USF Tampa Graduate Theses and Dissertations   USF Graduate Theses and Dissertations

January 2012

Design of a Reversible ALU Based on Novel Reversible Logic Structures

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

Design of a Reversible ALU Based on Novel Reversible Logic Structures

by

Matthew Morrison

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Engineering
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Nagarajan Ranganathan, Ph.D.
Srinivas Katkoori, Ph.D.
Hao Zheng, Ph.D.

Date of Approval:
March 8, 2012

Keywords: Comparator, Emerging Technologies, Low Power, Multiplexer, Quantum
Computing

DEDICATION

To my parents, Dr. Alfred and Kathleen Morrison, who taught me the most important

values in life: work hard, have respect, do what is right, and treat others the way you want

to be treated.

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF TABLES

iii

LIST OF FIGURES

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. Recent advances in reversible logic using and quantum computer algorithms allow for improved computer architecture and arithmetic logic unit designs. In this paper, a 2*2 Swap gate which is a reduced implementation in terms of quantum cost and delay to the previous Swap gate is presented. Next, a novel 3*3 programmable UPG gate capable of calculating the fundamental logic calculations is presented and verified, and its advantages over the Toffoli and Peres gates are discussed. The UPG is then implemented in a reduced design for calculating n-bit AND, n-bit OR and n-bit ZERO calculations. Then, two 3*3 RMUX gates capable of multiplexing two input values with reduced quantum cost and delay compared to the previously existing Fredkin gate is presented and verified. Next, 4*4 reversible gate is presented and verified which is capable of producing the calculations necessary for two-bit comparisons. The UPG and RC are implemented in the design of novel sequential and tree-based comparators. Then, two novel 4*4 reversible logic gates (MRG and PAOG) are proposed with minimal delay, and may be configured to produce a variety of logical calculations on fixed output lines based on programmable select input lines. A 5*5 structure (MG) is proposed that extends the capabilities of both the MRG and PAOG. The comparator designs are verified and its advantages to previous designs are discussed. Then, reversible implementations of ripple-carry, carry-select and Kogge-Stone carry look-ahead adders

are analyzed and compared. Next, implementations of the Kogge-Stone adder with sparsity-4, 8 and 16 were designed, verified and compared. The enhanced sparsity-4 Kogge-Stone adder with ripple-carry adders was selected as the best design, and its implemented in the design of a 32-bit arithmetic logic unit is demonstrated. The proposed ALU design is verified and its advantages over the only existing ALU design are quantitatively analyzed.

CHAPTER 1

INTRODUCTION

Reversible logic is a promising computing design paradigm which presents a method for constructing computers that produce no heat dissipation. Reversible computing emerged as a result of the application of quantum mechanics principles towards the development of a universal computing machine. Specifically, the fundamentals of reversible computing are based on the relationship between entropy, heat transfer between molecules in a system, the probability of a quantum particle occupying a particular state at any given time, and the quantum electrodynamics between electrons when they are in close proximity. 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.

1.1 – Outline of Thesis

In Chapter 2, a literature survey of the relevant works of reversible logic design and arithmetic is presented. The quantum mechanics principles that are fundamental to the principles of reversible computing are reviewed in order to properly define the scope of the argument presented in this paper. First, Clausius' definition of entropy [27] is presented, and the derivation of his equation of entropy of a system with respect to the

rate of heat generation dQ and the temperature function T, and where he defined the

instance where $\int \frac{dQ}{T} = 0$ as a reversible process is reviewed [28]. Boltzmann's work in

probabilistic kinetic gas energy is reviewed in order to demonstrate that the probability

that a particle inhibits a specific state at any given moment directly correlates to the

entropy of the system, and that a reversible system has an identical number of possible

initial states and final states [1]. Plank's application of Boltzmann's work into his

research of black-body radiation is presented in order to demonstrate that energy may

only be emitted in quantized amounts, meaning the integral presented by Clausius must

be finite and definite, allowing for a quantum representation of a reversible system [30].

Einstein's work into the emission and transformation of light is discussed since his

research led him to conclude that relating the statistical probability of a molecule to its

entropy is fully adequate for representing the thermal transformations in a system, and

that that wave function of an observable electron may be represented by a constant

unitary operator in the Hamiltonian space [31]. Feynman's work into quantum

electrodynamics is reviewed in order to show that space-time diagrams used to represent

interactions between electrons are utilized as a method for the calculation of any process

involving energy quanta, and how energy is transferred between quantum particles in a

reversible system [32][33].

The fundamentals of Turing's universal computing machine are presented in order

to provide context to the main ideas of this paper [34]. Then, Landauer's work suggesting

that heat generation per computing cycle is an unavoidable aspect of computing is

reviewed. Landauer presented an equation that determined the minimum amount of heat

generated in a reversible system that correlated the input and output states of the

computing structure to the states of a system [35]. Next, Bennett's response to Landauer, where he presents a specific form of Turing machine, is reviewed in order to show that binary computing structures may be realized in which every clock cycle is logically reversible by creating a bijective device with an identical number of input and output lines [36]. Toffoli and Fredkin's work in conservative logic and reversible computing is reviewed to show the fundamental design concepts of reversible logic structures [36][37]. Next, Deustch's work in quantum computational networks is shown in order to demonstrate how Feynman's research into quantum electrodynamics and Bennett's work in reversible Turing machines may be combined to create a mathematical model of a universal quantum computer [32][33][38].

In Chapter 3, a reduced implementation of the Swap gate is presented and verified. The concept of Programmable Reversible Logic is defined and theorems presented. Two reversible 4*4 gates are presented which may be implemented in a programmable manner such that they meet the input/output and cost requirements demonstrated in Theorems 3.1 and 3.2. Also, a novel 3x3 universal programmable gate (UPG) is presented which can be used in realizing the reversible functions: AND, OR, NAND, and NOR. The UPG is first used in the design of a reduced N-Bit ZERO calculation, which is important for ALU design, since zero calculations are important for determining branch-on-equal results for datapath design. Then, two reversible multiplexer gates are presented with improved cost and delay over the Fredkin gate. Then a 4x4 Reversible Comparator (RC) gate is presented which may be configured in a programmable manner for implementation in a reversible comparator. In Chapter 4, The UPG and RC are implemented in the design of a sequential and tree-based comparator.

These designs are compared in terms of quantum cost and delay to previous existing designs. In Chapter 5, a series of reversible adder designs using carry look ahead, carry select and fast carry adders with various sparsitys. Then, the UPG is used to design minimal reversible Kogge-Stone Cumulate Logic to produce a fast reversible adder that accounts for a good compromise between low cost and low delay. In Chapter 6, the presented reversible carry-select adder, reversible tree-based comparator and reversible multiplexer are implemented in the design of a 32-bit reversible ALU capable of producing ADD, SUB, AND, NAND, OR, NOR, XOR, XNOR, >, <, = and ZERO logical calculations.

CHAPTER 2

RELEVANT WORKS

In this section, we first cover the fundamental physical concepts behind reversible logic. Then, we will cover four specific aspects of reversible logic design. First, we will cover the fundamental reversible logic gates. Next, we will discuss reversible arithmetic, specifically adders and comparators. Next, we will cover the previously proposed ALU in the literature. Finally, we will discuss the two previously presented reversible comparators. Each of the topics presented in the section are shown in the Taxonomy Diagram in Fig. 2.1.
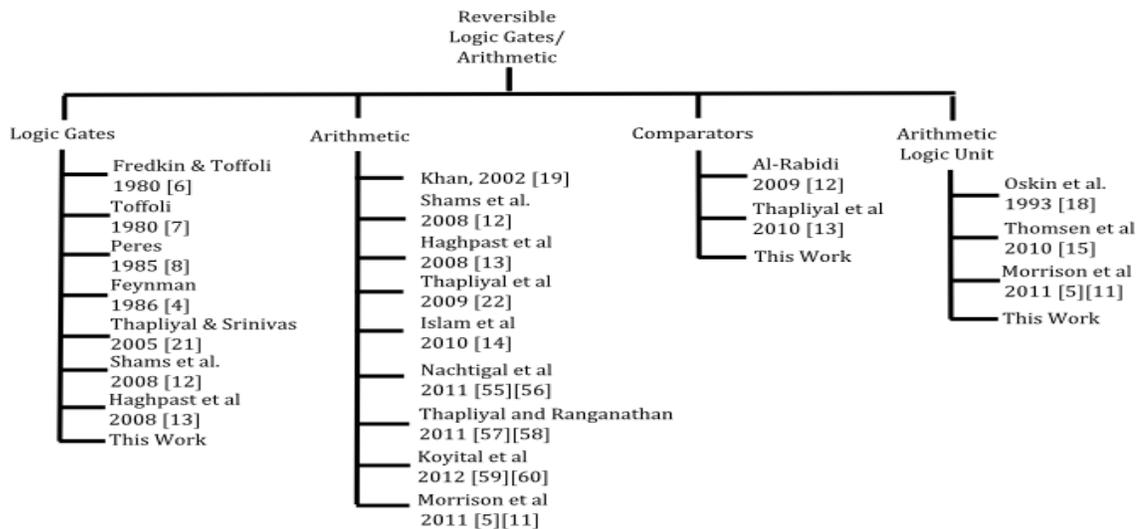
Fig 2.1: Taxonomy Diagram of Relevant Works

2.1 Entropy and the Reversible Process

Clausius demonstrated in [27] that 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. Whenever some quantity of heat Q is converted into work W, another quantity of heat must necessarily be transferred from a warmer to a colder body. The value of Q may be related to the converted work and the equivalent heat per unit of work A, and this relation is shown in (1):

$$Q = U + A * W \qquad (1)$$

The value for U is determined by the initial and final states of the system, W is the work done by the system, and A is the equivalent heat per unit of work. In a cyclical process – meaning that the initial state and final state of a system are identical – U is 0, which reduces the equation to $Q = A * W$.

Clausius then defined the equivalence-value, where two transformations may occur without requiring an additional energy transfer, as Q/T, where T is a function of the temperature. By substituting the temperature function into the equivalence-value, the transfer of temperature between two bodies may be represented by (2):

$$Q \left(\frac{1}{T_2} - \frac{1}{T_1} \right) \qquad (2)$$

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 dQ divided by the temperature function T:

$$\frac{Q_1}{T_1} + \frac{Q_2}{T_2} + \cdots + \frac{Q_N}{T_N} = \Sigma \frac{Q}{T} = \int \frac{dQ}{T} \qquad (3)$$

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 produces the equality

$\int \frac{dQ}{T} \geq 0$. The instance where $\int \frac{dQ}{T} = 0$ is a unique case where the system reaches all of the possible states, thus all of the transformations exactly cancel each other, and is known as a *reversible process*.

