

An Approach to Simplify Reversible Logic Circuits

Pabitra Roy¹, Subrata Das², Samar Sensarma³

Department of Information Technology, Academy of Technology, Hooghly, India^{1,2}
Department of Computer Science & Engineering, University of Calcutta, India³

Abstract

Energy loss is one of the major problems in traditional irreversible circuits. For every bit of information loss $kT \ln 2$ joules of heat is lost. In order to reduce the energy loss the concept of reversible logic circuits are introduced. Here we have described an algorithm for simplifying the reversible logic circuit and hence reduction of circuit cost and energy. The algorithm considers sub_circuit with respect to their number of lines and contiguous gates. The resulting sub_circuits are re-synthesized with smaller equivalent implementation. The process continues until circuit cost reaches good enough for Application or until a given computation budget has been exhausted. The circuit is constructed by NOT, CNOT and Toffoli gates only. By applying the algorithm and using the equivalent implementation we will get significant reduction of circuit cost and hence energy.

Keywords

Reversible gates, Quantum computing, Synthesis of reversible circuits, qubits.

1. Introduction

Analysis and synthesis of any circuit (digital and analog) considers cost optimization and minimum energy dissipation. We are dealing with only digital circuits and their reversible counterpart. The modus operandi behind is once again minimum energy loss or aesthetically speaking lower bound entropy content. The emerging area in the field of Quantum computing that taught us to think physically about computation. The explorations in this field may one day result in information processing devices with capabilities far beyond today's computing and communication systems. Reversible Computation is a fundamental aspect necessary for Quantum Computation. The overall advantage of making the computation reversible on a Classical Computer is that heat dissipation is reduced and new techniques like Quantum Computing can be approached with this foundational change in the way processing is done. Such combination of Classical and Quantum

Computers would be indispensable, since our perception is classical. According to Moore's Law by the year 2020 if it could be possible, the device geometry would come down to an extent that one atom will represent one bit, the smallest entity of information. At this scale the behavior of the device can only be dictated by the laws of quantum theory.

In this paper our aim is to find out how reversible circuit can be minimized in terms of number of gates and cost.

2. Related Work and Literature Review

The concept of Reversible logic was first introduced in [9]. New types of reversible gate (TR gate) and with that gate implementation of reversible subtractor are discussed in [8]. In [1] the authors proposed the design of a new type of four bit reversible comparator. In [6] the authors explored a number of promising techniques for synthesizing optimal and near optimal circuits that require little or no temporary storage. The design of reversible gate and reversible sequential circuit based on DNA computing were discussed in [2]. Compact storage mechanisms for synthesizing reversible circuits are discussed in [3]. In [4] the authors described the window optimization of reversible and quantum circuits. Good architecture for Full Adder circuits based on Toffoli Gate, Feynman Gate and New Gate were discussed in [5]. Synthesis of multipurpose binary reversible gates that can be used in regular circuits realizing Boolean function has been discussed in [7].

3. Preliminaries

In this section we discuss some introductory concepts of reversible gates.

Definition: A multi-output function $f: B^n \rightarrow B^m$ is a *reversible function* iff its number of inputs is equal to the number of outputs (i.e. $n = m$) and it maps each input pattern to a unique output pattern.

Reversible Logic Gates

NOT Gate: Among all the conventional gates NOT gate is only reversible, since from output we can predict the input; But NOT gate is not functionally complete. Hence the concept of Controlled NOT gate and Controlled NOT gates are introduced. [9]

Controlled NOT Gate (CNOT): The Controlled-NOT gate has two inputs and two outputs. It changes the second bit if the first bit is 1 and leaves the second bit unchanged otherwise. CNOT gate has a similar behavior of XOR gate with some extra information to make it reversible. It works in the following way. We have two wires, on one of which we put a circle (0), representing a control, and in other a cross(x). The "X" denotes a not operation, but this NOT is not a conventional one; it is controlled by the input to the O-wire. Input in the O-wire is called controlled bit and that X- wire is known as target bit.[9].

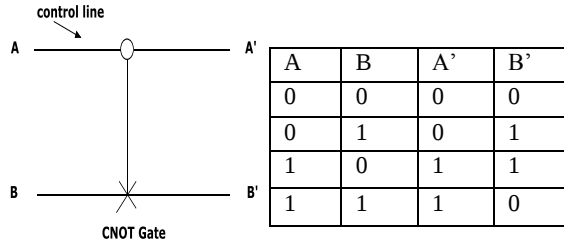


Fig.1: Controlled NOT Gate & its Truth table.

