Neural, Parallel, and Scientific Computations 20 (2012) 11-16

COMMENTS ON "QUANTUM NETWORKS FOR ELEMENTARY ARITHMETIC OPERATIONS"

PIJUSH KUMAR KOLEY[1] AND SRIMANTA PAL[2]

[1]Department of Instrumentation and Electronics Engineering,
Jadavpur University, Salt Lake Campus, Kolkata, India
[2]Electronics and Communication Sciences Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, India

ABSTRACT. This paper makes a few comments on the quantum subtractor circuit proposed by Vedral et al. (1996) in "Quantum networks for elementary arithmetic operations". First we demonstrate some flaws with this circuit. Then we propose a possible corrected circuit for this plain subtractor and a new combined circuit that can be used both as an adder and a subtractor.

AMS (MOS) Subject Classification. 68M15, 94C12

1. Introduction

A quantum computer can be viewed as a quantum network composed of quantum logic gates. These gates are different from their classical counterpart, which are irreversible in nature. Quantum gates perform unitary operation on one, two or more than two qubits (Barenco et al. 1995, DiVincenzo 1995, Lloyd 1995). Each qubit is an elementary unit of information and is represented by a *ket* vector $|0\rangle$ or $|1\rangle$. These qubits are analogous to classical bits 0 or 1. Any quantum operation is unitary, and hence is reversible (Bennett 1973, Bennett 1989), so we need reversible gates. Quantum addition and subtraction operations are performed using these reversible gates. In this paper we have discussed the plain quantum subtractor circuit proposed by Vedral et al. (1996) and show some flaws in this circuit. We then propose its corrected quantum algorithm for the plain quantum subtractor. Finally, we proposed a combined quantum circuit, which can act both as an adder and a subtractor using a control qubit.

2. Quantum Adder and Subtractor by Vedral et al.

In this section we discuss the plain quantum adder and subtractor proposed by Vedral et al. (1996). Their proposed plain quantum adder (Vedral et al. 1996) is correctly working and their quantum subtractor uses this quantum adder circuit. Here we study this quantum subtractor and highlight some flows in it.

Received July 27, 2011

Consider two real numbers x_1 and x_2, and store them in registers R_1 and R_2 respectively. Now our objective is to add x_1 and x_2, i.e., add the contents of R_1 and R_2 and store the output in another register R_3 (say) whose size is greater than that of R_1 and R_2 to prevent overflow. In quantum computing notation we can write the above statement as in Eqn (2.1):

$$|x_1\rangle|x_2\rangle|0\rangle \rightarrow |x_1\rangle|x_2\rangle|x_1 + x_2\rangle \tag{2.1}$$

Instead of storing the result in another register R_3 we can store it in R_2 register and to do so we need a little more complex operation as proposed by Vedral et al. (1996). The length of R_2 register is greater than that of R_1 register, i.e., if the length of R_1 register is n qubits, the R_2 register will be $n + 1$ qubits. In this case the addition operation can be written as in Eqn (2.2):

$$|x_1\rangle|x_2\rangle \rightarrow |x_1\rangle|x_1 + x_2\rangle \tag{2.2}$$

To achieve this, we need a temporary register of size $n-1$ qubits, whose every qubit

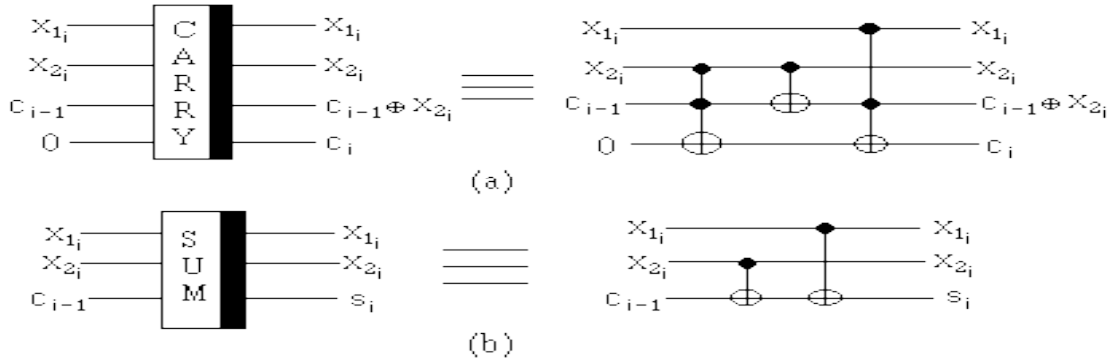

FIGURE 1. Quantum plain adder circuit by Vedral el al. (1996): (a) carry computation circuit, (b) sum computation circuit.

is initially set to $|0\rangle$. This temporary register is used to store carries for intermediate operations and the last carry will be stored in the most significant qubit of the R_2 register. Let x_{1_i} and x_{2_i} be the ith qubit of register R_1 and R_2 respectively. We now describe in Algorithm 1 for plain addition as proposed by Vedral et al. (1996).