In [28], Clausius defined the rate of change of entropy of a system to be equivalent to the rate of change of heat in a system divided by the temperature function. Therefore, the change in entropy in a system is determined by Eqn. (4):

$$S = S_0 + \int \frac{dQ}{T} \qquad (4)$$

The value of S denotes the transformation content of the body, and this value is defined as the *entropy* of the body. In a reversible system, the integral is equal to zero, reducing (4) to $S = S_0$. Resultantly, reversible systems generate zero entropy gain through their transformations.

Boltzmann presented a probabilistic expression for entropy in [1] by defining kinetic energy in the context of kinetic gas theory. The relationship, shown in (5), between the number of particles N of an ideal gas in an isolated system, a volume V, constant energy U is as a function to determine the number of microstates of the particles in the system by relating them to the mass of an atom m and Planck's constant.

$$S = \Omega(U, V, N) = \left[\left(\frac{2\pi m k T}{h^2} \right)^{3/2} \frac{V e^{5/2}}{N^{5/2}} \right] \approx e^N \qquad (5)$$

Boltzmann demonstrated that the entropy of a system was directly proportional the logarithm of the energy, volume and number of particles in the system, as in (5), as well as the gas constant. The entropy is also inversely proportional to Avogadro's

number. The relationship between the gas constant and Avogadro's number, which is the numerical constant representing the proportion between the logarithm of the microstates came to be known as Boltzmann's constant, k. This allowed for the equation in (5) to be reduced to $S = k\ln \Omega(U,V,N)$. Since the natural logarithm of $\Omega(U,V,N)$ is approximately equal to $\ln (e^N)$, this equation may be simplified to (6), where W is the number of possible energy states in the system:

$$S = k \ln (W) \qquad (6)$$

Planck determined in [29] that a given number N of resonators would produce an entropy of $S_N = N * S$, where S is the entropy of a single resonator, and that the total system entropy is found by the logarithm of its probability, $S_N = k_B\ln (W) + U$, supporting Boltzmann's claim. By directly relating the entropy to the number of resonators in the system, he determined that the distribution of energy elements can result only in a finite, integral number, which allowed him to postulate that electromagnetic energy could only be emitted in discrete quantized amounts.

Therefore, Clausius' definition pertaining to the heat given off in a system may be related to Boltzmann's equation, giving the result shown in (7), where $\int \frac{dQ}{T}$ must be a discrete, and finite integral:

$$kln(W_f) = kln(W_o) + \int \frac{dQ}{T} \qquad (7)$$

Resultantly, in a reversible system, since $\int \frac{dQ}{T} = 0$, the number of input states must be equal to the number of output states, since $kln(W_f) = kln(W_o)$, which gives $W_f = W_o$. Therefore, Planck's postulate allows for a quantum representation of a

reversible system when the number of input states is identical to the number of output states.

Einstein demonstrated in [30] the probability that the total radiation energy will be found at a given point at a randomly chosen instant is relative to the monochromatic radiation of frequency v and energy E is enclosed by reflecting walls in a volume v_0. Einstein demonstrated that the change in entropy of a system where the number of particles n and the temperature t remain constant by setting the probability $W_o = v_0{}^n$. Substituting this into Boltzmann's equation, $\int \frac{dQ}{T} = kn * ln\left(\frac{v}{v_0}\right)$. In a system where the volume is constant, the probability that each possible state is reached is identical. This led him to conclude that relating the statistical probability of a particle to its entropy is wholly sufficient for representing the thermal transformations in a system. Therefore, in a reversible system, the energy transferred during each transformation is in finite, quantized packets, and that probability that a particle in the system will reach each particular state is the same for every possible state.

2.2 - Reversible System Representation in Spacetime

In [30], Einstein related the equations for electrodynamic interactions presented by Maxwell [31] to an expression which govern how electrons must move in space and time when a force is applied to them. He used Lorenz transformations, which relate the space-time coordinates (x, y and z for the particle's position in space, and t for its position with respect to time) which have the relationship $x^2 + y^2 + z^2 = c^2t^2$ to the variables (τ, ξ, η, ζ) as shown in equations (8)-(11).

$$\tau = \frac{t - vx/c^2}{\sqrt{1 - v^2/c^2}} \qquad (8)$$

9

$$\xi = \frac{x - vt}{\sqrt{1 - v^2/c^2}} \qquad (9)$$

$$\eta = y \qquad (10)$$

$$\zeta = z \qquad (11)$$

These equations were then related to a rigid sphere R at rest to the relative moving system k, giving the equation $\xi^2 + \eta^2 + \zeta^2 = c^2\tau^2 = R^2$ [30]. Through this analysis, he remarked that the curvature of the path of an electron, when a magnetic force N acts upon it perpendicularly to the velocity of the electron may be related to the radius of the sphere which contains it, which is shown in eqn. (12):

$$R = \frac{mc^2}{\epsilon} * \frac{v/c}{\sqrt{1 - v^2/c^2}} * \frac{1}{N} \qquad (12)$$

Resultingly, he concluded that the speed of light must be constant to all uniformly moving observers in a system, which meant space and time must be related and inextricably intertwined.

Minowski followed in [32] by demonstrating that all quantum interactions may be related by their interactions in space and in time. In Fig 1, the oberserver at the present moment, represented by O, He presented his four-dimensional representation of spacetime as a two-dimensional lightcone diagram, with the horizontal axis representing space and the vertical axis, time (ct). The cone in the figure shows the evolution of an emission of light from the past (prior-cone) to the future (upper cone) through the observer O at the present moment.

Since Einstein demonstrated that all observers in a vacuum must observe the speed of light at a constant velocity, Minkowski showed that the entirity of physical reality must exist within the prior-cone and upper-cone, since one must travel faster than the speed of light to exceed them. Therefore, quantum interactions are representable in the context of spacetime.
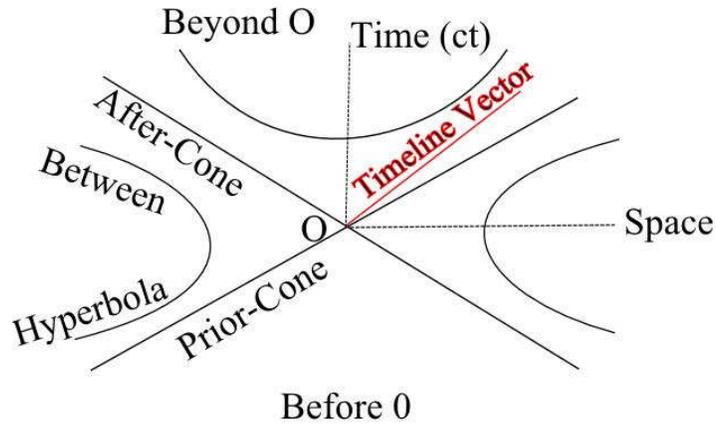
Fig 2.2: An Observer in Spacetime, As Presented By Minowski [34]

Schrödinger developed an equation in [33] which described the manner in which quantum state of a physical system changes with respect to time by expressing the phase of a plane wave as a complex phase factor. Specifically, his equation is a relationship between the momentum of the particle *p*, its wavelength λ, the partial derivative of its wavefunction ψ, its mass *m* and the potential well created by the particle *V*, and is shown in Eqn. 13:

$$i\hbar \frac{\delta}{\delta t} \psi(x, t) = -\frac{\hbar^2}{2m} \nabla^2 \psi(x, t) + V(x) \psi(x, t) \quad (13)$$

Feynman used the equations presented by Schrödinger to present a physical perspective of quantum electrodynamics in [34] by utilizing space-time diagrams to interpret electron interaction when they very close to each other in order to simplify the

matrix elements for complex processes. These perturbative representations of the transitions of quantum corpuscles from one state to another with respect to the corpuscle's amplitude led to a consistent method for the calculation of any process involving transfer of energy between quantum particles [35]. These representations are commonly called Feynman diagrams.

In a Feynman diagram, a state s_i at a point i in space-time is a given function of the points, such as s_j and s_k in some neighborhood of i. The relationship in (8) is dependent on states that are behind i in time, which allows for calculating of the next state in the diagram.

$$s_i = F_i(s_j, .., s_k, ...) \quad (13)$$

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. Based on information in the past, such as the momentum or two states of the positron, it is possible to produce a mathematical formulation of all future states of the positron based on their electromagnetic interactions [35]. Specifically, in a reversible process, a Feynman diagram will show every past configuration and future configuration, and that each possible state will be reached by the quanta.

An example of a Feynman diagram is shown in Fig. 2.3. This interaction represents the fundamental interaction between two electrons, where one election (in this case, the one on the left) gives the other electron a virtual quantum.
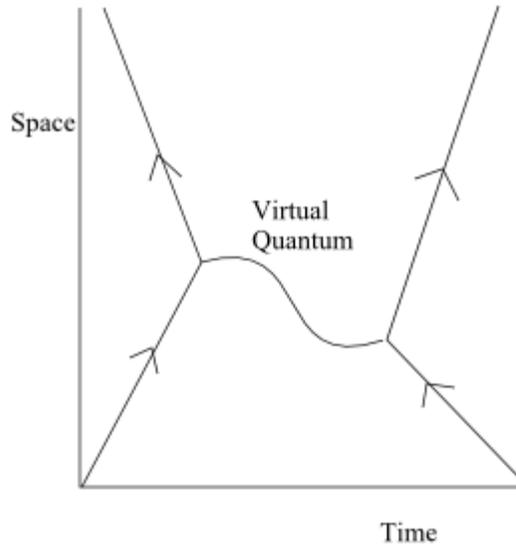
Fig 2.3: Feynman Diagram: A quantum interaction between charges

2.3 Reversibility in a Universal Computing Machine

Turing presented the notion of a *universal computing machine* in [36] as a single automatic machine where its operation is completely determined by its configuration, prints either a 0 or a 1, and is able to compute any computable sequence. Feynman noted in [6] that both natural laws and rules for computing are reversible, which allows for a computer to utilize quantum mechanics in order to produce computations without producing any entropy gain.

Landauer raised the possibility in [2] that it was not possible to design a computer that is physically reversible. He described a simple binary device as consisting of a particle in a bistable potential well, where the particle may either be at state '0' or state '1'. Using the bistable potential well, he described an irreversible operation called "restore to one," where the particle's output state was '1' regardless of the input state, meaning that the bijectivity of the device is lost, and it is not physically reversible.