Controlled-Controlled NOT Gate or TOFFOLI Gate (CCNOT):

We cannot do everything with just NOT and CNOT gates. Something more is needed. So the concept of CCNOT gate is introduced. This Gate operates on three qubits as follows: it negates the last bit (i.e. the target bit) of three if first two control bits are both set to 1. Otherwise target bit remains unaltered. This is represented graphically in the Fig.2

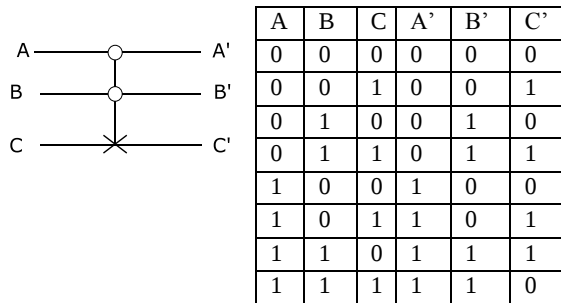


Fig.2: Controlled- Controlled NOT gate & its Truth table

From CCNOT gate we can generate NOT gate, CNOT gate, NAND gate. By itself CCNOT gate form a complete operator set.

4. Proposed Algorithm for Simplifying Reversible Circuits

In this section we discuss the algorithm for simplifying the reversible circuits under the assumptions that the circuit is constructed by NOT, CNOT and Toffoli gates only. Our optimization relies on determining if a selected sub_circuit can be re-implemented at lower cost. Here circuit cost is calculated as the sum of gate costs, where all gates of the same type contribute equally. For reversible circuits, the cost depends on the number of control lines. For example, a Toffoli gate with no or one control line has quantum cost of one, while a Toffoli gate with two control lines has quantum cost of five.

Sub_circuit optimization

In this section sub_circuit optimization of reversible circuits are introduced. Instead of considering the overall circuits as a whole, smaller sub_circuits are extracted. Then these sub_circuits are locally optimized. The resulting sub_circuits are re-synthesized with smaller equivalent implementation. An algorithm is introduced in the next subsection.

Algorithm_1: Sub_circuits generation based on number of lines and contiguous gates:

Input: Any given reversible circuits.

Output: Set of Sub_circuits.

- [1] $G := g_1, \dots, g_n$ is a given circuit.
- [2] $W := \phi$, $p = 1$
- [3] Initialize k such that $2 \leq k \leq L-1$ // L = width of the circuit.
- [4] $w_p := \phi$
- [5] for $i = 1$ to $n-1$ // n = total number of gates in the circuit.
- [6] If (number_of_lines (g_i) $\leq k$)
- [7] $w_p := g_i$, $t = g_i$
- [8] for $j = i+1$ to n
- [9] if (number_of_lines(w_p, g_j) $\leq k$)
- [10] if (contiguous(w_p, g_j, t, G))
- [11] $w_p := w_p \cup g_j$
- [12] else
- [13] $W := W \cup \{w_p\}$
- [14] $p \leftarrow p+1$
- [15] $w_p := g_j$, $t = g_j$
- [16] endif
- [17] endif

```

[18]endfor
[19] W:=W  $\cup$  {wp}
[20] p  $\leftarrow$  p +1
[21] wp:= $\emptyset$ 
[22]endif
[23] endfor

```

bool contiguous (sub_circuit w_p, gate g_j, gate t, circuit G)

```

[1] for each gh=t  $\in$  G to gj of G
[2] if (gh  $\notin$  wp)
[3] if !(can_swap (gh,gj)
[4] return false
[5] endif
[6] endif
[7] endfor
[8] return true
[9] end

```

Fig. 3: Algorithm for Sub_circuits generation

Algorithm Details

The algorithm considers sub-circuit with respect to their number of lines and contiguous gates. Starting with k=2, all sub-circuits with k circuit lines are considered. Afterwards k is incremented until the overall number of circuit lines minus one is reached (i.e. k<L, L=Max.Line or width of the circuit). Given G as a circuit (line 1), sub-circuits are generated and stored in the set W (line 2). For each k the approach starts from first gate of the circuit and end at n-1 gate. w_p represents the current sub_circuit, which is initially empty (line 4). Then the circuit is traversed from left to right. For each gate it tries to make sub-circuit by adding all gates that satisfy number of lines as well as are contiguous. For each gate g_j, it first checks whether the number of lines exceeds or not. If it exceeds the number of lines then it consider the next gate in the circuit. If it does not exceeds the number of lines then it checks whether by adding g_j to the sub_circuit makes it contiguous or not. If it makes contiguous then the sub_circuit is extended by g_j (line 8-11). If it does not make it contiguous then the current sub_circuit is added to W, and a new current sub_circuit is initialized with g_j (line 13-15). To check contiguous the rules R1 and R2 are used [3]. At the end we have a set of sub_circuits in W.

