**Project Description**

Now-a-days, the electronic systems are an integral part of a human’s life. As technology develops, the complexity of the system also increases. This gives rise to increase in power consumption, which is a great issue faced by the world today.

Energy dissipation due to information loss in high technology circuits and systems constructed using irreversible hardware was demonstrated by R. Landauer in the year 1960. According to Lindauer’s principle, the loss of one bit of information lost will dissipate kTln2 joules of energy where, k is the Boltzmann’s constant, T is the absolute temperature. In 1973, Bennett showed that to avoid kTln2 joules of energy dissipation in a circuit it must be built from reversible circuits. Reversible circuits (gates) have the same number of inputs and outputs and there is a one to one mapping between vectors of inputs and outputs. Thus, the vector of input states can be always uniquely reconstructed from the vector of output states.

Arithmetic Logic Unit (ALU)

ALU is essentially the heart of a CPU. Different kinds of computers have different ALU’s but all the ALU’s contain arithmetic unit and logic unit, which are the basic structures. This allows the computer to add, subtract, and to perform basic logical operations such as AND, OR etc.

Reversible Logic

A function is reversible if each input vector produces a unique output vector. Reversible logic is of growing importance to many future computer technologies. It is not possible to realize quantum computing without implementation of reversible logic. The main purposes of designing reversible logic are to decrease quantum cost, depth of the circuits and the number of garbage outputs. The reversible circuits form the basic building block of quantum computers as all quantum operations are reversible. Reversible logic is gaining importance in areas of CMOS design because of its low power dissipation. The traditional gates like AND, OR, XOR are all irreversible gates. Consider the case of traditional AND gate. It consists of two inputs and one output. As a result, one bit is lost each time a computation is carried out. According to the truth table shown in Fig.1, there are three inputs (1, 0), (0, 1) and (0, 0) that corresponds to an output zero. Hence it is not possible to determine a unique input that resulted in the output zero. In order to make a gate reversible additional input and output lines are added so that a one to one mapping exists between the input and output. This prevents the loss of information that is main cause of power dissipation in irreversible circuits. The input that is added to an m x n function to make it reversible is known as constant input (CI). All the outputs of a reversible circuit need not be used in the circuit. Those outputs that are not used in the circuit is called as garbage output (GO). The number of garbage output for a particular reversible gate is not fixed. The two main constraints of reversible logic circuit are

* Fan out not allowed.
* Feedbacks or loops not allowed.

Basic Definitions related to reversible logic

**Reversible logic gate:**

Reversible Gates are circuits in which number of outputs is equal to the number of inputs and there is a one to one correspondence between the vector of inputs and outputs. It not only helps us to determine the outputs from the inputs but also helps us to uniquely recover the inputs from the outputs.

**Constant inputs:**

This refers to the number of inputs that are to be maintaining constant at either 0 or 1 in order to synthesize the given logical function.

**Garbage outputs:**

Garbage is the number of outputs added to make an n-input k-output function reversible. We use the words constant inputs to denote the present value inputs that were added to an (n:k) function to make it reversible. The following simple formula shows the relation between the number of garbage outputs and constant inputs.

Input + constant input = output + garbage.

**Quantum cost:**

Quantum cost refers to the cost of the circuit in terms of the cost of a primitive gate. It is calculated knowing the number of primitive reversible logic gates (1\*1 or 2\*2) required to realize the circuit. The quantum cost of a circuit is the minimum number of 2\*2 unitary gates to represent the circuit keeping the output unchanged. The quantum cost of a 1\*1 gate is 1 and that of any 2\*2 gate is the same, which is 1.

