16 BIT ARITHMETIC AND LOGIC UNIT USING REVERSIBLE LOGIC GATES

Thota Sai Abhishek, Peddamile Sreemani, Majjigapu Pavan Kumar Reddy
Student, Student, Student
Electronics and Communication Engineering,
Gandhi Institute of Technology and Management, Hyderabad, India.

Abstract: **Reversible or data lossless circuits have applications in advanced computing, correspondence, PC illustrations and cryptography. Reversibility assumes a big job when vitality proficient calculations are thought of. Reversible rationale is employed to reduce the force scattering that happens in traditional circuits by forestalling the loss of knowledge. This paper proposes a reversible structure of a 16 bit ALU. This ALU comprises of eight activities, three arithmetic and five logical tasks. The number-crunching tasks incorporate addition, subtraction and multiplication and therefore the legitimate activities incorporate NAND, AND, OR, NOT and XOR. All the modules are being structured utilizing the essential reversible gates. The force and defer examination of the various sub modules is performed and a correlation with the traditional circuits is likewise done.**

Index Terms **– Reversible logic gates, 16-bit ALU, Low Power Dissipation.**

1. INTRODUCTION

In present day VLSI framework power scattering is high because of quick exchanging of inner signs. The intricacy of VLSI circuits increments with every year because of pressing increasingly more rationale components into littler volumes. Consequently power dispersal has become the primary territory of worry in VLSI structure. Reversible rationale has its nuts and bolts from thermodynamics of data preparing. As indicated by this, conventional irreversible circuits produce heat because of the loss of data during calculation. So as to keep away from this data misfortune the regular circuits are displayed utilizing reversible rationale. Landauer indicated that the circuits structured utilizing irreversible components disseminate heat because of the loss of data bits . It is demonstrated that the loss of one bit of data brings about dispersal of KT*log2 joules of warmth vitality where K is the Boltzmann consistent and T is the temperature at which the activity is performed.. A gate is considered to be reversible just if for every single contribution there is a one of a kind yield task. Henceforth there is a coordinated mapping between the information and yield vectors. A reversible rationale gate is a n – input, n-yield gadget showing that it has same number of sources of info and yields. A circuit that is worked from reversible gates is known as reversible rationale circuit. Right now, plan a 16 piece reversible ALU that can perform eight tasks at the same time. The eight tasks incorporate addition, subtraction, multiplication, division, AND, OR, NOT and XOR. All the modules are reenacted in modelsim SE 6.5 and incorporated utilizing Xilinx ISE 14.7.

2. REVERSIBLE LOGIC GATES

Reversible rationale is picking up significance in zones of CMOS structure due to its low power dissemination. The customary gates like AND, OR, XOR are for the most part irreversible gates. Think about the instance of customary AND gate. It comprises of two sources of info and one yield. Thus, the slightest bit is lost each time a calculation is done. As indicated by reality table appeared in Fig.1, there are three data sources (1, 0), (0, 1) and (0, 0) that compares to an yield zero. Subsequently it is preposterous to expect to decide a one of a kind info that brought about the yield zero. So as to make a gate reversible extra info and yield lines are added with the goal that a coordinated mapping exists between the information and yield. This forestalls the loss of data that is primary driver of intensity dispersal in irreversible circuits. The information that is added to a m x n capacity to make it reversible is known as steady information (CI). All the yields of a reversible circuit need not be utilized in the circuit. Those yields that are not utilized in the circuit is called as garbage output(GO). The quantity of trash yield for a specific reversible gate isn't fixed. The two requirements for a reversible gate are they should not have a feedback and the GO should be as least as possible.

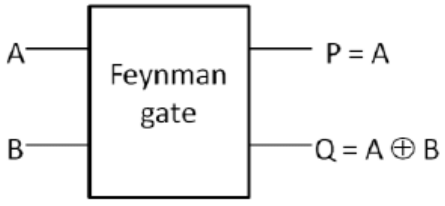

Figure 1 - 2 bit Feynman gate

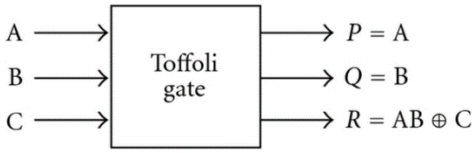

Figure 2 – Toffoli gate

3. PROPOSED DESIGN

The architectural design on which the 16 bit ALU is based on is shown in the diagram below

