Scientific Journal of Impact Factor(SJIF): 3.134

e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 6, June -2015

QUANTUM CIRCUIT OPTIMIZATION (FOR 3 QUBITS)

Yogesh B. Patel[1], Harikrishna B. Jethva[2]

Computer Department,
L. D. College of Engineering, Gujarat,

Abstract — *Logic gates like AND, OR, XOR are example of irreversible gates for which no input is calculated from each output. But in case of reversible gates like NOT, CNOT, HADAMARD, TOFFOLI, V, V+, etc. each input is calculated from each output, because no energy(information) loss is the best feature of reversible gates. And hence more researchers focus on reversible circuits, which are used in quantum circuits or networks. Quantum computing is a process that incorporates interacting physical systems that represent quantum bits and quantum gates. We present the quantum bit(qubit), the quantum register and the quantum gates. The qubit is described as a vector in a two dimensional Hilbert space and the quantum register, which comprises a number of qubits, as a vector in a multidimensional Hilbert space. Any gates transform the input bits to the output bits in some deterministic fashion according to the definition of the logic gate. Quantum gates are Hilbert space operators that rotate the qubit or the quantum register vectors. We present the currently used two optimization methods one is template based and second is window based for reversible and quantum circuits. Finally, we have implemented our method using the benefits of above two. Quantum circuit optimization is useful to reduce the quantum cost for those circuits, where optimization is applicable. Response time is reduced for complex computations.*

Keywords- *quantum bit, quantum register, quantum circuit, quantum cost, reversible gate, irreversible gate*

I. INTRODUCTION

"In about ten years or so, we will see the collapse of Moore's Law. In fact, already, we see a slowing down of Moore's Law," said by world-renowned physicist, Michio Kaku. He also said: "Computer power simply cannot maintain its rapid exponential rise using standard silicon technology, this law's collapse due to heat and leakage issues." He represented possible alternatives to the demise of Moore's Law: protein computers, DNA computers, optical computers, quantum computers and molecular computers.[15]

"Quantum computers can efficiently render every physically possible quantum environment, even when vast numbers of universes are interacting. Quantum computation is a qualitatively new way of harnessing nature," according to David Deutch, an Israeli-British physicist at the University of Oxford who pioneered the field of quantum computation and is a proponent of the many-worlds interpretation of quantum mechanics. Quantum computers, says Deutch, have the potential to solve problems that would take a classical computer longer than the age of the universe.

Quantum technologies offer new dimensions in field of computation and also in field of communication. Quantum technology and classical silicon technology are very different than each other. In case of Quantum computers, they have quantum bits (qubits) to store information. The behaviour of each qubit is governed by the laws of quantum mechanics, enabling qubits to be in a "superposition" state, that is, both a 0 and a 1 at the same time, until an outside event causes it to "collapse" into either a 0 or a 1.

In quantum computing to perform each operation special quantum circuits are designed. Now to get better performance like good throughput, fast response, etc… we can redesign these circuits. This redesign is done by either rearrangement and/or removal of unnecessary quantum gates. We can refer this redesigning process as an optimization process. As per our study, currently two optimization methods are available for quantum c ircuits. Among them first is template matching, second is window based optimization. Each has their own special technique or algorithm for optimization. Each does good job also.

Finally, we aimed to find out the best optimal approach/algorithm using the subparts of available optimizing techniques. We can apply this algorithm on any quantum combinational circuit having 3 qubits.

II. QUANTUM GATES

Each quantum gate has its own matrix representation. For a given circuit, we perform tensor product between the gates and this tensor product is nothing but it's a matrix multiplication. Consider an example of 5 gates. So we have 5 different matrices. Number of matrix multiplication are performed between these gates. A final output represents original circuit. Qubits are applied to this final matrix rather than applying them to individual member gates of circuit.

