2017

NOVEL RESOURCE EFFICIENT CIRCUIT DESIGNS FOR REBOOTING COMPUTING

Sai Subramanya Varun Thogarcheti
University of Kentucky, togarchetivarun@gmail.com
Author ORCID Identifier:
https://orcid.org/0000-0001-5442-6603
Digital Object Identifier: https://doi.org/10.13023/ETD.2017.474

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Sai Subramanya Varun Thogarcheti, Student

Dr. Himanshu Thapliyal, Major Professor

Dr. Cai-Cheng Lu, Director of Graduate Studies

NOVEL RESOURCE EFFICIENT CIRCUIT DESIGNS FOR REBOOTING COMPUTING

THESIS

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering
in the College of Engineering
at the University of Kentucky

By
Sai Subramanya Varun Thogarcheti
Lexington, Kentucky
Director: Dr. Himanshu Thapliyal
Lexington, Kentucky
2017

ABSTRACT OF THESIS

NOVEL RESOURCE EFFICIENT CIRCUIT DESIGNS FOR REBOOTING COMPUTING

CMOS based computing is reaching its limits. To take computation beyond Moores law (the number of transistors and hence processing power on a chip doubles every 18 months to 3 years) requires research explorations in (i) new materials, devices, and processes, (ii) new architectures and algorithms, (iii) new paradigms. The focus is on fundamental new ways to compute under the umbrella of rebooting computing. Therefore, this thesis highlights explicitly Quantum computing and Adiabatic logic, two new computing paradigms that come under the umbrella of rebooting computing. Quantum computing is investigated for its promising application in high-performance computing. The first contribution of this thesis is the design of two resource-efficient designs for quantum integer division. The first design is based on non-restoring division algorithm and the second one is based on restoring division algorithm. Both the designs are compared and shown to be superior to the existing work in terms of T-count and T-depth. The proliferation of IoT devices which work on low-power also has drawn interests to the rebooting computing. Hence, the second contribution of this thesis is proving that Adiabatic Logic is a promising candidate for implementation in IoT devices. The adiabatic logic family called Symmetric Pass Gate Adiabatic Logic (SPGAL) is implemented in PRESENT-80 lightweight algorithm. Adiabatic Logic is extended to emerging transistor devices such as FinFET, TFET and UTB-SOI.

KEYWORDS: Quantum Computing, Adiabatic Logic, T-Count, T-Depth, FinFET, TFET

Sai Subramanya Varun Thogarcheti

December 12, 2017

NOVEL RESOURCE EFFICIENT CIRCUIT DESIGNS FOR REBOOTING COMPUTING

By

Sai Subramanya Varun Thogarcheti

Dr. Himanshu Thapliyal

(Director of Thesis)

Dr. Cai-Cheng Lu

(Director of Graduate Studies)

December 12, 2017

(Date)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Computer Industry has fueled the information revolution over the past 50 years. Using personal computers, tablets and other smart-phones have become a part of everyday life. The rapid increases in the semiconductor technology and the implementation of complex computer architectures have enabled the computer performance to grow exponentially over the years. The 50-year reign of Moore's Law, with its exponential increase in integrated circuit density, has created this computer revolution. Moore's Law states that the number of transistors on a chip will double roughly every two years [9]. The chip industry has kept Moore's prediction alive until the last decade. However, due to limitations in operational performance, the progress in computational performance has substantially slowed down in the last ten years. The bounds on power dissipation of integrated circuits and increase in signal propagation delays have imposed the limitations on computer performance. Increasing the frequency to improve the performance of microprocessors had always been a trick followed by engineers. However, operating at a higher frequency came at an expense of increase in power. Operating frequency kept on increasing in the 1990's until the processors exceeded the 100W operating power level [10]. Exceeding the 100W power limit, would cause the circuit to self destruct. Therefore, increasing the frequency to

enhance the performance is no longer viable. Hence there is an urgent need to design new ways of computing [11].

Recognizing these problems, Institute of Electrical and Electronic Engineers (IEEE) has come up with a creative initiative called "Rebooting Computing". The focus of rebooting computing is on exploring fundamental new ways to compute. IEEE suggests that the next decade might see a "rebooting" of the entire computing industry, by redesigning the whole computer hardware and software from top to bottom [11]. Rebooting will enable continued growth of computing capabilities, keeping the computer revolution alive and well. Rebooting computing proposed spintronics, quantum computing, adiabatic and reversible computing as some of the promising fundamental new ways to compute. Hence, The focus of this thesis is to make significant contributions to quantum computing and adiabatic computing - the two new computing paradigms that come under the umbrella of rebooting computing.

Quantum computing is investigated for its promising application in high performance computing [12] [13]. Quantum computing focuses on theoretical computation systems that promise performance exponentially faster than any of today's computers. Quantum computing appears to be promising due to its applications in number theory, cryptography, search and scientific computation[12] [13]. There is a compelling need to design resource-efficient quantum circuits for arithmetic operations. Quantum circuits of arithmetic operations are needed to design quantum hardware for implementing quantum algorithms such as Shor's factoring algorithm, the discrete log problem, class number algorithm and triangle finding algorithm [14] [15]. Dividers are one of the significant computational units in quantum arithmetic [16] [14]. Integer division has applications in circuit designs of quantum algorithms, computation of power series, trigonometric functions [16–18]. This thesis presents two designs for quantum circuit integer division based on Clifford + T gates. The first quantum circuit is based on non-restoring division algorithm and the second one is

based on restoring division algorithm. Both of the designs seem to provide significant improvements when compared to the existing quantum division circuit.

The proliferation of IoT (Internet of Things) devices also has drawn interests to rebooting computing. The quality of life of individuals and societies would improve with the emergence of the Internet of Things (IoT). IoT has widespread applications in the field of manufacturing, automotive, medical, communication, finance, etc. IoT based devices such as Radio Frequency Identification (RFID) tags and smart cards are used to store and communicate secret or personal data over the Internet [19]. IoT devices such as RFID and smart cards have a constraint on power consumption and hardware resources. Further enhancements in IoT devices performance is only possible with advancements in low-power designing. Adiabatic logic is one of the rebooting computing paradigms that provides circuit design techniques used to design low-power hardware. Adiabatic logic can operate energy efficiently at low frequencies. Adiabatic logic design technique can also make the circuits resistant to powerful side-channel attacks such as Differential Power Analysis (DPA) attacks. Lightweight cryptography(LWC) is a subfield of cryptography which provides cryptographic solutions for resource-constrained IoT devices [20]. The properties of adiabatic logic can provide efficient solutions to the Lightweight Cryptographic circuits. Therefore, exploration of adiabatic logic in implementing the low-power LWC circuits for IoT devices is very essential. For this thesis work, we have explored the implementation of an adiabatic logic family - Symmetric Pass Gate Adiabatic Logic (SPGAL) in Lightweight cryptographic algorithm PRESENT-80. This application is extended to the emerging nanotechnology devices. SPGAL is implemented in FinFET, TunnelFET and UTB-SOI technologies.

1.1 Contribution of Thesis

This thesis presents resource-efficient designs in Quantum Computing and in Adiabatic Logic

1. Quantum division circuit based on Restoring division algorithm
2. Quantum division circuit based on Non-Restoring division algorithm
3. LWC based PRESENT-80 implementation in Adiabatic logic
4. Implementation of Adiabatic PRESENT-80 in Emerging Nano-technologies FinFET, TunnelFET and UTB-SOI.

1.2 Outline of Thesis

Chapter 2 provides a background on Quantum Computing and Adiabatic Logic. Chapter 3 presents designs of Quantum division circuits for the Restoring and Non-Restoring division Algorithms. Portions of Chapter 3 were previously published in [21]. Chapter 4 presents the implementation of PRESENT-80 algorithm in Adiabatic logic family called SPGAL. Chapter 5 presents the implementation of SPGAL in emerging nano-technology devices called FinFET, TunnelFET and UTB-SOI. Chapter 6 concludes the thesis. Portions of Chapters 4 and 5 were previously published in [22] and [23] (© [2017] IEEE) and [24] © 2017 ACM.

Chapter 2

Background

This chapter will cover any background information needed to understand the successive chapters. The main focus will be on the basics of Quantum Computing and Adiabatic Logic.

2.1 Quantum Computing

Among the emerging computing paradigms, quantum computing appears to be promising due to its wide applications in emerging technologies such as quantum dot cellular automata, cryptography, optical computing, etc. Quantum computation has seen vast progress over the years, both theoretically and experimentally. Quantum computing studies theoretical computation systems that makes direct use of quantum mechanical phenomena to perform operations on data [25]. A quantum computer operates by setting the qubits in a controlled initial state that represents the problem at hand by manipulating those qubits with a fixed sequence of quantum logic gates [26]. A quantum gate array is a set of these quantum logic gates with logical wires connecting their inputs and outputs. This definition of quantum gate arrays gives rise to completely reversible computation. Quantum circuits do not lose information during computation and quantum computation can only be performed when the system con-

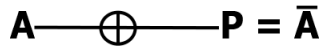

Figure 2.1: NOT gate

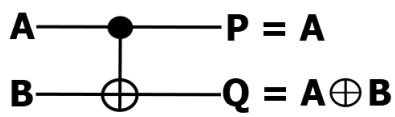

Figure 2.2: Feynman Gate

sists of quantum gates. Quantum circuits generate a unique output vector for each input vector, that is, there is a one-to-one mapping between the input and output vectors.

The quantum gates that are used for this thesis work are: NOT gate, Feynman gate, Toffoli gate and Peres gates.

2.1.1 The NOT Gate

NOT gate is a 1×1 gate. It is represented as shown in Fig. 2.1 [27].

2.1.2 The Feynman Gate

The Feynman gate also called $CNOT$ gate is a 2 input and 2 output gate with the mapping (A, B) to $(P = A, Q = A \oplus B)$. Here A and B are the inputs and P and Q are the outputs. The representation of Feynman gate is shown in Fig. 2.2 [27].

2.1.3 The Toffoli Gate

The Toffoli Gate is 3×3 reversible gate represented as shown in Fig. 2.3. The Toffoli Gate has a mapping of (A, B, C) to $(P = A, Q = B, R = A.B \oplus C)$ [27]. Here A, B and C are the inputs and P, Q and R are the outputs.

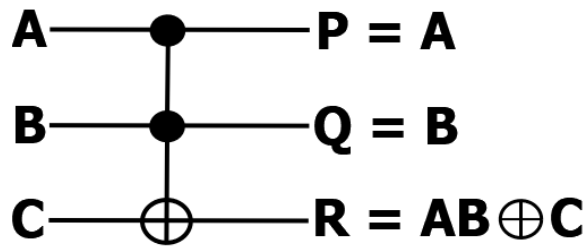

Figure 2.3: Toffoli Gate

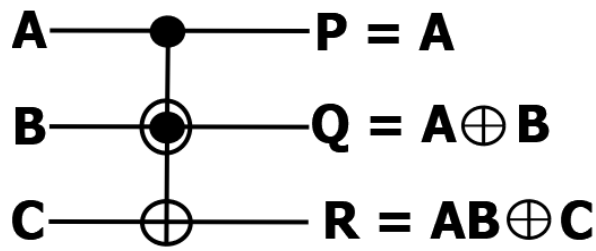

Figure 2.4: Peres Gate

2.1.4 The Peres Gate

The Peres Gate is 3×3 reversible gate represented as shown in Fig. 2.4. The Peres Gate has a mapping of (A, B, C) to $(P = A, Q = A \oplus B, R = A.B \oplus C)$ [27]. Here A, B and C are the inputs and P, Q and R are the outputs.

2.1.5 Clifford+T gates

Quantum computers of many qubits are extremely difficult to realize; thus, the number of qubits in the quantum circuits need to be minimized. The fabrication constraint of realizing quantum circuits with a large number of qubits has the objective of optimizing the number of ancilla qubits in a quantum circuits. Designing a scalable and reliable quantum computer is needed now as well as in the future; hence, fault-tolerant quantum circuits are required. Fault tolerant implementation of quantum circuits is gaining the attention of researchers because physical quantum computers are prone to noise errors. Fault tolerant implementations of quantum gates and quantum error correcting codes can be used to overcome the limits imposed by errors in

implementing quantum computing [28]. The most frequently used set of gates for this fault tolerant computation is the "Clifford+T" set of gates [29] [30]. Clifford+T gate family is illustrated in [1]. The NOT gate, Hadamard gate, T gate, Phase gate and CNOT gates constitute the Clifford+T set [8]. The definitions of these gates, their symbols and their matrix representations are shown in table 2.1. Using these gates will make the quantum circuits error-less.

Table 2.1: Definitions of Clifford $+T$ set gates [8]

Type of Gate	**Symbol**	**Matrix**
NOT	N	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Hadamard	H	$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
T gate	T	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i.\frac{\pi}{4}} \end{bmatrix}$
T gate Hermitian transpose	T^+	$\begin{bmatrix} 1 & 0 \\ 0 & e^{-i.\frac{\pi}{4}} \end{bmatrix}$
Phase	S	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
Phase gate Hermitian transpose	S^+	$\begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$
$CNOT$	C	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Clifford+T Implementation of Quantum Toffoli and Peres Gates

The Toffoli and Peres gates have to be implemented using the Clifford + T set. This subsection shows the functionality of these gates and explains their implementation in Clifford+T set.

Quantum Toffoli Gate:

The Toffoli gate, at times, can be vulnerable to errors. To avoid this, it can be implemented in terms of Clifford $+T$ set to make it fault tolerant. The T-gate implementation of Toffoli is shown in Fig. 2.5 [1].