Algorithm 1: Quantum adder

Input: x_1, x_2

Output: $x_1 + x_2$

Step 1. The carry c_i is computed using the following relation in (2.3)

$$c_i \leftarrow (x_{1_i}\ AND\ x_{2_i})\ OR\ (x_{1_i}\ AND\ c_{i-1})\ OR\ (x_{2_i}\ AND\ c_{i-1}) \tag{2.3}$$

where x_{1_i}, x_{2_i} represent the ith qubit of R_1 and R_2 register respectively and c_{i-1} represents the $(i-1)$th qubit of the temporary register.

Step 2. The sum is computed using the following relation in (2.4)

$$x_{2_i} \leftarrow x_{1_i}\ XOR\ x_{2_i}\ XOR\ c_{i-1}. \tag{2.4}$$

Step 3. [**Termination**] Stop

Figure 1(a) shows the circuit for computation of the carry. The inputs are x_{1_i}, x_{2_i}, c_{i-1}, and 0, while the outputs are x_{1_i}, x_{2_i}, $c_{i-1} \oplus x_{2_i}$, and c_i. In this circuit first the AND operation is performed between x_{2_i} and c_{i-1}; next the XOR operation is performed between x_{2_i} and c_{i-1}, and then a Taffoli gate (Taffoli 1980) is used. The two control inputs of the Taffoli gate are x_{1_i} and the output of the XOR operation. Its target qubit is the output of the AND operation. Figure 1(b) shows the quantum circuit for sum computation. In this circuit the three inputs are x_{1_i}, x_{2_i} and c_{i-1}. Here, first a XOR operation is performed between x_{2_i} and c_{i-1} and then again a XOR operation is performed between x_{1_i} and output of previous XOR operation.

Vedral et al. (1996) write "If we reverse the action of adder network (i.e., if we apply each gate of the network in the reversed order) with the input (x_1, x_2), the output will produce $(x_1, x_1 - x_2)\ldots$". So, according to their proposition if we apply the gate operations in reverse order then we should get the subtraction operation. Following their suggestion, the circuits for the borrow and difference calculation should be as shown in Figures 2(a) and 2(b) respectively.

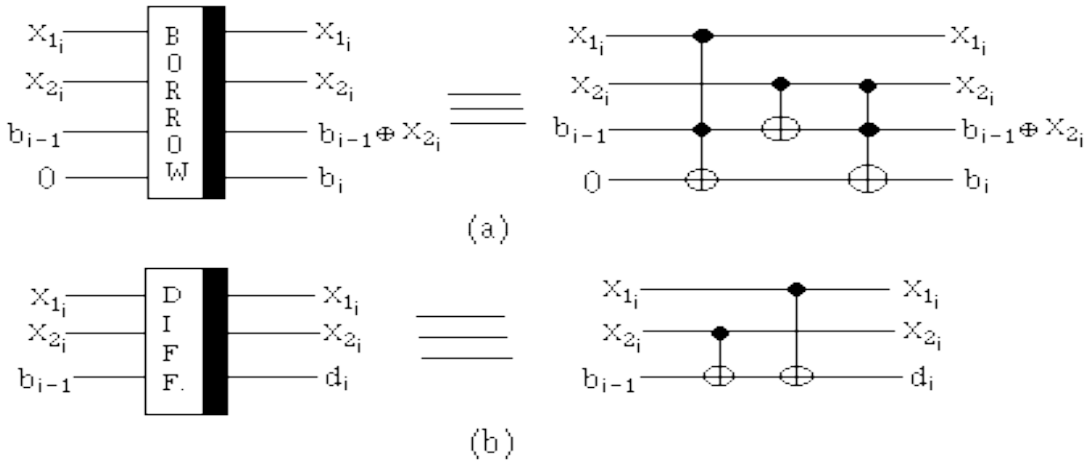

FIGURE 2. Quantum plain subtractor circuit by Vedral et al. (1996) (a) Borrow computation circuit (b) Difference (DIFF.) computation circuit

Now we illustrate the subtractor using examples as in Table 1.

