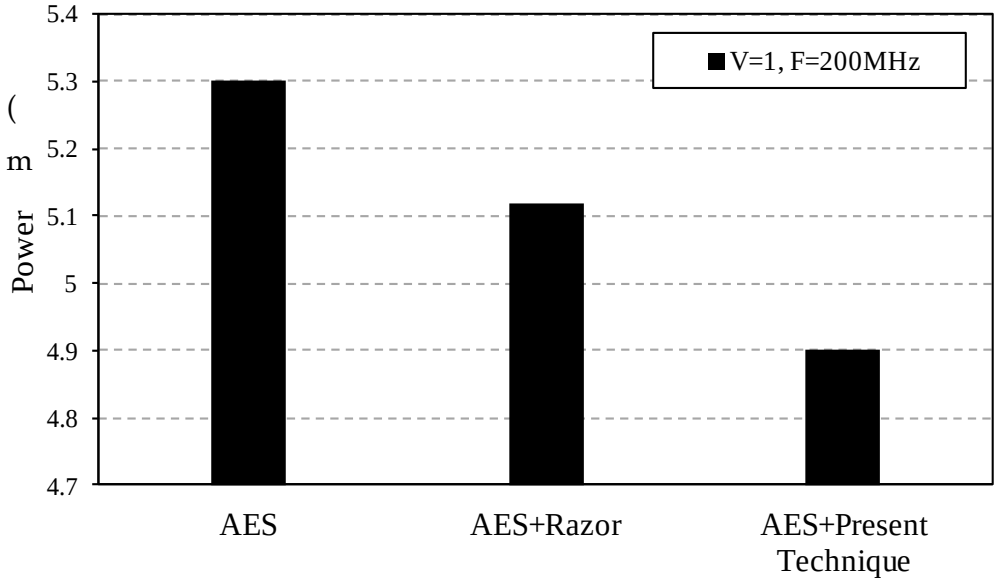


Figure 7.12: Power savings of present technique

Overall, the countermeasure provides more performance and power reductions compared to other aggressive scaling techniques. Less area overhead compared with other DPA countermeasures

7.14 Summary

In this chapter, a novel technique is presented to improve the immunity of embedded systems from DPA attacks. The presented countermeasure randomly varies voltage while keeping frequency constant or frequency while keeping voltage constant. Using this method, in case of frequency scaling, performance can be pushed further to exploit data heterogeneity. Further, the technique reduces the correlation coefficient for the correct key and improves it for the wrong keys. Also, the technique has less area overhead, power overhead and design overhead compared with other techniques. The proposed countermeasure is implemented in VHDL and downloaded on to the SPARTRAN 3E FPGA to evaluate it. Results are

provided to indicate the proposed technique improved immunity, performance improvement and ultra-power savings.

CHAPTER 8

Conclusions and Future Work

This chapter presents the conclusions of this research work and the possible future directions to extend the work further.

8.1 Conclusions of the Thesis

Smart and portable devices with limited battery supply require efficient power management techniques to extend the battery life. Supplementing the system power management and deciding to go for the low power modes when the system is idle is one way of offering power savings. A novel technique based on aggressive scaling is proposed in this work to address the issues of power efficiency and security. The aggressive scaling technique employs critical path monitor to extend the boundaries and to operate systems beyond the safety margins. This work identifies that monitoring of the critical path while frequency or voltage is scaling to a level exceeding the nominal values is important, not only for the efficient

use of excess margins but also for the stable and reliable operation of the system. These properties were realised by exploiting the critical path of the design as the main control input to the aggressive scaling unit. Using this method, the proposed technique detects and self-corrects the timing error to resume normal operation without much timing penalty.

The first objective of power efficiency is achieved by developing an aggressive scaling unit using an effective critical path monitor as the control trigger, as covered in chapter 5. The valuable property of sequential elements was employed in the critical path monitor design to detect real-time timing errors. The method was defined in such a way that the timing errors start to occur when the frequency or voltage is scaled above threshold levels. Once the error occurred, the proposed technique uses phase delayed clock to self-correct the error. In this way, the aggressive unit provides normal operation beyond the safety limits. As a requirement, the error detection and correction circuit should experience smooth transition of operating modes by using advanced PLL and on-chip power supply units to supply the required levels. The stability of the error detection circuit was found to depend on the number of timing error occurred and the operating frequency and voltage of the original design. However, these values can be pre-evaluated by using advanced CAD tools. The main benefits of the technique are less area, power and design overheads compared with the other latest techniques. The technique is validated by prototyping on to an FPGA in a Spartan 3E FPGA development board, as presented in chapter 5. Results show that the system can provide performance improvement of 20% or power savings of 30%.

The second objective is to improve the security of systems from power analysis attacks. These attacks are becoming more popular because of the

ease of their implementation. Also, devices are shared over a network known as IoT [14] is emerging. The security of these devices is becoming a major research challenge. To address this, researchers have proposed many techniques at different levels of abstraction. In this thesis, an effective countermeasure has been developed with less area, power and design overheads by considering the drawbacks of the previously reported countermeasures. The proposed countermeasure works on the principle of the earlier proposed aggressive scaling to improve the immunity of systems from DPA attacks. Like DVFS, aggressive scaling can also be used to introduce randomness in the power consumption. This prevents the attackers extracting the secret information by analysing the power leakage from the systems. The proposed technique (MRDVFS), uses not only the voltage or frequency variation but also the values can go up or down to add more randomness to the power profile, as discussed in detail in chapter 6. The latest encryption algorithm, AES is used to evaluate the resistance of the proposed circuit from DPA attacks. As a platform to design, develop and test the aggressive scaling unit as an effective countermeasure, a complete prototype model to replicate power analysis attacks was developed as detailed in chapter 4. The model simulated with encryption algorithm, proposed technique with PLL and controller; and downloaded on to the FPGA prototype board to capture real power traces. Statistical techniques are also implemented in Matlab to create hypothetical model and to compare this with real values to extract the hidden information.

Similarly, for validating the proposed power optimized circuit design experimentally, using the FPGA board with Spartan 3E FPGA was presented in chapter 4. In addition, results provided to indicate that the technique improves resistance of the system from DPA attacks. Further,

the technique provides performance improvements of approximately 20% and power savings of 30%, as presented in chapter 6.

8.2 Limitations and Future Work

The present research work could be extended in several directions. The following paragraphs introduce some limitations and relevant areas of future research.

One limitation is that the meta-stability circuit design is not considered in the proposed technique. The reason for this is the availability of meta-stable detectors, presented in [31] [55], [57]. The circuits can be added directly to the proposed aggressive scaling unit to overcome the stability issues.

This work was developed for single processor-core designs. The future work can include extending the architecture further to the multi-core processors or heterogeneous systems. Depending on the requirements, appropriate changes can be made to suit to the multi-core architectures to trade off performance and power consumption. In that case, the design need to evaluate in terms of area, power and design overheads. A more in-depth analysis on the state of the proposed aggressive scaling design may be beneficial for the multi-core processors power optimization in addition to the overheads involved.

Further, Integration of the present error correction and detection framework into an existing, commerical static timing analyser that calculates the worst case estimates after removal of the excess timing margins. However, this work poses several challenges to be widely accepted by the commercial CAD tools.

8.3 List of Publications

The research work in this thesis were presented and published in official proceedings of rigorously refereed conferences and journals through the following research papers:

P. Rathnala, A. Kharaz and T. Wilmshurst, A Novel Time-Borrowing Technique for Aggressive Voltage/Frequency Scaling, IEEE Transactions on VLSI, submitted

P. Rathnala, A. Kharaz and T. Wilmshurst, A Low Overhead Countermeasure Technique using Aggressive Voltage/Frequency Scaling against Power Analysis Attacks, IEEE Transactions on VLSI, submitted

P. Rathnala, A. Kharaz and T. Wilmshurst, An efficient adaptive voltage scaling using delay monitor unit, IEEE conference on Ph.D. Research in Microelectronics and Electronics (PRIME), 2015

P. Rathnala, A. Kharaz and T. Wilmshurst, Aggressive Voltage Scaling for Energy Efficiency, Derby Electrical and Electronics Research Showcase, 2015

P. Rathnala, A. Kharaz and T. Wilmshurst, A practical approach to differential power analysis using PIC microcontroller based embedded system, IEEE Computer Science and Electronic Engineering Conference (CEEC), 2014

P. Rathnala, A. Kharaz and T. Wilmshurst, Differential Power Analysis Attacks, Derby Electrical and Electronics Research Showcase, 2014

P. Rathnala, A. Kharaz and T. Wilmshurst, Fundamentals of Low Power Design Techniques for Embedded Systems, Derby Electrical and Electronics Research Showcase, 2013

References

[1] Massoud Pedram, "Power-optimal encoding for DRAM address bus (poster session)", *Proceedings of the 2000 international symposium on Low power electronics and design - ISLPED,* 2000

[2] Driving down power, http://automotive.electronicspecifier.com/power/driving-down-power

[3] M. Dale, 2008, "The Power of RTL Clock-gating," in *Chip Design Magazine*. vol. 2008.

