

**Article Citation Format**  
D. Allenotor (2017): A Design Methodology for CNOT-Based Quantum Circuits. Journal of Digital Innovations & Contemp Res. In Sc., Eng & Tech Vol. 5, No. 2. Pp 107-118

**Article Progress Time Stamps**  
**Article Type:** Research Article  
**Manuscript Received:** 11<sup>th</sup> June, 2017  
**Review Type:** Blind  
**Review/Acceptance Information Sent :** 24<sup>th</sup> June, 2017  
**Final Acceptance::** 28<sup>th</sup> June, 2017  
**DOI Prefix:** 10.22624  
Series ISSN - 2488-8699

## A Design Methodology for CNOT-Based Quantum Circuits

**D. Allenotor**

Department of Mathematics/Computer Science  
Federal University of Petroleum Resources  
Effurun, Delta State, Nigeria  
[allenotor.david@fupre.edu.ng](mailto:allenotor.david@fupre.edu.ng)

### ABSTRACT

Abstract— One of the current trends in the technology of small devices manufacturing is the reduction size of computer systems. By reducing the device overall size, it is possible to fit more inside a microprocessor. This will increase the computational power of the microprocessor. The rate of progress made so far shows that this trend cannot continue forever, as it may lead to sizing the entire microprocessor into a size smaller than an atom. Quantum devices design is one of the possible promises to get around improved performance, small size, and energy efficient small devices. In this paper, the idea of size miniaturization is addressed using quantum or Control-NOT (CNOT)-based technology, as an emerging research efforts. The idea is to develop a single-gate circuitry to transform the conventional AND-OR-NOT-based circuits to CNOT-only or all-CNOT circuits. Some of the major benefits of the all-CNOT circuits include low cost of small devices design/production, circuit miniaturization, reduced power consumption, and speedup in performance as a result of reduction in propagation delay. In this paper, we define a set of local transformation rules that is applicable to the design of classical circuits and based on these local transformations, we derive the equivalent for quantum circuits.

**Keywords:** Design, Construction, Gurglar, Alarm, Security, Opto-Isolator & Infrared-Beam.



This work is licensed under **The Creative Commons Attribution 4.0** License.  
To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons  
P.O.Box 1866, Mountain View, CA 94042, USA.

## 1. INTRODUCTION

In recent years, there have been continuous increase in the volume of computation carried out using the digital computers. Also, the growth and level of precision of scientific research ranging from complex mathematical factorization problem to analysis in genetic engineering have aggravated the high speed and mini sizes of classical computers to rapidly hitting their limits [1]. Classical computers are designed using conventional AND-OR-NOT-based circuits.

But, as new technological innovations continue to replace existing ones, the need to develop computing systems (quantum computers) that are capable of withstanding the surges in high speed and small sizes is imperative. These quantum systems are expected to replace the combinational or Boolean logic circuits – a circuit put together using any or a combination of AND-OR-NOT-based elements [2].

DeMorgan’s theorem is one scheme that converts heterogeneous circuits into the corresponding homogenous circuits. The resultant circuit is smaller in size and cost less to develop. However, the circuit is does not have the advantages of quantum computer system.

The main purpose of this term project is to introduce the logic synthesis for Quantum Boolean Circuits (QBC). The key difference [2] between the conventional and quantum circuits [16] is noticed in the base family of the logic gates. For the QBC, the design is based on the use of a C-NOT type of gates as the sole elements. A second difference is that the QBC design is based on only straight-line parallel wiring. A controlled-NOT gate is meant to be a reversible operation which, for ordered pairs of bits  $(a, b) \in \{0,1\}^2$  performs the operation  $CNOT(a, b) = (a, b \oplus a)$  where conventionally the control is represented by the first bit and the target by the second. Reversible operations such as CNOT [12] can be lifted to distributions over inputs by describing it as a permutation of computational basis states. In particular, since it can be mapped that:

$$\begin{aligned} (0,0) &\xrightarrow{CNOT} (0,0) \\ (0,1) &\xrightarrow{CNOT} (0,1) \\ (1,0) &\xrightarrow{CNOT} (1,1) \\ (1,1) &\xrightarrow{CNOT} (1,0) \end{aligned}$$

the permutation matrix which we use to represent CNOT is

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

It is easy to verify that this (and all other permutation matrices) are unitary. This is the operation which transforms the state of the first two qubits. The state of the two qubits [8] just before the CNOTCNOT gate is given by:

$$\begin{aligned} |\psi\rangle &= (H \oplus \mathbf{1}) |0\rangle|0\rangle \\ &= \frac{1}{\sqrt{2}} [|0\rangle|0\rangle + |1\rangle|0\rangle] \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \end{aligned}$$

