Original Article

Design And Implementation Of Modules Of 1-Bit ALU Using Proteus Design Suite

Vanshika Kumari[1*], Rajinder Tiwari[2*]

[1]Department of ECE, Model Institute of Engineering & Technology, Jammu, India
[2*]Dy. COE, Examination Cell, BBD University, Lucknow, India
[2*]Orcid ID: https://orcid.org/0000-0003-1734-5892
Email: rajinder.tiwari@ieee.org

*Corresponding Author: Dr. Rajinder Tiwari
[2*]Dy. COE, Examination Cell, BBD University, Lucknow, India
[2*]Orcid ID: https://orcid.org/0000-0003-1734-5892
Email: rajinder.tiwari@ieee.org
DOI: 10.47750/pnr.2022.13.S01.300

Received Date: 2022-08-05 **Publishing Date:** 2022-08-20

Abstract

ALU has proved to be an important module of a processor so as to carry out the most important computation for the assigned task in the domain of digital system based applications, for example, Digital Signal Processing. This has been the most important and desired application in the upcoming technology because of which it has proved to be the most aspired field of research under considerations. Keeping this in consideration, the author has put forward a discussion based on the performance analysis of an ALU of the processor with the help of the most dominant modules of the ALU on which the entire computation is based on. The entire simulation task has been carried out with the help of the Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. This software can provide all the resources needed for the test and it directly evaluates the correctness of hardware circuit design, directly debug the software with hardware schematic diagram and verify the function of the whole design and test controllable. This paper proposes a proteus design of different components of ALU. ALU performs different arithmetic and logical operations. The implementation of different components like adder, subtractor, encoder, decoder and multiplexer are carried out on this paper.

Keywords: ALU, Proteus Design Suite, Multiplexer, Adder, Subtractor, Encoder, Decoder

1. INTRODUCTION

In the current scenario of circuit designing, a tremendous need for different software for testing the circuits before making the hardware is seen. Major researches have been focusing on the functionality of digital circuits and leakage power expenditure. Proteus Design Suite is a software tool which is designed to test the electronic components and circuits. An ALU is a digital circuit which performs two operations i.e. arithmetic and logical operations. It is the essential building block of the Central Processing Unit in the computer. Different circuits like adder, multiplexer, subtractor etc. are used in order to implement the ALU. This paper discusses about various components of 1-bit ALU and their simulation on Proteus [1-5].

2. ARITHMETIC LOGIC UNIT (ALU)

ALU stand for Arithmetic and Logic unit. An arithmetic-logic unit is the part of a central processing unit that carries out arithmetic and logic operations on the operands in computer instruction words. In some processors, the ALU is divided into two units: an arithmetic unit (AU) and a logic unit (LU). Some processors contain more than one AU: for example, one for fixed-point operations and another for floating-point operations. ALUs serve as a combinational digital circuit that performs arithmetic and bitwise operations on binary numbers.

This is a foundational building block of arithmetic logic circuits for numerous types of control units and computing circuits including central processing units (CPUs), FPUs and graphics processing units. Except performing calculations related to addition and subtraction, ALUs handle the multiplication of two integers as they are designed to execute integer calculations; hence, its result is also an integer. However, division operations commonly may not be performed by ALU as division operations may produce a result in a floating-point number.

Instead, the floating-point unit (FPU) usually handles the division operations; other non-integer calculations can also be performed by FPU. An ALU usually takes two inputs, called operands, and a code, called OPCODE, which specifies the operation to be performed on the operands. ALU also has a result output which is the result of the operation on the operands. In some designs, usually the values (operands) fed to the ALU and/or the result generated from ALU are read from/stored in registers. A general symbolic representation of an ALU is shown in Figure 1 [6-15].