2.1 NOT GATE

Matrix Representation

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

Symbol

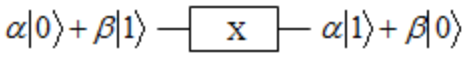


Figure 1. Not Gate

2.2 Z GATE

Matrix Representation

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Symbol

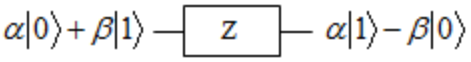


Figure 2. Z Gate

2.3 V GATE

Matrix Representation

$$V = \frac{1+i}{2}\begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}$$

Symbol

Figure 3. V Gate

2.4 V+ GATE

Matrix Representation

$$V = \frac{1+i}{2}\begin{bmatrix} -i & 1 \\ 1 & -i \end{bmatrix}$$

Symbol

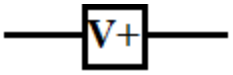


Figure 4. V+ Gate

2.5 HADAMARD GATE

Matrix Representation

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Symbol

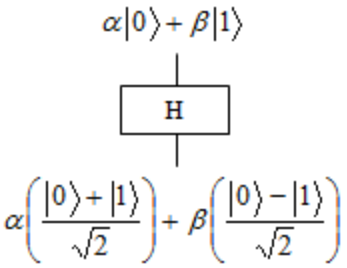


Figure 5. Hadamard Gate

2.6 Controlled SWAP GATE(For 3 Qubit)

Matrix Representation

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

Symbol

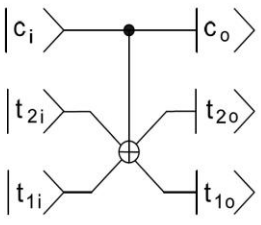


Figure 6. CSWAP Gate

2.7 Controlled NOT GATE

Matrix Representation

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

Symbol

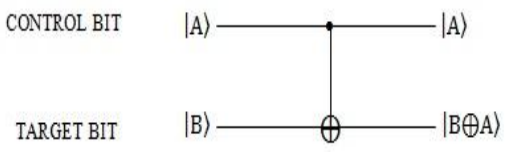


Figure 7. CNOT Gate

2.8 Controlled Z GATE

Matrix Representation

$$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Symbol

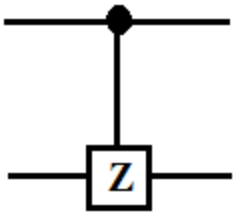


Figure 8. CZ Gate

2.9 Controlled V GATE

Matrix Representation

$$CV = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1+i}{2} & \frac{1-i}{2} \\ 0 & 0 & \frac{1-i}{2} & \frac{1+i}{2} \end{bmatrix}$$

Symbol

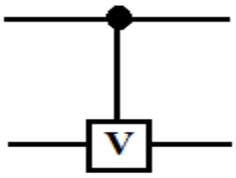


Figure 9. CV Gate

2.10 Controlled V+ GATE

Matrix Representation

$$CV+ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1-i}{2} & \frac{1+i}{2} \\ 0 & 0 & \frac{1+i}{2} & \frac{1-i}{2} \end{bmatrix}$$

Symbol

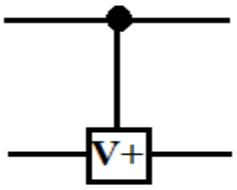


Figure 10. CV+ Gate

III. GATE PROPERTIES

3.1 For some gates, when we place two same gates in adjacent, their combination gives identity matrix. Like Hadamard gate, Not gate, Z gate, Swap gate.

3.2 For some gates, when we place two same gates in adjacent, their combination gives some third gate's matrix. Like V gate, V+ gate.

V x V = NOT

V^+ x V^+ = NOT

3.3 For some gates, when we place two different gates in adjacent, their combination gives some third gate's matrix. Like, V and V+ gates give identity matrix in output.

V x V+ = Identity

V and NOT gates give identity matrix in output.

V x NOT = V+

V+ and NOT gates give identity matrix in output.

V+ x NOT = V

IV. TEMPLATES

Templates are such circuits, which generates identity matrix. So, whenever we find any template in circuit then we can remove it completely. Following table provides list of templates.