Applying the action of the CNOT, the following can be obtained:

$$\begin{aligned}
 |\psi\rangle &= \text{CNOT}|\psi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\
 &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} [ |0\rangle|0\rangle + |1\rangle|1\rangle ]
 \end{aligned}$$

In this case, it is clear that the CNOT can be applied on the standard basis states  $|0\rangle|0\rangle$  and  $|1\rangle|0\rangle$ . This is made possible because of the linearity property of the CNOT operation.

In this project paper, a practical method is developed for quantum circuits. The developed quantum scheme is obtained using local transformation of Boolean logic circuits. The final quantum circuit makes the most efficient operation sequence for implementing logic functions in quantum computation. In this paper, a practical method is developed for quantum circuits. The developed quantum scheme is obtained using local transformation of Boolean logic circuits. The final quantum circuit makes the most efficient operation sequence for implementing logic functions in quantum computation. The novelty of this paper is the application of the existing set of available quantum transformations to actualize performance-intensive design for small devices. The overall objective is to achieve cost-reduction and power-efficient design.

The remaining parts of the article organized as follows. Section 2 reviews relevant literature. Supportive definitions for quantum mechanics is provided in Section 3. The relationship between Quantum Computing and Boolean operations that drives combinatorial circuits is provided in Section 4. A practical approach as a proof of concept and to show that the concept is realization is given in Section 5. Section 6 concludes the paper.

## 2. RELATED WORKS

A new methodology for logic synthesis for Quantum Boolean Circuits (QBC) is introduced in this paper. The key difference [2] between the conventional and quantum circuits is noticed in the base family of the logic gates. For the QBC, the design is based on the use of a C-NOT type of gates as the sole elements. A second difference is that the QBC design is based on only straight-line parallel wiring. Some schemes that are used for the evaluation of Boolean (algebraic) expression include algebraic methods [17], Karnaugh map [15], and Quine-McCluskey [14]. These two schemes have drawbacks in that they are intuitive and not programmable. However, Quine-McCluskey [14] is capable of automation but the overhead in terms of design is high. These schemes lack adequate support as a tool for the design of quantum circuits.

To discuss the design a small QBC, the quantum algorithm can be divided into two parts -- the quantum specific part, for example, Hadamard transformation for making a quantum superposition and the Quantum Fourier transformation [6]. The second design applies the quantum Boolean oracles [5], to obtain (conventional) Boolean functions. The purpose of this report is to use the rule set for the development of a design theory for quantum circuit whose Boolean logical part is implemented using the Control-NOT (C-NOT) circuit with a view to reducing the overall cost of design. The required operations needed to perform a quantum algorithm, required unitary operations should be expressed as a sequence of the basic operations, which can be implemented by a quantum computer.

It is proved that the combination of one-qubit operations and controlled-NOT (C-NOT) operations can make any unitary operations [1], and the implementation of these basic operations is the least requirement for a quantum system to be a quantum computer. Several methods have been suggested to get an operation sequence, or a quantum circuit, consisting of the basic quantum gates for a given unitary operation [1] and [4]. These methods are general, but they do not necessarily generate the most efficient operation sequence from the point of view of experimental implementation.

The evaluation of a binary or logic function, which is the very basic task in classical computation, is also frequently required in the quantum algorithms such as search and Deutsch algorithms [5] and [6]. One of the results of studies on quantum computation is that a quantum computer can calculate any function, which can be calculated by a classical computer [7]. Therefore, it is expected that a function can be evaluated in a quantum computer through a quantum combinational logic circuit corresponds to a classical combinational logic circuit. A quantum combinational logic circuit can be constructed with basic quantum logic gates such as quantum NOT, called C-NOT and Toffoli gates [13] as a classical combinational logic circuit is constructed with NOT, AND, and OR gates. An efficient method of making a well-designed quantum circuit for a logic function will be very useful because the successful implementation of a quantum algorithm depends heavily on the number of basic gates composing the quantum circuit due to the finite coherence time.