[4] H. Mahmoodi, V. Tirumalashetty, M. Cooke and K. Roy, "Ultra Low-Power Clocking Scheme Using Energy Recovery and Clock Gating," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 1, pp. 33-44, Jan. 2009.

[5] H. Singh, K. Agarwal, D. Sylvester and K. J. Nowka, "Enhanced Leakage Reduction Techniques Using Intermediate Strength Power Gating," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 11, pp. 1215-1224, Nov. 2007.

[6] M. S. Bhat, Srigowri, V. V. Rao and B. P. V. Pai, "Implementation of dynamic voltage and frequency scaling for system level power reduction," *Circuits, Communication, Control and Computing (I4C), 2014 International Conference on,* Bangalore, 2014, pp. 425-430.

[7] Y. Qadri, H.S. Gyjarathi and K.D.McDonald-Maier, "Low Power Processor Architectures and Contemporary Techniques for Power Optimization – A Review," *Journal of Computers*, Vol.4, No.10, 2009.

[8] Krisztián Flautner and David Flynn, "A Combined Hardware-Software Approach for Low-Power SoCs: Applying Adaptive Voltage Scaling and Intelligent Energy Management Software", *Design Conference on System-on-Chip and ASIC Design Conference*, 2003.

[9] D. D. Hwang, P. Schaumont, K. Tiri and I. Verbauwhede, "Securing embedded systems," in *IEEE Security & Privacy*, vol. 4, no. 2, pp. 40-49, March-April 2006.

[10] N. D. P. Avirneni and A. K. Somani, "Countering Power Analysis Attacks UsingReliable and Aggressive Designs," in *IEEE Transactions on Computers*, vol. 63, no. 6, pp. 1408-1420, June 2014.

[11] W. Wolf, "Multimedia applications of multiprocessor systems-on-chips," Design, *Automation and Test in Europe*, 2005, pp. 86-89 Vol. 3.

[12] C. H. Gebotys and B. A. White, "Methodology for attack on a Java-based PDA," In *CODES+ISSS '06*, pages 94–99, New York, NY, USA, 2006. ACM Press.

[13] E. Biham and A. Shamir, “Power Analysis of the Key Scheduling of the AES Candidates,” In *Second Advanced Encryption Standard (AES) Candidate Conference,* pages 343 –347, 2003.
[14] Internet of Things [IoT], http://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/
[15] D. F. YongBin Zhou, “Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing,”2005.
[16] S. B. Ors, F. Gurkaynak, E. Oswald, and B. Preneel, “Power-analysis attack on an asic aes implementation,” 2004.
[17] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, “ Examining smart-card security under the threat of power analysis attacks,” *IEEE Transactions on Computers*, 51(5):541–552, 2002.
[18] Jude Angelo Ambrose, “Power Analysis Side Channel Attacks: The Processor Design-level Context,” PhD dissertation, March 2009.
[19] P. Kocher, J. Jaffe, and B. Jun, Introduction to differential power analysis and related attack,” Technical Report, 1998.
[20] S.Mangard, “A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion,” In P. J. Lee and C. H. Lim, editors, Information Security and Cryptology - ICISC 2002, 5th *International Conference Seoul*, Korea, November 28-29, 2002, Revised Papers, volume 2587 of Lecture Notes in Computer Science, pages 343–358.
[21] P. Kocher, J. Jaffe, and B. Jun, “Using unpredictable information to minimize leakage from smartcards and other cryptosystems, “U.*S. Patent 6327661*, 1999.
[22] Mehdi Masoomi, Massoud Masoumi, Mahmoud Ahmadian K. N. Toosi, “A Practical Differential Power Analysis Attack against an FPGA Implementation of AES Cryptosystem,” 2010.
[23] N. Banerjee, K. Roy, H. Mahmoodi, and S. Bhunia, "Low power synthesis of dynamic logic circuits using fine-grained clock gating," in *Proceedings of the conference on Design, automation and test in Europe:* 2006.
[24] H. Shimada, H. Ando, and T. Shimada, "Pipeline stage unification: a low-energy consumption technique for future mobile processors," *International Symposium on Low power Electronics and Design Seoul, Korea, ACM*, 2003.
[25] M.Y. Qadri, H.S. Gyjarathi and K.D.McDonald-Maier, “Low Power Processor Architectures and Contemporary Techniques for Power Optimization – A Review”, *Journal of Computers*, Vol.4, No.10, Oct 2009.

[26] V.Venkatachalam and M. Franz, “Power reduction techniques for microprocessor Systems”, *ACM Computing Surveys (CSUR*), 37, 2005.
[27] M. R. Stan and W. P. Burleson, 1995, "Bus-invert coding for low-power I/O," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 3, pp. 49-58.
[28] Christian Piguet, 2005, “Low-Power Electronics Design”, CRC Press.
[29] National Semiconductor, 2009, “PowerWise Adaptive Voltage Scaling (AVS) for Portable Applications.”
[30] Findlay Shearer, 2008, “Power Management in Mobile Devices”, Newnes.
[31] D. Ernst et al., "Razor: a low-power pipeline based on circuit-level timing speculation,” 2003. *36th Annual IEEE/ACM International Symposium on Microarchitecture*, 2003, pp. 7-18.
[32] Rotem, E., Naveh, A., Moffie, M., and Mendelson, “Analysis of thermal monitor features of the intel pentium m processor,” In *TACS Workshop at ISCA-31*, 2004.
[33] AMD (2000), Technology, AMD white paper.
[34] Dongsheng Ma, Wing-Hung Ki and Chi-Ying Tsui, "An integrated one-cycle control buck converter with adaptive output and dual loops for output error correction," in *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 140-149, Jan. 2004.
[35] M. Barai, S. Sengupt and J. Biswas, "Optimized design of a high frequency digital controller for DVS-enabled adaptive DC-DC converter," 2008 *IEEE Power Electronics Specialists Conference, Rhodes*, 2008, pp. 1801-1807.
[36] H. R. Pourshaghaghi and J. P. de Gyvez, "Fuzzy-Controlled Voltage Scaling Based on Supply Current Tracking," in *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2397-2410, Dec. 2013.
[37] Mohamed Elgebaly and Manoj Sachdev, “Efficient Adaptive Voltage Scaling System Through On-Chip Critical Path Emulation,” in *Proceedings of the 2004 International Symposium on Low Power Electronics and Design (ISLPED’04),* 2004.
[38] Y.Ikenaga et al., “Fast Voltage Control Scheme with Adaptive Voltage Control Steps and Temporary Reference Voltage Overshoots for Dynamic Voltage and Frequency Scaling,” *IEEE Asian Solid-State Circuits Conference*, Japan, Nov. 2008.
[39] Weiwei Shan and Zhipeng Xu, “Timing Error Prediction based Adaptive Voltage Scaling for Dynamic Variation Tolerance,” 2014.
[40] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, “A dynamic voltage scaled microprocessor system,” in *IEEE International Solid-State Circuits Conference, ISSCC*. 2000.

[41] W. Lombardi, M. Altieri, Y. Akgul, D. Puschini and S. Lesecq, "Multivariable Voltage and Frequency Control for DVFS Management," *IEEE Conference on Control Applications*, France, Oct. 2014.
[42] Sangyoung Park and Jaehyun Park, "Accurate Modeling of the Delay and Energy Overhead of Dynamic Voltage and Frequency Scaling in Modern Microprocessors," *IEEE Transactions on Computer aided design of Integrated circuits and Systems*, vol. 32, No. 5, May 2013.
[43] NDP Avirneni, "RAKSHA:Reliable and Aggressive frameworK for System design using High-integrity Approaches", PhD dissertation, Iowa State University, 2012.
[44] Rizvandi, N, Taheri J, and Zomaya A, "Some observations on optimal frequency selection in dvfs-based energy consumption minimization," *Journal of Parallel and Distributed Computing*, 2001, 71(8):1154–1164.
[45] Texas Instruments Inc., "OMAP3530 and OMAP3525 Applications Processors Data Sheet."
[46] R. Mc Gowen et al., "Power and temperature control on a 90-nm Itanium family processor," in *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 229-237, Jan. 2006.
[47] Arthur Musah and Andy Dykstra, 2008, "Power-Management Techniques for OMAP35x Applications Processors", Texas Instruments White Paper.
[48] Samsung Electronics Co. Ltd, "S3C2410X – 32 Bit RISC Microprocessor User Manual."
[49] E.Kristjansson, 2008, "ARM7 low power design," ST Microelectronics, Whitepaper.
[50] Brant Ivey, 2011, "Low-Power Design Guide", AN1416, Microchip Technology Inc. 2011.
[51] ARM., "ARM: 1176 IEM Reference Methodology." #118
[52] David Weir, Cadence Design, "When do you know you have saved enough power?"
[53] Cyclos Semiconductor, "Addressing the Power-Performance IC Design Conundrum."
[54] S Das, "RAZOR: A Variability-Tolerant Design Methodology for Low power and Robust Computing," PhD dissertation, The University of Michigan, 2009.
[55] Augustus K Uht, "Going beyond Worst-Case Specs with TEAtime," *IEEE Computer Society*, 2004.
[56] A. K. Uht, "Uniprocessor performance enhancement through adaptive clock frequency control," in *IEEE Transactions on Computers*, vol. 54, no. 2, pp. 132-140, Feb. 2005.

