11-23-2013

Theory, Synthesis, and Application of Adiabatic and Reversible Logic Circuits For Security Applications

Matthew Arthur Morrison
University of South Florida, mamorris@mail.usf.edu

Follow this and additional works at: <https://digitalcommons.usf.edu/etd>

Part of the [Computer Engineering Commons](#)

Scholar Commons Citation

Morrison, Matthew Arthur, "Theory, Synthesis, and Application of Adiabatic and Reversible Logic Circuits For Security Applications" (2013). *USF Tampa Graduate Theses and Dissertations*.
<https://digitalcommons.usf.edu/etd/5082>

This Dissertation is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact digitalcommons@usf.edu.

Theory, Synthesis, and Application of Adiabatic and Reversible Logic Circuits For
Security Applications

by

Matthew A. Morrison

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Nagarajan Ranganathan, Ph.D.
Sanjukta Bhanja, Ph.D.
Srinivas Katkoori, Ph.D.
Jay Ligatti, Ph.D.
Kandethody Ramachandran, Ph.D.
Hao Zheng, Ph.D.

Date of Approval:
November 22, 2013

Keywords: Charge Based Computing, DPA Attacks, Encryption, Memory, Power

Copyright © 2014, Matthew A. Morrison

DEDICATION

To my parents, Alfred and Kathleen Morrison, and to my grandparents, Arthur and Betty Kempf, and Alfred and Dorothy Morrison, for making all the opportunities I have possible.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Nagarajan Ranganathan, for giving me the opportunity to pursue my research, and for his valuable guidance and assistance over these last four years. Ranga has always been an excellent motivator, guider, and leader, and completion of this project would not have been possible without his patience and belief in my ideas. I want to thank the members of my committee, Dr. Thomas Weller, Dr. Srinivas Katkoori, Dr. Hao Zheng, Dr. Sanjukta Bhanja, and Dr. Kandethody Ramachandran for their valuable advice and assistance in preparing my dissertation materials. I want to extend my gratitude to all the employees of the Computer Science and Engineering Department - Theresa Collins, Yvette Blanchard, Kimberly Bluemer, Joe Botto, and Kelly Schluer - for all their hard work in making my graduate school experience an excellent experience. I want to thank my colleagues in the VLSI System Design Research Laboratory – Soumyaroop, Ransford, Himanshu, Michael, Yue, Saurabh, and Santosh – for a fun and intellectually stimulating atmosphere in the lab. I want to thank other colleagues in CSE – Chris, Richard, Benjamin, Donald, Sam, Robert, Adrian, Bader, Kevin, and many others. I particularly want to thank Matt Lewandowski, whose excellent work on the VHDL library was instrumental in completion of the Integrated Qubit results presented in this dissertation. I want to thank all my friends in the USF Athletics community, particularly my friends in the Beef Studs, Student Bulls Club, Goalmouths, and players and families of the USF Softball team. Finally, I want to thank my mother and father for their patience and wisdom, and instilling in me the drive for excellence for which I am very grateful, as I know that attitude got me through many tough days.

TABLE OF CONTENTS

LIST OF TABLES	iv
LIST OF FIGURES	vi
ABSTRACT	xiv
CHAPTER 1: INTRODUCTION	1
1.1 – Motivation	2
1.2 – Contributions and Outline of Dissertation	6
CHAPTER 2: RELATED WORKS	10
2.1 – Physical Reversibility in a Quantum System	10
2.2 – Reversibility in a Universal Computing Machine	12
2.3 – Fundamentals of Reversible and Adiabatic Logic	24
2.4 – Basics of Dual-Rail Adiabatic Design	28
2.5 – Previous Work in Reversible Logic Synthesis	31
2.5.1 – Unitary Matrix and Truth Table Based Synthesis	32
2.5.2 – Graph and Group Based Synthesis	34
2.6 – Clocking Schemes in Sequential Adiabatic Logic	37
2.7 – Smart Card History	40
2.8 – Smart Card Standardization	43
2.8.1 – ISO/IEC 7816 Smart Card Standard	45
2.8.2 – ISO/IEC 14443/15693 Standards	46
2.9 – SIM Cards and the Java Card Standard	48
2.10 – Low Power Medical Device Encryption	51
2.11 – Differential Power Analysis Prevention	53
2.12 – Previous Work in Adiabatic Dynamic Logic	57
2.13 – Dynamic Information Flow Tracking	59
2.14 – Encryption and the Rijndael Algorithm	60
2.15 – Previous Work in S-Box Hardware Design	61
2.16 – Previous Work in Reversible Logic Architectures	63
CHAPTER 3: MITIGATING COMPUTING DEMONS	66
3.1 – Charge Based Computing Debate	66
3.2 – Proposed CMOS Source-Memory Device	69
3.3 – Simulation Results	71
3.4 – Noise Considerations	71
CHAPTER 4: PERMISSIBILITY OF SEQUENTIAL COMPUTING	77
4.1 – Sequential Reversible Logic Debate	77

4.2 – Definitions	80
4.3 – Mathematical Model	82
4.4 – Simulation of Reversible Logic and a Case Study	84
4.5 – Proposed 4*4 MLMRG for SRAM Implementation	87
4.6 – Proposed SRAM with Reversible Elements	88
4.7 – Proposed 4*4 Reversible Decoder (RD) Gate	89
4.8 – 4x2 SRAM Array with Reversible Elements	90
4.9 – DRAM Array with Reversible Elements	90
4.10 – Improved SRAM Structure	92
4.11 – Conclusion	93
 CHAPTER 5: INTEGRATED QUBIT BASED REVERSIBLE LOGIC	 95
5.1 – Reversible Design Goals	96
5.2 – Unitary Matrix Representation of Quantum Interaction	96
5.3 – Previous Work in Control-V Simulation	98
5.4 – Integrated Qubit Gates	99
5.5 – Presented Integrated Qubit Behavioral Model	99
5.6 – Reversible Logic Design Rules for VHDL	102
5.7 – Simulation of Fundamental Reversible Logic Structures	104
5.8 – VHDL QUBIT Implementation and Advantages	106
5.9 – Verification	108
5.10 – Proposed Integrated Qubit Buffer Gate as a Solution	111
5.11 – Proposed Optimization Algorithm	114
5.12 – Algorithm Verification and Comparison to Previous Results	121
5.13 – Conclusions	122
 CHAPTER 6: SYNTHESIS OF DUAL RAIL ADIABATIC LOGIC	 126
6.1 – Introduction	126
6.2 – Motivation and Background	128
6.2.1 – Fundamentals of Reversible and Adiabatic Logic	128
6.2.2 – Energy Consumption and Security Applications	129
6.3 – Presented Adiabatic Synthesis Algorithm	131
6.3.1 - Determining the Horizontal Offsets	132
6.3.2 – Determining the Initial Edges	133
6.3.3 – Optimization using ESPRESSO Boolean Minimization	136
6.3.4 – Swap Condition	139
6.3.5 – Formal Definition of Final Algorithm	140
6.4 – Comparison to Previous Benchmarks	145
6.5 – Conclusions	146
 CHAPTER 7: ADIABATIC DYNAMIC DIFFERENTIAL LOGIC	 153
7.1 – Comparison of Inverter A and Inverter B (Conventional)	153
7.2 – Comparison of Inverter C and Inverter D (SCRL)	156
7.3 – Comparison of Inverter E and Inverter F (ECRL)	158
7.4 – Forward Body Biased Adiabatic Toffoli Gate	160
7.5 – Forward Body Biased Adiabatic Fredkin Gate	162

7.6 – Performance Adiabatic Dynamic Differential Logic (PADDL) Cell	165
7.7 – Comparison to Previous Benchmarks	170
7.8 – A Note on Bijectivity in Secure IC Design	171
7.9 – Body-Biased Adiabatic Dynamic Differential Logic (BADDL) Cell	172
7.10 – Conclusions	176
CHAPTER 8: DUAL-RAIL ADIABATIC S-BOX FOR DPA MITIGATION	177
8.1 – Introduction	177
8.2 – Adiabatic S-Box Implementation	179
8.3 – Benchmark Comparison	199
8.4 – Conclusions	200
CHAPTER 9: CONCLUSIONS AND FUTURE WORK	203
9.1 – Conclusions	203
9.2 – Future Work	207
REFERENCES	210
APPENDICIES	241
Appendix 1 – Journal Reprint Permissions	241
ABOUT THE AUTHOR	End Page

LIST OF TABLES

Table 2.1 – Three Tape Reversible Turing Machine Operation.	19
Table 3.1 – Truth Table for Source-Memory Device Operations.	70
Table 3.2 – Energy Dissipation for PMOS and NMOS Circuit (Joules).	72
Table 4.1 – Inputs into Circuit in Fig. 4.1 Over 12 Time Cycles.	86
Table 4.2 – Outputs from Circuit in Fig. 4.1 Over 12 Time Cycles.	86
Table 4.3 – Inputs into Circuit in Fig. 4.2 Over 12 Time Cycles.	87
Table 4.4 – Outputs from Circuit in Fig. 4.2 Over 12 Time Cycles.	87
Table 5.1 – IQ Spin Operation Lookup Table.	101
Table 5.2 – V and $V^\dagger$ Behavioral Modeling System.	101
Table 5.3 – Improvement in Delay and Cost from Fig. 5.28 Based on Algorithm.	121
Table 5.4 – Improvement upon Synthesized Circuits.	124
Table 6.1 – Fundamental Gates and their Assigned Switch Values.	132
Table 6.2 – Synthesis of Fundamental Reversible Logic Structures Stages.	147
Table 6.3 – Synthesis of Fundamental Reversible Logic Structures Results.	148
Table 6.4 – Comparison of Synthesized Designs to Benchmarks.	149
Table 6.5 – Benchmark Power Comparison.	151
Table 7.1 – HSPICE Simulation Measurements for Inverters in Fig. 7.1-7.7.	159
Table 7.2 – Truth Table for Proposed PADDL Cell.	166
Table 7.3 – PADDL Cell Logic Outputs.	167
Table 7.4 – Benchmarks Average Power Consumption Comparison (10^{-8} W).	171

LIST OF FIGURES

Figure 1.1 – Instantaneous Power Waveform of Inverter in 22nm CMOS.	5
Figure 1.2 – Instantaneous Power Waveform of NAND Gate in 22nm CMOS.	6
Figure 1.3 – List of Contributions.	9
Figure 2.1 – Requirements for Proper Adiabatic Operation.	29
Figure 2.2 – Adiabatic ‘On’ Switches.	29
Figure 2.3 – Adiabatic ‘Off’ Switches.	29
Figure 2.4 – Basic Square Circuit of a Dual Rail Toffoli Gate.	30
Figure 2.5 – Adiabatic CMOS Design of a Dual Rail Toffoli Gate.	31
Figure 2.6 – HSPICE Simulation at 10 MHz of Adiabatic CMOS Toffoli Gate.	31
Figure 2.7 – NAND Gate (Red) and Adiabatic Toffoli Gate (Gold) Power.	31
Figure 2.8 – Taxonomy of Relevant Works in Differential Power Analysis.	55
Figure 2.9 – Taxonomy of Relevant Works in Adiabatic Security.	64
Figure 3.1 – Presented CMOS Source-Memory Device.	70
Figure 3.2 – Input Simulation Waveform.	70
Figure 3.3 – Energy, Time Delay Product and Power Analysis for 10mV.	72
Figure 3.4 – Energy, Time Delay Product and Power Analysis for 5mV.	73
Figure 3.5 – Energy, Time Delay Product and Power Analysis for 1mV.	73
Figure 3.6 – Energy, Time Delay Product and Power Analysis for 0.5mV.	74
Figure 3.7 – Input (Green) and Output (Yellow) Waveforms for Adiabatic Circuit.	74
Figure 3.8 – Instantaneous Power Waveform for Adiabatic Circuit.	74

Figure 3.9 – Supply Current Waveform for Adiabatic Circuit.	75
Figure 4.1 – Sequential Reversible Logic Structure.	85
Figure 4.2 – Reverse Circuit of Figure 4.1.	85
Figure 4.3 – VHDL Simulation of Sequential Fredkin Gate.	87
Figure 4.4 – VHDL Simulation of Sequential Reverse Fredkin Gate.	87
Figure 4.5 – Quantum Representation of Proposed MLMRG.	88
Figure 4.6 – MLMRG in Series with Peres and Reduced Representation.	88
Figure 4.7 – Modified MLMRG for Implementation in SRAM Array.	89
Figure 4.8 – SRAM Cell with Reversible Elements.	89
Figure 4.9 – Quantum Representation of RD Gate.	90
Figure 4.10 – 2-to-4 Decoder Logical Configuration of RD Gate.	90
Figure 4.11 – Reversible 4x2 SRAM Array.	91
Figure 4.12 – Reversible Dual-Port SRAM Cell.	91
Figure 4.13 – Synchronous Dual-Port SRAM Array.	92
Figure 4.14 – Reversible DRAM Logical Configuration.	92
Figure 4.15 – Improved Reversible SRAM Cell with RMUX Gate.	94
Figure 5.1 – Graphical Qubit Representation of the Bloch Sphere.	97
Figure 5.2 – Quantum Representation of the Control V and Control V+ Gates.	98
Figure 5.3 – Quantum Representation of Integrated Qubit Gates.	100
Figure 5.4 – Realization of Fredkin Gate with IQ Gates.	100
Figure 5.5 – Presented VHDL-93 QUBIT Syntax.	101
Figure 5.6 – VHDL Simulation Results for LIQP Gate.	102
Figure 5.7 – Signal Generation Example for Toffoli Gate.	103

Figure 5.8 – Signal Generation for CNOT Gate.	103
Figure 5.9 – Simulation Results for CNOT Gate.	103
Figure 5.10 – SWAP Gate Proposed in [182] (left) and Previous (right).	104
Figure 5.11 – Simulation Results for IQ-Based SWAP Gate.	104
Figure 5.12 – Simulation Results for Fredkin Gate.	105
Figure 5.13 – Simulation Results for Toffoli Gate.	105
Figure 5.14 – Simulation Results for Peres Gate.	106
Figure 5.15 – Simulation Results for RMUX1 Gate.	106
Figure 5.16 – Simulation Results for UPG.	106
Figure 5.17 – Xilinx ISE 13.2 Schematic Layout FRE Symbol.	107
Figure 5.18 – Quantum Representation of Peres Gate in Forward Operation.	108
Figure 5.19 – Quantum Representation of Peres Gate in Reverse Operation.	109
Figure 5.20 – Peres Gate Forward Operation Without Clocking Zone Regulation.	109
Figure 5.21 – Peres Gate Reverse Operation Without Clocking Zone Regulation.	109
Figure 5.22 – Quantum Representation of NOT Gate.	112
Figure 5.23 – Quantum Representation of Buffer Gate.	112
Figure 5.24 – Quantum Representation of Buffered Forward Peres Gate.	113
Figure 5.25 – Quantum Representation of Buffered Reverse Peres Gate.	113
Figure 5.26 – VHDL Simulation of Buffered Forward Peres Gate.	114
Figure 5.27 – VHDL Simulation of Buffered Reverse Peres Gate.	114
Figure 5.28 – Replacement Rules for Integrated Qubit Optimization .	116
Figure 5.29 – Graph and Circuit after CheckForMerge and CreateGraph.	119
Figure 5.30 – Reversible Circuit Presented in NCT and IQ.	123

Figure 5.31 – Reversible Circuit after CheckForMerge.	123
Figure 5.32 – Reduced Sequencing Graph.	125
Figure 5.33 – Final Reduced Circuit	125
Figure 6.1 – Dual-Rail Adiabatic S-box Power Waveform.	131
Figure 6.2 – CNOT Permutation, Unitary, and the Horizontal Offset Matrix.	133
Figure 6.3 – Offset Graphs.	134
Figure 6.4 – Offset Graphs (Left) and Edges (Right) for the CNOT Gate.	134
Figure 6.5 – Offset Graphs (Left) and Edges (Right) for the Toffoli Gate.	135
Figure 6.6 – Offset Graphs (Left) and Edges (Right) for the Peres Gate.	135
Figure 6.7 – Offset Graphs of Swap (Left) and Fredkin (Right).	136
Figure 6.8 – Offset Graph for the Inversion Gate.	138
Figure 6.9 – Edges Before (Left) and After (Right) for the Peres Gate.	138
Figure 6.10 – Full Synthesis of the Adiabatic Toffoli Gate.	139
Figure 6.11 – Full Synthesis of the Adiabatic Fredkin Gate.	141
Figure 6.12 – HSPICE Simulation of Synthesized Adiabatic Fredkin Gate	141
Figure 6.13 – Synthesis Flow of Proposed Algorithm.	142
Figure 6.14 – Synthesis Flow of Adiabatic Full Adder.	142
Figure 7.1 – Adiabatic Inverters With and Without Biasing.	154
Figure 7.2 – HSPICE Simulation of INVa and INVb at 1MHz.	155
Figure 7.3 – Comparison of Instantaneous Power Waveforms of INVa and INVb.	155
Figure 7.4 – HSPICE Simulation of INVc and INVd at 1MHz.	157
Figure 7.5 – Comparison of Instantaneous Power Waveforms of INVc and INVd.	157
Figure 7.6 – HSPICE Simulation of INVe and INVf at 1MHz.	159

Figure 7.7 – Comparison of Instantaneous Power Waveforms of INVe and INVf.	160
Figure 7.8 – Schematic Layout of CMOS Adiabatic Toffoli Gate.	161
Figure 7.9 – HSPICE Simulation of CMOS Adiabatic Body Biased Toffoli Gate.	161
Figure 7.10 – HSPICE Simulation Comparison of Adiabatic Toffoli Gates.	162
Figure 7.11 – Power of NAND Gate (Black) and Toffoli Gate (Red) at 0.5V.	162
Figure 7.12 – Schematic Layout of CMOS Adiabatic Fredkin Gate.	163
Figure 7.13 – HSPICE Simulation of CMOS Adiabatic Body Biased Fredkin Gate.	163
Figure 7.14 – HSPICE Simulation Comparison of CMOS Adiabatic Fredkin Gates.	164
Figure 7.15 – Power of MUX Gate (Red) and Fredkin Gate (Blue) at 0.5V.	165
Figure 7.16 – Basic Square Circuit Diagram for Proposed PADDL Cell.	166
Figure 7.17 – CMOS Schematic Diagram for Proposed PADDL Cell.	167
Figure 7.18 – HSPICE Simulation of PADDL Logic Cell at 1MHz.	168
Figure 7.19 – Power of NAND Gate (Blue) and Toffoli Gate (Red) at 1MHz.	168
Figure 7.20 – HSPICE Simulation of PADDL Logic Cell at 13.56MHz.	169
Figure 7.21 – Power of NAND Gate (Blue) and Toffoli Gate (Red) at 13.56MHz.	170
Figure 7.22 – Cascaded PADDL Cells with Logic Outputs Shown.	172
Figure 7.23 – Transient Analysis of the Impact of Body Biasing on BADDL.	173
Figure 7.24 – HSPICE Simulation of BADDL Logic Cell at 13.56MHz.	174
Figure 7.25 – Comparison of BADDL Logic Cells With and Without Body Bias.	175
Figure 8.1 – Isim Simulation of Forward Dual-Rail Adiabatic Sbox.	181
Figure 8.2 – Isim Simulation of Reverse Dual-Rail Adiabatic Sbox.	181
Figure 8.3 – Power Waveform for All 256 Inputs at 1MHz and 0.35 μ m.	182
Figure 8.4 – Power Waveform for 16 Random Inputs at 1MHz and 0.35 μ m.	182

Figure 8.5 – Power Waveform for All 256 Inputs at 13.56MHz and 0.35μm.	182
Figure 8.6 – Power Waveform for 16 Random Inputs at 13.56MHz and 0.35μm.	183
Figure 8.7 – Power Waveform for All 256 Inputs at 50MHz and 0.35μm.	183
Figure 8.8 – Power Waveform for 16 Random Inputs at 50MHz and 0.35μm.	183
Figure 8.9 – Power Waveform for All 256 Inputs at 10MHz and 0.25μm.	184
Figure 8.10 – Power Waveform for 16 Random Inputs at 10MHz and 0.25μm.	184
Figure 8.11 – Power Waveform for All 256 Inputs at 13.56MHz and 0.25μm.	184
Figure 8.12 – Power Waveform for 16 Random Inputs at 13.56MHz and 0.25μm.	185
Figure 8.13 – Power Waveform for All 256 Inputs at 0.1MHz and 0.18μm.	185
Figure 8.14 – Power Waveform for 16 Random Inputs at 0.1MHz and 0.18μm.	185
Figure 8.15 – Power Waveform for All 256 Inputs at 1.25MHz and 0.18μm.	186
Figure 8.16 – Power Waveform for 16 Random Inputs at 1.25MHz and 0.18μm.	186
Figure 8.17 – Power Waveform for All 256 Inputs at 13.56MHz and 0.18μm.	186
Figure 8.18 – Power Waveform for 16 Random Inputs at 13.56MHz and 0.18μm.	187
Figure 8.19 – Power Waveform for All 256 Inputs at 50MHz and 0.18μm.	187
Figure 8.20 – Power Waveform for 16 Random Inputs at 50MHz and 0.18μm.	187
Figure 8.21 – Power Waveform for All 256 Inputs at 1MHz and 0.13μm.	188
Figure 8.22 – Power Waveform for 16 Random Inputs at 1MHz and 0.13μm.	188
Figure 8.23 – Power Waveform for All 256 Inputs at 13.56MHz and 0.13μm.	188
Figure 8.24 – Power Waveform for 16 Random Inputs at 13.56MHz and 0.13μm.	189
Figure 8.25 – Power Waveform for All 256 Inputs at 1MHz and 90nm.	189
Figure 8.26 – Power Waveform for 16 Random Inputs at 1MHz and 90nm.	189
Figure 8.27 – Power Waveform for All 256 Inputs at 13.56MHz and 90nm.	190

Figure 8.28 – Power Waveform for 16 Random Inputs at 13.56MHz and 90nm.	190
Figure 8.29 – Power Waveform for All 256 Inputs at 50MHz and 90nm.	190
Figure 8.30 – Power Waveform for 16 Random Inputs at 50MHz and 90nm.	191
Figure 8.31 – Power Waveform for All 256 Inputs at 1MHz and 65nm.	191
Figure 8.32 – Power Waveform for 16 Random Inputs at 1MHz and 65nm.	191
Figure 8.33 – Power Waveform for All 256 Inputs at 13.56MHz and 65nm.	192
Figure 8.34 – Power Waveform for 16 Random Inputs at 13.56MHz and 65nm.	192
Figure 8.35 – Power Waveform for All 256 Inputs at 50MHz and 65nm.	192
Figure 8.36 – Power Waveform for 16 Random Inputs at 50MHz and 65nm.	193
Figure 8.37 – Power Waveform for All 256 Inputs at 1MHz and 45nm.	193
Figure 8.38 – Power Waveform for 16 Random Inputs at 1MHz and 45nm.	193
Figure 8.39 – Power Waveform for All 256 Inputs at 13.56MHz and 45nm.	194
Figure 8.40 – Power Waveform for 16 Random Inputs at 13.56MHz and 45nm.	194
Figure 8.41 – Power Waveform for All 256 Inputs at 50MHz and 45nm.	194
Figure 8.42 – Power Waveform for 16 Random Inputs at 50MHz and 45nm.	195
Figure 8.43 – Power Waveform for All 256 Inputs at 1MHz and 32nm.	195
Figure 8.44 – Power Waveform for 16 Random Inputs at 1MHz and 32nm.	195
Figure 8.45 – Power Waveform for All 256 Inputs at 13.56MHz and 32nm.	196
Figure 8.46 – Power Waveform for 16 Random Inputs at 13.56MHz and 32nm.	196
Figure 8.47 – Power Waveform for All 256 Inputs at 50MHz and 32nm.	196
Figure 8.48 – Power Waveform for 16 Random Inputs at 50MHz and 32nm.	197
Figure 8.49 – Power Waveform for All 256 Inputs at 1MHz and 22nm.	197
Figure 8.50 – Power Waveform for 16 Random Inputs at 1MHz and 22nm.	197

Figure 8.51 – Power Waveform for All 256 Inputs at 13.56MHz and 22nm.	198
Figure 8.52 – Power Waveform for 16 Random Inputs at 13.56MHz and 22nm.	198
Figure 8.53 – Power Waveform for All 256 Inputs at 50MHz and 22nm.	198
Figure 8.54 – Power Waveform for 16 Random Inputs at 50MHz and 22nm.	199

ABSTRACT

Programmable reversible logic is emerging as a prospective logic design style for implementation in modern nanotechnology and quantum computing with minimal impact on circuit heat generation. Adiabatic logic is a design methodology for reversible logic in CMOS where the current flow through the circuit is controlled such that the energy dissipation due to switching and capacitor dissipation is minimized. Recent advances in reversible logic using quantum computer algorithms allow for improved computer architectures. Production of cost-effective Secure Integrated Chips, such as Smart Cards, requires hardware designers to consider tradeoffs in size, security, and power consumption. In order to design successful security-centric designs, the low-level hardware must contain built-in protection mechanisms to supplement cryptographic algorithms such as AES and Triple DES by preventing side channel attacks, such as Differential Power Analysis (DPA). Dynamic logic obfuscates the output waveforms and the circuit operation, reducing the effectiveness of the DPA attack. Significant research exists in the design and analysis of locally optimal adiabatic elements towards mitigation of side channel attacks. However, none of these works have addressed the use of adiabatic logic in implementation of flexible and programmable hardware security policies. Nor has adiabatic logic been employed in hardware security applications such as trustworthy voting systems and data encryption standards.

In this dissertation, I address theory, synthesis, and application of adiabatic and reversible logic circuits for security applications. First, two major debates in reversible computing are addressed. These debates must be addressed in order to devise computational logic primitives in

any emerging quantum computing technology. First, we address whether charged based computing is limited due to the use of charge as a state variable. We propose the use of body biasing in CMOS adiabatic systems as a design methodology for reducing the need for gradually changing the energy barriers. Simulation results in HSPICE at 22nm are presented which show behavior of a source-memory device operating at sub-Landauer operation. Second, we address whether reversible logic can be used to design sequential computing structures, such as memory devices. we present an analysis of Quantum Turing Machines with sequential reversible logic structures, to show that the entropy gain is substantially less than the Landauer Barrier of $kT\ln(2)$, which is the limiting factor for irreversible computing. A mathematical proof is presented showing bit erasure does not occur in sequential reversible logic structures, and that these devices are physically reversible as long as appropriate delay elements are inserted in the feedback paths to prevent race conditions. This proof validates implementation of sequential reversible logic towards ultra-low power computing.

Next, a novel algorithm for synthesis of adiabatic circuits in CMOS is proposed. This approach is unique because it correlates the offsets in the permutation matrix to the transistors required for synthesis, instead of determining an equivalent circuit and substituting a previously synthesized circuit from a library. Parallelism is used, and the bijective properties of the device to achieve synthesis of the logic structure in $O(n)$ time. Then, using the ESPRESSO heuristic for minimization of Boolean functions method on each output node in parallel, we optimize the synthesized circuit. It is demonstrated that the algorithm produces a 32.86% improvement over previously synthesized circuit benchmarks.

For stronger mitigation of DPA attacks, we propose the implementation of Adiabatic Dynamic Differential Logic for applications in secure IC design. Such an approach is effective in

reducing power consumption, demonstrated using HSPICE simulations with 22nm predictive technology. The benefits of our design are demonstrated by comparing instantaneous power waveforms and observing the magnitude of differential power spikes during switching events. First, simulation results for body-biasing on sub-threshold adiabatic inverters show an improvement in differential power up to 43.28% for similar inverters without body biasing. Then, a High Performance Adiabatic Dynamic Differential Logic (PADDL) is presented for an implementation in high frequency secure ICs. This method improves the differential power over previous dynamic and differential logic methods by up to 89.65%. Finally, we propose a Body-Biased Adiabatic Dynamic Differential Logic (BADDL) for ultra-low power applications. Simulation results show that the differential power was improved upon by a factor of 199.16.

Then, we present an adiabatic S-box which significantly reduces energy imbalance compared to previous benchmarks. The design is capable of forward encryption and reverse decryption with minimal overhead, allowing for efficient hardware reuse.

CHAPTER 1

INTRODUCTION

Production of cost-effective Secure Integrated Chips, such as Smart Cards, requires hardware designers to consider tradeoffs in size, security, and power consumption. In order to design successful security-centric designs, the low-level hardware must contain built-in protection mechanisms to supplement cryptographic algorithms such as AES and Triple DES by preventing side channel attacks, such as Differential Power Analysis (DPA). Dynamic logic obfuscates the output waveforms and the circuit operation, reducing the effectiveness of the DPA attack. In this dissertation, I propose the implementation of Adiabatic Dynamic Differential Logic for applications in secure IC design for stronger mitigation of DPA attacks.

Quantum mechanics principles govern the physical limitations of computing circuits and systems. These systems dissipate energy due to bit erasure within their interconnected primitive structures, which is an important consideration as transistor density increases. Entropy gain in these environments is directly related to the probability of a quantum particle occupying any of its states. In order to design an ideal universal computer that dissipates arbitrarily-low energy, reversible logic must be implemented, since the laws of physics indicate toward reversibility in time.

The basic principle of reversible computing is that a bijective device with an identical number of input and output lines will have no heat dissipation. The electrodynamics of the system allow for prediction of all future states based on known past states, and the system reaches every possible state. There are two separate, yet equally important paradigms with

reversible logic. The first is logical reversibility, which is using the principles that govern reversible logic structure to determine the logical calculations necessary for feasible designs. The second is physical reversibility, which entails design a physical structure whose input values may be uniquely determined by the output at each computing cycle, and whose energy dissipation does not exceed the Landauer Barrier of $kT\ln(2)$ joules per computing cycle. The distinction between these two paradigms is important, because a logically reversible structure may still exceed the Landauer Barrier. For example, a CMOS inverter designed in 22nm technology operating at room temperature (298K) whose VDD is 1V and has an output capacitance of 100pF will dissipate 5×10^{-11} Joules per state transition, which is 1.75×10^{10} times greater than $kT\ln(2)$, even though inverters are logically reversible.

The *Adiabatic Theorem* was first presented by Born and Fock [173]. They describe a physical system as remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum. Therefore, by slowly changing the system conditions, the system itself adapts to the new configuration, modifying the probability density. This means that if the system starts in an eigenstate of the initial Hamiltonian, it will end in the corresponding eigenstate of the final Hamiltonian [1].

1.1 Motivation

Smart cards are small integrated circuits embedded onto plastic or tokens, and are used for authentication, identification, and personal data storage. They are used by the military, in ATMs, mobile phone SIM cards, by schools for tracking class attendance, and storing certificates for use in secure web browsing. They are also used internationally as alternatives to credit and debits cards by Europay, MasterCard and Visa. They are application specific, so their size and

software overhead may be minimized. Additionally, smart cards use tamper-resistant, secure file cryptosystems. They are more difficult to forge than tokens, money, and government-issued identification cards [92]. They can be programmed to deter theft by preventing immediate re-use, making them more effective than cards with magnetic strips. Due to their emphasis on security at both the software and hardware levels, smart-card technology is emerging as the platform of choice in key vertical markets [95]. Smart-card technology is moving toward multiple applications, higher interoperability, and multiple interfaces, such as TCP/IP, NFC, and contactless chips.

Smart cards consist of a secure Integrated Chip, which contains the main processor, ALU, processing registers, random access memory (RAM) for arithmetic processing, read-only memory (ROM) for storing the operating system, and Electrically Erasable Programmable Read-Only Memory (EEPROM) for data memory. The operating system controls data access and implements the cryptographic security algorithms. The international standard for both contactless smart cards electronic identification cards and smart cards is the ISO/IEC 7816 [93], and the contactless smart card is the ISO/IEC 14443 [94]. In this standard, Smart Cards use the Triple Data Encryption Standard, and the standard operating frequency is 13.56MHz.

Since the design of smart cards has been standardized, and their development is moving from single issuer models to co-operative private-public sector partnerships, a two-prong approach to Smart Card security is required: software-systems security and hardware-oriented security. Even though smart cards utilize operating systems with cryptographic kernels, the memory devices used to store them are not isolated in perfectly tamper-proof locations. As a result, analysis of a chip's operation metrics, such as differential power consumption, total execution time, magnetic field values, and radio frequencies allow attackers to gain sensitive

user data. The effectiveness of these side-channel attacks was demonstrated in [98]. Kocher demonstrated in [96] that attackers may be able to find fixed Diffie-Hellman exponents, factor RSA keys, and break other cryptosystems by analyzing power consumption and private key execution time.

The use of power consumption to obtain compromising information is known as a Differential Power Analysis (DPA) attack. The attacker analyzes information gleaned from the practical implementation details of otherwise secure algorithms [97]. Most modern computing systems use CMOS technology, and the dynamic power consumption of a CMOS gate is proportional to its input signals. Therefore, analyzing the output power consumption allows the attacker to determine a correlation between the data and the key, since the switching in the CMOS gates is dependent on those inputs. When the attacker knows the plaintext and the round subkey, they may determine the input to a logic function and may deduce their output by a look-up table. Public key algorithms can be analyzed using DPA by correlating candidate values for computation intermediates with power consumption measurements. For modular exponentiation operations, it is possible to test exponent bit guesses by testing whether predicted intermediate values are correlated to the actual computation.

The effectiveness of a DPA attack may be shown with a simple conventional inverter. Fig. 1.1 shows the simulation input, output and instantaneous power waveforms of a conventional CMOS inverter in 22nm technology. The supply voltage is 0.95V at 1MHz. The average power of the normal inverter is 2.0617×10^{-8} W, with a $P_{Peak_{rise}}$ of 4.5604×10^{-6} W, and a $P_{Peak_{fall}}$ of 1.0325×10^{-5} W, incurring a P_{diff} of 5.7644×10^{-6} W. This means that the peak in power when the input switches from 1 to 0 is 5.7644×10^{-6} W greater than the peak in power

when the input switches from 0->1. Fig. 1.1 shows that this difference is noticeable, and an attacker may correctly determine the logical layout of the circuit.

Figure 1.1. Instantaneous Power Waveform of Inverter in 22nm CMOS.

Simulation of a NAND gate, shown in Fig. 1.2, may be used to demonstrate a more complicated example where the DPA attack remains effective. The average power consumption of this NAND gate is 4.3522×10^{-9} W. The power peak at the '00' input is 1.2533×10^{-6} W, the '01' input is 1.4232×10^{-6} W, the '10' input is 1.4866×10^{-9} W, and the '11' input is 1.1093×10^{-7} W. These differences are sufficient to determine functionality of the circuit, as shown in Fig. 1.2, even if you only have the output and instantaneous power waveforms.

The primary drawback with addressing DPA attacks at the software level is that the power and current variations being analyzed by attacker occur at the hardware level, and no software algorithm, however effective, can affect the operation of a CMOS gate once it receives an input signal. For example, inserting random process interrupts to prevent sequential operation of an algorithm [99] may be circumvented by resynchronization and integration techniques [97]. Also, bit masking [100] can be defeated using DPA attacks.

Therefore, the most effective approach to prevention of DPA attacks is to include security-based logic within the hardware implementation itself in order to make it difficult for

the attacker to ascertain the necessary information to determine the inputs. The three most important metrics to consider when designing CMOS circuits for this purpose are power consumption, area, and operating frequency, since $E_{diss} = C_L * V_{dd}^2 * f$, where C_L is the load capacitance, V_{dd} is the supply voltage, and f is the operating frequency.

Figure 1.2. Instantaneous Power Waveform of NAND Gate in 22nm CMOS.

Thus, in summary, the motivation for this dissertation is to investigate the theory, synthesis, and application of adiabatic and reversible logic circuits for security applications.

1.2 Contributions and Outline of Dissertation

Reversible logic structures are satisfactory for design and implementation in computing structures and organization when those design rules ensure the logic structure is invertible [15]. Therefore, a universal computing machine may be implemented in order to perfectly simulate every finitely realizable physical system, since each electron in the quantum computer is represented by a constant unitary operator in the Hamiltonian space [16][17][18]. Adiabatic logic is an implementation of reversible logic in CMOS where current flow through the circuit is controlled in order to minimize energy dissipation due to switching. Significant research exists in the design and analysis of locally optimal adiabatic elements towards mitigation of side channel attacks. However, none of these works have addressed the use of adiabatic logic in

implementation of flexible and programmable hardware security policies. Nor has adiabatic logic been employed in hardware security applications such as trustworthy voting systems and data encryption standards.

First, I address two major sources of theoretical debate in adiabatic and reversible logic. In Chapter 3, I address whether manipulation of electron flow switching circuits may be done reversible with CMOS logic structures. I present simulation results of *a case study* of Adiabatic logic where a binary switching network dissipates less than $kT\ln(2)$ joules of energy per switching event. In Chapter 4, I address the permissibility of sequential logic in reversible computing systems. First, I present a purely mathematical proof that sequential reversible logic structures are physically possible. Then, I devise a set of computational logic primitives to ensure physical reversibility is maintained for all reversible logic structures with feedback paths.

Next, I present two synthesis algorithms for reversible and adiabatic logic. In Chapter 5, I present a robust behavioral model for the fundamental Integrated Qubit (IQ) gates towards the design of locally reversible logic structures. Modeling IQ gates, as opposed to only Control-V gates or Toffoli gates, allows for a more robust model that more accurately reflects a theoretical reversible computing structure. Then, I present an optimization method for reversible logic synthesis based on the Integrated Qubit (IQ) library. This algorithm runs in $O(N)$ time, and reduces the quantum cost of synthesized circuit by up to 45 percent. In Chapter 6, I present a parallel adiabatic synthesis algorithm for dual-rail adiabatic logic that improves circuit cost by 36.85% over previous benchmarks.

In Chapter 7, I apply the Forward Body Biasing technique presented in [119] to a conventional inverter, an SCRL inverter and an ECRL inverter to demonstrate the effectiveness of body biasing in dynamic differential logic. I present design and analysis using High

Performance Adiabatic Dynamic Logic (PADDL) for mitigating DPA attacks, which is a novel universal cell that performs AND, NAND, OR, NOR, XOR, and XNOR operations. The average power, instantaneous power, and differential power of the PADDL cell is compared to the same metrics of conventional NAND, NOR and XNOR gates. body biasing of NMOS transistors in PADDL is used to improve the operating frequency and differential power of ultra-low power devices. The device presented in this section is a Body-Biased Adiabatic Dynamic Differential Logic (BADDL) cell, and body biasing used to improve operation in frequencies dictated by the ISO/IEC 14443 standard.

Then, in Chapter 8, I present an adiabatic S-box which significantly reduces energy imbalance compared to previous benchmarks. The design is capable of forward encryption and reverse decryption with minimal overhead, allowing for efficient hardware reuse.

Major Contributions

Theory	<div style="background-color: cyan; padding: 5px; margin-bottom: 5px;">Case study in CMOS demonstrating Sub-Landauer operation of an adiabatic circuit. (Chapter 3)</div> <div style="background-color: magenta; padding: 5px; margin-bottom: 5px;">Mathematical proof demonstrating that sequential circuits are permissible in adiabatic circuits. (Chapter 4)</div> <div style="background-color: orange; padding: 5px;">Design and simulation of adiabatic body biased CMOS logic structures. (Chapter 6)</div>
Synthesis	<div style="background-color: cyan; padding: 5px; margin-bottom: 5px;">VHDL Behavioral Modeling and Error Checking of Integrated Qubit Based Reversible Logic (Chapter 5)</div> <div style="background-color: magenta; padding: 5px; margin-bottom: 5px;">Optimization Algorithm for Integrated Qubit Based Reversible Logic Structures (Chapter 5)</div> <div style="background-color: orange; padding: 5px;">Synthesis algorithm for dual-rail adiabatic logic that maps the CMOS switches to the permutation matrix (Chapter 6)</div>
Application	<div style="background-color: cyan; padding: 5px; margin-bottom: 5px;">Design and simulation of Body-Biased Adiabatic Logic Circuits (Chapter 7)</div> <div style="background-color: magenta; padding: 5px; margin-bottom: 5px;">Design and Simulation of Performance- Based Adiabatic Dynamic Differential Logic (PADDL) for DPA Attack Mitigation (Chapter 7)</div> <div style="background-color: orange; padding: 5px; margin-bottom: 5px;">Design and Simulation of Body Biased Adiabatic Dynamic Differential Logic (BADDL) for DPA Attack Mitigation (Chapter 7)</div> <div style="background-color: brown; padding: 5px;">Design and Simulation of Dual Rail Adiabatic S-Box for Implementation in the Rijndael Algorithm in 0.35um, 0.25um, 0.18um, 0.13um, 90nm, 65nm, 45nm, 32nm, and 22nm at a variety of frequencies. (Chapter 8)</div>

Figure 1.3. List of Contributions.

CHAPTER 2

RELATED WORKS

Mathematical models of computation are inherently abstract, since they are not affected by the natural laws of the universe. However, in order to produce a mathematical model that accurately reflects the nature of the computing structure, they must account for how the particles interact, as well as the nature of computing itself. Many of the presented works in reversible logic are *gedankenexperiments* [6] – thought experiments – concerning elastic collisions of classical billiard balls which represent the ideal collisions of quantum particles, or the flipping of quantized spins in order to represent binary logical operations. The result of these experiments is the derivation of a theoretical mathematical model for quantum computation for low-power systems that is applicable for any emerging technology.

2.1 Physical Reversibility in a Quantum System

In a system of quantum particles, it is not possible for a single transfer of heat from a body of lower temperature to a body of higher temperature without another connected change taking place at the exact same time [2]. In a system where N transformations take place, the total change in the equivalence-value is the sum of the equivalence values, which is equivalent to the rate of heat generation divided by the temperature function, $\sum_{i=1}^N \frac{Q_i}{T_i} = \int \frac{dQ}{T}$. Therefore, the sum of all heat transformations in a cyclical process, such as a Carnot engine, must be greater than or equal to zero, which leads to the equality $\int \frac{dQ}{T} \geq 0$. This came to be defined as the entropy of the system, and the difference between the initial and final entropy of the system is $\int \frac{dQ}{T} = S - S_o$ [3]. A

reversible system is the unique instance of the system that where each particle reaches all of its possible states. Since these transformations cancel each other out, the change in entropy is zero, meaning no heat is dissipated into the system.

The entropy of a system is directly proportional to the logarithm of the energy, volume and number of particles in the system, as well as the gas constant [4]. This relationship was simplified, since it is directly proportional to the gas constant and the natural logarithm of the number of microstates that the particles in the system may achieve W , as well as inversely proportional to Avogadro's number. The proportion between the gas constant and Avogadro's number is known as the Boltzmann's constant. The entropy of the system is determined using the equation $\int \frac{dQ}{T} = k \ln(W)$ and the change in entropy in the system is found by comparing the initial and final states of the system $\int \frac{dQ}{T} = k \ln(W_f) - k \ln(W_o)$, where W_f is the number of output states and W_o is the number of input states. Since the entropy is directly related to the number of particles in the system, the distribution of energy elements must be a finite, integral number [Planck 1901]. Electromagnetic energy could only be emitted in discrete quantized amounts, and the total number of possible states must be finite.

The quantum state of all quantum interactions are related to the particles' momentum p , its wavelength λ , the partial derivative of its wave function ψ , its mass m and the potential well created by the particle V , $i\hbar \frac{\partial}{\partial t} \psi(x,t) = -\frac{\hbar^2}{2m} \nabla^2 \psi(x,t) + V(x)\psi(x,t)$ [5]. This relationship was used to present a physical perspective of quantum electrodynamics by utilizing space-time diagrams to interpret electron interaction when they very close to each other [6]. These representations of the state transitions of quantum corpuscles with respect to their amplitude allow for calculations of any quantum process involving energy transfers [7]. A state s_i at a point i in space-time is a given

function $s_i = F_i(s_j \dots s_k \dots)$ of the points, where s_j and s_k are in the neighborhood of i , which is dependent on states that are behind i in time, which allows for calculating of the next state in the diagram. Therefore, in order to simulate time, the representative function F that must be able to predict a future state base on past states as well as know all past and future states. Specifically, in a reversible process, every past configuration of the electrons will be shown, that each possible state will be reached by the quanta, and their interactions will be such that the previous state is uniquely determinable.

2.2 Reversibility in a Universal Computing Machine

In this section, steps towards a mathematical model for synthesis of reversible logic structures will be discussed. Mathematical models of computation are inherent abstract, since they are not affected by the natural laws of the universe. However, in order to produce a mathematical model that accurately reflects the nature of the computing structure, they must account for the how the particles interact, as well as the nature of computing itself. We will discuss how these physical systems, although not yet physically achieved, are models, and why it is important to do so, even though experiment has yet to catch up with theory. Many of the presented works are *gedankenexperiments* – thought experiments – concerning elastic collisions of classical billiard balls which represent the ideal collisions of quantum particles, or the flipping of quantized spins in order to represent binary logical operations. The result of these experiments is the derivation of a theoretical mathematical model for quantum computation for low-power systems.

A universal computing machine is a single automatic device whose operation is completely determined by its interconnected primitive structures, and outputs either a logical ‘0’ or ‘1’, and is able to compute computable sequence [8]. It was initially argued that computing

devices could not perform exhaustively reversible operations, because it was inevitable to avoid calculations whose inputs could not be uniquely determined. This would result in energy dissipation and entropy gain [9]. The theoretical minimum energy dissipation of an irreversible computing device was calculated by relating the number of states attainable by the quantum particles at the input and output of the structure, as well as their probabilities, to the change in entropy, $\int \frac{dQ}{T} = k(\sum p_{out} \ln(p_{out})) - k(\sum p_{in} \ln(p_{in}))$. The scenario where the minimum amount of dissipated heat in an irreversible system occurs when all the input and output probabilities are identical, but there is one fewer output signal. Substituting these values into the entropy equation gives $k_B \ln(2^{N_{in}-1}) - k_B \ln(2^{N_{in}}) = \int_{N_{in}}^{N_{out}} \frac{dQ}{T}$. Solving for the quantity of heat Q , this gave $k_B T \ln(2)$, which is known as *Landauer's Barrier*.

Landauer's argument was that it was not possible to design a computing system comprised entirely of reversible systems, since it was inevitable that they would require a calculation without a single-valued inverse. The result was that any binary device with a single degree of freedom would dissipate energy proportional to the value kT . He presented three systems to support his claim. First, he established a binary system consisting of a single particle within a bistable potential well, where there are forces as each side of the well strong enough to prevent the particle from escaping the well, and a smaller potential in the middle of the potential well separating two local energy minima. Landauer defined the state where the particle was at the left hand side of the well as '0' and the right hand side as '1'. In this particular system, he defined the '1' state as the state of equilibrium where no force is required to attain the state, whereas a force is required for the particle to gain the energy to exceed the barrier in the middle of the well and attain the '0' state. A retarding force would have to be applied in order to keep

the particle at that '0' state. He states that this system is not time invertible, since there is no time-dependent force equation that may be used to determine whether a particle at state '1' was there at a previous time stage or was in state '0'. If the force required to bring the particle to state '0' is no longer applied at time t , and the time required to get back to state '1' is Δt , then we can no longer time invert the location of the particle after time $t + \Delta t$. This is because it is equally possible for a particle to be at state '1' already at time t , and then simply stay there for Δt . Additionally, the required time retarding force adds another potential state for the particle, which exceeds the binary nature of a computing system. A conventional computing system propagates signals through the structure in a manner independent of input data, and is only a function of the physical circuit connections.