### 3. QUANTUM COMPUTING - DEFINITIONS

The following basic definitions provided in [16] forms the basis of transformation rules that is applied to the local transformation rules for CNOT gates. In addition to these definitions, some commentaries relating to their applicability to quantum circuit design is given.

- a) Definition 1. A Control-NOT gate is denoted by  $[t, C]$ , where  $t$  is an integer and  $C$  is a finite set of integers. This definition for CNOT is constrained by the condition that  $(t \in C) \cdot |x_k\rangle$  is called the target bit and  $|x_k\rangle$  is called the control bit provided that  $(k \in C)$  condition is satisfied. Using this definition to analyze the Quantum Basic Circuits (QBC) in Figure 1, the leftmost CNOT  $[n+1, \{1, 2, n\}]$  this is identical to  $[t, C]$ . The effect of this CNOT gate is that it changes the  $|x_{n+1}\rangle$  into  $|x_{n+1} \oplus f(x_1 \dots x_n)\rangle$  state.
- b) Definition 2. A QBC of size  $M$ -over qubits [8]  $|x_1 \dots x_n\rangle$  is a sequence of CNOT gate. This definition provides a basis for comparison of quantum and conventional circuit model. For any given sequence of CNOT gates,  $[t_1, C_1], [t_2, C_2] \dots [t_m, C_m]$  where  $1 \leq t_i \leq N, C_i \in \{1, 2, \dots, N\}$
- c) Definition 3. The state of the circuit after  $i$ -th CNOT gate  $[t_i, C_i]$  is denoted by  $S_i = |x_1^i\rangle \dots |x_n^i\rangle$ .
- d) Definition 4. A QBC is said to be proper and to compute a Boolean function  $f(x_1, x_2, \dots, x_n)$  iff.  $S_0 = |a_1\rangle |a_2\rangle \dots |a_{n+1}\rangle |0\rangle \dots |0\rangle$ . This means that all the auxiliary bits are cleared at the beginning of computation.

#### 4. QUANTUM BOOLEAN CIRCUITS

Quantum computation engages a *quantum bit*, or *qubit* that plays a very significant role in the analysis and design of quantum circuits. [10] proposes a physical realization of these qubits and is given by  $|x\rangle = \alpha|0\rangle + \beta|1\rangle$ , where  $|x\rangle$  means the quantum state of the qubit  $x$ , and  $\alpha$  and  $\beta$  are complex quantities that satisfy  $|\alpha|^2 + |\beta|^2 = 1$  and the measured quantum state gives  $|0\rangle$  and  $|1\rangle$  with probabilities  $|\alpha|^2$  and  $|\beta|^2$  respectively. Thus, the qubit can be in a superposition of the states  $|0\rangle$  and  $|1\rangle$ . This is a major distinguishing factor of the quantum circuits from the conventional circuits. In quantum computing, unitary operator is applied to the qubits to obtain the final quantum state. The unitary operations applied to these quantum states constitute a Quantum Boolean Circuit (QBC). Consider the QBC as a quantum system with  $N$  qubits given by  $|x_1\rangle|x_2\rangle \cdots |x_N\rangle$ .

The resultant combinational logic circuits obtained have heterogeneous characteristics (since the complete circuit is built from different logic gates). Heterogeneous circuits are difficult to trouble shoot have high-energy demand. Also, the propagation delay is high and hence, the speed of such circuits is slow. Above all, since the heterogeneous circuits are composed of different gates, the cost (in terms of number of components) of production is high and computer systems developments based on this type of technology do not compete favorably with those developed from homogenous circuits. Both the heterogeneous and homogeneous classical circuits [3], [4], and [5] have the following basic characteristics:

- i. Classical circuit behavior is governed implicitly by classical physics.
- ii. Signal states are simple bit vectors, e.g.,  
 $\mathbf{x} = 11011101101$
- iii. Boolean algebra defines operations -  
 $\mathbf{x} = \mathbf{A}\bar{\mathbf{B}} + \mathbf{C}\mathbf{D}$ , where  $\bar{\mathbf{B}}$  is implemented using the NOT gate.
- iv. Small well-defined sets of universal gate types, e.g., AND, OR, NOT or their corresponding complements such as NAND, NOR, EXOR, COINCIDENCE.
- v. Circuits are easily implemented in fast, scalable and macroscopic technologies such as Complementary Metal Oxide Semi-conductor (CMOS).

