Reversible Logic ALU for 16 Bit Operations

S. Karthikeyan
Ph.D. Research Scholar,
SNS College of Technology,
Coimbatore, India.
skarthigp@gmail.com

Abstract: Reversible logic is one of the developing technologies having bright applications in quantum computing. In this paper the design of a 16bit reversible Arithmetic Logic Unit (ALU) with 15 operations is represented by making use of Double Peres gate, Fredkin gate, Toffolli gate, DKG gate and NOT gate. A modern VLSI architecture for ALU using reversible logic gates is proposed. ALU is one of the most essential components of CPU that can be part of a programmable reversible computing device such as a quantum computer. A first single bit is reversible ALU and second single bit is ALU are designed. After that 16single bit ALU's are cascaded combined in carry out of ALU acting upon LSB operation as an input to carry in of ALU acting upon the next LSB operation. Design is executed and verified by Verilog in modelsim Altera 6.6d.

Keywords: Reversible logic, Reversible ALU, MODEL SIM, Moore's law.

INTRODUCTION

It is well known that Moore's law will stop the function soon and something will happen in microelectronics in near future. By using faster and more complex digital systems being built, power consumption of CMOS circuits has become a major concern. Landauer proved that power loss is an integral feature of irreversible circuits that have information loss irrespective of the technology the circuit is implemented in. Also, Bennett showed that in order to keep a circuit from dissipating any power, it had to be composed of reversible gates. Reversible circuits (gates) have the same number of inputs and outputs and there is one-to-one mapping between vectors of inputs and outputs. The vector of input states can be always singularly reconstructed from the vector of output states. Because low power circuits cannot be built without the concepts of reversible logic, various technologies and circuits for reversible logic are recently studied. The Arithmetic Logic Unit (ALU) is the heart of a CPU. It allows the computer to add, subtract, and to perform basic logical operations such as AND, OR etc.

Since each computer needs to able to these simple functions, they are always included in a CPU. An ALU is a combinational logic circuit that have one or more inputs and only one output. ALU output is based on inputs applied at that instant as a function of time, and not on beyond conditions. The ALU is important consists of two inputs one input for selecting the desired operation and another one output for the result. The difficulty in ALU may vary from processor to processor and for reversible ALU one binary arithmetic and three logical operations are designed in this paper. In present work 16bit ALU with 15 operations are designed.

Related Work

Reversible Arithmetic logic unit with 4-operation AND, OR, X-OR and ADD. Design of a 4-bit 2's Complement Reversible Circuit Design of Control Unit for Low Power ALU with a Barrel Shifter Using Reversible Logic Design of 32 Bit Reversible ALU with 7-operations and Arithmetic & Logic Unit (ALU), Design using Reversible Control Unit with 9-operations are related and recent work in the reversible logic circuits.

REVERSIBLE LOGIC GATES

Reversible logic gates are used in Design circuit in this paper are follows NOT gate, Feynman Gate, Toffoli gate Fredkin Gate, Double Peres gate and DKG gate.

NOT Gate is the easiest Reversible gate and it is 1*1 gate. Not gate is shown in the Figure 1 and its quantum cost is Zero.

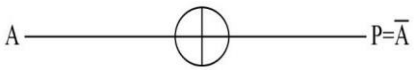

Fig.1. NOT Gate

FEYNMAN GATE Controlled NOT (CNOT) gate is an example for a 2*2 gate. The Reversible 2*2 gate with Quantum Cost of one. It is shown in the Figure 2

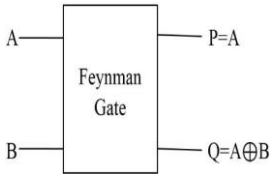

Fig.2. Feynman Gate

TOFFOLI GATE 2 CNOT gates is 3*3 Reversible gate with three inputs and three outputs. Its quantum cost is 5 and is shown in Figure 3

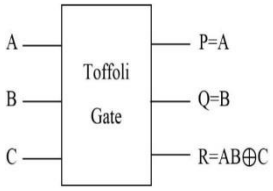

Fig.3. Toffolli Gate

FREDKIN is 3*3 gate maps inputs (A, B, C) to outputs (P=A, Q=A'B+AC, R=AB+A'C) having Quantum cost of 5 and It is shown in the Figure 4

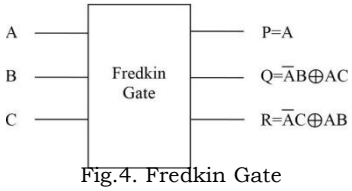

Fig.4. Fredkin Gate

DOUBLE PERES GATE is 4*4 gates with quantum cost of 6. It is shown in the Figure-5