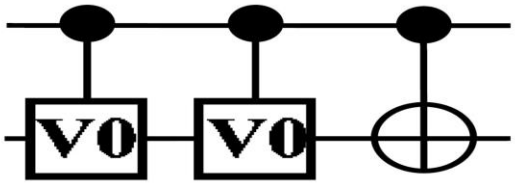


Figure 11. Template1

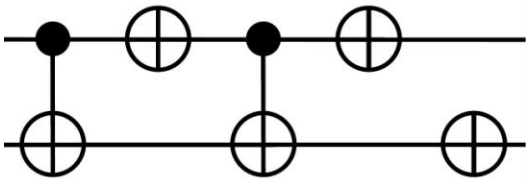


Figure 12. Template2

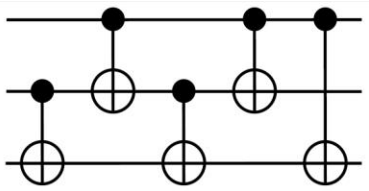


Figure 13. Template3

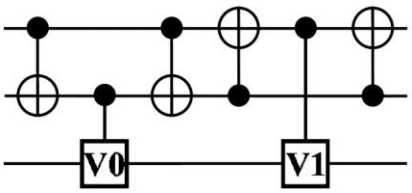


Figure 14. Template4

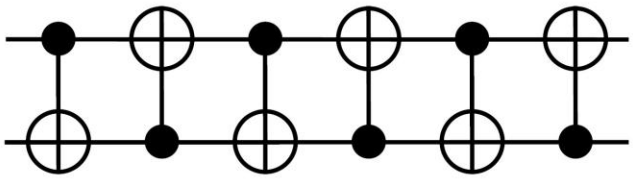


Figure 15. Template5

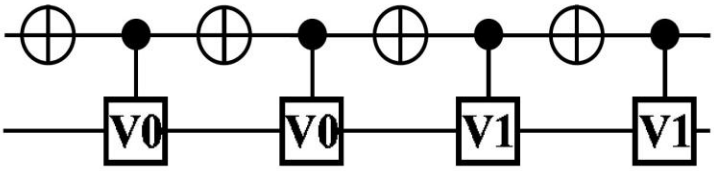


Figure 16. Template6

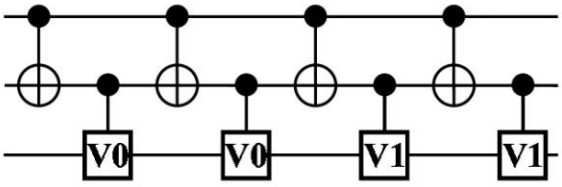


Figure 17. Template7

4.1 Moving Rule

Each circuit may not contain template, as it is. But it may possible after applying few rearrangements of gates, we can find out presence of one or more templates in rearranged circuit. For Rearrangement, we have to follow moving rule. Assuming gate A has control set C_A (C_A is an empty set in the case of an uncontrolled gate) and target T_A and gate B has control set C_B and target T_B, these two gates form a moving rule if, and only if, T_A NOT $\subseteq C_B$ and T_B NOT $\subseteq C_A$.

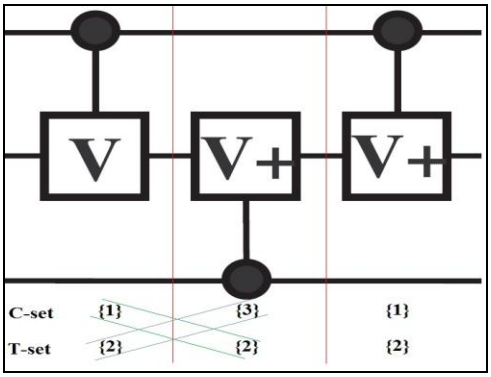


Figure 18. Before moving rule

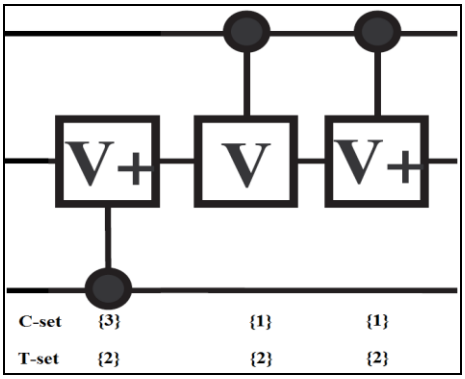


Figure 19. After moving rule

Using moving rule, we can perform rearrangement of gate within circuit.