Landauer then presented three classes of computing structures to define an irreversible process more thoroughly. The first system was a cryotronic system, which are theoretical devices which do not dissipate energy while holding information, but dissipate energy while switching from state to state. These are different from the previous device because neither the '0' or '1' state exist at a local minima, and thus are not equilibrium states, but this system is more representative of a computer's operation [10]. Logical structure may be designed using devices from the first class, but computing structures can be built that contain either only cryotrons, or only magnetic cores. The second type of system consists of steady-state devices which dissipate energy while holding onto information. These structures have transitions from the desired state to another stable steady state, and are sequential in nature. Devices such as latches, flip-flops and memory structures are in this class. He called the third system a "catch-all" device, where the information stored in the system is dependent on the time-variance of the propagation of the signal through the structure [11]. This is because memory capacity had been shown to be

increased by using small bistable volumes for each storage bit. They could not be made too small, however, since this resulted in increased quantum tunneling, and susceptibility to thermal agitation.

Next, Landauer made the connection between logical irreversibility and entropy generation by presenting a logically irreversible 3*3 logic structure. The probability of each input state – the values of A, B and C – occurring is 1/8. However, the probabilities of each output state – the values of P, Q and C – are different. The probability of $|0; 0; 0\rangle$ and $|0; 1; 1\rangle$ are 3/8 each, and the probability of $|0; 0; 1\rangle$ and $|1; 1; 1\rangle$ are both 1/8. By relating the probability of each output occurring to Boltzmann's equation, he was able to derive the minimal change in entropy for this device as $(3/2)k\ln(2)$.

Since the input state may not be uniquely determined by measuring the output state, it said that an irreversible device loses information, which results in entropy gain. In a binary computing device where all of the potential input and output states may be obtained, the difference of entropy is determined using $k\ln(2^{N_{out}}) - k\ln(2^{N_{in}}) = \int_{N_{in}}^{N_{out}} \frac{dQ}{T}$ When solved for Q in the instance where $N_{out} - N_{in} = 1$, the result $Q = kT\ln(2)$. gives the amount of the minimal heat generation Q per fixed computing cycle. The relationship between the energy dissipated per energy bit, $kT\ln(2)$, known as the *Landauer Barrier*, and is accepted as the theoretical lower limit of computation for irreversible devices. At room temperature (25° C), the value of the Landauer Barrier is 2.805×10^{-21} J per unit time. Landauer showed that this loss of information would result from two additional factors: incomplete switching from one state to another due to excessive clock speeds, and decay of stored information due to thermal fluctuations.

Bennett retorted to Landauer, claiming that simple, robust Turing Machine may be made logically reversible without resulting in information loss [12]. He noted the claim that it was not

practical to reverse the steps of the original computation of the Turing machine [13], which would lead to a practical reversible logic structure, was not complete because a reversible computing structure can be made to erase its history. Bennett explained that it is reasonable for a reversible computer to be able to erase all its results if the computing structure halts, meaning that the structure is still valid if having save the final output calculation and initial inputs. This is because, in a Turing machine, an output tape with random data may only be erased by irreversible means. However, the Turing history tape is not a set of random data. The data on the output is dependent on the logical calculations made by the computing structure. This allows for erasing the output values to reach the initial state.

However, this brought about another problem. If a deterministic, reversible calculation is made, and then the outputs immediately erased in order to reach the initial state through inverse calculation, then the desired output is lost, rendering the computing structure completely useless. Bennett solved this problem by allowing for a ‘Copy’ operation, which allows for preservation of the output data, which still permitting the initial tape to be erased. The copy operation is as follows: the calculations made by the reversible structure are stopped, thus suspending putting output values on the output tape. The tape that the copy is made to must be blank in order to be physically reversible. Once the copy is made, the result may be printed while the calculations are being made in reverse. The result is that the computing structure will restore the initial input while still containing a copy of the printed output. The added benefit is that there is no history saved within the computing structure, yet the device is fully reversible and deterministic.

To demonstrate this formally, let us consider a conventional Turing machine S and a reversible Turing machine R . R may be used to emulate all calculations possible in S , with the additional constraint that at the end of its computation, the structure must have its desired output

and initial input calculations. R will conduct its computation in the following manner. First, it performs the computation similarly to the irreversible structure, except that each intermediate result is saved. Saving these results prevents erasure of bits, which precludes the entropy gain incurred by an irreversible operation. Second, akin to any other computer, the device must print its output calculation. The final stage requires performing the calculations in reverse order, which ensures that the system reaches its initial state, satisfying the physical requirement for reversibility. Now, machine S consists of a read/write head, an infinite tape T divided into squares – each serving as a memory structure – and a controller unit A . The structure uses read-write-shift quintuples in order to govern the intermediate transition steps for calculations whose results are to be put onto the tape.

A Turing machine quadruple $S = (K, \Sigma, \delta, s)$, where K is the set of states, Σ is the alphabet, δ is the set of instructions, and s is the initial state of the machine. Therefore, if we define two quadruples consisting of n tapes, $\alpha \equiv A[t_1, \dots, t_n] \rightarrow [t'_1, \dots, t'_n]A'$ and $\beta \equiv B[u_1, \dots, u_n] \rightarrow [u'_1, \dots, u'_n]B'$, where A and B are the controller units for α and β , respectively, we may define three unique properties required for a reversible n -tape Turing machine. Machine R is made reversible from S by adding transition states which resemble the inverses of the original transitions. For example, the inverse of a Turing "read-write-shift" quintuple requires a "shift-read-write" quintuple in order to attain reversibility. Specifically, simple transition formulae are used in S wherein any tape is subject is a read-write operation or a shift operation, but not both. This distinction significantly reduces the complexity of the quintuples required to make a reversible Turing machine. This is accomplishable if α and β define an inverse mapping of S . This is possible if and only if the inverse is obtained through interchanging the initial control state with the final read-tape symbols, and changing the signs of all the shifts, meaning that

$A=B'$ and $B=A'$. The domains of α and β may overlap if and only if $A=B$, and that every tape i in both quadruples $(t_i$ and $u_i)$ must be equivalent. Additionally, the ranges of α and β may overlap if and only if meaning $A'=B'$ and that every tape i in the quadruple is analogous to the previous one, yet it depends on the final control state and the output tape symbols [14].

The reversible Turing machine may now be defined using these constraints. Bennett showed that, by applying these constraints, there exists for every one tape Turing machine S , there must exist a 3-tape Turing machine R that is reversible in nature such that there exists input and output strings I and P that contain no embedded blanks, and S only halts on the input string if R halts on the input string and the inverse controller B of the other two tapes. Additionally, machine S will product output string P given input strings I if and only if machine R produces I , B and P . Therefore, is S is a universal Turing machine, then R becomes a universal Reversible Turing machine. In fact, every job of computation that may be required by a computing structure is possible to achieve in a logically reversible computer, without inordinate increases in machine complexity, number of steps, unwanted output, or temporary storage capacity.

The table below shows an example of a reversible calculation using the 3-tape reversible Turing machine presented by Bennett. This example is for a specific problem in which the input is a known, calculable function of the output. The device acts as a conventional Turing machine in the first stage. However, the device implements bijectivity, which means that the values on the input at the end of every clock cycle may be uniquely determinable by studying the output. By saving all the output value calculations, the device may retrace all the calculations in reverse order. Since the previous stages were carried out reversibly, this means that all of the transformations cancel each other out, and is completely reversible in nature.

To demonstrate the difference in minimum energy dissipation in reversible and irreversible computing, let us consider the implementation of a Universal Programmable Gate (UPG) [19] that consists of Integrated Qubit gates. The AND gate is not bijective, since the user cannot determine the input combination when the output is a logical ‘0’. Using the AND gate’s truth table and the entropy equations, we determine that the minimum entropy gain of this device is $\int_{N_{in}}^{N_{out}} \frac{dQ}{T} = k_s \left(\frac{3}{4} * \ln\left(\frac{3}{4}\right) + \frac{1}{4} * \ln\left(\frac{1}{4}\right) \right) - k_s \left(4 * \frac{1}{4} * \ln\left(\frac{1}{4}\right) \right)$, which gives a heat dissipation of $\frac{3}{4} k_s T \ln(3)$, which is 1.2 times greater than the Landauer Barrier. The UPG has the same number of inputs and outputs, and is bijective, which gives the equation $\int_{N_{in}}^{N_{out}} \frac{dQ}{T} = k_B (8 * \ln(8)) - k_B (8 * \ln(8)) = 0$, which means the device is physically reversible.

Table 2.1. Three Tape Reversible Turing Machine Operation.

Stage	Action	Tape 1	Tape 2	Tape 3
0	Initial State	Input		
1	Forward S1 computation	Output	History 1	
2	Copy Output	Output	History 1	Output
3	Retraced S1 computation	Input		Output
4	Interchange Output with Input	Output		Input
5	Forward S2 computation	Input	History 2	Input
6	Reversible erasure of copy	Input	History 2	
7	Retraced S2 computation	Output		

In [43], Feynman asked two very important questions which govern our approach towards reversible logic design: *Can physics be simulated with a universal computer*, and *what kind of physics are we going to imitate?* The first question brings about the issue of local interconnections. The major issue with interconnections is that wires are a major source of energy dissipation with computing structures, due to changes in the wire’s voltage and internal resistance [44]. The second involves consideration of how computing is different from physical law. In a computing structure, there are a finite number of input and output combinations, as well

as a finite number of logical calculations. Even the size of the computer is finite. This is different from physics, where space can be measured in infinitesimal distances and wavelengths, and may be summed to infinite size of the universe. Since physical knowledge is always incomplete, the goal is to any mathematical model of a reversible logic structure is to design a *theoretical* structure that beats experiment at the present time. For example, consider the equation for quantum interactions in space-time $s_i = F_i(s_j, \dots, s_k, \dots)$. This allows for a description of the cellular automation by calculating a given point from points at earlier times, which allows for computation of the next values. However, what if the function F depends on all the points in the past? It turns out that the only type of computing system that can successfully imitate this nature of physics is a reversible system. Another important aspect of imitating physics is that we have to be able to predict probabilities. This is because it is impossible to be able to mathematically model each of the individual corpuscles – electrons, photons, atoms, mass particles and harmonic oscillators – in a system. Instead, the modeling accounts for known averages of the relevant dynamical quantities over the entire range of possible system configurations [45]. Within a given volume, there is a density of states and a distribution function, which expresses the statistical average of the number of particles within a given quantum energy state. Numerous statistical methods have been used to model these physical structures – Maxwell-Boltzmann, Fermi-Dirac and Bose-Einstein, to name the prominent ones [46]. In a computer, the total probability is represented by the number of input and output states, 2^N . A reversible computing system is a probabilistic computing system. Just like probability modeling of a system, it is not possible for the computer to model each electron, but the probabilistic computer of a probabilistic nature would allow for imitation of the simulation of the probabilities seen in nature.

To accomplish this, we must consider the nature of universal quantum simulators. For the example presented by Feynman, he considered a system consisting of Bose-Einstein particles. The constraints on a Bose-Einstein system are that have full-integral spins and are not subject to the Pauli Exclusion Principle. But these particles must still be indistinguishable, which means the distribution of the function is determined by the $g_i - 1$ partitions required to divide them into N_i groups [47]. Bose-Einstein systems are often composed of particles that are easily created and annihilated, which means that the number of particles in the system is not fixed. Feynman showed a system consisting for four mathematical operators to represent the creation and annihilation of these particles. The two-by-two matrices represent the two possible base, which are unoccupied or occupied. Consider the matrix for an annihilation, shown in the equations below, and results in the equation $\frac{1}{2}(\sigma_x - i\sigma_y)$. This is the instance where the particle initially occupies the space, and is then annihilated due to a quantum interaction. If it occupies the space, then it becomes unoccupied.

The other possible interaction in this system is *create*, which occurs when a photon is generated due to a quantum interaction. If the quantum state is unoccupied, then it becomes occupied. Additionally, the mathematical model for this system requires the number and the identity matrix. All four equations are shown below.

$$a=\text{Annihilate}=\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \frac{1}{2}(\sigma_x - i\sigma_y)$$

$$a^*=\text{Create}=\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \frac{1}{2}(\sigma_x + i\sigma_y)$$

$$n=\text{Number}=\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = a * a = \frac{1}{2}(1 + \sigma_z)$$

$$1=\text{Identity}=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{2}(\sigma_x + i\sigma_y)$$

The operators σ_x , σ_y , and σ_z are known as the Pauli matrices, and represent one-half spins. These operators allow for the derivation of locally-coupled Hamiltonian with corresponding operators which imitates a discrete quantum mechanical system with a finite number of degrees of freedom composed of Bose-Einstein particles.

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, 1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The theory presented by Feynman for Bose-Einstein systems was proven and expanded upon in [48] by Deustch. Since classical physics assumes a continuous system, it is impossible to simulate using a universal Turing machine. However, quantum physics is discrete in nature. Deustch modified the Church-Turing principle [18][49] in a physical manner to state: *every finitely realizable physical system can be perfectly simulated by a universal model computing machine operating by finite means*. This means that the laws of physics permit the existence of physical models for arithmetic. To this point, the most advanced work on quantum mechanical automaton was in [50], where the quantum cellular automation was used to know and predict the properties of the equations of motion. Deustch built upon this by developing a general, fully quantum model for computation. Such a computer is built upon all the normal Turing operations, plus eight additional operations. These are known as *unitary transformations*, which represent the quantum interactions in a two-dimensional Hilbert space. Given an irrational multiplier α of π , the first four Hilbert space transformations are shown to be $V_0 = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix}$, $V_1 = \begin{pmatrix} \cos(\alpha) & i\sin(\alpha) \\ i\sin(\alpha) & \cos(\alpha) \end{pmatrix}$, $V_2 = \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & 1 \end{pmatrix}$, and $V_3 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$. V_4 through V_7 are the inverse operators, found using the equation $A^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$.

This system may also use two convenient generators which correspond to the 90 degree spin rotations which may be used to compose any of the original eight Hilbert space

transformations. These are known as unitary Lie transformations, and their representation is

$$V_8 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \text{ and } V_9 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}.$$

In [51], Peres developed a method of constructing a locally reversible quantum-mechanical Hamiltonian, whose dynamical evolution may be represented by a series of logical operations. This is accomplished by ensuring each local operation is reversible, and that error-correcting codes are embedded within the Hamiltonian itself. This was done to address a concern that reversible logic was not possible in the presence of noise [52], which would cause bit information loss, and resultant heat dissipation. To accomplish this, Peres made a distinction between *local reversibility* and *global reversibility*. An example of a global reversible system would be a random number generator $x_n \rightarrow x_{n+1} = ax_n(\text{mod } b)$. It is much more difficult to obtain x_n from x_{n+1} than vice versa, since significantly more operations are required. A locally reversible system only consists of a few bits, which means that the reverse calculation requires the same small number of bits. As an example, he presented the “Universal Reversible Gate”, which is now commonly known as the Peres Gate, as a 3*3 reversible logic structure. $P = A$, $Q = A \oplus B$, and $R = AB \oplus C$.

This gate may be derived in terms of spin operations, and performing the Direct Product operation on them. The direct product of matrices involves two given matrices a and b , which will transform the vectors x and y , so $\xi = ax$ and $\eta = by$. Therefore, $a \otimes b$ is the matrix which transforms the appropriately ordered set of binary products $x_i y_j$, so that $\xi_i \eta_j = \sum_{r,s} a_{ir} b_{js} x_r y_s$ [53]. Using the space-time rules derived by Feynman, Peres constructed the Hamiltonian H which represents a unitary evolution of $\psi(t)$, where H is the logical operations done by the “hardware”, and t is the current state of the inputs to the machine, which represents the “software.” Since quantum mechanics in space-time are dependent on logical order as well as

time order, we may use the cursory state $|k\rangle$ after the execution state k , the unitary matrix U_k which transforms the logical state $\psi(k-1)$ to $\psi(k)$, and the inverse unitary matrix U_k^* which transforms the logical state $\psi(k)$ to $\psi(k-1)$, the resulting Hamiltonian for the reversible system is $\mathcal{H} = \hbar \sum_k \omega_k (|k\rangle\langle k-1|U_k + |k-1\rangle\langle k|U_k^*)$. It is determined that the probability of finding the cursor at the final position is $\sin^{2N}(\pi t/2T)$ and the initial position is $\cos^{2N}(\pi t/2T)$. This means that any measure performed at time T may not be a general linear combination. It is instead a quantum nondemolition combination [54]. This means that the quantum limit of weak forces in this system does not exist, which subsequently allows for a computer representation of a logical state in a quantum system that is unchanged by measurement. This ensures that we may develop a reversible Turing machine where we may measure all the resulting output calculations without worrying about Heisenberg Uncertainty.

2.3 Fundamentals of Reversible and Adiabatic Logic

Sadi Carnot demonstrated that an ideal heat engine whose steps are reversible will have the same energy efficiency, $(T_H - T_L)/T_H$, and that there does not exist a heat engine that is more efficient than a reversible engine [1]. This is because the engine would have to create work from nothing, which would violate Newton's Second Law of Thermodynamics. Any reversible Carnot engine must undergo four distinct stages: isothermal expansion, where the volume increases without a change in temperature; adiabatic expansion, where the volume increases while the temperature changes from T_H to T_L ; isothermal compression, where the volume decreases without a change in temperature; and adiabatic expansion, where the volume decreases while the temperature changes from T_L to T_H .

The adiabatic theorem states that "a physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the

eigenvalue and the rest of the Hamiltonian's spectrum" [20]. Since CMOS circuits operate on clock cycles, adiabatic logic design results in a gauge-invariant Berry phase. Normally, when waves are subjected to variations that are self-retracting, then the initial and final states of the system will differ. In order to prevent this, adiabatic systems are designed reversibly so that the system may always reach its initial state, regardless of the number of cycles it operates.

The main obstacle to using adiabatic logic in computing design is that complete adiabaticity means absolutely zero rate of entropy generation. This would require an infinite degree of isolation of system from uncontrolled external environment. In practice, a process is adiabatic to the extent to which their entropy generation approaches zero. The term used for this is "quasi adiabatic."

Therefore, the objective of adiabatic logic design is to use the principles of reversible logic in order to minimize energy dissipation in CMOS circuits. There are two issues that must be addressed in any adiabatic circuit. First, the implementation must result in an energy-efficient design of the combined power supply and clock generator. Second, reversible logic functions require greater logical overhead in order to meet the bijective requirement [21]. Therefore, the energy dissipated by switching of the circuit must be controlled and recycled instead of dissipated into the environment.

The reduction of energy dissipation is achieved through the use of a ramp function instead of the faster switching achieved in step functions. Therefore, transistors may be used in adiabatic operation, despite being demonstrated as lossy devices [21], and is achieved through the application of two rules. First, a transistor is always on when there is a significant current flowing through the transistor. Second, when there is a significant difference between the source and drain voltages, the transistor must be off. In [22], adiabatic circuits were shown to produce a

reduction in energy dissipation of 60% at 20MHz and 35% less energy at 100 MHz, and reversible dual-rail CMOS pass transistors were demonstrated in [23].

An issue for truly adiabatic circuits designed with dual-rail circuits is that they do not permit reversibility across multiple stages of sequential, pipelined logic. This issue was addressed using Split-Level Charge Recovery Logic (SCRL) [24]. Many other schemes were using as many as four-phase clocks to control the propagation of the signal through the datapath, which required greater overhead and constraint of the signals. Efficient Charge Recovery Logic (ECRL) [25] was presented to address this issue by minimizing the required overhead.

Truly adiabatic circuits require the ability to be physically bijective, meaning that the output signals may be placed on the outputs and the unique input signals may be reproduced on the input wires. A dual rail approach was used to accomplish this goal in [26], in which the three fundamental reversible logic structures were design and fabricated in $0.35\mu\text{m}$ technology, where $V_{tp} = 0.6V$ and $V_{tn} = -0.6V$. The circuits have no power supply inputs, meaning that all of the energy from the output signals originated from the input signals. This method improved upon SCRL and ECRL by significantly reducing the overhead required to perform evaluation and discharge, as well as improving the signal propagation, allowing for improving cascading of the devices.

Many early implementations of adiabatic CMOS circuits used diodes in the charge return path in order to provide Electrostatic Discharge protection or device isolation. Diodes operate as a uni-directional current barrier, which means they are physically irreversible. Therefore, for every operation of a device using a diode, a minimum energy dissipation occurs due to information loss. This is because a diode in a reversible operation would operate as “Maxwell’s demon”. James Maxwell proposed a quantum system where faster and slower particles were

separated into two compartments. A miniature demon guarded a door between the two chambers. When a faster particle approached the door, the demon opened it, allowing the particle to enter the other chamber. The result is that the temperature of one chamber will be raised and lowered in the other chamber, but this will have been done without expenditure of work, which would lower the entropy. The contradiction to Maxwell's demon was that, in order for the demon to know which particles were fast and slow, the demon would need to measure the particle's velocity, and the act of gaining this information would require energy, and increase the entropy of the system [139]. This means that a diode that does not dissipate energy would be forced to generate energy from nothing, meaning it would function the same as a perpetual-motion machine. Diodes limit the ability of a quasi-adiabatic device to approach zero entropy. Therefore, diodes are not permitted in adiabatic designs.

These are important considerations, since many of the “adiabatic” logic families that are discussed later in this dissertation are not explicitly or implicitly logically reversible. When we compare our synthesized adiabatic circuits to previous attempts at “adiabatic” secure logic, we will show significant improvement in leakage current and differential power simply because we followed the fundamental rules of reversible logic design.

Body biasing PMOS and NMOS transistors to adjust the threshold voltage (V_{TH}) has been shown to control sub-threshold leakage and avoid large static power dissipation and optimize system performance [27]. This is because the threshold voltage is a function of body-source voltage, which can be modulated for higher performance by forward bias. An added benefit is that the impact of short channel effects decreases as the bias is applied, which also reduces threshold voltage variations.

Applying a forward bias to the transistors decreases *the* threshold voltage and increase device performance. The application of a forward body bias improves V_t roll off behavior and enables the use of shorter gates. Therefore, a forward body-biasing scheme is preferable in some important aspects for extending bulk-Si CMOS technology scaling [28].

Forward body biasing has been used in CMOS to improve the power dissipation and switching capability of dual rail adiabatic inverters by permitting sub-threshold operation [29]. Additionally, body biasing in transistors has been shown to improve the vulnerability of a CMOS circuit against DPA attacks [30][31]. Since the threshold voltage of the pull-up transistors used in charge recovery varies the threshold voltage, the recovery itself is degraded, increasing the differential between peak and average power consumption and making the circuit more vulnerable to power analysis attacks. By implementing body biasing in the PMOS transistors, the dynamic power consumption was reduced by 50% on average, as well as reducing the data dependency on energy consumption.

2.4 Basics of Dual-Rail Adiabatic Design

Two rules must be followed in order to implement adiabatic logic in CMOS. First, a transistor must never be turned on when there is a voltage across it. This means that, if the voltage desired at the drain and source are different, then the transistor must be turned off. If this rule is violated, then energy is dissipated and information is lost. The second rule is that a non-zero voltage must never be applied across a transistor during any transition. If this occurs, then the internal resistance of the transistor is relatively small, resulting in a very high power spike and consequential energy dissipation.

To achieve these design goals, CMOS transistors are used as switches, and they are only “on” when we desire that source and drain voltages are the same. Additionally, the input signals

must be controlled using ramped switching instead of conventional square waveforms, and timed so that no two input signals are switching simultaneously, as this would violate the second rule. Fig. 2.1 shows the proper gate values required for the proper adiabatic operation. A PMOS transistor is “on” when the gate has a value of ‘0’, and a NMOS transistor is on when the gate has a value of ‘1’.

In our design methodology, we use the dual-rail adiabatic presented by Van Rentergem and De Vos [26]. In this method, each switch has a complimentary PMOS and NMOS transistor, which is included to improve circuit reliability. Fig. 2.2 shows the two possible ‘on’ switches that meet the design requirements, and Fig. 2.3 shows the switches with an ‘off’ representation.

Figure 2.1. Requirements for Proper Adiabatic Operation.

Figure 2.2. Adiabatic ‘On’ Switches.

Figure 2.3. Adiabatic ‘Off’ Switches.

The final dual-rail design has no V_{dd} or ground power supply inputs. This design choice is acceptable because the input signals are controlled so that they properly ramp the drain voltage of the transistor switches in order to turn on the transistor only when there is no voltage across it.

Supply voltages are constant, meaning that any transistor that is tied to V_{dd} or ground, has a differential voltage across the drain and source, and is “on” violates adiabatic and reversible principles.

As an example of the effectiveness of dual-rail adiabatic design, we show the design and HSPICE simulation of the Toffoli gate [15]. The Toffoli gate is a 3x3 reversible logic structure which achieves the logical outputs $P = A$, $B = Q$, and $R = AB \oplus C$. Fig. 2.4 shows the design of a dual-rail CMOS Toffoli gate which meets both adiabatic design rules. Fig. 2.5 shows the simulation of the Toffoli gate in HSPICE using the 22nm predictive technology model at 10MHz. The top set of waveforms are the A and A' signals, the second set of waveforms are the B and B' signals, and the third set are the C and C' input signals. The bottom set of inputs are the R and R' output signals. Fig. 2.6 shows the improvement in instantaneous power that the Adiabatic Toffoli gate earns over a conventional four-transistor NAND gate. The NAND gate violates the adiabatic and reversible rules in two ways. First, it is not bijective, so it is logically irreversible. Second, since the switching occurs faster than the adiabatic implementation of the Toffoli gate, a non-zero voltage is applied across a transistor during every transition. The result is that the internal resistances of the transistors decrease during the switch, resulting in the very high power spike visible in Fig. 2.7.

Figure 2.4. Basic Square Circuit of a Dual Rail Toffoli Gate.

Figure 2.5. Adiabatic CMOS Design of a Dual Rail Toffoli Gate.

Figure 2.6. HSPICE Simulation at 10 MHz of Adiabatic CMOS Toffoli Gate.

Figure 2.7. NAND Gate (Red) and Adiabatic Toffoli Gate (Gold) Power.

2.5 Previous Work in Reversible Logic Synthesis

Here, I discuss steps towards a mathematical model for synthesis of reversible logic structures. First, the various reversible libraries, as well as cost and delay metrics used as the

basis for reversible logic synthesis algorithms will be reviewed. Then, work in truth-table and graph-based synthesis of reversible logic structures will be studied. The three major design goals of reversible logic are as follows. First, minimization of the quantum cost - the number of 1*1 and 2*2 reversible calculations necessary to generate the logical output - will reduce the device's computational complexity.

Second, minimization of the delay - the logical depth of the device – will improve the throughput of the device. Third, reduction of the ancillary inputs and garbage outputs - inputs and outputs not implemented in the design of the gate and only serve to maintain reversibility of the device – will improve the design space require to implement the logic.

2.5.1 Unitary Matrix and Truth-Table Based Synthesis

A method was presented for the synthesis of combinational reversible logic structures using a library of Toffoli, Fredkin, CNOT and NOT gates [31]. The algorithm decomposes the output permutations, derived from the unitary matrix into a list of cycles. By removing a cycle, the algorithm is able to decompose it into a list of transpositions, then appends it back to the list of cycles. Once the list has no more decomposed cycles, then it removes the transpositions, creating the necessary sub-circuit in order to produce synthesis. This method was used to design the oracle for the Grover algorithm, along with appropriate Hadamard gates. It was extended in [32] to developed a method for synthesizing optimal reversible circuits within their library, and the algorithm ran in exponential time.

Rademacher-Walsh spectral transformation matrices, $T^0 = [1]$, $T^p = \begin{bmatrix} T^{p-1} & T^{p-1} \\ T^{p-1} & -T^{p-1} \end{bmatrix}$ and two-phase decomposition of Boolean functions were implemented to synthesize circuits up to 4 bits [33]. Dueck and Miller followed in [34] with a transformation algorithm based on Toffoli and Fredkin gates, which included SWAP and NOT gates. They matched the truth table

to the known valid results, and inserted gates to transform the output table to the input functions. They modified this approach in [35] using a set of reduction templates to reduce the output circuit from the algorithm presented in [36]. It was further modified in [37] with a synthesis method that used a bidirectional algorithm and then, applies the template of matching Toffoli/Fredkin networks in order to reduce the cost of the circuit. In [38], output permutation and control input reduction techniques were added to further reduce cost before the template matching method was implemented. Additional templates using the NCT library were presented in [39][40][41]. In [42], a precise formula was derived for the theoretical minimum implementation of the circuit in their library, in that there exists a reversible function that required at least $\left(\frac{2^N}{\ln(3)} + o(2^N)\right)$ gates as a lower bound, and that every reversible functions can be realized with no more than $n2^N$ gates. The result was that, in [43], the Peres gate was added to the bi-directional truth-table based synthesis method, which allowed for reduced implementation when considering Control-V gates.

In [44], transformation rules were developed by first determining the Hamming distance between any required swap operation or CNOT operation, then by searching through the truth-table, those circuit devices were used to develop the necessary circuit. Matrix-based bidirectional algorithms were investigated in [45] that used adjacent matrix which transformed two desired output values from one into the other, and then permutation was implemented to transform the resulting matrix into a reversible circuit.

The unitary matrix approach using Control-V and Control-V+ algorithms was implemented in [46] using a genetic algorithm based approach, which were limited to a 4*4 implementation due to high computational complexity. They defined the set of intermediate values for the control-V and control-V matrix operations, and then use the mutation, crossover

and selection operators of genetic algorithms in order to derive the circuit. This was meshed in [47] with the transformation based method in [48] to produce an more efficient implementation. In [49], the control-V gates to present a set of transformation rules in order to minimize the synthesized circuit using K-maps. The K-maps are used to extract the circuit to determine which nodes may be deleted to preserve the output calculations and reversibility.

Most of the approaches above have started from using either a unitary matrix or a truth table as input to their algorithm. In [49], a method called Synthesis with Output Permutation was introduced as a method to take output functions that have been permuted with their original specification in order to provide exact and heuristic approaches to generating Toffoli networks. By swapping the outputs, it improved a SAT-based approach and a template-based approach. A cycle-based approach was presented in [50], where the permutation was taken as the input, and the Canonical Cycle Form was built in order to extract 3-cycles from the original k-cycle CCF. The required gates to build the circuit out of the disjoint sets are then used to synthesize the circuit.

2.5.2 Graph and Group Based Synthesis

In [51], an approach to reversible logic synthesis based on AND, OR and EXOR operators (sum-of-product expressions), was used to synthesize a set of reversible Boolean functions. In [52], this approach was expanded upon by using positive-polarity Reed-Muller (PPRM) decompositions are performed on a set of input functions represented as a truth table, and then searches a library, represented as a tree of possible factors in priority order to try to find the best possible solutions. The advantage of PPRM is that they only use uncomplemented variables and can be derived easily from the function's sum of products expansion. In [53], the use of PPRM was modified using a heuristic approach in order to plot the reversible circuit into

partitions, then used the priority queue-based search tree as dictated in PPRM. In [54], the authors used the sum-of-product expressions to minimize the standard logic function, which allowed for conversion to a Fixed PPRM. The result was that, by minimizing the input to the PPRM algorithm, the output would be reduced as well. A tool called EXORCISM-4 [55] was used to develop a further reduced sum-of-products. Additionally, they presented heuristics which improved the search tree. Additional methods for improving the priority tree were presented in [56][57][58][59].

In [60], a synthesis technique was presented using shared binary decision diagrams, which was meant to counter the previous truth table approaches. This was permissible, since BDDs are advantageous in applications for Shannon expansions, which the Fredkin gate performs. Using the input permutations to generate binary decision diagrams that took into account every input function, he was able to match the instances where the output was ‘1’ to a synthesized circuit based on the complimentary edges using a library of NOT, CNOT, Toffoli, Fredkin and SWAP. The library and program for this approach, QMDD, was presented in [61]. In [62], quantified Boolean satisfiability (QBF) was used to generate binary decision diagrams to solve the quantified problem formulation. Once formulated, the BDDs were used to generate the reversible circuit.

In [63], reversible Neural Networks (RevNN) were used as a parallel and distributed computing methodology in order to realize the desired output functions using complex function expansions [64][65][66]. The algorithm adds ancillary inputs until the number of outputs and inputs match, then add constants representing ‘0’ and ‘1’ for each output. If that is not reversible, they assigns identical output tuples values which are half zero’s and half ones. Equivalent circuits were added to form the synthesized circuit for each substitution. Ant-colony optimization

[67], an algorithm used by ants to find the shortest path between a food source and their nest, was implemented in [68] in order to try to minimize the number of transformations using a best-path search problem, where artificial ants, starting from their output logical calculations (nest) attempt to find the best path to identity function (food). The algorithm (ants) selectively chooses generalized Toffoli gates during the search process to form the reversible logic circuit.

Group theory was introduced in [69] as a synthesis method, since the total number of possible output combinations on an $n \times n$ logic structure is $2^n!$. By investigation the groups isomorphic to a particular group, they were able to determine which element should represent the coset space, which significantly reduces the problem complexity. For example, a 2×2 gate has 24 output combinations, but the single coset approach allows for these logic structures to be used with only 7 different types of gates, with their inputs and outputs rearranged. Similarly, a 3×3 gate may have 40,320 output combinations, but the single coset approach allows for these to be achieved with only 76 gates. This approach was extended in [70] in order to consider qubit spins in order to minimize the library. In [71], hash tables were generated to compare the function of the output of the reversible logic structure to a cascade of quantum gates. This permitted mapping the output of a gate such as the Toffoli gate to a different permutation when the inputs are switched. The algorithm constructs the gate library, and then determines a minimized circuit using the previous group theory work. In [72], the authors generated a perfect Hash table and incorporated the Peres gate for improving the cost of the circuit.

In [73], a mapping method was presented which allowed users to input a set of non-Reversible functions and realize a set of reversible outputs which gave the desired logical calculations. The irreversible functions are divided into cells. An AND/OR cell is replaced with a cascade of two smaller cells, with the primary inputs to the second cell in the cascade being

provided by outputs of the first one. Additionally, XOR cells are used to provide fanout. Then, a gate packing optimization algorithm was used to reduce the cost of the circuit.

Davio trees were first implemented in reversible logic synthesis in [74]. Davio trees are known to provide optimized circuit synthesis in conventional designs [75]. For a logic function f and a variable x , the positive Davio expansion of f is expressed as $f = xf_{x:(0,1)} \oplus f_{x:(0)}$. If a node of a tree G has the variable x , it allows for positive Davio expansions. The benefit is that the Davio trees share nodes with the same input and output functions, which reduce the complexity of the circuit, as well as input and output lines, and delay. In [76], positive and negative Davio decomposition was implemented as a method to improve upon Shannon decomposition as well as BDDs. The result was that larger functions may be decomposed into smaller functions, which in-turn may be realized with reversible circuit with lower costs as well as fewer inputs and outputs. And in [77][78], a method was presented to generate a weighted, directed graph for reversible function representation and complexity measurement. The truth table is used to create a set of directed graphs connecting the output states that do not match the input states. The vertices represent the input state, and the edges represent the input combination that it receives instead of the identity. The graphs are mapped to circuits which represent the desired logical outputs.

2.6 Clocking Schemes in Sequential Adiabatic Logic

A power gating scheme was first proposed in [195] by Zhang et al., which used *Complimentary Pass-Transistor Adiabatic Logic (CPAL)* and a transmission gate as the power-gating switch. The T-gate for power gating is inserted between the single-phase power-clock and virtual power-clocks in order to detach power-gated CAL logic blocks. They used this methodology to design 8-bit full adders which achieve power savings between 56% and 73%

over a similar implementation without power gating. This was extended into a two-phase clocking scheme in [197] for the design of a power-clock generator used to supply the CPAL sequential circuits, by converting from a single-phase sinusoidal power-clock to the dual-phase clocking required by the CPAL. A two-phase static CMOS logic family (2PASCL) was presented in [200] which shows an improvement of 97% of dissipated energy compared to conventional CMOS logic circuits in 0.18 μ m CMOS. Zhang et al also implemented CPAL in the design of an up counter [220], adiabatic flip-flops with channel length bias [223], security wrappers for SoC technology [236], and an SRAM with an energy-efficient line driver [268].

In [205][208], single phase adiabatic logic implemented in an improved CAL family to gain a 36% improvement in energy with a 10% area penalty. In [209], a 32x32 Content Addressable Memory (CAM) with CPAL adiabatic address decoders and match-line driving circuits was presented. It is partially adiabatic, since the storage cells and driving control circuits were not designed using the CPAL design methodology. This approach did improve upon conventional CAM CMOS devices by 86 percent.

A bootstrap CMOS driver from four-phase adiabatic logic was presented in [201] which improved the area over clocked CPAL by 64% and energy delay product by 37%, as well as improving the capacitive load by a factor of 10. In [203], a 2N2P synchronized oscillator was used to implement a 4-phase clocking scheme that served to significantly reduce energy during idle times.

In [196], a family of cascadable adiabatic circuits called *Glitch-Free CAL* was presented by Reddy et al. that improved upon conventional power consumption by 50%. They are referred to as “glitch-free” because the adiabatic structures are easily cascadable without any glitch and are operated by a single power source, so they do not need multi-phase clocks. In [198], they

extended the GFCAL approach to design an inverter with two-supply waveforms and a single input that earned a power savings of 65% over standard CMOS. In [212], an adiabatic power gating scheme using Multi-Threshold CMOS (MTCMOS) was used to reduce leakage power in dual-gate adiabatic logic by an average of 56%. This approach was improved upon in [213] using the improved CAL logic family from [208] to reduce power leakage by 18% over the previous MTCMOS power gating scheme. In [214], transmission AND gates and wired OR gates were implemented as an array logic sinusoidal power supply for adiabatic logic gates. This improved upon the implementation of the 2PADCL approach by 87 percent.

In [204], a single-phase adiabatic scheme using AC voltage supplies was used to design partially adiabatic D, T, and JK flip-flops, and improved upon the power savings by using power-gating techniques with transmission gates. The drawback to this approach is that it uses adiabatic principles, but it is not physically reversible since it uses locally irreversible structures such as AND and NOR structures. This issue was addressed in [206], where a single phase, quasi-static Dynamic logic approach was used to design adiabatic D, T, and JK Flip Flops. This approach improved upon the CAL approach by up to 75 percent. In [202], the ECRL family of adiabatic circuits to implement a dual-transmission gate register used towards a 32x32 adiabatic register file, and earned a 62% improvement in power savings compared to a conventional register. In [210], the 2PASCL family was used to design a 4-bit ripple-carry adder and a D Flip-Flop. The clocking scheme in this approach is asymmetrical in order to improve the logic transition level. The presented D Flip-Flop improved upon the traditional design by 55% at 100MHz. In [216], an adiabatic shift register was presented which used a dual-phased clocking scheme and was simulated in HSPICE using TSMC 0.25um CMOS, and achieved an improvement of 96.8% in power consumption over the conventional CMOS shift register.

An adiabatic 6T SRAM device was presented in [242] which address the issue of non-adiabatic switching on the energy recovery path of the pass transistors in previous adiabatic designs by providing a forward body bias on the two bit line transistors. And an 8T adiabatic cell was presented in [243] which achieved a 90% improvement over conventional SRAM devices.

2.7 Smart Card History

Before the 1950s, identification and charge cards were paper based, which limited their functionality. The charge company Diners Club issued the first plastic charge card in 1950, which used an inexpensive material called polyvynol choloride (PVC) for durable and robust charge cards [297]. These cards had personal information embossed on the card, with visual features such as security printing to mitigate forgery. However, these cards were susceptible to forgery, and relied heavily upon the clerk's knowledge of the security features. This became problematic as plastic cards began to proliferate.

Security, as well as increased cost, resulted in the development of machine-readable cards. A magnetic strip was embedded into the PVC card, which served as an encryption mechanism to validate the card. This also minimized the paper trail required for authentication. The major drawback to magnetic strip cards is that an attacker with suitable equipment may gain access to the card and read, delete, and manipulate the information on the magnetic strip. The primary defense against this is the use of a Personal Identification Number (PIN), which requires the cardholder to input a number only they know in order to actually use the card. If the attacker is unable to ascertain the PIN, then they will not be able to gain their owner's information. This approach has two drawbacks. First, the PIN is not difficult to ascertain if it is stored on the card or the attacker is able to view the PIN by shoulder surfing. Second, if the PIN is stored on the bank's host system, the data transmission overhead is significantly increased, requiring greater

encryption for the signal and cost to the customer and the bank. Therefore, the need arose for a plastic charge card that performs reliable and secure off-line data transactions.

The first automated chip card was invented and patented in 1969 by Helmut Gröttrup and Jürgen Dethloff [298][299][300]. The card was designed to be both mechanically readable and electrically readable by a system identifier. The “foot” of the card, denoted by ‘38’ contains the integrated circuits used to provide the electrically readable information as well as the notches to hold the card in place while being accessed by the identifier. The rest of the card consisted of visible information such as the holder’s signature and their photograph. By utilizing slides and a rotator, the photograph and signature may be covered so that they cannot be seen. The innovative aspect of this design was that it combined a visual identification method with an electronic integrated circuit verification method. Jürgen Dethloff presented another chip card in 1976 which was the first to specifically guard against misuse and counterfeiting [303]. The security issue occurs when the attacker gains complete control of the chip card. The card bears the identification information, which is assumed to be forgeable. This invention required the use of a Personal Identification Number (PIN) to be entered when the chip card was inserted into the identifier. If the attacker is unable to ascertain the PIN, then they will not be able to gain their owner’s information.

In 1974, Roland Moreno designed the first printed circuit board (PCB) used to store information electronically on a plastic card. He based his invention on the signet ring, a stamp used to leave a raised impression in wax on correspondence between nobility in the 16th century [302]. It was designed to be portable and independent, and meant to be coupled temporarily with an identifier or data transfer device. The IC chip itself contains control circuits for storage on

information, a coupling means for data transfer between the two devices and an identification comparator using a confidential code, and one store element used for permanent storage.

Moreno's patent was revolutionary because it marked the beginning of affordable integrated circuits. This led to the development of ATM, banking and phone cards in France by 1984, developed by the French PTT and Motorola. The 1984 field trial was a breakthrough, as Smart Cards exceeded all expectations concerning tampering and high reliability [297]. Smart Cards were shown to be more effective in comparative tests with magnetic strip cards and optical storage cards. The added benefit was that, unlike the previous technologies, smart card design methodologies could be applied to other applications, making them more portable provided they use the same standards. By 1986, there were 60 million smart cards in Europe [304]. French banks adopted a standardized smart card in 1993, with Germany following suit in 1997. Austria became the first country to develop a nationwide electronic purse system using smart cards in 1997 [9]. The major problem with smart card development at this time was that the focus on design was different in each country, and therefore the smart cards were not interoperable or portable.

In 1993, Europay, Mastercard, and Visa jointly developed the *EMV specification* for smart cards, which ensured that any chip card used by the largest credit organizations would be mutually compatible [306]. There are now one billion smart cards that meet the designated EMV standard, at 15.4 million EMV acceptance terminals deployed around the world. This mean, 36 percent all transactions cards that use PVC, and 65 percent of terminals that accept PVC cards use the EMV standard [307]. The major benefit of the EMV standard was allowing the issuer the flexibility to define their own risk management of the Issues Application Data, and each payment

system being able to define its own EMV chip payment application specifications, while still allowing for mutually compatible systems.

Contactless smart cards were first developed in 1995 in Seoul Korea for electronic ticketing applications. These cards perform the same operations as conventional smart cards, except they involve the transfer of energy and data between the card and the reader without any electrical contact between them. The advantage of this approach is that the user does not necessarily have to hold the card in their hand when the card is required to be accessed, provided the card may be successfully read at the distance the user is from the terminal. This makes contactless smart cards suitable for applications where the user needs to be quickly identified, such as access management [308] and public transportation revenue collection [310].

2.8 Smart Card Standardization

Initially, the French and German telephone cards used different technologies. France used inexpensive EPROM, and Germany used improved EEPROM. The German chip did not require a programming voltage supply input, making the coupling devices for the different smart cards incompatible. As the smart card technology moved towards multiple applications, higher interoperability, and multiple interfaces, such as TCP/IP, NFC, and contactless chips, the need for a smart card standard grew. The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) jointly developed a standard for Smart Card design in 1987 [309].

There are three different types of smart cards which are used based on the specific application: memory smart cards, microprocessor smart cards, and crypto controllers. These chips are distinguished based on their storage memory, reading device interfaces, and the central processing units. Memory smart cards are specifically designed to contain non-volatile Electrically Erasable Programmable Read-Only Memory (EEPROM), as well as a specific logic

for I/O interfacing of the requested data. Microprocessor smart cards contain Random Access Memory (RAM), Read-Only Memory (ROM), and a CPU that can perform specific operations on the chip itself, giving it greater functionality than memory card. The crypto controllers add to the functionality of memory cards or microprocessor cards by encoding and decoding the stored data. They execute the calculation of keys in the required length using efficient cryptographic procedures. The crypto controllers use Ferro Electrical RAM (FERAM) because the writing of data into the memory can be executed much more frequently with reduced power consumption, which is advantageous for contactless smart cards. Since contactless smart cards send an RFID signal, being able to send a faster signal that consumes less power prevents specific types of attacks, making the signal more secure.

The ISO/IEC 7810 standard is used to determine the physical characteristics of ID cards. The ISO/IEC 7811 standard outlines the recording techniques for ID cards. This includes where on the cards embossing may be written, the locations of magnetic stripes – particularly whether they are read-only or read-write magnetic tracks, and the coercivity of the magnetic stripe. The ISO/IEC 7812 standard is specific to how the issuer of the ID card is determined, specifically the numbering system and the application and registration procedures for ID card issuers. ISO/IEC 7813 specifies standards for financial transaction cards. The test methods for ID cards are outlined in the ISO/IEC 10373 standards, and are separated into standards for general characteristics, cards with magnetic stripes, and optical memory cards.

In Sections 2.8.1 and 2.8.2, I will discuss the ISO/IEC 7816 and ISO/IEC 14443/10536 standard in greater detail. The 7816 standard is specific to Smart Card Integrated Circuits with Contacts. The 14443 standard is for Contactless ICs for proximity cards that successfully

transmit signals within 5 inches of the reading device, whereas the ISO 15693 device is specific to devices which successfully transfer data within 50 inches of the device.

2.8.1 ISO/IEC 7816 Smart Card Standard

The ISO/IEC 7816 standard has an eight-contact layout. It consists of an I/O port, as well as ports for the clock, the control-line, the supply voltage and ground. The supply voltage, V_{cc} , is provided to the smart card via contact C1. C2 is reserved for the reset signal which is used to initiate the reset sequence of instructions, C3 is the clock signal provided to the chip which controls the operation speed and provides a common framework for communication between the IFD and ICC. The C4 pin is designated for the CLK signal that is provided by the interface device when the card comes in contact. During the “answer to reset” initiation of the smart card, the CLK frequency is designated as f_i . Once the “answer to reset” is completed, the transmission frequency is designated as f_s (meaning “subsequent frequency”). Pin C5 is designated as the GND pin. C6 is the V_{pp} power connection used to program the EEPROM. This power connection is important, as it is required to supply programming and erasure of the non-volatile memory where the internal operating system is stored. During operation, V_{pp} may exist in either an idle state or an active state. The active state is activated by the interface device when the card desires to update the non-volatile memory. I/O signals are set back and forth from the embedded chip via Contact C7, and it has two modes as well: State A, which pertains to the transmission state, and State Z, which is the reception state. The C4 and C8 pins are reserved “for future use.”

The standard operating procedure of a contact Smart Card requires a specific set of operations when interfacing with the reading device. First, the contacts must make contact with the device and be activated. Second, a reset signal must be sent, which initializes the smart card so that it performs the operations requested by the reading device only, which it then sends an

answer signal confirming that it successfully reset the architecture. Once this occurs, the reading device informs the card of which operations it wishes to perform. Upon completion of the operation, the reader deactivates the smart card, allowing for the contacts to be unconnected, and the card to be safely removed [306].

