Rules for Synthesizing Quantum Boolean Circuits using Minimized Nearest-Neighbor Templates

Amlan Chakrabarti [1] and Susmita Sur-Kolay [2]
[1] University of Calcutta, Kolkata - 700 009, India,
[2] Indian Statistical Institute, Kolkata - 700 108, India
[1] achakra12@yahoo.com, [2] ssk@isical.ac.in

Abstract

Quantum Boolean circuit (QBC) synthesis issues are becoming a key area of research in the domain of quantum computing. While Minterm gate based synthesis and Reed-Muller based canonical decomposition techniques are adopted commonly, nearest neighbor synthesis technique for QBC utilizes the quantum logic gates involving only the adjacent target and control qbits for a given quantum network. Instead of Quantum Boolean circuit synthesis using C^kNOT gate, we have chosen the template-based technique for synthesis of QBC. This work defines new minimization rules using nearest neighbor templates, which results in reduced number of quantum gates and circuit levels. The need of proper relative placements of the quantum gates in order to achieve the minimum gate configuration has also been discussed.

Keywords: *Quantum Gates, Quantum Boolean Circuits, Nearest-Neighbour Synthesis.*

1. Introduction

The model of Quantum Computing stands on the understanding of quantum circuits and their application to solve computational problems. A quantum circuit is employed to process quantum bits (qbit). A qbit may be considered as the equivalent of a binary bit in a classical computer [1]. It can be taken as a particular spin state of an electron, or a certain polarization state of a photon. The spin state of an electron may be up (↑) or (↓) down; or the polarization state of a photon may be vertical (↕) or horizontal (↔). The two quantum mechanical states are represented in standard quantum mechanics [2] by standard ket notation |0> and |1>. The real difference between the classical and quantum states is that while in the former, the states are definite in quantum computing the states are superposed. For example, a quantum state is represented by superposition of two states as $y = a|0> + b|1>$ where a and b are the complex amplitudes representing the probabilities of state $|0>$ and $|1>$ respectively satisfying the condition $|a|^2+|b|^2 =1$. Unlike a classical computer in which a bit has exactly one value from the set {0,1}, a qbit represents both states simultaneously. The maximum number of possible states depends on the number of qbits i.e., n qbits can represent 2^n states. A 2-qbit vector can simultaneously represent the states |00>, |01>, |10>, |11> and the probability of their occurrence depends on the complex amplitude value $y=C_0 \mid 00> +C_1 \mid 01> +C_2 \mid 10> + C_3 \mid 11>$. Hence comes the concept of quantum register [6] of m qbits holding 2^m simultaneous values. This also implies that if we perform an operation on the contents of a register, all possible values are operated on simultaneously, thus leading to quantum parallelism [5]. However, in practice it is quite complex to achieve quantum parallelism, and is dependent on the property of quantum decoherence [3,7].

The organization of this paper is as follows. Preliminary concepts and definitions appear in Section 2. Nearest neighbor template based synthesis of QBC has been briefed in Section 3. Nearest neighbor templates for C^2NOT and CNOT combinations, and placement policy for minimizing the gate cost has been discussed in Section 4. Synthesis results for benchmark circuits using the minimization rules appear in Section 5. Concluding remarks are in Section 6.

2. Preliminaries

2.1. Reversible Logic

A Boolean function is reversible if each of the values in the input set can be mapped with a unique value in the output set. Landauer [8] proved that the usage of traditional irreversible circuits leads to power dissipation. Bennet [4] showed that a circuit consisting of only reversible gates does not dissipate power. Above all, some of the specialized computational applications like digital signal processing, computer graphics, cryptography, reconfigurable computing, etc. demands the preservation of input data.

2.2. Reversible Quantum Gates and Circuits

A reversible logic gate implements a reversible Boolean logic and it necessarily has equal numbers of input and output wires. We now discuss about a few reversible gates.

