International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 1, January-2014, pp: (443-448), Impact Factor: 1.252, Available online at: www.erpublications.com

Quantum Computer Arithmetic Logic Unit

Amit Kumar Dutta
JIS College of Engineering, Kalyani, WB, India

Abstract: The feasibility of Quantum Computer with Super Computing capability is addressed here. We describe a Arithmetic Logic Unit for Quantum Computer and show how the algebra is modified if the information is kept in phase for Radix-4 operation. We formulate Quaternary adder, subtractor, multiplier and divider circuits using basic quantum gates.

Keywords: Quantum Computer, Quaternary number system, adder, subtractor, multiplier and divider.

1. INTRODUCTION

Quantum Computer has shown a great potential for solving certain types of calculation intensive problems such as factoring large number by Shor's algorithm. It is very fast and huge memory can be easily implemented with fast access time. The difficulties are in the physical implementation of qubit and implementation of Datapath [1][5]. Research is continuing in the noise related decoherence for which error correcting codes are used.

The unit of memory for binary quantum computation is the qubit, a system existing in a linear superposition of two basic states |0> and |1>. The standard Dirac notation describing the state of qubit is given by q0=α|0> + β|1>. This means that q0 is in a superposition of the states 0 and 1 with complex coefficients describing their probabilities.

Modern computers are built using logic gates. Similarly Quantum computation also uses Quantum logic gates. All of the quantum logic gates are reversible. Here we consider some of the gates. Any computation like addition and subtraction can be done using the basic gates like Hadamard Gate, Pauli X- Gate, Y- Gate and Z gate [2]. We can represent the rotational gates by Phase Gate and T Gates.

Pauli's X-Gate $\left(\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right)$; Pauli's Y-Gate $\left(\sigma_y = \begin{pmatrix} 0 & -j \\ j & 0 \end{pmatrix}\right)$; Pauli's Z-Gate $\left(\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\right)$

Hadamard Gate $\left(H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\right)$; Phase Gate $\left(PH(\delta) = e^{j\delta}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right)$; Controlled Unitary matrix:

$$\left(\underset{\sim}{U} = \begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix}\right)$$

Controlled-NOT Gate (output= A XOR B, A is the input and B is the controlled input) and Toffoli Gate (output= A.B + C, where A & B are inputs and C is the controlled input).

Multi valued logic circuit is essential for General Purpose Computer or Quantum Computer but not obvious to realize [1][3][5][6][7][8]. In the last three decades research was conducted in multi valued logic circuit. This is also conducted in Quantum computing. We find that the quantum state can take imaginary values and the basic gates such as the Phase Gate modify the phase angle only. So if the information bit is kept in the phase angle, we can easily implement the radix-4 or radix-8 operation by multi valued logic circuits.

The Section I introduces the subject and the Section II formulates the Number Systems. The Section III designs the bit serial adder circuit in logical level and circuit level using Quantum Gates. We extend this idea to subtractor in the Section IV and design the multiplier which is not an array multiplier, but a bit serial array one in Section V. The Section VI discusses an algorithm for division. In the next Section we conclude the paper.

2. NUMBER SYSTEM

In the previous Section, we discussed about the basic quantum gates, X,Y,Z,T,S and Hadamard Gates etc. We define the number system by phase rotation in anticlockwise direction starting point at 1 as zero and get the radix 4 and radix 8 arithmetic operations like addition and subtraction.

We know that the S gate (Phase gate) gives a rotation of a phase angle of pi/2 and we can rotate the angle by pi/2 repeatedly to get the other numbers of radix 4 number system. So 1 will be zero, j will be one, -1 will be two and –j will be three.

For radix 8 number system, 1 will be zero, exp(jπ/4) will be one, exp(jπ/2) will be two, exp(j3π/4) will be three, exp(jπ) will be four, exp(j5π/4) will be five, exp(j6π/4) will be six and exp(j7π/4) will be seven. Now Figure 1 explains that we can add any two of the above mentioned numbers and get the addition done by S,X,Y,Z and T Gates for radix 4 number system.