2.8.2 ISO/IEC 14443/15693 Standards

The major issues in designing a contactless smart card are transferring the energy, clock signal, and data from the power supply in the reading device to the integrated circuit on the smart card. If the smart card is contactless, it will use a Radio Frequency (RF) interface, where the voltage regulator and modulator/demodulator are used to convert the I/O signals to a RFID signal to be aurally broadcast to the coupled device. The mask ROM contains the Smart Card operating system, and the EEPROM is used. The *transponder* is an RF interface which uses the remote memory to store data. The *base station* controls the protocols for communication by the RF interface, mitigate signal collisions, provide authentication and encryption, and interface the RF signal with the microcontroller.

The most widely used technique for RFID communication is *inductive coupling*, since it may be used for simultaneous transmission of energy to power the smart card and the data required to operate it. Energy transfer uses the performed by emitting a 13.56MHz magnetic field through a loosely coupled transformer. Once the contactless smart card passes through the terminal's magnetic field, a voltage is induced upon the coil of the card's RF interface, serving as a power source rectifier.

In order to increase the current levels in the chip to sufficient levels, a capacitor is connected in parallel with the inductor. Data transfer from the terminal to the smart card in contactless smart cards is accomplished using digital modulation techniques such as phase-shift,

amplitude-shift, and frequency-shift keying. When transferring the data back to the terminal, the card generates an amplitude modulation signal.

When the signal is broadcast from the smart card to the interface device, the data is generated by utilizing digital manipulation to obtain a load modulation signal. Once a smart card that is programmed to the required resonant frequency is brought within range of the interface device, the energy required to operate the chip is drawn from the load modulation field. This generates a couple between the inductor in the interface device and the inductor in the card. A coupled field increases the terminal's load current (I_0), which in turn increases the voltage consumption across the resistor in the RLC network (R_i). Therefore, the voltage that runs in parallel with the RLC network (U_0) can be varied – this is known as *amplitude modulation* – by varying the load resistor R_i . Since the data signal may then control the switching voltage consumption of R_2 , this permits the interface device to detect the incoming signal, and perform the required operations.

The variations in voltages on R_i are very small, since the coupling between the two inductive coils is very low (i.e. a few millivolts). Normally, this can only be detected with complicated circuits. This issue was solved by employing a subcarrier frequency which serves as two sidebands at a higher frequency of $f_c \pm f_s$. Then, using bandpass filters, the f_s frequency is isolated and then amplified and demodulated using the inductive coupling network. The drawback to this approach is that it required significantly more bandwidth than the normal direct amplitude modulation, and is limited to a small range of frequencies.

At smaller distances, such as those used in the ISO 10536 standard, data transfer may be accomplished through the use of *capacitive coupling*, where small conductive surfaces are embedded into the card itself, as well as the terminal on the interface device, which are used as

capacitive plates when the card is in range of the device. The surfaces are designed to obtain a capacitance of approximately 50pF, which is sufficient for data transmission. However, this is not sufficient for transferring energy from the interface device to the smart card. The energy required to power the card is still transferred via inductive coupling. This mixing of inductive and capacitive coupling is known as “close coupling” in the ISO 10536 standard, since it permits conduction within small distances.

2.9 SIM Cards and the Java Card Standard

The Subscriber Identity Module (SIM) Card is an embedded circuit used in mobile phones to be able to load the operational programs into the phone once the user has verified their identity through an encryption process. SIM Cards follow the ISO 7816 Smart Card standard, since the phone is used to broadcast and receive data, so the SIM contacts are always in place. Each SIM card has a unique identifier, known as an International Mobile Subscriber Identity (IMSI), a serial number (ICCID), and an authentication key (K_i), and also may require a PIN for gaining access to the phone.

The unique 128-bit authentication key, K_i , for each phone is stored in the device itself as well as in a mobile carrier’s database. The SIM card should not permit K_i to be obtained when the smart card ID being accessed. The authentication is performed using the Run GSM Algorithm, which passes data to the SIM card which is signed with the authentication key. The process for authentication the phone using the SIM card’s authentication key is as follows: First, the phone broadcasts the IMSI to the carrier, and the carrier searches for the IMSI in its database. Once the IMSI is found, the associated authentication key is pulled from the database. The carrier generates a random number RAND, and then signs the message (SRES_1) containing RAND with K_i and the IMSI. Once the mobile phone receives SRES_1, the SIM card

authenticates the message, and then responds to with a message, SRES_2, which contains the encryption key K_c . Once the carrier network receives SRES_2, it compares the K_i and K_c and the computed values. If the results match, then the carrier network authenticates the phone and SIM card for use on the network, and uses K_c to encrypt all messages passed between them.

At one level, the problem of authentication in mobile code systems is very similar to the problem of authentication in distributed systems in general. We have a good deal of both theory and experience in this area; we have methods, mostly cryptography-based, of ensuring within a reasonable doubt that a given person really sent a given message, and that the message has not been altered in transit. On the other hand, some of the known difficulties with standard authentication systems are even more serious in mobile code systems. If we use a digital signature for authentication, and that signature is based on an asymmetric cryptosystem, we must be able to reliably get the public half of the sender's key in order to verify the signature. No strong and general solutions to the key distribution problem are known. Solutions that work well in small groups of people that know each other, or in applications where authentication is only casually required, do not scale well to mobile code or electronic commerce systems where critical transactions with strangers may occur at a high rate, and require strong protection.

In mobile code systems, many programs may be obtained from unknown or untrusted sources. When a program attempts some action, we may be unable to identify a person to whom that action can be attributed, and it is not safe to assume that any particular person intends the action to be taken. When a program attempts some action, we cannot determine whether or not the action should be permitted by simply consulting the rights that have been granted to the user running the program, since the program may well not reflect the intent of that user. The person most relevant to determining the trustworthiness of a program may be someone not known to the

system. There are potentially many security domains corresponding to each user; different actions initiated by the same user may need to be treated differently. Significant security threats come from authorized users running programs which take advantage of the users' rights in order to accomplish undesirable results. SIM Cards must be portable. It is possible that mobile devices will allow “add-on” communications devices to be attached to them, so that a mobile device can temporarily acquire appropriate additional communications.

Power consumption is also an important consideration in SIM card coding standards. This is even more critical for mobile devices than for regular portable computers since mobile devices will have even less space for batteries and because the utility of the services they provide will be severely lessened if people can't rely on them operating for longer periods of time.

The standard for running Java-based applications on SIM cards that addressed all of these issues is the *Java Card Standard*. The purpose of the Java Card technology is to improve the security of the SIM card, as well as improving the portability of SIM cards between different mobile phones. They accomplish this by defining the Java Card Platform computing environment, akin to the JVM in regular computers, as well as the runtime library. The data in the Java Card is encapsulated within the application itself, and the applications are run in a separate hardware space than the operating system. The major difference in the Java Card VM environment and normal Java VMs is that the applications are run separately in order to restrict access between applications. This is an important consideration, since certain applications should only be able to access certain capabilities of the mobile phone. Like most other smart cards, the Java Card standard uses AES and the RSA asymmetric key algorithm. A Java Card coprocessor is an accelerator of Java Card Virtual Machine and implemented by the method of hardware. A low power Java Card Coprocessor need to be designed for the contactless IC card.

The Java Card language run on SIM cards is a subset of the standard Java language, and is Turing complete. The Java Card language is portable enough that it may be run on a Java complied and perform the exact same functionality. Due to the limited space capabilities, Java Card does not support many of Java's features and data types, such as char, multi-dimension arrays, or threads. Similarly, the Java Card bytecode is a subset of the Java 2 bytecode used in the Java VM on regular computer. Akin to the Java Card language, these optimizations are done due to limited resources and space.

2.10 Low Power Medical Device Encryption

This section is case study is from a paper entitled “Low-energy encryption for medical devices: Security adds an extra design dimension”, which was presented at the 2013 Design Automation Conference [317]. The main issue addressed in this paper is maintaining security and privacy in passively powered devices. The cryptographic algorithms require low power implementation that also needs to protect against active or passive tampering. They discuss the various abstraction layers required to mitigate side-channel attack and fault attacks. They conclude that medical devices require the same Kc cryptographic key method used in the ISO-14443 standard. Their new device consumed 5.1 μ J of energy in a 0.13 μ m CMOS technology for one point multiplication.

The authors of paper [317] address security and privacy in medical devices for low power implementations. They assume a wearable medical chip that could be worn or implanted, such as hearing aids or brain monitoring devices [321]. They specifically describe an example called Human++ of IMEC [322], which uses a cellular phone with a SIM card as a mini server in order to collect and process patient data. Mutual authentication is required in order to maintain privacy and not violate HIPAA law [323]. Additionally, they require implementation of security protocols to mitigate a violation that may occur as the result of an adversary gaining physical

access to perform side-channel attacks. The issue with side-channel attacks is that they may ever be used to pick up electromagnetic emanations on a contactless chip [324]. Therefore, the authors determined that incorporating low power design into the best practice paradigm for SIM cards crypto-processors would make for exemplary design for low power, efficient, and secure medical devices.

The authors first addressed the issue of security protocols. There are four main issues in security protocols that need to be addressed. First, any medical device security protocol must be clearly defined and require mutual authentication – akin to the cryptographic protocol in ISO 14443 standard – to prevent attacks and impersonation. Second, location based privacy is a required to mitigate tracking of the user. One such location based security policy is called LoPSiL [326], which provides a location based programming language. Third, asymmetric power consumption between the mobile server and the patient must be considered, since the mobile device is considered to be energy rich compared to the local chip [327]. Fourth, the protocol must minimize power consumption and implementation size as much as possible. The smallest implementation size for an SHA-1 authentication required only 5527 gates [328], whereas the ECC core uses 12k gates [329].

The protocol used in this paper is known as the Peter-Hermans protocol [326], which implements wide-forward-insider privacy, which covers most practical use cases of private RFID identification by using point multiplication. The authors present a point multiplication algorithm to extract performance, the size of the temporary storage, as well as the mitigation of side-channel attacks. Their presented ECC chip also uses a Montgomery powering ladder (MPL) in order to use only the x coordinate of the location in the storage space with 163 bits of memory, and the algorithm may be completed using only six 163-bit registers.

The authors then identify the major issues in optimizing security at the architecture level. At the architecture level, the designer must identify the root of trust in order to reduce the area and/or run time of the hardware security countermeasure. The crypto coprocess may only put sensitive data on the internal data bus, and must mask the data so that it may not be extracted through the ISA. They must also consider area-power-security trade-offs. At the circuit level, they use Wave Dynamic Differential Logic [103] in order to minimize differential power, they avoid data-dependent clock gating since an attacker can determine the security key if different registers are activated at different points of the encryption process, and they isolate the inputs to the datapaths in order to prevent the data from being correlated with the processor's power consumption. They conclude that medical devices require the same *Kc* cryptographic key method used in the ISO-14443 standard. Their new device consumed 5.1 μ J of energy in a 0.13 μ m CMOS technology for one point multiplication.

2.11 Differential Power Analysis Prevention

Sense Amplified Based Logic (SABL) was proposed in [101], and serves as a dynamical and differential logic (DDL) methodology. This implementation uses both complementary and non-complementary signals in order by pre-charging output nodes, which allows the IC to generate differential outputs before an attack can evaluate them. Their method was implemented in 0.18 μ m technology at 1.8V, and it was determined that normalized power variation was 116 times less pronounced than standard CMOS (SCMOS). The main drawback of SABL is that it is difficult to cascade the cells due to signal degradation. It is possible to design complementary circuits to work with SABL to improve the signal quality, but the result is that the differential power becomes sufficient for a DPA attack to be successful [102].

Wave Dynamic Differential Logic (WDDL) was introduced in [103], which uses secure compound gates and combines them with standard CMOS gates during the design automation process in order to mimic the behavior of SABL gates with reduced power consumption. This method gave a reduction of power variations by 52% using WDDL over the SABL method, but gained almost twice as much in terms of average power consumption, area and switching time. Reduced Complementary Dynamic and Differential Logic (RCDDL) was presented in [104] as a 42.29% improvement over WDDL in terms of power consumption. They both use complementary logic structures, but have different switching capacitances due to their inherent physical structures. The RCDDL library built consists of XOR and MUX cells, combined with AND/OR gate implemented using WDDL logic, and two SCMOS buffers with propagation delays equal to 0.3ns and 0.5ns.

Secure Differential Multiplexer Logic using pass transistors (SDMLp) was presented in [105], and uses evaluation and pre-discharge networks in order to implement complementary pass transistor logic (CPL) into dynamic CMOS behavior. The evaluation network consists of PMOS transistors, and the pre-discharge network consists of NMOS transistors. The major flaw with using only CPL is that it only satisfies the differential requirement for secure IC logic, but not the dynamic requirement of one switching per cycle. By implementing the pre-discharge network in SDMLp, the dynamic requirement is met, allowing for use in Secure IC design.

Figure 2.8. Taxonomy of Relevant Works in Differential Power Analysis.

The previous methods have shown significant improvement in differential power over standard CMOS circuits (SCMOS). The SDMLp method showed significant improvement by using an evaluation and discharge network, as well as dual-rail logic. However, they all rely on conventional, lossy logic methods for obtaining the required dynamic logical outputs.

An adiabatic approach was taken in [215] to mitigate DPA attacks using an AND/NAND logic approach. They were able to get an improvement of 47% over the RCCDL approach, but was beat by the SDMLp. The other drawback to their proposed design is that they use diodes in

their cells. Diodes are not physically reversible, and dissipate energy every clock cycle. Therefore, the proposed approach in [215] is not truly adiabatic or reversible. Additionally, a comparison of 2-input AND and NAND logic structures at 13.56MHz was presented in [216] for all ADL, 2N-2DL, ADCL, and dual-rail adiabatic logic. The drawback of these designs is that AND and NAND logic cannot be achieved reversibly with only two inputs, so there are still large differential inputs for certain inputs due to the necessary dissipation for irreversible operations. This means that the input is not truly dynamic, and an attacker can still determine the inputs uniquely from the outputs. In [230], a single-phase Secure buffer circuit was proposed to mitigate DPA attacks. They attempted to circumvent the diode rule by incorporating two diodes, one on the evaluation path, and one on the discharge path. This became problematic as they attempted to clock the design, as they continued to add diodes for every clock signal and every output signal.

Multi-phase adiabatic logic structures were used in [219] to improve the leakage current and power required to perform a DPA attack. The study in this paper was limited to evaluation of phased inverters, and did not address dynamic logic or Smart Card cells. In [235], adiabatic logic was used to design a Content Addressable Memory in a Smart Card in order to extend battery life by up to 18 months. In [236], charge-sharing symmetric adiabatic logic was used to design an 8-bit S-box AES circuit, and improved the differential power consumption over previously used adiabatic approaches by a factor of 53.17. And in [239], quasi-static partially adiabatic logic with diodes was used to present a Radio Frequency Identification architecture with a 64-bit RFID tag test circuit in a smart card which required less circuitry since they were able to eliminate the voltage regulator and full-wave rectifier circuits.

2.12 Previous Work in Adiabatic Dynamic Logic

In this section, I will discuss how adiabatic logic uses the principles of reversible logic to reduce the differential power even further for every possible logic output combination, allowing for a stronger mitigation of DPA attacks in a dynamic manner.

Wires and CMOS buffers dissipate energy despite the fact that they are both logically reversible. Conventional CMOS inverters are logically reversible, yet they do not produce the input waveform if you place the equivalent output waveform on the output pin. Clock signal distribution requires even more energy. Therefore, there are two issues that must be addressed in any CMOS adiabatic circuit. First, the implementation must result in an energy-efficient design of the combined power supply and clock generator. Second, reversible logic functions require greater logical overhead in order to meet the bijective requirement [21]. Therefore, the energy dissipated by switching of the circuit must be controlled and recycled instead of dissipated into the environment.

Adiabatic circuits use ramp functions in order minimize the energy dissipated due to energy as best as possible. First, a transistor is always on when there is a significant current flowing through the transistor. Second, when there is a significant difference between the source and drain voltages, the transistor must be off. In [335], adiabatic circuits were shown to produce a reduction in energy dissipation of 60% at 20MHz and 35% less energy at 100 MHz, and reversible dual-rail CMOS pass transistors were demonstrated for low power operation in [336].

CMOS adiabatic circuits require the ability to be physically bijective, meaning that the output signals may be placed on the outputs and the unique input signals may be reproduced on the input wires. A dual rail approach was used to accomplished this goal in [26][332], in which the three fundamental reversible logic structures were design and fabricated in 0.35 μ m

technology, where $V_{tp}=0.6V$ and $V_{tn}=-0.6V$. The circuits have no power supply inputs, meaning that all of the energy from the output signals originated from the input signals.

Adiabatic Dynamic Logic was first proposed in [106] as a method of implementing CMOS dynamic logic adiabatically, in order to gain order of magnitude in power reduction. They presented an inverter consisting of an NMOS transistor in parallel with a forward-biased diode with an evaluation and pre-charge phase. They reduced the power consumption from $26\mu W$ to $1.7\mu W$ using the adiabatic approach at 100MHz. However, these circuits are not truly adiabatic, since diodes are not reversible in nature [21].

The pre-charge and evaluate approach from [108] was implemented in [107] as a wave-pipelined Dynamic Adiabatic MOS (DAMOS) allowed for 73% energy recycling at 200MHz when fabricated in $0.25\mu m$ technology at 2.5V supply voltage. The issue of overhead and clocking was addressed in [109] with the implementation of a single-phase clock. They gained a reduction of 55% using $0.18\mu m$ technology with a tradeoff of doubling the operation time, although the operation was significantly reduced when compared to multi-phase clocking schemes. This was improved even further in [110] with Quasi-Static Single-phase Adiabatic Dynamic Logic (SPADL). They gained a 87% improvement in power consumption over static CMOS and 75% over a single-phase adiabatic clocking schemes by using a sinusoidal clock as well as a step function. In [111], adiabatic logic was used to demonstrate that adiabatic logic families, such as SCRL and SPADL, consume constant power during the pre-charge and evaluation stage, enabling mitigation of power analysis attacks.

In [231], Resonant Dynamic Logic was presented to use an inductance-capacitance network to store and discharge energy in an adiabatic system. They were able to achieve better than 50% energy dissipation reduction by using these networks at 500 MHz.

2.13 Dynamic Information Flow Tracking

Implementation of DIFT in hardware was proposed in [123] for the MINOS architecture, and specifically addressed control data attacks that overwrite the return address on memory pointers. The architecture presented in [124] addressed both control and non-control attacks by throwing an exception if a tag of an index is added to an untainted pointer with a pointer arithmetic instruction, and was meant to improve upon the buffer overflow protection provided by StackGuard [125] and StackGhost [126]. It also allowed for multi-grain mechanisms for managing tag storage, which reduced memory overhead to less than 2%. This approach improved upon the false positives for security exceptions, since many programs use bound checks for input validation. However, many false negatives still existed for common attacks. This issue was addressed in [127] by clearing the tag when tainted data is compared to untainted data and not clearing pointers on pointer arithmetic. DIFT was also implemented to prevent information leaks by tracking explicit and implicit information flow

The main drawback of DIFT implementation up to this point was that it was not flexible, and had a hard coded security policy. MINOS was unable to address any data attack that did not specifically address the architecture's control mechanism. The architecture presented in [128] is limited by the architecture's definition of pointer arithmetic instructions. The architecture in [127] still produced many false positives and negatives, since it assumes that comparisons implement validation through bounds checking. Software systems may also use validation methods such as using a modulo operation on hash table indices, or performing a logical AND operation on UID hash tables.

The Raksha architecture [129] proposed a set of flexible and programmable hardware security policies for tag tracking and propagation which allowed for fine-grain implementation in

software in order to allow for evolving attacks. It supports multiple active policies that address both high-level attacks at the software level, such as SQL injection and cross-site scripting, and low-level attacks at the hardware level such as buffer overflow attacks. Raksha's exception mechanism reduced the exception overhead to 30%, which improved upon software implementation using OS traps by 55.41% for performance benchmarks such as gzip, gcc, gap and twolf. It was also the first DIFT system to prevent high-level attacks such as directory traversals on unmodified binaries.

2.14 Encryption and the Rijndael Algorithm

Public key encryption algorithms permit message passing between a transceiver and an embedded device via Radio Frequency Identification (RFID) by using two different keys for encryption and decryption. Diffie and Hellman developed public-key encryption in order to reduce the reliance on secure key distribution channels [337]. They accomplished this by requiring the user to obtain both a public key and private key in order to encrypt and decrypt messages. Only the user possesses the private key, meaning that only a valid recipient may decrypt the message. The Rijndael algorithm [333] was selected in 2000 by the National Institute of Standards and Technology (NIST) as the Advanced Encryption Standard [338]. The Rijndael algorithm is an iterated round-based symmetric block cipher, which encrypts and decrypts data in 128-bit blocks with a key of 128, 192, or 256 bits in order to provide an effective protection of transmitted and stored data against cryptanalytic attacks [339][340]. By applying four transformations to the plain text - an initial round key addition, standard rounds, and a final round, the data is encrypted using the public key and the decrypted using the private key in an efficient manner.

The Rijndael algorithm [339][340] is the NIST’s accepted standard for the Advanced Encryption Standard (AES). Any Rijndael architecture must be able to implement a variable key/block length iterated block cipher using an initial data/key addition, a number of standard rounds, and a final round. The initial inputs into the 128-bit implementation of the Rijndael algorithm consist of an input plaintext block consisting of sixteen 8-bit states, and a cipher key consisting of sixteen 8-bit blocks. The *ByteSub* operation performs the multiplicative inverse of a Galois field, $GF(2^8)$ followed by an affine transformation. The decryption performs this process in reverse. The result is that the ByteSub operation acts as an 8-bit lookup table, where an input value correlates uniquely to an output value, and vice versa. For a 128-bit encryption, the plaintext is broken into 16 byte-size pieces. The four most significant bits refer to a “row” in a S-box table, and the four least significant bits refer to the “column” in the S-box table. The result used to replace the old byte in the key. The S-Box is considered the most important aspect of the AES cipher for speed, size, and mitigation of DPA attacks.

2.15 Previous Work in S-Box Hardware Design

Hardware designers of Rijndael ciphers must consider tradeoffs between high performance computing, low power computing, and design area. An initial implementation of the design in hardware was presented in [335], which required 54 clock cycles to perform an encryption, and achieved 311 Mbit/sec throughput. The main drawback to their design approach is the length of the critical path. The ByteSub, MixColumn, and AddRoundKey operations are performed for one column within one clock cycle. This was addressed in [340], where the authors presented two designs which considered tradeoffs in performance and area, and were synthesized in 0.6 μ CMOS technology. The standard implementation of the S-box design required 16 data cells and 4 S-boxes, whereas the high performance implementation required 16

Sboxes. However, the high performance implementation required only 34 clock cycles, while the standard implementation required 64 cycles, and this translated to a throughput of 241 M/sec and 128 Mbytes/sec, respectively. The encryption/decryption optimization modules for the Rijndael algorithm were first presented by Kuo et al [342]. The hardware modules were reused in order to optimize area and power consumption. Additionally, the round key was generated in real time, using the S-box, allowing the entire encryption process to be executed on a single chip architecture.

Intelligent S-box design must consider tradeoffs in power, area, and operating frequency. In [344], a five-stage smart card processor was augmented to include secure instructions capable of performing S-box operations. The S-box consumed 165 μ W in 0.25 μ m technology. The reduction in power was obtained through removal of compiler analysis. The power consumption was not masked, so they added masking signals to hide the power, which added another 45 μ W. An 8-bit architecture for the AES was presented in [345] which implemented the S-box as a set of LUTs. In [355], a four-stage pipelined S-box was used to reduce the throughput of [345] by a factor of 2.1. In [348], a high performance S-box with error correcting codes was presented to operate at 2.4GHz. A series of S-box designs was presented in [349] which allowed for tradeoffs in design metrics. These included using LUTs, unified architecture, bitslice architectures, and unified masked architectures, which were used to improve upon the designs in [350][351][352].

Numerous S-box designs have considered some form of masking the power signal in order to disassociate the waveform and the functionality of the circuit. Masking of the power signal was achieved in [355] by processing intermediate values during calculation of the S-box value by calculating the logarithm of the input values and affine transformations.

Dual-rail logic was implemented as a design methodology for mitigation of Differential Power Analysis (DPA) attacks [343]. By using dual-rail logic, special instructions may be encoded in order to provide data independence from the switching activity of the circuit [344]. Energy balancing is address in dual-rail logic by incorporating spaced between adjacent clock cycles, which guarantees all gates switch in every clock cycle. The main drawback to the dual-rail approach is that there is more logic, which requires a greater number of gates, which increases both area and power consumption. Additionally, spacers and negative gate optimization is required to implement a dual-rail circuit that successfully masks power signals.

Pass transistor and adiabatic logic schemes have been used to mitigate DPA attacks, but never concurrently. In [353], multiple adiabatic implementation methodologies - such as CSSAL, SyAL, 2N2P, and ECRL – to design low power S-boxes. The CSSAL showed significant improvement over the other methods at 0.18 μm . And in [356], a charge-sharing symmetric adiabatic logic (CSSAL) was used to improve upon the energy imbalance of the S-Box.

2.16 Previous Work in Reversible Logic Architectures

A conservative logic computer for use in a graphical simulator was first presented in [130], which utilized a garbage stack to collect extra bits from calculations, and used the ISA properties to produce the reverse calculations. The Insentropic ISA presented in [131] was the first to be able to conduct forward and reverse operations, and is a reduced, reversible version conventional PDP-10 ISA, the difference being that Insentropic does not contain byte, halfword, or floating point instructions. Pendulum was presented in [132][133], and was the first reversible microprocessor to produce a fully reversible instruction set, and the intermediate results were left uncomputed, akin to the Bennett reversible Turing machine. DEM1 and DEM2 were presented in [135] using the reversible control FSM technique presented in [134] in order to implement an

adiabatic reversible bidirectional MIPS datapath in 90nm technology at 200MHz. In [136], a reversible programming language, SyReC, was proposed and used in the implementation of a reversible ISA capable of 8 arithmetic instructions, 8 logic instructions, 5 branch/jump instructions, and 4 load/store instructions. And in [137], a reversible ISA, Bob, is capable of 17 instructions. They modified load/store instructions since they are inherently irreversible, with an *exchange* instruction, which swaps out register values with a value at a given address in memory.

Figure 2.9. Taxonomy of Relevant Works in Adiabatic Security.

In [211], adiabatic output pins were presented in order to improve upon the energy consumption of driving chip pads due to their large load capacitances. They proposed two output pads using the CPAL family and the PAL-2N family, and earned an improvement in power consumption of 55% at 100MHz.

CHAPTER 3

MITIGATING COMPUTING DEMONS

In this chapter, I address whether manipulation of electron flow switching circuits may be done reversible with CMOS logic structures. I present simulation results of *a case study* of Adiabatic logic where a binary switching network dissipates less than $kT\ln(2)$ joules of energy per switching event. It is shown that the energy of the bit is larger than $kT\ln(2)$, yet the resulting dissipation is less than $kT\ln(2)$. I show that the presented structure operates robustly under process and temperature variations. I then show that the device operates beneath the Landauer barrier when reversible operations are performed, but does not when irreversible operations are performed. I present simulation results where a CMOS-based 22nm device [138] conducts sub-Landauer operation outside of the noise margin, and the results are analyzed in terms of switching time, robustness and power consumption.

3.1 Charge Based Computing Debate

The concept of a reversible system was first challenged in a thought experiment known as *Maxwell's Demon*. James Maxwell proposed a quantum system where faster and slower particles were separated into two compartments. A miniature demon guarded a door between the two chambers. When a faster particle approached the door, the demon opened it, allowing the particle to enter the other chamber. The result is that the temperature of one chamber will be raised and lowered in the other chamber, but this will have been done without expenditure of work, which would lower the entropy. The contradiction to Maxwell's demon was that, in order for the demon

to know which particles were fast and slow, it would have to measure their velocity, and the act of gaining this information would require energy, and increase the entropy of the system [139].

Scaling of computing structures to smaller sizes and faster switching speed results in improved performance in accordance with “Moore’s Law.” However, energy dissipation due to interconnected primitives has limited improvements in operating frequency. The reason for this heat emission, and methods to mitigate its effects, is the source of significant debate. Rolf Landauer argued that not possible to design a computing system comprised entirely of reversible systems, since it was inevitable that they would require a calculation without a single-valued inverse. The result was that any binary device with a single degree of freedom would dissipate energy proportional to the value $E_B = kT \ln(2)$ joules for every loss of information, where $k = 1.3806 \times 10^{-23} \text{ J-K}^{-1}$ [9]. This gives $E_B = 2.853 \times 10^{-21} \text{ J}$ at 25°C . He also suggested that implementing a computing system reversible reintroduced the demon, since they require the quantum state of the particle to be measured in order to sort the molecules. In [12], C.H. Bennett retorted to Landauer, claiming that simple, robust Turing Machine may be made logically reversible without resulting in information loss. Fredkin and Toffoli used the findings of Bennett, as well as the physical properties of reversibility to develop a computation model to reflect and simulate fundamental physical principles [16]. Bennett also argued against Landauer’s demon by noting that the demon would run out of storage space and be forced to discard information. Bennett allowed for this eventuality in the design of his reversible Turing machine by allowing for a ‘Copy’ operation, which allows for preservation of the output data, which still permitting the initial tape to be erased.

Information is represented in modern computers with electric charge, where electron flow is manipulated by conducting gates in digital transistors. Cavin et al. argue in [140][141] that

manipulating electrons in computing structures using energy barriers requires another state variable, and this would require an irreversible operation, necessitating a minimum entropy gain of $kT\ln(2)$ every cycle *in every case*. They also state that this necessary charge would limit the scaling of CMOS structures, and that the associated gate and channel leakage would require new materials, processes and nonclassical MOSFET structures to be scaled to and beyond 22-nm technology. In [142], Bloecher et al. argue this analysis is flawed, since the copy operation from Bennett's Turing machine was not considered. They presented experimental results with a RLC circuit is excited by a sinusoidal wave where the dissipated energy was 160 times less than the Landauer Barrier. Cavin retorted in [143], claiming that the energy dissipated by the power source must be considered, and that their analysis considered all physical systems. This was disputed in [145], where it was noted that the original argument in [141] focused solely on charge carrying systems, and that consideration of the clock generator energy was not important, since resonant circuits sufficiently address the issue of recycling energy in the clock signals. In [144], Orlov et. al presented experimental results where an RC network of a 1.1k Ω resistor and a 100pF capacitor implemented as a source-memory device performed sub-Landauer at 50 μ V at switching speeds of 640 μ s.

The issue which remains in debate is whether switching devices such as CMOS transistors require a charge state variable, and whether this permits or forbids sub-Landauer operation. Conventional CMOS does waste energy, and has been shown to be lossy [146], due to the heat dissipation at each logic transition. Adiabatic logic is a CMOS design paradigm where the current flow through the circuit is controlled such that the energy is recycled rather than dissipated since the input and output charges are kept separate [147]. Therefore, the leakage power and energy barriers are reduced, mitigating the effects of Cavin's demon. Since the energy

required for a switching event in a CMOS circuit is $E_{diss} = C_L V_{dd}^2 f$, the tradeoff in using CMOS adiabatically is in operating frequency.

3.2 Proposed CMOS Source-Memory Device

For our simulation, I present a combinational reversible logic structure which utilizes a PMOS/NMOS network driven by a System input of +Vo, -Vo, and 0V. This is the source bit, which was first claimed to be reversible in [149], and is identical to the experiment performed in [144]. The difference is that a CMOS network is used instead of an RC network. The circuit will be operating in sub-threshold voltages in our simulations.

The circuit performs two reversible calculations: “Copy S to M” and “Erase with a Copy”. These operations were compared to an irreversible “Erase without a Copy” operation. The energy held by the bit (E_{bit}) is determined by the charge of the output capacitor, and is given using the equation $E_{bit} = 1/2 * C_L V_o^2$. In Table 3.1, the truth table for each of these operations is shown. The distinct states in this table are represented as a binary “1” (+Vo), a binary “0” (-Vo), and a “null” state which holds no information. The adiabatic network with CMOS transistors is shown in Fig. 3.1.

In our simulation, I show the following set of operations. First, the switch is moved from 0V to +Vo, and measured for the given rise time for the circuit. Each time slice for all the operations in this simulation are identical. This simulates “Copy S to M '1'”. After another time slice, the switch is moved to the 0V, which simulates “Erase '1'”. After another time slice, the switch is moved from 0V to -Vo, which simulates “Copy S to M '0'”. Again, the switch is moved back to 0V to simulate “Erase 0”. Finally, the switch is moved to the loop position, allowing for measurement of the next clock cycle based on the signal from the feedback-producing output. The input simulation waveform is shown in Fig. 3.2.

In order for a CMOS circuit to operate outside of the noise margin and maintain robustness, I must compare the energy dissipated from the circuit in Fig. 3.2 with the energy stored in the capacitor. The energy stored in a capacitor is represented by $E = \frac{1}{2} * C_L V_O^2$. In all of our simulations, I use an output capacitance of $C_L = 5fF$. A circuit with a supply voltage of 2mV would store $1 * 10^{-20}$ Joules.

Table 3.1. Truth Table for Source-Memory Device Operations.

Operation	S _{IN}	M _{IN}	S _{OUT}	M _{OUT}
Copy S to M	0	Null	0	0
Copy S to M	1	Null	1	1
Erase with a Copy	0	0	0	Null
Erase with a Copy	1	1	1	Null
Erase without a Copy	Null	0	Null	Null
Erase without a Copy	Null	1	Null	Null

Figure 3.1. Presented CMOS Source-Memory Device.

Figure 3.2. Input Simulation Waveform.

3.3 Simulation Results

The presented simulation results were obtained by using the 22nm low-power predictive technology model developed in [22] and run in HSPICE. Simulations were run for circuits with supply voltage ranging from 1V all the way down to 500 μ V. I placed a series of constraints on what I considered to be a successful operation of the circuit. Since I need to show that the circuit operates properly outside of noise margins, I defined a “successful operation” as one where the output voltage reached 95% of the supply voltage during all copy operations, and held no more than 5% of the voltage after Erase operations. Figs. 3.3-3.6 show HSPICE simulation results for the energy dissipation, time delay product, and average power consumed for each operation of the source/memory device. Each sweep ran the switching time from 100ms to 1ms. In each figure, there is a section where the tdp is equal to 0. This means that the simulation did not pass our criteria, and is considered an unsuccessful operation.

The first successful circuit I were able to simulate was with +Vo=1.4mV and a switching time of 60 μ s. The energy dissipation values are shown in Table 3.2. The presented CMOS has a supply voltage 28 times greater than the sub-Landauer RC circuit in [144], and a switching time 10.67 times faster while achieving the same operation. This circuit has a supply voltage of 1mV and a switching time of 250 μ s, which means the supply voltage is 20 times higher and the switching time is 2.56 times faster than the RC experiment presented by Snider. The operation of the 1mV/250 μ s device is shown in Fig. 3.7. I also present simulation results for instantaneous power and current results in Figs. 3.8 and 3.9, respectively.

3.4 Noise Considerations

In the previous section, I presented simulation results for a sub-Landauer 1mV, 250 μ s CMOS source-memory device. This device operates at a voltage that is significantly sub-

threshold. Therefore, to further support our simulation results, I present a discussion and simulation results concerning noise in CMOS operation.

Table 3.2. Energy Dissipation for PMOS and NMOS Circuit (Joules).

Operation	Ediss Vo = 1.4mV t _{sw} =60μs	Ediss Vo = 1mV t _{sw} =250μs
Copy '1'	2.8116E-21	1.8688E-21
Hold '1'	1.7954E-21	1.3235E-21
Erase '1'	1.7476E-21	8.6546E-22
Copy '0'	2.8102E-21	1.8744E-21
Hold '0'	1.4948E-21	1.5047E-21
Erase '0'	4.9396E-22	6.2761E-22

Figure 3.3. Energy, Time Delay Product and Power Analysis for 10mV.

Figure 3.4. Energy, Time Delay Product and Power Analysis for 5mV.

Figure 3.5. Energy, Time Delay Product and Power Analysis for 1mV.

Figure 3.6. Energy, Time Delay Product and Power Analysis for 0.5mV.

Figure 3.7. Input (Green) and Output (Yellow) Waveforms for Adiabatic Circuit.

Figure 3.8. Instantaneous Power Waveform for Adiabatic Circuit.

Figure 3.9. Supply Current Waveform for Adiabatic Circuit.

The issue of noise was initially raised in [150][151] as a detriment to reversible and adiabatic computing. They claimed that even the slightest noise disrupted the adiabatic process, since the noise affected the accuracy of reading the output values in the reversible Turing machine. This was countered in [82][152], where Peres developed a method of constructing a locally reversible quantum-mechanical Hamiltonian, whose dynamical evolution may be represented by a series of logical operations. This is accomplished by ensuring each local operation is reversible, and that error-correcting codes are embedded within the Hamiltonian itself. To accomplish this, Peres made a distinction between *local reversibility* and *global reversibility*.

The presented device was design to take advantage of energy recycling in CMOS in order to create a locally reversible circuit with appropriate error correcting codes. To demonstrate this, I show simulation results of a CMOS circuit that does not account for the appropriate error correction codes. This circuit will perform the “Erase Without a Copy” operation shown in Table 3.1.

Our device takes advantage of energy recycling in CMOS in order to create a locally reversible circuit with appropriate error correcting codes. The energy is recycled by controlling

the flow of the current through the circuit. During the Copy '1' operation, the circuit maintains positive current flow, as shown in Fig. 3.4. During the Erase '1' and Copy '0' operations, there is negative current flow, and the 'Erase 0' has positive current flow. This is a product of properly connecting the memory device to the appropriate voltage source in order to gradually ramp the voltage. This allows the system to properly capture all of the dissipated energy.

To demonstrate why this is an important consideration in both noise and reversible operation, I simulated the operation of the CMOS circuit which does not account for the appropriate error correction codes. This circuit performs the "Erase without a Copy" operation shown in Table 3.1. The operation stores a value in the memory, and then erases by dissipating the value in the memory. This is both logically and physically irreversible. This is because the value of the memory after "Erase without a Copy" is ground, regardless of the previously held value. This is physically irreversible since the memory is not connected to the appropriate voltage source, which negates any error correcting codes. This means that the capacitor has to be discharged in order to erase the bit. Therefore, a copy of the bit is not made, and the energy is dissipated into the surrounding environment. This means that physical reversibility is violated as well. The amount of energy dissipated in this operation is equivalent to $\frac{1}{2}Cv^2$. This value is 5×10^{-21} J, which is 1.77 times greater than the Landauer Barrier.

CHAPTER 4

PERMISSIBILITY OF SEQUENTIAL COMPUTING

In this chapter, I address the permissibility of sequential logic in reversible computing systems.¹² First, I present a purely mathematical proof that sequential reversible logic structures are physically possible. Then, I devise a set of computational logic primitives to ensure physical reversibility is maintained for all reversible logic structures with feedback paths. The presented proof in that paper assumes that the appropriate delays are inserted in the feedback paths to prevent race conditions. While I maintain that sequential reversible logic structures are physically reversible, I acknowledge the concern raised by Nielsen and Chung in their textbook *Quantum Computation and Quantum Information* [153] that failure to account for race conditions has an impact on reversibility and maintaining information. However, I show their claim that this completely prohibits sequential circuits in quantum computers is flawed, and I disagree with the notion presented in subsequent papers that reversible logic devices may only be represented using Directed Acyclic Graphs, such as [154][155][156][157][169] that cite that text as justification.

4.1 Sequential Reversible Logic Debate

Sequential computing structures are efficient, low-cost devices that use feedback paths in order to reuse subroutines, so that the device may refresh the information being stored before it is

¹ Portions of this chapter were published in the IEEE Symposium on Very Large Scale Integration [163]. Permission is included in Appendix A.

² Portions of this chapter were published in IEEE Conference on Nanotechnology [164]. Permission is included in Appendix A.

lost. There is a debate as to whether or not feedback causes bit erasure. This would prevent a computer from being able to retrace all of its computations, making the design of a Quantum Turing Machine with feedback impossible. Many texts and papers state emphatically that feedback is not permissible in a quantum computing structure under *any* circumstances. Others say this is “a common misperception” [21]. Paper espousing both points of view have been published within the last [160][157]. Establishing laws and metrics for these devices is essential to determining the natural bounds of computing design for any computing technology.

In order for a quantum reversible computing machine to satisfy the constraints set forth by Bennett, the device must be completely reversible in time, meaning that – given the time-dependent set of outputs – the circuit will be able to reproduce the inputs by operating the circuit in reverse. The focus of the remainder of this paper is how the introduction of feedback paths affects this necessary property of quantum computers.

Fredkin was the first to support the notion of sequential reversible circuits [160]. In his approach, there is an instantaneous combinational element and a feedback wire which has a delay element embedded to ensure the time iterative nature of the reversible structure is maintained. To that end, he presented a JK Flip Flop composed of a Fredkin gate and an inverter on the feedback path to demonstrate any computation that can be carried out by a conventional sequential network can also be carried out by a suitable conservative logic network, provided that an external supply of constants and an external drain for garbage are available. Toffoli presented a framework for condensing a time-iterative network into a reversible sequential network by having the same operation used repeatedly, and inserting a delay element in order to ensure that the time-dependence on the stages is maintained [15]. Through a proof, he concluded that any operations that may be computed by an arbitrary finite automaton may also be computed

with a reversible finite automaton. And Deutsch presented a 4-bit universal logic structure [149], identifying the initial state of the feedback path as a source bit, and claimed it was sequential and reversible in nature “in spite of the unit wire that loops backward.”

However, most of the research in reversible logic synthesis is based in the notion that sequential reversible logic structures are not feasible. They claim that “feedback is not permitted” and that graphical representations of reversible logic structures must be Directed Acyclic Graphs. They make the convention that loops are not permitted, since circuits containing cycles may become unstable, which would result in information loss, forbidding their use. Those who support sequential circuits in reversible logic will claim that since the circuit is composed entirely of reversible primitives, then it must be reversible. However, this is not entirely the case either. The commonly cited reason for rejecting sequential circuits comes from [153], which shows an example of an inverter with a feedback path without any delays, as well as fan-in and fan-out at the input and output. Inverters are logically reversible, but without a controlled feedback path, the input to the inverter will change immediately since the output changed. This violates the properties of a reversible Turing machine set by Bennett, since the input changed sooner than the time slice required by the inverter to update the signal. This creates a race condition, and since the memory of the reversible Turing machine is updated at every clock cycle, the output changes much faster than the clock updates the memory, resulting in information loss. Reversibility is violated when this occurs. Therefore, it is possible to synthesize a sequential logic structure whose combinational logic block is logically reversible, yet the device is not physically reversible.

There have been a number of papers that have addressed sequential reversible logic devices. Hari presented logical primitives which are acceptable in sequential reversible design

[161]. Mohammed et al presented a genetic-algorithm based approach to synthesizing sequential reversible logic structures [162]. Thapliyal et al. showed that reversible latches and registers may be designed and verified in Verilog [161] such that logical reversibility may be maintained. Many other researchers have expanded upon these works to present registers, flip-flops and memory devices that are logically reversible. Willingham et. al used positive feedback logic to design reversible Toffoli networks [189]. However, none of these papers address the skeptic's claims to their satisfaction. These authors, such as [154][155][156][157][169], state that feedback is not permitted in any reversible system.

The result is that there are two prominent views on this issue being expressed simultaneously in the literature. One side claims that sequential circuits are permissible under any circumstances. The other insists that these structures are not permissible at all. In our previously presented work [163], I disputed the claim that feedback is not permitted in reversible logic structures in any circumstance. Using the delay element presented by Toffoli and Fredkin, and considering the initial state of the feedback loop as a '0' source bit as suggested by Deutsch, it was demonstrated mathematically that a sequential reversible logic circuit may be designed their operations may be completely reversed, allowing for their use in reversible Turing machines. In this chapter, I expand upon this by adapting the model to account for the difference between the worst-case delay in the circuit and the delay in the feedback path in order to ensure that the signal on the feedback-dependent input arrives concurrently with the other inputs to prevent a race condition.

4.2 Definitions

I define here a *sequential reversible logic structure* as any logic device where one or more inputs to the structure are dependent on the logical calculations of one or more outputs and

the remainder of the structure is reversible. A *feedback-dependent input* is an input which receives the feedback wire. A *feedback-producing output* is an output where the feedback wire originates. And a *feedback path* is a wire which originates at a feedback-producing output, and ends at a feedback-dependent input. In order to properly apply the principles of quantum mechanics to these structures, I have to define the relationships between feedback paths, feedback-producing outputs and feedback-dependent inputs. Since a sequential reversible logic structure must have the same number of input and output lines, fan-in and fan-out are not permitted. This means that a sequential reversible logic structure must possess an identical number n of feedback-dependent inputs and feedback-producing outputs. In order for a sequential reversible logic structure to possess more feedback-dependent inputs than feedback-producing outputs, at least one of those outputs must be duplicated, which requires fan out. Similarly, in order for the structure to have more feedback-producing outputs than feedback-dependent inputs, at least one of those inputs must be dependent on two outputs, which requires fan in. Since the quantity of dependent inputs and producing outputs are the same, and fan in and fan out are not permitted, it is also true that each dependent input is dependent on one producing output only.

Since there is one path between each input and output, the number of feedback paths must be equal to n . In the initial clocking cycle, the feedback path acts in an identical fashion to a source bit, since the value of quantum bit will be produced on the output. A *source bit* is a gate which, once every computational step, produces a value of ‘0’ or ‘1’ on its output [149]. Source bits are defined as reversible, since there is a bijection between the produced value at the input of the gate, and the produced output. Therefore, in the initial clock cycle, the feedback path is physically reversible. For each subsequent clock cycle, the feedback path acts as a wire,

producing the value from the feedback-dependent output on the feedback-producing input. Since wires are reversible in nature, the feedback path is reversible for each subsequent clocking cycle, making the feedback path fully reversible in nature.

4.3 Mathematical Model

During the initial clock cycle of any sequential reversible logic device, the loss of one input state per d dependent inputs will result in the loss of 2^d potential input states. In addition, each ancillary input will reduce the total number of possible states by 2^a , since the outputs possible when a is the opposite value will never be attained. This results in a loss of 2^{d+a} potential input states. Since the device is reversible and the source bit producing the values for the ancillary inputs are reversible, the bijection will result in the loss of 2^{p+a} potential output states, where p is the number of feedback-producing outputs. Since the input and output states are bijective, the initial value of each dependent input is uniquely determinable. Therefore, since $d = p$, the total number of possible input states and output states are equivalent, and are equal to the quantity 2^{N-d+a} .

During each subsequent clock cycle, every feedback-producing output may be in a sequential reversible logic device where the feedback will either produce a '0' or a '1', or held at same value for every clock cycle. Therefore, the potential number of states of a feedback-producing output, s_i , is 2 in the first case and 1 in the second case. This means that the number of

lost potential output states for all the feedback-producing outputs is $2^{\sum_{i=1}^p 2^{-s_i}}$. The lost potential states only occur on the feedback-producing outputs, which are then produced on the feedback-dependent inputs for calculation on the next clock cycle. The rest of the device is reversible, which means that the total number of possible input states and output states for the next clock

cycle are equivalent, and are equal to $2^{N-a-\sum_{i=1}^p 2^{-s_i}}$.