C^kNOT Gates: A C^kNOT possesses $k+1$ number of input and output wires. It has k control inputs and the $k+1^{th}$ input is inverted at the output if all the k control inputs are at logic high. For $k=0$ it is a NOT gate which maps $x \rightarrow x \oplus 1$ as shown in Figure 1(a), for $k=1$ it is termed as controlled NOT or CNOT which maps $(x, y) \rightarrow (x, y \oplus x)$ as shown in Figure 1(b). The C^2NOT gate, also termed as TOFFOLI gate, it maps (x, y, z) $\rightarrow$ (x, y, z $\oplus$ x.y) as shown in Figure 1(c) .A C^kNOT is represented by k control qbits and a single target qbit, it maps $(x_0, x_1,\ldots\ldots,x_{k-1},y) \rightarrow (x_0,x_1,\ldots\ldots,x_{k-1},x_k \oplus x_0.x_1.x_2\ldots\ldots x_{k-1})$. The control and the target qbits are represented by • and ⊕ respectively. A generalized C^kNOT is shown in Figure 1(d).

Swap Gates: A swap gate is a 2x2 reversible gate. It interchanges the input bit values at the output. Figure 2 illustrates the internal architecture of a swap gate.

A QBC is a quantum system of N qbits specified by $|x_1>|x_2>\ldots\ldots|x_N>$ and a number of reversible quantum gates. The convention for QBC representation is to have the input qbits at the extreme left. These interact with a number of reversible quantum gates as desired, and the final output appears at the extreme right with all the input values restored at the output. The desired function is obtained with the help of a set of ancillary bits which are initialized at the input with |0>. Essentially, such a QBC is reversible and can be synthesized with a set of transformation rules.

2.3. Previous Work

Maslov and Dueck [13] have justified the use of circuit templates in QBC synthesis for minimizing the number of gates in the QBC. Younnes and Miller in their work [9] have introduced the techniques for representation of quantum Boolean circuits using Reed-Muller expansions and have mainly focused on generalized C^kNOT based circuit synthesis. Though C^kNOT gates are acceptable in high level logic design, the technology based implementation of quantum circuits demands the usage of only one, two and three qbit quantum logic gates like *NOT, CNOT, SWAP, C^2NOT*. Hence there is a need of defining efficient synthesis techniques in quantum circuits involving only the smaller qbit sized quantum gates, with small fan-in.

3. Nearest Neighbor Synthesis of QBC

Younnes and Miller [9] in their work have mentioned about the interaction between the adjacent only qbits is a needed technique for practical implementation of QBC and have utilized the SWAP gates to bring the control and target qbits adjacent. SWAP gates play a key role in bringing the control and the target qbits of any quantum gate on adjacent lines in a quantum gate network which is defined as the nearest neighbor configuration. The requirement of nearest neighbor relationship between the control and the target qbits is truly justified due to the limitation of the J-coupling force [14] required to perform multi-qbit logic operations and this works effectively only between the adjacent qbits.

We present below a set of circuit templates for non-adjacent qbit controlled CNOT and C^2NOT in our nearest neighbor based synthesis approach. We introduce the circuit templates for CNOT, and C^2NOT gates utilizing the SWAP gates in order to achieve the nearest neighbor circuit configuration. In Fig. 5, for a C^2NOT we use the notation (*ctrl1*,*ctrl2*,*target*) where the integers *ctrl1*, *ctrl2* and target are respectively the indices of the input qbits of the circuit for the top-control, bottom-control and the target qbit of this C^2NOT gate. The same convention is also followed for CNOT gate which is represented as CNOT(*ctrl*,*target*). According to the convention for index values of input qbit lines mentioned earlier, we assign index 1 to the bottommost control qbit input of the circuit, and the successive index values are assigned as we go upwards to the topmost qbit line. Thus, C^2NOT(4,3,1) represents a C^2NOT gate with its top-control on the 4th input qbit line, its bottom-control on the 3rd line and its target is on the lowest (1st) input qbit line. The exact number of SWAP gates required for nearest neighbor configuration is determined by the differences in the index values of the two control qbits with the target qbit, which can be calculated by the following rules:

Rule 1: *For a C^2NOT gate with indices of its input lines (ctrl1,ctrl2,target) in a QBC, the number of pairs of SWAP gates required is $s_t + s_b$ where s_t is max{(ctrl1 – target-2),0} and s_b is max{(ctrl2 – target - 1),0}.*

Example: In C^2NOT(4,3,1) (Fig. 3(a)) the difference in index value between the top-control and target is 4-1=3 and that between the bottom-control and the target is 3-1=2, hence we require two pairs of SWAP gates, one each to make the top-control and bottom-control as the nearest neighbor of the target qbit. In $C^2NOT(4,2,1)$ (Figure 3(c)) the difference between the top-control and target is 4-1=3 and that between the bottom-control and the target is 2-1=1 , hence we require only one pair of SWAP gate.