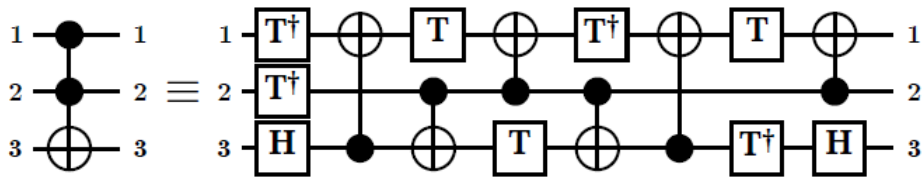

Figure 2.5: T gate implementation of Toffoli gate [1]

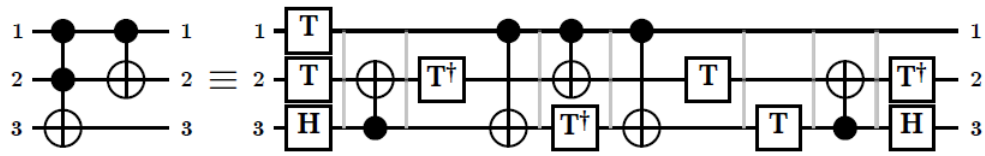

Figure 2.6: T gate implementation of Peres gate [1]

Peres Gate:

The T-gate implementation of Peres gate is shown in Fig. 2.6 [1].

2.1.6 Metrics Used for Evaluating Quantum Circuitry

In this thesis work, we shall be evaluating the quantum circuitry using the ancillaries, T-count and T-depth. Ancillaries are the number of ancilla qubits that are supplied to the circuit. T-count of a Clifford + T circuit is the total number of T and T^+ gates in the circuit. T-depth of a Clifford + T circuit is the number of levels in the circuit that contain one or more T and T^+ gates.[8]. The T-count of Toffoli gate shown in Fig. 2.5 is 7. T-depth of Toffoli gate is 4 [1]. The T-count of Peres gate shown in Fig. 2.6 is 7. T-depth of Peres gate is 4.

2.2 Adiabatic Computing

Adiabatic logic [31], [2] is one of the circuit design techniques used to design low-power and side channel attack-resistant hardware. Adiabatic logic can operate energy-efficiently at low frequencies, therefore it is one of the best candidate to implement low-power Lightweight Cryptography (LWC) circuits in IoT devices working at low

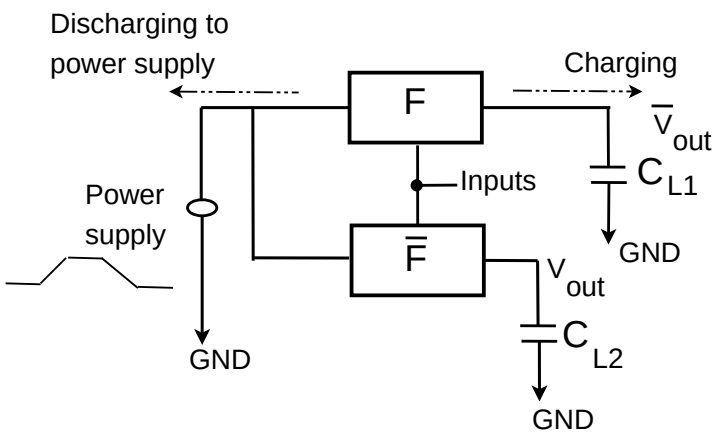

Figure 2.7: Adiabatic charging/discharging [2]

frequencies. A survey on side channel attack countermeasures for LWC has concluded that adiabatic logic is one of the promising techniques to design low-power and DPA-resistant hardware [32] [33].

Adiabatic logic uses power clocks to efficiently recycle the charge stored in the load capacitor [31]. Because of the recycling of the charge, adiabatic logic has reduced dynamic switching energy loss. Fig. 2.7 shows the energy recovery charging/discharging of the load capacitors. The energy dissipated in a energy recovery circuit when the charge is supplied through a constant current source is shown by

$$E_{diss} = \frac{RC}{T} C V_{dd}^2 \tag{2.1}$$

Where T is the charging/discharging time of the capacitor, C is the load capacitor and V_{dd} is the full swing of the power clock. If $T >> 2RC$ (time constant), the energy dissipated by the energy recovery circuit is less than the conventional CMOS circuit.

Adiabatic logic uses a time-varying voltage source and its slopes of transition are slowed down. This reduces the energy dissipation of each transition. In short the idea of adiabatic logic is to use a trapezoidal power-clock voltage rather than fixed supply voltage. As a consequence the power consumption of a circuit is reduced while at the same time its resistance against side-channel attacks is greatly enhanced. Low-power

```
generateRoundKeys()
for i = 1 to 31 do

    addRoundKey(STATE, Ki)
    sBoxLayer(STATE)
    pLayer(STATE)
endfor
addRoundKeys(STATE, K32)
```

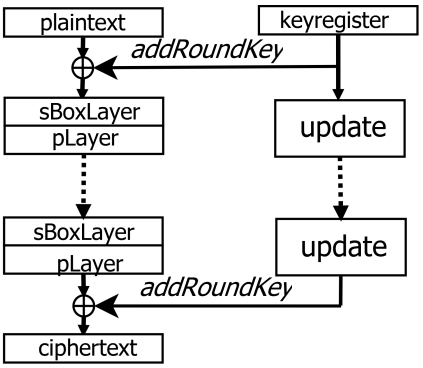

Figure 2.8: A top-level algorithmic description of the encryption routine of PRESENT [3]

adiabatic circuits could be especially valuable to implement in IoT devices such as RFID. To address the existing challenges in designing side channel attack countermeasure circuits for IoT devices, we considered the impact of adiabatic computing on the 64-bit input, 80-bit key based PRESENT algorithm.

2.2.1 PRESENT-80 Lightweight Algorithm

PRESENT [34] [3] is a light weight cipher which is designed for ultra constrained IoT devices such as RFID tags. PRESENT has obtained the ISO/IEC standard for lightweight cryptography. PRESENT can be employed in IoT circuits with minimal resources (1000 to 2000 Gate Equivalents). The PRESENT-80 algorithm is based on using S-Box as the main non-linear function. The block length of PRESENT is 64 bits and the length of key is 80 bits. Fig. 2.8 shows the algorithmic description of encryption routine for PRESENT-80 algorithm.

It can be seen from the Fig. 2.8 that algorithm is comprised of 31 rounds. Each of the 31 rounds is structured as follows:

AddRoundKey: The 64-bit plain-text is XORed with the 64-bit round key.

S-Box Layer: 16 4×4 identical S-Boxes are used in parallel as a non-linear substitution layer. All of the 16 S-Boxes comprise the S-Box layer.

P-Layer: After the S-Box layer, a permutation operation is performed to provide

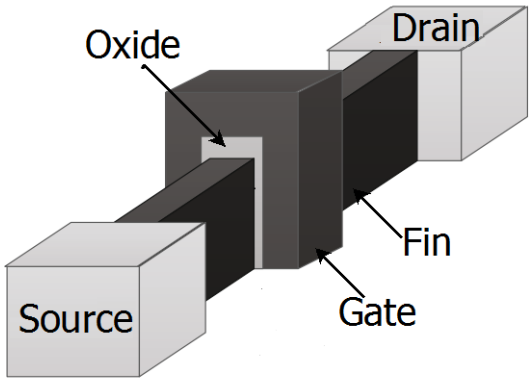

Figure 2.9: 3D structure of Short Gated FinFET

diffusion.

At the end of 31 rounds the final cipher text is produced at the output of addRoundKey. A key scheduler is used in the algorithm to provide the 64-bit round key from the 80-bit key.

2.2.2 FinFET

FinFET is a three dimensional structure that has a thin silicon body perpendicular to the plane of the wafer [35] [36]. The channel of the FinFET is wrapped by the gate in all three directions. FinFET provides strong gate control over the channels. This strong gate control over the channels reduces the short-channel effects, threshold current, and gate-dielectric leakage current when compared with MOSFETs [35]. Better gate control in FinFETs over MOSFETs results in higher on-state current, lower leakage, and faster switching speed.

FinFET devices come with two different modes of operation. In the Shorted Gate (SG) mode of FinFET, the back gate and front gate of the FinFET are tied together. FinFET acts as a three terminal device in SG mode. In the Independent Gate (IG) mode of FinFET, the front gate and back gate are connected to two different inputs. The SG mode is considered as a substitution for bulk CMOS and it has better

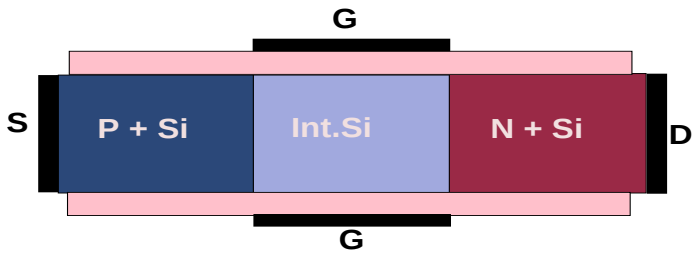

Figure 2.10: Physical structure of Homo-junction Tunnel FET [4]

performance as compared to the IG mode of FinFETs [36]. So, in this work, the SG mode FinFET implementation of Adiabatic logic is investigated. Fig. 2.9 shows the three dimensional structure of the SG mode FinFET device.

2.2.3 TunnelFET(TFET)

TFETs are emerging transistors that are considered to be a choice for low-power digital circuits. TFETs can have a subthreshold swing (SS) below 60 mV/dec, enabling a high on-current to off-current ratio. Lower SS enables TFET to have very low-leakage with higher performance than CMOS at lower voltages [37]. Among different types of proposed TFETs, III-V TFETs appear more promising due to their higher conduction current. In this work, we have used InAs homo-junction tunneling FETs for our simulations. Fig. 2.10 shows the physical structure of homo-junction Tunnel FET [4]. The advantage of TFET is it can operate at very low supply voltages.

2.2.4 Ultra-Thin-Body Silicon-On-Insulator (UTB-SOI)

Ultra-Thin-Body Silicon-On-Insulator (UTB-SOI) MOSFETs are considered to be a choice for low-power and low-leakage digital circuits [5]. The ultra-thin-body (UTB) structure of UTB-SOI (Fig. 2.11) eliminates the leakage paths between source and drain which provide a more evolutionary alternative to the vertical or surround-gate MOSFET. The thinner body in UTB-SOI results in lower leakage current. UTB-SOI device has the leakage current reduced by 10X for every nano meter drop in

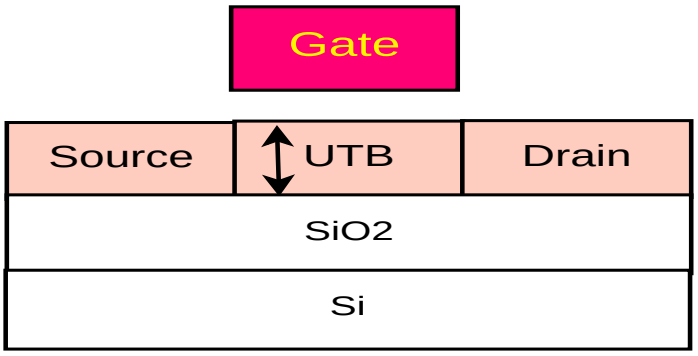

Figure 2.11: Physical structure of UTB SOI [5]

thickness of silicon [38]. Further, UTB-SOI can also support back-gating to change the threshold voltage, thereby further minimizing the leakage current. These advantages of UTB-SOI can be used to design cryptography circuits which have very low leakage power. So, in this work, we have exploited the useful properties of UTB-SOI in designing adiabatic logic family for IoT applications.

Chapter 3

Quantum Circuit Designs of Integer Division Optimizing T-count and T-depth

Quantum circuits of arithmetic operations are needed to design quantum hardware for implementing quantum algorithms such as Shor's factoring algorithm, the discrete log problem, class number algorithm and triangle finding algorithm [14] [15]. Dividers are one of the major computational units in quantum arithmetic and have applications in circuit designs of quantum algorithms [16] [14].

Quantum circuits that are based on Clifford+T gates can be made fault tolerant in nature permitting reliable and scalable quantum computation [29] [30]. The Clifford+T gate family is illustrated in [1]. The T gate is very costly to implement compared to the Clifford gates making reducing T-count and T-depth important optimization goals [30] [39]. Existing quantum hardware is limited in terms of number of available qubits [40]. Thus, ancillary qubits are a circuit overhead that needs to be kept to a minimum.

In the existing literature, there are a handful of integer divider designs based on

reversible gates targeting mostly reversible computing [41] [42] [43]. Among these designs we found only [44] to be suitable for quantum computing. The quantum integer division circuit in [44] implements the restoring division algorithm and uses the quantum Fourier transform to perform the division operation. However, the design in [44] is not optimized for T-depth and T-count. The quantum division circuit in [44] uses controlled phase shift gates. It is known that the controlled phase gates required by the design in [44] can only be approximated by Clifford+T gates [45]. The Clifford+T based approximations of the controlled phase gates have a high T gate cost [45]. Further, the T gate cost increases as the accuracy of the controlled phase gate approximation is improved [45]. Thus, implementing all the controlled phase gates required by the design in [44] with a high degree of accuracy will result in a design with high T-count and T-depth [45].

This chapter presents two designs for quantum circuit integer division based on Clifford+T gates. The first quantum circuit is based on the non-restoring division algorithm and the second quantum circuit is based on the restoring division algorithm. Both proposed quantum integer division circuits are based on (i) a new quantum conditional ADD operation circuit, (ii) a new quantum adder-subtractor and (iii) a new quantum subtraction circuit. Both the proposed restoring quantum integer division circuit and proposed non-restoring quantum integer division circuit are compared and shown to be superior to existing work in terms of T-depth and T-count.

3.1 Design of Quantum Circuits Used In Proposed Integer Division Circuits

The quantum circuits that are required for developing the proposed non-restoring and restoring integer division circuits are: (i) controlled adder-subtractor, (ii) quantum subtractor and (iii) conditional ADD operation circuit. The quantum circuit

designs of the quantum adder-subtractor, quantum subtractor and the conditional ADD operation circuit are discussed in the following sections.