On the other hand, quantum logical circuits are associated with the following characteristics [6]:

- i. Signal states are vectors interpreted as a superposition of binary “qubit” vectors with complex-number coefficients such that:

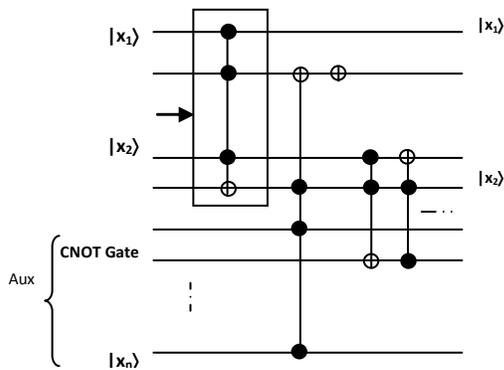
$$|\psi\rangle = \sum_{i=0}^{2^N-1} C_i |i_N i_{N-1} \cdots i_0\rangle \quad (1)$$

- ii. Operations are defined by linear algebra over Hilbert Space and can be represented by unitary matrices with complex elements.
- iii. Many universal gate sets exist but the best types are not obvious.
- iv. Circuits use microscopic technologies that are slow and fragile.

DeMorgan's theorem [17] is one scheme that converts heterogeneous circuits into the corresponding homogenous circuits. The resultant circuit is smaller in size and cost less to develop. However, the circuit is does not have the advantages of quantum computer system.

### 4.1 Quantum Boolean Circuits

In quantum computing, a quantum bit, or qubit, plays a very significant role in the analysis and design of quantum circuits. [10] proposes a physical realization of these qubits and is given by  $|x\rangle = \alpha|0\rangle + \beta|1\rangle$ , where  $|x\rangle$  means the quantum state of the qubit  $x$ , and  $\alpha$  and  $\beta$  are complex quantities that satisfy  $|\alpha|^2 + |\beta|^2 = 1$  and the measured quantum state gives  $|0\rangle$  and  $|1\rangle$  with probabilities  $|\alpha|^2$  and  $|\beta|^2$  respectively. Thus, the qubit can be in a superposition of the states  $|0\rangle$  and  $|1\rangle$ . This is a major distinguishing factor of the quantum circuits from the conventional circuits. In quantum computing, unitary operator is applied to the qubits to obtain the final quantum state. The unitary operations applied to these quantum states constitute a Quantum Boolean Circuit (QBC). Iwama et al. [2] define QBC as a quantum system with  $N$  qubits given by  $|x_1\rangle|x_2\rangle\cdots|x_N\rangle$ .



**Figure 1: Quantum Boolean Circuit.**

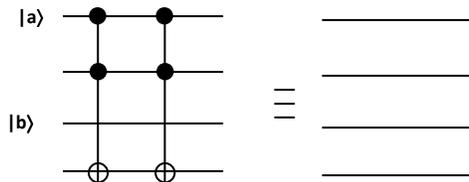
Figure 1 shows the structure of a QBC where  $|x_1\rangle|x_2\rangle\cdots|x_n\rangle$  are the inputs and the  $(n+1)$ -st qubit  $|x_{n+1}\rangle$  is the work bit of the QBC. The work bit takes the form of  $|x_{n+1}\rangle \oplus f(x_1, \dots, x_n)$ . This form is used to obtain the Boolean function  $f$ . During quantum circuit design, the auxiliary bits have a high being entangled, to circumvent the entanglement condition, all the auxiliary qubits are set and reset to  $|0\rangle$ .

### 4.2 Local Transformation Rules

Given any two classes of circuits - combinational or classical circuits (SC) derived from a Boolean expression and quantum circuits (SQ), We want to show that a certain relation exists that transforms SC to SQ. Where the circuits obtained from SQ are a minimized form of SC. Iwama et al. [2] give the following local transformation rules.

The basic idea that governs local rule transformation from circuit 1 (C1) to circuit 2 (C2) is based on the condition that if C1 and C2 are equivalent, then  $C1 \Leftrightarrow C2$  follows a simple transformation since  $C1 \Leftrightarrow S1 = S2 \Leftrightarrow C2$ .

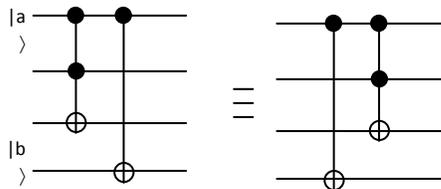
*Rule 1:* Iwama et al. [2] gave that any two adjacent logic gates will cancel out. i.e.,  $[t_i, C_i]. [t_i, C_i] \Leftrightarrow \epsilon$ . Where  $i$  is identical for any transformation. This rule can be diagrammed in a logical sense as shown in Figure 2.