Rule 2*: For a CNOT gate with indices of its input lines* (*ctrl*, *target*) *in a QBC, the number of pairs of SWAP gates required is* s_c *where* s_c *is* $max\{(ctrl - target - 1), 0\}$.

Example: In CNOT(4,1) (Figure 3(d)) the difference in index value between the control and the target qbit is 4 -1=3, hence we require 2 pair of SWAP gates, for similar reasons we require a single pair of SWAP gate for CNOT(3,1).The circuit in Figure 4(a) after being synthesized using the nearest neighbor template $C^2NOT(4,3,1)$ (Figure 3(b)) leads to the circuit as shown in Figure 4(b), and we can see an increase in the gate count and circuit level. Hence we need to focus on the minimization of gate count and number of levels in the QBC, which will reduce the quantum circuit cost.

A quantum Boolean circuit may involve generalized C^kNOT gates depending on the number variables involved and hence we have to convert each of the C^kNOT gates to an equivalent C^2NOT based representation. Figure 5 shows a C^2NOT equivalent circuit for a C^4NOT gate involving two ancillary qbits. The number of C^2NOTgate required for a single C^KNOT is 2(k-2)+1 and the number of ancillary qbits required is k-2, where k is the number of control qbits in the C^KNOT gate.

Observation: Any Quantum Boolean Circuit (QBC) can be synthesized using C^2NOT, CNOT, NOT and SWAP gates.

4. Minimized Nearest Neighbor Synthesis of QBC

The nearest neighbor template based synthesis of QBC generally involves a large number of quantum gates due to the conversion of individual C2NOT and CNOT to its equivalent nearest neighbor forms, hence we try to find out whether there is any scope of minimization in the number of gates in the circuit. In this section we propose the minimization rules for the QBC in terms of quantum gates and levels by using the nearest neighbor templates for the different combinations of C^2NOT and CNOT gates. The combinations of the C^2NOT and CNOT gate can be represented with minimized number of SWAP gates for nearest neighbor synthesis if they have the same target qbit and at least one of the control qbits common. We have only taken those combinations of the C^2NOT and CNOT gates, which are not in nearest neighbor configuration.

In Figure 6 we show the two equivalent nearest neighbor templates for each of the C^2NOT and CNOT gate combinations i.e. $C^2NOT(4,3,1)$ + CNOT(4,1), $C^2NOT(4,3,1)$ + CNOT(3,1) and $C^2NOT(4,2,1)$ + CNOT(4,1) , one by combining the nearest neighbor templates for the individual C^2NOT and CNOT gates, and the other one is generated by removing the redundant SWAP gates. We have used the notation of C^2NOT (ctrl1,ctrl2,target) + CNOT(ctrl,target) for representing the combination of C^2NOT and CNOT gates.

The nearest neighbor templates formed after the removal of the redundant SWAP gates reduces the total gate count and level count for the templates and hence they can be termed as the minimized nearest neighbor templates. From Figure 6 we can see that using the minimized nearest neighbor templates we have reduced the total number of gates and the number of levels by a value of 2, reduction in the number of gates and levels are larger for circuits having higher qbits. The reduction in the gate count and level count for the minimized nearest neighbor templates can be easily generalized by the following rules:

Rule 3: *If the top-control qbit of a C^2NOT gate and the control qbit of a CNOT gate are on the same qbit line in a QBC, then the number of pairs of SWAP gates required is one more than that required for the C^2NOT gate only.*

Example: In the combination $C^2NOT(4,3,1)$ + CNOT(4,1) the top-control qbit of the C^2NOT gate and the control qbit of the CNOT gate is on the same qbit line. By Rule1 $C^2NOT(4,3,1)$ requires 2 pairs of SWAP gate and hence considering CNOT(4,1) the combination will require 2+1=3 pairs of SWAP gate in the minimized nearest neighbour template as shown in Figure 6(a). The same is also true for the template shown in Figure 6(c)