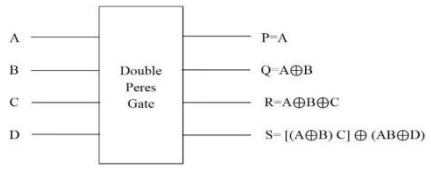

Fig.5. Double Peres Gate

DKG GATE is 4* 4 reversible DKG gate that can work singly as a reversible Full adder and a reversible Full
subtractor is shown in Figure 6. It can be verified that input pattern corresponding to a particular output pattern can be uniquely determined. If input A=0 it works as a reversible Full adder and if input A=1 it works as a reversible Full subtractor.

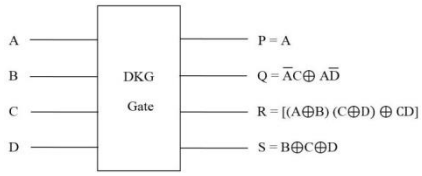

Fig.6. DKG Gate

METHODOLOGY

1-BIT-ALU

The ALU that is proposed is 15-operations. There ALU has 2 parts. First part has Double Peres Gate and base of the circuit is selected when select line s3 is zero. The operations performed are buffer, AND, OR, NAND, NOR, EX-OR, and EX-NOR. Second part is DKG Gate is the base of the circuit and is selected from select line s3 is one. The operations done by adder, increment, 2's complement, set, subtractor, decrement, not, and clear. The operations selected depending on various select lines are shown in below Table-1.

S3	S2	S1	S0	Operations
0	0	0	0	AND
0	0	0	1	NAND
0	0	1	0	OR
0	0	1	1	NOR
0	1	0	0	BUFFER A
0	1	0	1	EX-OR
0	1	1	0	BUFFER B
0	1	1	1	EX-NOR
1	0	0	0	ADDITION
1	0	0	1	INCREMENT
1	0	1	0	2's COMPLEMENT
1	0	1	1	SET
1	1	0	0	SUBSTRACTION
1	1	0	1	DECREMENT
1	1	1	0	NOT
1	1	1	1	CLEAR

Table-1.Operations

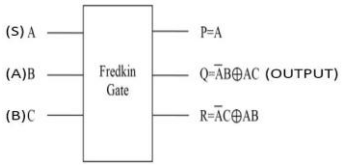

Fig.7. 2:1 MUX

The 2:1 Multiplexer is developed using Fredkin gate and 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 7.

Design of first 1-bit ALU with 15 operations is shown in the figure 8.

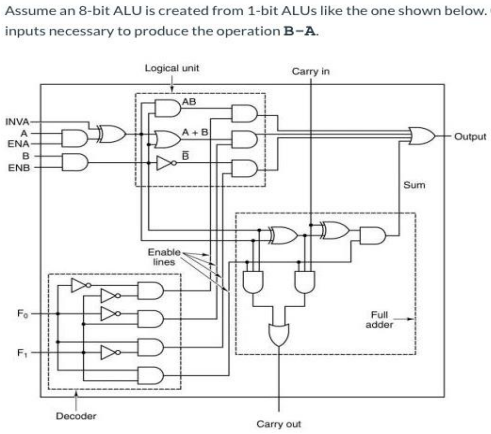

Fig.8. 1st 1-bit ALU

For 16-bit ALU we cascade 16 1-BIT-ALU as shown in the figure-10. We need that only first bit has to be added with one for increment and 2's complement or subtracted for decrement. There is a small change in SET and CLEAR operation. A separate 1-BIT-ALU is designed as shown in figure 9 and cascaded from second bit onwards till 16-bit. It is from (A1-A15) in the figure 10. A0 in the figure 10 is circuit shown figure 8.