Since the input states and output states are discrete, and a binary computing device may only reach a state of '0' or '1', then the total number of probabilities for each input line is 2. Therefore, the maximum number of probabilities for N inputs is 2^N. Landauer substituted this quantity for W in Boltzmann's equation (6) to relate these probabilities to the entropy change for each computing cycle. As an example, he presented a three-input, three-output device that had eight possible input states and only four possible output states, as shown in Table 2.1.

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, shown in (14)-(16).

Table 2.1
Truth Table Presented by Landauer

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

$$k(\sum p_{out} * ln(p_{out}) - \sum p_{in} * ln(p_{in})) \qquad (14)$$

$$k\left((2 * \frac{3}{8} * ln\frac{3}{8} + 2 * \frac{1}{8} * ln\frac{1}{8}) - (8 * \frac{1}{8} * ln\frac{1}{8})\right) \quad (15)$$

$$\frac{3}{2}kln(2) \qquad\qquad (16)$$

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 (17)-(19). When (19) is solved for Q in the instance where $N_{out} - N_{in} = 1$, the result (20) gives the amount of the minimal heat generation Q per fixed computing cycle.

$$kln(2^{N_{out}}) - kln(2^{N_{in}}) = \int_{N_{in}}^{N_{out}} \frac{dQ}{T} \qquad\qquad (17)$$

$$kN_{out}ln(2) - kN_{in}ln(2) = \int_{N_{in}}^{N_{out}} \frac{dQ}{T} \qquad\qquad (18)$$

$$kln(2)(N_{out} - N_{in}) = \int_{N_{in}}^{N_{out}} \frac{dQ}{T} \qquad\qquad (19)$$

$$Q = kTln(2) \qquad\qquad (20)$$

Bennett responded to Landauer's research by suggesting in [3] that a simple, general binary computational device may be realized in which every clock cycle is logically reversible. The device saves all of the intermediate calculations results in the first computational stage, which separates it from an irreversible device. During the next stage, the device outputs the calculated data that resulted from the reversible calculations. In the final stage, the device retraces all of the stages 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.

Determining the entropy difference in such a computing device where the number of input lines and output lines are identical, substituting $N_{out} = N_{in}$ for (20) reduces to the equation presented in (21). As noted in by Clausius, when the integral of the rate of change of heat divided by the temperature function equals zero, the transformations exactly cancel each other out, and the system in physically reversible.

$$\int \frac{dQ}{T} = kln(2)(N_{out} - N_{in}) = 0 \qquad (21)$$

Since the previous state must be determinable in a physically reversible system as demonstrated in Feynman, and that statistical probability of each state directly correlates to the thermal transformation between the input and output state as demonstrated by Einstein, such a binary reversible computing structure must be bijective and possess the same number of input and output lines.

Toffoli demonstrated in [7] that reversible logic structures are satisfactory for design and implementation in computing structures and organization when those design rules ensure the logic structure is invertible. Deutsch proposed in [37] that a universal computing machine may be implemented in order to perfectly simulate every finitely realizable physical system. This is done using a quantum computer that consists of a lattice of spinning electrons. The interactions of these quantum corpuscles within the machine may be represented in Feynman's diagrams, allowing for the behavior of these electrons to be simulated with cellular automation, and for a reversible implementation of such a device to be realized. Therefore, the dynamics of a quantum computer may be described using the wave function of an observable electron, since each electron in the quantum computer is represented by a constant unitary operator in the Hamiltonian space.

Deustch followed in [37] by stating that universal quantum gates strung together by unit wires are sufficient for the design of a quantum computational network Quantum gates are the generalization of classical logic gates. He defined a source bit of '0' or '1' as a gate which, once every computational step, produces a value of '0' or '1' on its output. He states that source bits are reversible gates, since there is a bijection between the produced value at the input of the gate, and the produced output. Since his publication, source bits have come to be known as ancillary inputs. Therefore, in a reversible system, if one input bit is always held at a particular value, the probability of input states and output states remain bijective, and are still adequate to determine the thermal transformation between quantum particles.

This model quantum computer consists of two parts. First, it consists of a processor with M 2-state observables, and is notated as in (22). In addition, the memory of a universal computing machine consists of an infinite sequence of 2-state observables, and is described in (23). Therefore, the state of a quantum computer Q can be represented, as in (24), as a unit vector in a Hamiltonian space spanned by the eigenvectors of the position x, the processor $\hat{n}_i$ and the memory $\hat{m}_i$.

$$\{\hat{n}_i\} \ (i \in \mathbb{Z}_M) \qquad (22)$$

$$\{\hat{m}_i\} \ (i \in \mathbb{Z}) \qquad (23)$$

$$|x; n; m\rangle \equiv |x; n_0, n_1 \dots n_{M-1}; \dots m_{-1}, m_0, m_1 \dots\rangle (24)$$

A Turing machine halts when two consecutive states are identical. The computation proceeds in steps of fixed duration t, and during each step only the processor and a finite part of the memory interact, the rest of the memory remaining static.

17

However, the equation (25) shows that any non-trivial computation in the quantum computer will never produce two consecutive identical states, which is a property of a physically reversible system.

$$\left|\psi(n * t)\right\rangle = \cup^n \left|\psi(0)\right\rangle \ (n \in \mathbb{Z}^+) \qquad (25)$$

Deustch followed in [37] by stating that universal quantum gates strung together by unit wires are sufficient for the design of a quantum computational network Quantum gates are the generalization of classical logic gates. He defined a *source bit* of '0' or '1' as a gate which, once every computational step, produces a value of '0' or '1' on its output. He states that source bits are reversible gates, since there is a bijection between the produced value at the input of the gate, and the produced output. Since his publication, source bits have come to be known as *ancillary inputs*. Therefore, in a reversible system, if one input bit is always held at a particular value, the probability of input states and output states remain bijective, and are still adequate to determine the thermal transformation between quantum particles.

2.4 Reversible Logic Gates

There are three basic 2x2 reversible logic gates. The Controlled-Not gate [4] – commonly called the Feynman gate - is designed to produce the following output states: $P = A$ and $Q = A \oplus B$. Since fanout is expressively forbidden in reversible logic, since a

fanout has one input and two outputs, the Feynman gate may be used to duplicate a signal when B is equal to 0.

Fig 2.4: Quantum Representation of Feynman Gate

The square-root-of-not gates utilize the unitary operators to produce reversible logic calculations when a select line is set at '1'. The Controlled-V and the Controlled-V+ gates are the two types of square-root-of-not gates. In both of these gates, when the control input is 0, the second input is propagated to the output. When two Controlled-V or two Controlled-V+ gates are activated in series, they act as an inverter. When a Controlled-V and Controlled-V+ gate are activated in series, they act as an identity. The corresponding unitary operator is propagated to the second output when the control input is 1, where the unitary operation for the Controlled-V is $\frac{i+1}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$, and $\frac{1}{i+1}\begin{pmatrix} 1 & -1/i \\ i & 1 \end{pmatrix}$ for the Controlled-V+ gate.
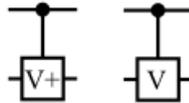
Figure 2.5: Quantum Representation of Controlled-V/V+ Gates

The third type of fundamental 2x2 reversible logic gate is the integrated qubit gate [26]. This gate is implemented with a Feynman gate with either a Controlled-V or Controlled V+ gate. The quantum cost of the integrated qubit gate is 1 and its worst-case delay is 1. The quantum configurations of all eight integrated-qubit gates are shown below in Fig. 2.6.
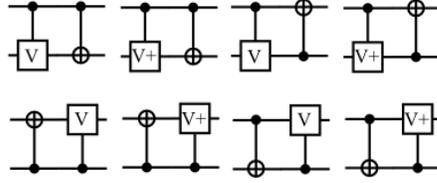
Fig 2.6: Quantum Representations of Integrated Qubit Gates

Next, there are four 3x3 reversible logic gates which may be implemented in a programmable manner. The Fredkin gate, proposed in [6], produces the following logical output calculations: $P = A$, $Q = A'B \oplus AC$ and $R = AB \oplus A'C$. Therefore, the outputs serve as a multiplexed output of the two data inputs based on the control input. It is realized using 2 Feynman gates, a Controlled-V gate and two integrated qubit gates. The quantum representation of the Fredkin gate is shown in Fig. 2.7.
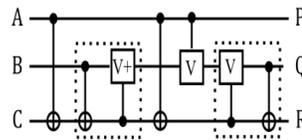
Fig 2.7: Quantum Representation of Fredkin Gate

The Toffoli gate, proposed in [7], is a 3x3 gate which produces the following output logical calculations: $P = A$, $Q = B$ and $R = AB \oplus C$. The quantum cost is 5 and the worst-case delay is 5. The Toffoli gate is known as a universal gate, because it may be used to produce AND, NAND, OR and NOR on the R output. The quantum representation of the Toffoli gate is shown below in Fig. 2.8.
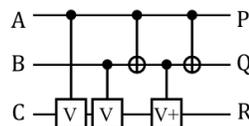
Fig 2.8: Quantum Representation of Toffoli Gate

The Peres gate, proposed in [8], has a quantum cost and worst-case delay of 4. It produces the following logical output calculations: $P = A$, $Q = A \oplus B$ and $R = AB \oplus C$. It

is advantageous to the Toffoli gate in that it produces the same logical output calculations on the R output at a reduced cost and delay. However, it is disadvantageous in designs where the programmer desires to propagate both the A and B inputs to the output. The quantum representation of the Peres gate is shown in Fig. 2.9.
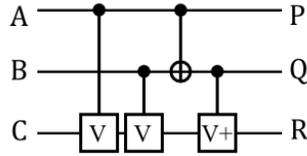
Fig 2.9: Quantum Representation of Peres Gate

The TRG gate, proposed in [9], has a quantum cost and worst-case delay of 4. It produces the following logical output calculations: $P = A$, $Q = A{\oplus}B$, and $R = AB'{\oplus}C$. The TRG may be implemented in the design of a full subtractor, and is advantageous in that cascaded TRG gates can be reduced, since the Controlled-V+ from the first TRG and the Controlled-V from the second TRG form an identity, and both can be omitted from the design. The quantum representation of the TRG gate is shown in Fig. 2.10.
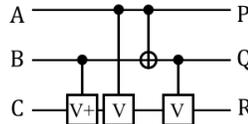
Fig. 2.10: Quantum Representation of the TRG Gate