[57] Mihir R Chouhury, V. Chandra, R. C. Aitken and K. Mohanram, “Time- Borrowing Circuit designs and Hardware Prototyping for Timing Error Resilience,” *IEEE transactions on Computers*, Vol 63, No 2, Feb 2014.

[58] A. N. D. Prasad and A. Somani, “Low Overhead Soft Error Mitigation Techniques for High-Performance and Aggressive Design,” *IEEE Transactions on Computers*, 2013.

[59] M. Fojtik et al., "Bubble Razor: Eliminating Timing Margins in an ARM Cortex-M3 Processor in 45 nm CMOS Using Architecturally Independent Error Detection and Correction," in *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 66-81, Jan. 2013.

[60] M. Fojtik et al., "Bubble Razor: An architecture-independent approach to timing-error detection and correction," 2012 *IEEE International Solid-State Circuits Conference*, San Francisco, CA, 2012, pp. 488-490.

[61] M. E. V. Alba et al., "An aggressive power optimization of the ARM9-based core using RAZOR," *TENCON 2012 - 2012 IEEE Region 10 Conference*, Cebu, 2012, pp. 1-5.

[62] K. Chae, S. Mukhopadhyay, Chang-Ho Lee and J. Laskar, "A dynamic timing control technique utilizing time borrowing and clock stretching," *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, San Jose, CA, 2010, pp. 1-4

[63] B. Zandian, W. Dweik, Suk Hun Kang, T. Punihaole and M. Annavaram, "WearMon: Reliability monitoring using adaptive critical path testing," *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, Chicago, IL, 2010, pp.

[64] C Zhang, D Ma and A Srivastava, “Integrated Adaptive DC/DC Conversion with Adaptive Pulse-Train Technique for Low-Ripple Fast Response”, *ISLPED 04*, 2004.

[65] P. C. Kocher, J. Jaffe and B.Jun, “Differential power analysis,” *CRYPTO99: Proceesings of the 19th Annual International Cryptology Conference on Advances in Cryptography*, London, UK, Springer – Verlag, 1999.

[66] E. Trichina, T. Korkishkoand, and K. H. Lee, “Small size, low power, side channel-immune AES coprocessor: Design and synthesis results,” in *Proc. AES*, vol. 3373, Lecture Notes in Computer Science, 2005, pp. 113–127.

[67] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, “Power and electromagnetic analysis: Improved model, consequences and comparisons,” Integration, *VLSI Journal*, vol. 40, no. 1, pp. 52–60, 2007.

[68] M. Masoumi and M. H. Rezayati, "Novel Approach to Protect Advanced Encryption Standard Algorithm Implementation Against

Differential Electromagnetic and Power Analysis," in *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 256-265, Feb. 2015.

[69] T.-H. Lee, C. Canovas, and J. Cledier, "An overview of side channel analysis attacks," in *Proc. ACM Symposium Information, Computer and Communications Security*, Tokyo, Japan, 2008, pp. 33–43.

[70] Z. Martinasek, V. Clupek, T. Krisztina, "General Scheme of Differential Power Analysis", *International Conference on Telecommunications and Signal Processing, TSP* 2013.

[71] David Flowers, "Data Encryption Routines for the PIC18," Microchip Technology Inc, AN953.

[72] M. Alam, S. Ghosh, D. R. Choudhury, I. Sengupta, "First-order DPA Vulnerability of Rijndael: Security and Area-delay Optimization Trade-off", *International Journal of Network Security*, Vol.15, No.3, PP.219-230, May 2013.

[73] Joan Daemen, Vincent Rijmen, "The Design of Rijndael: AES - The Advanced Encryption Standard", 2002.

[74] N. Pramstaller, "An AES ASIC-Implementation Resistant to Differential Power Analysis," Master's thesis, IAIK, University of Technology Graz, Austria, 2004.

[75] N. Pramstaller, E. Oswald, S. Mangard, F. K. G¨urkaynak, and S. Haene, "A Masked AES ASIC Implementation," in Austrochip 2004,Villach, Austria, Proceedings, E. Ofner and M. Ley, Eds., October 2004.

[76] E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen, "A side-channel analysis resistant description of the AES S-Box," in *Proc. 12th Int. Workshop FSE*, 2005, pp. 413–423.

[77] M. A. Hasan, "Power analysis attacks and algorithmic approaches to their countermeasures for Koblitz curve cryptosystems," in *IEEE Transactions on Computer*s, vol. 50, no. 10, pp. 1071-1083, Oct 2001.

[78] T. S. Messerges, "Securing the AES Finalists Against Power Analysis Attacks, "In *Proceedings of the 7th International Workshop on Fast Software Encryption*, pages 150–164, London, UK, 2001.

[79] K. Tiri, M. Akmal, and I. Verbauwhede, "A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards," in *Proc. 28th Eur. Solid State Circuits Conference*, Sep. 2002, pp. 403–406.

[80] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," Design, *Automation and Test in Europe Conference and Exhibition*, 2004.

[81] C. Tokunaga and D. Blaauw, "Securing encryption systems with a switched capacitor current equalizer," *IEEE Journal on Solid-State Circuits*, vol. 45, no. 1, pp. 23–31, Jan. 2010.
[82] Po-Chun Liu, Hsie-Chia Chang and Chen-Yi Lee, "A True Random-Based Differential Power Analysis Countermeasure Circuit for an AES Engine."
[83] K. Baddam, "Hardware Level Countermeasures against Differential Power Analysis", PhD dissertation, University of Southapmton, UK, 2012.
[84] Hspice reference manual, Synopsys, Inc, April 2006, http://www.synopsys.com.
[85] Spectre, Cadence, Inc, April 2006, http://www.cadence.com.
[86] Nanosim user guide, Synopsys, Inc, April 2006, http://www.synopsys.com
[87] Ultrasim, Cadence, Inc, April 2006, http://www.Cadence.com.
[88] PrimePower user guide, Synopsys, Inc, April 2006, http://www.synopsys.com
[89] J. Goodwin and P. R. Wilson, "Advanced encryption standard (AES) implementation with increased DPA resistance and low overhead," in *Proc. IEEE ISCAS*, May 2008, pp. 3286–3289.
[90] S. Mangard, E. Oswald, T. Popp, "Power Analysis Attacks: Revealing the Secrets of Smart Cards", *Springer-Verlag*, New York, 2007.
[91] Microchip Technology Inc., "PIC18F2420/2520/4420/4520 Data Sheet," 2003.
[92] Tim Wilmshurst, "An Introduction to the Design of Small-Scale Embedded Systems", September 2001.
[93] Tim Wilmshurst, "Designing Embedded Systems with PIC Microcontrollers, Principles and Applications", 2nd Edition, Elsevier, 2009.
[94] "FreeCapture- A PC remote control program for GDS-800S (RS-232/GPIB)", Version 2.0.
[95] J. Park, "Self-Tuning Dynamic Voltage Scaling Techniques for Processor Design," PhD dissertation, The University of Texas at Austin, May 2013.
[96] I. shin, J. Kim and Y. shin, "Aggressive Voltage Scaling through Fast Correction of Multiple Errors with Seamless Pipeline Operation," *IEEE transactions on circuits and Systems*, Vol 62, No 2, Feb 2015.
[97] K. A. Bowman et al., "Energy-Efficient and Metastability Immune Resilient Circuits for Dynamic Variation Tolerance," *IEEE journal of solid-state circuits*, Vol 44, NO 1, Jan 2009.

[98] C. Kwanyeob and S. Mukhopadhyay, "A Dynamic Timing Error Prevention Technique in Pipelines With Time Borrowing and Clock Stretching," *IEEE Transactions on Circuits and Systems*, 2014.
[99] A. N. D. Prasad and A. Somani, "Countering Power Analysis Attacks using Reliable and Aggressive Designs," *IEEE Transactions on Computers*, 2013.
[100] M. Fojtik et al., "Bubble Razor: Eliminating Timing Margins in an ARM Cortex-M3 Processor in 45nm CMOS Using Architecturally Independent Error Detection and Correction," *IEEE Journal of solid-state circuits*, vol . 48, no. 1, January 2013.
[101] E. Stott, J. M. Levine and P. Y. K .Cheung, "Timing Fault Detection in FPGA-based circuits," *IEEE 22nd annual international symposium on field-programmable custom computing machines (FCCM)*, May 2014.
[102] Shengqi Yang, W. Wolf, N. Vijaykrishnan, D. N. Serpanos and Yuan Xie, "Power attack resistant cryptosystem design: a dynamic voltage and frequency switching approach," *Design, Automation and Test in Europe*, 2005, pp. 64-69 Vol. 3.
[103] K. Baddam and M. Zwolinski, "Evaluation of Dynamic Voltage and Frequency Scaling as a Differential Power Analysis Countermeasure," *20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID'07)*, Bangalore, 2007, pp. 854-862.
[104] R. Ramanarayanan *et al.*, "18Gbps, 50mW reconfigurable multi-mode SHA Hashing accelerator in 45nm CMOS," *ESSCIRC, 2010 Proceedings of the*, Seville, 2010, pp. 210-213.
[105] L. Yan, B. Wu, Y. Wen, S. Zhang and T. Chen, "A Reconfigurable Processor Architecture Combining Multi-core and Reconfigurable Processing Unit," *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, Bradford, 2010, pp. 2897-2902.
[106] W. Shan, X. Fu and Z. Xu, "A Secure Reconfigurable Crypto IC With Countermeasures Against SPA, DPA, and EMA," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 7, pp. 1201-1205, July 2015.
[107] P. C. Liu, H. C. Chang and C. Y. Lee, "A True Random-Based Differential Power Analysis Countermeasure Circuit for an AES Engine," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 2, pp. 103-107, Feb. 2012.
[108] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proc. CHES*, 2004, pp. 16–29.
[109] M.L. Akkar and C. Giraud, "An implementation of DES and AES, secure against some attacks," in Proc. CHES, 2001, pp. 309–318.

