A power-aware and high performance of High Speed & Low Power Efficient Arithmetic Logic Unit (ALU)

Anjali Sharma[1], Ravish Gupta[2]
M. Tech (VLSI) Scholar[1], Assistant Professor[2]
Department of Electronics & Communication Engineering
Dr. C.V. Raman University Kota, Bilaspur

Abstract: - By increasing the demand of enhancing the ability of processors to handle the more complex and challenging processors has resulted in the integration of a number of processor cores into one chip. Still the load of on the processor is not less in generic system. An Arithmetic Logic Unit (ALU) is the heart of all microprocessors. It is a combinational logic unit that performs its logical or arithmetic operations. ALU is getting smaller and more complex nowadays to enable the development of a more powerful but smaller computer. In this proposed paper an 8 bit ALU chip has been design to the benefits of all the computations are done in parallel and available simultaneously, so no clock resources are wasted. The MUX is then simply used to select the required output.

Keywords: - Arithmetic Logic Unit (ALU), MUX, FPGA, VHDL, Graphics Processing Units, Arithmetic Operations, Logical Operations. low-power ALU design, microprocessor, multiplexer, Logic design, Multiplexing, Adders, Circuits, Arithmetic, Logic gates.

I. INTRODUCTION

An arithmetic logic unit (ALU) is a digital circuit that performs arithmetic and logical operations. The ALU is a fundamental building block of the central processing unit (CPU) of a computer, and even the simplest microprocessors contain one for purposes such as maintaining timers. The processors found inside modern CPUs and graphics processing units (GPUs) accommodate very powerful and very complex ALUs; a single component may contain a number of ALUs.

Most of a processor's operations are performed by one or more ALUs. An ALU loads data from input registers, an external Control Unit then tells the ALU what operation to perform on that data, and then the ALU stores its result into an output register. The inputs to the ALU are the data to be operated on (called operands) and a code from the control unit indicating which operation to perform. Its output is the result of the computation.

The coding of the ALU has been done in VHDL.VHDL (Very High Speed Integrated Circuits Hardware Description Language) is a hardware description language used in electronic design automation to describe digital and mixed-signal systems such as field-programmable gate arrays and integrated circuits. It is commonly used to write text models that describe a logic circuit. Such a model is processed by a synthesis program, only if it is part of the logic design. A simulation program is used to test the logic design using simulation models to represent the logic circuits that interface to the design.

After the coding had been completed in VHDL, the synthesis part was done using Xilinx-ISE. Xilinx ISE is a software tool for synthesis and analysis of HDL designs, which enables the developers to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

ABOUT ALU

Microprocessors/Microcontrollers have a single module that performs arithmetic operations on integer values. This is because many of the different arithmetic and logical operations can be performed using similar (if not identical) hardware. The component that performs the arithmetic and logical operations is known as the **Arithmetic Logic Unit**, or ALU.

The ALU is one of the most important components in a microprocessor, and is typically the part of the processor that is designed first. Once the ALU is designed, the rest of the microprocessor is implemented to feed operands and control codes to the ALU.