Several 4x4 and 5x5 gates have been described in the literature targeting low cost and delay which may be implemented in a programmable manner to produce a high number of logical calculations. The HNG gate, presented in [10], produces the following logical output calculations: $P = A$, $Q = B$, $R = (A{\oplus}B){\oplus}C$, and $S = (A{\oplus}B)C{\oplus}(AB{\oplus}D)$. The quantum cost and delay of the HNG is 6. When $D = 0$, the logical calculations produced on the R and S outputs are the required sum and carry-out

operations for a full adder. The quantum representation of the HNG is presented in Fig. 2.11.
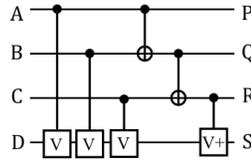
Fig. 2.11: Quantum Representation of the HNG Gate

2.5 Reversible Arithmetic Logic Units

A reversible arithmetic logic unit was designed by Thomsen, Glück, and Axelsen [18] that was based on the V-shaped design of the Van Rentergem adder [19]. The ALU had five fixed select lines, and produced the following logical outputs: ADD, SUB, NSUB, XOR and NOP. The least significant bit comprised of two Feynman gates and two Toffoli gates. Each additional bit also had two Fredkin gates.
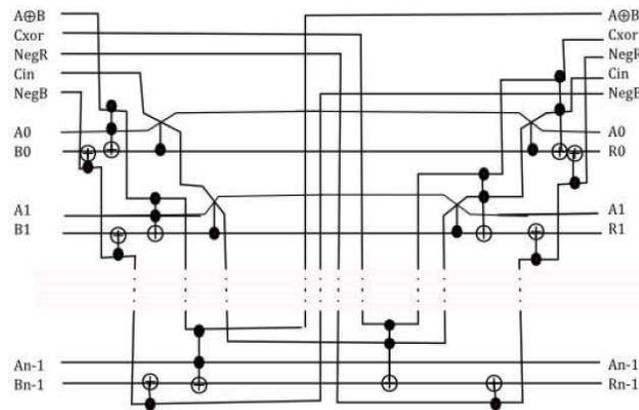
Fig 2.12 – Reversible ALU Presented by Thomsen et al

2.6 Reversible Comparators

A sequential reversible comparator was presented by in [14]. For an n-bit comparator, the quantum cost is $39n+9$, a delay is $24n+7$, and incurs a total of $8n$ garbage outputs. A tree-based reversible comparator was presented in [15] which utilized the TR

22

gate as an effective design element in cost and delay reduction. The cost of the device is $18(n - 1) + 9$. The delay of the device is $18 * log_2 n + 7$. The number of garbage outputs for the design is 6(n-1).

Table 2.2
Comparison of 32-Bit Reversible Comparators

	Quantum Cost	Delay	Garbage Outputs
Previous Sequential Design [14]	1257	775	256
Previous Tree-Based Design [15]	567	97	186

CHAPTER 3

PROPOSED REVERSIBLE LOGIC STRUCTURES

3.1 - Swap Gate

We describe a reduced implementation of the reversible Swap Gate, which is designed using two integrated qubit gates, and produces a swap of the two input values on the output gate. Previously, the swap gate was implemented using three Feynman gates which produced the outputs $P = A \oplus (A \oplus B)$ and $Q = A \oplus (A \oplus B) \oplus (A \oplus B)$, which produces the swap, and incurred a quantum cost and delay of 3. The proposed implementation is accomplished with a quantum cost and delay of 2, was verified using VHDL in Xilinx 12.4, and is shown in Fig. 2.1.
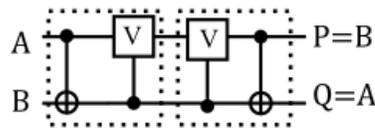
Fig 3.1: Integrated Qubit Gates Implemented as a Swap Gate

3.2 – Programmable Reversible Logic Theorems

Any arithmetic logic unit must be able to produce a variety of logical outputs [41][42], such as AND, OR, and XOR, based on inputs determined by the programmer for implementation in an instruction set architecture. Therefore, a reversible gate used for this purpose must be able to maximize the types of logical operations it can calculate while minimizing the number of select lines and logical output lines, cost and delay. To this end, a programmable reversible logic gate is defined here as a logic structure which possesses a bijection between input and output states and an equal number of inputs and

outputs wherein a subset of the inputs are fixed select lines, and a fixed subset of the outputs produce guaranteed logical calculations.

The designer of a programmable logic device must also consider which values to propagate to the output. In some instances, it may be beneficial to produce a copy of the input data values, whereas other designers may wish to propagate the input signals to the output signals. The following two theorems are presented in order to ensure reversibility is maintained in the design of these logic gates. In these theorems, the total number of inputs is j, the number of data inputs is d, the fixed input select lines is m, the quantity of chosen propagated values to the output is p and the fixed select output lines is n.

Theorem 3.1: An ideal programmable reversible logic gate with j inputs and outputs has a quantity of fixed select inputs m, fixed select outputs n, data inputs d and propagated outputs p such that $|d - p| = |m - n|$.

Proof: A reversible logic gate must have the same number of inputs and outputs. The number of fixed select inputs is the difference between the total inputs and the data inputs such that $j = d + m$. The number of fixed logical outputs, n, may be any value between 1 and $j - p$. When $m > n$, a number of garbage outputs g are incurred to maintain reversibility such that $m - n = p - d + g$. Therefore, in order to eliminate garbage outputs, the values for $m - n$ and $p - d$ must be identical. When $n > m$, a number of ancillary inputs a are incurred such that $n - m + a = d - p$. Therefore, in order to eliminate ancillary inputs, the values for $n - m$ and $d - p$ must be identical. Therefore, in order to maintain reversibility and eliminate ancillary inputs and garbage outputs, $|d - p| = |m - n|$.

Theorem 3.2: A programmable reversible logic gate with *m* select inputs may produce at maximum $n * 2^m$ logical calculations on the *n* logical outputs.

Proof**:** The *m* select inputs represent an input signal from the programmer, and allow for up to 2^m unique input combinations. For each unique input signal combination, there may be one logical calculation per output. Since there are *n* outputs, the maximum number of logical outputs is $n * 2^m$.

3.3 – Morrison-Ranganathan Gate (MRG)

In this section, a programmable 4x4 reversible logic structure - the MR (MRG) gate – is presented which produces outputs $P = A$, $Q = A \oplus B$, $R = (A \oplus B) \oplus C$, and $S = (AB \oplus D) \oplus ((A \oplus B) \oplus C)$. Fig. 6 shows the block diagram of the MRG gate.

The MRG gate has a quantum cost of 6, since it consists of three XOR gates, 2 Controller-V and one Controller-V+ gate. The worst-case delay of the MRG gate is 4. The quantum representation of the MRG gate is shown in Fig. 7 below. The truth table is shown in Table 2.

When the MRG is utilized as a programmable reversible logic gate with two select inputs, it will calculate four logical calculations on those two logical outputs: OR, NOR, XOR and XNOR. Table 3 shows the configurations of the MRG gate for different select input signal combinations, and the resultant logical output calculations. The MRG may be configured in a regular manner to simultaneously calculate if the two input bits are equal, their XOR value and their implication. This configuration is shown in Fig. 3.4.
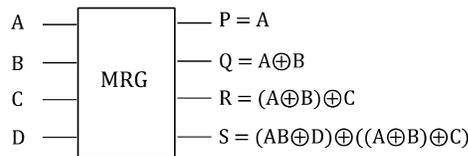
Fig. 3.2: Block Diagram of the MRG Gate
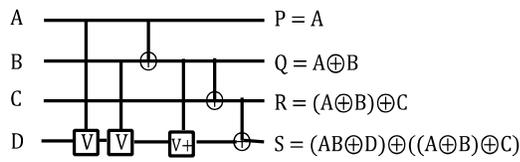
Fig. 3.3: Quantum Representation of the MRG Gate
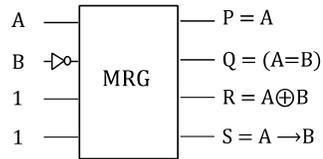
Fig. 3.4: MRG Logical Configuration

TABLE 3.1
MRG Truth Table

A	B	C	D	P	Q	R	S
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	0	1	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	1
1	1	0	0	1	1	0	1
1	1	0	1	1	1	0	0
1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	1

TABLE 3.2
MRG Programmable Inputs and Logical Outputs

S0 (C)	S1 (D)	R	S
0	0	XOR	OR
0	1	XOR	NOR
1	0	XNOR	NOR
1	1	XNOR	OR

27

3.4 – Peres And-Or Gate (PAOG)

In this section, a programmable 4x4 reversible logic structure - Peres And-Or (PAOG) gate – is presented which produces outputs $P = A$, $Q = A \oplus B$, $R = AB \oplus C$, and $S = (AB \oplus C) \oplus ((A \oplus B) \oplus D)$. Fig. 3.5 shows the block diagram of the PAOG gate. This gate is an extension of the Peres gate for ALU realization.
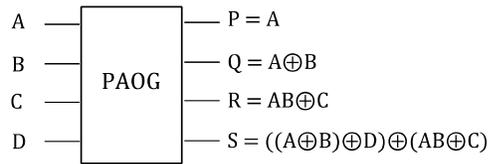
Fig. 3.5: Block Diagram of the PAOG

The PAOG gate is based on the Peres Gate. The fourth output is passed through two Feynman gates, which are tied to the outputs Q and R. The cost and delay are identical to the MRG. The quantum representation of the PAOG gate is shown in Fig. 3.6 below. The truth table is presented in Table 3.3.
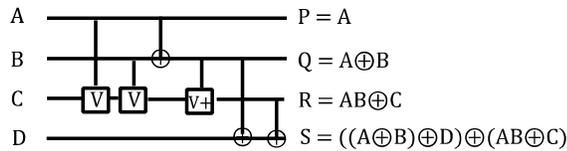
Fig. 3.6: Quantum Representation of the PAOG

When the PAOG is utilized as a programmable reversible logic gate with two select inputs, it will calculate four logical calculations on those two logical outputs: OR, NOR, AND and NAND. Table 5 shows the configurations of the PAOG gate for different select input signal combinations, and the resultant logical output calculations.

Table 3.3
PAOG Truth Table

A	B	C	D	P	Q	R	S
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	1	0	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	1	0
1	0	1	1	1	1	1	1
1	1	0	0	1	0	1	1
1	1	0	1	1	0	1	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	1

Table 3.4
PAOG Programmable Inputs and Logical Outputs

S0 (C)	S1 (D)	R	S
0	0	AND	OR
0	1	AND	NOR
1	0	NAND	NOR
1	1	NAND	OR

The designs of the MRG and PAOG were verified in Verilog using the ModelSIM SE 6.3 Advanced Programming and Debugging program.