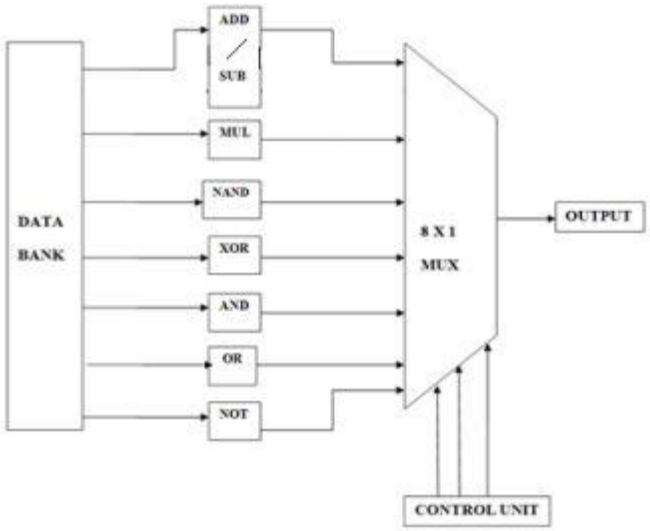

Figure 3 – ALU architecture

4. 16 BIT ADDER/SUBTRACTOR

The twofold full adder/subtractor handles each contribution alongside a convey in/obtain in that is produced as do/get out from the expansion of past lower request bits. In the event that two n bit twofold numbers are to be included or then again subtracted then n twofold full adder/subtractors ought to be fell. An equal adder/subtractor is the interconnection of various full adder/subtractor and applying the information sources at the same time. Right now 4 piece equal adder/subtractor circuit is structured utilizing a 4x4 reversible DKG gate. Fig.6 shows the reversible DKG gate. This gate can goes about as a adder or subtractor relying upon its control input A. when A is zero the gate acts as a full viper and when A is one the gate carries on as a subtractor.
For executing a 4-bit reversible adder/subtractor, 4 DKG gates are required. Thus the all out gate mean a 4-bit viper/subtractor is 4 and that for a 16 piece adder/subtractor is 16. Here the convey in (Cin) is spread starting with one gate then onto the next gate. The module is planned utilizing VHDL, mimicked in modelsim and combined utilizing Xilinx 14.7.

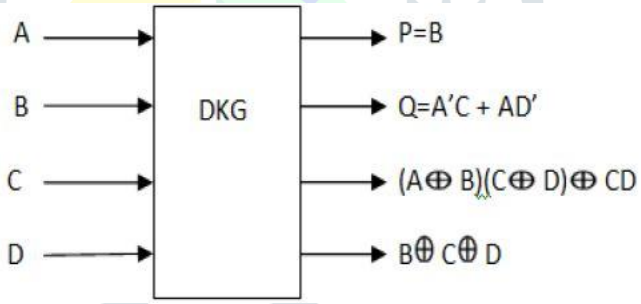

Fig 4 – 4-bit DKG gate

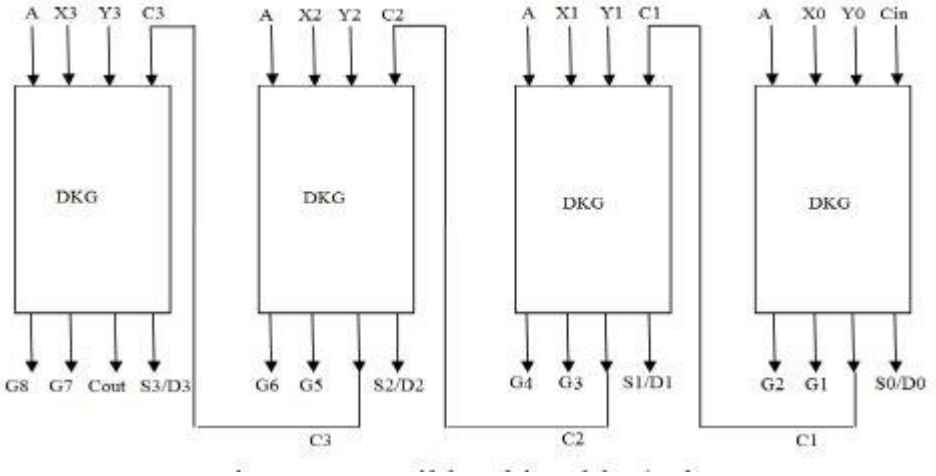

Fig 5 – Reversible 4-bit adder/subtractor

5. 16 BIT MULTIPLIER

Right now n x n reversible multiplier is planned utilizing TSG and Fredkin gates. The fundamental cell for such a multiplier is the full viper square. It is obvious from the figure that the TSG gate can go about as full viper when one of the sources of info is allocated a steady estimation of zero. When c is relegated as zero the TSG gate goes about as a full viper and the entirety and convey yield is acquired at S and R separately.