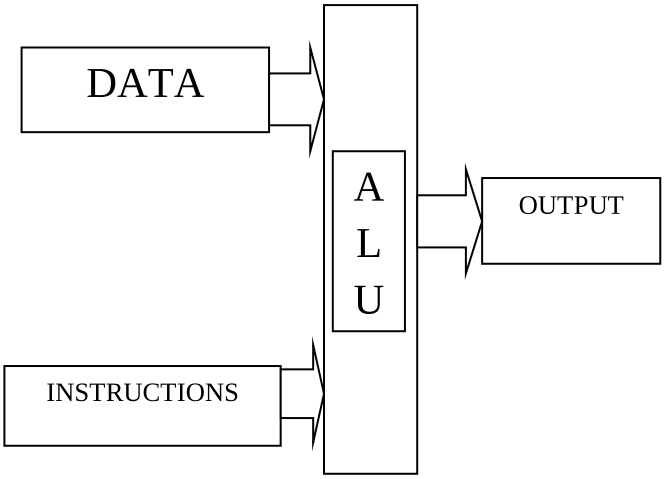


Figure 1.1: ALU block diagram

The arithmetic and logic unit (ALU) performs all arithmetic operations (addition, subtraction, multiplication, and division) and logic operations. Logic operations test various conditions encountered during processing and allow for different actions to be taken based on the results. The data required to perform the arithmetic and logical functions are inputs from the designated CPU registers and operands. The ALU relies on basic items to perform its operations. These include number systems, data routing circuits (adders/subtractions), timing, instructions, operands, and registers. **Figure 1.1** shows a representative block diagram of an ALU. An ALU loads data from input registers, an external Control Unit then tells the ALU what operation to perform on that data, and then the ALU stores its result into an output register. The Control Unit is responsible for moving the processed data between these registers, ALU and memory.

An ALU must process numbers using the same format as the rest of the digital circuit. The format of modern processors is almost always the two's complement binary number representation. Early computers used a wide variety of number systems, including ones' complement, two's complement sign-magnitude format, and even true decimal systems, with ten tubes per digit.

ALUs for each one of these numeric systems had different designs, and that influenced the current preference for two's complement, as this is the representation that makes it easier for the ALUs to calculate additions and subtraction.

FLOWCHART

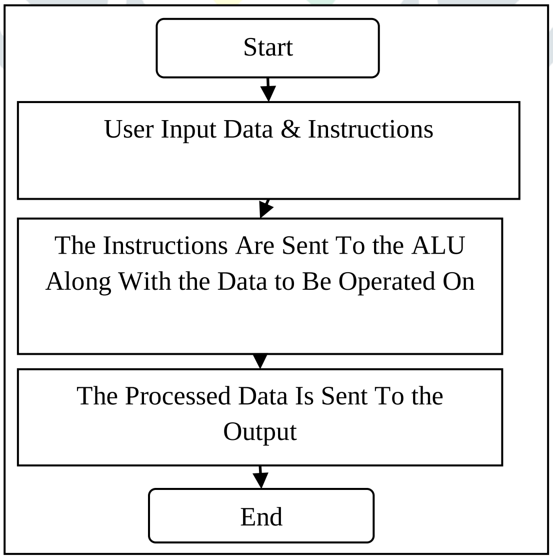

Start
User Input Data & Instructions
The Instructions Are Sent To the ALU Along With the Data to Be Operated On
The Processed Data Is Sent To the Output
End

Figure 1.2 – ALU Flowchart

Most ALUs can perform the following operations:

- Bitwise logic operations (AND, NOT, OR, XOR, NAND, NOR)
- Integer arithmetic operations (addition, subtraction, and sometimes multiplication and division, though this is more expensive).
- Other operations like greater than, equal to, exponential, modulus etc.

THE VARIOUS OPERATIONS

ARITHMETIC OPERATIONS	LOGICAL OPERATIONS
ADDITION	AND
SUBTRACTION	OR
MUTLIPLICATION	NOT
DIVISION	NAND
REMAINDER	NOR
MODULUS	XOR

Table-1: Arithmetic & Logical Operators

Project Flow

Project Flow A project is a collection mechanism for an HDL design under specification or test. Even though you don't have to use projects in ModelSim, they may ease interaction with the tool and are useful for organizing files and specifying simulation settings. The following diagram shows the basic steps for simulating a design within a ModelSim project.

As you can see, the flow is similar to the basic simulation flow. However, there are two important differences: • you do not have to create a working library in the project flow; it is done for you automatically. • Projects are persistent. In other words, they will open every time you invoke ModelSim unless you specifically close them.