3.1.1 Design of Quantum Subtractor

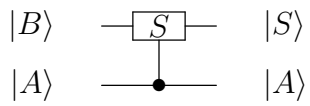

Figure 3.1: Graphic symbol of quantum subtractor. S represents the quantum subtraction operation

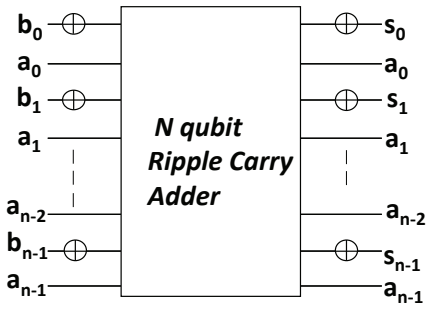

Figure 3.2: Circuit design of N qubit quantum subtractor based on N qubit quantum ripple carry adder

Fig.3.1 shows the symbol of the quantum subtractor circuit. The subtractor circuit takes two n qubit inputs $|A\rangle$ and $|B\rangle$. The input a is regenerated at the output. The n-qubit output $|S\rangle$ has the result of the subtraction of b and a. Fig.3.2 shows the circuit design of N qubit subtractor based on N qubit quantum ripple carry adder. As shown in Fig.3.2, a quantum ripple carry adder is required to develop a quantum subtractor circuit. We use the quantum ripple carry adder proposed in [46] for developing the quantum subtractor circuit. To perform subtraction, the input qubits $|B\rangle$ are complemented before being applied to the quantum ripple carry adder. Then, the ripple carry adder calculates $\bar{b} + a$. At the end of computation, the input

qubits $|B\rangle$ are complemented again. Thus, the quantum subtractor calculates $(\overline{\bar{b}+a})$ which is equivalent to $b-a$.

3.1.2 Design of Quantum Adder-Subtractor

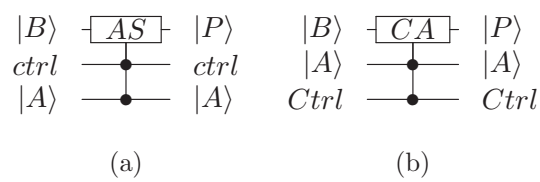

Figure 3.3: Graphic symbols of (a) Adder-Subtractor (b) Conditional ADD operation circuit. AS represents add or subtract operation. CA represents conditional add operation

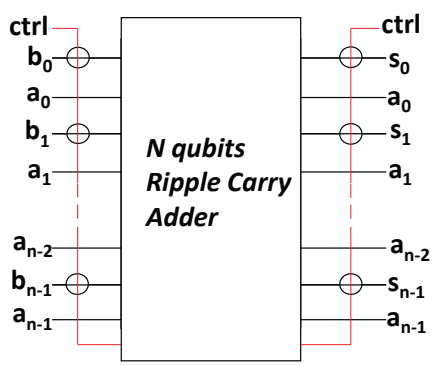

Figure 3.4: Circuit design of N qubit quantum adder-subtractor based on N qubit quantum ripple carry adder

Fig. 3.3(a) shows the graphic symbol of the quantum controlled addition or subtraction circuit. The quantum adder-subtractor circuit operates as follows: (i) when the input labeled *ctrl* is high (refer Fig. 3.3(a)), the circuit output is $|P\rangle = |B-A\rangle$, (ii) when the *ctrl* input is low, the circuit output is $|P\rangle = |B+A\rangle$.

The complete working circuit of the quantum adder-subtractor circuit is shown in Fig. 3.4. The quantum adder-subtractor circuit can be developed from an existing quantum ripple carry adder circuits such as those in [46] or [47]. We used the ripple carry adder in [46]. The quantum adder-subtractor calculates $(\overline{\bar{b}+a})$ when *ctrl* is

high. The expression $(\overline{\bar{b} + a})$ is equivalent to $b - a$.

3.1.3 Design of Quantum Conditional ADD Operation Circuit

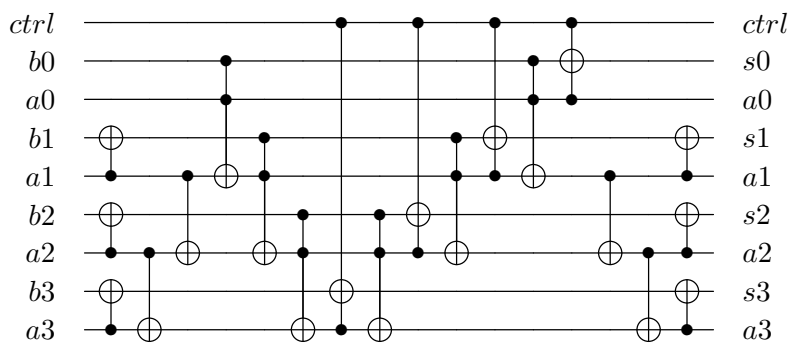

Figure 3.5: Circuit design of quantum conditional ADD operation circuit

Fig. 3.3(b) shows the graphic symbol of the quantum conditional ADD operation circuit. The quantum conditional ADD operation circuit operates as follows: (i) when the input labeled *ctrl* is high (refer Fig. 3.3(b)), the circuit output is $|P\rangle = |B + A\rangle$, (ii) when the *ctrl* input is low, the circuit output is $|P\rangle = |B\rangle$.

The complete working circuit of quantum conditional ADD operation circuit is shown in Fig.3.5 for 4 qubit operands. The quantum conditional ADD circuit uses a modified version of the ripple carry adder proposed in [46]. We were able to remove the qubit that performs the carry out for the adder in [46] as we do not need the carry out qubit in the proposed integer dividers. The addition architecture in [46] uses Peres gates to perform the addition. The Peres gate can be decomposed into a Feynman and a Toffoli gate. By replacing the Feynman gate with a Toffoli gate, we can use the control line (*ctrl*) to determine whether the conditional ADD circuit will perform addition or no operation. Although, Fig.3.5 is just shown for 4 qubit operands, it can easily be extended to any operand size.

3.2 Design of Non-Restoring Quantum Integer Division Circuit

The proposed non-restoring division algorithm for quantum circuits is shown in Table 3.1. In Table 3.1, the inputs to be given are: (a) ($|Q_{[0:n-1]}\rangle$, n qubit register in which the dividend is loaded; (b) $|D_{[0:n-1]}\rangle$, n qubit register in which the divisor is loaded; (c) $|R_{[0:n-1]}\rangle$, n qubit remainder register which is initiated to 0 at the start. At the end of computation, we get the quotient at $|Q_{[0:n-1]}\rangle$ and remainder at $|R_{[0:n-1]}\rangle$. The divisor is retained at the output. Also, $n+1$ garbage qubits are produced.

The quantum circuits that are required for developing the hardware implementation of the proposed non-restoring division algorithm are: (i) Leftshift operation circuit, (ii) controlled adder-subtractor, and (iii) conditional ADD operation circuit. We observed that we can eliminate the LeftShift operation circuit by combining $|R_{[0:n-2]}\rangle$ and $|Q_{[n-1]}\rangle$ to form an n qubit register there by saving the quantum resources.

The methodology to design our proposed quantum non-restoring integer division circuit is developed from the non-restoring division algorithm shown in Table 3.1. The Steps of the methodology are presented below.

3.2.1 Design Methodology for Quantum Non-Restoring Integer Division Circuit

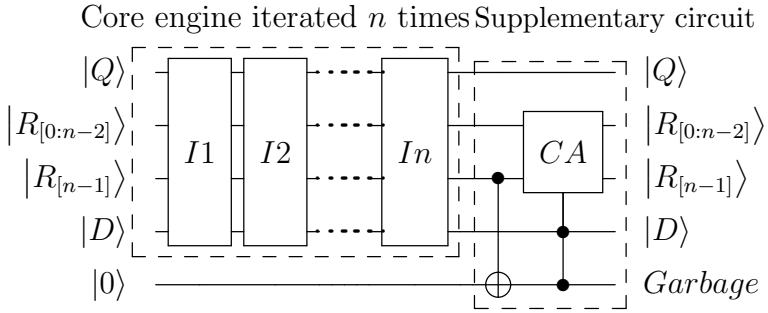

Figure 3.6: Quantum non-restoring integer divider circuit design

Table 3.1: Proposed quantum non-restoring division algorithm

Algorithm 1: *Proposed quantum non-restoring division algorithm*

```
function Non − Restore  (|Q_n⟩, |R_n⟩, |D_n⟩)
  for i = 0  to  n − 1 do
    /* Start Core Engine Phase */
    if(|R_[0:n−1]⟩ > 0)  then
      (|Q_[1:n−1]⟩, |R_[0:n−1]⟩) = LeftShift (|Q_[0:n−1]⟩, |R_[0:n−1]⟩);
      |R_[0:n−1]⟩ = |R_[0:n−1]⟩ + |D_[0:n−1]⟩;
    else
      (|Q_[1:n−1]⟩, |R_[0:n−1]⟩) = LeftShift (|Q_[0:n−1]⟩, |R_[0:n−1]⟩);
      |R_[0:n−1]⟩ = |R_[0:n−1]⟩ − |D_[0:n−1]⟩;
    end if;
    if(|R_[0:n−1]⟩ > 0)  then
      |Q_[0]⟩ = 1;
    else
      |Q_[0]⟩ = 0;
    end if;
    /* End Core Engine Phase */
  end for;
  //after n iterations//
    /* Start Supplementary Restoring Phase */
    if(|R_[0:n−1]⟩ > 0)  then
      |R_[0:n−1]⟩ = |R_[0:n−1]⟩;
    else
      |R_[0:n−1]⟩ = |R_[0:n−1]⟩ + |D_[0:n−1]⟩;
    end if;
    /* End Supplementary Restoring Phase */
  return R;
end function
```

From Table 3.1, we can see that the algorithm is divided into two phases. (i) Core Engine Phase and (ii) Supplementary Restoring Phase. The Core Engine Phase is iterated n times. Supplementary Restoring Phase takes place after the end of n iterations of the Core Engine Phase. The Supplementary Restoring Phase is repeated once. A quantum circuit is developed for each of these phases. The final circuit that performs the integer division using the non-restoring integer division algorithm is shown in Fig. 3.6. In Fig. 3.6, *I1* represents the first iteration of the Core Engine

Phase, *I2* represents the second iteration and *In* represents the final iteration.

Core Engine Phase

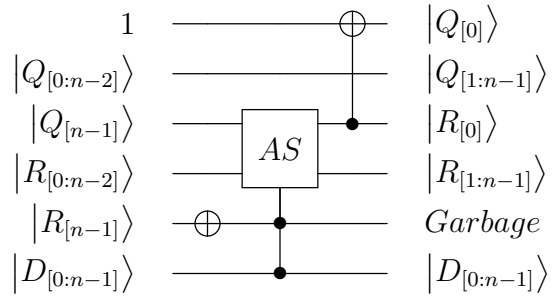

Figure 3.7: Quantum non-restoring integer divider circuit design for first iteration(core engine)

Fig. 3.7 represents the quantum circuit that does the operations that are marked under the Core Engine Phase in the algorithm in Table 3.1. We now elaborate on how the information moves in Fig. 3.7 .

- Step 1. $|D_{[0:n-1]}\rangle$ holds the divisor, $|R_{[0:n-1]}\rangle$ is initialised to zero, and $|Q_{[0:n-1]}\rangle$ holds the dividend.

- Step 2. We consider, $|Q_{[n-1]}\rangle$ and $|R_{[0:n-2]}\rangle$, as one combined register.

- Step 3. The combined register of Step 2 and $|D_{[0:n-1]}\rangle$ are applied as two n qubits inputs to the quantum adder-subtractor circuit. In Fig. 3.7, AS represents the adder-subtractor circuit. At the end of computation, register $|D_{[0:n-1]}\rangle$ emerges unchanged and the combined register now holds the sum or difference of the combined register and D.

- Step 4. Qubit $|R_{[n-1]}\rangle$ is complemented and applied as the *ctrl* qubit to quantum adder-subtractor circuit.

- Step 5. The *ctrl* qubit is left out as garbage.

- Step 6. An ancillary qubit set to 1 and qubit $|Q_{[n-1]}\rangle$ are applied to a CNOT gate. $|Q_{[n-1]}\rangle$ is the control qubit and 1 is the target qubit.

The Steps from 1 to 6 constitute the operations of the Core Engine Phase. From the algorithm in Table 3.1, it can be seen that Steps 2 to 6 of the Core Engine Phase are iterated n times. So, the circuit in Fig. 3.7 that represents the Core Engine Phase is also iterated n times (see Fig. 3.6). The outputs of the first iteration are given as inputs to the second iteration and so on for all n iterations.

Supplementary Restoring Phase

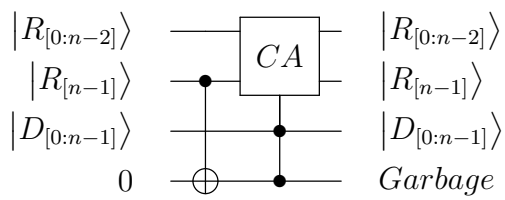

Figure 3.8: Quantum circuit implementation of the Supplementary Restoring Phase

After the end of n iterations of the Core Engine Phase, $|R_{[0:n-1]}\rangle$ might be negative at the end of n iterations. In that case, it has to be restored by adding the divisor. This restoration of the negative remainder is carried out by the Supplementary Restoring Phase quantum circuit shown in Fig. 3.8. The quantum circuit shown in Fig. 3.8 is the quantum implementation of the Supplementary Restoring Phase marked in the algorithm in Table 3.1. We now elaborate on how the information moves in the supplementary circuit.

- Step 1. The qubit $|R_{[n-1]}\rangle$ and an ancillary qubit set to 0 are applied as inputs to a CNOT gate. $|R_{[n-1]}\rangle$ is the control qubit and the ancillary qubit is the target qubit. The target now holds the value of $|R_{[n-1]}\rangle$.

- Step 2. The ancillary qubit is used as *ctrl* qubit to the conditional ADD operation quantum circuit.

- Step 3. Registers $|R_{[0:n-1]}\rangle$ and $|D_{[0:n-1]}\rangle$ are applied as inputs to conditional ADD operation quantum circuit. In Fig. 3.8, CA represents the conditional ADD operation circuit. $|D_{[0:n-1]}\rangle$ emerges unchanged and $|R_{[0:n-1]}\rangle$ will contain either the sum or emerge unchanged.