This means that, for every clock cycle, the total number of input states and output states may be calculated. In addition, the bijection between input states and output states of the device allow for the determination of the previous input states and subsequent output states. In addition, since ancillary inputs are implemented with reversible source bits, the use of ancillary inputs does not result in entropy gain. Therefore, the input state of each clock cycle of a sequential reversible logic device may be uniquely determined by observing the output values.

I have deduced the past states of sequential reversible logic structures, which means that I have sufficient information to retrace the previous calculations, allowing for use in the final stage of the reversible Turing machine. However, I still need to determine that the device is physically reversible. For the initial clock cycle, the possible number of input states is $w_o = 2^{N-d-a}$, and the total number of possible output states is $w_f = 2^{N-p-a}$. Since the device is bijective, the probability of achieving each input state is $\frac{1}{2^{N-d-a}}$, and the probability of each output state is $\frac{1}{2^{N-p-a}}$. Using the probability relation for change in entropy in a computing system, I get the following:

$$\begin{aligned} \int \frac{dQ}{T} &= k \left(\sum p_{out} * \ln(p_{out}) \right) - k \left(\sum p_{in} * \ln(p_{in}) \right) \\ \int \frac{dQ}{T} &= k \left(2^{N-p-a} * \frac{1}{2^{N-p-a}} * \ln\left(\frac{1}{2^{N-p-a}}\right) \right) - k \left(\left(2^{N-d-a} * \frac{1}{2^{N-d-a}} * \ln\left(\frac{1}{2^{N-d-a}}\right) \right) \right) \\ \int \frac{dQ}{T} &= (d-p)k_B * \ln(2) = 0 \end{aligned}$$

Since I know that the number of feedback-dependent inputs is equal to the number of feedback-producing outputs, the equation evaluates to $\int \frac{dQ}{T} = 0$, making the initial clock cycle physically reversible. The change in entropy for each subsequent clock cycle in a sequential reversible logic structure is determined by setting the number of possible input states to $w_o = 2^{N-a-\sum_{i=1}^d 2^{-s_i}}$ and the number of possible output states to $w_f = 2^{N-a-\sum_{k=1}^p 2^{-s_k}}$. This gives the equation to

determine the statistical probability of each input and output state for every subsequent clock cycle.

$$\int \frac{dQ}{T} = k_B (2^{N-a-\sum_{k=1}^p 2^{-s_k}} - k_B (2^{N-a-\sum_{i=1}^d 2^{-s_i}}))$$

$$\int \frac{dQ}{T} = k_B \ln(2) \left(\sum_{i=1}^d 2^{-s_i} - \sum_{k=1}^p 2^{-s_k} \right)$$

It is known that the number of feedback paths is equal to both d and p , and that each feedback path is reversible. Therefore, the value of s_i for each feedback-dependent input must be equivalent to the s_k value of the corresponding feedback-producing output. This means: $\sum_{i=1}^d 2^{-s_i} = \sum_{k=1}^p 2^{-s_k}$. Therefore, the entropy equation reduces to $\int \frac{dQ}{T} = 0$. This means that sequential reversible logic structures as I defined them are physically reversible in all cases.

4.4 Simulation of Reversible Logic and a Case Study

As a proof of concept, I present a simulation result of a sequential reversible logic structure simulated in the Integrated Qubit library presented in [166][167] with a VHDL library used to simulate the library [168]. A reversible Turing machine should take in a set of inputs, print those results, and then be able to take those results and perform the calculations in reverse. Therefore, let us consider a Fredkin gate with a sequential feedback loop as shown in Fig. 4.1. The circuit's reverse is shown in Fig. 4.2.

This device has 3 inputs and 3 outputs, with one feedback-dependent input and one feedback-producing output. In addition, three inputs are set at '0', so the device has 0 ancillary inputs using no source bits. I define the number of potential input states as 2^{N-d-a} . Since $N = 3$, $d = 1$ and $a = 0$, I obtain $2^{3-1-0} = 4$. Since there are only 4 possible input and output states in the first clock cycle, the device satisfies the conditions set in that claim. Since there is a bijection, when

the output of the device at the end of the first clock cycle is measured, the initial value of the dependent input may be uniquely determined.

Figure 4.1. Sequential Reversible Logic Structure.

Figure 4.2. Reverse Circuit of Figure 4.1.

At the end of the initial clock cycle, the feedback-producing input, may be a '0' or a '1' in either case. Therefore, the value s_i that output is 2. The possible number of input states is determined by substituting $N = 3$, $d = 1$, $a = 0$ and $s_I = 2$ gives , giving 8 possible states for each subsequent clock cycle. Therefore, I know that $\sum_{i=1}^1 2 - s_i = \sum_{k=1}^1 2 - s_k$ for all subsequent clock cycles.

Applying this information, I get the following:

$$\int \frac{dQ}{T} = k_B \ln(2) \left(\sum_{i=1}^d 2 - s_i - \sum_{k=1}^p 2 - s_k \right)$$

Since the entropy gain for every clock cycle is 0, the presented structure device is fully reversible. I will show that the input values may be recreated based on the inputs. Next, let us consider the set of inputs presented in Table 4.1 over 12 clock cycles. The values are the inputs to signals A and B in Fig. 4.2.

At $t=0$, the feedback-dependent input is consider as a signal bit with a value of '0'. Therefore, the results of the first input results in $\langle 0 \ 0 \ 0 \rangle$, and the value of '0' is placed back at the input. This gives the three inputs of $\langle 010 \rangle$ for $t=2$. This gives $\langle 010 \rangle$ for the output. For $t=3$,

the initial input values are $\langle 110 \rangle$, which gives the output of $\langle 101 \rangle$. A value of ‘1’ is placed on the feedback path, giving the initial value of $\langle 101 \rangle$ for the outputs. Table 4.2 shows the output values based in the initial outputs, and the value put on the feedback path f . To show reversibility, I place the value at $t = 12$ on the outputs of the device, and have the run backwards. That is the identical process to placing the that value on the input of Fig. 4.2. Those inputs are shown in Table 4.3.

At $t=0$, the feedback-dependent input is consider as a signal bit with a value of ‘0’. Therefore, the results of the first input results in $\langle 1 \ 1 \ 0 \rangle$, and the value of ‘0’ is placed back at the input. This is what was expected, since Table 4.1 shows the last iteration a $\langle 1 \ 1 \rangle$, and ‘0’ was the value on the input. Completing the reverse calculations, I get the results shown in Table 4.4. The A and B values are shown to be the reverse of Table 4.2. Additionally, the values placed on the feedback path are the exact reverse of the values in Fig. 4.1. Therefore, reversibility was achieved.

To prove our claim, I use the VHDL simulation of Integrated Qubit based Reversible Logic Structures that I presented in [116] to verify the design of these sequential reversible logic structures. The VHDL library was built upon the principles of the Integrated Qubits shown in Section 4. I show, using the VHDL simulations shown in Figures 4.3 and 4.4, that I may accurately produce a bijective sequential reversible logic structure.

Table 4.1. Inputs into Circuit in Fig. 4.1 Over 12 Time Cycles.

	1	2	3	4	5	6	7	8	9	10	11	12
A	0	0	1	1	1	1	1	0	0	0	1	1
B	0	1	1	0	1	0	0	1	0	1	0	1

Table 4.2. Outputs from Circuit in Fig. 4.1 Over 12 Time Cycles.

	1	2	3	4	5	6	7	8	9	10	11	12
P	0	0	1	1	1	1	1	0	0	0	1	1
Q	0	1	0	1	0	1	0	1	0	1	0	0
f	0	0	1	0	1	0	0	0	0	0	0	1

Table 4.3. Inputs into Circuit in Fig. 4.2 Over 12 Time Cycles.

	1	2	3	4	5	6	7	8	9	10	11	12
A	1	1	0	0	0	1	1	1	1	1	0	0
B	0	0	1	0	1	0	0	1	1	0	1	0

Table 4.4. Outputs From Circuit in Fig. 4.2 Over 12 Time Cycles.

	1	2	3	4	5	6	7	8	9	10	11	12
A	1	1	0	0	0	1	1	1	1	1	0	0
B	1	0	1	0	1	0	0	1	1	0	1	0
f	0	0	0	0	0	0	0	1	1	0	0	0

Figure 4.3. VHDL Simulation of Sequential Fredkin Gate.

Figure 4.4. VHDL Simulation of Sequential Reverse Fredkin Gate.

4.5 Proposed 4*4 MLMRG for SRAM Implementation

The proposed MLMRG gate has a quantum cost of 7 and the worst-case delay is 7. The quantum representation of the gate is shown in Fig. 4.4 below. An advantage of the MLMRG gate is that the input and output reversible gate are Controlled-V gates where the select line is the B input and the controlled input is D. Since two Controlled-V gates in series acts as an identity, placing two MLMRG gates in series allows for quantum cost and delay reduction by two.

Fig. 4.5 shows a special case where a MLMRG gate is placed in series with a Peres gate, where the first output from the MLMRG serve as the first input of the Peres gate, and the fourth output from the MLMRG is tied to the third input of the Peres gate. In this particular design, the quantum cost reduction is 3, since the last Controlled-V gate in the MLMRG is cancelled with the first Controlled-V in the Peres, and the Feynman gate in the Peres is reduced, since the desired value of $A \oplus B$ is propagated from the output of the MLMRG. Without this potential for reduction, the MLMRG and Peres design would incur a quantum cost of 11. The reduced design has a quantum cost of 9.

Figure 4.5. Quantum Representation of Proposed MLMRG.

Figure 4.6. MLMRG in Series with Peres and Reduced Representation.

4.6 Proposed SRAM with Reversible Elements

The proposed MLMRG may be configured for implementation as the decision logic. Fig. 4.7 shows this configuration, where the first two inputs are the Read and Write signals, the next two inputs are ancillary inputs where both are held at 0, and a Feynman gate is added to the output to restore the value of Write.

Figure 4.7. Modified MLMRG for Implementation in SRAM Array.

Next, a reversible SRAM cell is designed utilizing the configuration of the modified MLMRG which produces the same output functionality of conventional CMOS 6T SRAM cell [6]. The total quantum cost of the newly proposed SRAM cell is 21 and the worst-case delay of the newly proposed SRAM cell is 19.

Figure 4.8. SRAM Cell with Reversible Elements.

4.7 Proposed 4*4 Reversible Decoder (RD) Gate

In order to implement an array of SRAM cells, a 2-to-4 decoder is necessary for translating which cell corresponds to the address requested by the CPU. A novel reversible 4*4 RD gate which may be implemented as a 2-to-4 decoder is presented. The proposed gate has a quantum cost of 10 and a worst-case delay of 10. The quantum representation of the RD gate is presented in Fig. 4.8, and the input configuration which produces the necessary output logic for the 2-to-4 Decoder is shown in Fig. 4.10 below.

Figure 4.9. Quantum Representation of RD Gate.

Figure 4.10. 2-to-4 Decoder Logical Configuration of RD Gate.

4.8 4x2 SRAM Array with Reversible Elements

The configured RD gate is utilized to translate an input address and select the appropriate SRAM cells to read/write in a 4x2 SRAM array. Its configuration is shown in Fig. 4.10 below.

Next, the single-port SRAM was modified to produce a reversible Dual-Port SRAM cell. The logical configuration of this design is shown in Fig. 4.11 below. The Dual-Port SRAM cell has a quantum cost of 42 and a worst-case delay of 20.

The reversible Dual-Port SRAM was then configured in an implementation of an n-bit synchronous Dual-Port SRAM array. The logical configuration of the proposed array is shown in Fig. 4.12 below. The cost of this design is $42n + \sum_{i=1}^n (i - 1)$ and the delay is $19 + n$.

4.9 DRAM Array with Reversible Elements

Next, a novel reversible DRAM cell was designed and verified. The DRAM cell is based on a unique configuration of the inputs and outputs of a Fredkin gate, and its logical configuration is shown in Fig. 4.11 below. The reversible DRAM has a quantum cost of 8 and a worst-case delay of 8.

Figure 4.11. Reversible 4x2 SRAM Array.

Figure 4.12. Reversible Dual-Port SRAM Cell

The reversible DRAM cell in Fig. 4.13 and control logic utilizing modified Peres gates were used in the implementation of a 4x4 reversible DRAM array. The design also used two RD gates which served as 2-to-4 decoders for the row select and column select signals. The write signal is passed through a Peres control gate (PCG) consisting of two Peres gates. The logical

configuration of the reversible 4x4 DRAM is shown in Fig. 4.14. The quantum cost of the device is 414, and the worst-case delay is 39.

Figure 4.13. Synchronous Dual-Port SRAM Array.

Figure 4.14. Reversible DRAM Logical Configuration.

4.10 Improved SRAM Structure

Our presented SRAM design used two Fredkin gates, which each incur a quantum cost and delay of 5. Since the reversible SRAM device did not use the third output of either Fredkin gate - both were garbage outputs - the design may be reduced in terms of both cost and delay using the proposed RMUX gate, since the cost and delay are only 4. Therefore, the improved design of the proposed SRAM cell is 18 and the worst-case delay of the newly proposed SRAM

cell is 17. The modified reversible SRAM cell is presented in Figure 4.15. The design for the SRAM cell was verified and simulated using VHDL in Xilinx ISE 12.4.

4.11 Conclusion

Sequential computing structures allow for the utilization of subroutine reuse, which permit reduced, simple and efficient designs of computational devices. Reversible logic allows for the design of a universal computing machine where each input state is uniquely determinable by the results printed by the output state at each clocking cycle, and the system reaches every possible state, resulting in no heat dissipation.

In this chapter, I demonstrated that there is a bijection between the feedback-dependent inputs and feedback producing outputs in a sequential reversible logic device. Then, using the entropy equations, I was able to calculate the total number of possible input states and output states for the initial clock cycle, as well as each subsequent clock cycle. As a result, the device was shown to be bijective at each clock cycle. Then, it was demonstrated that the past and future states of each clock cycle of a sequential reversible logic structure may be uniquely determined. Using Boltzmann's equation, I concluded that a sequential reversible logic structure is physically reversible for every clock cycle. Therefore, I support the view that feedback is permitted in the design reversible logic based sequential computing structures.

Two novel 4*4 reversible logic gates were presented, verified and their advantages demonstrated. The MLMRG was implemented as the control for the read/write logic in an SRAM cell. The RD was implemented as a reversible decoder in an SRAM array. A SRAM cell implemented with these reversible elements was presented and verified, and then implemented in the design of a 4x2 reversible SRAM array. A dual-port SRAM cell was designed using these proposed reversible logic structures, and its implementation in a synchronous n-bit reversible

dual-port SRAM array is presented. A DRAM cell designed using reversible elements was presented, verified and implemented in a reversible 4x4 DRAM array.

Then, a previously proposed reversible SRAM cell was modified using the RMUX gate to improve the quantum cost and delay of the device. These new devices were used to implement an 8-bit Min/Max comparator device which stored a value when it was either the minimum or maximum value received over a period of time.

Figure 4.15. Improved Reversible SRAM Cell with RMUX Gate.

CHAPTER 5

INTEGRATED QUBIT BASED REVERSIBLE LOGIC

In this chapter, we present a robust behavioral model for the fundamental Integrated Qubit (IQ) gates towards the design of locally reversible logic structures^{1,2,3}. Modeling IQ gates, as opposed to only Control-V gates or Toffoli gates, allows for a more robust model that more accurately reflects a theoretical reversible computing structure. This method is an extension to existing programming language and modeling method that allows for reversible logic structures to be designed, simulated, and verified. To the best of our knowledge, this is the first work in the behavioral model of integrated qubit gates using a VHDL library presented in [16] that simulates Integrated Qubit operation, we show examples of designed reversible logic structures that violate reversibility. First, we prove that a Buffer Gate (BG) may be achieved in a quantum cost of 1 instead of 2 by effective use of unitary transformations, and then use the BG to show that proper reversible logic operation may be guaranteed. Next, we present an optimization method for reversible logic synthesis based on the Integrated Qubit (IQ) library. This algorithm runs in $O(N)$ time, and reduces the quantum cost of synthesized circuit by up to 45 percent. In addition, the process of replacing the gates in the synthesized circuits with IQ gates uses a locally optimal technique whose major benefits include reduction of cost as well as delay. This method works in conjunction with existing methods to further improve quantum cost and delay of a synthesized reversible logic circuit.

¹ Portions of this chapter were published in IEEE Symposium on Very Large Scale Integration [19]. Permission is included in Appendix A.

² Portions of this chapter were published in IEEE Symposium on Very Large Scale Integration [359]. Permission is included in Appendix A.

5.1 Reversible Design Goals

The three major design goals of reversible logic are as follows. First, minimization of the quantum cost - the number of 1×1 and 2×2 reversible calculations necessary to generate the logical output - will reduce the device's computational complexity. Second, minimization of the delay - the logical depth of the device - will improve the throughput of the device. Third, reduction of the ancillary inputs and garbage outputs - inputs and outputs not implemented in the design of the gate and only serve to maintain reversibility of the device - will improve the design space required to implement the logic.

5.2 Unitary Matrix Representation of Quantum Interaction

We may now introduce the matrix operations necessary to fully realize the locally reversible Turing machine. In [167], DiVincenzo provided a proof that a universal quantum circuit may be completely constructed using only 2×2 reversible gates. This became an important endeavor after Shor presented his algorithms for finding discrete logarithms and factoring integers [276]. Since the laws of physics only permit unitary transformations, deterministic computation is performable on a quantum computer if and only if it is reversible.

The representation of the states in the two-dimensional Hilbert space is found by using a complex projective line, which is a geometrical sphere known as the Bloch sphere, which contains points on the edge of the sphere that correspond to mutually orthogonal state vectors. The points on the sphere represent the system states, and the poles of the sphere represent the “spin up” and “spin down” electron states. It is important to note that the boundary condition on the Hilbert space is that only states on the surface of the sphere are “pure” states, whereas any state not on the surface is a mixed state [277]. Therefore, any reversible computing design is constrained to operations that achieve states on the surface of the Bloch sphere, or else there will

be information loss. The state representation of the spin of the electron on a Bloch sphere is given by $|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2)|1\rangle$. This may further be represented with rotation due to x and y as well as the negation and identity matrices already presented.

The NOT matrix, which may be applied to a one-qubit signal to invert the result, such that $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_0 \end{bmatrix}$. Using the unitary operators first presented in [10] by Deutsch, we may accurately represent a NOT operator in the Hilbert space using two qubit operators in series to represent their half spins, where $\varphi = \pi/2$. Therefore, the NOT matrix is properly represented by its square root matrix, $\frac{1+i}{2} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$.

$$R_y(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & \sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} R_z(\varphi) = \begin{pmatrix} e^{i\varphi} & 0 \\ 0 & e^{-i\varphi} \end{pmatrix}$$

Figure 5.1. Graphical Qubit Representation of the Bloch Sphere.

This operator is known as a *V gate* – or Square Root of Not - and performing two V gates in series will result in the NOT matrix $\frac{1+i}{2} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} * \frac{1+i}{2} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Similarly, if we obtain the Hermitian conjugate of the *V gate*, we obtain the *V+ gate*, $\frac{1-i}{2} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}$. Just like the V gate, performing two V+ operations will produce the NOT matrix. Since the V and V+ gates are Hermitian conjugates, the result is that they will produce the unity matrix will performed in series. The quantum representation of these gates is shown in Fig. 5.2.

This allows for the consideration of 2x2 Control-V and Control-V+ gates for use in reversible quantum computation. When the control signal is zero, the input value is matched on the output. When the control signal is ‘1’, then the unitary V operation is performed, just like with the V and $V+$ gates previously shown. When the resulting matrix for the Control-V gate is multiplied by itself the result is that it produces the CNOT matrix. Therefore, two Control-V or Control-V+ gates in series is equivalent to a Feynman gate. Just like the V and $V+$ gates, when the $V/V+$ operation occurs, we obtain the unity matrix. Therefore, two-bit quantum gates have been demonstrated to be sufficient to synthesize any unitary operation in any size Hilbert space [277].

Figure 5.2. Quantum Representation of the Control V and Control V+ Gates.

5.3 Previous Work in Control-V Simulation

The unitary matrix approach using Control-V and Control-V+ algorithms was implemented in [20] using a genetic algorithm based approach [278][279], which were limited to a 4*4 implementation due to high computational complexity. They defined the set of intermediate values for the control-V and control-V matrix operations, and then use the mutation, crossover and selection operators of genetic algorithms in order to derive the circuit. This was meshed in [48] with the transformation based method in [33] to produce an more efficient implementation. In [261], the control-V gates to present a set of transformation rules in order to minimize the synthesized circuit using K-maps. The K-maps are used to extract the circuit to determine which nodes may be deleted to preserve the output calculations and reversibility.

5.4 Integrated Qubit Gates

The next stage in reversible logic design is developing a library of two-bit quantum gates that allow for minimized construction of locally reversible logic structures. However, these operations may be combined into integrated Qubit gates, since their unitary evolution is a correct representation of space-time quantum mechanical operation [262], since a set of quantum acyclic gates can simulate quantum Turing machines in this manner [152]. This gate is implemented with a Feynman gate with either a Controlled-V or Controlled V+ gate. The XOR output of the Feynman gate is used as the control signal for the Controlled-V or V+ gate it is coupled with. The quantum cost of the integrated qubit gate is 1 and its worst-case delay is 1. The quantum configurations of these gates are shown below in Fig. 5.3. Fig. 5.4 shows that the Fredkin Gate [8] may be realized as a cost of 5 using IQ gates [262].

5.5 Presented Integrated Qubit Behavioral Model

We created a IEEE Standard 1076-1993 [278] package (often referred to as VHDL-93) to define, describe and efficiently model the reversible transformations that could be used within a reversible logic structure using integrated qubit gates. This library, called QUBIT, is a robust multiple-valued logic system for define the given deterministic quantum states and generalizing the non-deterministic states, which allow for the logic system to generalize a characterization of a Qubit that enters into a state other than the known deterministic logic values that exist in the system.

Table 5.1 shows the behavioral modeling system used to represent the spins of the qubits on the Bloch sphere when the Control-V receives binary inputs. Table 5.2 shows the expected results when the gate receives inputs A_0 and A_1 . The differentiation between the control line and the second input allows for real-time control checks on each gate that ensures proper logic is

being simulated, since the control lines are the qubit basis states. Passing a control line anything other than a '0' or '1' will cause simulation of the reversible logic structure to fail. Therefore, this differentiation, and introduction of variable 'U' allows for a robust system, as well as error checking. Additionally, any structure designed that has a 'v', 'V', 'p' or 'P' on the output has not been properly designed, since the final design should produce a binary output to match the binary input.

Figure 5.3. Quantum Representation of Integrated Qubit Gates.

Figure 5.4. Realization of Fredkin Gate with IQ Gates.

Fig. 5.5 shows the syntax of the LIQP gate procedure in the presented QUBIT library. The gate uses the lookup table for both the Control-V and the QXOR, which has the input/output lookup for the Feynman (CNOT) gate. As previous mentioned, this is a proper quantum mechanical representation of one two-qubit operation. Fig. 5.6 shows the resulting VHDL simulation of the LIQP gate.

Table 5.1. IQ Spin Operation Lookup Table.

Qubit State	State Description
0	Logical '0' is produced
1	Logical '1' is Produced
v	'0' is input, V Transformation Applied
V	'1' is input, V Transformation Applied
p	'0' is input, $V^\dagger$ Transformation Applied
P	'1' is input, $V^\dagger$ Transformation Applied
U	The Qubit is in an Unknown State

Table 5.2. V and $V^\dagger$ Behavioral Modeling System.

$A_0(\text{Control})$	A_1	$Q_1(V \text{ Gate})$	$Q_1(V^\dagger \text{ GATE})$
0	A_1	A_1	A_1
1	0	v	p
1	1	V	P
1	v	1	0
1	V	0	1
1	p	0	1
1	P	1	0

```

PROCEDURE LIQP ( SIGNAL Ai : IN QUBIT;
                  SIGNAL Bi : IN QUBIT;
                  SIGNAL Po : OUT QUBIT;
                  SIGNAL Qo : OUT QUBIT ) IS
  VARIABLE DELAY : TIME := 1 ns;
  VARIABLE Qa : QUBIT := 'U';
  VARIABLE Qb : QUBIT := 'U';
  BEGIN
    Qa := qxor_table(Ai,Bi);
    Qb := cvp_table(Ai,Qa);
    Po <= Ai AFTER DELAY;
    Qo <= Qb AFTER DELAY;
  END PROCEDURE;

```

Figure 5.5. Presented VHDL-93 QUBIT Syntax.

Figure 5.6. VHDL Simulation Results for LIQP Gate.

5.6 Reversible Logic Design Rules for VHDL

In order to successfully model the quantum interactions as well as develop a system that performs the desired reversible computation, the rules for reversible logic must be adapted into the behavioral model. In this section, we present a set of design rules and show how they were implemented.

First, we define each IQ gate as having two input signals and two output signals. This allows for the creation of intermediate signal values that are tied to the input/output signals of a fundamental gate. This permits easier debugging, and to maintain reversibility through the designed circuit.

This rule is realized in the syntax by declaring x signals for a line, where x is the value returned from the equation below. This works since the number of gates interacting with a particular line is equal to one then only a single new signal must be generated to interact with

$$\text{intermediate values, } f(x) = \begin{cases} \sum Gates_{line}, & Gates_{line} > 1 \\ 1, & Gates_{line} = 1 \end{cases}$$

Fig. 5.7 shows an example of how these signals are maintained in a designed reversible logic structure. The 3*3 Toffoli gate incurs a cost of 5, since it requires two Control-V, one Control-V+ and two CNOT gates. Line A encounters 3 gates, so it requires 3 intermediate signals, satisfying our equations. Similarly, line B had 4 intermediate signals and line C has 3 intermediate signals. The total number of intermediate signals is 10, which means that each 2*2

gate produces 2 intermediate signals, satisfying our constraint.

Figure 5.7. Signal Generation Example for Toffoli Gate.

The second rule we implement is that *control lines are Qubit Basis States*. In order for the reversible logic structure to properly operate, any control line must receive a ‘0’ or a ‘1’. If any other value in the lookup table is reached, our model throws an error, since the design rule has been violated. Such an error would typically occur from the improper use of multiple reversible gate transformations; creating a reversible logic structure consisting of a single RIQ or RIQP gate is an example of what would cause a Qubit to enter a state other than one from the presented multiple-valued logic system. Figs. 5.8 and 5.9 show an example of a CNOT gate’s design and simulation.

Figure 5.8. Signal Generation for CNOT Gate.

Figure 5.9. Simulation Results for CNOT Gate.

Third, we state that *unknown values exist*. This is due to the fact that a control line in an CNOT gate or IQ gate may not receive a ‘V’, ‘v’, ‘P’, or ‘p’ and still properly function. This rule

emphasizes good design habits and that Unknown values exist, and can be a resulting output value from a logic structure. The VHDL package allows for the ability of catching design errors, in the sense that there are strict rules for the control line variables of controlled reversible gates. Overall the gate structure and orderings are relaxed, but Unknown values still have the ability to appear in simulation. An example of logic that would produce an Unknown value is passing any variable other than a '0' or '1' in the Qubit Modeling System to a QXOR gate.

5.7 Simulation of Fundamental Reversible Logic Structures

Using this package an extensive library of fundamental reversible logic gates and reversible logic structures was designed and verified. In this section, we use our model to simulate the operations of two fundamental reversible logic structures. First, we simulate a Swap gate using two integrated qubit gates. This design, presented in [182], uses the IQ design paradigm to reduce the quantum cost from 3 to 2. Our behavioral model is advantageous in that it accurately simulates the quantum computation of a SWAP operation, with a smaller quantum cost and delay than any other behavioral model. Fig. 5.10 shows the IQ configuration, and Fig. 5.11 shows our simulation results.

Figure 5.10. SWAP Gate Proposed in [182] (left) and Previous (right).

Figure 5.11. Simulation Results for IQ-Based SWAP Gate.

Next, we simulate a Fredkin gate as shown in Fig. 5.4. This design was produced using only the CNOT gates, Control-V structure and IQ gates as proposed in [7]. Without the IQ gates, the minimum quantum cost and delay of the Fredkin gate would be seven. By using our behavioral modeling system, we are able to simulate the minimal quantum operations needed to perform the Fredkin's logical calculations. The resulting waveform is shown in Fig. 5.12. By getting the correct output waveforms without specifying the output functions of the Fredkin gate, the robustness of our behavioral model is demonstrated.

Additionally, Figs. 5.13 through Fig. 5.17 show simulation results for some fundamental reversible logic structures Toffoli [3], Measurement [10], Peres [261], RMUX1 [182] and UPG [182] that were simulated or designed using this method.

Figure 5.12. Simulation Results for Fredkin Gate.

Figure 5.13. Simulation Results for Toffoli Gate.

Figure 5.14. Simulation Results for Peres Gate.

Figure 5.15. Simulations Results for RMUX1 Gate.

Figure 5.16. Simulations Results for UPG.

5.8 VHDL QUBIT Implementation and Advantages

Implementing our behavioral model in VHDL allows us several advantages than if we created our tool in another behavioral language, such as Aida. First, VHDL allows for operator overloading, and the redefining of a function, procedure, and operator. This work uses a variety of functions and procedures for simulation and modeling each with purposes that are task specific. Simulation log file generation and file writing require that a string is used for being written to a file. Thus, function calls to convert a Qubit type to string to be written to a file

during test mode were used while procedures were used to write to the simulation log as nothing need be returned. Procedures are also beneficial because a procedure allows the use of signals, local variables, and the use of delay statements to model delay. This allows for quantum simulation not just for logical calculations, but to determine the delay of the propagation of the signal through the designed reversible logic structure.

Third, the user may use the VHDL Qubit library to create reversible logic structure layouts that may be used in the design of larger reversible logic structures. For this tool, we used Xilinx ISE 13.2 to create schematic symbols of gates that can be used in a visual design layout. With this schematic layout VHDL code can be automatically generated by Xilinx and require minimal changes to the code to become functional. Fig. 5.17 shows a Xilinx schematic symbol of a Fredkin gate composed of the Integrated Qubits in our QUBIT library.

Figure 5.17. Xilinx ISE 13.2 Schematic Layout FRE Symbol.

VHDL is also advantageous in that it permits the automatic generation of testbenches for robust testing. Using Xilinx 12.4 and 13.2 there was the ability to generate testbench files, although these files needed modification after every time a new testbench file was generated. The solution was a graphical testbench tool written in the C# programming language. The tool is designed to read the current working directory for *.vhd files, and group testbench and VHDL files in a tree view. Testing results showed that the program was able to properly generate a testbench that generates 2^n test inputs to formally verify the reversible unit under test. These

testbenches were also capable of being used in Xilinx 12.4 with ISIM M.81d Simulator and follow proper VHDL syntax.

5.9 Verification

Issues arise in simulation of reversible logic structures when the clocking zones are not properly constructed. Let us consider the following inputs into the Peres gate in Fig. 5.19. The Peres gate is a very commonly used reversible logic structure, since it is able to achieve the AND/NAND operations in fewer quantum operations than the Toffoli gate. The logic outputs of the Peres gate are $P = A$, $Q = A \oplus B$, and $R = AB \oplus C$. Many of these designs, such as the ripple-carry adder in [267] and the multiplier in [84] – to give two of *many* examples – use a block diagram design to implement the Peres gate. This means they assume that all of the outputs emerge simultaneously after four clock cycles. However, they do not account for this assumption with the appropriate delay considerations. For example, in the Peres gate, clocking zone one performs a Control-V operation on the B and C inputs, and the A input is left unchanged. Without consideration of the wire, as required in [3], the A signal is immediately propagated to the second clocking zone without delay.

Figure 5.18. Quantum Representation of Peres Gate in Forward Operation.

Figure 5.19. Quantum Representation of Peres Gate in Reverse Operation.

Fig. 5.20 shows the simulation of a Peres gate in forward operation in the VHDL package presented in [16], and Fig. 5.21 shows the Peres gate in reverse operation. All values of the input and output gates are assumed to be zero at start time, as shown in [30] by Deutsch. In this simulation, the input signals are changed every 10ns, and the delay is assumed to be 1ns. The simulation does show that, during most of the time slices, the circuit performs as expected. However, there are three time slices where this is not the case. In both simulations, this is shown where the output line is in red as opposed to the expected green.

Figure 5.20. Peres Gate Forward Operation Without Clocking Zone Regulation.

Figure 5.21. Peres Gate Reverse Operation Without Clocking Zone Regulation.

The reason this error occurs is due to propagation errors within the circuit. Consider the first error on the R output. This occurs two time cycles after the input vector changes values from $\langle 0, 0, 1 \rangle$ to $\langle 0, 1, 0 \rangle$. The reason is that the CNOT gate in time slice 3 is expecting the combination $\langle 0, 0 \rangle$ to hold for an additional time slice. However, the value on line B from the Control-V propagates past clocking zone 2 into clocking zone 3. Therefore, instead of getting $\langle 0, 0 \rangle$ as the expected input, it received $\langle 0, 1 \rangle$. This means that, instead of passing a 0 to the control line of the Control-V+ in clocking zone 4, it passes a '1' to the gate. The gate performs a Control-V+ operation on the input signal instead of passing the signal through. The result is that the output value does not hold a '0' or '1' on the Bloch sphere, resulting in lost information.

This becomes a significant problem in a reversible Turing machine. A reversible Turing machine is a 3-tape Turing machine R such that there exists input and output strings I and P that contain no embedded blanks, and S only halts on the input string if R halts on the input string and the inverse controller B of the other two tapes. Additionally, machine S will product output string P given input strings I if and only if machine R produces I , B and P . Therefore, if S is a universal Turing machine, then R becomes a universal Reversible Turing machine. This is accomplished by using a 'Copy' operation, which allows for preservation of the output data, which still permitting the initial tape to be erased. The data on the output history tape is dependent on the logical calculations made by the computing structure. This allows for erasing the output values to reach the initial state.

Let us assume that a reversible Turing machine has an output whose output values are determined by a Peres gate without proper regulation of clocking zones, as shown in Fig. 5.20. When the propagation error occurs, the value is not a proper value on the Bloch sphere. This means that the value on the output at that time is not able to be placed on the 'Copy' output,

since its value is not determinable. Therefore, during that time frame, the input string I is not determinable, because I is only determinable if and only if machine R produces I , B and P . This means that the resulting information loss results in both entropy gain and improper operation of the reversible Turing machine.

5.10 Proposed Integrated Buffer Gate as a Solution

In the next section, we will demonstrate how using buffers in the clocking zones on the paths where quantum interactions are not occurring prevents the race conditions that violate reversibility. Since we are providing the proof of concept for this proof in the integrated qubit implementation, we need to consider buffers in this technology. However, most reversible logic synthesis research focuses on reducing the quantum cost and delay of a circuit, and inserting buffers increases both of those metrics. Feynman presented a 2×2 buffer consisting of two CNOT gates [3] which restored the controlled signal back to the original value. A 1×1 buffer may also be designed using two inverters. Both of them incur a cost and delay of two. Not having a 1×1 buffer that has a cost and delay of one becomes problematic, since each clocking zone has a time slice equivalent to 1. This means that, without a 1×1 buffer with a delay of one, complete reversible operations in the integrated qubit library is not possible. Therefore, we present a 1×1 buffer to be used in the IQ library initially presented by DiVincenzo. The 1×1 buffer is logically equivalent to a V and a V^+ gate placed in series on the same wire.

LEMMA 5.1. A 1×1 buffer implemented in the Integrated Qubit library has a cost and delay of one.

PROOF. Consider the implementation of the NOT gate in the Integrated Qubit library. The identical implementation may be achieved by placing two V or V^+ gates in series. Two V^+ gates in series gives the following unitary operation:

$$\frac{1}{i+1} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} * \frac{1}{i+1} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} = \frac{1}{2i} \begin{pmatrix} 1+i^2 & i+i \\ i+i & 1+i^2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

This is the NOT operation. This means an operation of a V and V or a V+ and V+ together in series constitutes a cost of one. This is because these operations represent an entire spin, achieving the ‘0’ or ‘1’ state. When a V and a V+ are placed in series, they give the following unitary operation:

$$\frac{1+i}{2} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} * \frac{1}{i+1} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1-i^2 & i-i \\ i-i & 1-i^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

This is the identity matrix. The Buffer gate represents no spin, but still achieves the ‘0’ or ‘1’ state just like the NOT gate. And the V and V+ in series is cost-equivalent to placing a V and V or V and V+ in series, which incurs a cost and delay of one. Therefore, the 1x1 IQ reversible buffer also incurs a cost and delay of one.

In Fig. 5.22, we show an example of how two V gates are placed in series as an Integrated Qubit gate to accomplish the configuration for the NOT gate. We show the same in Fig. 5.23 for the new 1x1 Buffer gate with a cost of 1. In Fig. 5.24, we show the implementation of the forward and reverse operations of the Peres gate, but with proper regulation of clocking zones using Buffer gates. Since the buffer gate is used only in situations where the clocking zone is empty, the result is that the worst-case delay of the circuit is not impacted at all. The difference is that the delay of the other circuits has been modified to ensure proper operation.

Figure 5.22. Quantum Representation of NOT Gate.

Figure 5.23. Quantum Representation of Buffer Gate.

Figure 5.24. Quantum Representation of Buffered Forward Peres Gate.

Figure 5.25. Quantum Representation of Buffered Reverse Peres Gate.

To justify our assertion, we used the VHDL simulation again to show the difference between the buffered and un-buffered Peres gates in forward and reverse operation. As shown in Fig. 5.25, the operation of the forward Peres gate no longer incurs errors when the input vectors are $\langle 0,1,0 \rangle$, $\langle 1, 0, 0 \rangle$, and $\langle 1,1,1 \rangle$, as they did in the simulation shown in Fig. 5.20. Similarly, the operation of the forward Peres gate no longer incurs errors when the input vectors are $\langle 0,1,0 \rangle$, $\langle 1, 0, 0 \rangle$, and $\langle 1,1,1 \rangle$, as they did in the simulation shown in Fig. 5.21.

As a result, by using this buffering method, the presented designs in [31] and [32] - as well as *many* other reversible logic structures that use the Integrated Qubit library - are now logically correct. Additionally, since the authors assumed that the signals emerge from each gate at the time of the worst-case delay, all of their delay calculations are now correct. The only difference is that the designs incur a high quantum cost than previously calculated.

We would also like to note that DiVincenzo demonstrated previously that all 3x3 reversible logic structures may be achieved with six or fewer 1x1 or 2x2 integrated qubit gates. Therefore, such a 3x3 reversible logic structure has m 1x1 gates and n 2x2 gates, such that $m + n \leq 6$. This means that the structure has $m + n$ clocking zone. In a clocking zone with a 2x2 gate, only one buffer is required to properly mitigate the clocking zone, since there is only one wire that is not being operated only via quantum interaction. In a clocking zone with a 1x1 gate, two buffers are required. Therefore, in any 3x3 reversible logic structure, no more than $m + 2n$ buffers are required to ensure proper circuit operation.

Figure 5.26. VHDL Simulation of Buffered Forward Peres Gate.

Figure 5.27. VHDL Simulation of Buffered Reverse Peres Gate.

5.11 Proposed Optimization Algorithm

In the first stage of the algorithm, we switch the gates in the synthesized circuit with their equivalent in the library presented. Each Toffoli, Fredkin, Peres and Swap are replaced with the circuits from Table 5.1. In addition, we check two gates that are adjacent to one another to check

if there is a smaller implementation that may achieve the same logical calculation with a lower cost. The primary benefit from this technique is that, by replacing two or three adjacent gates with a smaller implementation, reduced quantum cost and delay are achieved. Consider the circuit in Fig. 5.28a. It consists of 4 Feynman gates and a Toffoli gate. The Toffoli gate is converted to the equivalent CV+ library equivalent.

Another reduction rule that is implemented at this stage is when two consecutive Toffoli gates have the same control signals. The algorithm would normally replace these gates with two 5-qubit Toffoli gates. However, that would incur a cost that is 2 more than necessary. Instead, we replace them with the circuit shown in Fig 5.28b. This automatically reduces the cost from 10 to 8, and achieves the identical logical calculation. It is important to note here that, if there is a third Toffoli gate in the sequence that meets those criteria, the algorithm must add the 5-qubit gate, since the Q value has been restored to the original value.

The third reduction rule that is implemented at this stage is when two consecutive Toffoli gates have the one control line in common, and the dependent line in common, as seen in Fig. 5.28a. The algorithm would normally replace these gates with two 5-qubit Toffoli gates. However, that would incur a cost that is 1 more than necessary. Instead, we replace them with the circuit shown in Fig. 5.28c. This automatically reduces the cost from 10 to 9, and achieves the identical logical calculation. Again, if there is a third Toffoli gate in the sequence that meets those criteria, the algorithm must add the 5-qubit gate, since the Q value has been restored to the original value.

The fourth reduction rule that is implemented at this stage is when a Toffoli is followed by a Feynman gate have the one control line in common, and the dependent line in common, as seen in Fig. 5.28d. The algorithm would normally replace these gates with a 5-qubit Toffoli gate and a

1-qubit Feynman. However, that would incur a cost that is 1 more than necessary. Instead, we replace them with the circuit shown in Fig. 5.28d. This automatically reduces the cost from 6 to 5, and achieves the identical logical calculation.

The resulting graph is broken up into nodes in the following manner: Given a reversible circuit with n inputs and n outputs with a cost of c , the resulting graph will have $2n+c-1$ nodes and $n+2c$ edges. Nodes 0 through $n-1$ represent the inputs. Nodes n through $n+c-1$ represent each quantum gate, in order of their delay. Nodes $n+c$ through $2n+c-1$ represent the outputs of the device. The edges have a direction, and since the algorithm is going to work from the outputs to the inputs, the direction goes from the outputs to the inputs.

Figure 5.28. Replacement Rules for Integrated Qubit Optimization.

Each node consists of the following: an integer pertaining to the `gateType` (for example, a `gateType = 0` is an input, `gateType = 1` is an output, `gateType = 2` is a Feynman (CNOT) gate, `gateType = 3` is a control-V, `gateType=4` is a control-V+, `gateType = 5` is a swap, and `gateType = 6` is an inverter), two Booleans `Control1` and `Control2` determining whether each input is a control line or a dependent line (false = dependent, true = control), two integers indicating where the next node is (`nextNode1` indicates where the top line goes, and `nextNode2` indicates where the bottom line goes), `nextType1` and `nextType2` – indication whether the next gate is control top (1), control bottom (2), dependent top (3) and dependent bottom (4) – and a boolean *include* indicating whether the gate is to be included in the final circuit.

Using the `nextNode1` and `nextNode2` values, as well as the `Control1` and `Control2` values, we can translate the circuit present in Fig. 5.28b into a sequencing graph, as shown in Fig. 5.29. The red edges represent when the node receives a control signal, and the blue edges represent when the node receives a dependent signal. The resulting graph is a Directed Acyclic Graph where there are n nodes with no incoming edges and n nodes with no outgoing edges.

The adjacency list consists of a pair of integers for each node. Each value of the list will be an integer from 0 to 6. The value 0 indicates there is no edge. If the top line is the control line, then the value will be 1. If the bottom line is the control line, then the value will be 2. If the top line is the dependent line, then the value will be 3, and if the bottom line is the dependent line, then the value will be 4. For example, consider node 4. The top line is the control line, and the edge comes from node 6, therefore the value of (4,6) will be 1. Also, the dependent line comes from node 5, therefore the value of node (4,5) in the adjacency list will be 4. Sometimes, both the control line and the dependent line both come from the same node. For example, consider node 11: the top line is dependent, since the value of the XOR depends on the control line, therefore,

the value for (11,12) is 3. In addition, the control line, which is the bottom line, also comes from 12. Therefore, the value will also be 2. Therefore, the node value of (11,12) is (3,2).

There are some important properties to note for the nodes. Nodes 0 through $n-1$ will only consist of one 5 or one 6. Since each input corresponds to each output, we can guarantee that a reversible graph will have n values of 5 or 6 in the adjacency list. Consider Fig. 5.29: (0,10) is 5, (1,4) is 5, (2,4) is 6 and (3,5) is 6. Since there are 4 inputs, this portion of the matrix is constructed correctly. Nodes $n+c$ through $2n+c-1$ will only consist of one value of 1 through 4, since they will either come from a control value of a dependent value of a node. On the outputs, (9,14) is 1, (9,16) is 4, (12,13) is 1 and (12,15) is 4. Since there are 4 outputs, this portion of the matrix is constructed correctly.

The nodes themselves have properties which reflect the nature of a reversible circuit. Each input node has one incoming edge and every output node has one outgoing edge, both of which are reversible in nature [4]. Every inverter edge has one input and one output edge, which is also reversible in nature [6]. Every other node has two input edges and two output edges, which is also reversible. Therefore, the total number of nodes has a linear relationship with the quantum cost and the number of inputs, such that $n_{total} = C - n_{swap} + 2N$, which allows for linear time implementation.

Next, using a set of rules, we traverse the nodes encountered by each process, and based on the type of node, we determine whether the node should be deleted. If the node is deleted, the two input wires are connected with the output wires. If the node is a CNOT, we check to see if the next gate is another CNOT or a Control-V/V+ gate. If both of the input lines to a CNOT are identical to the output lines of the Control-V gate, then they may be merged into a IQ gate. If a CNOT gate has the same control line as a CNOT gate, and the input lines are the same as the

previous CNOT's output lines, then they produce an identity, and both may be eliminated from the circuit. There is a similar method for Control-V/V+ structures, which is based off our knowledge that two Control-V in series produce a CNOT (reducing the cost from 2 to 1) and a Control-V and Control-V+ in series with the same control line and input/output lines produce an identity (reducing the cost from 2 to 0). After the circuit is traversed, we use two rules to determine if an edge or node should be removed. First, if neither input edge is marked, then the corresponding gate is not required. Second, if a node has one marked input line which is a control line, and the corresponding line is also a control line, then the gate is not necessary, and set the nextGate values of the previous node to the next gate values of the next node. Consider the circuit in Fig. 5.29. After the set of rules used in checkForMerge is run on the circuit, it has been reduced from a quantum cost of 10 to a cost of 8.

Figure 5.29. Graph and Circuit after CheckForMerge and CreateGraph.

Next, we optimize the circuit by performing a Reversible_DFS search from the desired outputs. Consider an implementation of the circuit in Fig. 29b where the user desires outputs P and R . Therefore, Q and S are garbage outputs. Therefore, we will perform a Depth-First Search from the desired outputs. The graph is represented as two adjacency matrices, where each matrix

$A0$ represents the control edges, and matrix $A1$ represents the dependent matrix. The advantage of adjacency list representation is that depth-first search may be performed in $O(V+E)$ time. In addition, by implementing parallelism and appropriate critical section protection – where each node is a critical section – the algorithm may perform this search from each desired output concurrently in $O(V+E)$ time. We determine that the total number of edges is directly proportional to the quantum cost. Therefore, the runtime of the Reversible DFS is $O(C)$. After running *Reversible_DFS*, we use a set of rules to check each node to determine whether or not it should be deleted. Using *deleteNode*, if a node is to be deleted, we merge the input lines and output lines of the node, and drop the node, since it has been proven to only contribute logically to a garbage output.

The user inputs a reversible circuit, $R(N,C)$ where N is the number of inputs/outputs, and C is the cost of the circuit. The algorithm converts the given circuit to the Integrated Qubit Library and generates the new circuit. Creation of the adjacency list takes $O(C)$ time. Next, the algorithm asks the user for the desired logical outputs and stores them in U . For each node in U , the algorithm generates a concurrent thread and then runs *Reversible_DFS* on each thread. This allows this step to be run in $O(C)$ time. Table 5.3 shows the cost and improvement.