In Table 1 we find that for rows 2 and 4 *no borrow* is produced, but we know that for these two cases borrow should be produced. Again for each of rows 6 and 7 a borrow is produced, but we know that for these two cases no borrow should be produced. So the plain subtractor is not correctly working.

Table 1: Truth table for subtractor using Vedral et al. (1996) method

x_{1_i}	x_{2_i}	b_{i-1}	d_i	b_i
0	0	0	0	0
0	**0**	**1**	**1**	**0**
0	1	0	1	1
0	**1**	**1**	**0**	**0**
1	0	0	1	0
1	**0**	**1**	**0**	**1**
1	**1**	**0**	**0**	**1**
1	1	1	1	1

3. **Modified Quantum and Subtractor**

In this section we discuss two issues (i) the possible correction for Vedral's (1996) proposed quantum network for quantum subtractor and (ii) our proposed combined architecture for quantum adder and subtractor.

3.1. **Correction for Vedral's et al. (1996) Network.** Vedral's et al. (1996) quantum network is already discussed in Section 2. Now we propose a modified architecture to perform the quantum subtraction operation. The computational procedure is discussed in Algorithm 2.

Algorithm 2: Modified quantum subtractor.

Input: x_1, x_2

Output: $x_2 - x_1$

Step 1. The borrow b_i is computed using the following relation (3.1):

$$b_i \leftarrow (\overline{x}_{1_i}\ AND\ x_{2_i})\ OR\ (\overline{x}_{1_i}\ AND\ b_{i-1})\ OR\ (x_{2_i}\ AND\ b_{i-1}) \tag{3.1}$$

where x_{1_i} and x_{2_i} represent the ith qubit of registers R_1 and R_2 respectively, $\overline{x}_{1_i}$ is the complement of x_{1_i} and b_{i-1} represents $(i-1)$th qubit of the temporary register.

Step 2. The difference is computed using the relation (3.2):

$$d_i \leftarrow x_{1_i}\ XOR\ x_{2_i}\ XOR\ b_{i-1} \tag{3.2}$$

If we store the difference in register R_2, then the operation can be written as follows:

$$x_{2_i} \leftarrow x_{1_i}\ XOR\ x_{2_i}\ XOR\ b_{i-1} \tag{3.3}$$

Step 3. [**Termination**] Stop.

Figure 3(a) shows the quantum circuit for difference computation. The three inputs x_{1_i}, x_{2_i}, and b_{i-1}, are XOR-ed and the quantum network produces the output x_{1_i}, x_{2_i}, and d_i. The borrow computation is showed in Figure 3(b). This circuit consists of four inputs, among them one is zero and the other three are x_{1_i}, x_{2_i}, and b_{i-1}. It produces the borrow b_i along with other necessary outputs. Note that, to fix the

problem of the algorithm proposed by Vedral et al. (1996) we connected a quantum NOT gate to x_{1_i} line as shown in Figure 3(b).

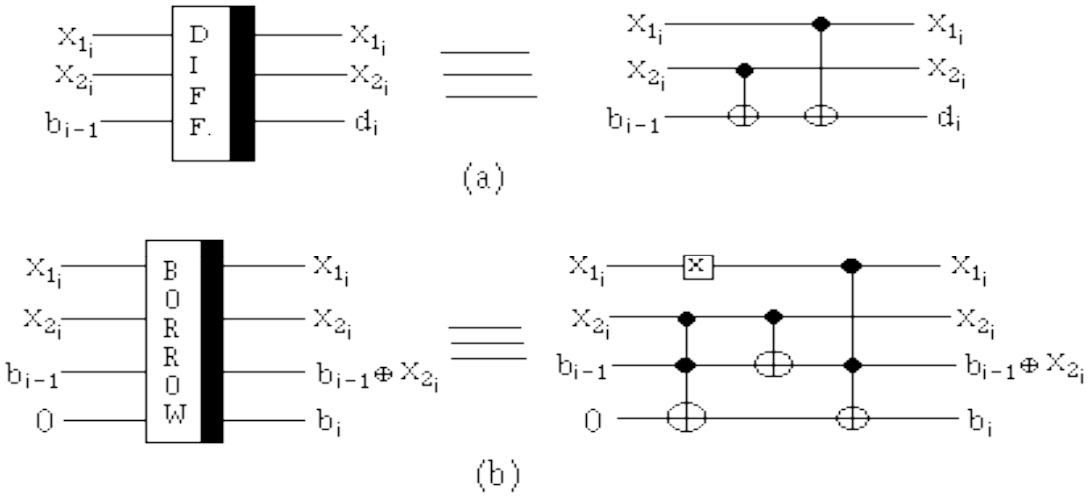

FIGURE 3. Proposed quantum plain subtractor circuit (a) Difference (DIFF.) computing circuit, (b) Borrow computing circuit.