- Step 4. The control qubit $|R_{[0:n-1]}\rangle$ is left out as garbage.

- Step 5. After Step 4, we have the Quotient in $|Q_{[0:n-1]}\rangle$, and the remainder in $|R_{[0:n-1]}\rangle$. The divisor $|D_{[0:n-1]}\rangle$ is unchanged.

3.2.2 Cost Comparison With Existing Work

Table 3.2: Resource Count of Proposed Non-Restoring Algorithm Division Circuit

Designs	Adder-Subtractor	conditional ADD operation circuit	Non-Restoring Divider
T-count	$(14n - 14)$	$(21n - 14)$	$14n^2 + 21n - 28$
T-depth	8	16	$8 * n + 7$
Ancilla qubits	0	0	$2 * n + 1$

Table 3.3: Comparison of Resource Count Between Proposed and Existing Work

	1	Proposed	% impr. w.r.t. 1
T-count	$\approx 400n^2$	$14n^2 + 21n - 28$	$\approx 96\%$
T-depth	$130 * n$	$8 * n + 7$	$\approx 93\%$
Ancilla qubits	$2n$	$2 * n + 1$	-

1 is the work in [44]

The resources used in the design of the proposed quantum non-restoring integer division circuit is presented in Table 3.2. As shown in Table 3.2, the proposed design will require $2 * n + 1$ ancillary qubits. n ancillary qubits are used during initialization of remainder register and the remaining $n+1$ are transformed to garbage output. The

T-count required by the design is given by summing the cost of adder-subtractor and conditional ADD operation quantum circuit at each stage. T-count of the proposed quantum non-restoring integer division circuit is $14n^2 + 21n - 28$. The T-depth required by the design is given as $8 * n + 7$.

Comparison of resource costs between the proposed quantum non-restoring integer division circuit and the existing work is shown in Table 3.3. To calculate the T-count and T-depth for [44] we use T-count and T-depth values from approximate phase gate implementations reported in [45]. The implementations with the poorest accuracy are used. This is because the T gate cost increases significantly as a function of accuracy. Table 3.3 shows that the proposed quantum circuit of integer division has an improvement ratio of 93% in terms of T-depth, and 96% in terms of T-count.

3.3 Design of Restoring Quantum Integer Division Circuit

The proposed restoring division algorithm is shown in Table 3.4. In Table 3.4, the inputs to be given are: (a) ($|Q_{[0:n-1]}\rangle$, n qubit register in which the dividend is loaded ; (b) $|D_{[0:n-1]}\rangle$, n qubit register in which the divisor is loaded; (c) $|R_{[0:n-1]}\rangle$, n qubit remainder register which is initiated to 0 at the start. The algorithm repeats n times. At the end of n iterations, we get the quotient at ($|Q_{[0:n-1]}\rangle$ and the remainder at $|R_{[0:n-1]}\rangle$. The divisor is retained at the output.

The quantum circuits that are required for developing the hardware implementation of the proposed restoring division algorithm are (i) Leftshift operation circuit, (ii) n qubit quantum subtractor and (iii) Conditional ADD operation circuit. We observed that we can eliminate the LeftShift operation circuit by combining $|R_{[0:n-2]}\rangle$ and ($|Q_{[n-1]}\rangle$ to form an n qubit register which is actually equal to performing an left shift operation. By combining the qubits in this way, we do not have to use a

separate left shift operation circuit.

The methodology to design our proposed quantum restoring integer division circuit is developed from the restoring division algorithm shown in Table 3.4. The Steps of the methodology are presented below.

Table 3.4: Proposed Restoring division algorithm for quantum circuits

---

Algorithm 1 : *Proposed Restoring division algorithm*

---

function *Restore* $(|Q_n\rangle, |R_n\rangle, |D_n\rangle)$
  for $i = 0$ *to* $n-1$ **do**
    $(|Q_{[1:n-1]}\rangle, |R_{[0:n-1]}\rangle) = \textsc{Leftshift}\ (|Q_{[0:n-1]}\rangle, |R_{[0:n-1]}\rangle)$;
    $(|R - D_{[0:n-1]}\rangle = |R_{[0:n-1]}\rangle - |D_{[0:n-1]}\rangle$;
    if$(|R_{[0:n-1]}\rangle > 0)$ **then**
      $|Q_{[0]}\rangle = 1$
      $|R_{[0:n-1]}\rangle = |R - D_{[0:n-1]}\rangle$;
    else
      $|Q_{[0]}\rangle = 0$;
      $|R_{[0:n-1]}\rangle = |R - D_{[0:n-1]}\rangle + |D_{[0:n-1]}\rangle$;
    end if;
  end for;
//repeat for n iterations//
return R;
end function

3.3.1 Design Methodology for Quantum Restoring Integer Division Circuit

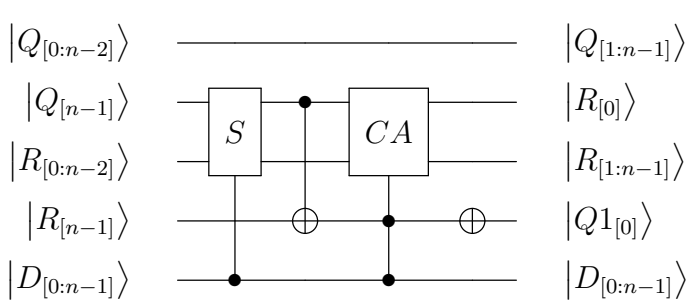

Figure 3.9: Quantum restoring integer divider circuit design for a single iteration

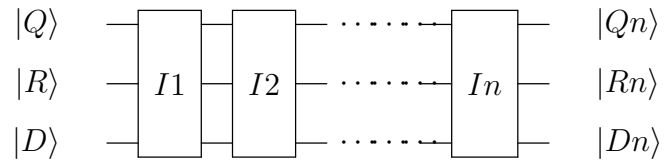

Figure 3.10: Quantum restoring integer divider circuit design(for n iterations)

Fig.3.9 shows the quantum circuit generated for the quantum restoring division circuit after 1 iteration of our design methodology. The Steps of the proposed methodology are repeated n times. Hence, the circuit in Fig. 3.9 is also iterated n times. This is done by using the outputs of the first iteration as inputs for the next iteration. Fig. 3.10 shows the complete quantum restoring division circuit where $I1$ represents the first iteration, $I2$ represents second iteration and In represents the final iteration. We now elaborate on how information moves through the circuit shown in Fig. 3.9.

- Step 1. The $|D_{[0:n-1]}\rangle$ holds the divisor, $|R_{[0:n-1]}\rangle$ is initialised to zero, and $|Q_{[0:n-1]}\rangle$ holds the dividend.

- Step 2. We consider, $|Q_{[n-1]}\rangle$ and $|R_{[0:n-2]}\rangle$, as one combined register.

- Step 3. The combined register mentioned above in Step 2, and $|D_{[0:n-1]}\rangle$ are given as inputs to the quantum subtractor circuit. Register $|D_{[0:n-1]}\rangle$ emerges unchanged. The combined register now holds the result of subtraction of R and D registers. Let us call this result as $|R - D_{[0:n-1]}\rangle$.

- Step 4. Qubits $|R - D_{[n-1]}\rangle$ and $|R_{[n-1]}\rangle$ are supplied to a CNOT gate. $|R - D_{[n-1]}\rangle$ is the control qubit and the $|R_{[n-1]}\rangle$ is the target qubit. The target now holds the value of $|R - D_{[n-1]}\rangle$ because $|R_{[n-1]}\rangle$ is always zero throughout the computation.

- Step 5. Qubit $|R_{[n-1]}\rangle$ is the control qubit to the conditional ADD operation circuit.

- Step 6. Registers $|R - D_{[0:n-1]}\rangle$ and $|D_{[0:n-1]}\rangle$ are the two n qubit inputs to the conditional ADD operation circuit. Register $|D_{[0:n-1]}\rangle$ emerges unchanged. The combined register will contain either the sum or emerge unchanged..

- Step 7. $|R_{[n-1]}\rangle$ is complemented.

Steps 2 through 7 are repeated n times. At the end of n iterations, the Quotient will be in $|Q_{[0:n-1]}\rangle$, the remainder in $|R_{[0:n-1]}\rangle$ and the divisor emerges unchanged.

3.3.2 Cost Comparison With Existing Work

Table 3.5: Resource Count of Proposed Restoring Division Circuit

	Subtractor	conditional ADD operation circuit	Restoring Divider
T-count	$(14n - 14)$	$(21n - 14)$	$35n^2 - 28n$
T-depth	8	16	$18 * n$
Ancilla qubits	0	0	n

Table 3.6: Comparison of Resource Count Between Proposed and Existing Work

	1	Proposed	% impr. w.r.t. 1
T-count	$\approx 400n^2$	$35n^2 - 28n$	$\approx 91\%$
T-depth	$130 * n$	$18 * n$	86.15%
Ancilla qubits	$2n$	n	50%

1 is the work in [44]

The resources used in the design of the proposed quantum restoring integer division circuit is presented in Table 3.5. As shown in Table 3.5, the proposed design will require n ancillary qubits during initialization of the remainder register. The T-count required by the design is given by summing the cost of subtractor and conditional ADD operation quantum circuit at each stage. T-count of the proposed quantum

restoring integer division circuit is $35n^2 - 28n$. The T-depth required by the design is given as $18 * n$.

Comparison of resource estimation between proposed quantum circuit of integer division and the existing quantum circuit of integer division in [44] is shown in Table 3.6. To calculate the T-count and T-depth for [44] we use T-count and T-depth from approximate phase gate implementations reported in [45]. The implementations with the poorest accuracy were used. This is because the T-count increases significantly as a function of accuracy. Table 3.6 showed that the proposed quantum circuit of integer division has an improvement ratio of 86.15% in terms of T-depth, and 91% in terms of T-count.

3.4 Conclusion

In this chapter, we have presented two designs for quantum circuit integer division based on Clifford+T gates. The first quantum circuit presented is based on the non-restoring division algorithm and the second quantum circuit presented is based on the restoring division algorithm. The design of sub-components used in the proposed quantum integer division circuits such as the quantum conditional ADD operation circuit, quantum adder-subtractor and quantum subtraction circuit are also shown. The proposed quantum integer division circuits are shown to be superior to existing designs in terms of T-depth and T-count. We conclude that the proposed non-restoring division circuit can be integrated in a larger quantum data path system design where T-count and T-depth are of primary concern. We also conclude that the proposed restoring division circuit can be integrated in a larger quantum data path system design to implement quantum algorithms where qubits are limited and T-count and T-depth must be kept to a minimum.

Existing quantum circuit implementations do not include the additional qubit

transformations that account for the available instruction set architecture, the hardware connectivity and layout constraints of a particular technology [48, 49]. For example, in trapped ion quantum computers (such as those presented in [50] and [51]) the ions are stored as a linear chain. Thus, interactions between qubits is restricted to at most two neighbors. Such constraints may significantly impact how quantum circuits are implemented in practice. The proposed quantum integer division circuit designs do not take into account technology constraints. However, the T-count and T-depth cost savings of our quantum integer division circuits are unaffected by these hardware considerations. To efficiently implement quantum algorithms, new designs need to be investigated for integer division that minimize the overhead imposed by technology constraints.

Chapter 4

Adiabatic Computing Based Low-Power and DPA-Resistant Lightweight Cryptography

Lightweight cryptography(LWC) is a subfield of cryptography that provides cryptographic solutions for resource-constrained IoT devices [20]. However, the secret or personal information stored and communicated through the LWC devices can be obtained through side-channel attacks [52]. Among the various side-channel attacks reported in the literature, the Differential Power Analysis (DPA) attack is considered to be one of the powerful side-channel attacks to reveal the secret information from the secure devices[53]. DPA attack reveals the secret key by correlating the instantaneous power consumed by the cryptographic device with the input data and the secret key. To guess the secret key, DPA uses statistical methods and evaluate the power traces with uniform plain texts. DPA requires no knowledge about the hardware implementation of the cipher and can be applied to any black box hardware implementation. These features of DPA makes it one of the powerful side channel attacks.

Various hardware related DPA countermeasures have been developed over the years, but none of these countermeasures are suitable to implement in resource constrained IoT devices [32] [54]. For example, a recent DPA-resistant implementation of the lightweight cryptography algorithm called PRESENT based on widely used DPA-resistant Wave Dynamic Differential Logic (WDDL) consumes at least 3X times more power than its CMOS based implementation [55].

Adiabatic logic [31] is one of the circuit design techniques used to design low-power and DPA-resistant hardware. *Adiabatic logic can operate energy-efficiently at low frequencies, therefore it is one of the best candidate to implement low-power LWC circuits in IoT devices working at low frequencies.* A survey on DPA countermeasures for LWC has concluded that adiabatic logic is one of the promising techniques to design low-power and DPA-resistant hardware [32] [33].

To establish the utility of adiabatic logic as a low-power and DPA-resistant solution for LWC, this thesis work investigates the Symmetric Pass Gate Adiabatic Logic (SPGAL) based implementation of the PRESENT-80 algorithm.

4.1 Symmetric Pass Gate Adiabatic Logic (SPGAL)

Symmetric Pass Gate Adiabatic Logic (SPGAL) was recently proposed as a low-power and DPA-resistant solution for LWC based IoT devices [6]. Fig. 4.1 shows the general structure of SPGAL logic gates. F and $\bar{F}$ in Fig. 4.1 represent the logic function and its compliment in the SPGAL gates. In SPGAL gates, F and $\bar{F}$ are designed in such a way that the load capacitors are balanced. Transistors M1 and M2 are used to recover the charge from the load capacitances while M3 and M4 are used to discharge the redundant charge present in the load capacitances before the evaluation of the next cycle of inputs.