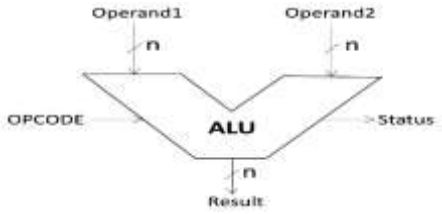

Fig-1: A Symbolic Representation of an ALU [16]

3. FULL ADDER DESIGN

Table-1: Truth table of Full Adder

Inputs			Outputs	
A	**B**	**C**	**Sum**	**Carry**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

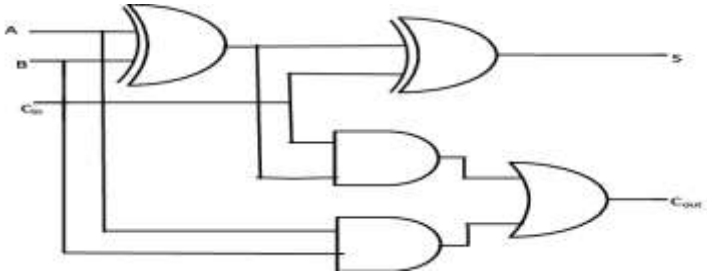

Fig-2: Circuit diagram of Full Adder [17]

Full Adder is the adder that adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM. A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to another. A 1-bit full adder adds three operands and generates 2-bit results. The circuit diagram and truth table of full adder is shown in fig 2 and table1 respectively.

$$Sum = (A \oplus B) \oplus C_{in} \quad (1)$$

$$Carry = (A \cdot B) \quad (2)$$

4. FULL SUBTRACTOR DESIGN

$$Difference = \bar{A}\bar{B}B_{in} + A\bar{B}B_{in} + ABB_{in} + \bar{A}BB_{in} \quad (3)$$

$$B_{out} = BB_{in} + \bar{A}B + \bar{A}B_{in} \quad (4)$$

Table-2: Truth table of Full Subtractor

Inputs			Outputs	
A	**B**	**C**	**Difference**	**Borrow**
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

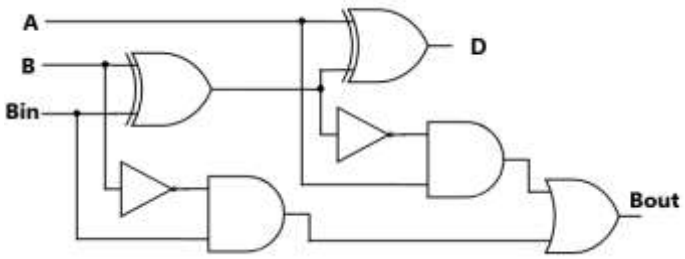

Fig-2: Circuit diagram of Full Subtractor [18]

A full subtractor is a combinational circuit that performs subtraction of two bits, one is minuend and other is subtrahend, considering borrow of the previous adjacent lower minuend bit. This circuit has three inputs and two outputs**.** The three inputs A, B and Bin, denote the minuend, subtrahend, and previous borrow, respectively.
The two outputs, D and Bout represent the difference and output borrow, respectively. The circuit diagram of full subtractor is shown in fig.2 and truth table of the full subtractor is shown in table 2 [19-25].

5. 4:1 MULTIPLEXER DESIGN

In 4:1 MUX, there will be 4 input lines and 1 output line. And to control which input should be selected out of these 4, we need 2 selection lines. Thus, it is evident from the fig.4 below that I0, I1, I2 and I3 are the input lines and S1, S2 are the two selection lines.

$$Y = \overline{S_0 S_1} D_0 + \bar{S_0} S_1 + D_1 + S_0 \bar{S_1} D_2 + S_0 S_1 D_3 \quad (5)$$

The combination of binary numbers given as a selection line will determine the output of the MUX. The circuit diagram and truth table of the 4:1 multiplexer is shown below in the fig 3 and table 3 respectively [26-32].