[110] M. Masoomi, M. Masoumi and M. Ahmadian, "A practical differential power analysis attack against an FPGA implementation of AES cryptosystem," *Information Society (i-Society), 2010 International Conference on*, London, 2010, pp. 308-312.
[111] S. Ordas, M. Carbone, G. Ducharme, S. Tiran and P. Maurine, "Efficiency of the RDVFS countermeasure," *Faible Tension Faible Consommation (FTFC), 2014 IEEE*, Monaco, 2014, pp. 1-4.
[112] S. B. Ors, F. K. G ¨ urkaynak, E. Oswald, and B. Preneel, " Power Analysis Attack on an ASIC aes Implementation", Proceedings of the International Conference on Information Technology: Coding and Computing, ITCC 04, Volume 2, Washington, USA, *IEEE Computer Society*, 2004.
[113] S. B. Ors, F. K. G ¨ urkaynak, E. Oswald, and B. Preneel, "Power analysis attack on an asic aes implementation," ITCC 04, Proceedings of the International Conference on Information Technology: Coding and Computing , *IEEE Computer Society* , Vol 2. Washington, DC, USA, 2004.
[114] J. Wu, Y. Shi and M. Choi, "Measurement and Evaluation of Power Analysis Attacks on Asynchronous S-Box," in *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 10, pp. 2765-2775, Oct. 2012.
[115] J. Wu, Y. Shi and M. Choi, "FPGA-based measurement and evaluation of power analysis attack resistant asynchronous S-Box," *Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE*, Binjiang, 2011, pp. 1-6.
[116] W. Shan *et al.*, "Evaluation of Correlation Power Analysis Resistance and Its Application on Asymmetric Mask Protected Data Encryption Standard Hardware," in *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 10, pp. 2716-2724, Oct. 2013.
[117] A. L. Masle and W. Luk, "Detecting power attacks on reconfigurable hardware," *22nd International Conference on Field Programmable Logic and Applications (FPL)*, Oslo, 2012, pp. 14-19.
[118] Brumley and D. Boneh, "Remote timing attacks are practical," In *Proceedings of the 12th USENIX Security Symposium*, August 2003.
[119] J. Quisquater and D. Samyde, "ElectroMagnetic Analysis (EMA): Measures and CounterMeasures for Smart Cards," In *E-smart*, pages 200–210, 2001.
[120] T. S. Messerges, "Using second-order power analysis to attack dpa resistant software," In *Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems*, pages 238–251, London, UK, 2000.

[121] K. Baddam, M. Zwolinski, “Evaluation of Dynamic Voltage and Frequency Scaling as a Differential Power Analysis Countermeasure”, *20th International Conference on VLSI Design (VLSID 07), IEEE*, 2007.

[122] P. C. Liu, H. C. Chang and C. Y. Lee, "A Low Overhead DPA Countermeasure Circuit Based on Ring Oscillators," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 7, pp. 546-550, July 2010.

[123] S. V. Kosonocky, A. J. Bhavnagarwala, K. Chin, G. D. Gristede, A.-M. Haen, W. Hwang, M. B. Ketchen, S. Kim, D. R. Knebel, K. W. Warren, and V. Zyuban, "Low-power circuits and technology for wireless digital systems," *IBM Journal of Research and Development*, vol. 47, pp. 283-298, 2003.

[124] A.C. Vaidya, V.Harpale, Feb 2012, “Review: Power Optimisation of Embedded Systems – An ARMed Approach,” *Proceedings of the National Conference, NCNTE-2012*, Mumbai, India.

[125] Krisztián Flautner and David Flynn, “A Combined Hardware-Software Approach for Low-Power SoCs: Applying Adaptive Voltage Scaling and Intelligent Energy Management Software”, *System-on-Chip and ASIC Design Conference*, 2003.

APPENDIX A

Table A.1 Simulation results of correlation analysis with countermeasure; correlation coefficients for all possible keys (94, 112 used as keys)

Serial No	Key = 112, with correlation	Key = 94, with correlation
0	0.2716	0.2264
1	0.3019	0.1318
2	0.0423	0.1225
3	0.3045	0.3293
4	0.2108	0.0126
5	0.0325	0.2951
6	0.0928	0.3044
7	0.1823	0.2654
8	0.3192	0.0329
9	0.3216	0.0873
10	0.0525	0.1118
11	0.3235	0.2266
12	0.3191	0.0455
13	0.1618	0.2404
14	0.2668	0.0356
15	0.0473	0.2179
16	0.1406	0.1647
17	0.3052	0.2597
18	0.2641	0.2383
19	0.3198	0.3012
20	0.2186	0.297
21	0.0119	0.1114
22	0.283	0.2329
23	0.3113	0.0659
24	0.2262	0.0102
25	0.2526	0.248
26	0.2477	0.1667
27	0.1307	0.16
28	0.2185	0.3016
29	0.0571	0.2033
30	0.2353	0.2059

Continued on next page..

Table A.1 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
31	0.0106	0.2865
32	0.0923	0.2685
33	0.57	0.1922
34	0.0324	0.061
35	0.2745	0.08
36	0.2316	0.2955
37	0.1057	0.0096
38	0.3167	0.1633
39	0.0115	0.056
40	0.1462	0.3262
41	0.1272	0.2376
42	0.2552	0.1668
43	0.2651	0.157
44	0.0623	0.75
45	0.1633	0.2273
46	0.1485	0.0141
47	0.2154	0.0238
48	0.2365	0.1739
49	0.2516	0.0322
50	0.092	0.2727
51	0.2266	0.2725
52	0.2184	0.2408
53	0.0542	0.05
54	0.0397	0.2199
55	0.7	0.1729
56	0.3199	0.3243
57	0.1135	0.2163
58	0.1951	0.2668
59	0.0746	0.1513
60	0.2504	0.1441

Continued on next page..

Table A.1 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
61	0.085	0.2751
62	0.1687	0.0278
63	0.233	0.0444
64	0.297	0.0578
65	0.3198	0.1303
66	0.1824	0.2771
67	0.0462	0.2678
68	0.0498	0.0202
69	0.0858	0.1331
70	0.2802	0.1756
71	0.0848	0.1389
72	0.2714	0.219
73	0.0812	0.2093
74	0.3098	0.0973
75	0.1167	0.1439
76	0.0655	0.0052
77	0.0837	0.328
78	0.2053	0.0557
79	0.1578	0.0354
80	0.1172	0.1241
81	0.2769	0.066
82	0.1951	0.1632
83	0.1832	0.1132
84	0.3057	0.3172
85	0.0953	0.3068
86	0.2524	0.0176
87	0.2512	0.246
88	0.1268	0.0897
89	0.1893	0.1409
90	0.0253	0.1826

Continued on next page..

Table A.1 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
91	0.018	0.3142
92	0.1769	0.1392
93	0.2597	0.3277
94	0.3113	0.1005
95	0.0433	0.2337
96	0.1896	0.2221
97	0.1565	0.1797
98	0.004	0.2327
99	0.1124	0.2222
100	0.0541	0.0594
101	0.2648	0.0427
102	0.1037	0.333
103	0.1762	0.057
104	0.0552	0.0109
105	0.2007	0.1871
106	0.0877	0.294
107	0.218	0.2231
108	0.2297	0.0635
109	0.2494	0.123
110	0.1502	0.1536
111	0.0279	0.3272
112	0.0763	0.0521
113	0.3044	0.2852
114	0.0508	0.2149
115	0.2753	0.1254
116	0.1794	0.0636
117	0.332	0.1428
118	0.0261	0.1607
119	0.1476	0.6
120	0.0356	0.1965

Continued on next page..

Table A.1 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
121	0.3206	0.0754
122	0.0015	0.1282
123	0.2583	0.1943
124	0.2724	0.0839
125	0.2896	0.0968
126	0.0281	0.2057
127	0.1333	0.0884
128	0.0866	0.2748
129	0.2667	0.3276
130	0.1438	0.2434
131	0.3035	0.1146
132	0.0606	0.1947
133	0.0879	0.0359
134	0.0485	0.3021
135	0.0454	0.2932
136	0.2898	0.2726
137	0.1932	0.0869
138	0.1833	0.1981
139	0.0483	0.0075
140	0.2843	0.1418
141	0.2074	0.1042
142	0.117	0.0538
143	0.1711	0.0596
144	0.1339	0.141
145	0.0253	0.0314
146	0.08	0.1995
147	0.0411	0.157
148	0.0613	0.232
149	0.08	0.2333
150	0.1391	0.2128

Continued on next page..