We also can think of negative number as radix 4's complement number. The first digit if it is 1 it will be a negative number, like 1 321. It's 4's complement number is 0 013 (0012+0001).

3. HALF ADDER

We know that the radix 4 number system is given by {1,j,-1,-j} and the sum is given by multiplication so that the phases gets added as given by Table -I.

Now in Quantum Gate CNOT we give input A: a|0> +b|1> and B: c|0> + d|1> the output states will be entangled and ac|00>+ad|01>+bc|11>+bd|10>. In CNOT state 10 gets exchanged to 11 and vice versa. Now if we choose b=0, d=0 that is input a|0> and c|0> then output will be ac|00> and in CNOT it remains unchanged. So the ac is the sum which is multiplication means addition in phases. A and c can be {1,j,-1,-j}.

The carry is difficult to get as it is (j,-j)+(-j,j)+(-1,-1)+(-1,-j)+(-j,-1)+(-j,-j). The first solution is given after converting 1==00,j==01,-1==11,-j==10. The sequence is gpoklj as explained in the Figure 2. The second solution is not included here. The input can be {1,j,-1,-j} The sequence is hplkon, where the alphabets are the states: 0000 a,0001 b, 0010 c, 0011 d, 0100 e, 0101 f, 0110 g, 0111 h, 1000 I, 1001 j, 1010 k, 1011 l, 1100 m, 1101 n, 1110 o and 1111 p states.

We understand how a single bit adder works. We can form a bit serial adder like given in [8] and form a adder circuit which works in either parallel or in connected way as shown in [8] using bit serial clock. Here the shift registers are implemented by qubit number storage.

As we use the single bit Quantum adder in bit serial adder structure, we use serial clock which is faster than the parallel clock by N times. [9] shows the simplest bit serial adder, where the carry is fed back to itself through a qubit storage. If we put 16 single bit adders, in bit serial addition we get N times more addition than the equivalent parallel adder. [9] explains a bit serial adder structure which can be converted to Quantum adder structure where the all the adders are connected in series, carry is propagated through qubits to the next adder. Here the data is fed one after other to the adder serially and addition takes 16 clock cycles for the first addition and next 15 additions are done in 15 serial clock cycle. Thus the addition is done between two qubits of storage. Here the serial clock is found to be the largest time to propagate from a qubit to the next qubit in the adder structure.

4. HALF SUBTRACTOR

A and B have signal constellation {1,j,-1,-j} in radix 4 number system and we have to get the subtraction A-B. The truth table is given in Table-II. We can implement it like this:

The difference is found by a table as shown in Figure-3 and Borrow=(1,j)+(1,-1)+(j,-1)+(1,-j)+(j,-j)+(-1,-j).
The Borrow logic can be generated like CARRY_out in adder circuit.

5. MULTIPLIER

The Multiplier-Accumulator (MAC) unit is the main arithmetic processing unit of the Arithmetic Logic Unit (ALU). The multiplier unit is a series connected 16x16 single bit adder which allows 16 multiplications to take place simultaneously. Here N is equal to 16. It accepts up to 2*N input operands and outputs N of 32-bit result of the multiplication stored in the accumulators [9]. So as it radix 4 number system, it is a 32 bit multiplier all together. The difference in the Quantum multiplier from normal multiplier is that all the single bit adders are Quantum adder circuit. Because it is a radix 4 operation, the ANDing operation [9] is substituted by look up table and addition. The look up table can be found by mod-4 operation. A 4bit x 4 bit multiplier is implemented as shown in Figure 4. Signal is bifurcated using CNOT gate with one input ac|0> and the controlled input as 0|0>. Both the output are ac|00> [2].