R1: If gates g and h share no wires and are consecutive in the circuit then in the original circuit g does not follow h and h does not follow g. Swapping g and h will not affect the functionality of the circuit.

R2: If gates g and h share some wires, but control wires of one gate are not connected to the target wire of the other gate. Then swapping g and h will not affect the functionality of the circuit.

Algorithm _2: Simplifying Given Reversible circuits:

Assumptions: The circuit is constructed by NOT, CNOT and Toffoli gates only.

Input: Given reversible circuit:

Output: Reduced Circuit with lower cost.

step1: Generate sub_circuits using algorithm_1 for particular value of k.

step2: Remove the duplicate sub_circuit and single gate sub_circuit.

step3: Re-synthesis the sub_circuit with equivalent implementation with lower cost. If equivalent implementation results lower cost, then substitute the sub_circuit with smaller implementation. Two or more sub_circuits can be re-synthesized at the same time if they do not have any common gates.

step4: Repeat step1 to step3 until circuit cost reaches good enough for Application or until a given computation budget has been exhausted.

Fig. 4: Algorithm for Simplification of Reversible logic Circuits

Now We Explain the Sub-circuit Optimization with a Motivating Example

We can find equivalent implementation of a sub_circuit by applying transformation-based approach. It traverses each line of the truth table and adds gates to the circuit until the output value match the input value. Gates are chosen so that they do not alter already considered truth table lines.[4] In this way we try to find the equivalent implementation of the sub_circuit with lower cost. If the equivalent implementation results lower cost than substitute the sub_circuit with the equivalent implementation. As smaller sub_circuits are iteratively considered, the cost may be reduced. Exact synthesis is also applicable as they generate circuits with a minimal number of gates or cost, respectively. As require significant run time and thus are only applicable to small circuits. Since the sub_circuits are restricted to both the number of gates and the number of lines, exact synthesis represents a promising choice for sub_circuit optimization. [4].

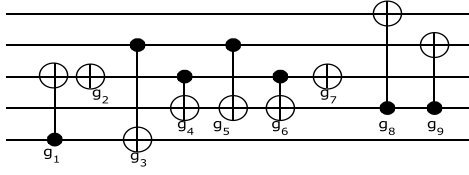


Fig. 5: Circuit with NOT and CNOT gates [4]

For Example: Applying the above algorithm to the circuit of figure5: with $k=3$ we get the following sub_circuits.

$w_1=\{g_1, g_2, g_3\}, w_2=\{g_7, g_8\}, w_3=\{g_2, g_3\}, w_4=\{g_7, g_8\},$
 $w_5=\{g_3, g_5\},$
 $w_6=\{g_9\}, w_7=\{g_4, g_5, g_6, g_7, g_9\}, w_8=\{g_5, g_6, g_7, g_9\},$
 $w_9=\{g_6, g_7, g_8\}, w_{10}=\{g_7, g_8\}, w_{11}=\{g_8, g_9\}$

After removing the duplicate sub_circuit we have

$W = \{w_1, w_2, w_3, w_5, w_7, w_8, w_9, w_{11}\}$

If we take the sub_circuit w_7 for re-synthesize then the smaller equivalent implementation of w_7 is shown in the Fig. 7.

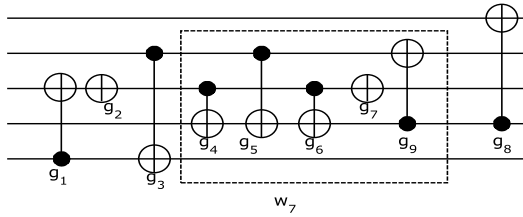


Fig. 6: Sub_circuit $w_7=\{g_4, g_5, g_6, g_7, g_9\}$



Fig. 7: Equivalent Sub_circuit of w_7

The sub_circuit is substituted by equivalent implementation and the reduced circuit is shown in the Fig. 8.