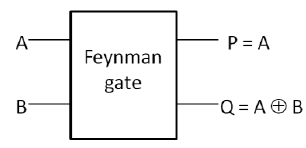
**List of reversible logic gates:**

**NOT Gate**

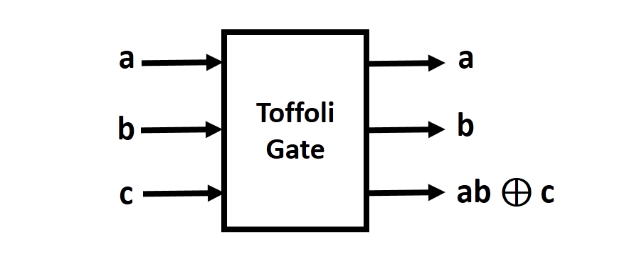
The simplest Reversible gate is NOT gate and is a 1\*1 gate. The Reversible 1\*1 gate is NOT Gate with zero Quantum Cost is as shown in the Figure. If input is A then output is P=.

P

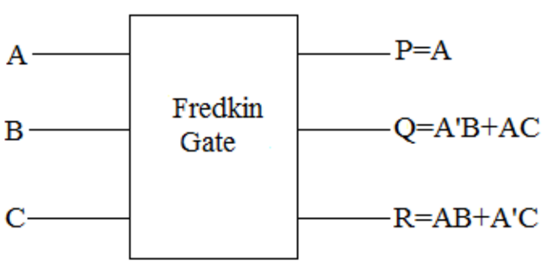
**FEYNMAN Gate**

The Feynman gate which is a 2\*2 gate and is also called as Controlled NOT and it is widely used for fan out purposes. The inputs (A, B) and outputs P=A, Q= A XOR B. It has Quantum cost one. It is the only 2x2 reversible gates available and is commonly used for fan out purposes.

**TOFFOLI Gate**

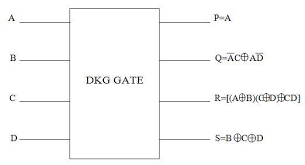
Toffoli gate is a 3\*3 gate 1with inputs (A, B, C) and outputs P=A, Q=B, R=AB XOR C. It has quantum cost of 5 and garbage outputs of 2. It is also called as CCNOT gate. It was invented by Tommaso Toffoli, is a universal reversible logic gate, which means that any reversible circuit can be constructed from Toffoli gates.

**FREDKIN Gate**

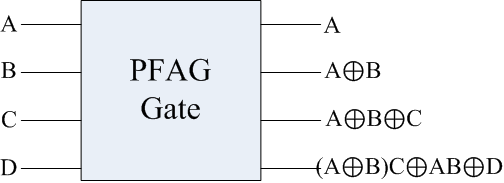
Fredkin gate which is a 3\*3 gate with inputs (A, B, C) and outputs P=A, Q=A'B+AC, R=AB+A'C. It has Quantum cost 5.

**DKG Gate**

It is 4\* 4 reversible gates that can work singly as a reversible Full adder and a reversible Full subtractor is shown in Figure. It can be verified that input pattern corresponding to a particular output pattern can be uniquely determined. If input A=0, the proposed gate works as a reversible Full adder, and if input A=1, then it works as a reversible Full subtractor. Its Quantum cost is 6.



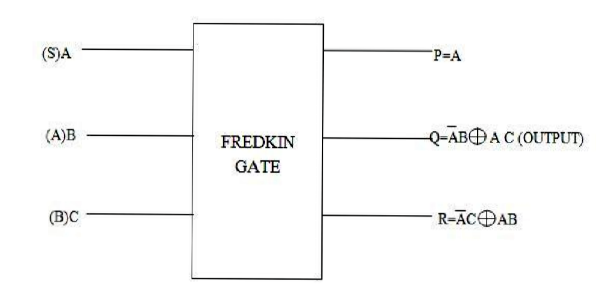
**PFAG Gate**

Peres Full Adder Gate (PFAG) shown in figure. The gate is achieved by cascading two 3\*3 Peres gate. The quantum realization cost of this gate is 8 since it includes two 3\*3 Peres gates. The gate can work singly as a reversible full adder circuit when its fourth input is set to zero (D=0). This gate requires only one clock cycle and produces no extra garbage outputs.