3.5 - Universal Programmable Gate (UPG)

In this section, a 3x3 reversible universal programmable gate (UPG) which implements the following logical configuration is presented: $P = A$, $Q = (A \oplus B) \oplus (AB \oplus C)$ and $R = AB \oplus C$. The output for Q is reducible to $(A + B) \oplus C$. The quantum cost and delay of the new gate is 4, and the quantum representation is shown in Fig. 3.7. The design of the UPG was verified in VHDL in Xilinx 12.4. This new

gate is named UPG since it may be configured as a programmable 3x3 gate to produce

the logical calculations AND, NAND, OR and NOR at low quantum cost when input C

as the fixed input select line, and Q and R as the fixed logical output lines. When $C = 0$,

$Q = A + B$ and $R = AB$. When $C = 1$, $Q = \overline{A + B}$ and $R = \overline{AB}$.
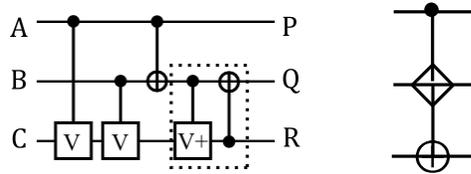
Fig 3.7: Quantum Representation and Logic Symbol of UPG
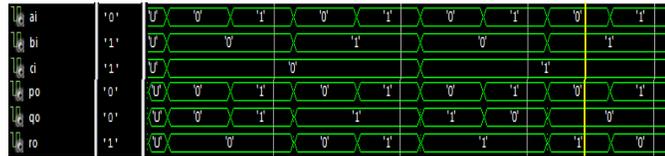
Fig 3.8: UPG Simulation in VHDL

Table 3.5
UPG Truth Table

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

The quantum cost and delay of the UPG is identical to the Peres gate. The new gate is advantageous in that, when used in the programmable configuration described above, the UPG produces 4 unique logical calculations, whereas the Peres may only produce 3 (XOR, AND and NAND). In addition, the UPG is advantageous to the Toffoli gate in that it has an improved quantum cost and is able to produce all the desired logical outputs without any manipulation of the inputs. The UPG is disadvantageous to the Toffoli gate in an implementation where the programmer desires to propagate the inputs to the output.

Claim: The UPG is the smallest programmable reversible logic structure in terms quantum cost and delay that can obtain the logical calculations for AND, NAND, OR, and NOR for two data inputs and one select input.

Proof: In order to obtain the AND and NAND calculation, a minimum of four 2x2 fundamental reversible logic structures are required. There must be two Controlled-V, one Controlled-V+ and one Feynman gate. This is because an input value of '0' must be changed to '1' only when A and B are both '1', and an input value of '1' must be changed to '0' when both A and B are both '1'. This inversion is done minimally with the use of two Controlled-V gates, which will only invert when both are activated. When none of them are activated, the signals are passed through. When only one of them is activated, a Controlled-V+ must be on the same line, and may be activated if A or B are exclusively activated. This is achieved minimally using a Feynman gate whose Q output controls the activation line for the Controlled-V+ gate. This configuration is shown as the Peres Gate. When the last Controlled-V+ in the Peres Gate is replaced with an Integrated Qubit gate, which incurs the same cost, the output calculation for $Q = (A \oplus B) \oplus$

$(AB \oplus C)$. When $C = 0$, this becomes $Q = (A \oplus B) \oplus AB = A + B$. When $C = 1$, the equation becomes $Q = A \oplus B \oplus (AB)' = (A + B)'$. Therefore, both AND, NAND, OR and NOR for two data inputs may be obtain using the UPG by using one select output.

3.6 - N-bit OR and ZERO Detection

Since the UPG has a reduced quantum cost and increased unique logic outputs, it may be implemented in an n-bit AND or n-bit OR calculation. The cost and delay of this design is *4n*. This is the cheapest implementation of an n-bit reversible OR calculation.

Another benefit of this design is that the OR output may be used for testing if the output of any calculation is ZERO. If the first *n-1* bits are passed through *n-1* UPG gates where $C = 0$, and the nth bit has $C = 1$, then the R output of the most significant UPG will be 1 if and only if all of the data inputs are 0. For a 32-bit ALU, the quantum cost and is 124. The design of a 4-bit ZERO calculation is shown in Fig. 3.9.
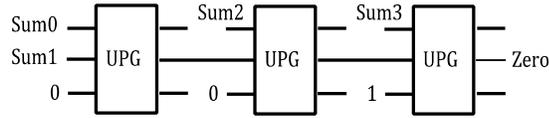
Fig 3.9: Four-Bit Zero Calculator using Proposed UPG gates

3.7 - Reduced Reversible Logic Multiplexer Gates

We describe A 3x3 Reversible Multiplexer (RMUX1) gate is proposed which implements the following logical configuration: $P = A$, $Q = A'B + AC$ and $R = A'C + AB'$. The output Q serves as the multiplexed output of B and C, where A is the select line. The quantum cost and worst-case delay of the new gate is 4, and the quantum representation is shown in Fig. 3.10. The design of the RMUX was verified in VHDL in Xilinx 12.4. The RMUX1 truth table is present in Table 3.6.
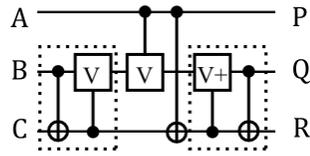
Fig 3.10: Quantum Representation of Proposed RMUX1

Table 3.6
RMUX1 Truth Table

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

Fig 3.11: RMUX1 Gate Simulation in Xilinx

The second RMUX gate (RMUX2) implements the following logical configuration: $P = A$, $Q = A'B + AC$ and $R = A \oplus (B \oplus C)$. Just like the RMUX1, the Q output produces the multiplexed output of B and C, where A is the select line. The quantum cost and worst-case delay of the new gate is 4, and the quantum representation is shown in Fig. 3.12. The design of the RMUX2 was verified in VHDL in Xilinx 12.4.

The proposed RMUX gates are advantageous to the Fredkin gate in designs where low cost and fast multiplexing are desired, since the RMUX gates are able to produce a multiplexed output of the B and C inputs in a reduced cost and delay of one compared to the Fredkin. Therefore, in order to multiplex n values, the RMUX provides an improvement in quantum cost of n-1 and an improvement in delay of $log_2 n$. However, it is disadvantageous to the Fredkin gate in designs where both inputs are derived on the output gate, such as Shannon expansion, which gives the Fredkin gate added functionality.
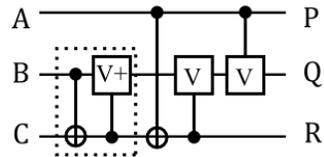
Fig 3.12: Quantum Representation of Proposed RMUX2

Table 3.7
RMUX2 Truth Table

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	1	1	1
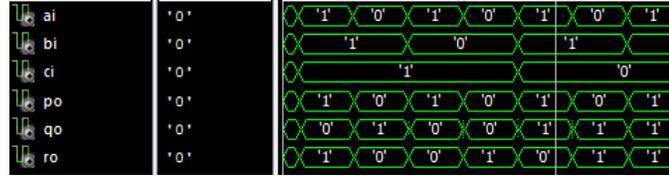
Fig 3.13: RMUX2 Gate Simulation in Xilinx

3.8 - Reversible Comparator Gate

A 4x4 Reversible Comparator (RC) gate is proposed that implements the following logical configuration: $P = A$, $Q = (A \oplus B) \oplus (B \oplus AB)$, $R = B \oplus C \oplus AB$, and $S = A \oplus B \oplus D$. The quantum cost and delay of the proposed RC is 5, and the quantum representation is shown in Fig. 3.14. The logical outputs based on the programmable inputs of the RC are shown in Table 3.9. The design of the RLU was verified in VHDL in Xilinx 12.4.
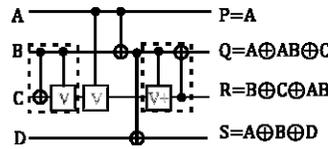
Fig 3.14: Quantum Representation of Proposed RC Gate
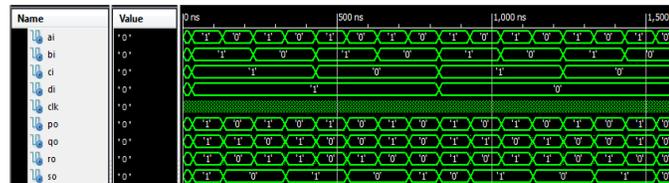
Fig 3.15: RC Gate Simulation in Xilinx

The configuration of the RC gate when $C = 0$ and $D = 1$ is a special case which allows for its implementation as a comparator. This configuration of select inputs produces the logical calculations: $Q = AB'$, $R = A'B$, and $S = A \oplus B \oplus 1$. The logical calculation for AB' is equivalent to $A > B$, and the logical calculation for $A'B$ is

35

equivalent to $B > A$. The logical calculation for $A \oplus B \oplus 1$ is equivalent to $A = B$. Therefore, it may be determined whether A is greater than, equal to or less than B.

Table 3.8
RC Truth Table

A	B	C	D	P	Q	R	S
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	1
0	1	0	0	0	0	1	1
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	1
1	0	0	1	1	1	0	0
1	0	1	0	1	0	1	1
1	0	1	1	1	0	1	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	1

Table 3.9
RC Programmable Inputs and Logical Outputs

C	D	Q	R	S
0	0	AB'	$A'B$	$A \oplus B$
0	1	AB'	$A'B$	$\overline{A \oplus B}$
1	0	$A + B'$	$A' + B$	$A \oplus B$
1	1	$A + B'$	$A' + B$	$\overline{A \oplus B}$

Ideally, it is desirable to produce all three logical calculations, AB', $A'B$, and $A \oplus B \oplus 1$ with 3 logical outputs and no propagated or garbage outputs. However, this is not possible, since the bijective nature of the circuit would be lost.

Claim: It is not possible to design a 3x3 programmable reversible comparator gate where the logical outputs AB', $A'B$, and $A\oplus B\oplus 1$.

Proof: Let us assume that there exists a 3x3 programmable reversible logic circuit producing the logical outputs AB', $A'B$, and $A\oplus B\oplus 1$ simultaneously. This means that the all three outputs of the reversible logic structure are logical output calculations, and that there are no propagated or garbage outputs. Therefore, there must be 3 inputs such that two of them are the data inputs A and B, and the third input line must be an ancillary input or a select line input. Therefore, the logical output combination stated above must be obtained either when the third input is a 0 or a 1. Since the output combination was achieved in a reversible structure, then the output states must be bijective. However, the input vector {A,B,a0} for {0,0,X} and {1,1,X) will both produce output vectors {0,0,1} as an output for the logical output combinations. Therefore, since the structure is not bijective, it is not reversible. This is a contradiction.