Table-3: Truth table of 4:1 Mux

Selection Lines		Output
S_0	S_1	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

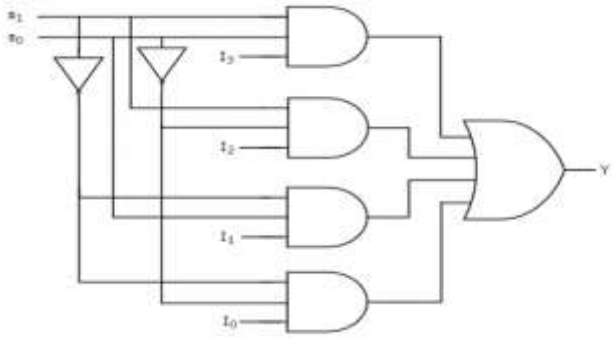

Fig-3: Circuit diagram of 4:1 Mux [33]

6. 4 to 2 ENCODER DESIGN

An Encoder is a combinational circuit that performs the reverse operation of Decoder. It has maximum of 2^n input lines and 'n' output lines. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2^n input lines with 'n' bits.

$$A_1 = Y_3 + Y_2 \quad (6)$$

$$A_0 = Y_3 + Y_1 \quad (7)$$

It is optional to represent the enable signal in encoders. Let 4 to 2 Encoder has four inputs Y_3, Y_2, Y_1 & Y_0 and two outputs A_1 & A_0. At any time, only one of these 4 inputs can be '1' in order to get the respective binary code at the output. The circuit diagram and truth table of the encoder is shown below in the fig 4 and table 4 respectively [34-42].

Table-4: Truth table of 4:2 Encoder

Inputs				Outputs	
Y_3	Y_2	Y_1	Y_0	A_1	A_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

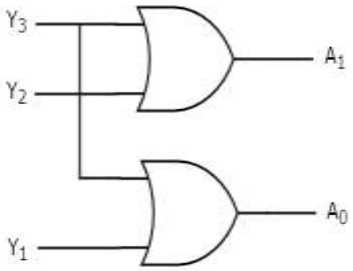

Fig-4: Circuit diagram of 4:2 Encoder [43]

7. 2 to 4 DECODER DESIGN

Decoder is a combinational circuit that has 'n' input lines and maximum of 2^n output lines. One of these outputs will be active High based on the combination of inputs present, when the decoder is enabled. That means decoder detects a particular code.

The outputs of the decoder are nothing but the min terms of 'n' input variables lines, when it is enabled. In 2:4 decoder, decoders have two inputs namely A0, A1, and four outputs denoted by D0, D1, D2, and D3. The following figure shows the circuit diagram and truth table of 2 to 4 decoder [44-47].

$$Y_3 = E.A_1.A_0 \quad (8)$$

$$Y_2 = E.\bar{A}_1.A_0 \quad (9)$$

$$Y_2 = E.A_1.\bar{A}_0 \quad (8)$$

$$Y_1 = E.\bar{A}_1\bar{A}_0 \quad (8)$$

Table-5: Truth table of 2 to 4 Decoder

Enable		Inputs	Outputs			
E	**A_1**	**A_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

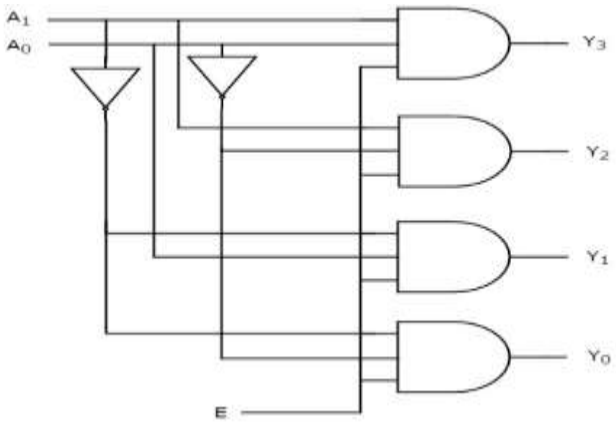

Fig-5: Circuit diagram of 2 to 4 Decoder [48]

8. PERFORMNCE ANALYSIS WITH PROTEUS DESIGN SUITE

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

Proteus can provide all the resources needed for the test, directly evaluate the correctness of hardware circuit design, directly debug the software with hardware schematic diagram, verify the function of the whole design, and test controllable, easy to evaluate and easy to implement. Schematic capture in the Proteus Design Suite is used for both the simulation of designs and as the design phase of a PCB layout project.

It is therefore a core component and is included with all product configurations. There are different modules of proteus. Some of them are as following.