V. ALGORITHM

We have developed an algorithm which utilizes the moving rule to rearrange gates. Next, gate properties and temples are used to remove the unnecessary gates from the circuit. Before looking at algorithm, we visit first matrix generation algorithm for individual gate of circuit.

5.1 Matrix Generation Algorithm

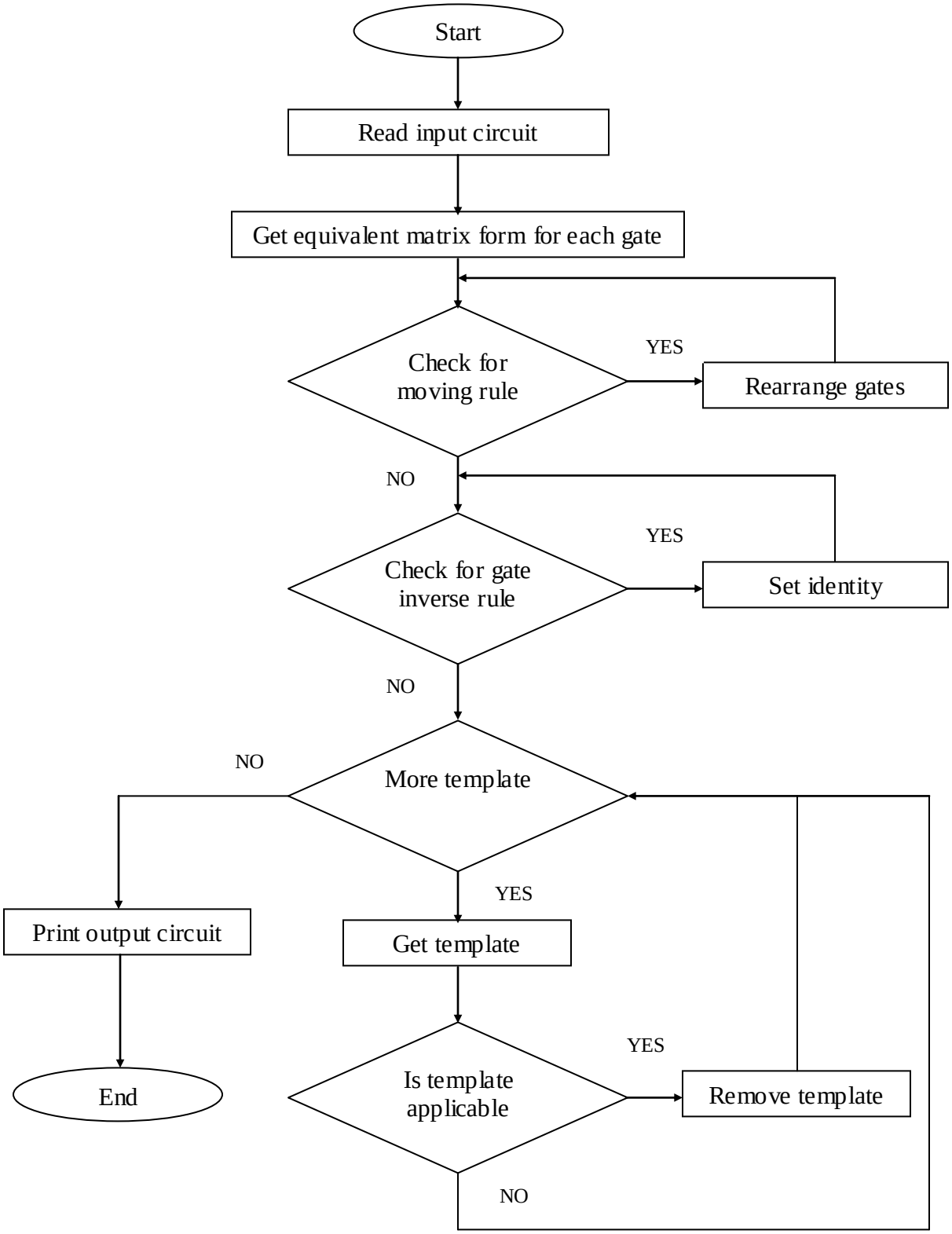
```
INPUT: BASE_MATRIX for particular gate, OUTPUT_MATRIX with each element 0
OUTPUT: OUTPUT_MATRIX with elements mapped from BASE_MATRIX
ALGORITHM_BEGIN
Total_No = total number of lines in circuit
Target_Line = line number where gate is positioned
Difference = Total_No – Target_Line
Total_repeatition = 1 << (Target_Line – 1)
Gap = 1 << Difference
FOR m = 0 to Total_repeatition increment m by 1
    MULTIPLIER = 1 << (Difference + 1)
    MULTIPLIER = MULTIPLIER * m
    FOR k = 0 to Gap increment k by 1
            INDEX = MULTIPLIER + k
            FOR i = 0 to 2 increment i by 1
                    FOR j = 0 to 2 increment j by 1
                            IF i = 0 AND j = 0
                                    OUTPUT_MATRIX (INDEX, INDEX) = BASE_MATRIX(0,0)
                            IF i = 0 AND j = 1
                                    OUTPUT_MATRIX (INDEX, INDEX + Gap) =BASE_MATRIX(0,1)
                            IF i = 1 AND j = 0
                                    OUTPUT_MATRIX (INDEX + Gap, INDEX) =BASE_MATRIX(1,0)
                            IF i = 1 AND j = 1
                                    OUTPUT_MATRIX (INDEX + Gap, INDEX + Gap)  =
                                                                    BASE_MATRIX(1,1)
                    END FOR
            END FOR
    END FOR
END FOR
END
```