The proposed algorithm was written in C. The program generated a circuit in VHDL in the proposed Control-V/V+/IQ library based on the circuits presented in [11][33]. In this section, we will first show a setp-by-step implementation of our algorithm against a previously presented synthesized circuit, and then compare our results with a through set of previously existing synthesized circuits from previous algorithms. Consider the reversible circuit derived from the positive Davio tree in [29]. The circuit consists of 5 Toffoli gates and 2 Feynman gates.

ALGORITHM 5.1. IQSynthesis

```
Algorithm(Circuit R(N,C))
  Convert R to R', where R' is in IQ Library
  Generate Adjacency List  $L$  from  $R'$ 
  For each node  $i \in L$ 
     $checkForMerge(L)$ 
  Get user-indicated logical inputs  $U$ 
  Generate a parallel thread for each node  $i \in U$ 
  Concurrently run  $Reversible\_DFS(L, U[i])$  for each thread  $i$ .
  For each node  $i \in L$ 
    if( $checkForDelete(i)$ ) // If true, delete the node
      deleteNode(i.prevNode1, i.prevNode2, i, i.nextNode1, nextNode2)
  Generate new circuit  $R$  from  $L$  and return  $R$ 
```

Table 5.3. Improvement in Delay and Cost from Fig. 5.28 Based on Algorithm.

User Select	Result Cost	Improvement (%)	Ideal Cost
P	5	50	5
Q	0	100	0
R	0	100	0
S	6	40	0
P, Q	5	50	5
P, R	5	50	5
P, S	6	40	0
Q, R	0	100	0
Q, S	6	40	0
P,Q,R	5	50	5
P,Q,S	6	40	0
P,R,S	6	40	6
Q,R,S	6	40	6
P,Q,R,S	6	40	6

5.12 Algorithm Verification and Comparison to Previous Results

The equivalent circuit in this library incurs an initial cost of 27, as shown in Fig. 5.29. The authors designated outputs U and V as garbage outputs. For the implementation of the proposed algorithm, we designate inputs P , Q , R , S and T as the desired logical outputs. After running the algorithm, the circuit in Fig. 5.29 was reduced from a cost of 27 to a cost of 24, which is an improvement of 11.11%. The results of each step of the algorithm are shown in Figs. 5.30-5.32.

The algorithm was implemented on the final circuits produced by many of the algorithms presented in Section 5.11. The results of those implementations, including the paper presented, the initial cost in the IQ library, the final cost after implementing the algorithm, and the percent improvement in cost are shown in Table 5.4. Our results show that we earn significant reduction in quantum cost over previously synthesized circuits.

5.13 Conclusions

Simulation results show that our VHDL quantum behavioral model based on Integrated Qubit gates is a more robust and efficient method for the design, simulation, and verification of reversible logic structures than previously existing models. It was also shown the use of VHDL in this method allows for the use of a natively robust programming language that allows for concurrency from the built in capabilities of this language. This method offers a new means of design and verification of reversible structures. The ability for reuse of structures allows for ease of design for larger complex structures, while offering assistance in the verification and test phases by asserting the designer of invalid or poor design practices. Additionally, rather than the design of these structures purely at the quantum representation this method provides new means for designing at higher levels of abstractions, in that, the designer can now use higher level block diagrams.

We presented in this paper a method of optimizing synthesized reversible logic structures based upon converting a previously synthesized circuit into the Integrated Qubit library, and then allowing the user to designate the desired logical outputs. Our presented method of replacing gates with their IQ implementation allowed for reduction in delay by replacing two adjacent gates with a smaller implementation that achieves that same logical calculation. This allows us to merge quantum bits in order to reduce cost and delete bits that only contribute to the calculation

of garbage outputs. The synthesized circuits are still logically reversible in nature, but are reduced in terms of quantum cost. By implementing parallelism in our design, we successfully implemented this algorithm $O(N)$ time. We demonstrated that the quantum cost of the circuits may be significantly reduced.

Figure 5.30. Reversible Circuit Presented in NCT and IQ.

Figure 5.31. Reversible Circuit after CheckForMerge.

Table 5.4. Improvement upon Synthesized Circuits.

Circuit	User Outputs	Initial IQ Cost	Cost After Algorithm	Improvement (%)
Example 1 [20]	P,Q	16	13	18.75%
Example 1 [20]	P,Q,R	16	14	12.5%
Example 2 [20]	P,Q	7	6	14.28%
Example 2 [20]	P,Q,R	7	7	0%
Example 3 [20]	P,Q,R	15	13	13.33%
Example 3 [20]	P,Q	15	12	20%
Example 4 [20]	P,Q,R	17	13	23.5%
Example 6 [20]	P,Q,R	7	5	28.57%
Example 6 [20]	Q,R	7	4	42.86%
Fig. 8 (Full Adder) [20]	R, S	11	6	45.45%
Fig 1a [21]	P,Q,R	10	8	20%
Fig 2 [21]	P, Q	3	0	100%
Fig 2 [22]	P,Q,R,S,T, U, V	220	204	7.27%
Fig 2 [22]	P, S, T	220	201	8.63%
Fig 2 [25]	P, R, S	17	13	23.5%
Fig 3 [25]	P, Q, R, S	12	9	25%
Fig 4 [25]	A, B, C	6	5	16.67%
Fig 4 [25]	A, C	6	4	33.33%
Fig 5 [25]	B, C, D, E, G	23	18	21.74%
Fig 5 [27]	A, B, C	7	6	14.28%
Fig 5 [27]	B, C	7	5	28.57%
example1 [27]	A,B,C	7	6	14.28%
example1 [27]	A, B	7	5	28.57%
example2 [27]	A,B,C	5	5	0%
example2 [27]	A,C	5	4	20%
example3 [27]	A,B,C	7	6	14.28%
example3 [27]	A, C	7	5	28.57%
Fig 3c [28]	A, B, C	15	13	13.33%
Fig 3c [28]	A, B	15	11	26.67%
Fig 3c [28]	B	15	10	33.33%
Fig 4 [29]	P,Q,R, S,T	27	24	11.11%
Fig 10 [30]	U	13	12	7.69%
Fig 11 [30]	P,Q,R,S,T,U	15	14	6.67%
Fig 11 [30]	P, R, T, U	15	12	20%
Fig 16a [30]	P,Q,R,S	12	10	16.67%
Fig 16a [30]	P, S	12	9	25%
Fig 16b [30]	P,Q,R,S,T	13	9	30.77%

Figure 5.32. Reduced Sequencing Graph.

Figure 5.33. Final Reduced Circuit.

CHAPTER 6

SYNTHESIS OF DUAL RAIL ADIABATIC LOGIC

6.1 Introduction

Reversible logic is a promising computing design paradigm which presents a method for constructing computers that produce arbitrarily low heat dissipation [12]¹. The basic principle of reversible computing is that a bijective device with an identical number of input and output lines will produce a computing environment where the electrodynamics of the system allow for prediction of all future states based on known past states, and the system reaches every possible state, resulting in no heat dissipation [160][15]. Reversible logic is essential in the future implementations in CMOS [149], quantum computing [16][17], optical computing [330] and DNA computing [113], since these structures are required to break the $kT\ln(2)$ barrier for energy dissipation, which has recently been experimentally demonstrated [114].

Adiabatic logic is an implementation of reversible logic in CMOS where the current flow through the circuit is controlled such that the energy dissipation due to switching and capacitor dissipation is minimized [24]. This is accomplished by recycling circuit energy rather than dissipating it into the surrounding environment. This is beneficial for CMOS implementations, since the input and output charges are kept separate. Adiabatic logic implementations of CMOS have been used to improve power consumption in comparison to pass transistor logic [23]. Adiabatic logic requires the use of ramp functions instead of the faster switching achieved in step

¹ Portions of this chapter have been accepted for publication to the IEEE Transactions on Computer Aided Design [293]. © 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

functions. Dual-rail adiabatic logic implemented with reversible principles has also been demonstrated to significantly reduce differential power consumption [116]. Therefore, dual-rail adiabatic logic shows significant promise as a design methodology in applications where security is the primary design metric, and higher operating frequency is not as desirable. One such application is Smart Card design, where mitigation of Differential Power Analysis attacks is highly desirable, and the operating frequency is 13.56MHz according to the ISO 14443 standard.

In this chapter, we present a novel algorithm for synthesis of adiabatic logic structures in CMOS. We achieve synthesis by comparing the binary representation of the output value and its offset from its location in the unitary matrix. Once the initial circuit is synthesized, the number of transistors required for between each input and output node are minimized using the ESPRESSO heuristic branch-and-bound method [142] for prime implicants. The resulting circuit is synthesized and simulated using HSPICE. In Section 6.2, we review the fundamentals of adiabatic logic and review the previous work in reversible and adiabatic logic synthesis. In Section 6.3, we present the algorithm and implementation of the optimization method, showing step-by-step synthesis of the Toffoli, Peres, Swap, and Inversion gates. Our approach is novel because we directly correlate the horizontal offsets in the permutation matrix with the necessary switches required for synthesis instead of using a library of equivalent functions. We designed the proposed algorithm in C++. The program generates a CMOS circuit in HSPICE that is used for simulation and verification. Multiple SPICE decks were used for simulation, and the 22nm predictive technology model designed by the Nanoscale Integration and Modeling (NIMO) Group (ptm.asu.edu) [138] is used in the simulation in Fig. 6.5. The benchmarks in Table 6.2 are from the RevLib library [89], and their code in C++ (revlib.org) was used to determine the output combinations we used in our algorithm. The circuits are simulated at 13.56MHz in order

to show their benefit in designs where reduction of differential power for security purposes is the most desirable metric, and slower operation is acceptable.

6.2 Motivation and Background

There are two issues that must be addressed in any CMOS adiabatic circuit. First, the implementation must result in an energy-efficient design of the combined power supply and clock generator. Second, reversible logic functions require greater logical overhead in order to meet the bijective requirement [21]. Therefore, the energy dissipated by switching of the circuit must be controlled and recycled instead of dissipated into the environment.

6.2.1 Fundamentals of Reversible and Adiabatic Logic

The reduction of energy dissipation is achieved through the use of a ramp function instead of the faster switching achieved in step functions. Therefore, transistors may be used in adiabatic operation, despite being demonstrated as lossy devices [21], and is achieved through the application of two rules. First, a transistor is always on when there is a significant current flowing through the transistor. Second, when there is a significant difference between the source and drain voltages, the transistor must be off. In [22], adiabatic circuits were shown to produce a reduction in energy dissipation of 60% at 20MHz and 35% less energy at 100 MHz, and reversible dual-rail CMOS pass transistors were demonstrated in [23].

CMOS adiabatic circuits require the ability to be physically bijective, meaning that the output signals may be placed on the outputs and the unique input signals may be reproduced on the input wires. A dual rail approach was used to accomplish this goal in [26][331], in which the three fundamental reversible logic structures were design and fabricated in 0.35 μ m technology, where $V_{tp} = 0.6V$ and $V_{tn} = -0.6V$. The circuits have no power supply inputs, meaning that all of the energy from the output signals originated from the input signals. This

method improved upon SCRL and ECRL by significantly reducing the overhead required to perform evaluation and discharge, as well as improving the signal propagation, allowing for improving cascading of the devices.

6.2.2 Energy Consumption and Security Applications

The quantum principles of adiabaticity and reversibility focus on zero-energy dissipation in ideal conditions. The terminology for an “adiabatic circuit” is different in that the use of ramp functions attempts to minimize the energy dissipated due to energy. Wires and CMOS buffers dissipate energy despite the fact that they are both logically reversible. Conventional CMOS inverters are logically reversible, yet they do not produce the input waveform if you place the equivalent output waveform on the output pin. Clock signal distribution requires even more energy. This is because switching in CMOS requires a certain amount of energy [140][141]. The extent to which this switching energy limits energy dissipation is in dispute [111][143][144][145][114]. What is not in dispute is that faster switching times and greater operating frequency require more energy dissipation.

One of the motivations for this synthesis algorithm is to minimize the number of switches and paths required to implement dual-rail adiabatic logic in security circuits. One of the drawbacks to the dual-rail methodology is that it requires more transistors and more inputs and outputs. This increases the length of the wire, which leads to greater energy dissipation. Additionally, the larger the circuit, the greater the buffering required, which also increases energy dissipation. Some adiabatic design methodologies, such as Split-Level Charge Recovery Logic (SCRL) [23], address this issue. Many other schemes were using as many as four-phase clocks to control the propagation of the signal through the datapath, which required greater overhead and constraint of the signals. Efficient Charge Recovery Logic (ECRL) [25] was

presented to address this issue by minimizing the required overhead. This algorithm was designed with these drawbacks in mind. By reducing the number of switches, we reduce the size of the circuit, the wire length, and the power consumption of the final circuit.

In Fig. 6.1, we shown the instantaneous power consumption of ByteSub stage of the Rijndael algorithm [332] used for the AES encryption standard (known as the S-box) that we synthesized using our proposed algorithm. The synthesized circuit is a 16-by-16 circuit, and consists of 8 inputs and 8 inverted inputs, as well as 8 outputs with their inverse represented on the other 8 outputs. The waveform represents all 256 possible output combinations. The dual-rail adiabatic circuit consumes an average power of 1.1850×10^{-6} W. The highest power spike was 26.4×10^{-6} W, and the lowest power spike was 18.6×10^{-6} W, which represents a 30% difference in differential power.

The S-box is also an instance where the dual-rail design methodology is advantageous to conventional and single-rail implementations. Since pass-transistor logic is used, the circuits are reversible. Placing the signals on the output pins allows us to uniquely determine the input signals. Most implementations of the Rijndael cipher require an encryption S-box and a decryption S-box. Since the dual-rail adiabatic circuit is logically and physically bijective, the gain in size incurred by the dual-rail methodology may be offset by reusing the same structure for both encryption and decryption. The S-box synthesized by our proposed algorithm requires 4806 switches. Instead of making a second S-box for decryption, we may simply place a 3-by-3 Fredkin gate – used for multiplexing - at each input and output in order to dictate flow of the circuit. Each Fredkin gate incurs a cost of 8 switches, meaning that a 16-by-16 S-box would require 32 Fredkin gates, which is an added cost of 256 switches. This approach reduces the cost of S-bix encryption/decryption from 9612 switches to 5062 switches.

Figure 6.1. Dual-Rail Adiabatic S-box Power Waveform.

Most reversible logic synthesis algorithms correlate the required logical transitions to a synthesis library consisting of logic gates. The reduction we achieve in our proposed algorithm is due to mapping directly to the implementation technology, instead of inserting the library equivalent circuits. In our comparison, we replace the individual cells in the best-known reversible designs from the RevLib benchmark library [24] – shown in Table 6.1 – in order to determine the improvement of our proposed approach.

6.3 Presented Adiabatic Synthesis Algorithm

In this section, we present our proposed adiabatic synthesis algorithm. We show how we determine the horizontal offsets used in the algorithm, and prove that the sum of the horizontal offsets is zero. In Section 6.3.1, we present our algorithm for the initial synthesis of the circuit. In Section 6.3.2, we show how using ESPRESSO reduction and Petrick’s methods for Boolean minimization for each output node provides significant minimization of the final circuit. In Section 6.3.3, we formally define the proposed algorithm, show full synthesis examples, and compare our results to previous work and benchmarks.

Table 6.1. Fundamental Gates and their Assigned Switch Values.

Gate	Library	Benchmark Switches Required
CNOT	MCT	4
Swap	EQ	12
3x3 Toffoli	MCT	8
3x3 Peres	MCT+P	12
3x3 Fredkin	MCF	8
4x4 Toffoli	MCT	12
5x5 Toffoli	MCT	16

6.3.1 Determining the Horizontal Offsets

The first stage of the algorithm involves reading the desired output values and determining the horizontal offset from the permutation matrix to the unitary matrix. Since all reversible and adiabatic logic structures are logically and physically bijective, no row or column will have more than one value of '1'. Fig. 6.2 shows the permutation matrix of the fundamental CNOT gate on the left and the unitary matrix on the right. The CNOT gate is a 2x2 reversible logic structure, which means it has four possible output combinations. The horizontal columns represent the input combinations, the vertical columns are the output combinations, and the '1' represents the matching between the input and the output. For example, the fourth column and third row of the CNOT permutation matrix is a '1', meaning that input '11' correlates to the output '10'. In the unitary matrix, the fourth row has the value '1' in the fourth column. The 1x4 matrix on the right represents the calculated horizontal offsets. We define the horizontal offset as the difference between the expected location in the unitary matrix and the actual location in the permutation matrix. The equation for each offset is $Off_i = Perm_i - Unit_i$. For example, the permutation matrix of the CNOT has the '1' in the fourth row in the third column, and the unitary matrix has the '1' in the fourth column. By using the equation, we determined the offset as $Off_4 = Perm_4 - Unit_4 = 3 - 4 = -1$.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

Figure 6.2. CNOT Permutation, Unitary, and the Horizontal Offset Matrix.

Fig. 6.3(a) shows a graph of the CNOT matrix offsets, and how the end up back at ‘11’ with offsets of -1 and 1. Fig. 6.3(b) shows the graph for the 2x2 inversion gate. Fig. 6.8(c) shows the graph for the Swap gate. Fig. 6.3(d) shows the graph for the 3x3 Toffoli Gate. Fig. 6.3(e) shows the graph for the 3x3 Fredkin gate. Fig. 6.3(f) shows the graph for the 3x3 inversion gate, and Fig. 6.3(g) shows the graph for the 3x3 Peres gate [82].

6.3.2 Determining the Initial Edges

We determine the edges and switches for the initial circuit by creating an edge for each node where the offset is a non-zero value, and comparing the binary representations of the initial and final node of the edge in the graph. If the comparison shows that all but one value are identical, then those values represent the switch along the path. The value that is different represents the initial and final node on the path. If the comparison shows that multiple values are different, and only one value is the same, then the algorithm will create an edge for every difference bit from the destination value to the origin value, with the other origin values representing the switches. If the comparison shows that no values are identical, then the algorithm will create an edge from the origin node to the destination node for each node, and the rest of the nodes represent the switches.

Fig. 6.4 shows how the initial edges are determined for the CNOT gate. In this case, the number of identical values is 1, which is the A input. Therefore, the switch value is $A=I$, which will eventually be replaced by adiabatic ‘on’ switch for ‘1’ from Fig. 6.2. The first edge is determined by the edge with the origin of {11} and destination of {10}. This means that the

origin of the edge is for B and the destination of the edge is Q'. The representation of the edge switches is {1-}. Similarly, the second edge is determined by the edge with the origin of {10} and destination of {11}. This means that the origin of the edge is for B' and the destination of the edge is Q.

Figure 6.3. Offset Graphs.

Figure 6.4. Offset Graphs (Left) and Edges (Right) for the CNOT Gate.

Fig. 6.5 shows how the initial edges are determined for the Toffoli gate. In this case, the number of identical values is 2. Therefore, the switch values are $A=1$ and $C=1$, which will eventually be replaced by adiabatic 'on' switches for '1' from Fig. 6.2. The first edge has an origin of {111} and destination of {110}. This means that the origin of the edge is for C and the destination of the edge is R'. The representation of the edge switches is [1-]. Similarly, the second edge has an origin of {110} and destination of {111}. This means that the origin of the edge is for C' and the destination of the edge is R.

Figure 6.5. Offset Graphs (Left) and Edges (Right) for the Toffoli Gate.

Fig. 6.6 shows how the initial edges are determined for the Peres gate. The graph has instances where one bit is common, such as the edge between $\{111\}$ and $\{100\}$, and where all but one bit are common, such as the edge between $\{110\}$ and $\{100\}$. The result is that six edges are produced between four node pairs. The pairing of B and Q' has switch combinations of $\{1-0\}$ and $\{1-1\}$. Since this combination of edges may be reduced to $\{-1\}$, there is an opportunity for optimization.

Figure 6.6. Offset Graphs (Left) and Edges (Right) for the Peres Gate.

There is a unique case for optimization of reversible logic circuits where the values of two inputs are swapped. This occurs in a Swap gate with no switches required, or in a Fredkin gate with the control signal providing a Controlled Swap. At this stage of the algorithm, we check for a *SwapCondition*. A *SwapCondition* exists when there are exactly two bits different in the origin and destination nodes, and the edge from the destination node has a destination to the

origin node. Fig. 6.6 shows the Offset graphs for the Swap and Fredkin gates to illustrate how the *SwapCondition* is met. In the Swap gate, there are exactly two bits of difference, and the two bits within the origin node are different. This indicates a swap where no switches are required. For the Fredkin gate, there are exactly two bits of difference, and the two bits within the origin node are different. They do share an identical bits, $A = I$, which means the swap only occurs when that condition is met. Therefore, when $A=0$, the signals are passed to the normal output value.

In this algorithm, all cases of the *SwapCondition* will be set aside and not synthesized until the last stage. This is because all of the other cases involve determining when the input signal will be propagated to its corresponding output or its inverse. In the case of the *SwapCondition*, the signal will be propagated to a completely different output. For example, in the Swap gate, the *SwapCondition* determines that the A input is propagated to the Q output, the A' input to Q' , the B input to P , and B' input is propagated to P' output. This simplifies the next two stages of the algorithm, and reduces the overall cost. For example, the Swap gate is reduced from a cost of 8 switches to the ideal cost of zero switches, and the Fredkin gate is reduced from a cost of 16 to the ideal cost of 8.

Figure 6.7. Offset Graphs of Swap (Left) and Fredkin (Right).

6.3.3 Optimization using ESPRESSO Boolean Minimization

The edges produced by the first part of the algorithm are bijective and onto, as well as having the property that every edge that originates at input node i ends and at the same output node j . This property allows for the use of minimization of Boolean functions in order to minimize the number of origin node to the destination node.

We implemented the ESPRESSO heuristic method [142] for our Boolean minimization of circuit edges. Fig. 6.7 shows the ESPRESSO minimization for the inversion gate. The inversion gate has the highest potential for optimization. The initial result is that eight edges are produced between four node pairs. The pairing of A and P' has switch combinations of $\{-0\}$ and $\{-1\}$. Since this combination of edges may be reduced to $\{--\}$, there is an opportunity for optimization. In fact, each of these edges may be reduced to $\{--\}$, which means no switches would be required. The input-output node combination for $A \rightarrow P'$ and $A' \rightarrow P$ have minterms of $\{1-\}$ and $\{0-\}$, which is reduced to $\{--\}$. The input-output node combination for $B \rightarrow Q'$ and $B' \rightarrow Q$ have minterms of $\{-1\}$ and $\{-0\}$, which is reduced to $\{--\}$. Therefore, the number of switches required in the design of the inversion gate is zero.

Fig. 6.8 shows the ESPRESSO minimization for the Peres gate. The input-output node combinations for $B \rightarrow Q'$ and $A' \rightarrow P$ both have minterms of $\{0-1\}$ and $\{1-1\}$, which is reduced to $\{--1\}$. Therefore, the number of switches required in the design of the inversion gate is reduced from 12 to 8.

The final stage of the algorithm involves finding the edges from the input node to the identical output node. Until this point, the edges go from the input node to the inverted output node. For example, all the edges from the $node_0$ will appear at the output $node_1$, since $node_0$ corresponds to input A and output $node_1$ corresponds to output P' . We need to create edges that go from input $node_0$ to output $node_0$. If output $node_i$ is the regular node, and $node_{i+1}$ is the inverted node for dual-rail implementation, then the value of output $node_i$ must never equal the value of output $node_{i+1}$. Using this rule, we determine the edges by running ESPRESSO minimization again, and using the 'don't care' values from the previous iteration as the minterms

for this iteration. The result of this stage is that the circuit becomes physically and logically reversible, and both constraints for adiabatic implementation are met.

Figure 6.8. Offset Graph for the Inversion Gate.

Figure 6.9. Edges Before (Left) and After (Right) for the Peres Gate.

As an example, we show the entire synthesis for the Toffoli gate, where the logical outputs represent $P = A$, $Q = B$, and $R = AB \oplus C$. Fig. 6.10(a) shows the initial permutation matrix for the Toffoli gate, and Fig. 6.10(b) shows the horizontal offset matrix. Fig. 6.10(c) shows the edges and nodes for the offsets. The algorithm determines that only two edges exist, reducing the complexity of the input required for the ESPRESSO portion of the algorithm. Each edge only has one bit of difference from the input and output node. Therefore, the edge from

$\{111\} \rightarrow \{110\}$ has an origin of C and a destination of R', and the switches are A=1 and B=1. Also, the edge from $\{110\} \rightarrow \{111\}$ has an origin of C' and a destination of R, and the switches are A=1 and B=1. The result of this portion of the algorithm is shown in Fig. 6.10(d). Since each input-output node combination has only one edge, the ESPRESSO Method is not required. For the final stage of the algorithm, since the minterms of each edge are $\{11-\}$, this means that the minterms creating the opposite edge are $\{00-\}$, $\{01-\}$, and $\{10-\}$. Using the ESPRESSO Method, the prime implicants are $\{0--\}$ and $\{-0-\}$. Therefore, two edges from each input-output node combination are used in the circuit, each with one switch, giving the final circuit shown in Fig. 6.9(e).

Figure 6.10. Full Synthesis of the Adiabatic Toffoli Gate.

6.3.4 Swap Condition

Here, we address the edges that meet the *SwapCondition* established earlier in the algorithm. The swap condition portion of algorithm determines when the edge should go from an origin to the swapped destination, and when the signal should propagate to the original output.

We connect the swapped outputs, and the switches on that edge are the common signals. Then, using ESPRESSO, we use those switches as the ‘don’t cares’ to determine the values required to go to the normally corresponding output.

Consider the example of the Fredkin gate in Fig. 6.11. where the logical outputs represent $P = A$, $Q = A'B \oplus AC$, and $R = AB \oplus A'C$. Figure 6.11(a) shows the initial permutation matrix for the Fredkin gate, and Fig. 6.10(b) shows the horizontal offset matrix. Fig. 6.10(c) shows the edges and nodes for the offsets. The algorithm determines that only two edges exist, and that both meet the *SwapCondition*. The common bit is $A=1$, therefore, the edge from $B \rightarrow R$, $B' \rightarrow R'$, $C \rightarrow Q$, and $C' \rightarrow Q'$ has the switch equivalent to $A=1$, as shown in Fig. 6.11(d). Using the ESPRESSO minimization method, we determine that $A=0$ is the switch required to go to the initial edge. Therefore, the circuit requires 8 total switches.

6.3.5 Formal Definition of Final Algorithm

Here, we formally define our proposed algorithm for synthesis of dual-rail adiabatic circuits. The algorithm was written in C++ using the gcc compiler version 3.4.6. The graph G and horizontal offset matrix $h[n^2]$ are global variables. The second algorithm is called *CreateInitialEdges*. It is the formal definition of the portion of the algorithm presented in Section 6.2. The third and final algorithm is the *ESPRESSO_Reduction*. It is the formal definition of the portion of the algorithm presented in Section 6.3.

We designed the proposed algorithm in C++ using the gcc compiler Version 3.4.6. The program generates a CMOS circuit in HSPICE that is used for simulation and verification. Multiple SPICE decks were used for simulation, and the 22nm predictive technology model designed by the Nanoscale Integration and Modeling (NIMO) Group (ptm.asu.edu). The

synthesis flow is shown in Fig. 6.13. As an example of a synthesis method with a larger set of gates required, we show the synthesis of a full adder circuit in Fig. 6.14.

Figure 6.11. Full Synthesis of the Adiabatic Fredkin Gate.

Figure 6.12. HSPICE Simulation of Synthesized Adiabatic Fredkin Gate.

Figure 6.13. Synthesis Flow of Proposed Algorithm.

Figure 6.14. Synthesis Flow of Adiabatic Full Adder.

ALGORITHM 6.1. AdiabaticSynthesis

Input: Number of inputs n , Array of output combinations $output[n^2]$
Output: Graph $G(V,E)$ representing the synthesized adiabatic circuit.

```
index = 0; numDiff = 0; horizontal[ $n^2$ ]; Graph  $G(V,E)$ ;  
for  $i=0$  to  $n^2-1$   
    if  $i \neq output[i]$   
        horizontal[ $i$ ] =  $output[i]-i$ ;  
        if SwapCondition is false  
            CreateInitialEdges( $i$ )  
for  $i=0$  to  $n^2-1$   
    ESPRESSO_Reduction( $i$ , horizontal[ $i$ ])  
  
for  $i=0$  to  $n^2-1$   
    SwapCondition( $i$ , horizontal[ $i$ ])  
return  $G$ 
```

ALGORITHM 6.2. CreateInitialEdges

Input: Input i
Output: Edges written to $G(V,E)$ representing the switches between the in and out nodes.
 $destination = i + horizontal[i]$; $common_bits = 0$; $bits = \lg_2(n^2)$
 $i_bits[bits]$ is the binary representation of i .
 $dest_bits[bits]$ is the binary representation of $destination$.
for $i=0$ to $bits-1$
 if $i_bits[i] == dest_bits[i]$ **then** $common_bits++$;
if $common_bits == 0$
 for $j=0$ to $bits-1$
 $edge[bits] = dest_bits$; $edge[i] = '-'$;
 add $edge[i]$ to Graph G , with org node = $dest_bits[j]$, and dest node = $org_bits[j]$
else if $common_bits == 1$
 for $j=0$ to $bits-1$
 $edge[j] = i_bits[j]$;
 if $i_bits[j] \neq dest_bits[j]$ **then** $edge[j] = '-'$;
 add $edge[i]$ to Graph G , with origin node = j , and dest. node = j_inv ;
else
 $edge[bits]$;
 for $j=0$ to $bits-1$
 if $i_bits[j] == dest_bits[j]$ **then** $edge[j] = i_bits[j]$;
 for $j=0$ to $bits-1$
 $edge[j] = i_bits[j]$;
 if $i_bits[j] \neq dest_bits[j]$ **then** $edge[j] = '-'$;
 add $edge[i]$ to Graph G , with origin node = j , and destination node = j_inv ;

ALGORITHM 6.3. ESPRESSO_Reduction

Input: Input i

Output: Edges written to $G(V,E)$ representing the switches between the input and output nodes.

```
vector<int> minterms; vector<int> dont_cares;  
int num_edges = Graph.vertex(i).getNumEdges();  
int origin = Graph.vertex(i).getOrigin();  
int destination = Graph.vertex(i).getDestination();  
for j=0 to num_edges-1;  
    add integer value of the edge to minterms;
```

```
for j=0 to inputs2-1;  
    if minterms does not contain j then add j to dont_cares  
run ESPRESSO(minterms, dont_cares);  
delete all edges from origin to destination
```

```
for j=0 to number of prime implicants;  
    add edge to Graph G with origin, destination, and prime implicant representing the  
switches  
if origin%2 == 0 then origin++; destination--;  
else then origin--; destination++;  
run ESPRESSO(dont_cares, minterms);
```

```
for j=0 to number of prime implicants;  
    add edge to Graph G with origin, destination, and prime implicant representing the  
switches
```

ALGORITHM 6.4. SwapCondition

Input: Input i

Output: Edges written to $G(V,E)$ representing the switches between the input and output nodes.

```
vector<int> minterms; vector<int> dont_cares;  
int num_edges = Graph.vertex(i).getNumEdges();  
int origin = Graph.vertex(i).getOrigin();  
int destination = Graph.vertex(i).getDestination();  
for j=0 to num_edges-1;  
    add integer value of the edge to minterms;
```

```
for j=0 to inputs2-1;  
    if minterms does not contain j then add j to dont_cares  
    add edge to Graph G with origin, destination, and prime implicant representing the switches  
from Diff_A to Diff_B and the negation edge, and Diff_B to Diff_A  
run ESPRESSO(dont_cares, minterms);  
    add edge to Graph G with origin, destination, and prime implicant representing the switches  
from Diff_A to Diff_A and the negation edge, and Diff_B to Diff_B
```

6.4 Comparison to Previous Benchmarks

Table 6.1 compares the synthesized circuits by the algorithm with the benchmarks presented in the *RevLib* library, which consists of reversible logic structures from the MCT, MCF, MCT+P, and EQ libraries. The Swap gate is given a value of 12 because in all of the provided libraries, the Swap requires 3 CNOT gates, each of which requires 4 switches.

Table 6.2 shows the number of transistors required for the synthesis of fundamental reversible logic structures. The table shows the desired output sequence presented by the user in the first column. The second column shows the generated horizontal offsets based upon these outputs. The third column shows the initial number of switches generated by the *CreateInitialEdges* portion of the Algorithm (Algorithm 2). The fourth column shows the number of switches remaining after the *ESPRESSO_Reduction* is performed. The fifth column shows the final number of switches after the *ESPRESSO_Reduction* uses the ‘don’t cares’ to obtain the opposite edges in order to ensure the circuit meets both the logical requirements for reversible logic, and the physical requirements for adiabatic implementation. The sixth column shows the reversible function that was achieved by the synthesis algorithm, and the last column shows the equivalent benchmark circuit in the *RevLib* library [89]. To make the comparison, the corresponding reversible logic gates were assigned their switch values in the adiabatic library, which are shown in Table 6.3 and 6.4. The benchmarks 3_17, 4_49, hwb4_12, and hwb5_13 are considered as “worst case scenarios for several synthesis algorithms” in RevLib. In these cases, the presented algorithm beat the previous best results by 27%. On average, of the benchmarks included in this paper, the presented algorithm improved upon the previous results by 36%. Next, we ran power analysis on each benchmark and synthesized circuit using 22nm predictive technology at 13.56MHz. These results are shown in the last two columns of Tables 6.5-6.8.

6.5 Conclusions

We presented a novel algorithm for synthesis of adiabatic logic structures in CMOS. The resulting circuits are synthesized and simulated using HSPICE. The synthesis results show that, on average, the proposed algorithm represents an improvement of 36% over the best known reversible designs with the optimized dual-rail cells. Synthesis of a dual-rail adiabatic S-box for the Rijndael cipher show improvement in differential power over single-rail and conventional implementations. These results suggest that dual-rail adiabatic logic is a promising design methodology for circuits where security is this most important design metric, and frequencies are slower, such as the ISO 14443 Smart Card standard.

Table 6.2. Synthesis of Fundamental Reversible Logic Structures Stages.

Output Sequence	Horizontal Offsets	CIE	# Switch Reduce	Switch Swap Cond.
{0,1}	{0, 0}	0	0	0
{1,0}	{-1, 1}	0	0	0
{0,1,2,3}	{0, 0, 0, 0}	0	0	0
{3,2,1,0}	{3, 1, -1, 3}	0	0	0
{0,2,1,3}	{0, 1, -1, 0}	0	0	0
{3,1,2,0}	{3, 0, 0, 3}	0	0	0
{0,1,3,2}	{0, 0, 1, -1}	2	2	2
{0,3,2,1}	{0, 2, 0, -2}	2	2	3
{3,2,0,1}	{3, 1, -2, -2}	6	2	2
{3,0,1,2}	{3,-1,-1,-1}	6	2	2
{0,1,2,3,4,5,6,7}	{0,0,0,0,0,0,0,0}	0	0	0
{7,6,5,4,3,2,1,0}	{7,5,3,1,-1,-3,-5,-7}	48	0	0
{0,1,2,3,4,5,7,6}	{0,0,0,0,0,0,1,-1}	4	4	4
{7,6,5,4,3,2,0,1}	{7,5,3,1,-1,-3,-6,-6}	44	4	4
{0,1,2,3,4,7,6,5}	{0,0,0,0,0,2,0,-2}	4	4	4
{7,6,5,4,3,0,1,2}	{7,5,3,1,-1,-5,-5,-5}	44	4	4
{0,1,2,7,4,5,6,3}	{0,0,0,4,0,2,0,-4}	4	4	4
{7,6,5,0,3,2,1,4}	{7,5,3,-3,-1,-3,-5,-3}	44	4	4
{0,1,2,3,6,7,5,4}	{0,0,0,0,2,2,-1,-3}	12	6	6
{7,6,5,4,1,0,2,3}	{7,5,3,1,-3,-5,-4,-4}	36	6	6
{0,1,2,3,4,6,5,7}	{0,0,0,0,0,1,-1,0}	0	0	8
{7,6,5,4,3,1,2,0}	{7,5,3,1,-1,-5,-3,-7}	40	0	8
{0,1,2,6,4,5,3,7}	{0,0,0,3,0,0,0,-3,0}	0	0	8
{7,6,5,1,3,2,4,0}	{7,5,3,2,-1,-3,-4,-7}	40	0	8
{7,6,5,1,3,2,4,0}	{7,5,3,2,-1,-3,-4,-7}	40	0	8
{7,6,5,1,3,2,4,0}	{7,5,3,2,-1,-3,-4,-7}	40	0	8
{0,3,2,1,6,5,4,7}	{0,2,0,-2,2,0,1,-3}	12	8	8
{7,4,5,6,1,2,3,0}	{7,3,3,3,-3,-3,-3,-7}	40	8	8
{16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0}	{15,13,11,9,7,5,3,1,-1,-3,-5,-7,-9,-11,-13,-15}	192	0	0
{0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,14}	{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,-1}	6	6	6
{16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0}	{15,13,11,9,7,5,3,1,-1,-3,-5,-7,-9,-11,-14,-14}	186	6	6
{0 1 2 3 6 7 5 4 10 11 9 8 13,12 15 14}	{0,0,0,0,2,2,-1,-3,2,2,-1,-3,1,-1,1,-1}	48	18	20
{0,1,2,3,6,7,5,4,10,11,9 8 13 12 15 14}	{0,0,0,0,2,2,-1,-3,2,2,-1,-3,1,-1,1,-1}	48	18	20
{15 14 13 12 9 8 10 11 5 4 6 7 2 3 0 1}	{15,13,11,9,5,3,4,4,-3,-5,-4,-4,-10,-10,-14,-14}	144	16	20

Table 6.3. Synthesis of Fundamental Reversible Logic Structures Results.

Output Sequence	Horizontal Offsets	Final switches	Reversible Function	Achieved Benchmark
{0,1}	{0, 0}	0	2x2 Unitary	NA
{1,0}	{-1, 1}	0	2x2 Inversion	inv_2
{0,1,2,3}	{0, 0, 0, 0}	0	2x2 Unitary	NA
{3,2,1,0}	{3, 1, -1, 3}	0	2x2 Inversion	inv_2
{0,2,1,3}	{0, 1, -1, 0}	0	2x2 Swap	swap_2
{3,1,2,0}	{3, 0, 0, 3}	0	2x2 Inversion and Swap	swap_2
{0,1,3,2}	{0, 0, 1, -1}	4	2x2 CNOT	cnot
{0,3,2,1}	{0, 2, 0, -2}	4	2x2 CNOT	cnot
{3,2,0,1}	{3, 1, -2, -2}	4	2x2 Inversion and CNOT	cnot
{3,0,1,2}	{3,-1,-1,-1}	4	2x2 Inversion and CNOT	cnot
{0,1,2,3,4,5,6,7}	{0,0,0,0,0,0,0,0}	0	3x3 Unitary	NA
{7,6,5,4,3,2,1,0}	{7,5,3,1,-1,-3,-5,-7}	0	3x3 Inversion	inv_3
{0,1,2,3,4,5,7,6}	{0,0,0,0,0,1,-1}	8	3x3 Toffoli	tof_3
{7,6,5,4,3,2,0,1}	{7,5,3,1,-1,-3,-6,-6}	8	3x3 Inversion and Toffoli	tof_3
{0,1,2,3,4,7,6,5}	{0,0,0,0,2,0,-2}	8	3x3 Toffoli	tof_3
{7,6,5,4,3,0,1,2}	{7,5,3,1,-1,-5,-5,-5}	8	3x3 Inversion and Toffoli	tof_3
{0,1,2,7,4,5,6,3}	{0,0,0,4,0,2,0,-4}	8	3x3 Toffoli	tof_3
{7,6,5,0,3,2,1,4}	{7,5,3,-3,-1,-3,-5,-3}	8	3x3 Inversion and Toffoli	tof_3
{0,1,2,3,6,7,5,4}	{0,0,0,0,2,2,-1-3}	12	3x3 Peres	peres_4
{7,6,5,4,1,0,2,3}	{7,5,3,1,-3,-5,-4,-4}	12	3x3 Inversion and Peres	peres_4
{0,1,2,3,4,6,5,7}	{0,0,0,0,1,-1,0}	8	3x3 Fredkin	fred_3
{7,6,5,4,3,1,2,0}	{7,5,3,1,-1,-5,-3,-7}	8	3x3 Inversion and Fredkin	fred_3
{0,1,2,6,4,5,3,7}	{0,0,0,3,0,0,0,-3,0}	8	3x3 Fredkin	fred_3
{7,6,5,1,3,2,4,0}	{7,5,3,2,-1,-3,-4,-7}	8	3x3 Inversion and Fredkin	fred_3
{0,3,2,1,6,5,4,7}	{0,2,0,-2,2,0,1,-3}	16	3x3 UPG	NA
{7,4,5,6,1,2,3,0}	{7,3,3,3,-3,-3,-3,-7}	16	3x3 Inversion and UPG	NA
{16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0}	{15,13,11,9,7,5,3,1,-1,-3,-5,-7,-9,-11,-13,-15}	0	4x4 Inversion	inv_4

Table 6.3. (Cont.).

Output Sequence	Horizontal Offsets	Final switches	Reversible Function	Achieved Benchmark
{0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,14}	{0,0,0,0,0,0,0,0,0,0,0,0,1,-1}	12	4x4 Toffoli	tof_4
{16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0}	{15,13,11,9,7,5,3,1,-1,-3,-5,-7,-9,-11,-14,-14}	12	4x4 Inversion and Toffoli	tof_4
{0,1,2,3,6,7,5,4,10,11,9,8,13,12,15,14}	{0,0,0,0,2,2,-1,-3,2,2,-1,-3,1,-1,1,-1}	20	4x4 Full Adder	rd32
{15,14,13,12,9,8,10,11,5,4,6,7,2,3,0,1}	{15,13,11,9,5,3,4,4,-3,-5,-4,-4,-10,-10,-14,-14}	20	4x4 Inversion and Full Adder	rd32

Table 6.4. Comparison of Synthesized Designs to Benchmarks.

Benchmark	Library	Benchmark Switches Required	Switches Produced by Synthesis Algorithm	Switches Improvement
3_17_12	MCT	30	28	6.67%
3_17_13	MCT	32	28	12.5%
3_17_14	EQ	28	28	0%
4_49	MCT	112	78	30.35%
millier_11	MCT	32	24	25%
millier_12	MCT	24	24	0%
fredkin_5	EQ	18	16	11.11%
fredkin_6	MCT	24	16	33.33%
peres_8	EQ	12	12	0%
toffoli_double_4	EQ	24	22	8.33%
toffoli_double_3	EQ	20	22	-8.33%
hwb4_49	MCT	108	96	11.11%
hwb4_50	MCT+P	108	96	11.11%
hwb4_51	MCF	156	96	38.46%
hwb4_52	MCT	60	96	-37.5%
hwb7_59	MCT	4188	3392	19.00%
hwb7_61	MCT	3100	2486	19.80%
alu_9	MCT	32	32	0%
alu1_198	MCT	238	196	14.03%
alu2_96	MCT	6280	3392	45.98%
alu2_199	MCT	4523	2624	41.98%
9symml_193	MCT	5552	3186	57.38%

Table 6.4. (Cont.).

Benchmark	Library	Benchmark Switches Required	Switches Produced by Synthesis Algorithm	Switches Improvement
radd_250	MCT+ MCP	540	492	8.88%
graycode6_47	MCT	20	20	0%
radd_250	MCT+ MCP	540	492	8.88%
ham3_102	MCT	24	28	-14.28%
ham3_104	MCT+ MCP	20	28	-28.57%
mini_alu_167	MCT	112	96	14.28%
mini_alu_305	MCT	104	96	7.69%
decod24-bdd_294	MCT	48	32	33.33%
decod24-v0_38	MCT	32	32	0%
decod24-v0_40	EQ	28	32	-14.28%
decod24-v2_41	MCT	54	32	40.74%
decod24-v2_43	EQ	28	32	-14.28%
decod24- enable_124	MCF	48	48	0%
decod24- enable_125	MCT	52	48	7.69%
decod24- enable_126	MCT	152	48	68.42%
sym6_145	MCT	464	228	50.86%
sym9_146	MCT	184	156	15.21%
rd32_271	MCT	44	32	27.27%
sys6-v0_111	MCT	132	96	27.27%
win_266	MCT	190	172	9.47%
pm1_249	MCT	288	192	33.34%
ryy6_256	MCT	1544	572	62.95%
alu-v4_36	MCT	48	28	41.67%
alu-v4_37	MCT	34	28	17.64%
alu-v0_26	MCT	34	28	17.64%

Table 6.5. Benchmark Power Comparison.

Benchmark	Average Power Benchmarks (W)	Average Power Synthesis Algorithm (W)
3_17_12	7.3971E-09	6.9040E-09
3_17_13	7.8902E-09	6.9040E-09
3_17_14	6.9040E-09	6.9040E-09
4_49	2.7616E-08	1.9232E-08
millier_11	7.8902E-09	5.9177E-09
millier_12	5.9177E-09	5.9177E-09
fredkin_5	4.4383E-09	3.9451E-09
fredkin_6	5.9177E-09	3.9451E-09
peres_8	2.9588E-09	2.9588E-09
toffoli_double_4	5.9177E-09	5.4245E-09
toffoli_double_3	4.9314E-09	5.4245E-09
hwb4_49	2.6630E-08	2.3671E-08
hwb4_50	2.6630E-08	2.3671E-08
hwb4_51	3.8465E-08	2.3671E-08
hwb4_52	1.4794E-08	2.3671E-08
hwb7_59	1.0326E-06	8.3637E-07
hwb7_61	7.6437E-07	6.1297E-07
alu_9	7.8902E-09	7.8902E-09
alu1_198	5.8684E-08	4.8328E-08
alu2_96	1.5485E-06	8.3637E-07
alu2_199	1.1152E-06	6.4700E-07
9symml_193	1.3690E-06	7.8557E-07
radd_250	1.3315E-07	1.2131E-07
graycode6_47	4.9314E-09	4.9314E-09
cm42a_207	7.1012E-08	4.7341E-08
ham3_102	5.9177E-09	6.9040E-09
ham3_104	4.9314E-09	6.9040E-09
mini_alu_167	2.7616E-08	2.3671E-08
mini_alu_305	2.5643E-08	2.3671E-08
decod24-bdd_294	1.1835E-08	7.8902E-09
decod24-v0_38	7.8902E-09	7.8902E-09
decod24-v0_40	6.9040E-09	7.8902E-09
decod24-v2_41	1.3315E-08	7.8902E-09
decod24-v2_43	6.9040E-09	7.8902E-09
decod24-enable_124	1.1835E-08	1.1835E-08
decod24-enable_125	1.2822E-08	1.1835E-08
decod24-enable_126	3.7479E-08	1.1835E-08
sym6_145	1.1441E-07	5.6218E-08

Table 6.5. (Cont.).

sym9_146	4.5369E-08	3.8465E-08
rd32_271	1.0849E-08	7.8902E-09
sys6-v0_111	3.2547E-08	2.3671E-08
win_266	4.6848E-08	4.2410E-08
pm1_249	7.1012E-08	4.7341E-08
ryy6_256	3.8070E-07	1.4104E-07
alu-v4_36	1.1835E-08	6.9040E-09
alu-v4_37	8.3834E-09	6.9040E-09
alu-v0_26	8.3834E-09	6.9040E-09
alu_9	7.8902E-09	7.8902E-09
alu1_198	5.8684E-08	4.8328E-08
alu2_96	1.5485E-06	8.3637E-07
alu2_199	1.1152E-06	6.4700E-07
9symml_193	1.3690E-06	7.8557E-07
radd_250	1.3315E-07	1.2131E-07
graycode6_47	4.9314E-09	4.9314E-09

CHAPTER 7

ADIABATIC DYNAMIC DIFFERENTIAL LOGIC

In this chapter, we will apply the Forward Body Biasing technique presented in [119] to a conventional inverter, an SCRL inverter and an ECRL inverter to demonstrate the effectiveness of body biasing in dynamic differential logic¹. The inverters simulated are a conventional inverter with no body bias, a conventional inverter with body biasing, SCRL inverters with and without body biasing, and ECRL inverters with and without body biasing. Fig. 7.1 shows the layout of each of these inverters. The presented results in HSPICE using the 22nm predictive technology model presented in [138]. To show the improvement at subthreshold operation, we set the supply voltage of the circuits 0.5V, and the input frequency is 13.56 MHz.