Fig. 4.2 (a) shows the schematic of the SPGAL buffer. M3 and M4 form the

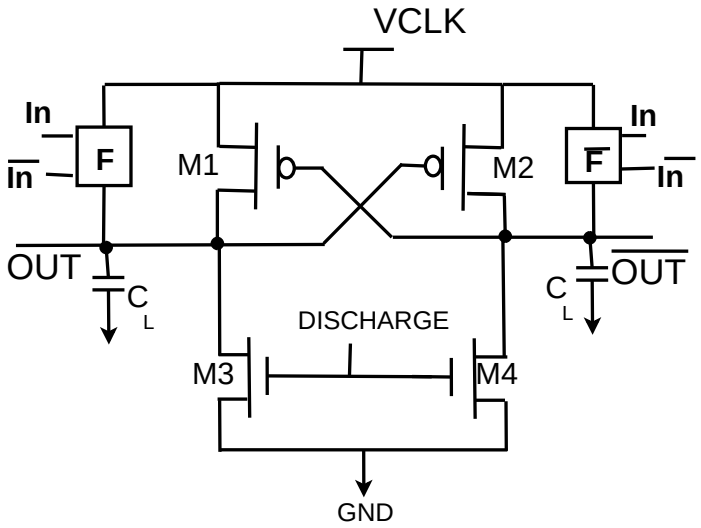

Figure 4.1: General structure of a SPGAL logic gate [6]

logic functions. M1 and M2 are used to recover the charge from the load capacitors. M5 and M6 are used to reset the outputs before the evaluation of the next cycle. Fig. 4.2 (b) shows the timing diagram of the SPGAL buffer. At T1, the inputs are passed to the SPGAL buffer. At T2, VCLK rises from GND to V_{dd} and the output load capacitors are charged through M3 or M4. At T3, VCLK will be at V_{dd} and the inputs will slowly fall back to ground. At T4, the charges present in the load capacitors is recovered back to VCLK through M1 or M2. However, V_{tp} charge in the load capacitors cannot be recovered back to VCLK which leads to information leakage. In the SPGAL design, the redundant charge is discharged to ground by using the discharge signal. Power clocks required for this circuit is generated by a dedicated circuit. Examples of such adiabatic clock generation circuitry are explained in [56].

To implement complex circuit designs in SPGAL, four trapezoidal clocks with each having a 90° phase shift with respect to its advance clock should be employed. Symmetric designs and resetting the outputs before the evaluation of next outputs make SPGAL gates more secure than the existing countermeasures against DPA attacks. Further, the SPGAL family is energy-efficient as compared to the existing adiabatic logic based DPA countermeasure circuits due to the reduction of non-adiabatic loss.

More details on this Symmetric Pass Gate Adiabatic Logic (SPGAL) can be found in [6]. SPFAL is one of the other secure adiabatic logic families [57].

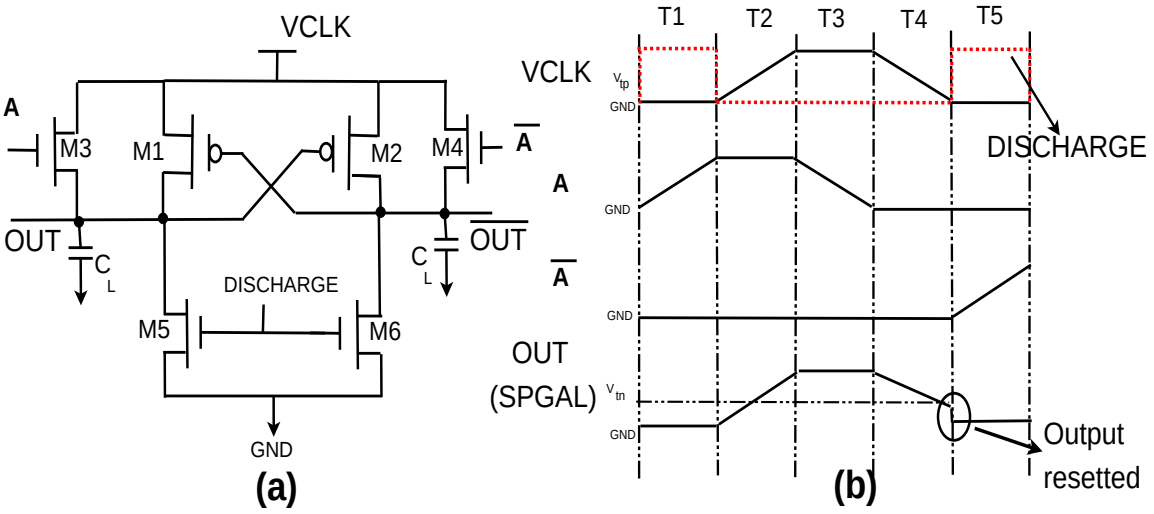

Figure 4.2: a) Schematic of SPGAL buffer b) Timing diagram of SPGAL buffer [6]

4.2 Implementation of PRESENT-80 Using Adiabatic Logic

Due to the higher power consumption and large area, CMOS-based DPA countermeasure circuits such as Sense Amplifier Based Logic (SABL) [58] are not suitable to implement in LWC devices. To protect the IoT devices against DPA attacks, an algorithmic countermeasure against DPA attack has been proposed in [59]. However, the countermeasure against DPA provided in [59] is not applicable for all LWC algorithms. As such, low-power adiabatic circuits could be especially valuable to implement in IoT devices such as RFID. To address the challenges in designing DPA countermeasure circuits for IoT devices, we considered the impact of adiabatic computing on the 64-bit input, 80-bit key based PRESENT algorithm.

Side-channel attacks based on DPA can be mounted on PRESENT to extract the keys. The existing countermeasures for DPA attacks are not suitable for circuits

working under energy constraints; for example, WDDL based PRESENT consumed 3X more power than its CMOS implementation [55].

4.2.1 SPGAL Implementation of PRESENT-80

In this section, we discuss the implementation of the PRESENT-80 algorithm using SPGAL gates. As discussed in the previous section, SPGAL is a low-power and DPA secure adiabatic logic family that uses four phase trapezoidal clocks to recover the energy from the load capacitors to the power clock. Four trapezoidal clocks with each having a 90° phase shift with respect to its advance clock are employed during the implementation. Note that in adiabatic circuits, the output of each gate is valid after one phase cycle of the clock and therefore it is possible to connect the circuits in a sequential manner.

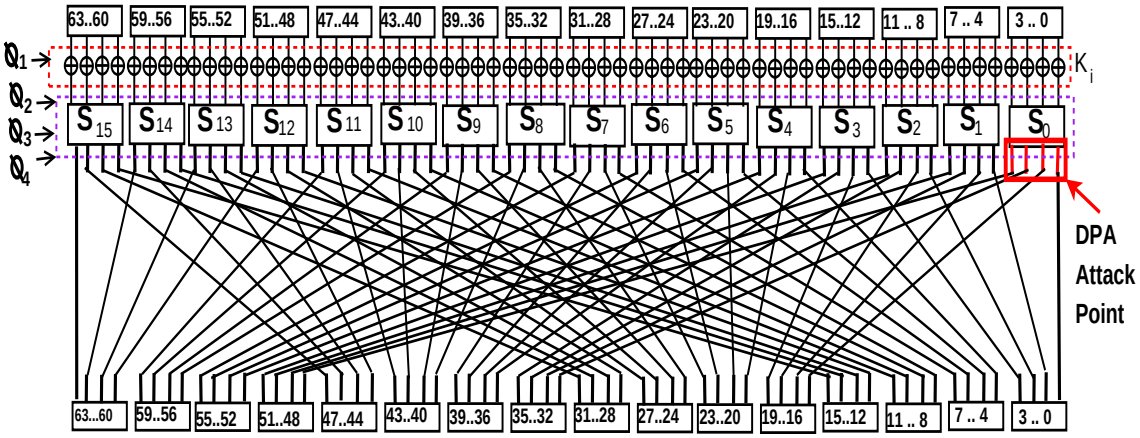

Figure 4.3: One round implementation of PRESENT-80 using SPGAL gates

Fig. 4.3 shows one round of the PRESENT-80 algorithm with a four phase clocking scheme. In our design of PRESENT-80, AddRoundKey is implemented with the first phase ($\phi 1$) of the clock while the PRESENT 4×4 S-Box is implemented with $\phi 2$, $\phi 3$ and $\phi 4$ as shown in Fig. 4.3. Fig. 4.4 shows the four phase clocks which are used to implement PRESENT-80 using SPGAL gates.

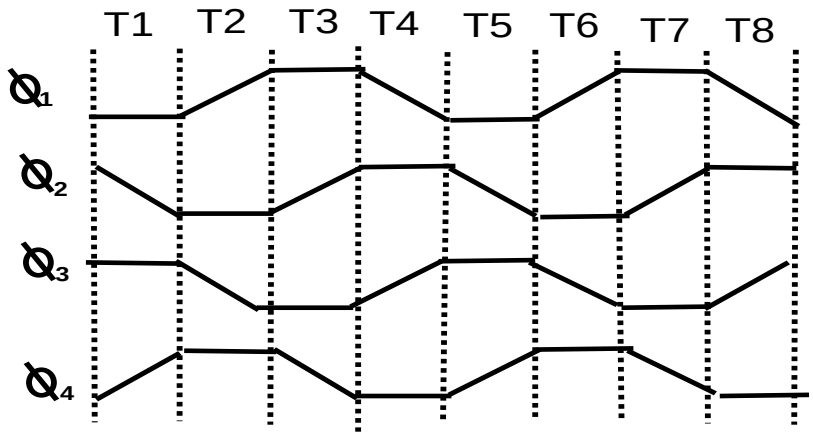

Figure 4.4: 4 phase clocking scheme to implement PRESENT-80

4.2.2 Implementation Results of PRESENT-80

In this work, we have implemented PRESENT-80 using SPGAL logic gates. For comparison purposes, we have implemented PRESENT-80 using CMOS gates and Sense Amplifier Based Logic (SABL) gates. SABL is one of the prominent CMOS based circuit level DPA countermeasure circuits in the literature [58]. Simulations are performed in SPECTRE simulator using $22nm$ CMOS bulk technology. All the circuits are simulated using SPECTRE simulator in PTM $22nm$ [60] technology with a V_{DD} of 1V. All of the simulations presented in this work are performed at 12.5 MHz which is close to the operating frequency of RFID (13.56 MHz).

Table 4.1 shows the implementation results of PRESENT-80 using SPGAL, CMOS and SABL logic gates. From Table 4.1, we can see that the SPGAL implementation of one round of PRESENT-80 has 83% improvement in terms of average power consumed and average current consumed as compared to its corresponding CMOS implementation. The SPGAL implementation of PRESENT-80 also has 82% of improvement in terms of average energy consumed as compared to the CMOS implementation. The comparison results in Table 4.1 also show that the SPGAL based implementation of one round of PRESENT-80 also has very high improvement results in all the metrics as compared to the SABL implementation. It has to be noted that the current con-

sumption of the SPGAL PRESENT-80 is the sum current of all the power supplies. The SPGAL based PRESENT-80 has reduced current consumption due to recovery of charge whereas in the conventional CMOS based PRESENT-80 the charges are discharged to ground leading to the additional current and power consumption.

Gate Equivalent (GE) represents the size of the circuit in terms of two input NAND gates. From our simulations, we found that the SPGAL based PRESENT implementation has 16% lesser GE count as compared to the SABL implementation of PRESENT-80. However, the SPGAL based PRESENT-80 has 38% more GE count than its CMOS equivalent as the CMOS-based design utilizes the single rail logic.

Table 4.1: Comparison of metrics between CMOS, SABL and SPGAL implementations of PRESENT-80

Metric	**CMOS**	**SABL** [58]	**SPGAL**	**% imp of SPGAL w.r.t CMOS**	**% imp of SPGAL w.r.t SABL**
Avg. power (μW)	7.890	15.30	1.32	83	91
Avg. current (μA)	7.954	15.33	1.35	83	91
Avg. energy (pJ)	20.83	40.46	3.564	82	90

4.3 DPA Attack on PRESENT-80

Although CMOS and emerging transistors based SPGAL show better performance in terms of energy and power consumptions, it is important to validate their security against DPA attack. When considering the DPA attack, it is essential to identify the intermediate blocks to perform the DPA attack. In this work, a DPA attack is performed on the output of the PRESENT S-Box (S-layer) as shown in Fig. 4.3. We have performed the DPA attack as per the steps described in [61]. Simulations are performed at 12.5 MHz. Fig. 4.5(a) shows a successful DPA attack on the CMOS

based PRESENT-80 design. In DPA attacks, usually a large number (greater than 100,000) of input plain texts are fed to the crypto processor. However, in this thesis, we performed a simulation based DPA attack without any electrical noises. Moreover, the benchmark PRESENT-80 core does not have other analog and digital modules of the crypto processor that consume additional current. Therefore, for the CMOS-based implementation of the PRESENT-80 algorithm, the secret key was revealed using fewer traces (5233 input traces).

Further, It has been shown in Fig.4.5(b) that the DPA attack was unsuccessful for the SPGAL based PRESENT-80. From our simulation results, the secret key was not revealed in the SPGAL based PRESENT for more than 50,000 input traces.