**Figure 2: Diagrammatic Representation of Rule 1.**

Logically,  $a\bar{a}b\bar{a}b = d$ . The circuit on the left is transformed to the one on the right by the application of Rule 1. The fundamental concerns of circuit miniaturization are achieved (reduced size and power consumption).

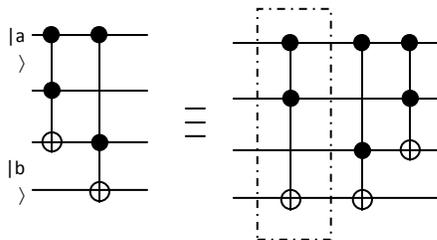
*Rule 2:* The application of Rule 2 follows the commutative principle in linear algebra with a constraint that  $t_1 \in C_2$  and  $t_2 \in C_1$ . It however guarantees that the circuit on the left-hand side and that on the right-hand side are equivalent. According to Iwama et al. [2] the placement of basic elemental components in a circuit follows a commutative approach.



**Figure 3: Diagrammatic Representation of Rule 2.**

*Rule 3:* Iwama et al. [2] gave Rule 3 as follows:

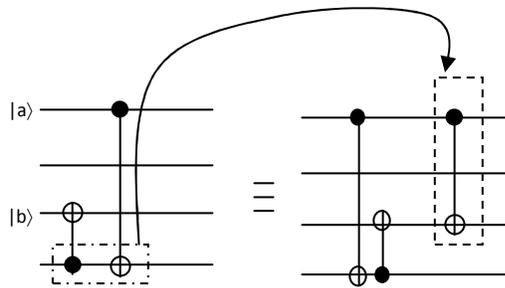
if  $t_1 \in C_2$  and  $t_2 \in C_1$ , then  $[t_1, C_1]. [t_2, C_2] \Leftrightarrow [t_2, C_2]. [t_1, C_1]. [t_1, C_1 \cup C_2 - \{t_2\}]$ . The logical circuit that illustrates this rule is given in Figure 3.



**Figure 4: Diagrammatic Representation of Rule 3.**

*Rule 4:* Iwama et al. [2] gave Rule 4 as follows:

$[t_1, C_1]. [t_2, C_2] \Leftrightarrow [t_2, C_1 \cup C_2 - \{t_1\}]. [t_2, C_2]. [t_1, C_1]$  if  $t_1 \in C_2$  and  $t_2 \in C_1$ . This rule is the dual of Rule 3. It provides a relationship between two gates that are opposite. The logical circuit that illustrates this rule is given in Figure 4.

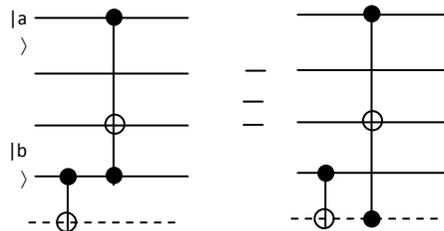


**Figure 5: Diagrammatic Representation of Rule 4.**

*Rule 5:* Iwama et al. [2] gave Rule 5 as follows:

$[t_1, C_1]. [t_2, C_2] \Leftrightarrow [t_1, \{C_1\}]. [t_2, C_2 \cup C_1 - \{t_1\}].$

Rule 5 follows a condition that  $t_1$  must be greater than  $n+1$  and that there is no  $CNOT_{t_1}$  before  $[t_1, \{C_1\}]$ . The logical circuit given in Figure 6 explains Rule 5.



**Figure 6: Diagrammatic Representation of Rule 5.**

In Figure 6,  $|0\rangle$  is called auxiliary qubit and initialized to  $|0\rangle$ . The output is given as:

□□□□□□□□□□□□□□□□

*Rule 6:* Iwama et al. [2] expression of Rule 6 shows that whenever there is an integer  $t$  such that there is no  $CNOT_t$  before  $[t, C]$  then the circuit does not really exist. According to Iwama et al. [2],  $[t, C] \Leftrightarrow \epsilon$ .

### 5. PRACTICAL APPROACH

To model a quantum circuit from a corresponding Boolean logic circuit (classical circuit), consider Equation (2), a Karnaugh Map (K-map) method for minimizing Boolean expression.