7.1 Comparison of Inverter A and Inverter B (Conventional)

The output waveform simulations for Inverters (a) and (b) from Fig 7.1 now referred to as INVa and INVb, are presented in Fig. 7.2, and the measurement analysis is presented in Table 7.1. Performing transient analysis of INVa at 1MHz gave optimized rise and fall times when the transistor sizing was $W_P/W_N = 540nm/324nm$. We obtained a body bias of $V_b = 0.14V$ by determining the minimum difference in the peak power on the output switches for INVb. The result is that INVb has improved average power, $P_{Peak_{fall}}$ and differential power over INVa, with the tradeoff coming in increased $P_{Peak_{rise}}$. The average power of INVb is 5.5678×10^{-10} W, which represents a 4.58% improvement over INVa. Additionally, the INVb value for $P_{Peak_{fall}}$ is 2.8299×10^{-8} , which is an improvement of 7.08%. The INVb value for $P_{Peak_{rise}}$ is 2.7697×10^{-8} W,

¹ Portions of this chapter were published in IEEE Conference on VLSI Design [116]. Permission is included in Appendix A.

which is an increase of 28.35%. This increase is beneficial for the design, however, since this results in a significant reduction in P_{diff} , where has a value of 6.0202×10^{-9} W. This is an improvement in the differential power of INVa of 43.28%.

Figure 7.1. Adiabatic Inverters With and Without Biasing.

Fig. 7.2 shows the benefit of this improvement in differential power. The instantaneous power of INVa (second from the bottom) shows a distinct difference in power spikes when the output rises and when the output falls, making it easier for an attacker to determine the input signal based on the output signal and the instantaneous power waveform. The power spikes for INVb (bottom) are very similar, mitigating the effectiveness of a simple DPA attack. The top waveform shows the input signal. The second and third waveforms show the output signals of INVa and INVb, respectively. The fourth waveform shows the instantaneous power of the INVa (red) inverter and the INVb inverter (black). Though the output signals are similar, the improvement difference in power spikes for the switching events for INVb is demonstrated.

Fig. 7.3 shows their instantaneous power waveforms compared to the power spikes presented in Fig. 7.1. The conventional inverter is in maroon. INVa is in blue, and INVb is in pink. The INVb design improves upon the differential power of the conventional inverter by a factor of 957.51.

Figure 7.2. HSPICE Simulation of INVa and INVb at 1MHz.

Figure 7.3. Comparison of Instantaneous Power Waveforms of INVa and INVb.

7.2 Comparison of Inverter C and Inverter D (SCRL)

Here, we conduct the same simulations on the design for Inverters (c) and (d) from Fig. 7.1, now referred to as INVc and INVd, and the measurement analysis is also presented in Table 7.1. Performing transient analysis of INVb at 1MHz gave optimized rise and fall times when the transistor sizing was $W_P/W_N = 500nm/300nm$. The simulation results are presented in Fig. 7.4. The top waveform is the signal. The second waveform shows the phi and phibar signals used for evaluation and discharge. The third waveform from the top shows the comparison of the *out* signal for INVc (pink) and INVd (black). The third waveform shows the comparison of the *outbar* signal for INVc (red) and INVd (blue). They show improvement in rise and fall time, and operation for INVd. The bottom waveform shows the instantaneous power of INVc (blue) and INVd (red).

At 1MHz, the SCRL inverter INVc has an average power consumption of 9.1178×10^{-11} W, which is an improvement of 84.37% over INVa. The INVc value for $P_{Peak_{fall}}$ is 7.1045×10^{-10} W, an improvement of 76.67% over INVa. Also, the INVc value for $P_{Peak_{rise}}$ is 4.0357×10^{-9} W, which is an improvement of 79.66% over INVa.

The inverter INVd was found to be biased at the midpoint, $V_b = 0.25V$, since the voltage waveforms are held at the midpoint when switching does not occur in the SCRL implementation. The INVd value for $P_{Peak_{fall}}$ is 5.6502×10^{-10} , an improvement of 23.75% over INVc, and 81.44% over INVa. The INVd value for $P_{Peak_{rise}}$ is 3.3563×10^{-9} , an improvement of 16.83% over INVc, and 83.08% over INVa. The differential power of INVd is an improvement of 15.28% over INVc, and an improvement of 73.07% over INVa.

Fig. 7.5 shows their instantaneous power waveforms compared to the power spikes presented in Fig. 7.1. The conventional inverter is in black. INVc is shown in blue, and has a

significantly smaller differential power, but is still visible. INVf is shown in red, and has the smallest differential power and is red, and is barely visible at this scale, demonstrating its effectiveness. The INVd design improves upon the differential power of the conventional inverter by a factor of 2065.2.

Figure 7.4. HSPICE Simulation of INVc and INVd at 1MHz.

Figure 7.5. Comparison of Instantaneous Power Waveforms of INVc and INVd.

7.3 Comparison of Inverter E and Inverter F (ECRL)

We continue the same simulations on the design for Inverters (e) and (f) from Fig. 7.1, now referred to as INVc and INVd, which are the ECRL inverters. The measurement analysis is also presented in Table 7.1. The simulation results are shown in Fig. 7.6. The top waveform shows the input signals, in and inbar, required for dual rail application. The second waveform shows the phi and phibar signals used for evaluation and discharge. The third waveform from the top shows the comparison of the *out* signal for INVe (red) and INVf (blue). The third waveform shows the comparison of the *outbar* signal for INVe (black) and INVf (maroon). They show improvement in rise and fall time, and operation for INVf. The bottom waveform shows the instantaneous power of INVe (blue) and INVf (red).

At 1MHz, the ECRL inverter INVe has an average power consumption of 9.1178×10^{-11} W, which is an improvement of 90.27% over INVa. The INVe value for $P_{Peak_{fall}}$ is 1.1649×10^{-9} W, an improvement of 96.17% over INVa. Also, the INVc value for $P_{Peak_{rise}}$ is 6.3389×10^{-14} W, which is an improvement of 99.99% over INVa.

The inverter INVf was found to be biased at the midpoint, $V_b = 0.25V$, since the voltage waveforms are held at the midpoint when switching does not occur in the SCRL implementation. The average power consumption of the INVf inverters is 3.0568×10^{-11} W, which is a 94.76% improvement over INVa. The INVf value for $P_{Peak_{fall}}$ is 3.4382×10^{-10} , an improvement of 70.48% over INVe, and 98.87% over INVa. The INVf value for $P_{Peak_{rise}}$ is 3.3563×10^{-9} , an improvement of 16.83% over INVc, and 83.08% over INVa. The differential power of INVd is 1.3083×10^{-14} W. an improvement of 79.36% over INVe, and an improvement of 99.99% over INVa.

Fig. 7.7 shows the instantaneous power waveforms of INVe and INVf compared to the power spikes presented in Fig. 7.1. The INVf design improves upon the differential power of the conventional inverter by a factor of 16772.09. The conventional inverter is in black. INVe is shown in blue, and has a significantly smaller differential power, but is still visible. INVf is shown in red, and has the smallest differential power and is red, and is barely visible at this scale, demonstrating its effectiveness.

Table 7.1. HSPICE Simulation Measurements for Inverters in Fig. 7.1-7.7.

Inv Design	W_P (nm)	W_N (nm)	V_{Bias}	P_{Avg} ($W \cdot 10^{-10}$)	$P_{PeakRise}$ ($\cdot 10^{-8}$)	$P_{Peakfall}$ ($\cdot 10^{-8}$)	P_{diff} ($\cdot 10^{-8} W$)
INV _A	540	324	0	5.8353	1.9843	3.0458	1.0615
INV _B	540	324	0.14V	5.5678	2.7697	2.8299	6.0202
INV _C	500	300	0	.91178	.40357	.0741	0.3295
INV _D	500	300	0.25V	1.0501	.33563	.05650	0.2791
INV _E	500	300	0	.56746	$6.3389 \cdot 10^{-6}$	.1166	0.1695
INV _F	500	300	0.25V	.30568	$1.3083 \cdot 10^{-6}$	.03438	0.0344

Figure 7.6. HSPICE Simulation of INVe and INVf at 1MHz.

Figure 7.7. Comparison of Instantaneous Power Waveforms of INVe and INVf.

7.4 Forward Body Biased Adiabatic Toffoli Gate

The body biasing method used to improve the average, peak, and differential power of the conventional and adiabatic inverters was used to achieve similar improvements in a CMOS adiabatic Toffoli gate. The gate layout of the circuit is shown in Fig. 7.8. The top waveform represents the A and A' signals. The second waveform represents the B and B' signals. The third waveform represents the C and C' signals. The fourth waveform represents the R and R' outputs. The bottom waveform is the instantaneous power. The HSPICE simulation waveforms are shown in Fig. 7.9. The average power of the body-biased Toffoli gate is 2.1865×10^{-10} W, and the peak power is 1.9348×10^{-8} W. The largest differential power in peaks is 2.29×10^{-9} W.

The simulation in Fig. 7.10 shows the difference in operation between the body biased Toffoli gate and the non-body biased Toffoli gate at 0.5V. The top waveform shows the R and R' outputs for the non-body biased Toffoli gate. The mid-level outputs for the non-body biased show degraded signals, whereas the body-biased outputs have proper operation. The non-body

biased Toffoli gate has an average power of 4.4622×10^{-10} W, and the peak power is 2.0126×10^{-8} W. The largest differential power in positive peaks is 2.6101×10^{-9} W. Therefore, the body biased Toffoli gate has improvements of 50.9% in average power, a 3.86% improvement in peak power, and 12.25% improvement in differential power.

Figure 7.8. Schematic Layout of CMOS Adiabatic Toffoli Gate.

Figure 7.9. HSPICE Simulation of CMOS Adiabatic Body Biased Toffoli Gate.

The difference in power is accounted for in the degraded output signal of the non-body biased Toffoli gate. Since there is no body biasing, the output signal for R and R' is unable to get back to the midpoint of 0.25V. The more the signal is degraded, the more the reversibility of the circuit is compromised. As a result, the energy dissipation and power consumption of the device increases.

Fig. 7.11 shows a comparison between the instantaneous power of the body-biased Toffoli gate and the non-body biased NAND gate, both at 0.5V supply voltage. Even though the NAND gate has 4 transistors and the Toffoli gate requires 16, the Toffoli gate operates at an improvement of 68.49% in average power, a 93.41% improvement in peak power, and an improvement in differential power by a factor of 128.43.

Figure 7.10. HSPICE Simulation Comparison of Adiabatic Toffoli Gates.

Figure 7.11. Power of NAND Gate (Black) and Toffoli Gate (Red) at 0.5V.

7.5 Forward Body Biased Adiabatic Fredkin Gate

The same body biasing technique was used to improve the average, peak, and differential power of the CMOS body-biased Fredkin gate. The gate layout of the circuit is shown in Fig.

7.12. The HSPICE simulation waveforms are shown in Fig. 7.13. The top, second and third waveform represents the A and A' signals, the B and B' signals, and the C and C' signals, respectively. The fourth waveform represents the Q and Q' outputs. The fourth waveform represents the R and R' outputs. The bottom waveform is the instantaneous power. The average power is 1.8829×10^{-10} W, and the peak power is 1.2077×10^{-8} W. The largest differential power in positive peaks is 3.0310×10^{-9} W.

Figure 7.12. Schematic Layout of CMOS Adiabatic Fredkin Gate.

Figure 7.13. HSPICE Simulation of CMOS Adiabatic Body Biased Fredkin Gate.

The simulation in Fig. 7.14 shows the difference in operation between the body biased Fredkin gate and the non-body biased Fredkin gate at 0.5V. The top waveform shows the Q and Q' outputs for the non-body biased Toffoli gate. The mid-level outputs for the non-body biased

show degraded signals, whereas the body-biased outputs have proper operation. This is similar for the R and R' outputs, which are shown for the non-body biased and body biased Fredkin gates in waveforms 3 and 4 in Fig. 7.14, respectively. The non-body biased Fredkin gate has an average power of 5.3726×10^{-10} W, and the peak power is 1.2959×10^{-8} W. The largest differential power in positive peaks is 2.5363×10^{-9} W. Therefore, the body biased Fredkin gate has improvements of 64.95% in average power, a 7.80% improvement in peak power, and 16.32% improvement in differential power. The difference in power is accounted for in the degraded output signal of the non-body biased Fredkin gate.

Figure 7.14. HSPICE Simulation Comparison of CMOS Adiabatic Fredkin Gates.

Fig. 7.15 shows a comparison between the instantaneous power of the body-biased Fredkin gate and the non-body biased pass transistor MUX, both at 0.5V supply voltage. Even though the MUX has 4 transistors and the Fredkin gate requires 16, the Fredkin gate operates at an improvement of 89.56% in average power, a 95.14% improvement in peak power, and an improvement in differential power by a factor of 74.27.

Figure 7.15. Power of MUX Gate (Red) and Fredkin Gate (Blue) at 0.5V.

7.6 Performance Adiabatic Dynamic Differential Logic (PADDL) Cell

In this section, we present method for implementation of High-Performance Adiabatic Dynamic Differential Logic (PADDL) design methodology for mitigating DPA attacks in high-performance applications. The data presented in this section was obtained using HPSICE simulations using the 22nm predictive technology model presented in [138].

The objective of PADDL is to design as a universal cell capable of dynamically performing all of the fundamental two-input logical calculations (AND, NAND, OR, NOR, XOR and XOR) with the minimal differential power for each logical calculation. The device is both logically and physically bijective. This means that the input waveforms may be uniquely determined by reading the output waveforms, a necessity in implementation of low power reversible and adiabatic designs.

The logical calculations of the output signals of PADDL are $P = \bar{A}$, $\bar{P} = A$, $Q = \overline{(A + B) \oplus C}$, $\bar{Q} = (A + B) \oplus C$, $R = \overline{AB \oplus C}$, and $\bar{R} = AB \oplus C$. The truth table of the device is shown in Table 7.2, and the logic outputs of PADDL are presented in Table 7.3. Fig. 7.16 shows the design process of the PADDL cell. The objective of the basic square circuit diagram is to determine the switches required for an input signal to flow from an input to an output. Consider

Fig. 7.16(a): in order for the output Q to be '1' when input C is a '1', either A or B must be a '1', which would close the switch. The circuit diagram shows whether the switch will open or close when the appropriate input signal is a '1'. The output Q is determined with the circuit Fig. 7.16(a) and the output R is determined with circuit Fig. 7.16(b).

Figure 7.16. Basic Square Circuit Diagram for Proposed PADDL Cell.

Table 7.2. Truth Table for Proposed PADDL Cell.

A	A'	B	B'	C	C'	P	P'	Q	Q'	R	R'
0	1	0	1	0	1	1	0	1	0	1	0
0	1	0	1	1	0	1	0	0	1	0	1
0	1	1	0	0	1	1	0	0	1	1	0
0	1	1	0	1	0	1	0	1	0	0	1
1	0	0	1	0	1	0	1	0	1	1	0
1	0	0	1	1	0	0	1	1	0	0	1
1	0	1	0	0	1	0	1	0	1	0	1
1	0	1	0	1	0	0	1	1	0	1	0

Fig. 7.17 shows the gate level design of the PADDL cell derived from the. The device has 32 transistors, each of which have their gate, drain and source tied to an input or output signal. The PMOS transistors are biased to the nominal supply voltage, which is 0.8V in the 22nm model in [138], and the NMOS transistors are biased to ground. The advantage of this approach is that evaluation and discharge signals are not required, meaning that less power is consumed by the circuit, even though the device has more transistors.

The simulation waveform of the PADDL cell is presented in Fig. 7.18. The first, second and third waveforms show the A and A', B and B', and C and C' waveforms, respectively. The fourth waveform represents the Q and Q' output signals. The fifth waveform represents the R and R' output signals. The bottom waveform is the instantaneous power. The average power of the PADDL device at 1MHz is 1.4634×10^{-9} W. Therefore, it consumes 1.4634×10^{-15} J of energy every computing cycle. Like to the ECRL inverter, the instantaneous power dissipation is similar for every switching event, regardless of the input and output signals. The highest switching peak power of the device is 9.6867×10^{-8} W, and the smallest switching peak power is 8.6750×10^{-8} W, giving a differential power of 1.0117×10^{-8} W.

Table 7.3. PADDL Cell Logic Outputs.

Control Signal	P	P'	Q	Q'	R	R'
A=0	A'	A	$\overline{B \oplus C}$	$B \oplus C$	C'	C
A=1	A'	A	C'	C	$\overline{B \oplus C}$	$B \oplus C$
B=0	A'	A	$\overline{A \oplus C}$	$A \oplus C$	C'	C
B=1	A'	A	C'	C	$\overline{B \oplus C}$	$B \oplus C$
C=0	A'	A	$\overline{A + B}$	$A + B$	$\overline{AB}$	AB
C=1	A'	A	$A + B$	$\overline{A + B}$	AB	$\overline{AB}$

Figure 7.17. CMOS Schematic Diagram for Proposed PADDL Cell.

The instantaneous power of the PADDL device is compared to a NAND gate of identical transistor sizing in Fig. 7.18. The average power of the 4-transistor NAND gate is 2.4926×10^{-9} W, meaning the PADDL circuit gives an improvement of 41.29%, despite have 28 more transistors. Furthermore, the differential power of the largest and smallest peaks is 1.1413×10^{-6} W, meaning that the PADDL cell improves upon the differential power by a factor of 112.81.

Figure 7.18. HSPICE Simulation of PADDL Logic Cell at 1MHz.

Figure 7.19. Power of NAND Gate (Blue) and Toffoli Gate (Red) at 1MHz.

The operation of the PADDL cell at 13.56MHz is shown in Fig. 7.20, and the same comparison of the PADDL cell and the NAND gate operation is shown in Fig. 7.21. The first, second and third waveforms show the A and A', B and B', and C and C' waveforms, respectively. The fourth waveform represents the Q and Q' output signals. The fifth waveform represents the R and R' output signals. The bottom waveform is the instantaneous power. The operating frequency of 13.56MHz is chosen since it is the standard operating frequency of smart cards using the ISO/IEC 14443.

The average power of the PADDL device at 13.56MHz is 8.5963×10^{-9} W, therefore it consumes 6.3394×10^{-16} J of energy every computing cycle. The highest switching peak power of the device is 1.4430×10^{-6} W, and the smallest switching peak power is 1.3274×10^{-6} W, giving a differential power of 1.156×10^{-7} W. The average power of the NAND gate is 2.6382×10^{-8} W, giving the PADDL cell an improvement of 67.42%. The largest switching peak power of the NAND gate at 13.56MHz is 8.9768×10^{-6} W, and the smallest peak is 1.0433×10^{-6} W, giving a differential power of 7.8355×10^{-6} W. Therefore, the PADDL cell improves upon the differential power by a factor of 67.78.

Figure 7.20. HSPICE Simulation of PADDL Logic Cell at 13.56MHz.

Figure 7.21. Power of NAND Gate (Blue) and Toffoli Gate (Red) at 13.56MHz.

7.7 Comparison to Previous Benchmarks

Here we compare our presented PADDL method with previous benchmarks in mitigation of DPA Attacks, SDMLp [102], RCCDL [103], and WDDL [104]. We reproduced those circuits in 22nm technology using the methods presented in those papers. The average power consumption of each of these methods, as well as the conventional implementation in CMOS, is presented in Table 7.4.

The presented PADDL design is advantageous to the previous designs in average power for each of the fundamental calculations AND, NAND, OR, NOR, XOR and XNOR. PADDL improves upon SDMLp by 76.41%, over RCCDL by 93.98%, and by 89.65% over WDDL. The implementation of SDMLp is the previously best implementation, since it uses evaluate and discharge phases, similar to the SCRL and ECRL methods presented in Section 7.3. Locally, SDMLp is advantageous in terms of required transistors, since implementation of SDMLp requires 16 transistors as opposed to the 32 transistors needed in our proposed implementation. However, this advantage is erased when cascading the cells together. The hardware overhead required ensuring proper timing of evaluation and discharge stages of each cell increases

exponentially as the length of the critical path of the device increases [21]. The PADDL circuit does not require any overhead for maintaining evaluation and discharge phases, making it the better cell for larger implementations, such as DES encryption circuits. However, improving the area of the PADDL device is important. We address this issue in the next section through the use of body biasing in subthreshold operation of the adiabatic dynamic differential logic.

Table 7.4. Benchmarks Average Power Consumption Comparison (10^{-8} W).

Logic	CMOS	WDDL	RCDDL	SDMLp	PADDL
AND	2.9182	6.9751	11.99717	3.705	0.8596
NAND	2.6382	6.4056	11.01763	3.705	0.8596
OR	2.8106	7.2350	12.4442	3.718	0.8596
NOR	3.0702	7.0847	12.18568	3.718	0.8596
XOR	3.3451	11.0587	19.02096	3.508	0.8587
XNOR	3.3451	11.0587	19.02096	3.508	0.8587
Avg	3.0212	8.3029	14.2811	3.643	0.8593
Std Dev	0.2626	1.9653	3.380437	0.0961	0.0004

7.8 A Note on Bijectivity in Secure IC Design

The objective of PADDL is to disassociate the instantaneous power waveform from the output in order to make it more difficult to determine the functionality of the circuit by reading the instantaneous power waveform and the output signals. We should note that the PADDL cell is bijective, so the input signals may be uniquely determined by studying the output signals. In this case, the functionality of the cell can be easily determined by studying the output. This circuit is a 3*3 dual-rail device, so the function may easily be determined by reading 2^3 input signals. However, since the PADDL cell is universal, it may be combined with other PADDL cells in order to generate larger circuits, complicating the effectiveness of this strategy. For example, a 3-input NAND gate would require 7 inputs, since it requires two cascaded PADDL cells, as shown in Fig. 7.22. Therefore, instead of only needing to read 8 outputs, an attacker would have to consider 512 inputs in order to properly ascertain the circuit's functionality.

Furthermore, the triple DES encryption standard uses a cipher key size of 56 bits, meaning that an attacker would have to analyze 7.2057×10^{16} output signals to properly reverse engineer the circuit.

Figure 7.22. Cascaded PADDL Cells with Logic Outputs Shown.

Also, unlike SCRL and ECRL, PADDL does not require additional evaluation and discharge signals to generate the results further in the cascade. Every cell in the other methods requires a unique evaluation and discharge signal. This means that the overhead required to manage the input and output signals is significantly reduced. This is beneficial, since DPA mitigation methods such as SABL have difficulty propagating the signal through the circuit due to signal degradation. The PADDL approach uses the existing signals for evaluation and discharge, which is advantageous over WDDL, RCCDL and SDMLp. Therefore, even though the transistor count is higher in PADDL, the added power required to generate the evaluate and discharge signals in the other methods makes a DPA attack easier.

7.9 Body-Biased Adiabatic Dynamic Differential Logic (BADDL) Cell

In this section, we present a method for implementation of Adiabatic Dynamic Differential Logic at subthreshold operation for ultra-low power implementation. To accomplish this, we use the forward body biasing method shown in Section 7.4 to reduce the average power consumption, differential power consumption, and enables the use of shorter gates. This method is Body-Biased Adiabatic Dynamic Differential Logic (BADDL). The design of BADDL is

essentially the same as the PADDL design presented in Fig. 7.12. The difference is that the nominal voltage is 0.5V, below the threshold voltage of the transistors. We performed our simulation results at 13.56MHz, the standard operating frequency of the ISO/IEC 14443 smart card.

Fig. 7.23 shows the transient analysis performed on the BADDL to determine the proper body biasing for the cell. The top shows the peak power during each of the output combinations. After a body bias of 0.09V, the improvement of the differential power is insignificant, however the magnitudes of the spikes increase linearly. The worst-case fall time for the Q output is shown in the middle of the figure. During certain output combinations, the device struggles to get the signal back to the midpoint due to the NMOS transistors. Body biasing steadily improves this issue. The bottom shows the average power consumption, which reaches a minimum at 0.25V. Therefore, the worst-case switching of the output and average power consumption is optimized when the body bias is set to the midpoint.

Figure 7.23. Transient Analysis of the Impact of Body Biasing on BADDL.

Fig. 7.24 shows the operation of the BADDL device at 13.56MHz. The average power of the BADDL device at 13.56MHz is 9.6093×10^{-10} W; it therefore consumes 6.3394×10^{-16} J of energy every computing cycle. The first, second and third waveforms show the A and A', B and B', and C and C' waveforms, respectively. The fourth waveform represents the Q and Q' output signals. The fifth waveform represents the R and R' output signals. The bottom waveform is the instantaneous power. This is an 88.82% improvement over PADDL, and a 97.4% improvement over SDMLp. The highest switching peak power of the device is 4.4800×10^{-8} W, and the smallest switching peak power is 3.9069×10^{-8} W, giving a differential power of 5.7305×10^{-9} W. This design gains an improvement of 56.64% over PADDL. Therefore, the BADDL cell improves upon the differential power of the conventional NAND gate by a factor of 199.16.

Figure 7.24. HSPICE Simulation of BADDL Logic Cell at 13.56MHz.

Fig. 7.25 shows the improvement in operation of the BADDL output signals by using body biasing. The top waveform in Fig. 7.19 shows the Q and Q' outputs for the BADDL with a

body bias of 0.25V. The second waveform shows these outputs in a PADDL implementation at 0.5V. The outputs for input combinations {001} and {011} are degraded for the non-body biased implementation, whereas they are not degraded for the BADDL. The third and fourth waveforms represent the R and R' outputs for the PADDL and BADDL circuits, respectively. The outputs for input combinations {101} and {111} are degraded for the non-body biased implementation, whereas they are not degraded for the BADDL.

Figure 7.25. Comparison of BADDL Logic Cells With and Without Body Bias.

Since the NMOS transistor is body biased, we were able to reduce the width to 300nm. Additionally, we were able to reduce the transistor length for both the PMOS and NMOS transistors from 50nm to 30nm, which is the minimum transistor length permitted by the model.

The tradeoff in designs for PADDL and BADDL is performance vs. low power. The BADDL design successfully simulates at 13.65 MHz, sufficient for the ISO/IEC 14443 standard. However, the signal quickly begins to degrade after 20MHz due to the low supply voltage. The PADDL effectively simulates in frequencies in excess of 100MHz.

7.10 Conclusions

We use forward body biasing to improve the operation of adiabatic CMOS logic structures, as well as peak, average and differential power at a supply voltage of 0.5V and 1MHz. Simulations in HSPICE using 22nm predictive technology shows that an adiabatic body-biased ECRL inverter may earn up to a 96% improvement over a conventional inverter at 0.5V. The body-biased inverter improves upon the differential power of a conventional inverter at the nominal supply voltage by a factor of 16000. This body biasing method was applied to adiabatic Toffoli and Fredkin gates. The designs improved over their non-body biased implementation in all power metrics, as well as improved output signals. The designs improved the differential power over conventional NAND and MUX gates by factors of 128.43 and 74.27, respectively. These results suggest forward body-biased adiabatic logic is a promising design method for ultra-low power application. We propose an Adiabatic Dynamic Differential Logic design methodology for mitigation of DPA attacks on secure integrated chips. We designed and simulated two universal cells. The first design is a High Performance Adiabatic Dynamic Differential Logic (PADDL), which is optimized for very high operating frequencies. This design improves upon previously presented benchmarks [105] by 76.41% for average power due to a reduced reliance of evaluation and discharge networks. The PADDL cell also improved upon the differential power of a conventional NAND gate by a factor of 112. The second design, Body-Biased Adiabatic Dynamic Differential Logic (BADDL), uses forward body biasing to improve the switching time and differential power of ultra-low power. As a proof of concept, body biasing on adiabatic inverters was shown to improve upon the differential power of a conventional inverter by a factor of 16000. This method was implemented in BADDL, and simulation results show that the differential power was improved upon by a factor of 199.16.

CHAPTER 8

DUAL-RAIL ADIABATIC S-BOX FOR DPA MITIGATION

Design of low power cryptographic hardware systems require built-in protection mechanisms to mitigate side channel attacks¹. Recent research in dual-rail adiabatic logic in CMOS has shown potential for reduction of differential power. In this paper, propose dual-rail adiabatic logic as an advantageous design methodology for lower frequency circuits where security is the primary design objective. We present an adiabatic S-box which significantly reduces energy imbalance compared to previous benchmarks. The design is capable of forward encryption and reverse decryption with minimal overhead, allowing for efficient hardware reuse.

8.1 Introduction

Encryption protects transmitted data from modification or corruption by using checksums and hash functions in order to authenticate users, facilitate non-repudiation, and maintain confidentiality. Encryption disguises plain text by scrambling its content into a cipher text by substituting the characters through mapping the plain text to another element, and then rearranging the elements via transformation. Devices that use encryption standards such as AES are the target of fraud and theft due to their proliferation and their desired content. In particular, smart cards that meet the ISO 14443 standard [5][6] are susceptible to side-channel attacks, which are based on correlations of leaked secondary information and the IC output signals. In RFID devices, these include electromagnetic emanations (EM leakage) [7], measuring the

¹ Portions of this chapter have been accepted for publication to the IEEE Transactions on Computer Aided Design [293]. © 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

amount of time required to perform private-key operations [8], and analysis of noisy power consumption [9]. One of the most effective attacks is a Differential Power Analysis (DPA) attack [9][10], where the attacker analyzes the power consumption in the IC and compares it to the IC's output signals. These attacks are effective, since most modern computing technology is CMOS-based, and the power consumption tendencies of these devices are well studied. Reducing the power consumption of the circuit makes a DPA attack more difficult. Additionally, the primary obstacle in the design of effective, secure hardware devices - such as Smart Cards and medical devices - is overcoming limited power and area resources. RFID devices are powered by the incoming signal, so power masking methods which consume more power are not ideal. In Implantable Medical Devices, replacing the battery requires invasive surgery [39].

Recent research in dual-rail adiabatic logic has shown that the reversible logic design paradigm is promising for mitigation of DPA attacks by significantly reducing the differential power consumption of the circuit [11][12]. By minimizing bit erasure in the circuit, and control current flow through the circuit in order to minimize energy dissipation due to switching, the peaks in power are more consistent.

In this chapter, we propose that dual-rail adiabatic logic is an advantageous design methodology for low power circuits where security is the primary design metric at low frequency such as the 13.56MHz used in the ISO 14443 Smart Card standard. In Section 8.2, we review the motivation for our proposed architecture. In Section 8.3, we present an efficient dual-rail adiabatic implementation of the S-box architecture for mitigation of DPA attacks on the Rijndael algorithm. The novelty of the presented S-box is that it is the first to incorporate dual-rail methodologies for masking power supply, adiabatic logic for average power reduction, and pass transistor logic for reduced area. This methodology allows for a bijective, efficient S-box that is

also physically reversible, permitting design reuse for both encryption and decryption with minimal overhead, reducing the area trade-off that is normally incurred when using dual-rail adiabatic logic. We demonstrate the approach with ISim simulations from our Verilog design, and HSpice waveforms showing the power balance in each input combination. In Section 8.4, we compare the design to previous benchmarks using a variety of technology files in order to provide the best possible comparison.

8.2 Adiabatic S-Box Implementation

The ByteSub portion of the Rijndael algorithm involves using a 16x16 lookup table to determine the output. The first 4-bits correspond to the row of the table, and the next 4 bits correspond to the column of the S-box table. The values of the table are bijective, meaning that the input value may be uniquely determined by knowing the output. And since the result is an 8 bit output, we may construct a fully logically and physically reversible S-box using adiabatic CMOS circuits. We synthesized our S-box and generated the circuit in Verilog. The Verilog code was compiled using the Xilinx IDE Design Suite 13.4, with the Virtex-5 XC5VLX110T device. Figs. 8.2 and 8.3 were generated using the Isim simulation tool. The left side of the figure represented the forward operation. Signals A-Hb represent the 8-bit inputs and their dual-rail implementation, and signals P-Wb are the output values and their dual rail results. In the right side of the figure, the operation is reversed, so the previous outputs are now inputs, and vice versa. The result is that the input values are derivable by placing the output signal at the original output. This shows that the circuit is both logically and physically bijective. The Verilog circuit was imported in Design Compiler in order to generate the HSpice file for all the technology files used in Table 8.1.

Our presented design requires 9612 transistors for forward encryption for the S-Box. Compared to most benchmarks, this is significantly higher – upwards of 56 percent in some cases. However, an advantage of the use of dual-rail adiabatic design in this approach is that the physical reversibility allows for the same circuit to be used for both encryption and decryption. Most Rijndael implementations require forward and reverse S-box hardware implementations. Our method requires only one additional control signal, and 32 Fredkin gates to control the flow through the device. The key may go through the reverse S-box in the same order, reducing software overhead, and the S-box may be used for encryption and decryption, reducing hardware overhead. Therefore, using only 256 additional transistors and one control signal, we are able to perform decryption by reversing the flow through the circuit.

In Figs. 8.1 and 8.2, we present a verification that the synthesized dual-rail adiabatic SBox is functionally correct, and operates at 13.56MHz, the standard frequency for smart card designs for the ISO 14443 standard. We used Design Compiler to generate the HSpice netlist, and simulated the S-box to perform power analysis. Fig. 8.3 shows the instantaneous power waveform for the S-Box in HSpice at 13.56 MHz in 90nm technology. Fig. 8.4 is a closer look at the waveform for 16 random inputs to show the effectiveness of the design method.

The shown Figs. 8.5-8.54 are similar simulations all the way from 0.35 μ m down to 22nm at a variety of frequencies to show that the proposed method scales across technologies.

Figure 8.1. Isim Simulation of Forward Dual-Rail Adiabatic Sbox.

Figure 8.2. Isim Simulation of Reverse Dual-Rail Adiabatic Sbox.

Figure 8.3. Power Waveform for All 256 Inputs at 1MHz and 0.35 μ m.

Figure 8.4. Power Waveform for 16 Random Inputs at 1MHz and 0.35 μ m.

Figure 8.5. Power Waveform for All 256 Inputs at 13.56MHz and 0.35 μ m.

Figure 8.6. Power Waveform for 16 Random Inputs at 13.56MHz and 0.35 μ m.

Figure 8.7. Power Waveform for All 256 Inputs at 50MHz and 0.35 μ m.

Figure 8.8. Power Waveform for 16 Random Inputs at 50MHz and 0.35 μ m.

Figure 8.9. Power Waveform for All 256 Inputs at 10MHz and 0.25 μ m.

Figure 8.10. Power Waveform for 16 Random Inputs at 10MHz and 0.25 μ m.

Figure 8.11. Power Waveform for All 256 Inputs at 13.56MHz and 0.25 μ m.

Figure 8.12. Power Waveform for 16 Random Inputs at 13.56MHz and 0.25 μ m.

Figure 8.13. Power Waveform for All 256 Inputs at 0.1MHz and 0.18 μ m.

Figure 8.14. Power Waveform for 16 Random Inputs at 0.1MHz and 0.18 μ m.

Figure 8.15. Power Waveform for All 256 Inputs at 1.25MHz and 0.18 μ m.

Figure 8.16. Power Waveform for 16 Random Inputs at 1.25MHz and 0.18 μ m.

Figure 8.17. Power Waveform for All 256 Inputs at 13.56MHz and 0.18 μ m.

Figure 8.18. Power Waveform for 16 Random Inputs at 13.56MHz and 0.18 μ m.

Figure 8.19. Power Waveform for All 256 Inputs at 50MHz and 0.18 μ m.

Figure 8.20. Power Waveform for 16 Random Inputs at 50MHz and 0.18 μ m.

Figure 8.21. Power Waveform for All 256 Inputs at 1MHz and 0.13 μ m.

Figure 8.22. Power Waveform for 16 Random Inputs at 1MHz and 0.13 μ m.

Figure 8.23. Power Waveform for All 256 Inputs at 13.56MHz and 0.13 μ m.

Figure 8.24. Power Waveform for 16 Random Inputs at 13.56MHz and 0.13 μ m.

Figure 8.25. Power Waveform for All 256 Inputs at 1MHz and 90nm.

Figure 8.26. Power Waveform for 16 Random Inputs at 1MHz and 90nm.

Figure 8.27. Power Waveform for All 256 Inputs at 13.56MHz and 90nm.

Figure 8.28. Power Waveform for 16 Random Inputs at 13.56MHz and 90nm.

Figure 8.29. Power Waveform for All 256 Inputs at 50MHz and 90nm.

Figure 8.30. Power Waveform for 16 Random Inputs at 50MHz and 90nm.

Figure 8.31. Power Waveform for All 256 Inputs at 1MHz and 65nm.

Figure 8.32. Power Waveform for 16 Random Inputs at 1MHz and 65nm.

Figure 8.33. Power Waveform for All 256 Inputs at 13.56MHz and 65nm.

Figure 8.34. Power Waveform for 16 Random Inputs at 13.56MHz and 65nm.

Figure 8.35. Power Waveform for All 256 Inputs at 50MHz and 65nm.

Figure 8.36. Power Waveform for 16 Random Inputs at 50MHz and 65nm.

Figure 8.37. Power Waveform for All 256 Inputs at 1MHz and 45nm.

Figure 8.38. Power Waveform for 16 Random Inputs at 1MHz and 45nm.

Figure 8.39. Power Waveform for All 256 Inputs at 13.56MHz and 45nm.

Figure 8.40. Power Waveform for 16 Random Inputs at 13.56MHz and 45nm.

Figure 8.41. Power Waveform for All 256 Inputs at 50MHz and 45nm.

Figure 8.42. Power Waveform for 16 Random Inputs at 50MHz and 45nm.

Figure 8.43. Power Waveform for All 256 Inputs at 1MHz and 32nm.

Figure 8.44. Power Waveform for 16 Random Inputs at 1MHz and 32nm.

Figure 8.45. Power Waveform for All 256 Inputs at 13.56MHz and 32nm.

Figure 8.46. Power Waveform for 16 Random Inputs at 13.56MHz and 32nm.

Figure 8.47. Power Waveform for All 256 Inputs at 50MHz and 32nm.

Figure 8.48. Power Waveform for 16 Random Inputs at 50MHz and 32nm.

Figure 8.49. Power Waveform for All 256 Inputs at 1MHz and 22nm.

Figure 8.50. Power Waveform for 16 Random Inputs at 1MHz and 22nm.

Figure 8.51. Power Waveform for All 256 Inputs at 13.56MHz and 22nm.

Figure 8.52. Power Waveform for 16 Random Inputs at 13.56MHz and 22nm.

Figure 8.53. Power Waveform for All 256 Inputs at 50MHz and 22nm.

Figure 8.54. Power Waveform for 16 Random Inputs at 50MHz and 22nm.

8.3 Benchmark Comparison

Since many of the presented works in AES Rijndael design use different implementation technology sizes, a comparison of energy consumption and implementation size is not necessarily fair, since a 22nm low power technology model, such as the model we used to get our results, will inherently produce lower power consumption and area than a cipher using a 0.35 μ m technology model, since the supply voltage will be significantly lower. Transistor count, wire count, and energy imbalance, a metric presented in [22], provides a much more fair comparison across technology files. Given two input combinations e_1 and e_2 , the variation between the power consumption (energy imbalance) of the circuit during those time frames may be found using the equation: $\left| \frac{e_1 - e_2}{e_1 + e_2} \right| * 100\%$.

In comparing the previous benchmarks, we used a variety of technology files and operating frequencies in order to provide a pair and thorough comparison. Using Design Compiler, we were able to generate HSpice files optimized for power and area using TSMC models for 0.35 μ m and 0.25 μ m, IBM files for 0.18 μ m, 0.13 μ m, and 90nm, and PTM files for 65nm, 45nm, 32nm, and 22nm technologies at a variety of frequencies. In addition to providing a fair comparison to previous work, we synthesized the S-box in multiple technologies to show

that the proposed approach is effective as technology is scaled. Security for implantable medical devices must demonstrate long term effectiveness, as attacks only get stronger with Moore’s law [40]. Table 8.2 shows the comparison of implementation technology, supply voltage, single or dual-rail, transistors required for encryption and decryption of the S-box, the total area, the average power consumption, and the energy imbalance. The improvement of our proposed method over the previous best case for energy imbalance, the CSSAL in [35], is 41% at 50MHz.

The design is advantageous in terms of worst-case EI and P_{avg} for all benchmarks. In most cases, the encryption area is significantly higher, as we expected. The physical reversibility of our proposed structure significantly reduces the area trade-off when the area overhead required for S-box decryption is also considered. This is especially advantageous over the CSSAL, which is not physically bijective, resulting in a significant transistor overhead for decryption.

8.4 Conclusions

We propose an Adiabatic Dynamic Differential Logic design methodology for mitigation of DPA attacks on secure integrated chips. We present a dual-rail adiabatic S-box for implementation in the Rijndael cipher for the AES algorithm for low power and low frequency applications, such as Smart Cards at 13.56 MHz. We synthesized and simulated our proposed circuit in numerous implementation technologies in order to provide a fair and accurate comparison between previously synthesized benchmarks. Our circuit improves the upon energy imbalance over previous work by an average of 65 percent, which makes our circuit an effective mitigant of DPA attacks. The tradeoff is operating frequency. The initial tradeoff in area is offset by using the reversible property of the dual-rail adiabatic circuit to allow for design reuse for both encryption and decryption, which is not physically possible in all the previous benchmarks.

Table 8.1. Comparison of S-Box Benchmarks.

Benchmark	Single/ Dual	Tech	Vdd (V)	Freq (MHz)	Xstr Encry	Xsts Decryption	Total Xstr	Area μm^2	Pavg (μW)	Worst-Case EI
[343] (Open Core)	Single	0.35 μm	3.3	333	3180	3092	6272	44593	14000	57%
[343] (comp)	Single	0.35 μm	3.3	333	2362	2242	4606	101364	32100	52%
[343] Open core	Dual	0.35 μm	3.3	333	6672	6432	13104	32975	9812	36%
[343] (Comp)	Dual	0.35 μm	3.3	333	4628	4442	9072	68603	8906	33%
<i>Proposed</i>	<i>Dual</i>	<i>0.35μm</i>	<i>3.3</i>	<i>50</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>25663</i>	<i>1037</i>	<i>1.51%</i>
<i>Proposed</i>	<i>Dual</i>	<i>0.35μm</i>	<i>3.3</i>	<i>13.56</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>25663</i>	<i>446.2</i>	<i>1.87%</i>
<i>Proposed</i>	<i>Dual</i>	<i>0.35μm</i>	<i>3.3</i>	<i>1</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>25663</i>	<i>6.59</i>	<i>0.736%</i>
[344] (No Masking)	Dual	0.25 μm	2.5	13.56	n/a	n/a	n/a	n/a	165	50%
[344] (Masking)	Dual	0.25 μm	2.5	13.56	n/a	n/a	n/a	n/a	210	24%
[353]	Single	0.25 μm	2.5	10	4880	2976	7856	8560	122	63%
[354]	Single	0.25 μm	0.8	10	4880	2976	7856	8560	6.5	47%
[354]	Single	0.25 μm	2.5	10	5516	3316	8880	8150	416	51%
[354]	Single	0.25 μm	2.5	10	10340	9956	20296	15282	232	36%
<i>Proposed</i>	<i>Dual</i>	<i>0.25μm</i>	<i>2.5</i>	<i>13.56</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>6874.2</i>	<i>143.7</i>	<i>4.13%</i>
<i>Proposed</i>	<i>Dual</i>	<i>0.25μm</i>	<i>2.5</i>	<i>10</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>6874.2</i>	<i>106.2</i>	<i>3.97%</i>
[349] (LUT)	Single	0.18 μm	1.8	0.1	1205 gates	1361 gates	2566 gates	n/a	1.85	n/a
[348] (unified)	Single	0.18 μm	1.8	0.1	1287 gates	1181 gates	2468 gates	n/a	1.85	n/a
[348] (Bitslice)	Single	0.18 μm	1.8	0.1	1177 gates	1192 gates	2369 gates	n/a	1.85	n/a
[348] (masked LUT)	Single	0.18 μm	1.8	0.1	2373 gates	2083 gates	4456 gates	n/a	2.34	n/a
[349] (masked unify)	Single	0.18 μm	1.8	0.1	2190 gates	2190 gates	4380 gates	n/a	2.33	n/a
[349]	Single	0.18 μm	1.8	0.1	1570 gates	1622 gates	3192 gates	n/a	5	n/a
[350] (Round-based)	Single	0.18 μm	1.8	0.1	1650 gates	1542 gates	3192 gates	n/a	3.86	n/a
[350] (Serialized)	Single	0.18 μm	1.8	0.1	1075 gates	1104 gates	2179 gates	n/a	2.52	n/a
[346]	Single	0.18 μm	1.8	0.1	195 gates	180 gates	375 gates	5179	1.75	8.2%
[347]	Single	0.18 μm	1.8	0.1	135 gates	122 gates	257 gates	4740	1.64	19.1%
<i>Proposed</i>	<i>Dual</i>	<i>0.18μm</i>	<i>1.8</i>	<i>0.1</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>3474</i>	<i>0.342</i>	<i>0.159%</i>
[356] (CSSAL)	Dual	0.18 μm	1.8	1.25	8115	8063	16178	n/a	4.187	4.9%
[356] (SyAL)	Single	0.18 μm	1.8	1.25	6095	5973	12068	n/a	6.4	9.50%
[356] (2N2P)	Single	0.18 μm	1.8	1.25	4176	4176	8352	n/a	8.675	74.74%
[356] (ECRL)	Single	0.18 μm	1.8	1.25	3166	3270	6436	n/a	6.375	65%
<i>Proposed</i>	<i>Dual</i>	<i>0.18μm</i>	<i>1.8</i>	<i>1.25</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>3474</i>	<i>0.69</i>	<i>2.88%</i>
[356] (CSSAL)	Single	0.18 μm	1.8	12.5	8115	8063	16178	n/a	83.375	7.976%
[356] (SyAL)	Single	0.18 μm	1.8	12.5	6095	5973	12068	n/a	166	16.69%
[356] (2N2P)	Single	0.18 μm	1.8	12.5	4176	4176	8352	n/a	235.75	80.89%
[356] (ECRL)	Single	0.18 μm	1.8	12.5	3166	3270	6436	n/a	172.75	78.78%
<i>Proposed</i>	<i>Dual</i>	<i>0.18μm</i>	<i>1.8</i>	<i>12.5</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>3474</i>	<i>26.97</i>	<i>3.30%</i>
<i>Proposed</i>	<i>Dual</i>	<i>0.18μm</i>	<i>1.8</i>	<i>13.56</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>3474</i>	<i>30.48</i>	<i>3.73%</i>

Table 8.1. (Cont.).