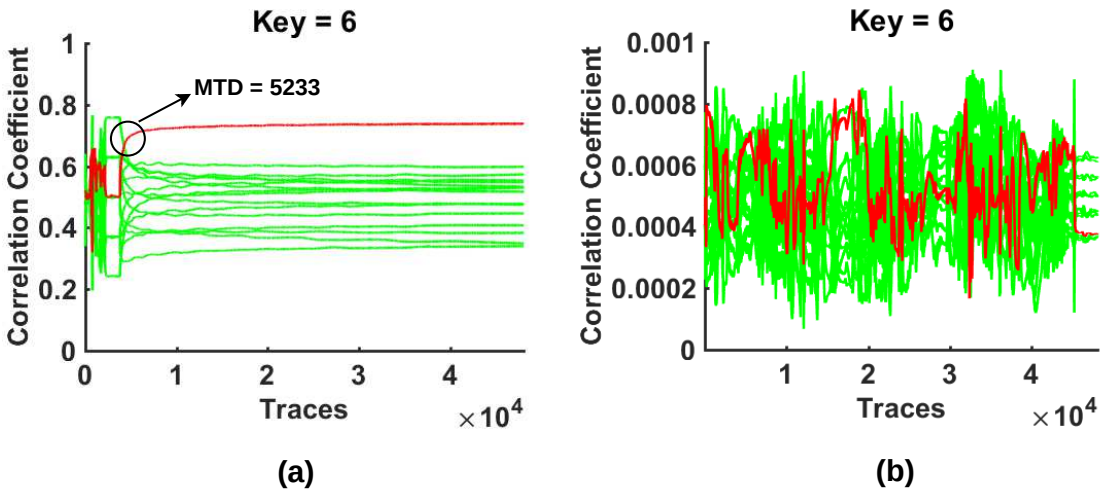

Figure 4.5: DPA attack results of PRESENT implemented using a) CMOS gates b) SPGAL gates

4.4 Conclusion

In this work, we have demonstrated adiabatic computing as a promising platform for low-power and LWC in IoT devices. PRESENT-80 Lightweight algorithm has been used as the benchmark circuit for this thesis work. From the simulation results, it is shown that the SPGAL based PRESENT-80 consumes less current, less power and is

more energy-efficient in comparison to its equivalent CMOS-based and SABL-based implementations. It is also demonstrated that SPGAL circuits are more resistant to DPA attacks as compared to their equivalent CMOS circuits. Improvement in power dissipation along with security against DPA makes the adiabatic computing (SPGAL) an ideal candidate to implement IoT based devices where power consumption and security are major concerns. The low-power and DPA-resistance properties of the adiabatic based PRESENT benchmark circuit have opened avenues for the low-power and DPA-resistant implementation of lightweight cryptographic algorithms for IoT devices.

Chapter 5

Adiabatic Computing for Emerging Nanotechnologies

Entering the smart society today, the amount of the information and data is growing explosively. Corresponding to the growth, demands for low-power, high-performance integrated circuits become even stronger. The slowdown of Moores law intensifies the search of the next transistor and memory technologies beyond CMOS. For conventional MOS structure, as the channel length shrinks, the gate does not have full control over the channel which is not desirable. One of its effects is to cause more sub-threshold leakage from drain to source, which is not good from power consumption point of view. In conventional MOS, the gate cannot control leakage path. This can be improved using various MOS structures which allow the scaling of a transistor beyond conventional MOS scaling limit [62]. Several emerging transistor devices are proposed in the last decade. This emerging transistor devices extends Moore's law, allowing semiconductor manufacturers to create CPUs and memory modules that are smaller, perform faster, and consume less energy. FinFET, TFET (Tunnel FET) and UTB-SOI (Ultra Thin Body - Silicon on Insulator) are some of the most promising emerging nanotechnologies. In this section, we will discuss the adiabatic

implementation of FinFET, TFET and UTB-SOI in PRESENT-80 benchmark circuit for Lightweight cryptography.

5.1 SPGAL Implementation in FinFET, TFET and UTB-SOI

FinFET-SPGAL and TFET-SPGAL are implemented in $20nm$ technology. The results are compared with $22nm$ CMOS-SPGAL. Further, leakage analysis is also performed on UTB-SOI.

5.1.1 FinFET Based SPGAL (FinSAL)

In this work, the Short Gated (SG) mode FinFET implementation of SPGAL gates are investigated for LWC. In this work, we have used Predictive Technology Model for $20nm$ FinFETs for simulation [60]. Since SG mode is considered as the substitution for bulk CMOS, the MOSFETs are replaced by SG FinFETs. The FinFET implementation of SPGAL (FinSAL) has outperformed the CMOS based SPGAL gates in terms of power consumption and security in terms of resistance against DPA attacks. FinSAL has been recently proposed in [7]. Fig. 5.1 shows the FinFET based SPGAL (FinSAL) implementation of XOR gate. The FinSAL XOR gate consumes less energy as compared to the FinFET based conventional XOR gate due to recovery of charge in each phase of clock cycle. Fig. 5.2 shows the uniform current consumption of the FinSAL XOR gate for various input transitions. The uniform current consumption of FinSAL XOR gates shows that FinSAL gates can counteract DPA attack at cell level. Low operating voltage, low-power consumption and uniform current consumption irrespective of input data of FinSAL gates makes it suitable to implement in LWC for IoT applications. This motivated us to investigate the FinSAL gates for use in LWC to design low-power and DPA-resistant IoT devices. More details on FinSAL can be

found in [7].

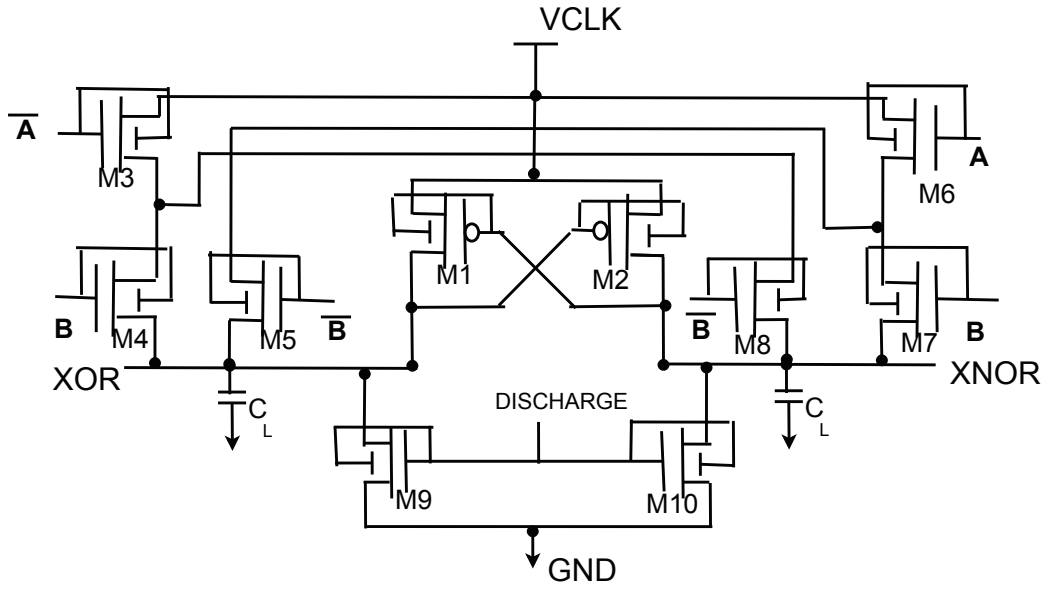

Figure 5.1: Schematic of FinSAL XOR gate [7]

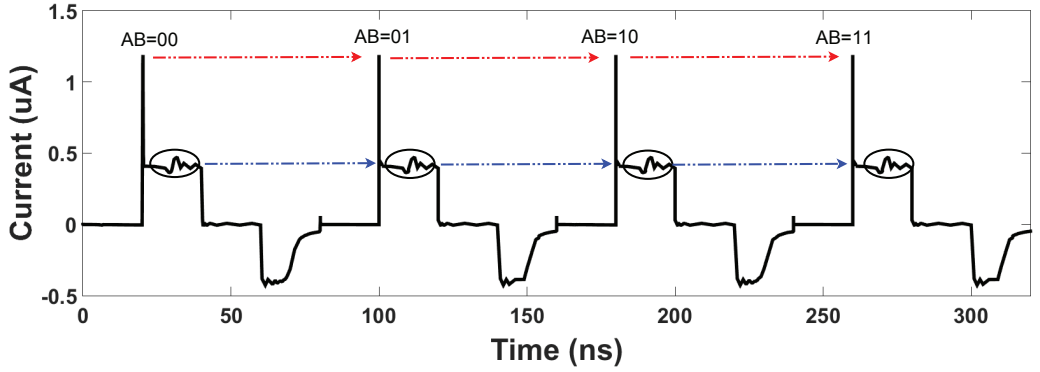

Figure 5.2: Uniform current consumption of FinSAL XOR gate

5.1.2 TFET Based SPGAL (TunSAL)

In this work, we have used InAs homo-junction tunneling FETs for our simulations. We have investigated the advantages of SPGAL gates with Tunnel FET (TFET). TFET based SPGAL gates are referred as TunSAL in this work. Fig. 5.3 shows TunSAL XOR gate. TunSAL XOR gate has balanced load capacitance with symmetric design similar to the CMOS counterpart. In TunSAL gates, F and $\bar{F}$ (refer Fig. 4.1)

are replaced by N type TFETs and the charge recovery path in SPGAL designs are replaced by P type TFETs. For energy recovery designs, it is critical to determine the supply voltage with different transistors and different technology nodes. The advantage of TFET is it can operate at very low supply voltages. In this thesis, the supply voltage of 0.3V has been used to simulate the TFET based circuits at $20nm$ technology. PTM technology model files have been used [60]. With the scaling of supply voltage, TunSAL circuits have reduced power consumption as compared to the CMOS based SPGAL circuits. Fig. 5.4 shows the uniform current consumption of the TunSAL XOR gate for various input transitions. The uniform current consumption of TunSAL XOR gates shows that TunSAL gates can counteract DPA attack at cell level. Low operating voltage, low-power consumption and uniform current consumption irrespective of input data of TunSAL gates makes it suitable to implement in LWC for embedded computing devices. For the purpose of fair comparison, we have compared the TunSAL with FinSAL (FinFET based SPGAL) and CMOS-SPGAL based circuits.

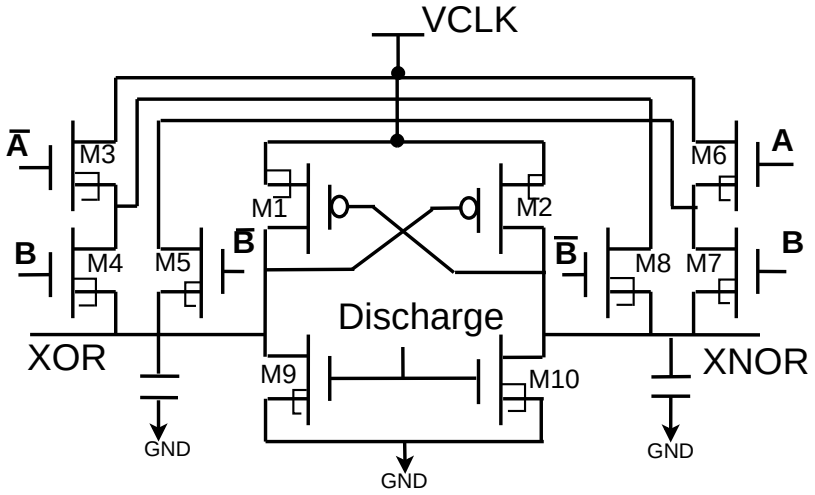

Figure 5.3: Schematic of TunSAL XOR gate

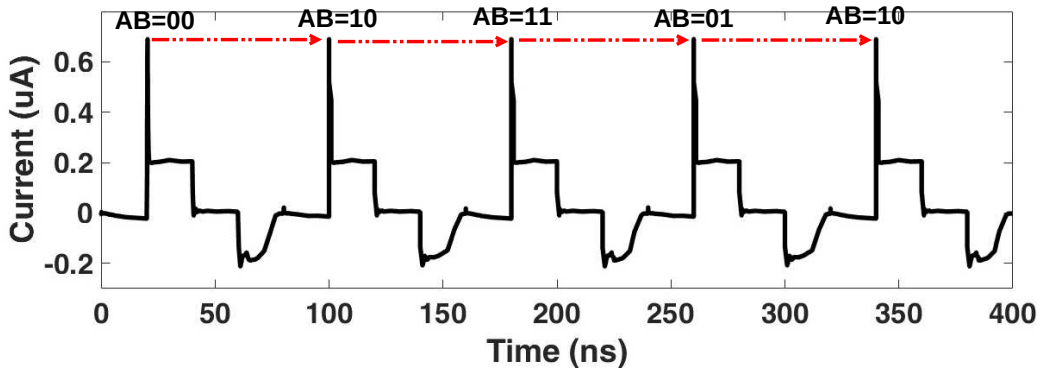

Figure 5.4: Uniform current consumption of TunSAL XOR gate

5.1.3 Analysis of FinSAL and TunSAL

In this work, CMOS and FinFET based circuit simulations are performed in Cadence Virtuoso using PTM model files [60]. An input voltage of 1V is used for simulating CMOS and CMOS-SPGAL gates at $22nm$. An input voltage of 0.9V is used for FinFET gates at $20nm$ and TFET gates are simulated at 0.3V at $20nm$ using [63].

The security parameter Normalized Energy Deviation (NED) is used to indicate the percentage difference between minimum and maximum energy consumption for all possible input transitions. Normalized Standard Deviation (NSD) indicates the energy consumption variation based on the inputs. Table 5.1 shows the simulated and calculated results of the CMOS-SPGAL XOR gate compared with FinSAL and TunSAL XOR gates. From Table 5.1, it can be inferred that FinSAL and TunSAL XOR gates have very negligible NED values. The reason for this lower NED values is the uniform current consumption of SPGAL designs. Further, the TunSAL XOR gate has reduced energy consumption as compared to the CMOS-SPGAL and FinSAL XORgates. Fig. 5.5 helps us to understand the relation between the energy deviation (NED) and the supply voltages for each device with energy recovery computing. With lowering of supply voltages, the TunSAL-XOR gate offers more security as it has minimum energy deviation. We can also infer from Fig. 5.5 that, CMOS-SPGAL XOR gate does not function properly for voltages less than 0.6 V. Similarly, FinSAL

XOR gate fails to function correctly for voltages less than 0.5 V. Hence, the NED values for FinSAL (less than 0.5 V) and CMOS-SPGAL (less than 0.6 V) are not presented in Fig. 5.5. Further, we can see that TunSAL XOR gate has very negligible energy deviations from 0.2 - 0.5 V. FinSAL shows superior performance compared to CMOS-SPGAL from 0.5 V to 1 V. Low energy deviations makes TunSAL and FinSAL gates excellent candidates for LWC applications.