3.9 – Morrison Gate (MG)

Next, we propose the design of a 5*5 programmable reversible logic gate structure utilized in the implementation of an ALU. Fig. 3.16 shows the block diagram of the MG, and the logical calculations based on the programmable inputs are presented in Table 2.10. The cost of the MG is 7, and the worst-case delay is 7. The design for the programmable MG was verified and simulated using VHDL in Xilinx 12.4.
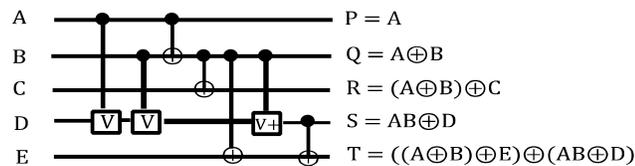
Fig 3.16: Quantum Representation of Proposed MG.

Table 3.10
MG Programmable Inputs and Logical Outputs

C	D	E	R	S	T
0	0	0	$A \oplus B$	AB	$A + B$
0	0	1	$A \oplus B$	AB	$(A + B)$
0	1	0	$A \oplus B$	(AB)	$(A + B)$
0	1	1	$A \oplus B$	(AB)	$A + B$
1	0	0	$A = B$	AB	$A + B$
1	0	1	$A = B$	AB	$(A + B)$
1	1	0	$A = B$	(AB)	$(A + B)$
1	1	1	$A = B$	(AB)	$A + B$

CHAPTER 4

REVERSIBLE COMPARATOR USING RC AND UPG

4.1 - Proposed Reversible Comparator Design

In this section, the RC and UPG are implemented in the design an n-bit sequential comparator capable of simultaneously outputting bitwise AND, bitwise OR, $A > B$, $A < B$ and $A = B$. The design of a two-bit comparator is presented in Fig. 24, and a behavioral verification in Xilinx 12.4 is presented in Fig. 4.1. An n-bit sequential implementation of this comparator incurs a cost of *20n-12*. The worst-case delay of this device is is *10n-2*. The design also incurs *5n+1* garbage outputs.
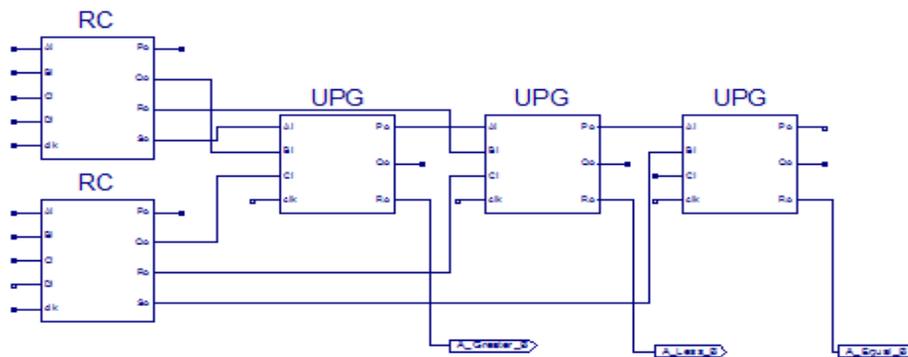
Fig 4.1: Reversible 2-Bit Comparator Composed from RC and UPG
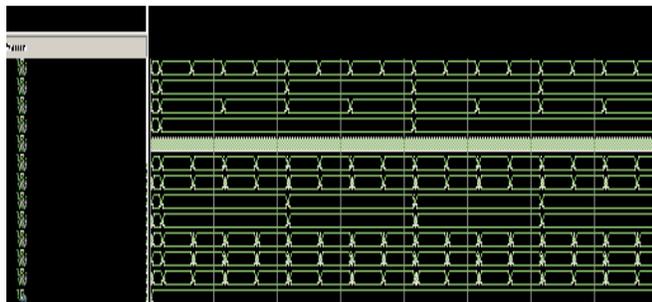
Fig. 4.2: VHDL Behavioral Verification of 2-Bit Comparator

4.2 - Tree-Based Comparator Design

Next, the 2-bit comparator is modified in order to implement a 32-bit tree-based comparator. The design requires 32 RLU gates to produce the initial comparator values for each bit, incurring a cost of 256. When the comparisons are arraigned in tree-based, the 32-bit comparator requires 31 of these UPG configurations, incurring a cost of 12 each. As a result, the proposed tree-based reversible comparator has a quantum cost of 628, which is identical to the sequential design. The tree-based comparator allows for an significant improvement in delay, since the delay is reduced from $10n - 2$ to $8 + 10log_2 n$ in a tree-based implementation. In the 32-bit tree-based comparator design using the RLU and UPG, the delay is reduced from 318 to 58, an 81.8% improvement. The design of a 4-bit tree-based reversible comparator is shown in Fig. 4.3, and was verified in VHDL in Xilinx 12.4.
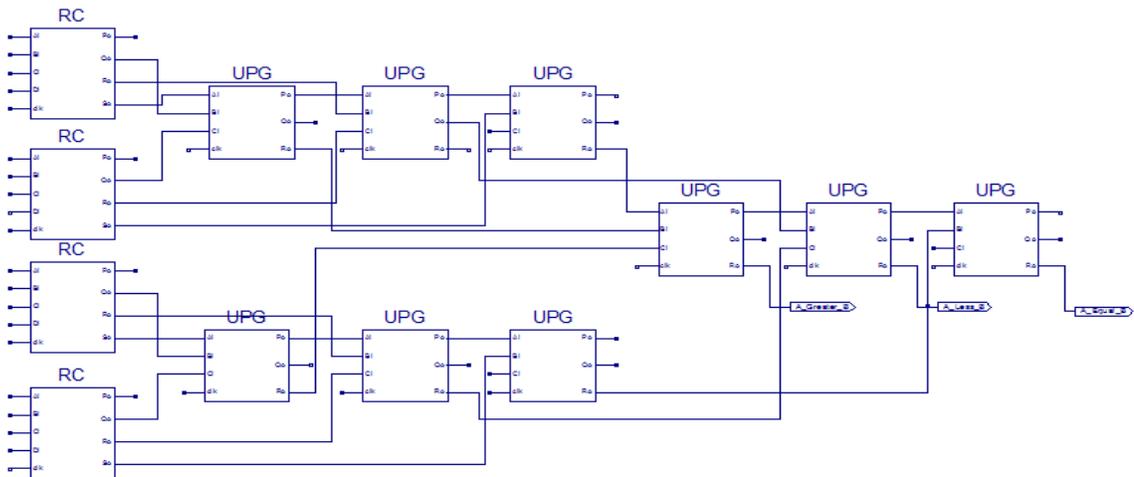
Fig 4.3: Reversible 4-Bit Tree-Based Comparator

4.3 - Comparison to Previous Designs

A sequential reversible comparator was presented by in [14]. For an n-bit comparator, the quantum cost is 39n+9, a delay is 24n+7, and incurs a total of 8n garbage

outputs. A tree-based reversible comparator was presented in [15] which utilized the TR gate as an effective design element in cost and delay reduction. The cost of the device is $18(n - 1) + 9$. The delay of the device is $18 * log_2 n + 7$. The number of garbage outputs for the design is 6(n-1).

The comparison of the tree-based designs shows that the proposed design is an improvement over the previous work in terms of delay. The quantum delay of the proposed 32-bit tree-based design gives a 40% improvement in delay. The quantum cost is the same as the sequential design, which is 9.7% higher than the previous tree-based design. However, since the proposed design also outputs a 32-bit bitwise OR and a 32-bit bitwise AND as part of its functionality, it allows for reduced implementation in a reversible ALU. In order for the previous tree-based design to implement bitwise OR and a bitwise AND, it would incur an addition cost of 5 per bit (one Feynman and one UPR), which would also incur 1 additional garbage bit. In a 32-bit ALU, this would result in an additional cost of 160, raising the cost to 727. In this instance, the improvement in cost of the proposed design over the previous tree-based design is 13.6 percent. Also, since the *2n* bits per RLU are being utilized, they are no longer garbage outputs, reducing the garbage outputs in that implementation to 95.

Table 4.1
Comparison of Proposed and Previous Reversible Comparators

	Quantum Cost	Delay	Garbage Outputs
Previous Sequential Design [14]	1257	775	256
Proposed Sequential Design	532	315	160
Previous Tree-Based Design [15]	567	97	186
Proposed Tree-Based Design	532	55	160

CHAPTER 5

ENHANCED CARRY LOOK-AHEAD ADDER

In order to design the most efficient 32-bit reversible arithmetic logic unit, we designed and compared reversible implementation of ripple-carry, carry-select and carry look-ahead adders [11]. A reversible ripple-carry adder and a reversible carry-select adder are designed using the new ALU. The ripple-carry adder has a cost of 40n-3 and a delay of 4n + 13. The most-significant bit of the ALU ties the Sum output to the SLT input of the least significant bit, and the SLT input for all other bits is 0. The reversible carry-select adder uses the carry out of the first n/2 bits as the control signal to a Fredkin gate implemented as a multiplexer, since the next n/2 bits are calculated with both a carry-in of 0 and a carry-in of 1, which requires a cost of 40(3n/2)-3 and a delay of 2n + 19.

5.1 - Reduced Reversible Kogge-Stone Cumulate Logic

Next, a reversible carry look-ahead adder was presented which was based on the Kogge-Stone adder [9]. First, a RKS Cumulate utilized in the calculation of the carry out signal was designed and verified. The cost of the RKSC is 14 and it has a worst-case delay of 4.

The previously presented RKSC may be reduced in terms of cost, delay and garbage outputs by using the proposed UPG. The cost of the new RKSC is reduced from 14 to 12 by replacing the MRG with a UPG. The MRG was the previous best gate to use in terms of quantum cost and delay to obtain an OR structure, with an incurred cost and

delay of 6, as well as two garbage outputs [5]. Exchanging the MG with the UPG reduces

the cost and delay by two. The presented design is shown in Fig. 5.1.
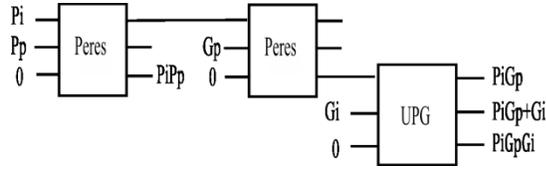
Fig 5.1: Reduced RKSC Layout

5.2 - Comparison of Reversible Ripple-Carry and Carry-Select Adders with Sparsity

The Kogge-Stone adder may be enhanced to reduce overhead and design complexity by generating a carry every n-bits instead of every bit, and the carry is used for the carry-in of an n-bit ripple-carry or carry-select adder. The number n is defined as sparsity. This implementation was designed and tested for sparsity-4, 8 and 16 with the corresponding n-bit ripple-carry and carry-select adders. The cost and delay comparison of each implementation is shown in Table 5.1. The design for each adder presented in Table 5.1 was verified using VHDL in Xilinx 12.4.