Table A.1 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
151	0.0166	0.0112
152	0.3009	0.0229
153	0.3149	0.1065
154	0.1636	0.177
155	0.1631	0.2181
156	0.1126	0.1359
157	0.3	0.2733
158	0.1231	0.2395
159	0.0371	0.3229
160	0.2601	0.1771
161	0.1299	0.1084
162	0.0806	0.0352
163	0.1346	0.2037
164	0.0322	0.2596
165	0.044	0.1412
166	0.314	0.0303
167	0.3187	0.0888
168	0.1917	0.0512
169	0.0199	0.0937
170	0.0783	0.1467
171	0.1177	0.1757
172	0.2737	0.1525
173	0.0051	0.2918
174	0.64	0.1727
175	0.0563	0.3145
176	0.2164	0.2126
177	0.2439	0.3192
178	0.2159	0.0802
179	0.1503	0.2254
180	0.1823	0.0964

Continued on next page..

Table A.1 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
181	0.0988	0.2239
182	0.2482	0.2317
183	0.063	0.0227
184	0.2289	0.0849
185	0.0612	0.0747
186	0.1228	0.2226
187	0.2085	0.2815
188	0.2601	0.1148
189	0.027	0.2602
190	0.3098	0.2251
191	0.2586	0.0022
192	0.1623	0.2007
193	0.1453	0.1289
194	0.1489	0.3053
195	0.1021	0.0004
196	0.1695	0.1541
197	0.1703	0.1414
198	0.2725	0.1536
199	0.83	0.2567
200	0.2148	0.1075
201	0.1262	0.2616
202	0.2705	0.1571
203	0.1776	0.0119
204	0.1169	0.0586
205	0.313	0.2406
206	0.292	0.1578
207	0.1834	0.0509
208	0.2075	0.1137
209	0.1957	0.82
210	0.0692	0.0639

Continued on next page..

Table A.1 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
211	0.1004	0.2461
212	0.157	0.0809
213	0.0768	0.3058
214	0.2814	0.0897
215	0.0649	0.2552
216	0.0753	0.0629
217	0.0569	0.0958
218	0.0759	0.0304
219	0.1452	0.1921
220	0.1037	0.2278
221	0.3078	0.1822
222	0.1434	0.1419
223	0.0616	0.2148
224	0.3016	0.2159
225	0.3266	0.2263
226	0.1463	0.2119
227	0.037	0.3151
228	0.086	0.0696
229	0.1362	0.2364
230	0.1983	0.0787
231	0.0874	0.0398
232	0.2009	0.2024
233	0.2371	0.15
234	0.0739	0.1529
235	0.0391	0.2206
236	0.0989	0.2568
237	0.1063	0.1167
238	0.1414	0.2207
239	0.1693	0.1387
240	0.0285	0.2806

Continued on next page..

Table A.1 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
241	0.0875	0.2776
242	0.267	0.0855
243	0.0097	0.2045
244	0.3096	0.1941
245	0.2434	0.1802
246	0.1629	0.29
247	0.1928	0.0883
248	0.0791	0.106
249	0.1529	0.0397
250	0.321	0.3133
251	0.1823	0.2152
252	0.1737	0.1598
253	0.077	0.2131
254	0.1737	0.1816
255	0.1362	0.2158

Table A.2 Simulation results of correlation analysis without countermeasure; correlation coefficients for all possible keys (94, 112 used as keys)

Serial No	Key = 112, with correlation	Key = 94, with correlation
0	0.6292	0.1813
1	0.3661	0.2403
2	0.4856	0.1742
3	0.3845	0.3312
4	0.0172	0.0729
5	0.2977	0.0353
6	0.4309	0.0366
7	0.3475	0.0212
8	0.2482	0.1349
9	0.6248	0.1495
10	0.553	0.1219
11	0.5661	0.2545
12	0.2484	0.2093
13	0.3955	0.2573
14	0.5817	0.311
15	0.6223	0.3242
16	0.4456	0.064
17	0.1379	0.0463
18	0.4359	0.2321
19	0.048	0.0313
20	0.2712	0.1751
21	0.4446	0.1768
22	0.6225	0.287
23	0.5406	0.1616
24	0.323	0.1312
25	0.5045	0.2238
26	0.278	0.2471
27	0.6479	0.1734
28	0.6586	0.1159
29	0.5761	0.05
30	0.2593	0.1954

Continued on next page...

Table A.2 – continued from previous page

Serial No	Key = 112, without correlation	Key = 94, without correlation
31	0.3032	0.0874
32	0.1645	0.0148
33	0.5229	0.2516
34	0.5886	0.0809
35	0.6091	0.1475
36	0.3722	0.2293
37	0.3992	0.1197
38	0.0993	0.2454
39	0.5998	0.1316
40	0.3003	0.2278
41	0.1371	0.2347
42	0.5998	0.1474
43	0.5084	0.0065
44	0.5883	0.1103
45	0.19	0.1414
46	0.4488	0.0901
47	0.4429	0.0657
48	0.0819	0.2739
49	0.2715	0.1433
50	0.1835	0.2959
51	0.4778	0.1304
52	0.1889	0.2564
53	0.5975	0.1323
54	0.5511	0.2695
55	0.26	0.2517
56	0.3319	0.1258
57	0.4632	0.072
58	0.5562	0.2635
59	0.4064	0.3164
60	0.3832	0.1092

Continued on next page...

Table A.2 – continued from previous page

Serial No	Key = 112, without correlation	Key = 94, without correlation
61	0.2174	0.2238
62	0.3043	0.1462
63	0.4759	0.2778
64	0.5896	0.2563
65	0.4806	0.0558
66	0.0124	0.2873
67	0.4499	0.33
68	0.2923	0.1715
69	0.2919	0.2948
70	0.078	0.196
71	0.5431	0.0516
72	0.2166	0.0666
73	0.1642	0.1357
74	0.2285	0.2496
75	0.2505	0.2752
76	0.3644	0.2633
77	0.3746	0.1062
78	0.2639	0.178
79	0.2654	0.03
80	0.3436	0.0372
81	0.4384	0.0454
82	0.6339	0.2262
83	0.4816	0.1651
84	0.2667	0.0632
85	0.5546	0.165
86	0.0896	0.0492
87	0.0403	0.0183
88	0.0562	0.2836
89	0.1093	0.1869
90	0.2161	0.3099

Continued on next page...

Table A.2 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
91	0.2012	0.2322
92	0.0078	0.1943
93	0.3599	0.89
94	0.0636	0.293
95	0.0977	0.3296
96	0.4208	0.0002
97	0.5729	0.2885
98	0.6495	0.2042
99	0.3806	0.33
100	0.6646	0.1759
101	0.369	0.1598
102	0.3436	0.2671
103	0.2205	0.0759
104	0.2867	0.166
105	0.3279	0.3003
106	0.0474	0.1916
107	0.5918	0.2817
108	0.0431	0.2462
109	0.2908	0.1953
110	0.5511	0.0822
111	0.94	0.2221
112	0.409	0.0278
113	0.5458	0.2087
114	0.5908	0.2203
115	0.6207	0.2433
116	0.1272	0.2969
117	0.1724	0.3274
118	0.5986	0.2563
119	0.3956	0.1938
120	0.3359	0.3094

Continued on next page...

Table A.2 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
121	0.4085	0.1934
122	0.5463	0.0057
123	0.3546	0.0403
124	0.1347	0.2876
125	0.3026	0.1614
126	0.2853	0.2816
127	0.644	0.0698
128	0.4134	0.1841
129	0.4636	0.21
130	0.4801	0.0107
131	0.2313	0.2049
132	0.3447	0.1208
133	0.3711	0.0165
134	0.1043	0.1632
135	0.3747	0.0642
136	0.4632	0.041
137	0.2843	0.0685
138	0.5575	0.0488
139	0.4876	0.063
140	0.24	0.0142
141	0.3028	0.2117
142	0.2576	0.094
143	0.517	0.1795
144	0.4895	0.2317
145	0.2869	0.1664
146	0.4625	0.1786
147	0.6301	0.1484
148	0.5228	0.0413
149	0.4704	0.1635
150	0.0729	0.2843

Continued on next page...

Table A.2 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
151	0.26	0.2913
152	0.3939	0.0901
153	0.3063	0.0695
154	0.0336	0.1883
155	0.1525	0.2134
156	0.5561	0.139
157	0.0104	0.0687
158	0.5758	0.316
159	0.052	0.0274
160	0.446	0.0352
161	0.3335	0.0473
162	0.1453	0.0555
163	0.3811	0.207
164	0.0815	0.1912
165	0.4474	0.0174
166	0.3997	0.3104
167	0.0373	0.2429
168	0.0376	0.2459
169	0.1017	0.0211
170	0.0131	0.2868
171	0.2901	0.3115
172	0.5548	0.3281
173	0.4116	0.2863
174	0.3468	0.2619
175	0.5759	0.1711
176	0.0651	0.0592
177	0.6054	0.1329
178	0.072	0.0446
179	0.3447	0.0103
180	0.0954	0.313

Continued on next page...