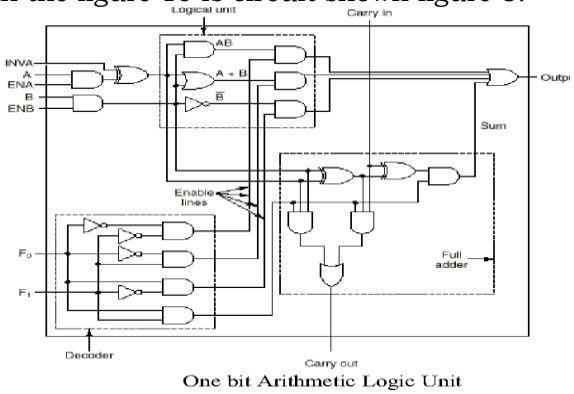

Fig.9. 2nd 1-BIT-ALU

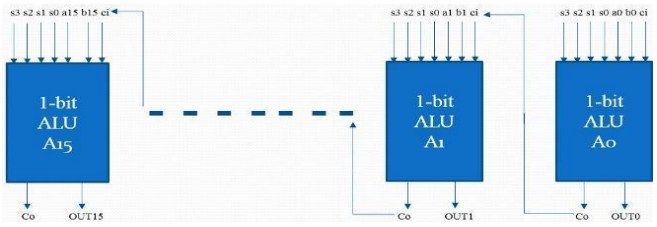

Fig.10.16-BIT-AL

SIMULATION RESULT

The simulation result for 16-bit-ALU design in Verilog are verified by modelsim Altera 6.6d is shown in figures 11, 12, 13 and 14.

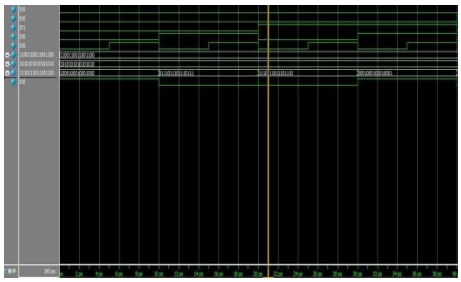

Fig.11. Result for 1st 4-operations (AND, NAND, OR and NOR)

The Figure-11 shows the result of first 4 operations (AND, NAND, OR and NOR). These are operations are selected depending on select lines s3-s0 as shown in table-2. The Graph is shown for each 4-operations because it cannot contain in the same screen. In Proposed work for ALU is 16-bit and has 15-operational features whereas as in the existing work the simulation result and design for 8 to 10 features at the maximum.

Table-2 shows the first 4 operations selected depending on select lines. The operation is executed by two inputs A & B and output is given out.

S3	S2	S1	S0	Operation	Input(A)	Input(B)	OUTPUT
0	0	0	0	AND	1100110011001100	1010101010101010	1000100010001000
0	0	0	1	NAND	1100110011001100	1010101010101010	0111011101110111
0	0	1	0	OR	1100110011001100	1010101010101010	1110111011101110
0	0	1	1	NOR	1100110011001100	1010101010101010	0001000100010001

Table-2

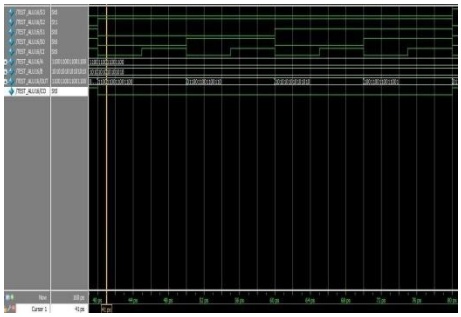

Fig.12. Results for 2nd 4-operations (Buffer A,EX-OR, Buffer B, EX-NOR)

The Figure-12 shows the result for second 4-operations Buffer-A, EX-OR, Buffer B, EX-NOR. These are selected depending on select lines s3-s0as shown in the table-3

S3	S2	S1	S0	Operation	Input(A)	Input(B)	OUTPUT
0	1	0	0	Buffer-A	1100110011001100	1010101010101010	1100110011001100
0	1	0	1	EX-OR	1100110011001100	1010101010101010	0110011001100110
0	1	1	0	Buffer-B	1100110011001100	1010101010101010	1010101010101010
0	1	1	1	EX-NOR	1100110011001100	1010101010101010	1001100110011001

Table-3

Table-3 shows the second 4 operations selected depending on select lines. The operation is executed on two inputs A & B and output is given out. The operations done by Buffer-A, EX-OR, Buffer B, and EX-NOR.

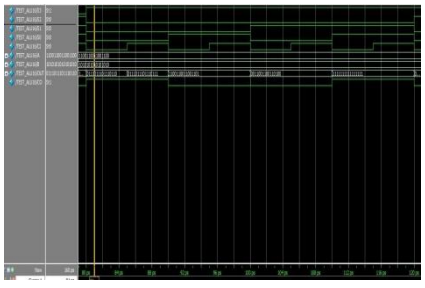

Fig.13. Results for 3rd 4-operations (ADD, INCREMENT, 2'S COMPLEMENT, SET)

These are selected depending on select lines s3-s0 as shown in the table-4

S3	S2	S1	S0	Operation	Input(A)	Input(B)	OUTPUT
1	0	0	0	ADD	1100110011001100	1010101010101010	0111010111010110
1	0	0	1	INCREMENT A	1100110011001100	1010101010101010	1100110011001101
1	0	1	0	2'S COMPLEMENT	1100110011001100	1010101010101010	0011001100110100
1	0	1	1	SET	1100110011001100	1010101010101010	1111111111111111

Table-4

Table-4 shows the Third 4 operations selected depending on select lines. The operation is performed on 2 input's A & B and output is given out here Co is consider for ADD & SET operations.