6. DIVISION

Here we formulate an algorithm using addition, subtraction and multiplication to the division. The algorithm is stated as follow:
We subtract Bx(n) from A and integrate it and equate it to x(n+1).
In analog domain it is always stable and we use Laplace to solve it.
$\int(A - BX(t))dt = X(t)$
In Laplace domain, $\frac{A}{s} - BX(s) = sX(s)$ and solving this equation we get, $(s) = A/B(\frac{1}{s} - \frac{1}{B+S})$. This in time domain is $X(t)=A/B(1-\exp^{(-Bt)})$ so it converges to A/B as time goes to infinity. A/B is the wanted division.
In Discrete domain the equation is $\sum(A - BX(n))T = X(n+N)$ where T is the step size and kept BT~=1. Now it converges very quickly and assume it takes ten steps. We can implement it such that it takes 65 serial clock period where A and B are 16 bits wide. We can implement it by Quantum adder, subtractor and multiplier in radix 4 number system.

7. RESOLVING ENANGLEMENT

We found that by entanglement we can get the sum which is ac|00> or ac|0>. Now we have to measure the value of ac|0> and it is not obvious. We first compare ac|0> with 1|1> by sending both through unitary matrix. The output will be αac|0>+γ.1|1> and β ac|0>+δ.1|1>. Then send the first signal through H gate. The output will be H|1>=α ac|0>-γ .1|1>. We can measure it and if it is small number ac==1. Again we can pass it(α ac|0>+δ 1|1>) through unitary matrix and -1|1> in the input, the output will be ac|0>. So the data is not lost. This is possible if α =β=γ=δ=1. The circuit will be as shown in Figure 5.

8. CONCLUSION

Addition, subtraction and multiplication are the main operations of Arithmetic Logic Unit. We propose an ALU of Quantum Computer which is similar in operation to the Classical Computer. We can improve the speed of addition, subtraction and multiplication than the conventional computer. As the information is kept in phase of electron's spin and it will be least disturbed by noise. The division algorithm is very fast but hardware intensive and not accurate.

REFERENCES

[1]. S. B. Mandal, A. Chakrabarti, S. Sur-Kolay, "A synthesis method for Quaternary Quantum logic circuits," Progress in VLSI Design and Test Lecture Notes in Computer Science Volume 7373, 2012.
[2]. R. Koc," Quantum Gates," University of Gaziantep, Class Notes.
[3]. A. Muthukrishnan, C.R. Stroud Jr., "Multivalued logic gates for Quantum Computation," Physical Review A, Volume 62, 2000.
[4]. Eisuke Abe, "Quantum Circuits," Notes, School of Quantum Computing, Keio University.
[5]. N. Isailovic, "An investigation into realities of a Quantum Datapath," Ph.D. Disseration, UC Berkeley, 2010.
[6]. D.M. Miller, G.W. Dueck, D. Maslov, "A synthesis method for MVL reversible logic," ISMVL, 2004.
[7]. V. Patel K. S., K. S. Gurumurthy, "Arithmetic operation in MultiValued logic," VLSICS, Vol. 1, No. 1, March 2010.
[8]. D.M. Miller, R. Wille, Z. Sasanian, "Elementary Quantum Gate realization for multiple control Toffoli Gates," ISMVL, 2011.
[9]. A. K . Dutta, "Vector arithmetic logic unit," IJERSTE, Vol. 2 Issue 12, December -2013.

TABLE-I

A	B	SUM	CARRY
1	1	1	1
1	J	J	1
1	-1	-1	1
1	-j	-j	1
J	1	J	1
J	J	-1	1
J	-1	-j	1
J	-j	1	J
-1	1	-1	1
-1	J	-j	1
-1	-1	1	J
-1	-j	J	J
-j	1	-j	1
-J	J	1	J
-j	-1	j	J
-j	-j	-1	j

TABLE-II

A	B	DIFF	BORROW
1	1	1	1
1	J	J	j
1	-1	-1	J
1	-j	-j	J
J	1	-j	1
J	J	1	1
J	-1	J	J
J	-j	-1	J
-1	1	-1	1
-1	J	-j	1
-1	-1	1	1
-1	-i	J	J
-j	1	J	1
-J	j	-1	1
-j	-1	-j	1
-j	-j	1	1