In this case we compute the borrow and the difference with expressions used in classical digital subtraction and the truth table for this quantum circuit is shown in Table 2. Here the outputs are exactly the same as we get in classical digital plain subtractor.

Table 2: Truth table for practical subtractor method

x_{1_i}	x_{2_i}	b_{i-1}	d_i	b_i
0	0	0	0	0
0	**0**	**1**	**1**	**1**
0	1	0	1	1
0	**1**	**1**	**0**	**1**
1	0	0	1	0
1	**0**	**1**	**0**	**0**
1	**1**	**0**	**0**	**0**
1	1	1	1	1

3.2. **Combined Architecture.** Now we propose a combined quantum circuit for adder and subtractor as shown in Figure 4. The idea is to use an extra control qubit. The difference and sum computation follow the same algorithm as described above but only difference is that the borrow is replaced by the carry. So there will be no change in the quantum circuits for the two operations. Sum or difference computation quantum circuit is shown in Figure 4(a). The carry (or borrow) computation quantum circuit is shown in Figure 4(b). In this quantum circuit, the control qubit c is 0 when

it will compute the carry and if the control qubit c is 1 then the borrow will be computed.

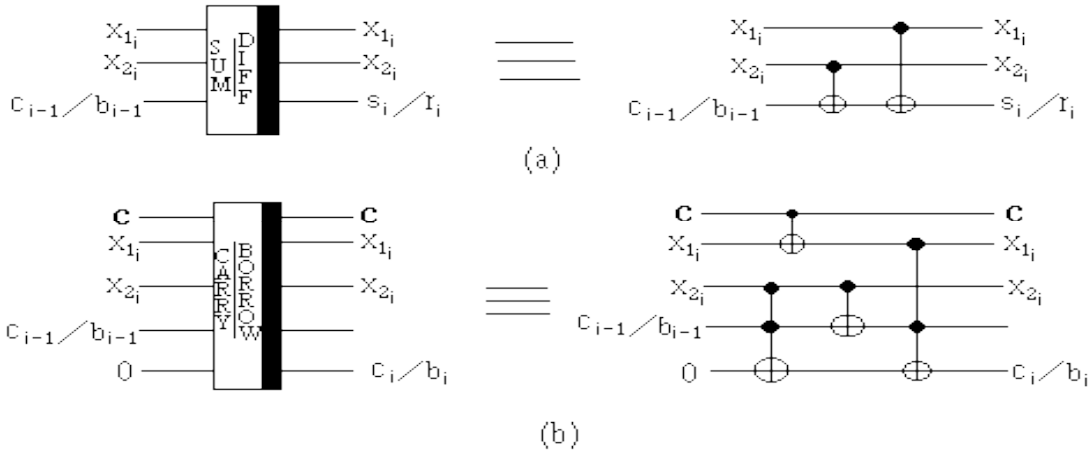

FIGURE 4. Quantum plain adder or subtractor circuit (a) sum or difference computation (b) carry or borrow computation.

4. **Conclusion**

In this paper we first pointed out a flaw in the quantum circuit proposed by Vedral et al. (1996) for the computation of subtraction. We suggested a possible modification of the architecture for the correction of the quantum subtractor. Finally, we proposed a composite architecture for the operations on quantum addition and subtraction.

REFERENCES

[1] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin and H. Weinfurther, Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467, 1995.

[2] C.H. Bennett, Logical reversibility of computation, *IBM J. Res. Develop.*, 17:525–532, 1973.

[3] C.H. Bennett, Time/space trade-offs for reversible computation, *SIAM J. Comput.*, 18:766–776, 1989.

[4] D.P. DiVincenzo, Two-bit gates are universal for quantum computations, *Phys. Rev. A*, 51:1015–1022, 1995.

[5] S. Lloyd, Almost any quantum logic gate is universal. *Phys. Rev. Lett.*, 75:346–349, 1995.

[6] T. Taffoli, Reversible computing, an Automata, Language and Programming, *Seventh Colloquium, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Vol. 84, pp. 632–644, 1980 (J.W. de Bakker and J. van Leeuwen eds.).

[7] V. Vedral, A. Barenco and A. Ekert, A quantum networks for elementary arithmatic operations, *Phys. Rev.* A, 54:147–153, 1996.