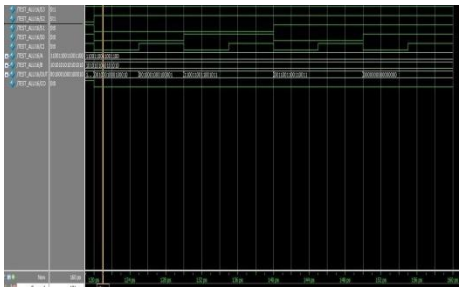

Figure-14. Results for 4th 4-operations
(SUBTRACTION, DECREMENT, NOT, CLEAR)

The Figure-14 shows the execution of Fourth 4-operations SUBTRACTION, DECREMENT, NOT, CLEAR. These are selected depending on select lines s3-s0 as shown in the table-4

S3	S2	S1	S0	Operation	Input(A)	Input(B)	OUTPUT	Co
1	1	0	0	SUBSTRACTION	1100110011001100	1010101010101010	0010001000100010	0
1	1	0	1	DECREMENT A	1100110011001100	1010101010101010	1100110011001011	1
1	1	1	0	NOT	1100110011001100	1010101010101010	0011001100110011	0
1	1	1	1	CLEAR	1100110011001100	1010101010101010	0000000000000000	0

Table-5 shows the Fourth 4 operations selected depending on select lines. The operation is performed on 2 input's A & B and output is given out. The operations done by SUBTRACTION, DECREMENT, NOT, and CLEAR.

CONCLUSION

In this paper 16-bit reversible ALU is developed and executed in Verilog by using MODEL SIM ALTERA 6.6d. The main aim is to design this paper is to improve the ALU features by increase to 15-operations and increase the thickness to 16-bit. For future research ALU can be extended to 32-bit and 64-bit and more features can also be added. This design is verified by using Verilog and it contain input to output for one-way functionality, if we can design the reversible logic circuit using tools support for 2-way functionality the reversible logic result can be simulated and analysed in much better.

REFERENCES

1. Y.syamala & A.V.N Tilak , "Reversible Arthematic logic unit", IEEE conference, Page no: 207-211,2009.
2. R. Landauer, "Irreversibility and heat generation in the computing process", IBM J. Research and Development, 5(3), pp.: 183-191, 1961.
3. C. H. Bennett, "Logical reversibility of computation", IBM J. Research and Development, 17, pp.: 525 – 532, November1973.
4. Feynman, R., "Quantum mechanical computers" Optics, News, and pp.: 11-20 11, 1985.
5. Toffolli T, "Reversible computing", *Technical Memo* MIT/LCS/TM-151, MIT Laboratory for Computer Science (February), 1980.
6. Fredkin, E. and T. Toffolli, "Conservative logic", Intl. J. Theoretical Phy., 21, pp.: 219-253, 1982.
7. B.Raghukanth, B.Murali Krishna, M. Sridhar, V.G. SanthiSwaroop"A DISTINGUISH BETWEEN REVERSIBLE AND CONVENTIONAL LOGIC GATES", International Journal of Engineering Research and Applications (IJERA), Vol. 2, pp.148-151, Issue 2, Mar-Apr 2012.
8. Y.Syamala and A. V. N. Tilak, "Reversible Arithmetic and Logical Unit" International Journal of Electrical and Electronics Engineers (IEEE), 2011.
9. R.Landauer, "Irreversibility and Heat Generation in the Computational Process", IBM Journal of Research and Development, Vol. 05, 1961.
10. C.H. Bennett, "Notes on the History of Reversible Computation", IBM Journal of Research and Development, Vol. 32, 1998.
11. Monika Rangari, Prof. Richa Saraswat and Dr. Rita
12. Jain, "Design of Reversible Logic ALU using Reversible logic gates with Low Delay Profile" International Journal of Advanced Research in Computer and Communication Engineering, Vol. 04, Issue 4, April 2015.
13. Ravi Raj Singh, Sapna Upadhyay, Saranya S, Soumya,Jagannath KB and Hariprasad SA, "Efficient Design of Arithmetic and Logical Unit using Reversible Logic Gates" International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 3, Issue 4, April 2014.