Rule 4: *If the bottom-control qbit of a C^2NOT gate and the control qbit of a CNOT gate in a QBC are on the same line then the total SWAP gate requirement is same as that of the C^2NOT gate alone.*

Example: In the combination $C^2NOT(4,3,1)$ + CNOT(3,1) the bottom-control qbit for the C^2NOT gate and the control qbit for the CNOT gate is in the same qbit line. By Rule1 $C^2NOT(4,3,1)$ requires 2 pairs of SWAP gate and hence the combination will require the same number of SWAP gate pairs as shown in Figure 6(b), no extra pair is required for the CNOT(3,1) gate.

Utilization of the nearest neighbor templates as shown in Figure 6 also depends on the proper placement of the C^2NOT and CNOT gates relative to each other such that the initial circuit configuration becomes favorable for the nearest neighbor synthesis.

Rule 5: *The C^2NOT and CNOT gates which work on the same target qbit and have at least one control qbit common, should be adjacent.*

Example: Figure 7(a) and 7(b) the shows the QBC and its equivalent nearest neighbor circuits. Figure 7 (a) uses the individual templates for the $C^2NOT(4,3,1)$, CNOT(4,1) gates placed at circuit level 2 and 4 respectively. In Figure 7(b) the $C^2NOT(4,3,1)$ and CNOT(4,1) gates both having the 4^{th} qbit as a common control bit are being adjacently placed at circuit level 2 and 3 which results to a reduction in the gate count and level count by 4 .

6. Results

We have applied our synthesis techniques for the reversible logic benchmark circuits and obtained the following results as shown in Table 1.

Table 1: Nearest Neighbor Synthesis for Reversible logic Benchmark Circuits

Name	# Gates	Extra Swap Gates	# Swap Gates after Minimization	# Ancillary Qbits
rd32	4	4	4	None
4 mod 5	5	6	6	None
5 mod 5	10	14	14	3
rd 53	16	82	76	2
rd84	28	182	138	None

The results show that though the nearest neighbor technique itself requires quite a large number of *SWAP* gates, a considerable amount of reduction is possible through the new *minimization* rules discussed in Section 4. The ancillary qbits are due to the decomposition of the generalized C^kNOT ($k \geq 3$) gates present in the given circuits, into a number of C^2NOT appropriately.

5. Conclusion

Our work focuses on defining the synthesis techniques for quantum Boolean circuits utilizing the nearest neigbour templates for CNOT and C^2NOT gates. We have presented the generalized rules for the creation of nearest neigbour templates and have also defined the cost-effective synthesis rules and placement policies which leads to the minimized QBC in respect to the number of quantum gates and circuit level. Our future work will be to define a general automation technique utilizing the proposed rules, for generating optimized QBC in terms of quantum gate cost.

6. References

[1] P.W. Shor, "Quantum Computing", *Documenta Mathematica* - Extra Volume ICM, pp. 1-1000, 1998.

[2] R. P. Feynman, "Quantum Mechanical Computers", *Foundations of Physics* , Vol.16, pp.507-531 ,1986.

[3] M. A. Nielsen and Isaac L. Chuang, "Quantum Computation and Quantum Information", *Cambridge University Press* ,2002.

[4] C. Bennett. "Logical reversibility of computation", *I.B.M. J. Res.* .Dev.,17, pp.525-532,1973.

[5] J. Preskil, "Quantum Computing: Pro and Con", *Proc. Royal Society Lond.* A454, pp.469-486 ,1998.

[6] H.K. Lo, S. Popescu and T. Spiller, "Introduction to quantum computation and information," Singapore : *World Scientific Publ.* ,1999.

[7] H. Buhrman, R. Cleve and A. Wigderson. "Quantum vs. classical communication and computation", *In Proceedings of the 30th Annual ACM Symposium on the Theory of Computation*, ACM Press, El Paso, Texas, pp. 63-68 ,1998.

[8] R. Landauer. "Irreversibility and heat generation in the computing process", *I.B.M. J. Res.* Dev, 5, pp.183-191, 1961.