Table A.2 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
181	0.3729	0.1004
182	0.0031	0.0985
183	0.5111	0.111
184	0.5658	0.1557
185	0.6112	0.2161
186	0.658	0.0084
187	0.3368	0.2807
188	0.1809	0.1863
189	0.0672	0.2847
190	0.3386	0.116
191	0.3904	0.1487
192	0.5086	0.0181
193	0.0553	0.059
194	0.4411	0.2209
195	0.3447	0.1103
196	0.114	0.2995
197	0.6257	0.0394
198	0.3937	0.3295
199	0.2938	0.18
200	0.6279	0.2356
201	0.4373	0.3332
202	0.3013	0.0959
203	0.5598	0.1382
204	0.3551	0.1549
205	0.3693	0.2547
206	0.4534	0.2727
207	0.2448	0.0334
208	0.1595	0.0594
209	0.3859	0.1199
210	0.5779	0.0189

Continued on next page...

Table A.2 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
211	0.2712	0.174
212	0.0751	0.1119
213	0.2959	0.0586
214	0.2001	0.0696
215	0.2676	0.3017
216	0.5556	0.2251
217	0.2691	0.1562
218	0.2601	0.304
219	0.2403	0.0347
220	0.0935	0.2485
221	0.1734	0.2454
222	0.0579	0.1873
223	0.2863	0.0614
224	0.1715	0.1991
225	0.1984	0.1
226	0.2832	0.0447
227	0.0795	0.0709
228	0.33	0.2983
229	0.4709	0.0238
230	0.1624	0.0808
231	0.5234	0.0179
232	0.0494	0.1472
233	0.2626	0.0044
234	0.0023	0.2991
235	0.1471	0.0656
236	0.0009	0.0311
237	0.1261	0.1025
238	0.095	0.152
239	0.1787	0.0339
240	0.1166	0.3318

Continued on next page...

Table A.2 – continued from previous page

Serial No	Key = 112, with correlation	Key = 94, with correlation
241	0.0924	0.1107
242	0.3993	0.0991
243	0.6007	0.0207
244	0.6263	0.0994
245	0.1475	0.0155
246	0.3218	0.1685
247	0.2507	0.2538
248	0.3492	0.2104
249	0.1766	0.03
250	0.0456	0.027
251	0.2909	0.2591
252	0.1159	0.3017
253	0.0174	0.1779
254	0.6365	0.0364
255	0.2871	0.2753

APPENDIX B

AES algorithm

The functions in AES algorithm can be grouped as AddRoundKey, SubBytes (Sbox), Shiftrows and Mixcolumns. The AddRoundKey operation combines the input data with a secret key. The Sbox function maps the input value with a corresponding value in the table. The table implementation can be done as a loo-up table or a dynamic computation. In this case, it is implemented as a look up table, 'Etable', loaded with pre-calculated values. Similarly, for decryption, 'Dtable' is used. It is same as ETable in encryption but provides inverse look up results. The C code for Sbox implementation is as follows:

```
for (i=0; i < size(ETable); i++)

{
    Sbox(i) = ETable(Sbox(i));

}
ETable[] =

{
0x63,0x7C,0x77,0x7B,0xF2,0x6B,0x6F,0xC5,0x30,0x01,0x67,0x2B,0x
FE,0xD7,0xAB,0x76,0xCA,0x82,0xC9,0x7D,0xFA,0x59,0x47,0xF0,0xA
D,0xD4,0xA2,0xAF,0x9C,0xA4,0x72,0xC0,0xB7,0xFD,0x93,0x26,0x36,
0x3F,0xF7,0xCC,0x34,0xA5,0xE5,0xF1,0x71,0xD8,0x31,0x15,0x04,0x
C7,0x23,0xC3,0x18,0x96,0x05,0x9A,0x07,0x12,0x80,0xE2,0xEB,0x27
,0xB2,0x75,0x09,0x83,0x2C,0x1A,0x1B,0x6E,0x5A,0xA0,0x52,0x3B,0
xD6,0xB3,0x29,0xE3,0x2F,0x84,0x53,0xD1,0x00,0xED,0x20,0xFC,0x
B1,0x5B,0x6A,0xCB,0xBE,0x39,0x4A,0x4C,0x58,0xCF,0xD0,0xEF,0xA
A,0xFB,0x43,0x4D,0x33,0x85,0x45,0xF9,0x02,0x7F,0x50,0x3C,0x9F,
0xA8,0x51,0xA3,0x40,0x8F,0x92,0x9D,0x38,0xF5,0xBC,0xB6,0xDA,0
x21,0x10,0xFF,0xF3,0xD2,0xCD,0x0C,0x13,0xEC,0x5F,0x97,0x44,0x1
7,0xC4,0xA7,0x7E,0x3D,0x64,0x5D,0x19,0x73,0x60,0x81,0x4F,0xDC,
0x22,0x2A,0x90,0x88,0x46,0xEE,0xB8,0x14,0xDE,0x5E,0x0B,0xDB,0
xE0,0x32,0x3A,0x0A,0x49,0x06,0x24,0x5C,0xC2,0xD3,0xAC,0x62,0x9
1,0x95,0xE4,0x79,0xE7,0xC8,0x37,0x6D,0x8D,0xD5,0x4E,0xA9,0x6C,
0x56,0xF4,0xEA,0x65,0x7A,0xAE,0x08,0xBA,0x78,0x25,0x2E,0x1C,0x
A6,0xB4,0xC6,0xE8,0xDD,0x74,0x1F,0x4B,0xBD,0x8B,0x8A,0x70,0x3
E,0xB5,0x66,0x48,0x03,0xF6,0x0E,0x61,0x35,0x57,0xB9,0x86,0xC1,
```

```
0x1D,0x9E,0xE1,0xF8,0x98,0x11,0x69,0xD9,0x8E,0x94,0x9B,0x1E,0x
87,0xE9,0xCE,0x55,0x28,0xDF,0x8C,0xA1,0x89,0x0D,0xBF,0xE6,0x4
2,0x68,0x41,0x99,0x2D,0x0F,0xB0,0x54,0xBB,0x16
};


DTable[] =

{
0x52,0x09,0x6A,0xD5,0x30,0x36,0xA5,0x38,0xBF,0x40,0xA3,0x9E,0x
81,0xF3,0xD7,0xFB,0x7C,0xE3,0x39,0x82,0x9B,0x2F,0xFF,0x87,0x34
,0x8E,0x43,0x44,0xC4,0xDE,0xE9,0xCB,0x54,0x7B,0x94,0x32,0xA6,0
xC2,0x23,0x3D,0xEE,0x4C,0x95,0x0B,0x42,0xFA,0xC3,0x4E,0x08,0x2
E,0xA1,0x66,0x28,0xD9,0x24,0xB2,0x76,0x5B,0xA2,0x49,0x6D,0x8B,
0xD1,0x25,0x72,0xF8,0xF6,0x64,0x86,0x68,0x98,0x16,0xD4,0xA4,0x
5C,0xCC,0x5D,0x65,0xB6,0x92,0x6C,0x70,0x48,0x50,0xFD,0xED,0xB
9,0xDA,0x5E,0x15,0x46,0x57,0xA7,0x8D,0x9D,0x84,0x90,0xD8,0xAB
,0x00,0x8C,0xBC,0xD3,0x0A,0xF7,0xE4,0x58,0x05,0xB8,0xB3,0x45,0
x06,0xD0,0x2C,0x1E,0x8F,0xCA,0x3F,0x0F,0x02,0xC1,0xAF,0xBD,0x0
3,0x01,0x13,0x8A,0x6B,0x3A,0x91,0x11,0x41,0x4F,0x67,0xDC,0xEA,
0x97,0xF2,0xCF,0xCE,0xF0,0xB4,0xE6,0x73,0x96,0xAC,0x74,0x22,0x
E7,0xAD,0x35,0x85,0xE2,0xF9,0x37,0xE8,0x1C,0x75,0xDF,0x6E,0x4
7,0xF1,0x1A,0x71,0x1D,0x29,0xC5,0x89,0x6F,0xB7,0x62,0x0E,0xAA,
0x18,0xBE,0x1B,0xFC,0x56,0x3E,0x4B,0xC6,0xD2,0x79,0x20,0x9A,0
xDB,0xC0,0xFE,0x78,0xCD,0x5A,0xF4,0x1F,0xDD,0xA8,0x33,0x88,0x
07,0xC7,0x31,0xB1,0x12,0x10,0x59,0x27,0x80,0xEC,0x5F,0x60,0x51,
0x7F,0xA9,0x19,0xB5,0x4A,0x0D,0x2D,0xE5,0x7A,0x9F,0x93,0xC9,0
x9C,0xEF,0xA0,0xE0,0x3B,0x4D,0xAE,0x2A,0xF5,0xB0,0xC8,0xEB,0x
BB,0x3C,0x83,0x53,0x99,0x61,0x17,0x2B,0x04,0x7E,0xBA,0x77,0xD
6,0x26,0xE1,0x69,0x14,0x63,0x55,0x21,0x0C,0x7D
};
```