Table 5.1
Comparison of Modified Reversible Carry Look-Ahead Adders

	Sparsity-4	Sparsity-8	Sparsity-16
Ripple-Carry	Cost: 606 Delay: 32	Cost: 462 Delay: 48	Cost: 300 Delay: 80
Carry-Select	Cost: 730 Delay: 29	Cost: 538 Delay: 37	Cost: 422 Delay: 45

In terms of quantum cost and delay, the best compromises are the sparsity-4 ripple-carry adder and the sparsity-8 carry-select adder. The carry select implementation

is a 11.2% improvement in terms of cost, and the ripple-carry is a 13.5% improvement in terms of delay. Since implementing the adder in a 32-bit ALU will essentially negate the advantage the carry-select implementation has in terms of cost, while the improvement in delay will always be 5, the sparsity-4 ripple-carry implementation is ideal for implementation in an ALU.

CHAPTER 6

NOVEL REVERSIBLE ARITHMETIC LOGIC UNIT

6.1 - Reversible ALU Design with MRG and PAOG

Two 1-bit ALUs are presented in this section. The first utilizes the MRG gate and HNG gate to produce six logical calculations: ADD, SUB, XOR, XNOR, OR and NOR. The ALU has 8 inputs and 8 outputs. The inputs consist of three data inputs (A, B and C_{in}) and five fixed input select lines. The eight outputs are: A, S0, S3 and S4 propagated to the output, $A \oplus B$, SUM, C_{out}, Overflow and Result. The cost of this 1-bit ALU is 24, and the worst-case delay is 16. For n-bit ALU devices, an addition cost of 2 is incurred per bit in order to propagate S1 and S2 to other bits. Therefore, the total cost for an n-bit ALU is 26n-2. The proposed ALU is shown in Fig. 15, and the logical results based on the input opcodes are presented in Table 6.1.

Table 6.1
ALU Opcodes and Logical Result for Fig. 6.1

S4	S3	S2	S1	S0	Result
0	0	0	0	0	XOR
0	0	0	1	0	=
0	1	0	0	0	OR
0	1	1	0	0	NOR
1	0	0	0	0	ADD
1	0	0	0	1	SUB
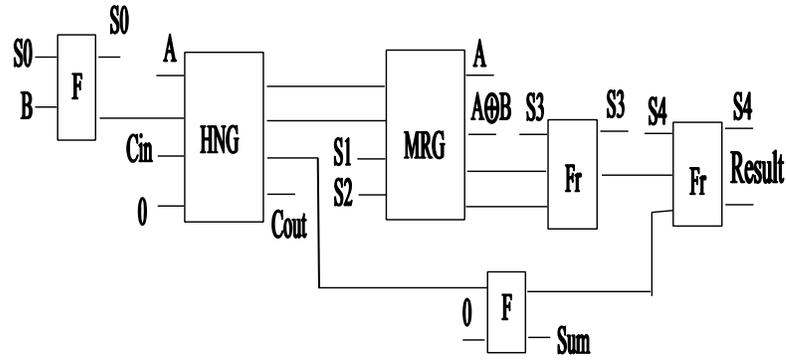
Fig. 6.1: Reversible ALU with MRG and HNG Gates

The second ALU utilizes the PAOG gate and HNG gate to produce six logical calculations: ADD, SUB, AND, NAND, OR and NOR. The cost and delay calculations are identical to the ALU in Fig. 6.2. The proposed ALU is shown in Fig. 16, and the logical results based on the input opcodes are presented in Table 6.2.

Table 6.2
ALU Opcodes and Logical Result for Fig. 6.2

S4	S3	S2	S1	S0	Result
0	0	0	0	0	AND
0	0	0	1	0	NAND
0	1	0	0	0	OR
0	1	1	0	0	NOR
1	0	0	0	0	ADD
1	0	0	0	1	SUB
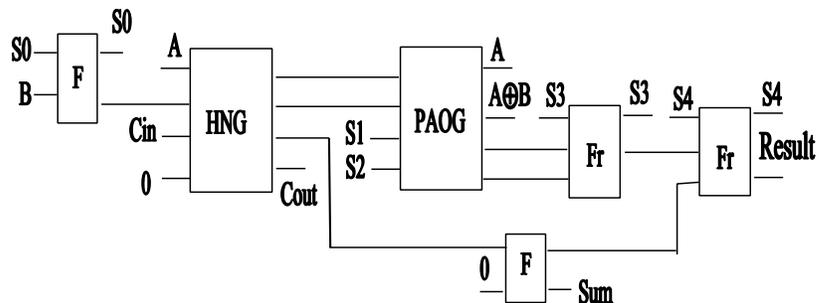
Fig. 6.2: Reversible ALU with MRG and HNG Gates

6.2 - Reversible ALU Design with MG

The MG gate is utilized in the implementation of a novel arithmetic logic unit based on those proposed in [8]. The ALU, in addition to producing the same logical calculations as the MG, is able to perform addition and subtraction by utilizing the HNG gate and store less-than operation. The cost an n-bit ALU is 37n-3 and had a worst-case delay of 4n+13. The proposed ALU is shown in Fig. 6.3, and the logical results based on the input opcodes are presented in Table 6.3. The design for the novel one-bit ALU was verified and simulated using VHDL in Xilinx 12.4.
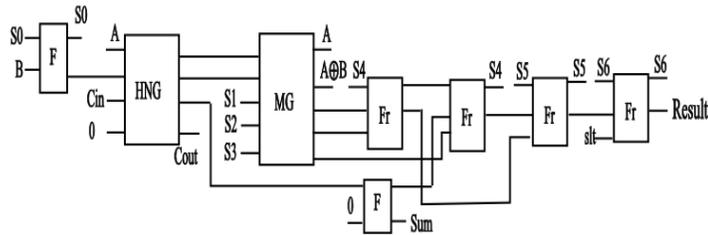
Fig. 6.3: Reversible ALU with MG and HNG Gates

Tabl 6.3
ALU Opcodes and Logical Result for Fig. 6.3

S6	S5	S4	S3	S2	S1	S0	Result
0	0	0	0	0	0	0	ADD
0	0	0	0	0	0	1	SUB
0	0	1	0	0	0	0	$A + B$
0	0	1	1	0	0	0	$(A + B)'$
0	1	0	0	0	0	0	$A \oplus B$
0	1	0	0	0	1	0	$A = B$
0	1	1	0	0	0	0	AB
0	1	1	0	1	0	0	$(AB)'$
1	0	0	0	0	0	0	SLT

6.3 – MG Implemented in a 32-Bit ALU with Kogge-Stone Adder

A 32-bit ALU utilizing a sparsity-4 carry look-ahead, adder implemented with ripple-carry has a cost of 1656 and a delay of 59. A 32-bit ALU utilizing a sparsity-8 carry look-ahead, adder implemented with carry-select has a cost of 1568 and a delay of 64. The carry-save implementation represents a 5.3% improvement in cost over the ripple-carry implementation, while the ripple-carry produces a 7.8% improvement in delay. Therefore, the modified carry look-ahead adder with sparsity-4 implemented with 4-bit ripple-carry addition is ideal for implementation in the reversible arithmetic logic unit. A 4-bit implementation of this adder is shown in Fig. 13 below.

The figure shows the propagate and generate signal logic for any adder which is not the most-significant 4-bits or the least significant 4-bits of adder. The least-significant adder is different in that the LSB does not have input propagate or generate signals, so the RKSCs are not necessary. This is shown in Fig. 12 The most-significant adder is different in that it does not need to generate any future propagate or generate signals other than the initial carry. Therefore, the HNG gates for the second, third and fourth bit which produce the propagate and generate signals are not required, nor is the RKSC for the third bit. This is shown in Fig. 14. The design for the entire 32-bit ALU was verified and simulated using VHDL in Xilinx 12.4.
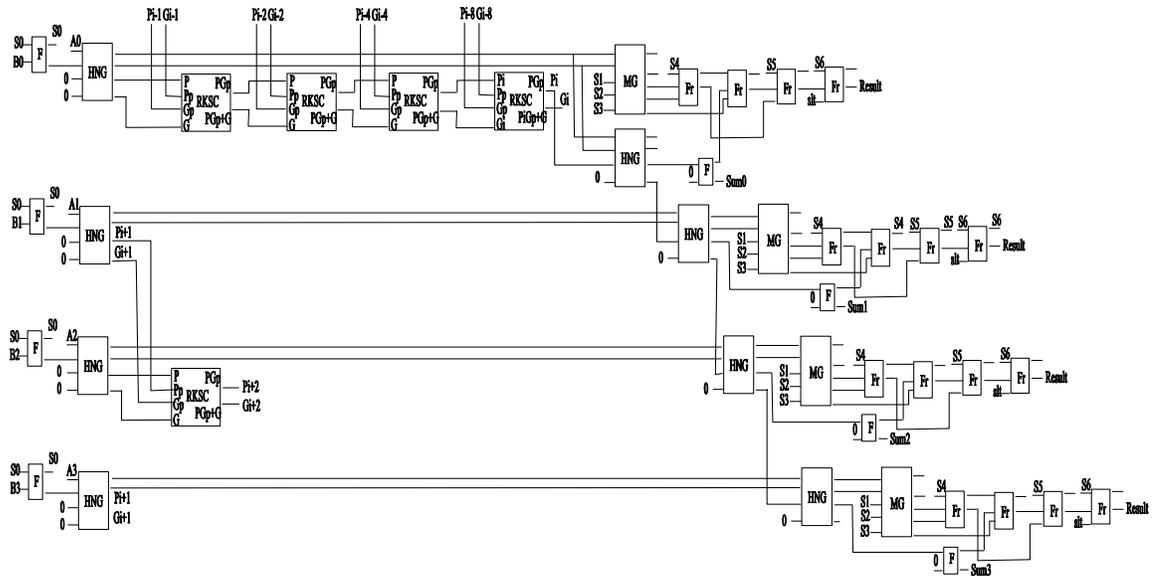
Fig. 6.4: Reversible 4-bit ALU with Modified Kogge-Stone Adder

6.4 – Reversible ALU with UPG, Comparator and Zero Implementation

A novel reversible ALU is proposed which implements the proposed comparator, multiplexers, UPG, RLU and the presented adder. The ALU outputs the following logical calculations: ADD, SUB, AND, NAND, OR, NOR, XOR, and XNOR, as well as output bits for A > B, A = B, A < B and ZERO. The total quantum cost of the proposed ALU is 1770. The worst-case delay of the device traverses from the first bit of the adder, through the ZERO calculation, and through the three RMUX1 gates, incurring a worst-case delay of 145. An 8-bit implementation of this ALU is presented in Fig. 29, and the opcodes necessary to produce the desired functions for the ALU are presented in Table 6.4.