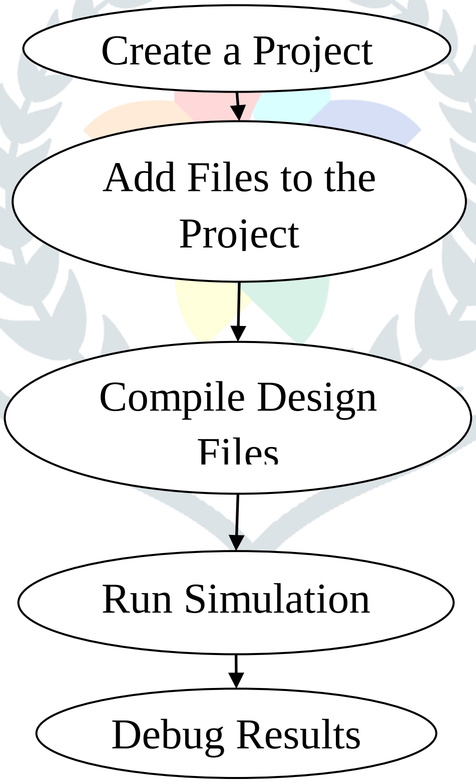


Figure 1.3 – Project Flowchart

II. Related Work

In the past, ALU and full adder circuits have been implemented for optimum area and delay, each with their distinct features that bring about optimum area and delay. Some of them have been briefly described below to give us an idea of earlier work and shed light on different optimization techniques. Past work related to multiple-input floating gate CMOS applications has been reviewed to give us an idea about its operation, design and simulation issues.

But et al have designed a low power 10-transistor full adder called Static Energy-Recovery Full Adder (SERF) using 10 transistors. A novel set of XOR and XNOR gates in combination with existing ones have been used. The XOR and XNOR circuits designed by them do not directly connect to power and ground lines, respectively. It uses four transistors for XOR and XNOR gates in CMOS.

Wang et al have shown an improved version of XOR and XNOR gates that make use of six transistors. In another design of CMOS 1-bit full adder cell, four transistor XOR and XNOR gates have been used. The cell offers higher speed and lesser power consumption than standard 1-bit full adder cell. Radhakrishnan has presented the design of low power CMOS full adder circuits using transmission function theory. This design uses six transistor CMOS XOR and XNOR gates. 16-bit, 2.4ns, 0.5nm CMOS ALU design, which consists of a logical and arithmetic unit (LAU), a magnitude comparator (CMP), an overflow detector (OVF) and zero flag detector (ZERO). The ALU employs a binary look-ahead carry (BLC) adder. All units in this design operate in parallel and high speed is achieved.

In the previous review 4-bit ALU design is based on multiple- input floating gate CMOS devices.

Harrison et al propose an analog floating memory element for on-chip storage of bias voltages. A floating gate technology has been used to eliminate off-chip biasing voltages in the existing systems by providing these voltages on-chip, with arrays of programmable floating- gate voltages. These arrays can be individually programmed by digital controls. In, short-wave ultraviolet light has been applied to floating-gate devices to adjust threshold voltages for optimal performance in a circuit.

Linear amplifiers with rail-to-rail operation with supply voltages less than 1V have been designed in floating gate CMOS. Floating gate MOS circuits have the inherent ability to adapt to the incoming and outgoing signals by continuously enabling various programming mechanisms. Based on this property, single floating gate FETs, that emulate computational and adaptive properties of biological synoptic elements, were developed. An example of continuously adapting floating- gate circuits is presented in. The auto zeroing floating gate amplifier (AFGA) uses tunnelling and pFET hot electron injection to adaptively set its dc operating point. No additional circuitry is required. Here a 4-bit ALU was designed which is operated at 3.0V, using multiple-input floating gate MOSFETs in standard 1.5μm CMOS process. The application of multi- input floating gate transistor, in designing the full adder, has enabled us to realize the full adder with fewer transistors, when compared to previous designs.

III. Problem Identification

As a fundamental part of the microprocessors, ALU performs computing operations and it is typically on the critical path. Therefore, the achievable operating frequency of the whole microprocessor is determined by the operating speed of ALU. At the same time, ALU is also one of the most active components in microprocessor, raising the power and thermal issues. Therefore, the lowest voltage supply to the circuit is needed to reduce power and reducing the propagation delay is required to enhance the speed of the ALU.

IV. PROPOSED APPROACH

Here we will discuss a new design that will save more power and area. The new MUX based ALU design Circuit is shown in below figure using MUX logic is used. This circuit saves power in such a way that even when Target device's clock is ON, the controlling device's clock is OFF and also when the target device's clock is OFF then also Controlling device's clock is OFF. This way we can save more power by avoiding unnecessary switching at clock net. This MUX based ALU design is on at positive-edge and off at negative-edge means clock power and dynamic power is saving at the time of negative-edge clock.