[9] A. Younes and J. Miller, " Representation of Boolean Quantum Circuits as Reed Muller Expressions", *arxiv: quant- ph/0304134*, May 2003.

[10] J.Kim, J-S.Lee, and S.Lee, "Implementation of the refined Deutsch-Jozsa algorithm on a three-bit NMR quantum computer", *Physical Review A*, Volume 62, 022312, 2000.

[11] J.Kim, J-S.Lee, and S.Lee, "Implementing unitary operators in quantum computation," Physical Review A, Volume 62, 032312, 2000.

[12] J.S.Lee, Y.Chung, J.Kim, and S.Lee,"A Pactical Method of Constructing Quantum Combinational Logic Circuits", *arXiv:quant-ph/*9911053, V1, Nov. 1999.

[13] D. Maslov and G. W. Dueck, "Toffoli network synthesis with templates", *Computer-Aided Design of Integrated Circuits and Systems IEEE Transactions*, Volume: 24, Issue: 6, pp.807- 817, June 2005.

[14] I. A. Grigorenko and D.V. Khveshchenko, "Single-Step Implementation of Universal Quantum Gates," *Physical Review Letters*, 95.110501, 2005.

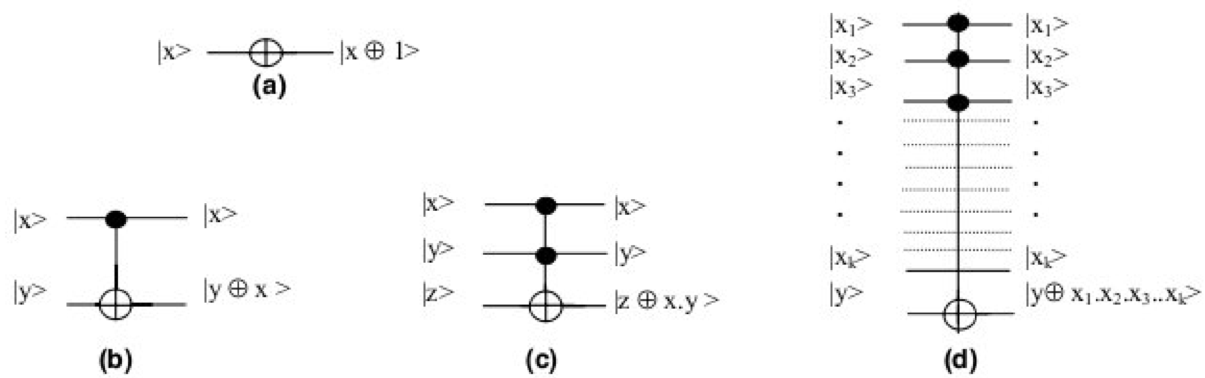


Figure 1: (a) NOT gate, (b) CNOT gate, (c) C^2NOT gate, (d) C^kNOT gate.