Table 5.1: Simulated and calculated results of CMOS-SPGAL XOR gate and FinSAL XOR gate compared with TunSAL XOR gate

Logic family	SPGAL	FinSAL	TunSAL
Device	MOSFET	FinFET	TFET
Technology	$22nm$	$20nm$	$20nm$
V_{DD}(V)	1	0.9	0.3
E_{min}(fJ)	0.266	0.058	0.044
E_{max}(fJ)	0.268	0.060	0.046
E_{avg}(fJ)	0.267	0.059	0.045
NED (%)	0.500	0.211	0.014
NSD(%)	0.200	0.100	0.030

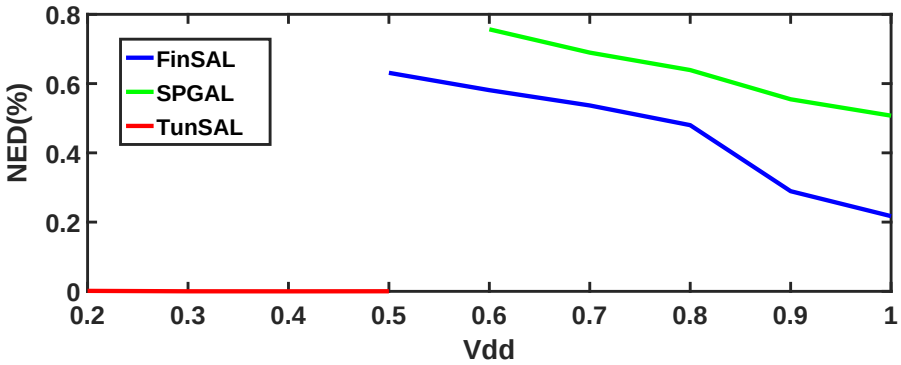

Figure 5.5: NED as a function of supply voltage

5.1.4 UTBSOI Based SPGAL

In this work, all the UTB-SOI based simulations are performed in Cadence Virtuoso using BSIM model files [64]. An input voltage of 1.8V is used for simulating CMOS

Table 5.2: Simulated and calculated results of SPGAL XOR and AND gates compared with adiabatic UTB SOI XOR and AND gates

Logic family	CMOS SPGAL-XOR	UTB SOI SPGAL-XOR	CMOS SPGAL-AND	UTB SOI SPGAL-AND
NED (%)	0.016	0.001	0.09	0.005
NSD(%)	0.008	0.0005	0.03	0.002
Avg. Leakage power(nW)	7.63	3.02	11.52	7.15

and CMOS-SPGAL gates. UTB-SOI SPGAL gates are simulated at 1.5V. NED and NSD values have been calculated for both bulk CMOS SPGAL and UTB-SOI SPGAL based XOR and AND gates. To prove that UTB-SOI consumes low leakage power, we have also compared the average leakage power for both bulk CMOS SPGAL and UTB-SOI SPGAL based XOR and AND gates

Table 5.2 shows the results of the bulk CMOS SPGAL XOR and AND gates compared with UTB-SOI SPGAL based XOR and AND respectively. From Table 5.2, we can infer that both bulk CMOS SPGAL and UTB-SOI SPGAL gates have very negligible energy deviations. However, UTB-SOI SPGAL based XOR and AND gates saves 60% and 38% of average leakage power as compared to bulk CMOS SPGAL based XOR and AND gates respectively.

5.2 FinFET, TFET and UTB-SOI implementations of PRESENT-80

In this work, we have implemented PRESENT-80 using FinFET, TFET and UTB-SOI based SPGAL logic gates. The results are compared with CMOS-SPGAL gates. All of the simulations presented are performed at 12.5 MHz which is close to the operating frequency of RFID (13.56 MHz).

From the comparison results in Table 5.3, we can see that FinSAL and Tun-

Table 5.3: Comparison of metrics between CMOS-SPGAL, FinSAL and TunSAL implementations of PRESENT-80

Metric	**CMOS-SPGAL**	**FinSAL**	**TunSAL**	**% imp of FinSAL w.r.t CMOS-SPGAL**	**% imp of TunSAL w.r.t CMOS-SPGAL**
Device	MOSFET	FinFET	TFET	-	-
Tech.(nm)	22	20	20	-	-
V_{DD}**(V)**	1	0.9	0.3	-	-
Avg. power (μW)	1.32	0.70	0.511	46	62
Avg. energy (pJ)	3.564	1.795	1.257	50	65

SAL implementations of PRESENT-80 has reduced power and energy consumption as compared to the CMOS-SPGAL. The FinSAL based PRESENT-80 consumes 46% and 50% of less power and energy consumption, respectively, as compared to the CMOS-SPGAL. Further, FinSAL also has 91% and 92% of less power and energy consumption, respectively, as compared to the CMOS based PRESENT-80. TunSAL has reduced power and energy consumption due to the reduced supply voltages as compared to FinFET and CMOS circuits. The TunSAL based PRESENT-80 consumes 62% and 65% of less power and energy consumption, respectively, as compared to the CMOS-SPGAL. Further, TunSAL has also 28% and 30% of less power and energy consumption, respectively, as compared to the FinSAL based PRESENT-80.

Table 5.4 shows the implementation results of PRESENT-80 using CMOS, CMOS SPGAL and UTB-SOI SPGAL logic gates. CMOS and CMOS-SPGAL are simulated at 1.8V and UTB-SOI SPGAL is simulated at 1.5V. The results of UTB-SOI SPGAL simulations are compared with the CMOS SPGAL simulation results and also with conventional CMOS simulation results for one round of PRESENT-80(refer Table 5.4). From the comparison results in Table 5.4, we can see that the UTB SOI SPGAL implementation of PRESENT-80 has reduced power and energy consump-

tion as compared to the CMOS and CMOS SPGAL implementations. The UTB-SOI SPGAL based PRESENT-80 consumes 92% and 91% of less power and energy consumption, respectively, as compared to the CMOS based PRESENT-80. UTB-SOI SPGAL based PRESENT-80 also consumes 36% and 33% of less power and energy, respectively, as compared to bulk CMOS SPGAL based PRESENT-80. The current consumption of UTB-SOI is also very less. It has to be noted that the current consumption of the UTB-SOI SPGAL PRESENT-80 is the sum current of all the power supplies. The UTB-SOI SPGAL based PRESENT-80 has reduced current consumption due to recovery of charge whereas in the conventional CMOS based PRESENT-80 the charges are discharged to ground leading to the additional current and power consumption.

Table 5.4: Comparison of metrics between CMOS, CMOS SPGAL and UTB-SOI SPGAL implementations of PRESENT-80

Metric	**CMOS**	**Bulk-CMOS SPGAL**	**UTB-SOI SPGAL**	**% imp of UTB-SOI SPGAL w.r.t CMOS**	**% imp of UTB-SOI SPGAL w.r.t SPGAL**
Avg. power (μW)	4079	513.6	328.7	92	36
Avg. current (μA)	2255	304.55	189.6	91	37
Avg. energy (nJ)	1.044	0.15	0.10	91	33

5.3 DPA Attack

Although FinSAL, TunSAL and UTB-SOI-SPGAL have shown better performance, it is important to validate their security against DPA attack. We have performed the simulations at 12.5 MHz and we have sampled the data with a sampling period of 1ns. Fig. 5.6(a) shows a successful DPA attack on the conventional CMOS based

PRESENT-80 design. For the CMOS-based implementation of the PRESENT-80 algorithm, the secret key was revealed using fewer traces (6130 input traces).

Further, It has been shown in Fig.5.6(b) that the DPA attack was unsuccessful for the UTB-SOI SPGAL based PRESENT-80. From our simulation results, the secret key was not revealed in the UTB-SOI SPGAL PRESENT for more than 50,000 input traces.

It has also been shown in Fig.5.7 that, DPA attack was unsuccessful for FinSAL and TunSAL gates.

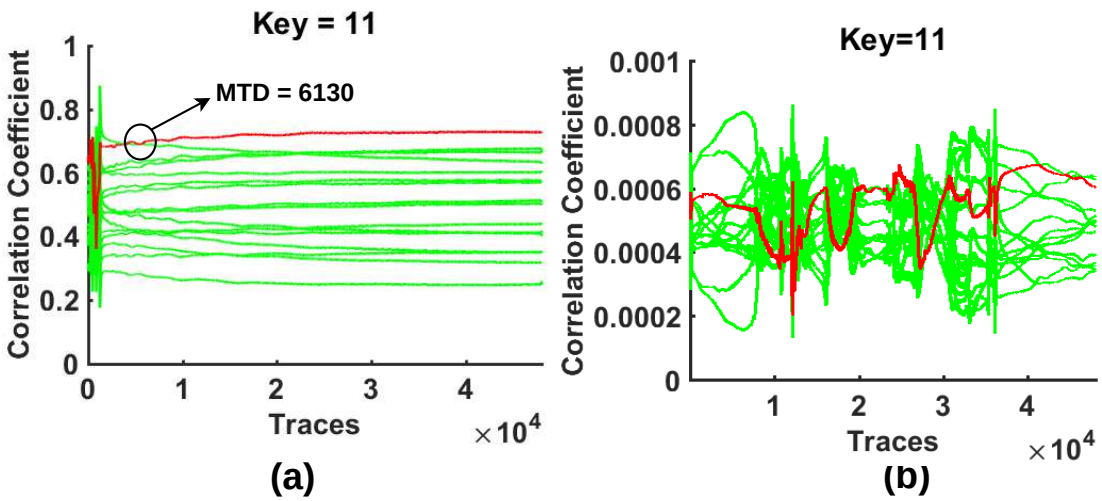

Figure 5.6: DPA attack results of PRESENT implemented using a) Conventional CMOS gates b) UTB-SOI SPGAL gates

5.4 Conclusion

In this work, we have demonstrated adiabatic computing in emerging transistors as a promising platform for low-power and LWC in IoT devices. From the simulation results, it is shown that the FinSAL consumes less current, less power and is more energy-efficient in comparison to its equivalent CMOS-based SPGAL at $22nm$ implementation at 0.5-0.9 V. TunSAL has shown amazing performance at 0.3V. UTBSOI based SPGAL has proven to show great improvements in energy, power and also

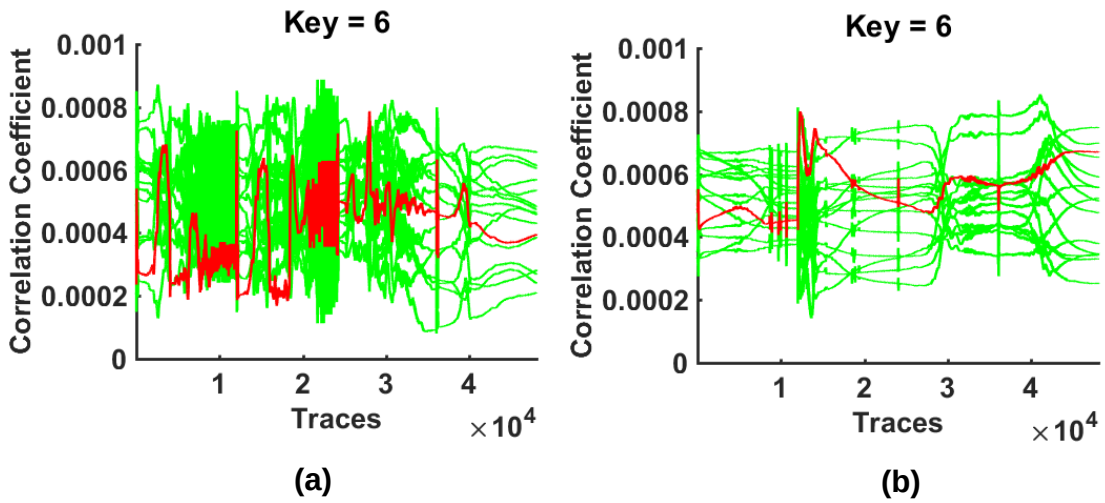

Figure 5.7: DPA attack results of PRESENT implemented using a) FinSAL gates b) TunSAL gates

in leakage power compared with its corresponding CMOS SPGAL. All the FinFET, TFET and UTB-SOI implementations of adiabatic PRESENT-80 were shown to be resilient to DPA attacks. Low leakage power, high energy efficiency and resilience against DPA attack makes adiabatic FinFET, TFET and UTB-SOI gates suitable to implement in LWC for IoT applications.

Chapter 6

Conclusions

In this thesis, significant contributions are made quantum computing and adiabatic computing - the two new computing paradigms that come under the umbrella of rebooting computing.

Two resource efficient integer division circuit designs were proposed for use in Quantum Computing. One design was based on restoring division algorithm and the other one on non-restoring division algorithm. The proposed quantum integer division circuits are shown to be superior to existing designs in terms of T-depth and T-count. The design of sub-components used in the proposed quantum integer division circuits such as the quantum conditional ADD operation circuit, quantum adder-subtractor and quantum subtraction circuit were also shown. The proposed non-restoring division circuit can be integrated into a more extensive quantum data path system where T-count and T-depth were of primary concern. The restoring division circuit can be used to implement quantum algorithms where qubits are limited and T-count and T-depth must be kept to a minimum. Both the designs were verified through Verilog simulations.

Next, implementation of adiabatic logic in Lightweight cryptography for IoT devices were examined. PRESENT-80, the lightweight cryptographic algorithm, was

used as a benchmark algorithm. The recently proposed Symmetric Pass Gate Adiabatic Logic (SPGAL) family was chosen for case study. It was proven that SPGAL implementation of PRESENT-80 consumes less power and energy as compared to its equivalent CMOS implementation. It was also shown that SPGAL implementation is resistant to DPA attacks, which are powerful side-channel attacks. To obtain the full leverage of adiabatic logic designs, SPGAL was implemented in the emerging transistor devices such as FinFET, TFET and UTB-SOI. All the FinFET, TFET and UTB-SOI based SPGAL designs had proven their resilience against DPA attacks. FinFET-SPGAL was shown to provide excellent improvements in terms of power and energy consumption compared to CMOS-SPGAL designs for 0.5 V to 0.9 V. TFET-SPGAL was demonstrated to be more secure and energy-efficient compared to FinFET-SPGAL and CMOS-SPGAL at 0.2 V - 0.5 V. UTB-SOI SPGAL resulted in less leakage power compared to its equivalent CMOS implementation.