Schematic Capture

Schematic capture in the Proteus Design Suite is used for both the simulation of designs and as the design phase of a PCB layout project. It is therefore a core component and is included with all product configurations.

Microcontroller Simulation

The micro-controller simulation in Proteus works by applying either a hex file or a debug file to the microcontroller part on the schematic. It is then co-simulated along with any analog and digital electronics connected to it. This enables its use in a broad spectrum of project prototyping in areas such as motor control, temperature control and user interface design. It also finds use in the general hobbyist community and, since no hardware is required, is convenient to use as a training or teaching tool.

PCB Design

The PCB Layout module is automatically given connectivity information in the form of a netlist from the schematic capture module. It applies this information, together with the user specified design rules and various design automation tools, to assist with error free board design. PCB's of up to 16 copper layers can be produced with design size limited by product configuration.

3D Design

The 3D Viewer module allows the board under development to be viewed in 3D together with a semi-transparent height plane that represents the boards enclosure. STEP output can then be used to transfer to mechanical CAD software such as Solid works or Autodesk for accurate mounting and positioning of the board.

9. RESULT AND DISCUSSIONS

9.1 Simulation of Full Adder

The functional correctness of full adder is verified with the use of Proteus to make the corresponding simulation. In the below figure a, b, c are the inputs of the full adder and the Cin and s are the outputs. The following waveform is carried out in the Proteus software.

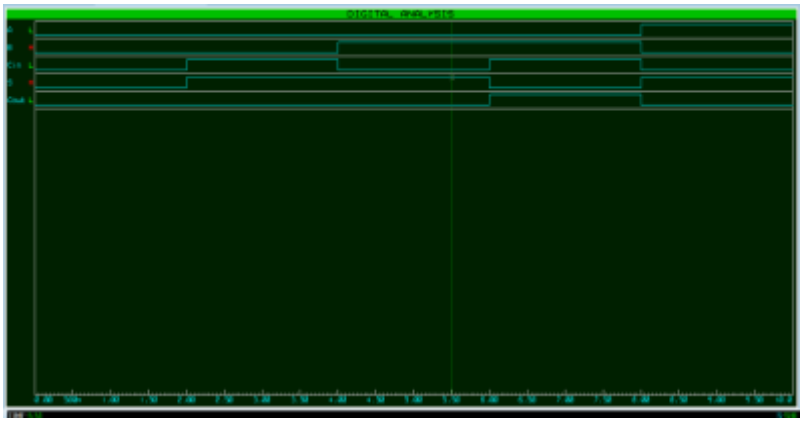

Fig-6: Simulation waveform of Full adder in Proteus

9.2. Simulation of Full Subtractor

The following figure shows the proteus simulation of full subtractor. In the below figure the a, b, bin are the inputs of the full subtractor and the d, Cout are the outputs of the full subtractor.

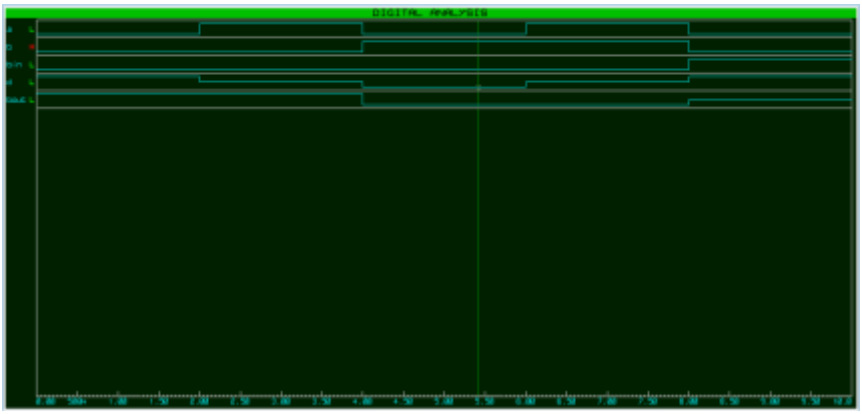

Fig-7: Simulation waveform of Full subtractor in Proteus

9.3 Simulation of 4:1 Multiplexer

The following figure shows the proteus simulation of 4:1 multiplexer. In the figure, a,b,c,d are the inputs of the multiplexer, s0 and s1 are the select lines and y is the output.