Given that:

$$f(a, b, c) = \sum_{i=0}^7 M_i(3,4,5,6) + d(7) \quad (2)$$

where d is a case for “don’t care”. A don’t care condition [8] is analogous to the use of a catalyst to aid chemical reaction. They only help in circuit minimization through a combinational approach. A combinational logic operation is basically the operation performed by a combinational logic circuit that accepts one or more inputs and produces only one output as in  $f:\{0,1\}^n \rightarrow \{0,1\}$ .

It is required to obtain a Boolean circuit and transform it to its equivalent quantum circuit. For this problem, a K-map realization gives the following:

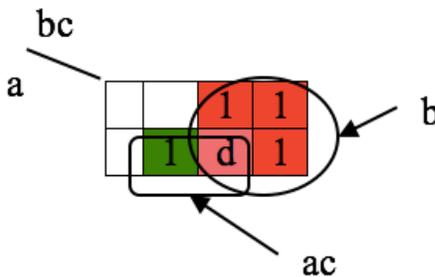


Figure 7: K-Map realization of  $f(a, b, c) = \sum_{i=0}^7 M_i(3,4,5,6) + d(7)$ .

From Figure 7, the minimized Boolean expression corresponding to  $f(a, b, c) = \sum_{i=0}^7 M_i(3,4,5,6) + d(7)$  is given by  $b + ac$ . This Boolean expression can be realized using an AND-OR implementation as follows.

AND

OR

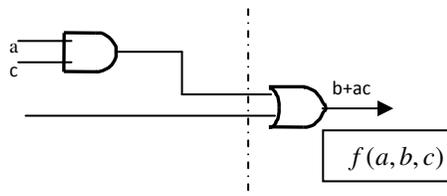
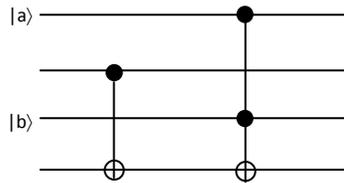


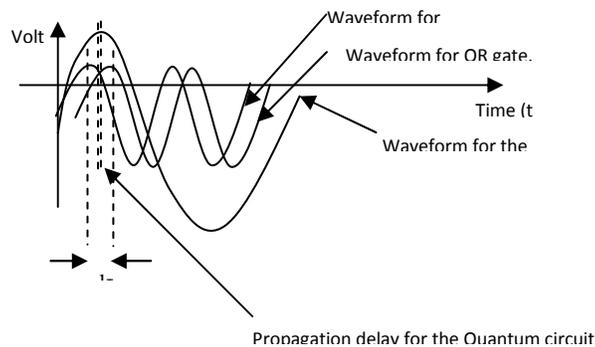
Figure 8: Minimized Logic Circuit realization for  $f(a,b,c)$ .

The equivalent quantum circuit in Figure 8 corresponds to the minimized classical logic circuit in Figure 9.



**Figure 9: Equivalent Quantum Circuit.**

**Circuit Analysis:** From the circuit of Figure 9, we observe that its comprised of two different logic gates - AND and OR. These two logic elements have different propagation delay ( $\tau_D$ ) - the time required by an operation to get to completion. Since these logic elements are put in cascade, a possibility exists that their respective  $\tau_D$  add up and makes the performance too slow. To circumvent the effects of this delay property, a quantum circuit (Figure 9) is obtained from the minimized logic circuit. The quantum circuit is comprised of the same type of elements of all CNOT (homogeneous). This makes the circuit less cost intensive to build, fast responds speed and above all, occupies less space. The waveform of Figure 10 describes the relative speedup of the two circuits of our analysis.



**Figure 10: Wave Form analysis of Classical and Quantum circuits.**

$\tau_T = \tau_{AND} + \tau_{OR}$  is the sum of the delay due to AND and OR gates. In the case of the quantum circuit,  $\tau_T$  sums up like parallel addition. That is, the sum of the propagation delay is maintained low.

## 6. CONCLUSION

The circuit miniaturization process using local transformation rule obtains quantum equivalent circuits of the given classical circuits. Although the resultant compact sized quantum circuits consume less power, faster, and cost less to develop, the circuit will no doubt face other problems that will form the frontier for wider research area. The compact sized circuits will become too complex to diagnose and troubleshoot. Hence maintainability becomes non-trivial. Also, the mean time to fail (MTTF) and Mean time between fails (MTBF) will be high for a complex circuit. Therefore, the pros and cons of miniaturization should be adequately weighed before embarking on the venture. For example, it is computationally intractable to allow one computation to be cascaded with another, because in general, cascading two unitary transforms results in a new transform with unrelated eigenvalues. Future research directions will look at parallelism in this regard.