The next function after Sbox is the shift rows operation is performed on the result. The process of shift rows involves shifting the data. The first row is unchanged, the second row is shifted left once, and the third row is shifted left twice and so on. The C code for this the other functions in AES algorithm mixcolumns and AddRoundKey is included as separate functions.

```
void Encode_ShiftRow(unsigned char* sTable)
{
    unsigned char temp;

    /* first row (row 0) unchanged */

    /* second row (row 1) shifted left by one */
    temp=sTable[1];
    sTable[1]=sTable[5];
    sTable[5]=sTable[9];
    sTable[9]= sTable [13];
    sTable [13]=temp;

    /* third row (row 2) shifted left by two */
    temp= sTable [2];
    sTable [2]= sTable [10];
    sTable [10]=temp;
    temp= sTable [14];
    sTable [14]= sTable [6];
    sTable [6]=temp;

    /* fourth row (row 3) shifted left by three (or right by one) */
    temp= sTable [3];
    sTable [3]= sTable [15];
    sTable [15]= sTable [11];
    sTable [11]= sTable [7];
    sTable [7]=temp;
}
#endif

void Encode_MixColumn(unsigned char* block)
{

for(i=0;i<16;i+=4)
                        {
                                aux1= block[i+0] ^ block[i+1];
                                aux3= block[i+2]^block[i+3];
                                aux = aux1 ^ aux3;
                                aux2= block[i+2]^block[i+1];

                                aux1 = xtime(aux1);
                                aux2 = xtime(aux2);
                                aux3 = xtime(aux3);

                                block[i+0]= aux^aux1^block[i+0];
                                block[i+1]= aux^aux2^block[i+1];
                                block[i+2]= aux^aux3^block[i+2];
                                block[i+3]=
block[i+0]^block[i+1]^block[i+2]^aux;
```

```
                }
        }
}

        /*   key addition */
        for(i=0;i<BLOCKSIZE;i++)
        {
                block[i] ^= key[i];
        }

        _roundCounter--;
    while(_roundCounter !=0);
}
#endif
```

APPENDIX C

FPGA Implementation

A two stage arithmetic pipeline deign consists of 16 bit adder and XOR gate is used to validate the technique proposed. The schematic view of the design is shown in Figure C.5. The two stage design is implemented in VHDL using Xilinx 14.7 IDE and targeted to Spartran 3E FPGA- xc3s500e-4fg320. The proposed technique uses delayed clock to capture the late arrival of data. DCM (Digital Clock Manager) is used to create the phase delayed clock. To create DCM: Project -> New Source -> IP (Core Generator and Architecture wizard. Figure C.1, Figure C.2 shows the DCM creation wizard. The input, output clock frequencies and the required signals are selected in the creation wizard.

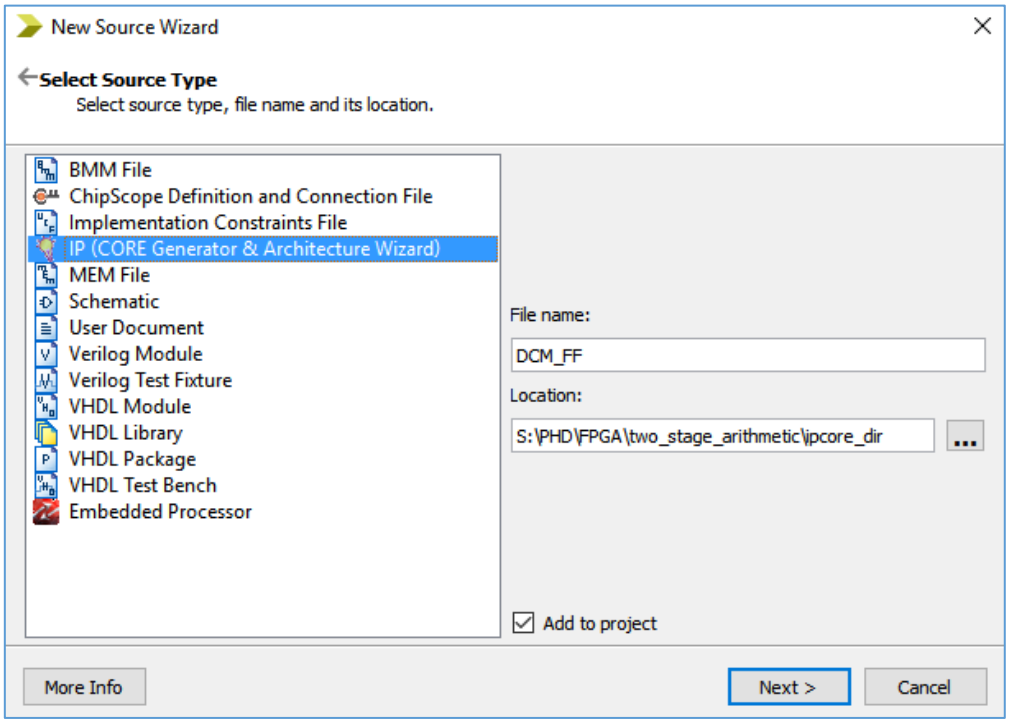


Figure C.1: DCM Creation Wizard

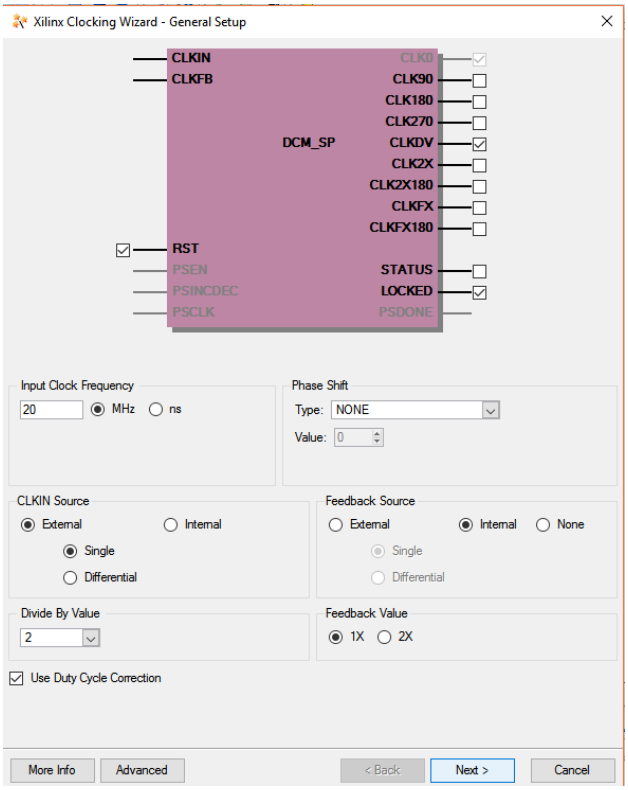


Figure C.2: DCM Creation Wizard: selection of input and out signals

Next, VHDL instantiation is done to use DCM in the original design. The divided clock signal (CLK_DIV) is given as the clock to the flip-flop in the case of errors. The selection can be done using a multiplexer. The DCM and selection circuit in VHDL is as follows:

```
component DCM is                    // VHDL Instantiation
port (
CLKFB : in std_logic;
CLKIN : in std_logic;
RST : in std_logic;
CLKDIV : out std_logic;
LOCKED : out std_logic;
end component;
begin
```

```
DCM port map (                          // port map in the design
    CLKFB => CLK,
    CLKIN => Sys_Clk,
    RST => Reset,
    CLKDIV => DCM_Clk;
    LOCKED => LOCKED);

    ff_clk <= Sys_clk when t_error = '1' else
    DCM_Clk;                            // Multiplexer design to change
clock to the flip-flop;
```

The implemented design is verified using Xilinx simulator ISim. The required input signal values are selected using the panel. The simulation is run and verified for the correctness and bit file is generated to program the FPGA. The same procedure is repeated for all other FPGA modules.