Benchmark	Single/ Dual	Tech	Vdd (V)	Freq (MHz)	Xstr Encry	Xsts Decryption	Total Xstr	Area μm^2	Pavg (μW)	Worst-Case EI
[355]	Single	0.18 μm	1.8	10	701 gates	725 gates	1426 gates	n/a	51	n/a
[356] (CSSAL)	Single	0.18 μm	1.8	50	8115	8063	16178	n/a	693.5	4.88%
[356] (SyAL)	Single	0.18 μm	1.8	50	6095	5973	12068	n/a	1517	29%
[32] (2N2P)	Single	0.18 μm	1.8	50	4176	4176	8352	n/a	1376.5	77.76%
[353] (ECRL)	Single	0.18 μm	1.8	50	3166	3270	6436	n/a	1111.5	76%
<i>Proposed</i>	<i>Dual</i>	<i>0.18μm</i>	<i>1.8</i>	<i>50</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>3474</i>	<i>246.7</i>	<i>2.33%</i>
[345] (area)	Single	0.13 μm	1.2	132	3.1kgates	3.0kgates	6.1kgates	n/a	37	n/a
[345] (power)	Single	0.13 μm	1.2	150	3.2kgates	3.3kgates	6.5kgates	n/a	30	n/a
[345] (speed)	Single	0.13 μm	1.2	290	3.9kgates	4.1kgates	5.0kgates	n/a	62	n/a
[345]	Single	0.13 μm	1.5	10	712 gates	725 gates	1437 gates	n/a	29	n/a
<i>Proposed</i>	<i>Dual</i>	<i>0.13μm</i>	<i>1.2</i>	<i>13.56</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>1512.3</i>	<i>11.9</i>	<i>2.48%</i>
<i>Proposed</i>	<i>Dual</i>	<i>0.13μm</i>	<i>1.2</i>	<i>1</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>1512.3</i>	<i>2.4</i>	<i>0.995%</i>
[347]	Single	90nm	1.8	2400	n/a	n/a	n/a	5699	5847	n/a
<i>Proposed</i>	<i>Dual</i>	<i>90nm</i>	<i>1.8</i>	<i>50</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>733.2</i>	<i>200.8</i>	<i>2.85%</i>
<i>Proposed</i>	<i>Dual</i>	<i>90nm</i>	<i>1.8</i>	<i>13.56</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>733.2</i>	<i>35.82</i>	<i>2.54%</i>
<i>Proposed</i>	<i>Dual</i>	<i>90nm</i>	<i>1.8</i>	<i>1</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>733.2</i>	<i>4.9</i>	<i>0.63%</i>
<i>Proposed</i>	<i>Dual</i>	<i>65nm</i>	<i>0.8</i>	<i>50</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>361.9</i>	<i>57.98</i>	<i>3.42%</i>
<i>Proposed</i>	<i>Dual</i>	<i>65nm</i>	<i>0.8</i>	<i>13.56</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>361.9</i>	<i>18.73</i>	<i>4.64%</i>
<i>Proposed</i>	<i>Dual</i>	<i>65nm</i>	<i>0.8</i>	<i>1</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>361.9</i>	<i>2.633</i>	<i>1.21%</i>
<i>Proposed</i>	<i>Dual</i>	<i>45nm</i>	<i>1.0</i>	<i>50</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>145.78</i>	<i>20.53</i>	<i>4.38%</i>
<i>Proposed</i>	<i>Dual</i>	<i>45nm</i>	<i>1.0</i>	<i>13.56</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>145.78</i>	<i>4.926</i>	<i>6.8%</i>
<i>Proposed</i>	<i>Dual</i>	<i>45nm</i>	<i>1.0</i>	<i>1</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>145.78</i>	<i>1.752</i>	<i>4.27%</i>
<i>Proposed</i>	<i>Dual</i>	<i>32nm</i>	<i>0.9</i>	<i>50</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>77.75</i>	<i>18.02</i>	<i>4.78%</i>
<i>Proposed</i>	<i>Dual</i>	<i>32nm</i>	<i>0.9</i>	<i>13.56</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>77.75</i>	<i>4.499</i>	<i>3.74%</i>
<i>Proposed</i>	<i>Dual</i>	<i>32nm</i>	<i>0.9</i>	<i>1</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>77.75</i>	<i>1.033</i>	<i>2.28%</i>
<i>Proposed</i>	<i>Dual</i>	<i>22nm</i>	<i>0.8</i>	<i>50</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>44.54</i>	<i>10.28</i>	<i>19%</i>
<i>Proposed</i>	<i>Dual</i>	<i>22nm</i>	<i>0.8</i>	<i>13.56</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>44.54</i>	<i>3.41</i>	<i>18%</i>
<i>Proposed</i>	<i>Dual</i>	<i>22nm</i>	<i>0.8</i>	<i>1</i>	<i>9612</i>	<i>512</i>	<i>10124</i>	<i>44.54</i>	<i>1.425</i>	<i>12%</i>

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

Quantum mechanics principles govern the physical limitations of computing circuits and systems. Reversible logic is a computing design methodology which uses the bijective nature of quantum reversibility to reduce power consumption in circuits as best as possible. Adiabatic computing is a paradigm which aims to reduce switching energy through the use of a ramp function instead of the faster switching achieved in step functions.

9.1 Conclusions

In Chapter 3, we address two major debates in reversible logic. First, we present an adiabatic source-memory device in CMOS which operates at sub-Landauer operation where the energies of the “0” and “1” states are identical, thus allowing for a symmetric binary memory. Simulation results show the circuit operates with a supply voltage that is 20 times higher and a switching frequency 2.6 times faster than a previously presented RC circuit. This circuit is presented as a case study that switching circuits are not required to consider charge as a state, since the copy operation of the memory device allows for energy recycling. Measurement of the CMOS value does not necessarily result in energy dissipation. Therefore, adiabatic switching mitigates demons in charge based computing by properly modifying the probability state in a local reversible CMOS based structure.

In Chapter 4, we demonstrated that there is a bijection between the feedback-dependent inputs and feedback-producing outputs in a sequential reversible logic device. Then, using entropy equations and principles presented in and the quantum computing principles,

we were able to calculate the total number of possible input states and output states for the initial clock cycle, as well as each subsequent clock cycle. As a result, the device was shown to be bijective at each clock cycle. Then, it was demonstrated that the past and future states of each clock cycle of a sequential reversible logic structure may be uniquely determined. Using Boltzmann's equation, we concluded that a sequential reversible logic structure is physically reversible for every clock cycle. Therefore, we support the view that feedback is permitted in the design reversible logic based sequential computing structures. Using a VHDL Library for the simulation of reversible logic structures with Integrated Qubit gates, we successfully simulated a reversible logic structure with both forwards and backwards implementation, showing that bijectivity is achieved.

In Chapter 5, we present simulation results which show that our VHDL quantum behavioral model based on Integrated Qubit gates is a more robust and efficient method for the design, simulation, and verification of reversible logic structures than previously existing models. It was also shown the use of VHDL in this method allows for the use of a natively robust programming language that allows for concurrency from the built in capabilities of this language. This method offers a new means of design and verification of reversible structures. The ability for reuse of structures allows for ease of design for larger complex structures, while offering assistance in the verification and test phases by asserting the designer of invalid or poor design practices. Additionally, rather than the design of these structures purely at the quantum representation this method provides new means for designing at higher levels of abstractions, in that, the designer can now use higher level block diagrams.

Next, we presented a method of optimizing synthesized reversible logic structures based upon converting a previously synthesized circuit into the Integrated Qubit library, and then

allowing the user to designate the desired logical outputs. Our presented method of replacing gates with their IQ implementation allowed for reduction in delay by replacing two adjacent gates with a smaller implementation that achieves that same logical calculation. This allows us to merge quantum bits in order to reduce cost and delete bits that only contribute to the calculation of garbage outputs. The synthesized circuits are still logically reversible in nature, but are reduced in terms of quantum cost. By implementing parallelism in our design, we successfully implemented this algorithm $O(N)$ time. We demonstrated that the quantum cost of the circuits may be significantly reduced.

In Chapter 6, we presented a novel algorithm for synthesis of adiabatic logic structures in CMOS. The resulting circuits are synthesized and simulated using HSPICE. The synthesis results show that, on average, the proposed algorithm represents an improvement of 36% over the best known reversible designs with the optimized dual-rail cells. Synthesis of a dual-rail adiabatic S-box for the Rijndael cipher show improvement in differential power over single-rail and conventional implementations. These results suggest that dual-rail adiabatic logic is a promising design methodology for circuits where security is this most important design metric, and frequencies are slower, such as the ISO 14443 Smart Card standard.

In Chapter 7, we use forward body biasing to improve the operation of adiabatic CMOS logic structures, as well as peak, average and differential power at a supply voltage of 0.5V and 1MHz. Simulations in HSPICE using 22nm predictive technology shows that an adiabatic body-biased ECRL inverter may earn up to a 96% improvement over a conventional inverter at 0.5V. The body-biased inverter improves upon the differential power of a conventional inverter at the nominal supply voltage by a factor of 16000. This body biasing method was applied to adiabatic Toffoli and Fredkin gates. The designs improved over their non-body biased implementation in

all power metrics, as well as improved output signals. The designs improved the differential power over conventional NAND and MUX gates by factors of 128.43 and 74.27, respectively. These results suggest forward body-biased adiabatic logic is a promising design method for ultra-low power application. We propose an Adiabatic Dynamic Differential Logic design methodology for mitigation of DPA attacks on secure integrated chips. To consider the tradeoff in performance and power consumption, we designed and simulated two universal cells. The first design is a High Performance Adiabatic Dynamic Differential Logic (PADDL), which is optimized for very high operating frequencies. This design improves upon previously presented benchmarks [105] by 76.41% for average power due to a reduced reliance of evaluation and discharge networks. The PADDL cell also improved upon the differential power of a conventional NAND gate by a factor of 112. The second design, Body-Biased Adiabatic Dynamic Differential Logic (BADDL), uses forward body biasing to improve the switching time and differential power of ultra-low power. As a proof of concept, body biasing on adiabatic inverters was shown to improve upon the differential power of a conventional inverter by a factor of 16000. This method was implemented in BADDL, and simulation results show that the differential power was improved upon by a factor of 199.16.

Finally, in Chapter 8, we propose an Adiabatic Dynamic Differential Logic design methodology for mitigation of DPA attacks on secure integrated chips. We present a dual-rail adiabatic S-box for implementation in the Rijndael cipher for the AES algorithm for low power and low frequency applications, such as Smart Cards at 13.56 MHz. We synthesized and simulated our proposed circuit in numerous implementation technologies in order to provide a fair and accurate comparison between previously synthesized benchmarks. Our circuit improves the upon energy imbalance over previous work by an average of 65 percent, which makes our

circuit an effective mitigant of DPA attacks. The most significant tradeoff is operating frequency. The initial tradeoff in area is offset by using the reversible property of the dual-rail adiabatic circuit to allow for design reuse for both encryption and decryption, which is not physically possible in all the previous benchmarks.

9.2 Future Work

My objective for future work is to design a framework for secure reliable cost-effective patient identification in hospital settings using Adiabatic Near Field Communicator circuits. my low power, secure adiabatic chips to make a new type of hospital patient bracelet. My design methodology, which I present in my dissertation, reduces peak and average power consumption. Not only is this beneficial for operation of small chips - like those used in implantable medical devices - but regulating the power spikes makes it more difficult to steal the information. The current bracelets use barcodes, which create two problems. First, they do not store any information. The scanner has to send the bar code to a database, which gets the user information, and then has to be sent to scanner to be read. Both of those transmissions are long range, which create problems for security. Second, the bar code scanners do not work in the dark, so the nurses have to turn on the light every time they wish to update the user information. So, I am proposing using a Near Field Communicator designed with my secure low-power design - such as a Smart Card - that can store the information locally and securely. Then, you don't need the database or the long range transmissions. Also, since the circuits will work at any time, you would not have to wake up the patient every time. One aspect of my proposed project is using the security to store the patient information locally, maintain the 5 rights of medication, and stay within HIPAA laws.

There are many avenues for pursuing research in order to obtain this objective. First, I will pursue research in dual-rail adiabatic memory devices, such as SRAM, DRAM, cache memory hierarchy, NAND/NOR memory, and non-volatile memory. Previous research in adiabatic SRAM shows that using ramp functions with conventional 6T SRAM cells significantly improves the read and write noise margins [184]. Since there has been debate concerning sequential reversible logic, investigation of reversible and adiabatic memory is not very mature. Therefore, I intend to research how implementing memory using dual-rail logic affects implementation size, speed, and noise margins.

Next, using the memory structures and designing an dual-rail adiabatic ALU based on the design presented in my Master's Thesis [358], I will design a reversible MIPS datapath. I compare the power consumption and area to the previously existing "adiabatic" architectures, such as Pendulum [132] and Bob [137]. It will be novel in that I will implement Dynamic Information Flow Tracking similarly to Raksha [129] in order to provide multiple levels of security, both at the transistor and architectural level. I will build upon the presented S-box to design a complete dual-rail Rijndael cipher. Using the synthesis algorithm and hardware reuse, we will propose a design that will be efficient and show significant improvement in differential power consumption compared to previous hardware implementation, and the hardware reuse will incur less software overhead than previous methods.

Hardware security in medical devices has four primary design paradigms: average and differential power consumption, implementation size, protocol design, and location-based security [318]. Location-based security mitigates attacks which aim to track the user. In a medical bracelet, tracking of patients must be limited to the hospital employees who are attending the patient. Once we build the dual-rail adiabatic architecture with the Rijndael cipher,

we will be able to implement the LoPSiL, a location-based policy-specification language [327] which was the first programming language to specify location-dependent security policies for mobile devices. Since mobile devices use the ISO 7816 Smart Card standard, implementing LoPSiL using a dual-rail adiabatic architecture is a promising design goal for achieving a secure reliable cost-effective patient identification device.

REFERENCES

- [1] T. Kato, "On the Adiabatic Theorem of Quantum Mechanics". Journal of the Physical Society of Japan 5 (6): 435–439, 1950.
- [2] R. Clausius, "On a Modified Form of the Second Fundamental Theorem in the Mechanical Theory of Heat," Philosophical Magazine and Journal of Science, vol. 12, iss. 77, pp. 81-98, 1856.
- [3] R. Clausius, "On Several Convenient Forms of the Fundamental Equations of the Mechanical Theory of Heat," Philosophical Magazine and Journal of Science, 1865.
- [4] L. Boltzmann, "On the Relation Between the Second Fundamental Law of the Mechanical Theory of Heat and the Probability Calculus with Respect to the Theorems of Heat Equilibrium," *Wiener Berichte*, 1877.
- [5] E. Schrödinger, "Quantization as a Problem of Proper Values. Part I.", *Annalen der Physik*, vol. 79, 1926.
- [6] R. Feynman, "Space-Time Approach to Quantum Electrodynamics," *Physical Review*, vol. 76, 1949.
- [7] R. Feynman, "Mathematical Formulation of the Quantum Theory of Electromagnetic Interaction," *Physical Review*, vol. 80, 1950.
- [8] A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. s2-42, iss. 1, pp. 230-265, Jan. 1937.
- [9] R. Landauer, "Irreversibility and Heat Generation in the Computational Process," *IBM Journal of Research and Development*, vol. 5, 1961, pp. 183-91.
- [10] R. Landauer and J.A. Swanson, "Frequency Factors in the Thermally Activated Process", *Phys. Rev. Letters* 121, 1961, pp. 1668-1674.
- [11] J.A. Swanson, "Physical versus Logical Coupling in Memory Systems", *IBM Journal*, vol. 4, Iss. 3, Jul. 1960, pp. 305-310.
- [12] C. Bennett, "Logical Reversibility of Computation," *IBM Journal of Research and Development*, vol. 17, 1973, pp. 525-532.
- [13] R.W. Keyes and R. Landauer, "Minimal energy dissipation in logic", *IBM Journal*, Vol. 14, Iss. 2, Mar. 1970, pp. 152-157.

- [14] M. Davis, *Computability and Unsolvability*, McGraw-Hill Book Co., Inc., New York, 1958, pp. 25-26.
- [15] T. Toffoli, "Reversible Computing," Technical Report MIT/LCS/TM-151, 1980.
- [16] R. Feynman, "Simulating Physics with Computers," *International Journal of Theoretical Physics*, 1982.
- [17] D. Deutsch, "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer," *Proceedings of the Royal Society of London*, vol. 400, 1982.
- [18] R. Feynman, "Quantum Mechanical Computers," *Foundations of Physics*, vol. 16, iss. 6, 1986.
- [19] M. Morrison, M. Lewandowski and N. Ranganathan, "Design of a Tree-Based Comparator and Memory Unit Based on a Novel Reversible Logic Gates," *ISVLSI*, 2012, pp. 231-236.
- [20] Mohammadi, M.; Eshghi, M.; Bahrololoom, A.; , "Behavioral model of V and V+ gates to implement the reversible circuits using quantum gates," *TENCON 2008 - 2008 IEEE Region 10 Conference* , vol., no., pp.1-6, 19-21 Nov. 2008.
- [21] M.P. Frank, "Common mistakes in adiabatic logic design and how to avoid them." *Embedded Systems and Applications*, 216-222, 2003.
- [22] Y. Yibin Ye and K. Roy , "Energy recovery circuits using reversible and partially reversible logic," *IEEE Trans. Circuits and Systems I*: vol.43, no.9, pp.769-778, Sep 1996.
- [23] Hisakado, T.; Iketo, H.; Okumura, K.; , "Logically reversible arithmetic circuit using pass-transistor," in *ISCAS*, vol.2, pp. 853-6, 2004.
- [24] S. G. Younis, "Asymptotically Zero Energy Computing Using Split-Level Charge Recovery Logic", Ph.D. Dissertation, Massachusetts Institute of Technology, June 1994.
- [25] Y. Moon and D. K. Jeong, "An efficient charge recovery logic circuit." *IEEE Journal of Solid State Circuits*, 31(4), 1996, 514-522.
- [26] Van Rentergem, Yvan, and Alexis De Vos. "Optimal design of a reversible full adder." *International Journal of Unconventional Computing* 1.4 (2005): 339.
- [27] Kerntopf, P.; , "Synthesis of multipurpose reversible logic gates," *Digital System Design*, 2002. *Proceedings. Euromicro Symposium on* , vol., no., pp. 259- 266, 2002.
- [28] Kerntopf, P. Binary decision diagram based on single and multiple generalized Shannon expansions. In *Proc. 6th International Symposium on Representations and Methodology of Future Computing Technology*, Trier, Germany, March 2003, 183.190.

- [29] Jing Hu; Guangsheng Ma; Gang Feng; , "Efficient Algorithm for Positive-polarity Reed-muller Expansions of reversible circuits," *Microelectronics*, 2006. ICM '06. International Conference on , vol., no., pp.63-66, 16-19 Dec. 2006.
- [30] Zhijin Guan; Xiaolin Qin; Ziming Ge; Yiqing Zhang; , "Reversible Synthesis with Minimum logic function," *Computational Intelligence and Security*, 2006 International Conference on , vol.2, no., pp.968-971, 3-6 Nov. 2006.
- [31] A. Mishchenko and M. Perkowski, "Fast heuristic minimization of exclusive sum-of-products," in *Proc. Int. Workshop Appl. Reed-Muller Expansion Circuit Des.*, Aug. 2001, pp. 242-250.
- [32] V.V. Shende; Prasad, A.K.; Markov, I.L.; Hayes, J.P.; , "Reversible logic circuit synthesis," *Computer Aided Design*, 2002. ICCAD 2002. IEEE/ACM International Conference on , vol., no., pp. 353- 360, 10-14 Nov. 2002.
- [33] V.V. Shende; Prasad, A.K.; Markov, I.L.; Hayes, J.P.; , "Synthesis of reversible logic circuits," *Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on* , vol.22, no.6, pp. 710- 722, June 2003.
- [34] Miller, D.M.; , "Spectral and two-place decomposition techniques in reversible logic," *Circuits and Systems*, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on , vol.2, no., pp. II-493- II-496 vol.2, 4-7 Aug. 2002.
- [35] Maslov, D.; Dueck, G.W.; , "Reversible cascades with minimal garbage," *Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on* , vol.23, no.11, pp. 1497- 1509, Nov. 2004.
- [36] Dueck, G.W.; Maslov, D.; Miller, D.M.; , "Transformation-based synthesis of networks of Toffoli/Fredkin gates," *Electrical and Computer Engineering*, 2003. IEEE CCECE 2003. Canadian Conference on , vol.1, no., pp. 211- 214 vol.1, 4-7 May 2003.
- [37] Maslov, D.; Dueck, G.W.; Miller, D.M.; , "Simplification of Toffoli networks via templates," *Integrated Circuits and Systems Design*, 2003. SBCCI 2003. Proceedings. 16th Symposium on , vol., no., pp. 53- 58, 8-11 Sept. 2003.
- [38] Maslov, D.; Dueck, G.W.; Miller, D.M.; , "Fredkin/Toffoli templates for reversible logic synthesis," *Computer Aided Design*, 2003. ICCAD-2003. International Conference on , vol., no., pp. 256- 261, 9-13 Nov. 2003.
- [39] D. M. Miller and G. W. Dueck. "Spectral techniques for reversible logic synthesis." In *6th International Symposium on Representations and Methodology of Future Computing Technologies*, March 2003.
- [40] Maslov, D.; Dueck, G.W.; Miller, D.M.; , "Toffoli network synthesis with templates," *Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on* , vol.24, no.6, pp. 807- 817, June 2005.

- [41] Maslov, D.; Miller, D.M.; Dueck, G.W.; , "Templates for reversible circuit simplification," *Communications, Computers and signal Processing*, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on , vol., no., pp. 609- 612, 24-26 Aug. 2005.
- [42] Maslov, D.; Dueck, G.W.; Miller, D.M.; , "Synthesis of Fredkin-Toffoli reversible networks," *Very Large Scale Integration (VLSI) Systems*, *IEEE Transactions on* , vol.13, no.6, pp.765-769, June 2005.
- [43] Maslov, D.; Dueck, G.W.; , "Reversible cascades with minimal garbage," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.23, no.11, pp. 1497- 1509, Nov. 2004.
- [44] G. W. Yang, X. Y. Song, W. N. N. Hung, M. Perkowski, "Fast synthesis of exact minimal reversible circuits using group theory", *Proceedings of IEEE ASP-DAC 2005*, Shanghai, China, 2005, Vol. 2, pp. 18-21.
- [45] Ardestani, E.K.; Zamani, M.S.; Sedighi, M.; , "A Fast Transformation-Based Synthesis Algorithm for Reversible Circuits," *Digital System Design Architectures, Methods and Tools*, 2008. DSD '08. 11th EUROMICRO Conference on , vol., no., pp.803-806, 3-5 Sept. 2008.
- [46] Dong Wang; Hanwu Chen; Wanning Zhu; , "Bidirectional matrix-based algorithm for 4-qubit reversible logic circuits synthesis," *Evolutionary Computation (CEC)*, 2010 IEEE Congress on , vol., no., pp.1-5, 18-23 July 2010.
- [47] Mohammadi, M.; Eshghi, M.; Bahrololoom, A.; , "Behavioral model of V and V+ gates to implement the reversible circuits using quantum gates," *TENCON 2008 - 2008 IEEE Region 10 Conference* , vol., no., pp.1-6, 19-21 Nov. 2008.
- [48] Mohammadi, M.; Eshghi, M.; , "Behavioral description of quantum V and V+ gates to design quantum logic circuits," *Systems, Signals and Devices*, 2008. IEEE SSD 2008. 5th International Multi-Conference on , vol., no., pp.1-5, 20-22 July 2008.
- [49] Arabzadeh, M.; Saeedi, M.; Zamani, M.S.; , "Rule-based optimization of reversible circuits," *Design Automation Conference (ASP-DAC)*, 2010 15th Asia and South Pacific , vol., no., pp.849-854, 18-21 Jan. 2010.
- [50] Robert Wille, Daniel Große, Lisa Teuber, Gerhard W. Dueck, Rolf Drechsler: *RevLib: An Online Resource for Reversible Functions and Reversible Circuits*. Int'l Symp. on Multi-Valued Logic, 2008.
- [51] Z. Sasanian, M. Saeedi, M. Sedighi and M. Samani, "A Cycle-Based Synthesis Algorithm for Reversible Logic," *Design Automation Conference*, 2009, pp. 745-750.
- [52] Kerntopf, P.; , "Synthesis of multipurpose reversible logic gates," *Digital System Design*, 2002. *Proceedings. Euromicro Symposium on* , vol., no., pp. 259- 266, 2002.

- [53] Kerntopf, P. Binary decision diagram based on single and multiple generalized Shannon expansions. In Proc. 6th International Symposium on Representations and Methodology of Future Computing Technology, Trier, Germany, March 2003, 183-190.
- [54] Jing Hu; Guangsheng Ma; Gang Feng; , "Efficient Algorithm for Positive-polarity Reed-muller Expansions of reversible circuits," Microelectronics, 2006. ICM '06. International Conference on , vol., no., pp.63-66, 16-19 Dec. 2006.
- [55] Zhijin Guan; Xiaolin Qin; Ziming Ge; Yiqing Zhang; , "Reversible Synthesis with Minimum logic function," Computational Intelligence and Security, 2006 International Conference on , vol.2, no., pp.968-971, 3-6 Nov. 2006.
- [56] A. Mishchenko and M. Perkowski, "Fast heuristic minimization of exclusive sum-of-products," in Proc. Int. Workshop Appl. Reed-Muller Expansion Circuit Des., Aug. 2001, pp. 242-250.
- [57] Jing Hu; Guangsheng Ma; Donghai Li; , "Symbolic Optimization Method for Reversible Circuits," EUROCON, 2007. The International Conference on "Computer as a Tool" , vol., no., pp.521-524, 9-12 Sept. 2007.
- [58] Saeedi, M.; Zamani, M.S.; Sedighi, M.; , "On the Behavior of Substitution-based Reversible Circuit Synthesis Algorithms: Investigation and Improvement," VLSI, 2007. ISVLSI '07. IEEE Computer Society Annual Symposium on , vol., no., pp.428-436, 9-11 March 2007.
- [59] Jing Zhong; Muzio, J.C.; , "Improved Implementation of a Reed-Muller Spectra Based Reversible Synthesis Algorithm," Communications, Computers and Signal Processing, 2007. PacRim 2007. IEEE Pacific Rim Conference on , vol., no., pp.202-205, 22-24 Aug. 2007.
- [60] D. Kole, H. Rahaman, D. Das, and B. Bhattacharya, "Optimal Reversible Logic Circuit Synthesis Based on a Hybrid DFS-BFS Technique," *ISED*, Feb. 2011, pp. 208-212.
- [61] Kerntopf, P.; , "A new heuristic algorithm for reversible logic synthesis," Design Automation Conference, 2004. Proceedings. 41st , vol., no., pp.834-837, 7-11 July 2004.
- [62] Miller, D.M.; Thornton, M.A.; Goodman, D.; , "A Decision Diagram Package for Reversible and Quantum Circuit Simulation," Evolutionary Computation, 2006. CEC 2006. IEEE Congress on , vol., no., pp.2428-2435.
- [63] Wille, R.; Le, H.M.; Dueck, G.W.; Grosse, D.; , "Quantified Synthesis of Reversible Logic," Design, Automation and Test in Europe, 2008. DATE '08 , vol., no., pp.1015-1020, 10-14 March 2008.

- [64] Al-Rabadi, A.N.; , "Reversible logic neural networks," *Neural Networks*, 2004. Proceedings. 2004 IEEE International Joint Conference, vol.4, no., pp. 2677- 2682 vol.4, 25-29 July 2004.
- [65] M. T. Hagan, H. B. Dcmuth, and M. Bealc, *Neural Network Design*, PWS Publishing Company, 1996.
- [66] S . Haykin, *Neural Networks: A Comprehensive Foundation*. Second Edition, Prentice Hall, 1999.
- [67] J. M. Zurada, *Introductions to Artificial Neural Systems*, West Publishing Company. 1992.
- [68] M. Dorigo, *Optimization, learning and natural algorithms*, Ph.D. thesis, Politecnico di Milano, 1995.
- [69] C. Lu, S. Wang, "Reversible Logic Synthesis Based on a Modified Tabulation Method," *ICMT*, 2011, pp. 2541-2544.
- [70] Van Rentergem, Y.; De Vos, A.; De Keyser, K.; , "Using group theory in reversible computing," *Evolutionary Computation*, 2006. *CEC 2006. IEEE Congress on* , vol., no., pp.2397-2404.
- [71] G.Yang, F. Xie, X. Song, W. Hung, M. Perkowski, "A Constructive Algorithm for Reversible Logic Synthesis," *IEEE Congress on Evolutionary Computation*, July 2006, pp. 2416-2421.
- [72] Zhiqiang Li; Hanwu Chen; Baowen Xu; Xiaoyu Song; Xiling Xue; , "An Algorithm for Synthesis of Optimal 3-qubit Reversible Circuits Based on Bit Operation," *Genetic and Evolutionary Computing*, 2008. *WGEC '08. Second International Conference on* , vol., no., pp.455-458, 25-26 Sept. 2008.
- [73] Min Li; Yexin Zheng; Hsiao, M.S.; Chao Huang; , "Reversible logic synthesis through ant colony optimization," *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2010 , vol., no., pp.307-310, 8-12 March 2010.
- [74] Zilic, Z.; Radecka, K.; Kazamiphur, A.; , "Reversible Circuit Technology Mapping from Non-reversible Specifications," *Design, Automation & Test in Europe Conference & Exhibition*, 2007. *DATE '07* , vol., no., pp.1-6, 16-20 April 2007.
- [75] Takahashi, K.; Hirayama, T.; , "Reversible logic synthesis from positive Davio trees of logic functions," *TENCON 2009 - 2009 IEEE Region 10 Conference* , vol., no., pp.1-4, 23-26 Jan. 2009.

- [76] S.-B. Ko, "A new partitioning method for LUT-based FPGAs," in *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on*, vol. 1, 4-7 2003, pp. 103 – 106 vol.1.
- [77] Soeken, M.; Wille, R.; Drechsler, R.; , "Hierarchical synthesis of reversible circuits using positive and negative Davio decomposition," *Design and Test Workshop (IDT), 2010 5th International* , vol., no., pp.143-148, 14-15 Dec. 2010.
- [78] Yexin Zheng; Chao Huang; , "A novel Toffoli network synthesis algorithm for reversible logic," *Design Automation Conference, 2009. ASP-DAC 2009. Asia and South Pacific* , vol., no., pp.739-744, 19-22 Jan. 2009.
- [79] W.V. Quine, "The problem of simplifying truth tables.", *Amer. Math. Monthly*, Vol. 59 , No. 8, pp. 521–531, 1952.
- [80] E. McCluskey, "Minimization of Boolean Functions", *The Be; System Technical Journal*, Vol. 35, pp. 1417-1444, November 1996.
- [81] S. K. Petrick, "On the minimization of Boolean functions", *Proc. Intl. Conf. Information Processing*, Paris:Unesco, pp.422–423, 1959.
- [82] A. Peres, "Reversible Logic and Quantum Computers," *Physical Review*, vol. 32, iss. 6, 1985, pp. 3266-3267.
- [83] P. Gupta, A. Argawal, and N. Jha, "An Algorithm for Synthesis of Reversible Logic Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, iss. 11, Nov. 2006, pp 2317-23.
- [84] M. Haghparast, S. J. Jassbi, K. Navi and O. Hashemipour, "Design of a Novel Reversible Multiplier Circuit using HNG Gate in Nanotechnology", *World Applied Sci. J.*, Vol. 3, , 2008, pp. 974-978.
- [85] R. Landauer, "Irreversibility and Heat Generation in the Computing Process," *IBM J.Res. Develop.*, vol. 5, 1961, pp 183-91.
- [86] K. Likharev and A. Korotkov, "Single-Electron Parametron: Reversible Computation in a Discrete-State System", *Science Magazine*, Vol. 273 no 5276 pp. 763-765, Aug. 1996.
- [87] A. Mishchenko and M. Perkowski, "Logic synthesis of reversible wave cascades," in *Proc. Int. Workshop Logic Synthesis*, Jun. 2002.
- [88] Y. Pang, S. Wang, Z. He, J. Lin, S. Sultana, "Positive Davio-based synthesis algorithm for reversible logic," *ICCD '11*, Oct. 2011, pp. 212-218.
- [89] R. Willie, D. Große, G. Dueck and R. Drechsler, "Reversible Logic Synthesis with Output Permutation," *22nd International Conference on VLSI Design*, 2009, pp. 189-194.

- [90] K. Yakahashi, T. Hirayama, "Reversible Logic Synthesis from Positive Davio Trees of Logic Functions," TENCON 2009, Jan 2009, pp. 1-4.
- [91] Guowu Yang; Xiaoyu Song; Hung, W.N.N.; Perkowski, M.A.; , "Bi-direction synthesis for reversible circuits," VLSI, 2005. Proceedings. IEEE Computer Society Annual Symposium on , vol., no., pp. 14- 19, 11-12 May 2005.
- [92] Attoh-Okine, N.O.; Shen, L.D.; , "Security issues of emerging smart cards fare collection application in mass transit," Vehicle Navigation and Information Systems Conference, pp.523-526, 1995.
- [93] International Standard Organization, "ISO-IEC 7816".
- [94] International Standard Organization, "ISO-IEC 14443".
- [95] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em sidechannel(s)," in CHES '02. London, UK, UK: Springer-Verlag, 2003, pp. 29–45.
- [96] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in CRYPTO '96. London, UK, UK: Springer-Verlag, 1996, pp. 104–113.
- [97] C. Clavier, J.-S. Coron, and N. Dabbous, "Differential power analysis in the presence of hardware countermeasures," in CHES '00. London, UK, UK: Springer-Verlag, 2000, pp. 252–263.
- [98] P. Kocher, "Differential Power Analysis", Springer-Verlag, LNCS 1666, pp. 388-397, Crypto'99, 1999.
- [99] J. Daemen and V. Rijmen, "Resistance Against Implementation Attacks: A Comparative Study of the AES Proposals", in Proc. of the Second Advanced Encryption Standard (AES) Candidate Conf. March 1999.
- [100] S. Chari, C.S. Jutla, J.R. Rao, and P. Rohatgi, "Towards Sound Approaches to Counteract Power-Analysis Attacks", in Proc. of CRYPTO '99, Lecture Notes in Computer Science, vol. 1666, 1999, pp. 398-412.
- [101] K. Tiri, M. Akmal, and I. Verbauwhede, "A dynamic and differential cmos logic with signal independent power consumption to withstand differential power analysis on smart cards", ESSCIRC, 2002, pp. 403–406.
- [102] L. Ramakrishnan, "SDMLp - Secure Differential Multiplexer Logic : Logic Design For DPA Resistant Cryptographic Circuits", M.S. Thesis, University of Cincinnati, 2012.
- [103] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA-resistant ASIC or FPGA implementation," in Proc. DATE, 2004, pp. 246–251.

- [104] V. Sundaresan, S. Rammohan, and R. Vemuri, "Power invariant secure-IC design methodology using reduced complementary dynamic and differential logic," in *Very Large Scale Integration, 2007. VLSI – SoC, 2007. Oct. 2007*, pp. 1–6.
- [105] Ramakrishnan, L.N.; Chakkaravarthy, M.; Manchanda, A.S.; Borowczak, M.; Vemuri, R.; "SDMLp: On the use of complementary Pass transistor Logic for design of DPA resistant circuits," *Hardware-Oriented Security and Trust (HOST)*, 2012.
- [106] Dickinson, A.G.; Denker, J.S., "Adiabatic dynamic logic," *Solid-State Circuits, IEEE Journal of* , vol.30, no.3, pp.311,315, Mar 1995.
- [107] De, V.K.; Meindl, J.D., "A dynamic energy recycling logic family for ultra-low-power gigascale integration (GSI)," *Low Power Electronics and Design, 1996., International Symposium on* , vol., no., pp.371,375, 12-14 Aug 1996.
- [108] Merkle, R.C., "Towards Practical Reversible Logic," *Physics and Computation, 1992. PhysComp '92., Workshop on* , vol., no., pp.227-228, 2-4 Oct 1992.
- [109] Yang, M.M.; Barby, J.A., "A novel fast low voltage dynamic threshold true single phase clocking adiabatic circuit," *Circuits and Systems, 2004. ISCAS '04., vol.2, no., pp.II,289-92 Vol.2, 23-26 May 2004*.
- [110] Monteiro, C.; Takahashi, Y.; Sekine, T., "Resistance against power analysis attacks on adiabatic dynamic and adiabatic differential logics for smart card," *Trans. ISPACS*), 2011, vol., no., pp.1,5.
- [111] A. Bérut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, and E. Lutz, "Experimental verification of Landauer's principle: linking information and thermodynamics", *Nature*, vol. 483 (2012), pp. 187-189.
- [112] Pramstaller, N.; Gurkaynak, F.K.; Haene, S.; Kaeslin, H.; Felber, N.; Fichtner, W.; , "Towards an AES crypto-chip resistant to differential power analysis," *ESSCIRC 2004*. pp. 307- 310, 2004.
- [113] P. Picton. "A Universal Architecture for Multiplevalued Reversible Logic," *MVL Journal*, 5, 2000, pp.27-37.
- [114] Snider, G.L.; Blair, E.P.; Boechler, G.P.; Thorpe, C.C.; Bosler, N.W.; Wohlwend, M.J.; Whitney, J.M.; Lent, C.S.; Orlov, A.O.; , "Minimum energy for computation, theory vs. experiment," *IEEE-NANO*, pp.478-481, 15-18 Aug. 2011.
- [115] S. G. Younis and T. F. Knight, Jr., "Asymptotically zero energy split-level charge recovery logic," in *International Workshop on Low Power Design*, pp.177-182, 1994.

- [116] M. Morrison and N. Ranganathan, "Forward Body Biased Adiabatic Logic for Peak and Average Power Reduction in 22nm CMOS", *IEEE Symposium on Very Large Scale Integration and Design*, 2014.
- [117] Jingyang Li; Yimeng Zhang; Yoshihara, T., "A novel charge recovery logic structure with complementary pass-transistor network," *SoC Design Conference (ISOCC)*, 2012 International , vol., no., pp.17,20, 4-7 Nov. 2012.
- [118] Hokazono, A.; Balasubramanian, S.; Ishimaru, K.; Ishiuchi, H.; Tsu-Jae King Liu; Chenming Hu; , "MOSFET design for forward body biasing scheme," *Electron Device Letters, IEEE* , vol.27, no.5, pp. 387- 389, May 2006.
- [119] Tschanz, J.W.; Kao, J.T.; Narendra, S.G.; Nair, R.; Antoniadis, D.A.; Chandrakasan, A.P.; De, V.; , "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *Solid-State Circuits, IEEE Journal of* , vol.37, no.11, pp. 1396- 1402, Nov 2002.
- [120] Kioi, K.; Kotaki, H.; Kakimoto, S.; Fukushima, T.; Sato, Y., "Forward body-bias MOS (FBMOS) dual rail logic using an adiabatic charging technique with sub -0.6 V operation," *Electronics Letters* , vol.33, no.14, pp.1200,1201, 3 Jul 1997.
- [121] Khatir, M.; Ejlali, A., "A Body Biasing Method for Charge Recovery Circuits: Improving the Energy Efficiency and DPA-Immunity," *VLSI (ISVLSI)*, 2010 IEEE Computer Society Annual Symposium on , vol., no., pp.195,200, 5-7 July 2010.
- [122] Dickinson, A.G.; Denker, J.S., "Adiabatic dynamic logic," *Solid-State Circuits, IEEE Journal of* , vol.30, no.3, pp.311,315, Mar 1995.
- [123] J. Newsome and D. X. Song. Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software. In the *Proceedings of the Network and Distributed System Security Symposium* , San Diego, CA, Feb. 2005.
- [124] J. R. Crandall and F. T. Chong. MINOS: Control Data Attack Prevention Orthogonal to Memory Model. In the *Proceedings of the 37th Intl. Symposium on Microarchitecture*, Portland, OR, Dec. 2004.
- [125] Cowan, Crispin, et al. "StackGuard: Automatic adaptive detection and prevention of buffer-overflow attacks." *Proceedings of the 7th USENIX Security Symposium*. Vol. 81. 1998.
- [126] M. Frantzen and M. Shue y . Stackghost: Hardware facilitated stack protection. In *Proc. 10th USENIX Security Symposium* , Washington, D.C., Aug. 2001.
- [127] S. Chen, J. Xu, N. Nakka, Z. Kalbarczyk, and R. K. Iyer. Defeating Memory Corruption Attacks via Pointer Taintedness Detection. In the *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*, Yokohama, Japan, June 2005.

- [128] J. R. Crandall and F. T. Chong. MINOS: Control Data Attack Prevention Orthogonal to Memory Model. In the Proceedings of the 37th Intl. Symposium on Microarchitecture, Portland, OR, Dec. 2004.
- [129] M. Dalton, H. Kannan, and C. Kozyra, "Raksha: A Flexible Information Flow Architecture for Software Security", ACM SIGARCH Computer Architecture News. Vol. 35. No. 2. ACM, 2007.
- [130] A.L. Ressler, "The design of a conservative logic computer and a graphical simulator", Master's Thesis, MIT Artificial Intelligence Laboratory, 1981.
- [131] J. S. Hall, "A reversible instruction set architecture and algorithms", in Physics and Computation, pp/ 128-134, Nov. 1994.
- [132] Vieri, Carlin James. Pendulum: A reversible computer architecture. Diss. Massachusetts Institute of Technology, 1995.
- [133] Vieri, Carlin, et al. "A fully reversible asymptotically zero energy microprocessor." Power Driven Microarchitecture Workshop. Citeseer. 1998.
- [134] Vasudevan, D.; Goudarzi, Maziar; Popovici, E.; Schellekens, M., "A Reversible MIPS multi-cycle control FSM design," Quality Electronic Design, 2009. ASQED 2009. 1st Asia Symposium on , vol., no., pp.336,342, 15-16 July 2009.
- [135] Vasudevan, D.; Schellekens, M.; Zeinolabedini, N.; Popovici, E., "Prototyping a bidirectional processor design based on reversible principles," Integrated Circuits (ISIC), 2011 13th International Symposium on , vol., no., pp.325,328, 12-14 Dec. 2011.
- [136] Wille, R.; Soeken, M.; Grosse, D.; Schönborn, E.; Drechsler, R., "Designing a RISC CPU in Reversible Logic," Multiple-Valued Logic (ISMVL), 2011 41st IEEE International Symposium on , vol., no., pp.170,175, 23-25 May 2011.
- [137] M.K. Thomsen, H. Axelsen, and R. Glück. "A reversible processor architecture and its reversible logic design." Reversible Computation. Springer Berlin Heidelberg, 2012. 30-42.
- [138] PTM 22nm Low Power HSPICE Model. ptm.asu.edu/modelcard/LP/22nm_LP.pm.
- [139] L. Szilard, "On the Decrease of Entropy in a Thermodynamic System by the Intervention of Intelligent Beings," Z. Phys. 53:840, 1929.
- [140] V. V. Zhirnov, R. K. Cavin, J. A. Hutchby, and G. I. Bourianoff, "Limits to binary logic switch scaling: a gedanken model," *Proceedings of the IEEE*, vol. 91, no. 11, pp. 1934–1939, 2003.

- [141] K. Cavin, V. V. Zhirnov, J. A. Hutchby, and G. I. Bourianoff, "Energy Barriers, Demons, and Minimum Energy Operation of Electronic Devices", *Fluct. Noise Lett.*, Vol. 5, 2005.
- [142] P. McGeer, J. Sanghavi, R. Brayton, and A. Sangiovanni-Vincentelli, "ESPRESSO-SIGNATURES: A New Exact Minimizer for Logic Functions", *Proceedings of the Design Automation Conference*, pp. 618-621, 1993.
- [143] V.V. Zhirnov and R. K. Cavin, "Comment on Fundamental limits of energy dissipation in charge-based computing", *Applied Physics Letters*, Vol. 98, 2010.
- [144] G.P. Boechler, J. M. Whitney, C. S. Lent, A. O. Orlov, and G. L. Snider, Response to "Comment on 'Fundamental limits of energy dissipation in charge-based computing'", *Appl. Phys. Lett.* 98, 2011.
- [145] G.P. Boechler, J.M. Whitney, C.S. Lent, A.O. Orlov, and G.L. Snider, "Fundamental Limits of energy dissipation in charge-based computing", *Applied Physics Letters*, Vol. 97, 2010.
- [146] P. Heydari, S. Abbaspour, and M. Pedram, "A comprehensive study of energy dissipation in lossy transmission lines driven by CMOS inverters", *Proc. IEEE Custom Integrated Circuits Conf.*, 2002.
- [147] R.C. Merkle, "Towards Practical Reversible Logic," *Physics and Computation, 1992. PhysComp '92., Workshop on* , vol., no., pp.227-228, 2-4 Oct 1992.
- [148] R.C. Merkle, "Helical Logic", *Nanotechnology*, 1992.
- [149] D. Deustch, "Quantum Computational Networks," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* , vol. 425, iss. 1868, 1989, pp. 73-90.
- [150] W. Porod, R. O. Grondin, D. K. Ferry, and G. Porod, "Dissipation in Computation", *Phys. Rev. Lett.* Vol. 52, 1984, pp. 232-235.
- [151] W. Porod, R. Grondin, D. K. Ferry, and G. Porod, "Porod et. al Respond", *Phys. Rev. Lett.* Vol. 53, 1984, pg. 1206.
- [152] A. Peres, *Quantum Theory: Concepts and Methods*, (Kluwer, 1993), Chap. 8.6.
- [153] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.

- [154] M. Perkowski et al. "A general decomposition for reversible logic." *Proc. RM 1* (2001): 119-138.
- [155] A. N. Al Ribaldi, *From Fundamentals to Quantum Computing*, Springer, 2004.
- [156] P. Gupta, A. Argawal, and N. Jha, "An Algorithm for Synthesis of Reversible Logic Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, iss. 11, Nov. 2006, pp 2317-23.
- [157] V.Kamalakannan, P. Rao, V. Patil, "Analysis of Reversible Binary Adder/Subtractor Circuit", *Int'l J. Advanced and Innovative Research*, Vol. 1, Iss. 4, Sept. 2012.
- [158] H. Thapliyal and M. B. Srinivas, "A Beginning in the Reversible Logic Synthesis of Sequential Circuits," *Proceedings of Military and Aerospace Programmable Logic Devices (MAPLD) International Conference*, 2005.
- [159] Thapliyal, H. and Ranganathan, N. 2010. Design of reversible sequential circuits optimizing quantumcosts, delay, and garbage outputs. *ACM J. Emerg. Technol. Comput. Syst.* 6, 4, Article 14 (December 2010).
- [160] E. Fredkin and T. Toffoli, "Conservative Logic," *International Journal of Theoretical Physics*, vol. 21, iss. 3-4, Apr. 1982, pp 219-53.
- [161] Hari, Siva Kumar Sastry, et al. "Efficient building blocks for reversible sequential circuit design." *Circuits and Systems*, 2006. MWSCAS'06. 49th IEEE International Midwest Symposium on. Vol. 1. IEEE, 2006.
- [162] Mohammadi, M.; Eshghi, M.; Haghparast, M., "On design of multiple-valued sequential reversible circuits for nanotechnology based systems," *TENCON 2008 - 2008 IEEE Region 10 Conference* , vol., no., pp.1,6, 19-21 Nov. 2008.
- [163] M. Morrison and N. Ranganathan, "Analysis of Reversible Logic Based Sequential Computing Structures Using Quantum Mechanics Principles," *ISVLSI*, 2012, pp. 219-224.
- [164] M. Morrison, M. Lewandowski, R. Meana and N. Ranganathan, "Design of Static and Dynamic RAM Arrays Using a Novel Reversible Logic Gate and Decoder," *To Appear in the IEEE 11th International Conference on Nanotechnology*, 2011.