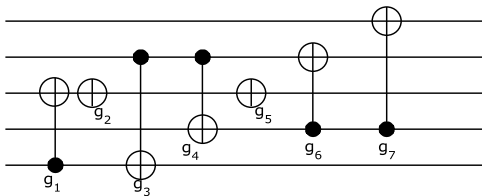


Fig. 8: Reduced Circuit of Fig. 5.

Now again we apply the algorithm on the reduced circuit to generate the sub_circuit for $k=3$. We get the following sub_circuits.

$w_1=\{g_1, g_2, g_3, g_5\}, w_2=\{g_2, g_3, g_5\}, w_3=\{g_3, g_4, g_6\},$
 $w_4=\{g_4, g_5, g_6\}, w_5=\{g_5, g_6\}, w_6=\{g_6, g_7\}$
 $W=\{w_1, w_2, w_3, w_4, w_5, w_6\}$

Now we take w_1 for re-synthesize with smaller equivalent implementation and it is shown in the Fig. 10

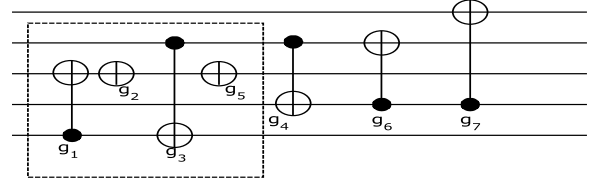


Fig. 9: Sub_circuit $w_1=\{g_1, g_2, g_3, g_5\}$

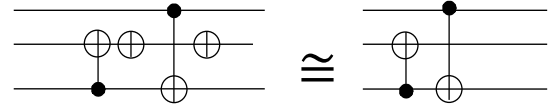


Fig. 10: Equivalent Sub_circuit of w_1

Now the reduced circuit is shown in the Fig. 11

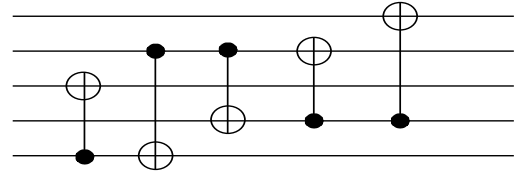


Fig. 11: Reduced circuit of Fig. 9.

By applying the above algorithm and using the equivalent implementation we will get significant reduction of cost.

Cost Table: Toffoli gate costs:

Table-I: Cost of Toffoli Gates

Size	Name	Cost
1	NOT	1
2	CNOT	1
3	Toffoli(t3)	5
4	t4	13
5	t5	29
6	t6	61

Results (The Experimental Circuits are given in the Appendix.

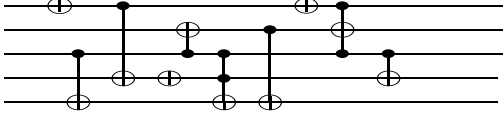
Table-II: Experimental results

Circuits	#Lines	#Gates	Cost	#Reduced Gates	Reduced Cost
Circuit_1	5	10	18	6	14
Circuit_2	5	9	9	5	5
Circuit_3	6	18	158	16	108
Circuit_4	7	28	117	20	109
Circuit_5	8	20	80	15	43
Circuit_6	10	21	100	14	54

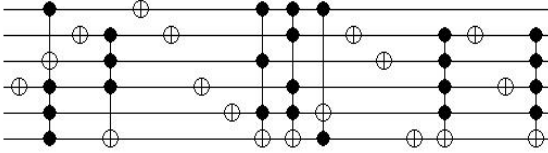
5. Conclusion

Our work focuses on synthesis techniques for reversible logic circuits using number of lines and contiguous gates. The proposed synthesis technique operates on circuits consisting only NOT, CNOT and CCNOT gates. Our proposed work gives the significant reduction of both cost and number of gates. If we are able to design a classical reversible computer the Quantum computation and Quantum algorithms can very well be simulated to produce error free results. However it was Einstein who said "If you can think about it, it will be possible now or in future".