TABLE-III

A	B	MULT	CARRY
1	1	1	1
1	J	1	1
1	-1	1	1
1	-j	1	1
J	1	1	1
J	J	J	1
J	-1	-1	1
J	-j	-J	1
-1	1	1	1
-1	J	-1	1
-1	-1	1	J
-1	-j	-1	J
-j	1	1	1
-J	J	-J	1
-j	-1	-1	J
-j	-j	J	-1

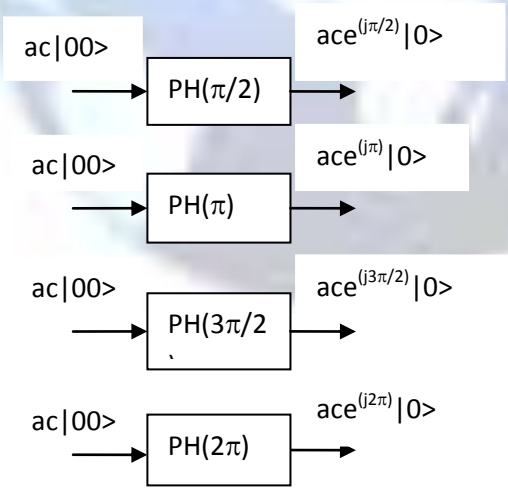

Figure 1. Radix 4 number system in phase is permissible using Phase gate.

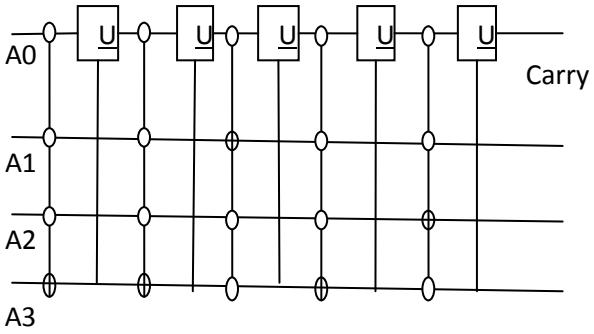

Figure 2. Carry Logic generator. A0,A1,A2 and A3 are 0 or 1. U is unitary transform. Others are Toffoli gate. The sequence is gpoklj.