5.2 Optimization Algorithm

```
INPUT: Circuit
OUTPUT: Optimized Circuit
ALGORITHM_BEGIN
Step 1: Calculate quantum cost of input circuit.
Step 2: Get equivalent matrix form for each gate.
Step 3: Check and Apply moving rule till it applicable in circuit.
Step 4: Check and Apply gate-inverse rule till it applicable in circuit.
Step 5: For each template
            IF template is found in circuit THEN
                    Remove particular template from circuit
Step 6: Calculate Quantum Cost
```


END

5.3 Flow Chart of Optimization Algorithm



5.4 Experiment and Result

We have implemented our algorithm in java.
Algorithm has complexity of O(Total_line^4).
We have tested our algorithm on 10 sample circuits. Among them, here we are presenting two of them with results.

5.4.1 Circuit 1

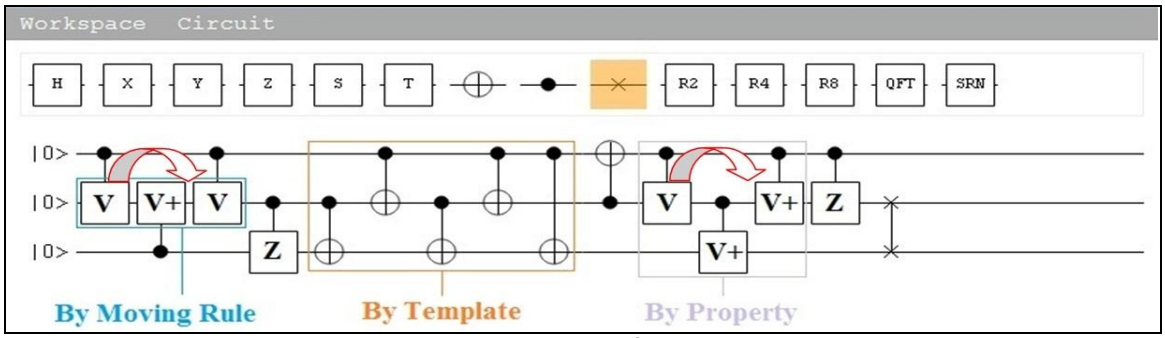


Figure 20. Circuit before optimization

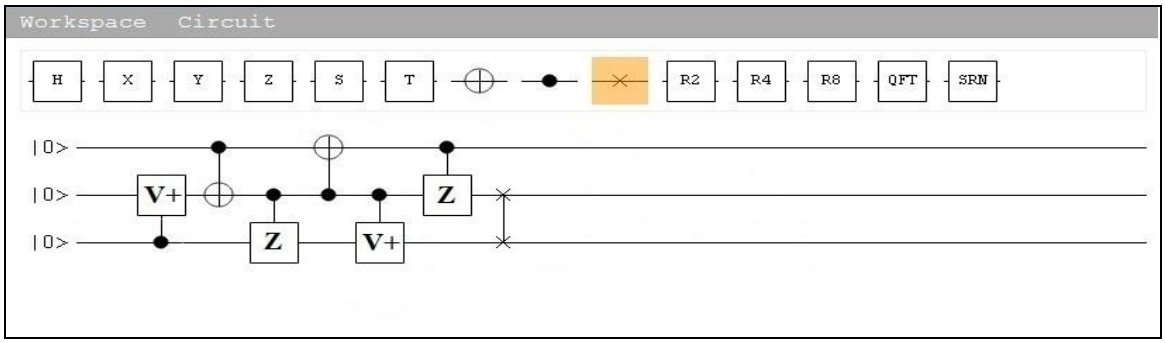


Figure 21. Circuit after optimization

5.4.2 Circuit 2 (3_17_15, Benchmark)

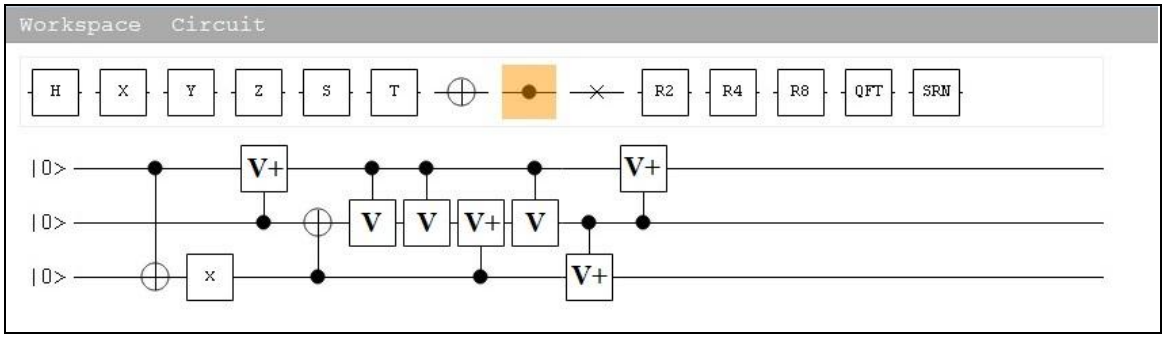


Figure 22. Circuit before optimization

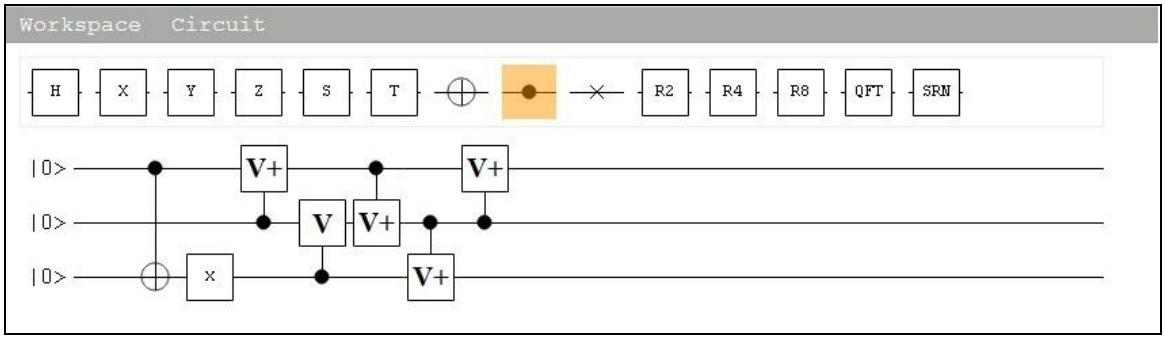


Figure 23. Circuit after optimization

CONCLUSION