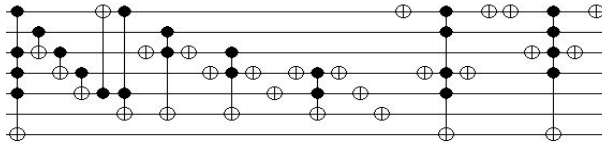
Appendix:



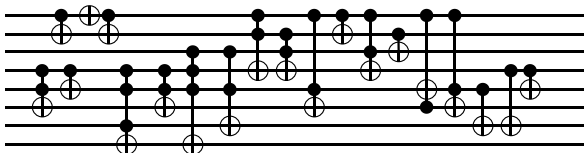
Circuit_1



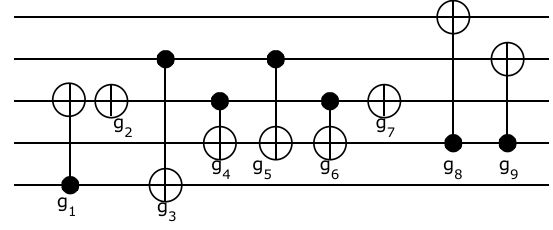
Circuit_2



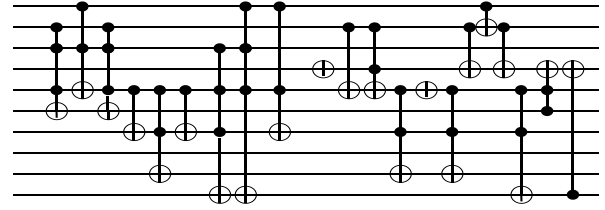
Circuit_3



Circuit_4



Circuit_5



Circuit_6

References

- [1] Lihui Ni, Zhijin Guan, Xiaoyu Dai, Wenjuan Li, "Using New Designed NLG gate for the realization of Four-bit reversible Numerical comparator", International Conference Educational and Network Technology, pp 254-258, 2010.
- [2] Tao Song, Shudong Wang, Xun Wang, "The Design of Reversible Gate and Reversible Sequential Circuit Based on DNA", 3rd international Conference on Intelligent System and Knowledge Engineering, pp 114-118, 2008.
- [3] Aditya K. Prasad, Vivek V Sende, Ketan N Patel, Igor L. Markov, John P. Hayes, "Data Structure and Algorithms for Simplifying Reversible Circuits", ACM Journal on Emerging Technologies in Computing Systems (JETC), Volume 2 Issue 4, PP 277 – 293, October 2006.
- [4] Mathias Soeken, Robert Wille, Gerhard W. Dueck, Rolf Drechsler, "Window Optimization of Reversible and Quantum Circuits", Workshop on Design and Diagnostics of Electronic Circuits and Systems - DDECS, pp- 341 - 345, 2010.
- [5] Hafiz Md. Hasan Babu, Md. Rafiqul Islam, Ahsan Raja Chowdhury, Sayed Mostahed Ali Chowdhury, "Reversible logic synthesis for minimization of full-adder circuit", Euromicro Symposium on Digital System Design, pp 50-54, 2003.
- [6] Vivek V Sende, Aditya K. Prasad, Igor L. Markov, John P. Hayes, "Reversible Logic Circuit Synthesis", International Conference on Computer Aided Design, IEEE, 2002.
- [7] Pwiel Kerntopf, "Synthesis of Multipurpose Reversible Logic Gates", Euromicro Symposium on Digital system Design (DSD02), IEEE, 2002.

- [8] Himanshu Thapliyal, Nagarajan Ranganathan, "Design of Efficient reversible Binary Subtractors Based on New Reversible Gates", IEEE Computer Society Annual Symposium on VLSI, pp 229-234. 2009.
- [9] Feynman, Richard Phillips, J. G. Hey, and Robin W. Allen. Feynman lectures on computation. Addison-Wesley Longman Publishing Co., Inc., 1998.



Pabitra Roy received B.Sc in Physics (Hons.) (2006), B.Tech (2009) & M.Tech (2011) in Computer Science & Engineering from University of Calcutta, India. He is presently working as Assistant professor in the dept. of Information Technology, Academy Of Technology, Aedconagar, Hooghly,

712121.



Subrata Das received B.Sc in Physics (Hons.) (2005), B.Tech (2008) in Information Technology & M.Tech (2010) in Computer Science & Engineering from University of Calcutta, India. He is presently working as Assistant professor in the dept. of Information Technology, Academy Of Technology, Aedconagar, Hooghly, 712121.



Samar Sensarma is founder faculty member of the department of Computer Science and Engineering, University of Calcutta, India. He is the first Ph.D. (Tech.) in Computer Science from University of Calcutta. He has more than ninety papers published in peer reviewed journals and conferences.