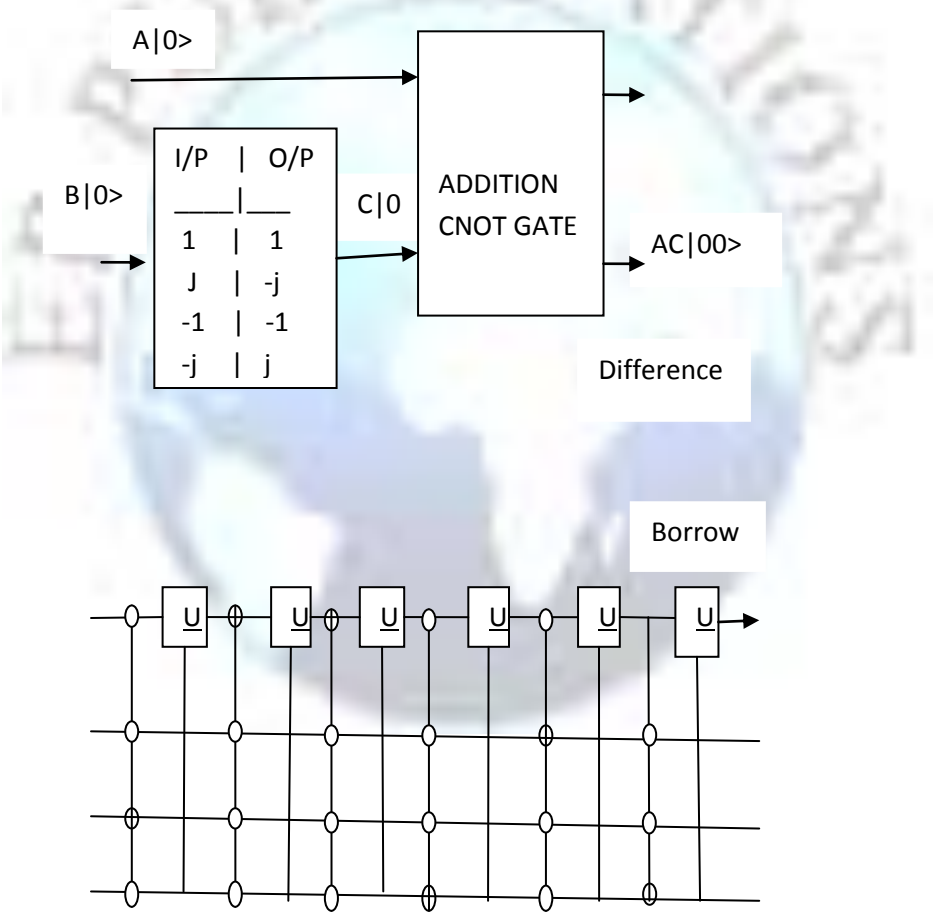

Figure 3. The Difference and the Borrow_out generation of Half Subtractor. The sequence for Borrow is bdldcgh.

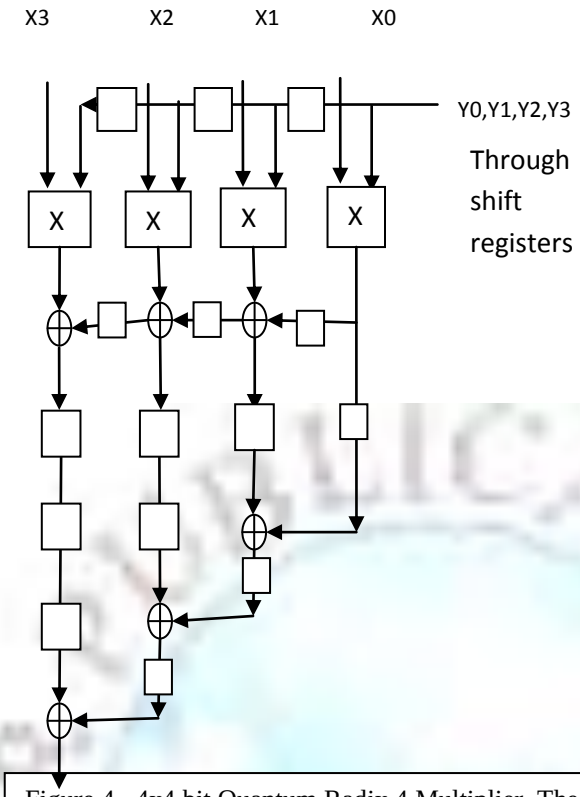

Figure 4. 4x4 bit Quantum Radix 4 Multiplier. The shift registers are implemented by Phase Gate (2π) and bit multiplication (X) is done by table. It does not have any MUX. So it can do multiplication for any number system with its adder.

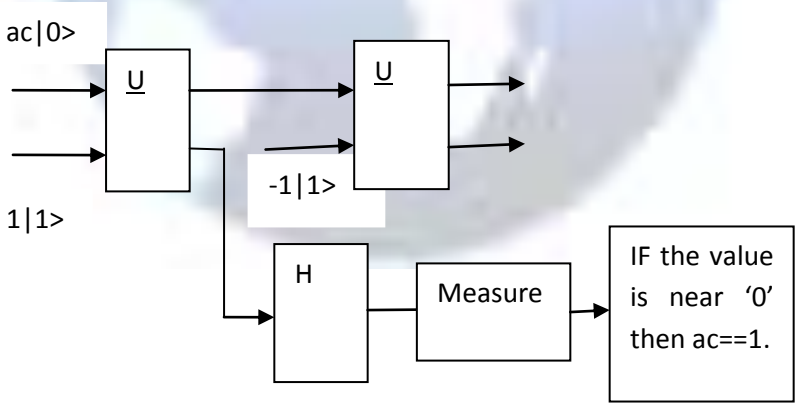

Figure 5. Measurement of entangled values. $\alpha,\beta,\gamma,\delta$ are all 1 in U.