As we know, optimization means try to do some better than the current. By providing greater computing capabilities, quantum computer will provide strong platform to solve those problems, which are unsolved till today. Quantum computing will create new scope for researchers and also for developers. And might be chance, we or next generation will see new look of this world. But the negative side of quantum computing is "when will quantum computers available for normal users at economic cost?"

Using presented optimization method, we can optimize quantum circuits up to 3 qubits. We can get good optimized outcomes. Hence we can reduce the quantum cost(QC) of circuit if possible. Due to the reduction in quantum cost, response time of circuits will be decreased. As soon as new templates will be invented, we can reduce quantum cost by more numbers.

REFERNCES

[1]. V. V. Zhirnov et al. **Limits to binary logic switch scaling – agedanken model**. Proc. of the IEEE, 91(11):1934.1939, Nov.2003.
[2]. R. Landauer. **Irreversibility and heat generation in the computingprocess.** IBM J. Res. AndDev., 5:183.191, 1961.
[3]. C. H. Bennett. **Logical reversibility of computation.** IBM J.Res. and Dev., 17:525.532, Nov. 1973.
[4]. M. Nielsen and I. Chuang. **Quantum Computation and QuantumInformation.** Cambridge University Press, 2000.
[5]. **IBM's test-tube quantum computer makes history.** IBM T.J. Watson Research Center, http://researchweb.watson.ibm.com/resources/news/20011219 quantum.shtml, Dec. 2001.