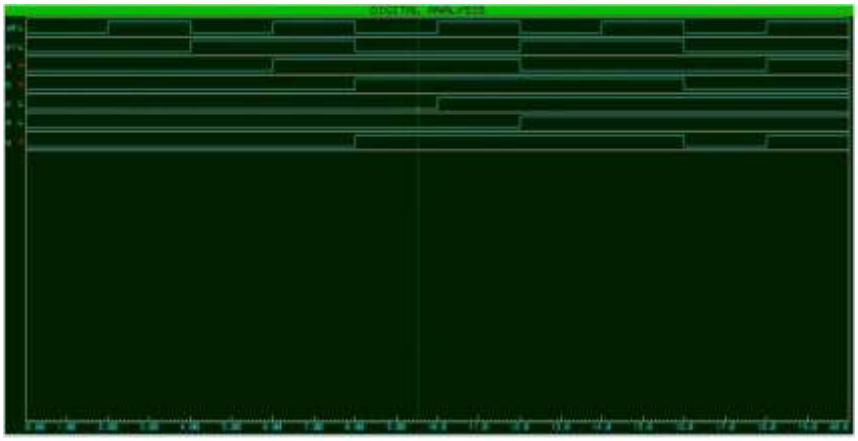

Fig-8: Simulation waveform of Full subtractor in Proteus

9.4. Simulation of 4 to 2 Encoder

The following figure shows the simulation of 4 to 2 encoder. In the below figure y0, y1, y2, y3 are the inputs of the encoder and a0 and a1 are the outputs of the encoder.

Fig-9: Simulation of 4 to 2 Encoder in Proteus

9.5. Simulation of 2 to 4 Decoder

In the below figure, the functional correctness of decoder is verified with the use of Proteus to make the corresponding simulation. In the below figure a0, a1 are the inputs of the decoder and the d0, d1, d2, d3 are the outputs. The following waveform is carried out in the Proteus software.

Fig-10: Simulation of 2 to 4 Decoder in Proteus

CONCLUSION

The 1-bit ALU is designed by integrating various modules like full adder, full subtractor, multiplexer, encoder, decoder. The performance evaluation of these modules is also carried out in the proteus software. The results and simulations are also carried out using proteus tool. As a future work different modules of 1-bit ALU can also be designed and included into this ALU using Proteus tool.

ACKNOWLEDGEMENT

The author is thankful to Ms. Alka Das (Chairperson, BBD Group), Mr. Viraj Sagar Das (President BBD Group), Prof. (Dr.) Arun Kumar Mittal (Vice-Chancellor, BBD University), Prof. (Dr.) Manish Gupta (COE BBD University) and members of Examination Department for their motivation, kind cooperation, and suggestions.