- [165] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 3 ed., Boston: Addison Wesley, 2005, pp. 715-738.
- [166] D.P. DiVincenzo, "Two-bit gates are universal for quantum computation", *Phys. Review A*, Vol. 51, No. 2, Feb. 1995, pp. 1015-1022.
- [167] D. P. DiVincenzo and J. A. Smolin,. "Results on two-bit gate design for quantum computers". Proceedings of the Workshop on Physics and Computation, PhysComp '94.
- [168] M. Lewandowski, N. Ranganathan, and M. Morrison, "Behavioral Modeling of Integrated Qubit Gates in Quantum Reversible Logic Structures", *To Appear in ISVSLI 2013*, Aug. 2013.
- [169] B. Gergel and J. E. Rice, "Reversible Logic Synthesis Example Using a Transformation Based Algorithm," ULSI Workshop, 2005.
- [170] W. C. Athas, L. "J." Svensson, J. G. Koller, N. Tzartzanis, and E. Chou, "A framework for practical low-power digital CMOS systems using adiabatic-switching principles," in Proc. Int. Workshop on Low Power Design, Napa Valley, CA, 1994, pp. 189-194.
- [171] C.H. Bennett, "Thermodynamics of Computation: A Review", *Int'l Journal of Theoretical Physics*, Vol 21, 1982, pp. 905-940.
- [172] F. Bloch, "Nuclear induction", *Phys Rev.*, Vol. 70, Iss 7, pp. 460.
- [173] M. Born and V. A. Fock (1928). "Beweis des Adiabatenatzes". *Zeitschrift für Physik A* 51 (3–4): 165–180.
- [174] S. Chiwande, and P. Dakhole, "VLSI design of power efficient Carry Skip Adder using TSG & Fredkin reversible gate," *Devices, Circuits and Systems (ICDCS)*, 2012 International Conference on , vol., no., pp.370-373, 15-16 March 2012.
- [175] M. Davis, *Computability and Unsolvability*, McGraw-Hill Book Co., Inc., New York, 1958, pp. 25-29.
- [176] J. S. Denker, S. C. Avery, A. G. Dickinson, A. Kramer, and T. R. Wik, "Adiabatic computing with the 2N-2N2D logic family," in Proc. Int. Workshop on Low Power Design, Napa Valley, CA, 1994, pp. 183-187.

- [177] E.A. Desloge, Statistical Physics, Holt, Rinehart and Winston, New York, 1986.
- [178] A. Hokazono; Balasubramanian, S.; Ishimaru, K.; Ishiuchi, H.; Tsu-Jae King Liu; Chenming Hu; , "MOSFET design for forward body biasing scheme," Electron Device Letters, IEEE , vol.27, no.5, pp. 387- 389, May 2006.
- [179] J. Huang, X. Ma, and F. Lombardi, "Energy Analysis of QCA Circuits for Reversible Computing," Nanotechnology, 2006. IEEE-NANO 2006. Sixth IEEE Conference on , vol.1, no., pp. 39- 42, 17-20 June 2006.
- [180] K. Likharev and A. Korotkov, "Single-Electron Parametron: Reversible Computation in a Discrete-State System", Science Magazine, Vol. 273 no 5276 pp. 763-765, Aug. 1996.
- [181] J.C. Maxwell, Theory of Heat, Elibron Classics, New York: Dover, 2001.
- [182] M. Morrison, M. Lewandowski, and N. Ranganathan," Design of a Tree-Based Comparator and Memory Unit Based on a Novel Reversible Logic Structures," ISVLSI, 2012, pp. 231-236.
- [183] S. Nakata et al. "Increase in Read Noise Margin of Single-Bit-Line SRAM Using Adiabatic Change of Word Line Voltage," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2013.
- [184] A. Orlov, C.S. Lent, C. Thorpe, G. Boechler, and G.L. Snider, "Experimental Test of Landauer's Principle at the Sub-kT Level", Japanese Journal of Applied Physics, June 2012.
- [185] C. L. Seitz, A.H. Frey, S. Mattisson, S.D. Rabin, D.A. Speck, J.A. van de Snepscheut, "Hot-clock NMOS", In Proc. of the 1985 Chapel Hill Conference on VLSI, 1985.
- [186] T. Sleator, J. Smolin, and H. Weinfurter, "Elementary gates for quantum computation", Phys Rev. A, Vol. 53, 1995, pp. 3457-3467.
- [187] N. Tzartzanis and W. Walker, "A Reversible Poly-Phase Distributed VCO," Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International , vol., no., pp.2452-2461, 6-9 Feb. 2006.
- [188] D. Willingham and I. Kale, "Using Positive Feedback Adiabatic Logic to Implement Reversible Toffoli Gates," Proceedings of the 26th NORCHIP Conference, 2008.

- [189] “Intel 22nm 3-D Tri-Gate Transistor Technology”, <http://newsroom.intel.com/docs/DOC-2032>.
- [190] “Intel’s Revolutionary 22 nm Transistor Technology” http://download.intel.com/newsroom/kits/22nm/pdfs/22nm-Details_Presentation.pdf.
- [191] W.V. Quine, "The problem of simplifying truth tables.", Amer. Math. Monthly, Vol. 59 , No. 8, pp. 521–531, 1952.
- [192] E. McCluskey, “Minimization of Boolean Functions”, The Bell System Technical Journal, Vol. 35, pp. 1417-1444, November 1956.
- [193] S. K. Petrick, "On the minimization of Boolean functions", Proc. Intl. Conf. Information Processing, Paris:Unesco, pp.422–423, 1959.
- [194] Garcia, J.C.; Montiel-Nelson, J.A.; Nooshabadi, S., "A CMOS adiabatic inverter operating with a single clock power supply to reduce non-adiabatic loss," Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on , vol., no., pp.968,971, Nov. 30 2008-Dec. 3 2008
- [195] Weiqiang Zhang; Li Su; Jinghong Fu; Jianping Hu, "A power-gating scheme for CAL circuits using single-phase power-clock," Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on , vol., no., pp.846,849, Nov. 30 2008-Dec. 3 2008.
- [196] Reddy, N. S S; Satyam, M.; Kishore, K. L., "Cascadable adiabatic logic circuits for low-power applications," Circuits, Devices & Systems, IET , vol.2, no.6, pp.518,526, December 2008.
- [197] Weiqiang Zhang; Dong Zhou; Xuanyan Hu; Jianping Hu, "The implementations of adiabatic flip-flops and sequential circuits with power-gating schemes," Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on , vol., no., pp.767,770, 10-13 Aug. 2008.
- [198] Reddy, N. S S; Satyam, M.; Kishore, K.L., "Minimization of energy dissipation in glitch free and cascadable adiabatic logic circuits," TENCON 2008 - 2008 IEEE Region 10 Conference , vol., no., pp.1,5, 19-21 Nov. 2008.

- [199] Takahashi, Y.; Sekine, T.; Yokoyama, M., "Theoretical analysis of power clock generator based on the switched capacitor regulator for adiabatic CMOS logic," Micro-Nanoelectronics, Technology and Applications, 2008. EAMTA 2008. Argentine School of , vol., no., pp.17,22, 18-19 Sept. 2008.
- [200] Anuar, N.; Takahashi, Y.; Sekine, T., "Fundamental logics based on two phase clocked adiabatic static CMOS logic," Electronics, Circuits, and Systems, 2009. ICECS 2009. 16th IEEE International Conference on , vol., no., pp.503,506, 13-16 Dec. 2009.
- [201] Garcia, J.C.; Montiel-Nelson, J.A.; Nooshabadi, S.; Sosa, J.; Navarro, H., "Bootstrapped Adiabatic Complementary Pass-Transistor Logic Driver Circuit for Large Capacitive Load and Low-energy Applications," Digital System Design, Architectures, Methods and Tools, 2009. DSD '09. 12th Euromicro Conference on , vol., no., pp.196,199, 27-29 Aug. 2009.
- [202] Linfeng Li; Jianping Hu; Lili Yu, "Low-Power Register File Using N-type and P-type Adiabatic Logic Circuits," Circuits, Communications and Systems, 2009. PACCS '09. Pacific-Asia Conference on , vol., no., pp.342,345, 16-17 May 2009.
- [203] Teichmann, P.; Fischer, J.; Schmitt-Landsiedel, D., "A robust synchronized 2N2P LC oscillator with a shut-down mode for adiabatic logic circuits," Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on , vol., no., pp.241,244, 24-27 May 2009.
- [204] Haiyan Ni; Jianping Hu, "Single-phase adiabatic flip-flops and sequential circuits with power-gating scheme," ASIC, 2009. ASICON '09. IEEE 8th International Conference on , vol., no., pp.879,882, 20-23 Oct. 2009.
- [205] Chanda, M.; Dandapat, A.; Rahaman, H., "Low-power sequential circuit using single phase Adiabatic Dynamic Logic," Computers and Devices for Communication, 2009. CODEC 2009. 4th International Conference on , vol., no., pp.1,4, 14-16 Dec. 2009.
- [206] Garcia, J.C.; Montiel-Nelson, J.A.; Nooshabadi, S., "Analysis and comparison of high performance CMOS adiabatic drivers," Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on , vol., no., pp.3146,3149, 24-27 May 2009.
- [207] Samanta, S., "Adiabatic computing: A contemporary review," Computers and Devices for Communication, 2009. CODEC 2009. 4th International Conference on , vol., no., pp.1,4, 14-16 Dec. 2009.

- [208] Jinghong Fu; Jianping Hu; Xiaoyan Luo, "The Implementation of Single-Phase Power-Gating Adiabatic Circuits Using Improved CAL Circuits," Circuits, Communications and Systems, 2009. PACCS '09. Pacific-Asia Conference on , vol., no., pp.334,337, 16-17 May 2009.
- [209] Qingbo Xu; Lifang Ye; Jianping Hu; Lijun Huang, "The Implementation of Low-Power CAM with Fully Adiabatic Driving for Large Node Capacitances," Computer Science and Information Engineering, 2009 WRI World Congress on , vol.3, no., pp.413,417, March 31 2009-April 2 2009.
- [210] Anuar, N.; Takahashi, Y.; Sekine, T., "Two phase clocked adiabatic static CMOS logic," System-on-Chip, 2009. SOC 2009. International Symposium on , vol., no., pp.083,086, 5-7 Oct. 2009.
- [211] Jianping Hu; Binbin Liu; Dong Zhou; Xiaoyan Luo, "Low-Power Adiabatic Pins for Driving Chip Pads," Computer Science and Information Engineering, 2009 WRI World Congress on , vol.3, no., pp.408,412, March 31 2009-April 2 2009.
- [212] Lifang Ye; Jianping Hu; Binbin Liu, "Leakage Reduction of DTGAL Circuits with MTCMOS Power-Gating," Electrical and Control Engineering (ICECE), 2010 International Conference on , vol., no., pp.4706,4709, 25-27 June 2010.
- [213] Li Su; Jianping Hu, "An Adiabatic Single-Phase MTCMOS Scheme for Leakage Reduction in Nano-Scale CMOS Processes," Electrical and Control Engineering (ICECE), 2010 International Conference on , vol., no., pp.3271,3274, 25-27 June 2010.
- [214] Tomita, Y.; Takahashi, Y.; Sekine, T., "Adiabatic array logic," Signals and Electronic Systems (ICSES), 2010 International Conference on , vol., no., pp.269,272, 7-10 Sept. 2010.
- [215] Sana, P.K.; Satyam, M., "An Energy Efficient Secure Logic to Provide Resistance against Differential Power Analysis Attacks," Electronic System Design (ISED), 2010 International Symposium on , vol., no., pp.61,65, 20-22 Dec. 2010.
- [216] Takahashi, Y.; Sekine, T.; Yokoyama, M., "A comparison of adiabatic logic as a countermeasures against power analysis attacks," System Science and Engineering (ICSSE), 2010 International Conference on , vol., no., pp.615,618, 1-3 July 2010.

- [217] Fengna Mei; Pengjun Wang, "Design of Ternary clocked adiabatic shift register," Computer Application and System Modeling (ICCASM), 2010 International Conference on , vol.2, no., pp.V2-641,V2-645, 22-24 Oct. 2010.
- [218] Srivastava, A.; Xu, Y.; Soundararajan, R., "Energy Recovery Techniques for CNT-FET Circuits," Electronic System Design (ISED), 2010 International Symposium on , vol., no., pp.81,84, 20-22 Dec. 2010.
- [219] Xuefei Bai; Lu Huang; Yifei Wang; Xinwei Hu, "Evaluation of DPA Attack Resistance of Transistor-Based Adiabatic Logic Styles," e-Business and Information System Security (EBISS), 2010 2nd International Conference on , vol., no., pp.1,3, 22-23 May 2010.
- [220] Weiqiang Zhang; Yu Zhang; Shi Xuhua; Jianping Hu, "Leakage Reduction of Power-Gating Sequential Circuits Based on Complementary Pass-Transistor Adiabatic Logic Circuits," Innovative Computing & Communication, 2010 Intl Conf on and Information Technology & Ocean Engineering, 2010 Asia-Pacific Conf on (CICC-ITOE) , vol., no., pp.282,285, 30-31 Jan. 2010.
- [221] Rosenbaum, D.; Perkowski, M., "Mapping Binary Functions to a Practical Adiabatic Quantum Computer," Multiple-Valued Logic (ISMVL), 2010 40th IEEE International Symposium on , vol., no., pp.270,275, 26-28 May 2010.
- [222] Maurya, A.K.; Kumar, G., "Energy Efficient Adiabatic Logic for Low Power VLSI Applications," Communication Systems and Network Technologies (CSNT), 2011 International Conference on , vol., no., pp.460,463, 3-5 June 2011.
- [223] Jianping Hu; Yu Zhang, "Adiabatic flip-flops based on CPAL with channel length bias," Electronics, Communications and Control (ICECC), 2011 International Conference on , vol., no., pp.2502,2505, 9-11 Sept. 2011.
- [224] Yadav, R.K.; Rana, A.K.; Chauhan, S.; Ranka, D.; Yadav, K., "Adiabatic technique for energy efficient logic circuits design," Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on , vol., no., pp.776,780, 23-24 March 2011.
- [225] Haiyan Ni; Jianping Hu, "Near-threshold sequential circuits using Improved Clocked Adiabatic Logic in 45nm CMOS processes," Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on , vol., no., pp.1,4, 7-10 Aug. 2011.

- [226] Lukac, M.; Ben Shuai; Kameyama, M.; Miller, D.M., "Information-Preserving Logic Based on Logical Reversibility to Reduce the Memory Data Transfer Bottleneck and Heat Dissipation," Multiple-Valued Logic (ISMVL), 2011 41st IEEE International Symposium on , vol., no., pp.131,138, 23-25 May 2011.
- [227] Burignat, S., "Reversible computation, a quantum-inspired low-consumption viable technology?," Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings (SPA), 2011 , vol., no., pp.1,2, 29-30 Sept. 2011.
- [228] Yadav, R.K.; Rana, A.K.; Chauhan, S.; Ranka, D.; Yadav, K., "Four phase clocking rule for energy efficient digital circuits — An adiabatic concept," Computer and Communication Technology (ICCCT), 2011 2nd International Conference on , vol., no., pp.209,214, 15-17 Sept. 2011.
- [229] Theis, T.N., "Principles of operation for a fast, low-voltage digital switch," Semiconductor Device Research Symposium (ISDRS), 2011 International , vol., no., pp.1,1, 7-9 Dec. 2011.
- [230] Sana, P.K.; Satyam, M., "A low power secure logic style to counteract differential power analysis attacks," VLSI Design, Automation and Test (VLSI-DAT), 2011 International Symposium on , vol., no., pp.1,4, 25-28 April 2011.
- [231] Jingyang Li; Yimeng Zhang; Yoshihara, T., "A novel charge recovery logic structure with complementary pass-transistor network," SoC Design Conference (ISOCC), 2012 International , vol., no., pp.17,20, 4-7 Nov. 2012.
- [232] Bezzam, I.; Krishnan, S.; Mathiazhagan, C., "Low power SoCs with resonant dynamic logic using inductors for energy recovery," VLSI and System-on-Chip (VLSI-SoC), 2012 IEEE/IFIP 20th International Conference on , vol., no., pp.307,310, 7-10 Oct. 2012.
- [233] Inoue, K.; Takeuchi, N.; Ehara, K.; Yamanashi, Y.; Yoshikawa, N., "Simulation and Experimental Demonstration of Logic Circuits Using an Ultra-Low-Power Adiabatic Quantum-Flux-Parametron," Applied Superconductivity, IEEE Transactions on , vol.23, no.3, pp.1301105,1301105, June 2013.
- [234] Gebotys, C.H.; Zhang, Y., "Security wrappers and power analysis for SoC technology," Hardware/Software Codesign and System Synthesis, 2003. First IEEE/ACM/IFIP International Conference on , vol., no., pp.162,167, 1-3 Oct. 2003

- [235] Tessier, R.; Jasinski, D.; Maheshwari, A.; Natarajan, A.; Weifeng Xu; Burleson, W., "An energy-aware active smart card," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.13, no.10, pp.1190,1199, Oct. 2005
- [236] Monteiro, Cancio; Takahashi, Yasuhiro; Sekine, Toshikazu, "Low power secure AES S-box using adiabatic logic circuit," Faible Tension Faible Consommation (FTFC), 2013 IEEE , vol., no., pp.1,4, 20-21 June 2013
- [237] Xuefei Bai; Lu Huang; Yifei Wang; Xinwei Hu, "Evaluation of DPA Attack Resistance of Transistor-Based Adiabatic Logic Styles," e-Business and Information System Security (EBISS), 2010 2nd International Conference on , vol., no., pp.1,3, 22-23 May 2010.
- [238] Khazamipour, A.; Radecka, K., "A New Architecture of Adiabatic Reversible Logic Gates," Circuits and Systems, 2006 IEEE North-East Workshop on , vol., no., pp.233,236, 18-21 June 2006.
- [239] King-Keung Mok; Cheong-Fat Chan, "A 13.56 MHz adiabatic smart card / RFID," ASIC, 2007. ASICON '07. 7th International Conference on , vol., no., pp.874,877, 22-25 Oct. 2007.
- [240] Yemiscioglu, G.; Lee, P., "16-Bit Clocked Adiabatic Logic (CAL) logarithmic signal processor," Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on , vol., no., pp.113,116, 5-8 Aug. 2012.
- [241] Garcia, J.C.; Montiel-Nelson, J.A.; Nooshabadi, S.; Sosa, J.; Navarro, H., "Bootstrapped Adiabatic Complementary Pass-Transistor Logic Driver Circuit for Large Capacitive Load and Low-energy Applications," Digital System Design, Architectures, Methods and Tools, 2009. DSD '09. 12th Euromicro Conference on , vol., no., pp.196,199, 27-29 Aug. 2009.
- [242] Kioi, K.; Kakimoto, S.; Sat, Y., "Forward body-bias SRAM circuitry on bulk Si with twin double-well," Electronics Letters , vol.33, no.23, pp.1929,1931, 6 Nov 1997.
- [243] Jiaoyan Chen; Vasudevan, D.; Popovici, E.; Schellekens, M.; Gillen, P., "Design and analysis of a novel 8T SRAM cell for adiabatic and non-adiabatic operations," Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on , vol., no., pp.434,437, 12-15 Dec. 2010.

- [244] Sheng Zhang; Jianping Hu; Dong Zhou, "A low-power adiabatic content-addressable memory," *Circuits and Systems*, 2007. MWSCAS 2007. 50th Midwest Symposium on , vol., no., pp.1285,1288, 5-8 Aug. 2007.
- [245] Avery, S.; Jabri, M., "A three-port adiabatic register file suitable for embedded applications," *Low Power Electronics and Design*, 1998. Proceedings. 1998 International Symposium on , vol., no., pp.288,292, 10-12 Aug. 1998.
- [246] Wang Pengjun; Yu Junjun; Xu Jian, "Design of Clocked Transmission Gate Adiabatic Logic Circuit Based on the 3ECEAC," *Circuits and Systems*, 2006. APCCAS 2006. IEEE Asia Pacific Conference on , vol., no., pp.430,433, 4-7 Dec. 2006.
- [247] Wu Yangbo; Hu Jianping, "Low-Power Content Addressable Memory Using 2N-2N2P Circuits," *Communications, Circuits and Systems Proceedings*, 2006 International Conference on , vol.4, no., pp.2657,2661, 25-28 June 2006.
- [248] Wang Fang; Jia Song; Ji Lijiu, "Design of adiabatic multi-port register file," *ASIC*, 2005. ASICON 2005. 6th International Conference On , vol.1, no., pp.287,290, 24-27 Oct. 2005.
- [249] Jianping Hu; Sheng Zhang; Dong Zhou, "A Low-Power CAM with Fully Adiabatic Driving for Match-Lines and Search-Lines," *Integrated Circuits*, 2007. ISIC '07. International Symposium on , vol., no., pp.321,324, 26-28 Sept. 2007.
- [250] Nakata, Shunji; Suzuki, H.; Honda, R.; Kusumoto, T.; Mutoh, S.; Makino, H.; Miyama, M.; Matsuda, Y., "Adiabatic SRAM with a shared access port using a controlled ground line and step-voltage circuit," *Circuits and Systems (ISCAS)*, Proceedings of 2010 IEEE International Symposium on , vol., no., pp.2474,2477, May 30 2010-June 2 2010.
- [251] Nakata, S.; Kusumoto, T.; Miyama, M.; Matsuda, Y., "Adiabatic SRAM with a large margin of VT variation by controlling the cell-power-line and word-line voltage," *Circuits and Systems*, 2009. ISCAS 2009. IEEE International Symposium on , vol., no., pp.393,396, 24-27 May 2009.
- [252] Wang Fang; Jia Song; Ji Lijiu, "Power Comple ity Analysis of Adiabatic SRAM," *ASIC*, 2005. ASICON 2005. 6th International Conference On , vol.1, no., pp.334,337, 24-0 Oct. 2005.
- [253] Somasekhar, D.; Yibin Ye; Roy, K., "An energy recovery static RAM memory core," *Low Power Electronics*, 1995., IEEE Symposium on , vol., no., pp.62,63, 9-11 Oct. 1995.

- [254] Jamima, H.; Takahashi, Y.; Sekine, T., "Low-power adiabatic SRAM," Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on , vol., no., pp.1,4, 7-9 Dec. 2011.
- [255] Bala, G.J.; Perinbam, J.R.P., "A novel low power 16×16 content addressable memory using PAL," VLSI Design, 2005. 18th International Conference on , vol., no., pp.791,794, 3-7 Jan. 2005.
- [256] Jianping Hu; Weiqiang Zhang; Yinshui Xia, "A novel low-power adiabatic SRAM with an energy-efficient line driver," Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on , vol.2, no., pp.1151,1155 Vol.2, 27-29 June 2004.
- [257] Joohee Kim; Ziesler, C.H.; Papaefthymiou, M.C., "Energy recovering static memory," Low Power Electronics and Design, 2002. ISLPED '02. Proceedings of the 2002 International Symposium on , vol., no., pp.92,97, 2002.
- [258] Samson, M.; Mandavalli, S., "Adiabatic 5T SRAM," Electronic System Design (ISED), 2011 International Symposium on , vol., no., pp.267,272, 19-21 Dec. 2011.
- [259] Jun-Jun Yu; Peng-Jun Wang, "Design of Adiabatic SRAM Based on CTGAL Circuit," Solid-State and Integrated Circuit Technology, 2006. ICSICT '06. 8th International Conference on , vol., no., pp.2118,2120, 23-26 Oct. 2006.
- [260] Jianping Hu; Xu, Tiefeng; Junjun Yu; Yinshui Xia, "Low power dual transmission gate adiabatic logic circuits and design of SRAM," Circuits and Systems, 2004. MWSCAS '04. The 2004 47th Midwest Symposium on , vol.1, no., pp.I,565-8 vol.1, 25-28 July 2004.
- [261] Arabzadeh, M.; Saeedi, M.; Zamani, M.S.; , "Rule-based optimization of reversible circuits," *Design Automation Conference (ASP-DAC), 2010 15th Asia and South Pacific* , vol., no., pp.849-854, 18-21 Jan. 2010.
- [262] A. Barenco, C.H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, "Elementary gates for quantum computation", *Phys Rev. A*, Vol. 53, 1995, pp. 3457-3467.
- [263] Thapliyal, H.; Ranganathan, N., "A new design of the reversible subtractor circuit," Nanotechnology (IEEE-NANO), 2011 11th IEEE Conference on , vol., no., pp.1430,1435, 15-18 Aug. 2011

- [264] Morrison, M.; Ranganathan, N., "Design of a Reversible ALU Based on Novel Programmable Reversible Logic Gate Structures," VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on , vol., no., pp.126,131, 4-6 July 2011
- [265] Kotiyal, S.; Thapliyal, H.; Ranganathan, N., "Design of a reversible bidirectional barrel shifter," Nanotechnology (IEEE-NANO), 2011 11th IEEE Conference on , vol., no., pp.463,468, 15-18 Aug. 2011
- [266] Morrison, M.; Lewandowski, M.; Meana, R.; Ranganathan, N., "Design of a novel reversible ALU using an enhanced carry look- ahead adder," Nanotechnology (IEEE-NANO), 2011 11th IEEE Conference on , vol., no., pp.1436,1440, 15-18 Aug. 2011
- [267] Thapliyal, H.; Ranganathan, N., "Design of Reversible Latches Optimized for Quantum Cost, Delay and Garbage Outputs," VLSI Design, 2010. VLSID '10. 23rd International Conference on , vol., no., pp.235,240, 3-7 Jan. 2010
- [268] Thapliyal, H.; Ranganathan, N.; Ferreira, R., "Design of a comparator tree based on reversible logic," Nanotechnology (IEEE-NANO), 2010 10th IEEE Conference on , vol., no.,
- [269] Kotiyal, S.; Thapliyal, H.; Ranganathan, N., "Mach-Zehnder Interferometer Based All Optical Reversible NOR Gates," VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on , vol., no., pp.207,212, 19-21 Aug. 2012
- [270] Thapliyal, H.; Ranganathan, N., "Tutorial T2: Reversible Logic: Fundamentals and Applications in Ultra-Low Power, Fault Testing and Emerging Nanotechnologies, and Challenges in Future," VLSI Design (VLSID), 2012 25th International Conference on , vol., no., pp.13,15, 7-11 Jan. 2012.
- [271] Nachtigal, M.; Ranganathan, N., "Design and analysis of a novel reversible encoder/decoder," Nanotechnology (IEEE-NANO), 2011 11th IEEE Conference on , vol., no., pp.1543,1546, 15-18 Aug. 2011.
- [272] Thapliyal, H.; Ranganathan, N.; Kotiyal, S., "Design of Testable Reversible Sequential Circuits," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.21, no.7, pp.1201,1209, July 2013.
- [273] Thapliyal, H.; Ranganathan, N., "Design of Efficient Reversible Binary Subtractors Based on a New Reversible Gate," VLSI, 2009. ISVLSI '09. IEEE Computer Society Annual Symposium on , vol., no., pp.229,234, 13-15 May 2009

- [274] Nachtigal, M.; Thapliyal, H.; Ranganathan, N., "Design of a reversible single precision floating point multiplier based on operand decomposition," Nanotechnology (IEEE-NANO), 2010 10th IEEE Conference on , vol., no., pp.233,237, 17-20 Aug. 2010
- [275] Morrison, M.; Ranganathan, N., "Design of a Moore finite state machine using a novel reversible logic gate, decoder and synchronous up-counter," Nanotechnology (IEEE-NANO), 2011 11th IEEE Conference on , vol., no., pp.1445,1449, 15-18 Aug. 2011
- [276] Filanovsky, I.M.; Sánchez-Sinencio, E.; Ramírez-Angulo, J.; Thapliyal, H.; Ranganathan, N.; Soderstrand, M.A.; Silva-Martinez, J.; Yun Chiu; Gopalan, K.; Acharya, V.; Saxena, V.; Salama, K.N., "[Seven tutorials]," Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on.
- [277] Standard 1076-1993, "IEEE Standard VHDL Language Reference Manual," *IEEE*, p. 249, 1993.
- [278] P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring", *Proc. of 35th Foundations of Computer Science*, Nov 1994, pp. 124-134.
- [279] F. Bloch, "Nuclear induction", *Phys Rev.*, Vol. 70, Iss 7, pp. 460.
- [280] Martin Lukac, Mihail Pivtoraiko, Alan Mishchenko, Marek Perkowski, "Automated Synthesis of Generalized Reversible Cascades Using Genetic Algorithms" *Proc. Fifth Intern. Workshop on Boolean Problems*, 33-45.
- [281] David Edward Goldberg "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989, ISBN 0201157675 Freiberg, Sachsen, Germany, September 19-20.
- [282] Thapliyal, H.; Ranganathan, N., "A new reversible design of BCD adder," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011 , vol., no., pp.1,4, 14-18 March 2011
- [283] Thapliyal, H.; Ranganathan, N., "Reversible Logic-Based Concurrently Testable Latches for Molecular QCA," *Nanotechnology, IEEE Transactions on* , vol.9, no.1, pp.62,69, Jan. 2010.
- [284] Thapliyal, H.; Ranganathan, N., "Concurrently testable FPGA design for molecular QCA using conservative reversible logic gate," *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on* , vol., no., pp.1815,1818, 24-27 May 2009

- [285] Ramaswamy, V.N.; Namuduri, K.R.; Ranganathan, N., "Context modeling of wavelet coefficients in EZW-based lossless image coding," *Acoustics, Speech, and Signal Processing*, 1999. Proceedings., 1999 IEEE International Conference on , vol.6, no., pp.3165,3168 vol.6, 15-19 Mar 1999.
- [286] Ramaswamy, V. N.; Namuduri, K.R.; Ranganathan, N., "Lossless image coding using embedded zerotree wavelet framework: Part I. EZW coding," *Signals, Systems & Computers*, 1998. Conference Record of the Thirty-Second Asilomar Conference on , vol.1, no., pp.145,148 vol.1, 1-4 Nov. 1998.
- [287] Kotiyal, S.; Thapliyal, H.; Ranganathan, N., "Design of A ternary barrel shifter using multiple-valued reversible logic," *Nanotechnology (IEEE-NANO)*, 2010 10th IEEE Conference on , vol., no., pp.1104,1108, 17-20 Aug. 2010.
- [288] Thapliyal, H.; Ranganathan, N., "Testable Reversible Latches for Molecular QCA," *Nanotechnology*, 2008. NANO '08. 8th IEEE Conference on , vol., no., pp.699,702, 18-21 Aug. 2008.
- [289] Thapliyal, H.; Ranganathan, N., "Design, Synthesis and Test of Reversible Circuits for Emerging Nanotechnologies," *VLSI (ISVLSI)*, 2012 IEEE Computer Society Annual Symposium on , vol., no., pp.5,6, 19-21 Aug. 2012.
- [290] Kotiyal, S.; Thapliyal, H.; Ranganathan, N., "Mach-Zehnder interferometer based design of all optical reversible binary adder," *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2012 , vol., no., pp.721,726, 12-16 March 2012.
- [291] Nachtigal, M.; Thapliyal, H.; Ranganathan, N., "Design of a reversible floating-point adder architecture," *Nanotechnology (IEEE-NANO)*, 2011 11th IEEE Conference on , vol., no., pp.451,456, 15-18 Aug. 2011.
- [292] Thapliyal, H.; Ranganathan, N., "Reversible logic based concurrent error detection methodology for emerging nanocircuits," *Nanotechnology (IEEE-NANO)*, 2010 10th IEEE Conference on , vol., no., pp.217,222, 17-20 Aug. 2010.
- [293] M. Morrison and N. Ranganathan, "Synthesis and Optimization of Dual Rail Adiabatic Logic Using A Novel Parallel Algorithm", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2013.

- [294] M. Morrison, N. Ranganathan, and J. Ligatti, "Design of Adiabatic Dynamic Differential Logic for DPA-Resistant Secure Integrated Circuits", Submitted to IEEE Transactions on Very Large Scale Integration, 2013.
- [295] M. Morrison and N. Ranganathan, "Issues in Reversible Computing: Mitigating Demons in Charge Based Computing, and Permissibility of Reversible Sequential Logic", Submitted to Submitted to ACM Journal on Emerging Technology Special Issue on Reversible Computing, 2013.
- [296] M. Morrison and N. Ranganathan, "Design of an Integrated-Qubit Based Quantum Reversible ALU Based On Novel Reversible Logic Structures", Submitted to IEEE Transactions on Circuits and Systems I, 2013.
- [297] W. Rankl and W. Effing, *Smart Card Handbook*, 2nd Edition, John Wiley & Sons, Chichester, England, 2000.
- [298] http://worldwide.espacenet.com/publicationDetails/biblio?DB=EPODOC&II=86&ND=3&adjacent=true&locale=en_EP&FT=D&date=19720718&CC=US&NR=3678250A&KC=A.
- [299] http://worldwide.espacenet.com/publicationDetails/originalDocument?FT=D&date=19730523&DB=EPODOC&locale=en_EP&CC=GB&NR=1317915A&KC=A&ND=4.
- [300] http://worldwide.espacenet.com/publicationDetails/biblio?DB=EPODOC&II=80&ND=3&adjacent=true&locale=en_EP&FT=D&date=19730531&CC=GB&NR=1318850A&KC=A.
- [301] Shelfer, K. The Intersection of Knowledge Management and Competitive Intelligence: Smart Cards and Electronic Commerce. Knowledge Management For The Information Professional. Information Today, Inc. Medford, New Jersey. 1999.
- [302] <http://www.archpatent.com/patents/4092524>.
- [303] http://worldwide.espacenet.com/publicationDetails/originalDocument?FT=D&date=19780808&DB=EPODOC&locale=en_EP&CC=US&NR=4105156A&KC=A&ND=4.
- [304] <http://news.google.com/newspapers?nid=861&dat=20001223&id=dLZIAAAIBAJ&sjid=jYEMAAAIBAJ&pg=6102,6779167>.
- [305] http://www.researchgate.net/publication/228314623_The_Internationalization_of_Electronic_Purse_Networks_National_Responsiveness_vs._Benefits_of_Integration/file/d912f5100ef19d2b84.pdf.
- [306] https://www.emvco.com/best_practices.aspx?id=217.

- [307] <http://www.tfn.net/techno/smartcards/cardspec.pdf>.
- [308] Dominique Brule, Mattias Eld, Jerome Juvin, Christian Lackner. MIFARE4Mobile™ enabling electronic ticketing and access management applications on mobile devices. MIFARE4 Mobile Industry Group, 2012.
- [309] "4th Asian Transport Revenue Collection Forum - Overview". Asia Pacific Smart Card Association.
- [310] ISO/IEC 7816– Integrated Circuit Cards with Contacts – Part I: Physical Characteristics http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=29257.
- [311] http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816-2.aspx.
- [312] http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816-3.aspx.
- [313] http://wg8.de/wg8n1496_17n3613_Ballot_FCD14443-3.pdf.
- [314] http://wg8.de/wg8n1344_17n3269_Ballot_FCD14443-4.pdf.
- [315] Ahmed Patel, Kenan Kalajdzic, Laleh Golafshan, Mona Taghavi (2011). "Design and Implementation of a Zero-Knowledge Authentication Framework for Java Card". International Journal of Information Security and Privacy 5 (3). IGI. pp. 1–18.
- [316] Junwei He; Liji Wu; Xiangmin Zhang, "Design and implementation of a low Power Java Coprocessor for dual-interface IC Bank Card," *ASIC (ASICON), 2011 IEEE 9th International Conference on* , vol., no., pp.965,969, 25-28 Oct. 2011.
- [317] Junfeng Fan; Reparaz, O.; Rozic, V.; Verbauwhede, I., "Low-energy encryption for medical devices: Security adds an extra design dimension," Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE , vol., no., pp.1,6, May 29 2013-June 7 2013.
- [318] Messerges, T.S.; Dabbish, E.A.; Sloan, R.H., "Examining smart-card security under the threat of power analysis attacks," Computers, IEEE Transactions on , vol.51, no.5, pp.541,552, May 2002.
- [319] Di WU, Liji WU, Xiangmin ZHANG. Design and FPGA Implementation of JAVA CARD Coprocessor for EMV Compatible IC Bankcard, 978-1-4244-3870(2009).
- [320] TID (2009), Telefonica I+D, <http://www.tid.es/>.
- [321] W. P. Burleson, S. S. Clark, B. Ransford, and K. Fu. Design Challenges for Secure Implantable Medical Devices. In DAC 2012, June 2012. Invited paper.

- [322] V. Pop, R. de Francisco, H. Pflug, J. Santana, H. Visser, R. J. M. Vullers, H. de Groot, and B. Gyselinckx. Human++: Wireless Autonomous Sensor Technology for Body Area Networks. In ASP-DAC 2011, pages 561–566, 2011.
- [323] <http://www.hhs.gov/ocr/privacy/index.html>.
- [324] P. Kocher, “Differential Power Analysis”, Springer-Verlag, LNCS 1666, pp. 388-397, Crypto'99, 1999.
- [325] R. Peeters and J. Hermans. Wide Strong Private RFID Identification Based on Zero-Knowledge. IACR Cryptology ePrint Archive, 2012:389, 2012.
- [326] Ligatti, Jay, Billy Rickey, and Nalin Saigal. "LoPSiL: A location-based policy-specification language." *Security and Privacy in Mobile Information and Communication Systems*. Springer Berlin Heidelberg, 2009. 265-277.
- [327] K. Simoens, R. Peeters, and B. Preneel. Increased Resilience in Threshold Cryptography: Sharing a Secret with Devices That Cannot Store Shares. In *Pairing 2010*, volume 6487 of LNCS, pages 116-135, 2010.
- [328] M. O'Neill. Low-cost SHA-1 Hash Function Architecture for RFID Tags. In *Workshop on RFID Security - RFIDSec*, pages 41–51, 2008.
- [329] Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede. Elliptic-Curve-Based Security Processor for RFID. *IEEE Trans. Computers*, 57(11):1514–1527, 2008.
- [330] G. J. Milburn, "Quantum Optical Fredkin Gate," *Physical Review Letters*, vol. 62, iss. 18, 1989, pp. 2124-2127.
- [331] A. De Vos: "Reversible computing", Wiley-VCH, Weinheim (2010).
- [332] J. Daemen and V. Rijmen. "Rijndael for AES". In *AES Candidate Conference*, pages 343–348, 2000.
- [333] Saeedi, Mehdi, and Igor L. Markov. "Synthesis and optimization of reversible circuits—a survey." *ACM Computing Surveys (CSUR)* 45.2 (2013): 21.
- [334] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A Compact Rijndael Hardware Architecture with S-Box Optimization,” *ASIACRYPT 2001*, pp. 239-254, 2001.
- [335] Hisakado, T.; Iketo, H.; Okumura, K.; , "Logically reversible arithmetic circuit using pass-transistor," in *ISCAS*, vol.2, pp. 853-6, 2004.
- [336] W. Diffie and M. Hellman. "New directions in cryptography." *Information Theory, IEEE Transactions on* 22.6, 1976, pp. 644-654.

- [337] "NIST Federal Information Processing Standards (FIPS) PUB 197 Advanced Encryption Standard", Nov 2001.
- [338] J. Nechvatal. "Report on the Development of the Advanced Encryption Standard", 2000.
- [339] E. Thiagarajan and M. Gourishetty. "Study of AES and its Efficient Software Implementation".
- [340] H. Kuo and I. Verbauwhede. "Architectural Optimization for a 1.82 Gbits/sec VLSI Implementation of the AES Rijndael Algorithm". In *CHES*, volume 2162, pages 51–64, 2001.
- [341] S. Mangard, M. Aigner, and S. Dominikus. "A Highly Regular and Scalable AES Hardware Architecture". *IEEE Transactions on Computers*, 52(4):483–491, April 2003.
- [342] Sokolov, D., Murphy, J., Bystrov, A., & Yakovlev, A. (2005). Design and analysis of dual-rail circuits for security applications. *Computers, IEEE Transactions on*, 54(4), 449-460.
- [343] H. Saputra et. al, "Masking the Energy Behaviour of DES Encryption," Proc. DATE, 2003.
- [344] P. Hämäläinen, T. Alho, M. Hännikäinen and T. D. Hämäläinen, "Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core," DSD 2006, pp. 577-583, Aug 2006.
- [345] C. Wang, and N.M. Heys, "Using a pipelined S-box in compact AES hardware implementations", NEWCAS, pp.101,104, 20-23 June 2010.
- [346] Mathew, J.; Rahaman, H.; Jabir, A.M.; Mohanty, S.P.; Pradhan, D.K., "On the design of different concurrent EDC schemes for S-Box and GF(p)," Quality Electronic Design (ISQED), 2010 11th International Symposium on , vol., no., pp.211,218, 22-24 March 2010.
- [347] Wei Zhao; Yi Wang; Renfa Li, "A unified architecture for DPA-resistant PRESENT," IIT 2012, pp.244,248, 18-20 March 2012.
- [348] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT - An Ultra-Lightweight Block Cipher," in *CHES* 2007, vol. 4727, Springer-Verlag, 2007, pp. 450-466.
- [349] C. Rolfes, A. Poschmann, G. Leander, and C. Paar, "Security for 1000 Gate Equivalents," in *SECSI* 2008, 2008.

- [350] Zeng, Juanli et al, "Improvement on masked S-box hardware implementation", Innovations in Information Technology (IIT), IEEE International Conference, 2012, pp 113-116.
- [351] Y.K. Lai; Li-Chung Chang; Lien-Fei Chen; Chi-Chung Chou; Chun-Wei Chiu, "A novel memoryless AES cipher architecture for networking applications," Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on , vol.4, no., pp.IV,333-6 Vol.4, 23-26 May 2004.
- [352] Zeng, Yong-hong, et al. "A low-power Rijndael S-Box based on pass transmission gate and composite field arithmetic." *Journal of Zhejiang University SCIENCE A* 8.10 (2007): 1553-1559.
- [353] Wolkerstorfer, J., Oswald, E., Lamberger, M., 2002. An ASIC implementation of the AES S-Boxes. LNCS, 2271:67-78.
- [354] S. Morioka and A. Satoh, "An Optimized S-Box Circuit Architecture for Low Power AES Design", Proceedings Workshop on Cryptographic Hardware and Embedded Systems 2002 (CHES 2002), San Francisco, August 2002. pp. 172-186.
- [355] Monteiro, C.; Takahashi, Y.; Sekine, T., "Low power secure AES S-box using adiabatic logic circuit," *Faible Tension Faible Consommation (FTFC)*, 2013 IEEE , vol., no., pp.1,4, 20-21 June 2013.
- [356] N. Kosaraju, "A VLSI Architecture for Rijndael, the Advanced Encryption Standard", *M.S. Thesis, University of South Florida*, November, 2013.
- [357] M. Morrison, "Design of a Reversible ALU using Novel Reversible Logic Structures", *M.S. Thesis, University of South Florida*, March 2012.
- [358] S. Kotiyal, "Design Methodologies for Reversible Logic Based Barrel Shifters", *M.S. Thesis, University of South Florida*, March 2012.
- [359] M. Morrison and N. Ranganathan, "A novel optimization method for reversible logic circuit minimization," *VLSI (ISVLSI)*, 2013 IEEE Computer Society Annual Symposium on , vol., no., pp.182,187, 5-7 Aug. 2013.

APPENDIX 1 – JOURNAL REPRINT PERMISSIONS

This copyright permission is for a paper published in the IEEE Symposium on Very Large Scale Integration [163]. Portions of this paper appear in Chapter 4.

[Home](#) [Account Info](#) [Help](#)

Requesting permission to reuse content from an IEEE publication

Title: Analysis of Reversible Logic Based Sequential Computing Structures Using Quantum Mechanics Principles

Conference Proceedings: VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on

Author: Morrison, M.; Ranganathan, N.

Publisher: IEEE

Date: 19-21 Aug. 2012

Copyright © 2012, IEEE

Logged in as:
Matthew Morrison

[LOGOUT](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

This copyright permission is for a paper published in the IEEE Symposium on Very Large Scale Integration [359]. Portions of this paper appear in Chapter 5.

[Home](#) [Account Info](#) [Help](#)

Requesting permission to reuse content from an IEEE publication

Title: A novel optimization method for reversible logic circuit minimization

Conference Proceedings: VLSI (ISVLSI), 2013 IEEE Computer Society Annual Symposium on

Author: Morrison, M.; Ranganathan, N.

Publisher: IEEE

Date: 5-7 Aug. 2013

Copyright © 2013, IEEE

Logged in as:
Matthew Morrison

[LOGOUT](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

This copyright permission is for a paper published in the IEEE Conference on Nanotechnology [164]. Portions of this paper appear in Chapter 4.

[Home](#)[Account Info](#)[Help](#)

Requesting permission to reuse content from an IEEE publication

Title: Design of static and dynamic RAM arrays using a novel reversible logic gate and decoder

Conference Proceedings: Nanotechnology (IEEE-NANO), 2011 11th IEEE Conference on

Author: Morrison, M.; Lewandowski, M.; Meana, R.; Ranganathan, N.

Publisher: IEEE

Date: 15-18 Aug. 2011

Copyright © 2011, IEEE

Logged in as:
Matthew Morrison

[LOGOUT](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

This copyright permission is for a paper published in the IEEE Conference on VLSI Design [116]. Portions of this paper appear in Chapter 7.

RightsLink®

Home

Account
Info

Help

Title: Forward Body Biased Adiabatic Logic for Peak and Average Power Reduction in 22nm CMOS

Conference Proceedings: VLSI Design and 2014 13th International Conference on Embedded Systems, 2014 27th International Conference on

Author: Morrison, M.; Ranganathan, N.

Publisher: IEEE

Date: 5-9 Jan. 2014

Logged in as:
Matthew Morrison

LOGOUT

Copyright © 2014, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

This copyright permission is for a paper published in the IEEE Symposium on Very Large Scale Integration [19]. Portions of this paper appear in Chapter 5.

[Home](#)[Account Info](#)[Help](#)

Requesting permission to reuse content from an IEEE publication

Title: Design of a Tree-Based Comparator and Memory Unit Based on a Novel Reversible Logic Structure

Conference Proceedings: VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on

Author: Morrison, M.; Lewandowski, M.; Ranganathan, N.

Publisher: IEEE

Date: 19-21 Aug. 2012

Copyright © 2012, IEEE

Logged in as:
Matthew Morrison

[LOGOUT](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Material accepted for publication into the IEEE Transactions on Computer Aided Design [293] was used in Chapters 6 and 8. The paper is not yet published into IEEEExplore, making it not yet unavailable in the Copyright Clearance Center. I requested permission for use for this dissertation. I was directed to include a footnote, which appears on both the first page of Chapter 6 and the first page of Chapter 8. Below is a screenshot capture of the e-mail exchange where I requested permission, and was informed what needed to be done in this unique circumstance.

ABOUT THE AUTHOR

Matthew Morrison received his Bachelor of Science in Computer Engineering in 2010 from the University of South Florida and his Master of Science degree from USF in 2012. He is currently pursuing his Doctoral degree in Computer Science and Engineering at USF. His research interests include design, synthesis, and analysis of adiabatic and reversible logic circuits for gate and system level power aware design, low power hardware encryption circuits, and design and simulation of emerging technologies. From 1999-2005, he served as a Nuclear Electronics Technician and Reactor Operator in the United States Navy. He has taught several courses in the Computer Science and Engineering department at USF, including the undergraduate and graduate Computer Architecture courses, and Foundations of Engineering. He was the recipient of a 2012 ACM Turing Student Scholar Award. He was also the recipient of the 2013 USF Provost's Award for Outstanding Teaching by a Graduate Teaching Assistant. He is a Graduate Ambassador to the Dean of the Graduate School. He is a student member of the IEEE, the ACM, and the IEEE Computer Society. He has accepted an appointment as an Assistant Professor of Electrical Engineering at the University of Mississippi.