[6]. D. Maslov, D. M. Miller and G. W. Dueck.**Templatesfor Reversible Circuit Simplification.**Communications, Computers and signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference, 10.1109/PACRIM.2005.1517363.
[7]. Dmitri Maslov, and Gerhard W. Dueck, Member, IEEE.**LEVEL COMPACTION IN QUANTUM CIRCUITS** IEEE Congress on Evolutionary Computation, 2006.
[8]. Paul Isaac Hagouel and Ioannis G. Karafyllidis.**Quantum Computers: Registers, Gates and Algorithms.** PROC. 28th INTERNATIONAL CONFERENCE ON MICROELECTRONICS 2012.
[9]. D.Maslov, D. M. Miller, G. W. Dueck and Camille Negrevergne.**Quantum Circuit Simplificationand Level Compaction.** IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, 2008.
[10]. ANDREI BÃUTU1, ELENA BÃUTU.**Quantum Circuit Designby Means of Genetic Programming.** Rom. Journ. Phys., 2007.
[11]. Mathias Soeken, Robert Wille, Gerhard W. Dueck and Rolf Drechsler.**Window Optimization of Reversible and Quantum Circuits.**IEEE 13th International Symposium on April 2010.
[12]. www.google.co.in/
[13]. http://en.wikipedia.org/
[14]. http://www.dwavesys.com/
[15]. http://www.dailygalaxy.com/my_weblog/2012/05/is-the-age-of-silicon-coming-to-an-end-physicist-michio-kaku-says-yes.html
[16]. http://spectrum.ieee.org/tech-talk/computing/hardware/how-dwave-built-quantum-computing-hardware-for-the-next-generation