Finally, the ALU may be adapted to account for ZERO calculation, which is desired in the construction of an ALU which needs to calculate branch instructions for CPU implementation. Using 32 UPG structures in the configuration shown in Fig. 2.4, ZERO may be calculated with an added cost and delay of 124. Therefore, the ALU incurs

a cost of 1972, and is able to calculate ADD, SUB, AND, NAND, OR, NOR, XOR, XNOR, A>B, A<B and ZERO.

Table 6.4
Opcodes and Logical Result for Proposed ALU

S5	S4	S3	S2	S1	S0	Result
0	1	X	X	X	0	ADD
0	1	X	X	X	1	SUB
0	0	0	X	0	X	$A + B$
0	0	0	X	1	X	$(A + B)'$
1	X	X	0	X	X	$A \oplus B$
1	X	X	1	X	X	$(A \oplus B)$'
0	0	1	X	0	X	AB
0	0	1	X	1	X	$(AB)'$

6.5 – Comparison to Previous Work

The proposed 32-bit ALU is compared in terms of cost and delay to the ALU proposed by Thomsen et al. in [28], the ALU based on the PAOG/MRG presented in [15], and the ALU based on the MG presented in [16]. The proposed ALU is higher in terms of cost than the previous ALUs, and has a slightly higher delay than the ALUs in [15] and [16].

The presented ALU has two major advantages. First, it produces more logical calculations than the previous ALU. Second, the worst-case delay for an n-bit ALU of the presented ALU is significantly lower, since the carry in signal in the previous work passes through *2n-2* Fredkin gates and the second input of *n* Toffoli gates, whereas the carry signal in the new ALU only passes through *n* HNG gates, 1 Feynman and 1 Fredkin. Therefore, the presented ALU has a better quantum delay for n-bit ALUs where $n \geq 2$. These comparisons are presented in Table 6.5.

Table 6.5
Reversible 32-bit ALU Comparison

	Cost	Delay	Logical Calculations
[18]	694	470	5: ADD, SUB, NSUB, XOR, NOP
Fig 6.1/6.2	830	135	6: ADD, SUB, OR, NOR, XOR, XNOR
Fig 6.3	1568	108	8: ADD, SUB, AND, OR, NAND, NOR, XOR, XNOR
Fig 6.5	1848	95	11: ADD, SUB, AND, NAND, OR, NOR, XOR, XNOR, >, <, =

CHAPTER 7 - CONCLUSIONS

First, a reduced implementation of the 2x2 reversible Swap gate utilizing integrated qubit gates was presented, which was an improvement over the previous implementation using Feynman gates. The concept of a programmable reversible logic structure and theorems were proposed that introduce new metrics for reversible logic design. Next, a 3x3 reversible UPG gate was presented and verified which is was a functional improvement on the Peres Gate, and gave an improved quantum cost and delay over the Toffoli gate, and is able to produce all the desired logical outputs – AND, NAND, OR, NOR, without any additional logical structures. Next, two 3x3 RMUX gates wer proposed which were able to produce a multiplexed output of two input bits based on a select signal which has reduced cost and delay compared to the Fredkin gate. A 4x4 RC gate was presented and verified which is capable of being programmed as a reversible comparator. The RC and UPG were implemented in the design of sequential and tree-based n-bit comparators. The tree-based comparator provided a significant improvement in both quantum cost and delay over previously presented tree-based comparators. Two highly programmable, low-cost and low-delay 4*4 reversible logic gates were presented, verified and compared to similar logic structures already published. The proposed MRG and PAOG matched the HNG as the best existing 4*4 reversible gates in terms of cost, delay and logical output calculations. The gates were then implemented in reversible arithmetic logic units. These new ALU designs are advantageous to previously published

work in implementations that favor low delay and high logical calculation output, which is desirable for realization of a reversible central processing unit.

The proposed designs are then integrated in the design of a novel reversible arithmetic logic unit. The ALU was verified, and then compared to previous reversible ALU research. The ALU design provides a significant improvement in functionality over previously proposed reversible arithmetic logic units.

REFERENCES

[1]   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.

[2]   R. Landauer, "Irreversibility and Heat Generation in the Computational Process," *IBM Journal of Research and Development*, vol. 5, 1961, pp. 183-91.

[3]   C. Bennett, "Logical Reversibility of Computation," *IBM Journal of Research and Development*, vol. 17, 1973, pp. 525-532.

[4]   R. Feynman, "Quantum Mechanical Computers," *Foundations of Physics*, vol. 16, iss. 6, 1986.

[5]   M. Morrison and N. Ranganathan, "Design of a Reversible ALU Based on Novel Programmable Reversible Logic Gate Structures," *IEEE International Symposium on VLSI*, 2011, pp. 126-131.

[6]   E. Fredkin and T. Toffoli, "Conservative Logic," *International Journal of Theoretical Physics*, vol. 21, 1980, pp. 219-53.

[7]   T. Toffoli, "Reversible Computing," *Technical Report MIT/LCS/TM-151*, 1980.

[8]   A. Peres, "Reversible Logic and Quantum Computers," *Physical Review*, vol. 32, iss. 6, 1985, pp. 3266-3267.

[9]   H. Thapliyal and N. Ranganathan, "Design of Efficient Reversible Binary Subtractors Based on A New Reversible Gate," *Proc. of the IEEE Computer Society Annual Symposium on VLSI*, 2009, pp 229-234.

[10]   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.

[11]   M. Morrison, M. Lewandowski, R. Meana and N. Ranganathan, "Design of a Novel Reversible ALU with Enhanced Carry Look-Ahead Adder," *To Appear in the IEEE 11th International Conference on Nanotechnology*, 2011

[12]   A. N. Al-Rabadi, "Closed-system quantum logic network implementation of the viterbi algorithm," *Facta universitatis-Ser.: Elec. Energy.*, vol. 22, no. 1, pp. 1–33, April 2009.

[13]   H. Thapliyal, N. Ranganathan, and R. Ferreira, "Design of a Comparator Tree Based on Reversible Logic," *10th Proceedings of the IEEE International Conference on Nanotechnology*, 2010, pp 1113-6.

[14]   M. Emam and L. Elsayed, "Reversible Full Adder/Subtractor," *Xlth International Workshop on Syrnbolic and Nurnerical Methods, Modeling and Applications to Circuit Design*, 2010.

[15]   P. Kogge and H. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," *IEEE Transactions on Computers*, vol. C-22, iss. 8, 1973, pp. 786-793.

[16]   M. Thomsen, R. Glück, and H. Axelsen, "Reversible Arithmetic Logic Unit for Quantum Arithmetic," *Journal of Physics A: Mathematical and Theoretical*, vol. 43, iss. 38, 2010.

[17]   Y. Van Rentergem and A. De Vos, "Optimal Design of a Reversible Full Adder," *International Journal of Unconventional Computing*, vol. 1, 2005, pp 339-355.

[18]   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.

[19]   M. Morrison and N. Ranganathan, "Design of a Moore Finite State Machine using a Novel Reversible Logic Gate, Decoder and Synchronous Up-Counter," *To Appear in the IEEE 11th International Conference on Nanotechnology*, 2011.

[20]   R. Merkle, "Two Types of Mechanical Reversible Logic," *Nanotechnology*, vol. 4, 1993, pp 114-31.

[21]   H. Thapliyal and M. Srinivas, "Novel Reversible 'TSG' Gate and Its Application for Designing Components of Primitive Reversible/Quantum ALU," *Tenth Asia-Pacific Computer Systems Architecture Conference*, 2005.

[22]   H. Thapliyal and N. Ranganathan, "Testable Reversible Latches for Molecular QCA," *IEEE Conference on Nanotechnology*, 2008, pp. 699-702

[23] H. Thapliyal and N. Ranganathan, "Design of Reversible Sequential Circuits Optimizing Quantum Cost, Delay, and Garbage Outputs," *ACM Journal on Emerging Technologies in Computing Systems*, 2010.

[24] 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.

[25] M. Morrison and N. Ranganathan, "Design of a Moore Finite State Machine using a Novel Reversible Logic Gate, Decoder and Synchronous Up-Counter," *To Appear in the IEEE 11th International Conference on Nanotechnology*, 2011.

[26] J. Smolin and D. Divincenzo, "Five Two-bit Quantum Gates Are Sufficient to Implement the Quantum Fredkin Gate," *Physical Review A*, vol. 53, 1996, pp 2855-6.

[27] 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.

[28] R. Clausius, "On Several Convenient Forms of the Fundamental Equations of the Mechanical Theory of Heat," *Philosophical Magazine and Journal of Science*, 1865.

[29] M. Planck, "On the Law of Distribution of Energy in the Normal Spectrum," *Annalen der Physik*, vol. 4, pp. 556-562, 1901.

[30] A. Einstein, "On a Heuristic Viewpoint Concerning the Emission and Transformation of Light," *Annalen der Physik*, vol. 17, 1905.

[31] J. Maxwell, "On Physical Lines of Force," *Philisophical Magazine and Journal of Science*, vol. 4, March 1861, pp. 161-175.

[32] H. Minkowski, "Space and Time," Physikalische Zeitschrift, January 1909, pp. 104-115.

[33] E. Schrödinger, "Quantization as a Problem of Proper Values. Part I.", *Annalen der Physik,* vol. 79, 1926.

[34] R. Feynman, "Space-Time Approach to Quantum Electrodynamics," *Physical Review*, vol. 76, 1949.

[35] R. Feynman, "Mathematical Formulation of the Quantum Theory of Electromagnetic Interaction," *Physical Review*, vol. 80, 1950.

[36]   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.

[37]   R. Feynman, "Quantum Mechanical Computers," *Foundations of Physics*, vol. 16, iss. 6, 1986.

[38]   D. Deutsch, "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer," *Proceedings of the Royal Society of London*, vol. 400, 1982.

[39]   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

[40]   J. von Neumann, "The First Draft Report on the EDVAC," *IEEE Annals of the History of Computing*, vol. 15, iss. 4, 1993, pp 27-75

[41]   M. Oskin, F. T. Chong, and I. L. Chuang, "A Practical Architecture for Reliable Quantum Computers ," *Computer*, vol. 35, 2002, 1993, pp 79-87.