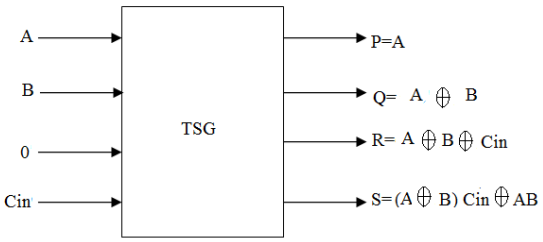

Figure 6 - Reversible TSG gate as a full adder

Augmentation incorporates two fundamental advances. Initial step is the age of fractional items and the subsequent one is the expansion of the created incomplete items. The halfway items can be created utilizing Fredkin gate. At the point when one of the contribution to the Fredkin gate is doled out a consistent estimation of zero, at that point the Fredkin gate carries on like an AND gate and incomplete item can be produced as appeared in Fig.10. The created incomplete items would now be able to be included utilizing Parallel wave convey viper that is structured by falling the TSG gate. For a 4x4 multiplier, 16 Fredkin gates are required for creating the whole incomplete items. Three phases of reversible wave convey adder utilizing TSG gate is required for including the above created incomplete items and getting the last 8bit item.

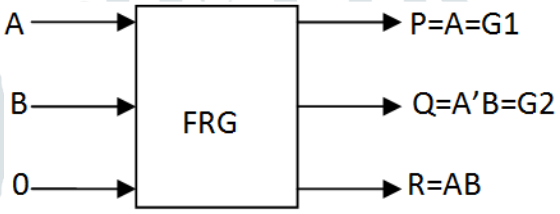

Figure 7 – FRG gate used as AND

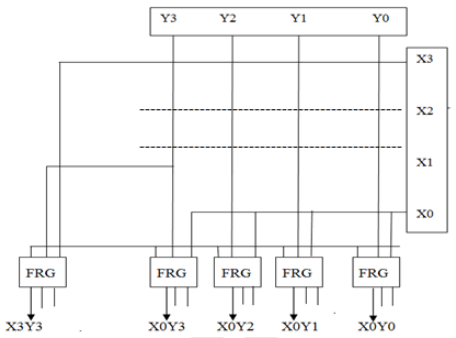

Figure 8 - Generation of partial product

6. MULTIPLIER ALGORITHM

The accompanying calculation can be utilized to plan a 8x8 multiplier utilizing 4x4 multiplier.

1. Leave An and B alone two 8 piece numbers that are to be duplicated.
2. Gap An into equivalent parts A1 and A0 with the end goal that A0 demonstrates the 4bit LSB and A1 shows the 4bit MSB. Gap B likewise similarly as B0 and BI.
3. Duplicate A0 and B0 utilizing reversible 4bit multiplier. This structures the main fractional item PR1. Hold the 4bit LSB of PR1.
4. Presently duplicate A1 and B0. This structures the fractional item PR2. At that point duplicate A0 and B1. This structures fractional item PR3.
5. Include PR2, PR3 and the 4bit MSB of PR1 utilizing reversible equal adder planned from TSG gate. This structures the brief outcome TR1. Hold the 4bit LSB of TR1.
6. Increase A1 and B1 bringing about the fractional item PR4. Presently include PR4 and the rest of the bits of TR1 (barring the 4bit LSB) bringing about the impermanent outcome TR2.
7. Presently connect TR2 with the 4bit LSB of TR1 and PR1. This structures the last 16 piece item. The equivalent can be applied for planning higher multipliers utilizing lower multipliers.

Henceforth the all out gate mean a 4x4 multiplier is 29 gates (16 Fredkin gates +13 TSG gates).

7. 16 BIT LOGICAL UNIT

We will be using a PFAG gate and one Feynman gate for realizing the logical circuit of the ALU of 1-bit size

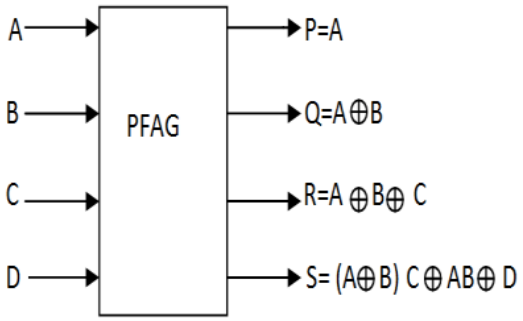

Figure 9- 4X4 PFAG gate

A	B	C	D	P	Q	R	S
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	1	0
1	0	0	1	1	1	1	1
1	0	1	0	1	1	0	1
1	0	1	1	1	1	0	0
1	1	0	0	1	0	0	1
1	1	0	1	1	0	0	0
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

Table 1 – Truth table of PFAG

When CD=00 the PFAG square goes about as an AND gate and the yield of the AND activity between A furthermore, B is acquired at S. When CD=01 the NAND activity among An and B is performed and the yield is gotten at S. When CD=10 the OR activity among An and B is performed and gotten at S. The XNOR activity among An and B is acquired at R. When CD=11 the NOT activity of An is done and gotten at P1. The XOR activity of An and B is gotten at Q independent of the estimation of C and D.