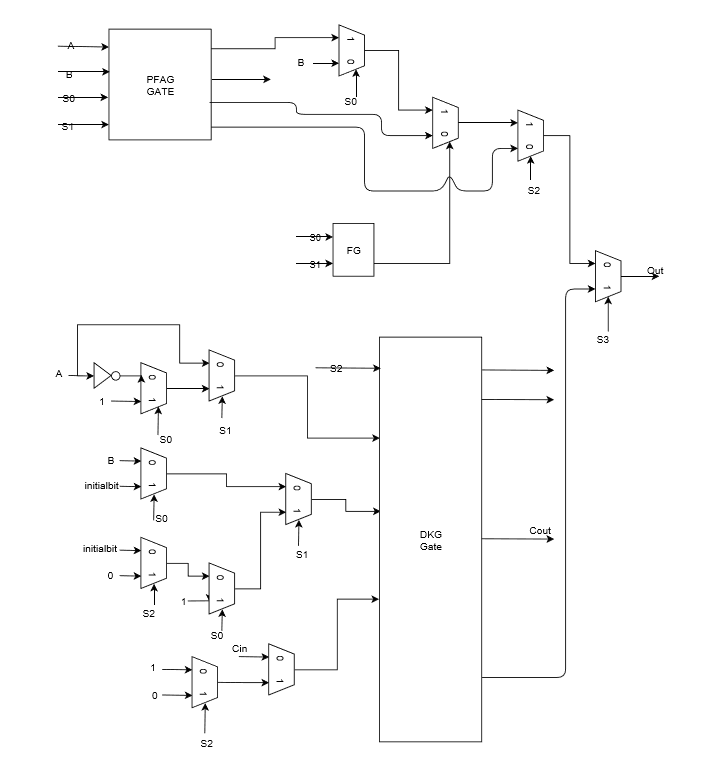
**ALU Design**

ALU has 2 parts, 1st which has PFAG Gate as base of the circuit and is selected when select line s3 is zero. The operations performed here are buffer, AND, OR, NAND, NOR, EX-OR, and EX-NOR. 2nd part has DKG Gate as base of the circuit and is selected when select line s3 is one. The operations performed here are add, increment, 2’s complement, set, subtract, decrement, not, and clear. The operations selected depending on various select lines are shown in the table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operations | S3 | S2 | S1 | S0 |
| AND | 0 | 0 | 0 | 0 |
| OR | 0 | 0 | 0 | 1 |
| NAND | 0 | 0 | 1 | 0 |
| NOR | 0 | 0 | 1 | 1 |
| XOR | 0 | 1 | 0 | 0 |
| Buffer A | 0 | 1 | 0 | 1 |
| Buffer B | 0 | 1 | 1 | 0 |
| XNOR | 0 | 1 | 1 | 1 |
| Addition | 1 | 0 | 0 | 0 |
| Increment | 1 | 0 | 0 | 1 |
| 2’s Complement | 1 | 0 | 1 | 0 |
| SET | 1 | 0 | 1 | 1 |
| Subtraction | 1 | 1 | 0 | 0 |
| Decrement | 1 | 1 | 0 | 1 |
| NOT | 1 | 1 | 1 | 0 |
| CLEAR | 1 | 1 | 1 | 1 |

Multiplexer is designed using Fredkin gate when we make A as select line and (B & C) as input. B or C is selected depending on A is 0 or 1 respectively. The block diagram is shown in figure.