REFERENCES

1. Priyal Grover and Hemant Verma, “Design, Layout and Simulation of 8 bit Arithmetic and Logical Unit” Inter-national Journal of Electrical and Electronics Engineers (IEEE),Vol. 07, Issue 02, July- December 2015.
2. Shefali Mamtaj, Biswajit Das, Anurima Rahama “An Optimized Realization of ALU for 12 Operation by using a Control Unit of Reversible Gates”, International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), Vol. 04, Issue 01, January 2014.
3. Ajay Kumar Sharma and Anshul Jain, “Design of Low Power Low Area Arithmetic and Logical Unit” Inter-national Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol. 02, Issue 12, December 2014.
4. Chetan Kumar, Dr. Rekha.K .R and Dr. Natraj K .R, “Implementation of 16 bit Arithmetic and Logical Unit using Toffoli Reversible Logic Gate” International Journal of Innovative Science, Engineering and Technology (IJISET), Vol. 01, Issue 06, August 2014.
5. Vijay G. Roy, P. R. Indurkar and D.M. Khatri, **“**Low Power 8 bit Quantum ALU Implementation using Reversible Logic Structure”, International Journal of Science and Research (IJSR), Vol. 03, Issue 07, july 2014.
6. Akanksha Dixit and Vinod Kapse, “Arithmetic and Logical Unit Design using Reversible Control Unit”, International Journal of Engineering and Innovative Technology (IJEIT), Vol. 01, Issue 06, June 2012.
7. Zhijin Guan, Wenjuan Li, Weiping Ding, Yueqin Hang and Lihui Ni, “An Arithmetic and Logical Unit Design based on Reversible Logic Gates”, International Journal of Electrical and Electronics Engineers (IEEE), 2011.
8. Avinash G. Keskar and Vishal R. Satpute, “Design of 8 bit Novel Reversible Arithmetic and Logical Unit” Inter-national Journal of Electrical and Electronics Engineers (IEEE),2011.
9. Y.Syamala and A. V. N. Tilak, “Reversible Arithmetic and Logical Unit” International Journal of Electrical and Electronics Engineers (IEEE), 2011.
10. R. Landauer, “Irreversibility and Heat Generation in the Computational Process”, IBM Journal of Research and Development, Vol. 05, 1961.
11. C.H. Bennett, “Notes on the History of Reversible Computation”, IBM Journal of Research and Development, Vol. 32, 1998.
12. Monika Rangari, Prof. Richa Saraswat and Dr. Rita Jain, “Design of Reversible Logic ALU using Reversible logic gates with Low Delay Profile” International Journal of Advanced Research in Computer and Communication Enginee-ring, 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.
14. Darshan H, Mohanraj R, Kavya H B, Monisha U K and Saroja Maralabhavi, “Design and Synthesis of 8 Bit Reversible Arithmetic & Logical Unit (ALU)” ITSI Transactions on Electrical and Electronics Engineering (ITSI-TEEE), Vol. 3, Issue 03, 2015.
15. Khushboo Ahirwar, Sachin Bandewar and Anand Kumar Singh, “FPGA Implementation ALU Based on Reversible Logic” International Journal of Engineering Research & Technology (IJERT), Vol. 3 Issue 1, January 2014.
16. Pradeep singla and Naveen Kr. Malik, “ A Cost Effective Design of Reversible programmable logic array” International Journal Of Computer Application, Vol. 41, March 2012.
17. [C.H. Bennett, “Logical Reversibility of Computation” IBM J. Research and Development, November 1973.
18. M. Nielsen, I. Chuang. Quantum computation and quantum information. Cambridge, UK: Cambridge University Press, 2000.
19. E.Fredkin, T.Toffoli, “Conservative logic”, International Journal of Theoretical Physics, 21:219-253, 1982.
20. Michael P. Frank, “Reversible Computing,” invited article, Developer 2.0 magazine, Jasubhai Digital Media, January 2004.
21. S. Kim, J.H. Kwon, S. I. Chiae, An 8-b nRERL microprocessor for ultra-low-energy applications. ASP-DAC 2001: 27-28
22. C.H. Bennett, “Logical Reversibility of Computation”, IBM J. Research and Development, pp. 525-532, November 1973.
23. C H Bennett, "Notes on the History of Reversible Computation", IBM Journal of Research and Development, vol. 32, pp. 16-23, 1998.
24. Peres, “Reversible logic and quantum computers”, phys. Rev .A ,Gen. Phys., vol. 32, no. 6, pp. 32663276, Dec. 1985.
25. J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, ``Embedded deterministic test," IEEE Trans. Computer Aided Des. Integr. Circuits Syst., vol. 23,no. 5, pp. 776_792, May 2004.
26. Brglez, D. Bryan, K. Kozminski, "Combinational Profiles of sequential benchmark circuits", Proc. IEEE ISCAS, pp. 1929-1934, 1989.
27. S. Abu-Issa and S. F. Quigley, “Bit-swapping LFSR and scan-chain ordering: A novel technique for peak- and average-power reduction in scan based BIST,” IEEE Trans. Computer Aided Des. Integrated Circuits Syst., vol. 28, no. 5, pp. 755–759, May 2009.
28. J. Rajski, J. Tyszer, G. Mrugalski, and B. Nadeau-Dostie, “Test generator with preselected toggling for low power built-in self-test,” in Proc. Eur.Test Symp., May 2012, pp. 1–6.
29. Y. Sato, S. Wang, T. Kato, K. Miyase, and S. Kajihara, “Low power BIST for scan-shift and capture power,” in Proc. IEEE 21st Asian Test Symp., Nov. 2012, pp. 173–178.
30. K. Moghaddam, J. Rajski, M. Kassab, and S. M. Reddy, “At-speed scan test with low switching activity,” in Proc. IEEE VLSI Test Symp., Apr. 2010, pp. 177–182.
31. Peres, “Reversible logic and quantum computers”, phys. rev. A, Gen. Phys., vol. 32, no. 6, pp. 32663276, Dec. 1985.
32. H.G Rangaraju, U. Venugopal, K.N. Muralidhara, K. B. Raja,” Low power reversible parallel binary adder / subtractor” arXiv.org/1009.6218,2010. 12) J.M. Rabaey and M. Pedram, “Low Power
33. Akanksha Dixit and Vinod Kapse (2012), “Arithmetic & Logic Unit (ALU) Design Using Reversible Control Unit”, IJEIT, Vol. 1, No. 6.
34. Bennett C H (1973), “Logical Reversibility of Computation”, IBM J. Research and Development, Vol. 17, pp. 525-532.
35. Himanshu Thapliyal and Nagarajan Ranganathan (2011), “A New Reversible Design of BCD Adder”, IEEE Conference on Design and Automation, pp. 1-4.
36. Jayashree H V, Nagamani A N and Bhagyalakshmi H R (2012), “Modified TOFFOLI GATE and its Applications in Designing Components of Reversible Arithmetic and Logic Unit”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, No. 7, ISSN: 2277 128X.
37. 6. Lekshmi Viswanath and Ponni M (2012), “Design and Analysis of 16 Bit Reversible ALU”, IOSRJCE, Vol. 1, No. 1, p. 46, ISSN: 2278-0661.
38. Rajinder Tiwari, Deepika Bastawade, Preeta Sharan, Anil Kumar, “Performance Analysis of Reversible ALU in QCA” Indian Journal of Science & Technology, vol: 10(29), pp: 01-05, 2017.