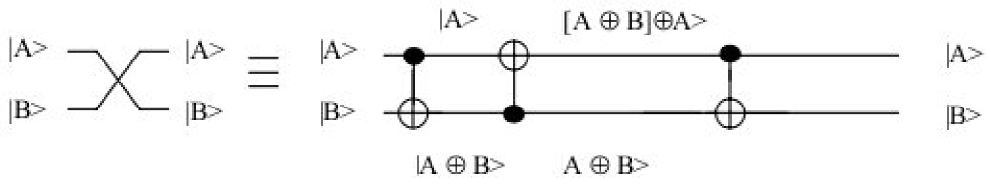


Figure 2: SWAP Gate

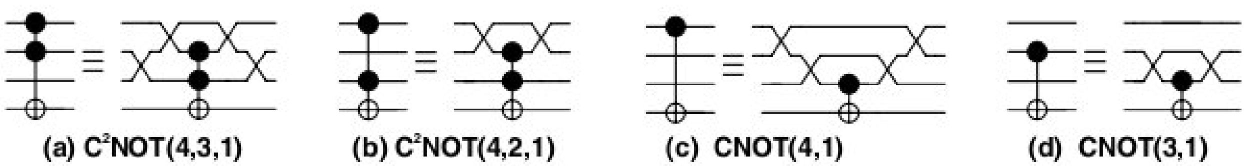


Figure 3: Nearest neighbor templates

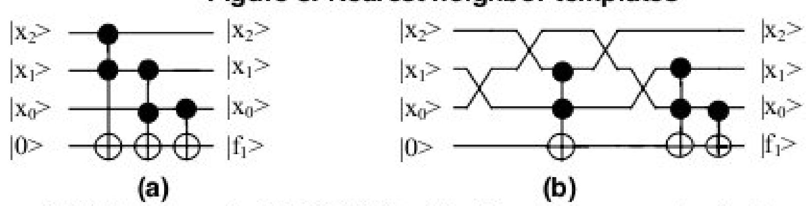


Figure 4: (a) An example QBC (b) Synthesis using nearest neighbor templates

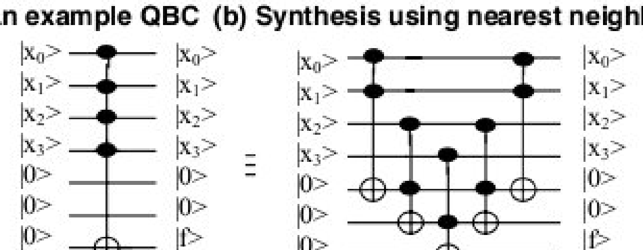
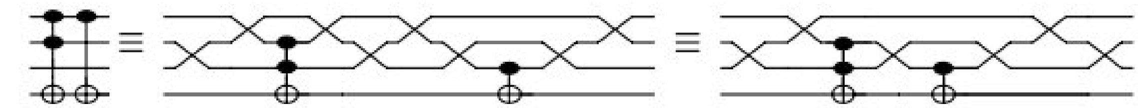
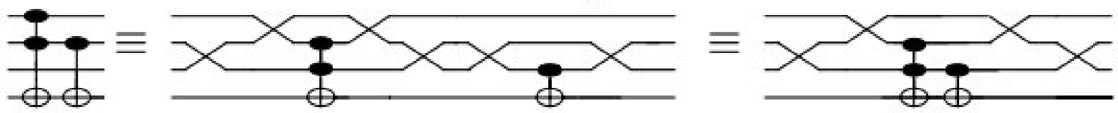


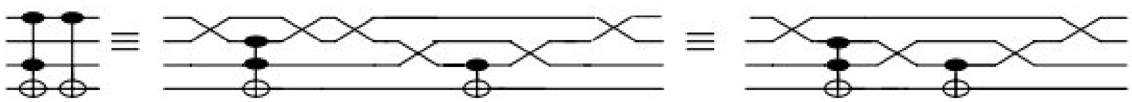
Figure 5: C^2NOT gate equivalent circuit for C^4NOT gate



(a)

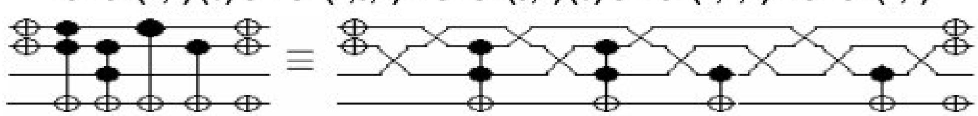


(b)

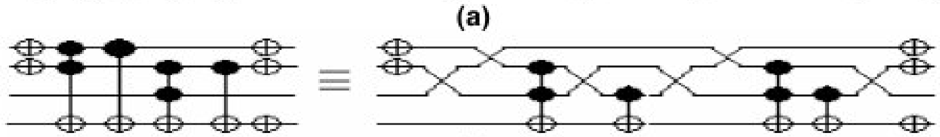


(c)

Figure 6: Nearest neighbor templates for C^2NOT and CNOT combinations (a) $C^2NOT(4,3,1)$ + CNOT(4,1) (b) $C^2NOT(4,3,1)$ + CNOT(3,1) (c) $C^2NOT(4,2,1)$ + CNOT(4,1)



(a)



(b)

Figure 7: (a) Nearest neighbor circuit using individual templates of C^2NOT (4,3,1), CNOT(4,1) and CNOT (3,1) gates (b) Minimized nearest neighbor circuit using the templates of $C^2NOT(4,3,1)$ + CNOT(4,1),CNOT (3,1) and placed according to Rule 5