**BLOCK DIAGRAM OF 1-BIT ALU**



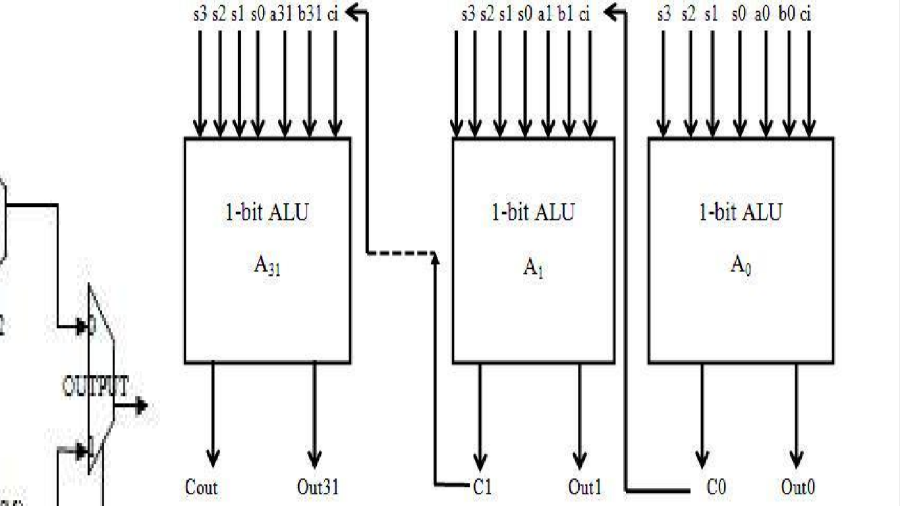
Block Diagram Explanation

* The 2:1 MUX are made using Fredkin gate (FRG).
* The Toffoli gate is used as a AND gate to produce S0.S1 product.
* The Feynman gate is used at two places to produce XOR of S0 and S1.
* The DKG gate is used as an adder/subtractor where sum is S and carry is R.
* Using the PFAG gate at the top the logic operations

AND, OR, NAND, NOR, XOR, XNOR

are performed with C(S1) and D(S0) as select lines with A and B as inputs.

* The BUFFER-A and BUFFER-B operation or performed using the MUXs near PFAG gate.
* The addition operation is performed by sending inputs A, B to B, C and Cin to D of DFK gate and setting A to 1’b0.
* Increment operation is performed by sending input A to B, 1’b1 to C, 0 to D of DKG in addition mode gate for the 1st bit ALU and changing C to 1’b0 for the other ALUs.
* 2’s complement is performed by sending NOT (A) to B, 1’b1 to C, 0 to D of DKG gate for the 1st bit ALU and changing C to 1’b0 for the other ALUs.
* Set operation is performed by sending 1’b1 to B, C, D of DKG gate.
* The subtraction operation is performed by sending input A, B to B, C and Bin to D pf DKG gate and setting a to 1’b1.
* Decrement operation is performed by sending input A to B, 1’b1 to C, 0 to D of DKG gate in subtraction mode for the 1st bit ALU and changing C to 1’b0 for the other ALUs.
* NOT operation is performed by sending NOT(A) to B, and 1’b0 to C, D of DKG gate.
* Clear operation is performed by sending 1’b1 to B, C and 1’b0 to D of DKG gate in addition mode.

Each 1-bit ALU is cascaded as shown in the figure to form a 32-bit ALU. The initalbit value is given as 1 for first bit and for the rest of the bits it is given as zero.

**Refrences**

* C. H. Bennett, “Logical reversibility of computation”, IBM J. Research and Development, 17, pp. 525 – 532, November1973.
* Feynman, R., “Quantum mechanical computers” Optics, News, and pp.: 11-20 11, 1985.
* Fredkin, E. and T. Toffolli, “Conservative logic”, Intl. J. Theoretical Phy., 21, pp.: 219-253, 1982.
* Lihui Ni, Zhijin Guan, and Wenying Zhu, “A General Method of Constructing the Reversible Full-Adder”, Third International Symposium on Intelligent Information Technology and Security Informatics, pp.109-113, 2010 5
* R. Landauer, “Irreversibility and heat generation in the computing process”, IBM J. Research and Development, 5(3), pp.: 183-191, 1961.
* Toffoli T, “Reversible computing”, Technical Memo MIT/LCS/TM-151, MIT Laboratory for Computer Science (February), 1980.