8. SIMULATION RESULTS

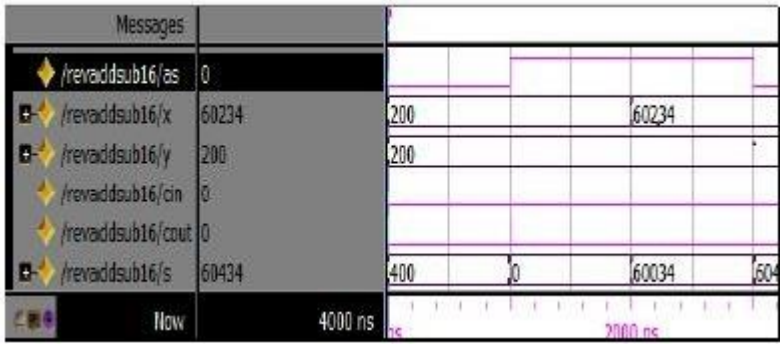

Figure 10- 16 bit adder/subtractor

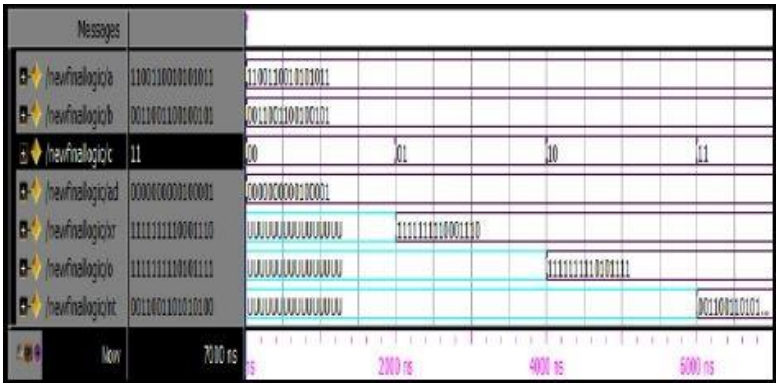

Figure 11- 16 bit logical circuit

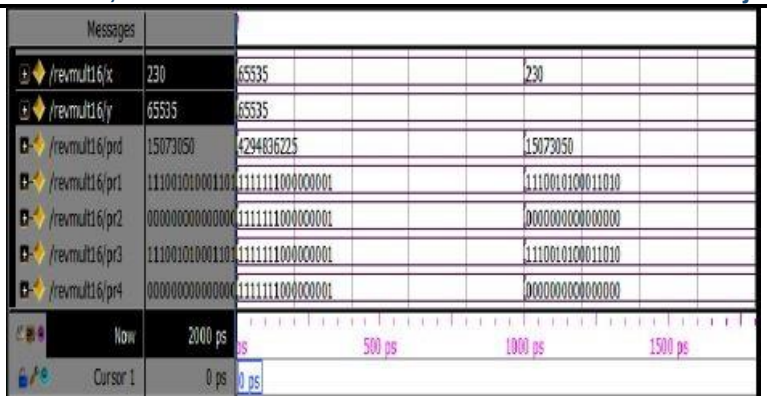

Figure 12- 16x16 multiplier

ACKNOWLEDGEMENT

We would like to thank our Pro Vice Chancellor- Dr.N Siva Prasad, Head of the Department- Dr.K Manjunathachari and our respected guide-Mr V.Shiva Prasad who have helped us in making this project. Without their support, this project would not have been made possible.

I would also like to thank the different resources and papers which acted as a guide in helping us do the project.

REFERENCES

[1] Ravish Aradhya H V, Praveen Kumar B V, Muralidhara K N "Design of Control unit for Low Power ALU Using Reversible Logic" *International Journal of Scientific & Engineering Research Volume 2, Issue 9,* September-2011.

[2] CPU designers have used a variety of names for the arithmetic logic unit, including "ALU", "integer execution unit", and "E-box". Paul V. Bolotoff. "Functional Principles of Cache Memory" 2007.

[3] 1990. Aviienis, A. "Signed-digit Number Representations for Fast Parallel Arithmetic," in *Computer Arithmetic*, Vol. II (ed. E. E. Swartzlander Jr), 54-65. Los Alamitos, CA: IEEE Computer Society Press. (Reprinted from *IRE Trans. El. Comp.*, EC-10 (1961), 389-400.)

[4] 1994. Ercegovac, M., and Lang, T. *Division and Square Root*. Boston: Kluwer Academic Publishers.

[5] Y.Taur and T.H.Ning.Fundamentals of modern VLSI DevicesNew York: Cambridge Univ. Press .1998.