39. Rajinder Tiwari, Anil Kumar, Preeta Sharan “Design and Implementation of 4:1 Multiplexer for Reversible ALU using QCA” published in the proceeding of 2nd International Conference Microelectro-nics and Telecommunication Engineering (ICMETE 2018), Aug 2018, PP No: 191-196.
40. Rajinder Tiwari*, Vikas Rajiv, Preeta Sharan, Anil Kumar. "An Innovative Low Power Reversible ALU for Quantum Processor using QCA", International Journal of Innovative Technology and Exploring Engineering, 2019.
41. Lihui Ni, Zhijin Guan, Wenying Zhu. "A General Method of Constructing the Reversible FullAdder", 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, 2010.
42. Mariam Zomorodi Moghadam, Keivan Navi. "Ultra-area-efficient reversible multiplier", Microelectronics Journal, 2012.
43. K. Walus, Mazur, G. Schulhof, G.A. Jullien. "Simple 4-Bit Processor Based On QuantumDot Cellular Automata (QCA)", 2005 IEEE International Conference on ApplicationSpecific Systems, Architec-ture Processors (ASAP'05), 2005.
44. Kamaraj, P. Marichamy. "Design and implementation of arithmetic and logic unit (ALU) using novel reversible gates in quantum cellular automata", 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017.
45. C.H. Bennett, “Logical Reversibility of Computation”, IBM J.Research and Development, pp. 525-532, November 1973.
46. Bennett, C.H., “Logical reversibility of computation”, IBM J. Research and Development, vol. 17: pp. 525-532, 1973.
47. 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.
48. B. Raghu Kanth1, B. Murali Krishna2, G. Phani Kumar3, J. Poornima4, K. Siva Rama Krishna “A Comparitive Study Of Reversible Logic Gates”International Journal of VLSI & Signal Processing Applications, Vol.2,Issue 1, Feb 2012,