The designs proposed in this thesis provide a solid foundation for future work. One such direction would be designing larger quantum circuits where any of the proposed division circuits can be used based on the requirements. More complex functional units such as quantum multipliers, quantum fast Fourier transform (FFT) units, quantum arithmetic logic units (ALUs) can be designed by taking advantage of the divider designs proposed in this thesis. Another possible future work could be to use the adiabatic logic in sub-threshold logic. The emerging transistor devices can also be implemented in designing circuits for sub-threshold logic.

References

[1] Matthew Amy, Dmitri Maslov, Michele Mosca, and Martin Roetteler. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(6):818–830, 2013.

[2] William C Athas, Lars J Svensson, Jefferey G Koller, Nestoras Tzartzanis, and E Ying-Chin Chou. Low-power digital systems based on adiabatic-switching principles. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2(4):398–407, 1994.

[3] Axel York Poschmann. Lightweight cryptography: cryptographic engineering for a pervasive world. In *PH. D. THESIS*. Citeseer, 2009.

[4] Dmitri Nikonov. Tunneling FETs. `https://nanohub.org/resources/18351`. [Online].

[5] Chenming Hu. Finfet and other new transistor technologies. *Univ. of California*, 2011.

[6] S Dinesh Kumar, Himanshu Thapliyal, Azhar Mohammad, and Kalyan S Perumalla. Design exploration of a symmetric pass gate adiabatic logic for energy-efficient and secure hardware. *Integration, the VLSI Journal*, 2016.

[7] S Dinesh Kumar, Himanshu Thapliyal, and Azhar Mohammad. Finsal: Finfet based secure adiabatic logic for energy-efficient and dpa resistant iot devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[8] D Michael Miller, Mathias Soeken, and Rolf Drechsler. Mapping ncv circuits to optimized clifford+ t circuits. In *International Conference on Reversible Computation*, pages 163–175. Springer, 2014.

[9] I Present. Cramming more components onto integrated circuits. *Readings in computer architecture*, 56, 2000.

[10] Dmitri E Nikonov and Ian A Young. Benchmarking of beyond-cmos exploratory devices for logic integrated circuits. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, 1:3–11, 2015.

[11] Paolo A. Gargini Alan Kadin Elie K. Track Thomas M. Conte, Erik DeBenedictis. Rebooting computing developing a roadmap for the future of the computer industry, 2015.

[12] Yasuhito Kawano and Michele Mosca. *Theory of Quantum Computation, Communication, and Cryptography: Third Workshop, TQC 2008 Tokyo, Japan, January 30-February 1, 2008, Revised Selected Papers*, volume 5106. Springer, 2008.

[13] Ilia Polian and Austin G Fowler. Design automation challenges for scalable quantum architectures. In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*, pages 1–6. IEEE, 2015.

[14] Peter Selinger et. al. *The Quipper System.* 2016. Available at: http://www.mathstat.dal.ca/ selinger/quipper/doc/.

[15] S Beauregard. Circuit for Shor's algorithm using 2n+3 gubits. *Quantum Information & Computation*, 3(2):175–185, Mar 2003.

[16] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[17] Jun Li, Xinhua Peng, Jiangfeng Du, and Dieter Suter. An efficient exact quantum algorithm for the integer square-free decomposition problem. *Scientific reports*, 2, 2012.

[18] Song Y Yan. *Quantum attacks on public-key cryptosystems.* Springer, 2013.

[19] Elisabeth Ilie-Zudor, Zsolt Kemény, Fred van Blommestein, László Monostori, and André van der Meulen. A survey of applications and requirements of unique identification systems and rfid techniques. *Computers in Industry*, 62(3):227–252, 2011.

[20] Kerry A McKay, Larry Bassham, Meltem Sönmez Turan, and Nicky Mouha. Report on lightweight cryptography. 2016.

[21] Himanshu Thapliyal, T.S.S Varun, E Munoz-Coreas, Keith A. Britt, and Travis S. Humble. Quantum circuit designs of integer division optimizing t-count and t-depth. In *Proceedings of the 2017 IEEE International Symposium on Nanoelectronic and Information Systems.* IEEE, Dec, 2017.

[22] Himanshu Thapliyal, TSS Varun, and S Dinesh Kumar. Adiabatic computing based low-power and dpa-resistant lightweight cryptography for iot devices. In *VLSI (ISVLSI), 2017 IEEE Computer Society Annual Symposium on*, pages 621–626. IEEE, 2017. Reprinted, with permission, from IEEE.

[23] Himanshu Thapliyal, TSS Varun, and S Dinesh Kumar. Low-power and secure lightweight cryptography via tfet-based energy recovery circuits. In *(ICRC), 2017 IEEE International Conference on Rebooting Computing*, pages 1 – 4. IEEE, 2017. Reprinted, with permission, from IEEE.

[24] Himanshu Thapliyal, TSS Varun, and S Dinesh Kumar. Utb-soi based adiabatic computing for low-power and secure iot devices. In *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, page 16. ACM, 2017. http://doi.acm.org/10.1145/3064814.3064825.

[25] KJ Sharma. Understanding quantum computing. *IJSEAS*, 1(6):370–388, 2015.

[26] Vipul Singh, Nishanth Dikkala, and Pushpak Bhattacharyya. A quantum computing approach to part-of-speech tagging: A quantum viterbi decoding algorithm.

[27] Himanshu Thapliyal. Mapping of subtractor and adder-subtractor circuits on reversible quantum gates. In *Transactions on Computational Science XXVII*, pages 10–34. Springer, 2016.

[28] Edgard Muñoz-Coreas and Himanshu Thapliyal. Design of quantum circuits for galois field squaring and exponentiation. In *VLSI (ISVLSI), 2017 IEEE Computer Society Annual Symposium on*, pages 68–73. IEEE, 2017.

[29] Alexandru Paler, Ilia Polian, Kae Nemoto, and Simon J Devitt. Fault-tolerant, high-level quantum circuits: form, compilation and description. *Quantum Science and Technology*, 2(2):025003, 2017.

[30] Xinlan Zhou, Debbie W. Leung, and Isaac L. Chuang. Methodology for quantum logic gate construction. *Phys. Rev. A*, 62:052316, Oct 2000.

[31] Philip Teichmann. *Adiabatic logic: future trend and system level perspective*, volume 34. Springer Science & Business Media, 2011.

[32] Amir Moradi and Axel Poschmann. Lightweight cryptography and dpa countermeasures: A survey. In *Financial Cryptography and Data Security*, pages 68–79. Springer, 2010.

[33] Marilyn Wolf. Ultralow power and the new era of not-so-vlsi. *IEEE Design & Test*, 33(4):109–113, 2016.

[34] Axel Poschmann, Gregor Leander, Kai Schramm, and Christof Paar. New lightweight crypto algorithms for rfid. In *2007 IEEE International Symposium on Circuits and Systems*, pages 1843–1846. IEEE, 2007.

[35] Digh Hisamoto, Wen-Chin Lee, Jakub Kedzierski, Hideki Takeuchi, Kazuya Asano, Charles Kuo, Erik Anderson, Tsu-Jae King, Jeffrey Bokor, and Chenming Hu. Finfet-a self-aligned double-gate mosfet scalable to 20 nm. *Electron Devices, IEEE Transactions on*, 47(12):2320–2325, 2000.

[36] Prateek Mishra, Anish Muttreja, and Niraj K Jha. Finfet circuit design. In *Nanoelectronic Circuit Design*, pages 23–54. Springer, 2011.

[37] Suman Datta, Huichu Liu, and Vijaykrishnan Narayanan. Tunnel fet technology: A reliability perspective. *Microelectronics Reliability*, 54(5):861–874, 2014.

[38] Navid Paydavosi, Sriramkumar Venugopalan, Yogesh Singh Chauhan, Juan Pablo Duarte, Srivatsava Jandhyala, Ali M Niknejad, and Chenming Calvin Hu. Bsimspice models enable finfet and utb ic designs. *IEEE Access*, 1:201–215, 2013.

[39] S. J. Devitt, A. M. Stephens, W. J. Munro, and K. Nemoto. Requirements for fault-tolerant factoring on an atom-optics quantum computer. *Nature Communications*, 4:2524, October 2013.

[40] IBM. *Quantum Computing - IBM Q*. 2017. Available at: https://www.research.ibm.com/ibm-q/.

[41] N. M. Nayeem, A. Hossain, M. Haque, L. Jamal, and H. M. H. Babu. Novel reversible division hardware. In *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*, pages 1134–1138, Aug 2009.

[42] S. V. Dibbo, H. M. H. Babu, and L. Jamal. An efficient design technique of a quantum divider circuit. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2102–2105, May 2016.

[43] Faraz Dastan and Majid Haghparast. A novel nanometric fault tolerant reversible divider. *International Journal of the Physical Sciences*, 6(24):5671–5681, October 2011.

[44] Alireza Khosropour, Hossein Aghababa, and Behjat Forouzandeh. Quantum division circuit based on restoring division algorithm. In *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on*, pages 1037–1040. IEEE, 2011.

[45] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. Fast and efficient exact synthesis of single-qubit unitaries generated by clifford and t gates. *Quantum Info. Comput.*, 13(7-8):607–630, July 2013.

[46] Himanshu Thapliyal and Nagarajan Ranganathan. Design of efficient reversible logic-based binary and bcd adder circuits. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 9(3):17, 2013.

[47] Steven A Cuccaro, Thomas G Draper, Samuel A Kutin, and David Petrie Moulton. A new quantum ripple-carry addition circuit. *arXiv preprint quant-ph/0410184*, 2004.

[48] Keith A Britt and Travis S Humble. High-performance computing with quantum processing units. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):39, 2017.

[49] Keith A Britt and Travis S Humble. Instruction set architectures for quantum processing units. *arXiv preprint arXiv:1707.06202*, 2017.

[50] Norbert M Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A Landsman, Kenneth Wright, and Christopher Monroe. Experimental comparison of two quantum computing architectures. *Proceedings of the National Academy of Sciences*, page 201618020, 2017.

[51] Esteban A Martinez, Thomas Monz, Daniel Nigg, Philipp Schindler, and Rainer Blatt. Compiling quantum algorithms for architectures with multi-qubit gates. *New Journal of Physics*, 18(6):063029, 2016.

[52] Debasis Bandyopadhyay and Jaydip Sen. Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69, 2011.

[53] Paul Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi. Introduction to differential power analysis. *Journal of Cryptographic Engineering*, 1(1):5–27, 2011.

[54] Himanshu Thapliyal and Mark Zwolinski. Reversible logic to cryptographic hardware: a new paradigm. In *Circuits and Systems, 2006. MWSCAS'06. 49th IEEE International Midwest Symposium on*, volume 1, pages 342–346. IEEE, 2006.

[55] Davide Bellizia, Giuseppe Scotti, and Alessandro Trifiletti. Implementation of the present-80 block cipher and analysis of its vulnerability to side channel attacks exploiting static power. In *Mixed Design of Integrated Circuits and Systems, 2016 MIXDES-23rd International Conference*, pages 211–216. Department of Microelectronics and Computer Science, Lodz University of Technology, 2016.

[56] Yibin Ye and Kaushik Roy. Qserl: Quasi-static energy recovery logic. *IEEE Journal of Solid-State Circuits*, 36(2):239–248, 2001.

[57] S Dinesh Kumar, Himanshu Thapliyal, and Azhar Mohammad. Ee-spfal: A novel energy-efficient secure positive feedback adiabatic logic for dpa resistant rfid and smart card. *IEEE Transactions on Emerging Topics in Computing*, 2016.

[58] Kris Tiri, Moonmoon Akmal, and Ingrid Verbauwhede. A dynamic and differential cmos logic with signal independent power consumption to withstand differential power analysis on smart cards. In *Solid-State Circuits Conference, 2002. ESSCIRC 2002. Proceedings of the 28th European*, pages 403–406. IEEE, 2002.

[59] Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-channel resistant crypto for less than 2,300 ge. *Journal of Cryptology*, 24(2):322–345, 2011.

[60] Arizona State University. Ptm models, 2012.

[61] Jun Wu, Yiyu Shi, and Minsu Choi. Measurement and evaluation of power analysis attacks on asynchronous s-box. *Instrumentation and Measurement, IEEE Transactions on*, 61(10):2765–2775, 2012.

[62] Ronak Lad Pavan H Vora. A review paper on cmos, soi and finfet technology.

[63] Huichu Liu, Vinay Saripalli, Vijaykrishnan Narayanan, and Suman Datta. Iii-v tunnel fet model, Apr 2015.

[64] University of California Berkley. Bsim models, 2013.

Vita

Sai Subramanya Varun Thogarcheti

Education

G. Pulla Reddy Engineering College, India
Bachelor of Science in Electronics and Communication Engineering, May 2015

Experience

Graduate Research Assistant
August 2015-May 2016
University of Kentucky
Lexington, KY

Publications

Himanshu Thapliyal, TSS Varun, Edgard Muñoz-Coreas. "Quantum circuit design of integer division optimizing ancillary qubits and T-count, arXiv, vol. 1609.01241, 2016.
Himanshu Thapliyal, TSS Varun, Edgard Muñoz-Coreas, Keith A. Britt and Travis S. Humble. "Quantum Circuit Designs of Integer Division Optimizing T-count and T-depth." Proceedings of IEEE International Symposium on Nanoelectronic and Information Systems (INIS), 2017.
Himanshu Thapliyal, TSS Varun, S. Dinesh Kumar. "UTB-SOI Based Adiabatic Computing for Low-Power and Secure IoT Devices" Cyber and Information Security Research Conference 2017, Article No 16.
Himanshu Thapliyal, TSS Varun, S. Dinesh Kumar. "Adiabatic Computing for Low Power and DPA Resistant Lightweight Cryptography for IoT Applications." IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2017, 10.1109/ISVLSI.2017.115.
Himanshu Thapliyal, TSS Varun, S. Dinesh Kumar. "Low-Power and Secure Lightweight Cryptography Via TFET-Based Energy Recovery Circuits" IEEE International Conference on Rebooting Computing (ICRC), pages 1-4, 2017.