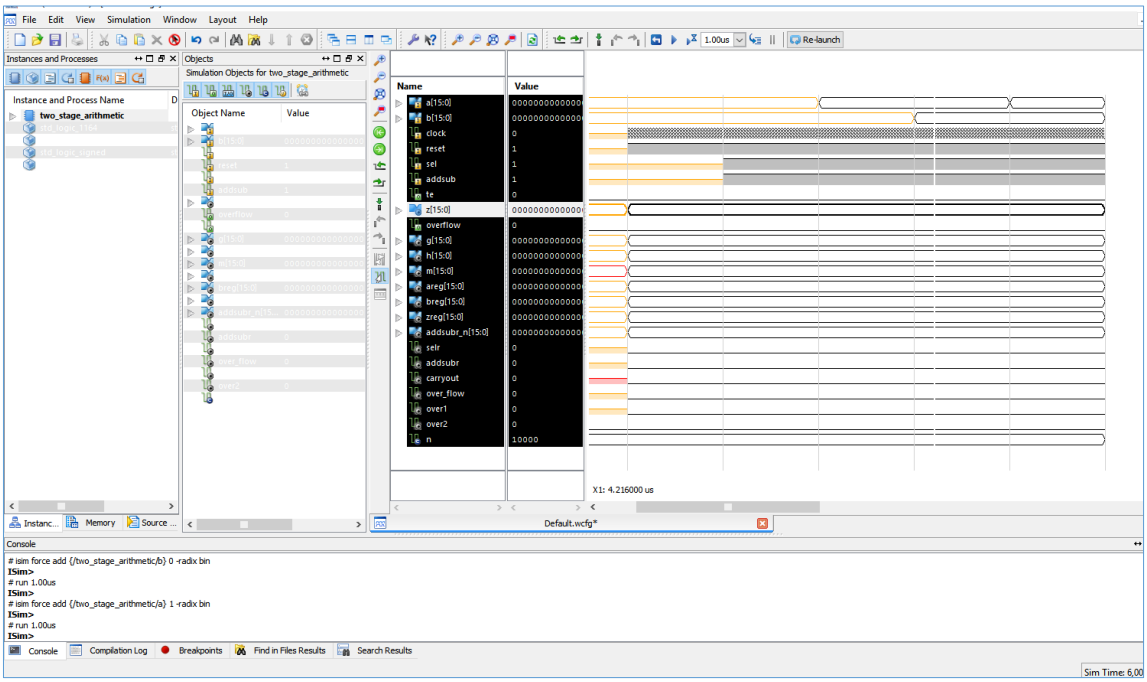


Figure C.4: Simulation waveforms using Xilinx Isim

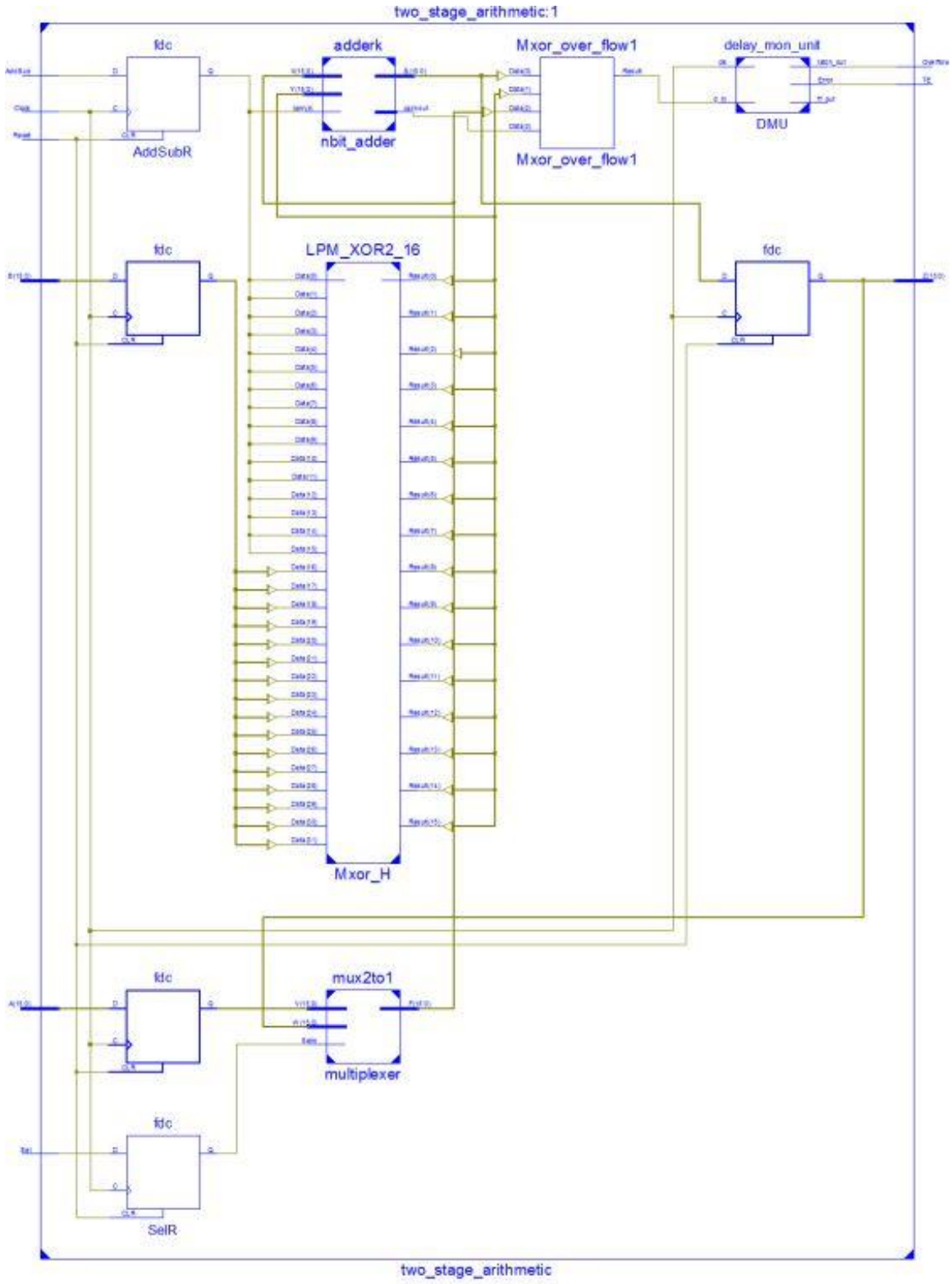


Figure C.5: Schematic of the two stage arithmetic design

APPENDIX D

Voltage Actuator

The Simulink model of the switching regulator with a DC-DC buck converter along with a digital closed loop system is shown in Figure D.1. A switching regulator is the most commonly used configuration in embedded systems, especially aggressive scaling systems, to have a smooth transition between different modes. It is essential for many reasons, including the fact that it enables a stable operation while providing fast response. The model has four main units, ADC, PWM, digital compensator and buck converter.

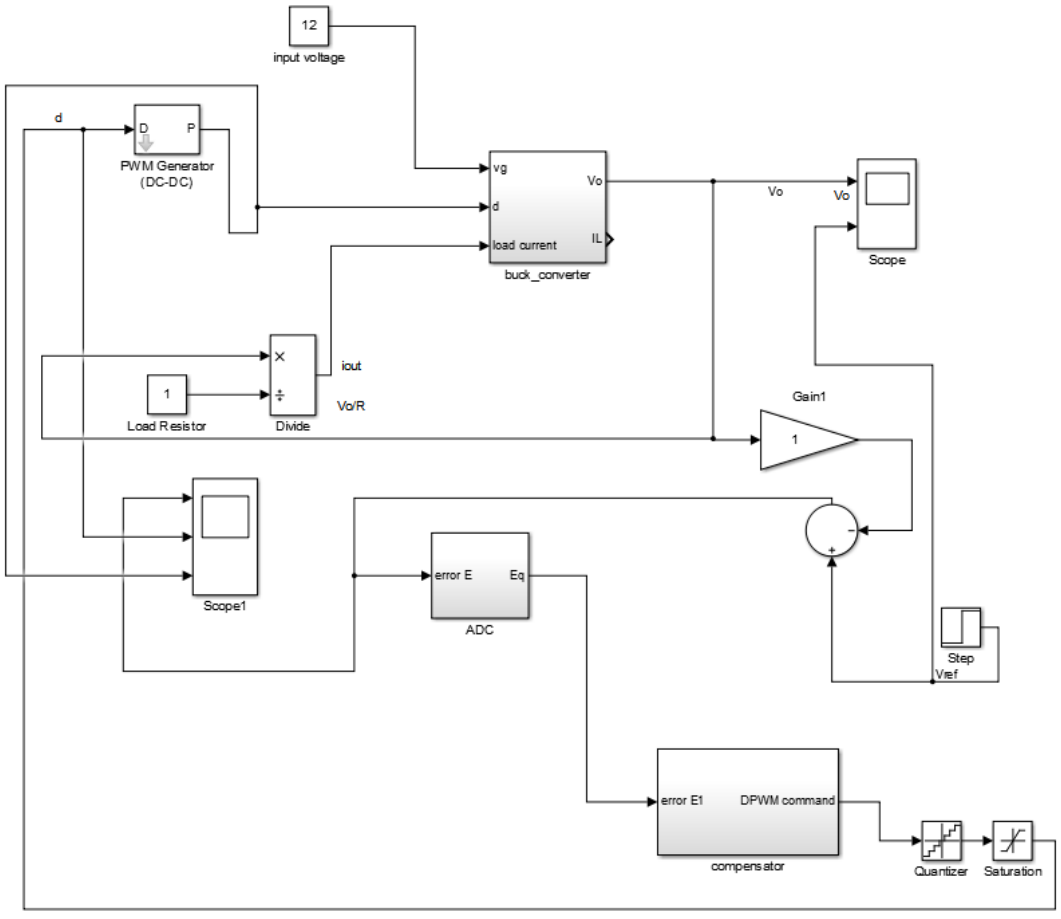


Figure D.1: Simulink model of switching regulator