## REFERENCES

- [1] Krishna V. Palem. Energy Aware Algorithm Design via Probabilistic Computing: From Algorithms and Models to Moore's Law and Novel (Semiconductor) Devices, in Proceedings of the international conference on Compilers, architectures and synthesis for embedded systems, 2003, pp. 113-116.
- [2] Kazuo Iwama and Yahiko Kambayashi and Shigeru Yamashita. Transformation Rules for Designing CNOT-Based Quantum Circuits. In Proceedings of the 39th conference on Design automation, 2002, pp. 419-424.
- [3] Massoud Pedram. Power minimization in Integrated Circuit Design: Principles and Applications, Vol. 1, No. 1. *ACM Trans. Des. Autom. Electron. Syst.* 1996, pp. 3-56.
- [4] John L. Fike. Predicting Fault Detectability in Combinational Circuits - A New Design Tool. In Proceedings of the 12th design automation conference. 1975, pp. 290-295.
- [5] Li-Pen Yuan and Sung-Mo Kang. A Sequential Procedure for Average Power Analysis of Sequential Circuits. In Proceedings of the 1997 international symposium on Low power electronics and design. 1997, pp. 231-234.
- [6]. Vivek V. Shende, Aditya K. Prasad, Igor L. Markov, and John P. Hayes. Reversible Logic Circuit Synthesis. In Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design. 2002, pp. 353-360.
- [7] John P. Hayes. Tutorial: Basic Concepts in Quantum Circuits. In Proceedings of the 40th conference on Design automation. 2002, pp. 893-893.
- [8]. J. T. Chu. Some Methods for Simplifying Switching Circuits Using "Don't Care" Conditions. *Journal of ACM*. Vol.8, No. 4. 1961, pp. 497-512.
- [9] Jennifer Carini. A Simulation of Quantum Logic Gates and Qubits Using Ruby. *J. Comput. Sci. Coll.* 19, 5 (May 2004), 2004, pp. 337-338.
- [10] Siddhartha Kasivajhula. Quantum Computing: A Survey. In *Proceedings of the 44th annual Southeast regional conference (ACM-SE 44)*. ACM, New York, NY, USA, 2006, pp. 249-253.
- [11] Andris Ambainis, Leonard J. Schulman, and Umesh Vazirani. Computing with Highly Mixed States. *J. ACM* 53, 3 (May 2006), pp. 507-531.
- [12] Anderson Braga de Avila, Renata Hax Sander Reiser, Adenauer Correa Yamin, and Mauricio Lima Pilla. Scalable Quantum Simulation by Reductions and Decompositions through the Id-Operator. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC '16)*. ACM, New York, NY, USA, 2016.

- [13] Daniel Grobe, Xiaobo Chen, Gerhard W. Dueck, and Rolf Drechsler. 2007. Exact Sat-Based Toffoli Network Synthesis. In *Proceedings of the 17th ACM Great Lakes symposium on VLSI (GLSVLSI '07)*. ACM, New York, NY, USA, 2007, pp. 96-101.
- [14] Tarun Kumar Jain, D. S. Kushwaha, and A. K. Misra. Optimization of the Quine-McCluskey Method for the Minimization of the Boolean Expressions. In *Proceedings of the Fourth International Conference on Autonomic and Autonomous Systems (ICAS '08)*. IEEE Computer Society, Washington, DC, USA, 2008, pp. 165-168.
- [15] Ali M.A. Rushdi and Motaz H. Amashah. Using Variable-Entered Karnaugh Maps to Produce Compact Parametric General Solutions of Boolean Equations. *International Journal of Computer Mathematics*. (July 14, 2011), 2011, pp. 3136-3149.
- [16] Michael A. Nielsen & Isaac L. Chuang, *Quantum. Computation and Quantum Information*, 10th Anniversary Edition Press, 2010. Cambridge University Press.
- [17] Charles H. Roth, Jr. and Larry L. Kinney, *Fundamentals of Logic Design*, Cengage Learning. 7<sup>th</sup> Ed. 2014.