V. IMPLEMENTATION DETAIL

We have implemented different type of techniques and proposed technique to reduce power. All the techniques are performed at different technology with temperature, voltage and frequency variation and their Dynamic, static and total power has been computed, in this we are applying MUX based techniques on an 8-bit Arithmetic logical unit (ALU). The results are in tables shown below. All experiments are done on Xilinx14.1 EDA tool. Mentor Graphics Model SIM. For power calculation we are using XPOWER. Sparten-6 (45nm) FPGA platform is used for result and analysis.

Table 2: - ALU Function

Function Select				Output Equals	Function
S2 Cin	S1		S0		
0	0	0	0	F=A	Transfer A
0	0	0	1	F=A+1	Increment A
0	0	1	0	F=A+B	Addition
0	0	1	1	F=A+B+1	Add with carry
0	1	0	0	F=A-B-1	Subtraction with borrow
0	1	0	1	F=A-B	Subtraction
0	1	1	0	F=A-1	Decrement A
0	1	1	1	F=A	Transfer A
1	0	0	X	F=A|B	OR
1	0	1	X	F=A^B	X-OR
1	1	0	X	F=A&B	AND
1	1	1	X	F=~A	Complement A

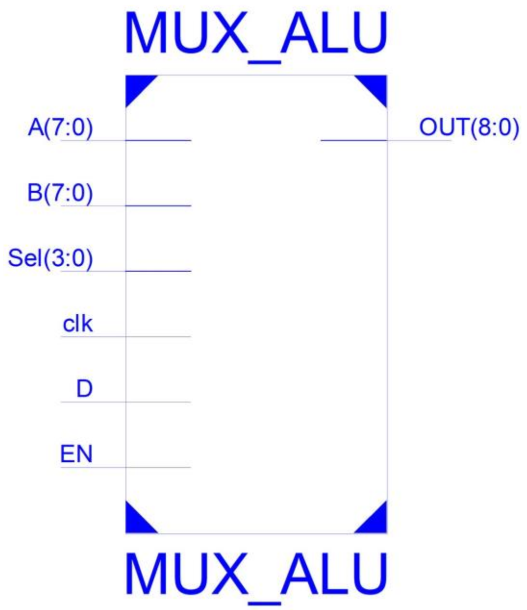


Figure 5.1: Top Level Schematic of ALU

VI. RESULTS

MUX_ALU Project Status (03/01/2019 - 03:09:54)			
Project File:	MUX_BASED.xise	**Parser Errors:**	No Errors
Module Name:	MUX_ALU	**Implementation State:**	Synthesized
Target Device:	xc6slx4-3tqg144	• **Errors:**	No Errors
Product Version:	ISE 14.1	• **Warnings:**	2 Warnings (2 new)
Design Goal:	Balanced	• **Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	
Environment:	System Settings	• **Final Timing Score:**	

Table 3:- Device Utilization Summary results

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	1	4800	0%
Number of Slice LUTs	35	2400	1%
Number of fully used LUT-FF pairs	1	35	2%
Number of bonded IOBs	32	102	31%
Number of BUFG/BUFGCTRLs	1	16	6%
Number of DSP48A1s	1	8	12%

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Sun 16. Mar 03:09:53 2014	0	2 Warnings (2 new)	2 Infos (2 new)
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

CIRCUIT SIMULATION & VERIFICATION

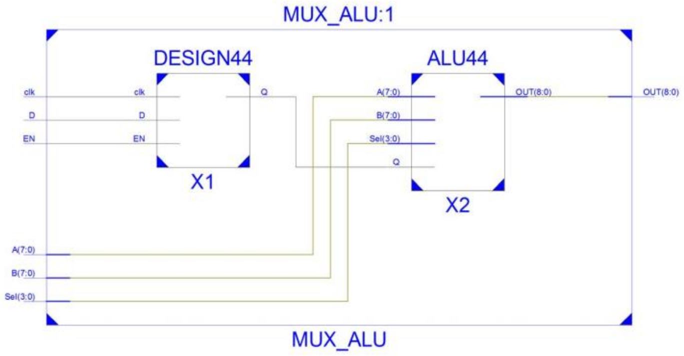


Figure 5.2: RTL Schematic of ALU

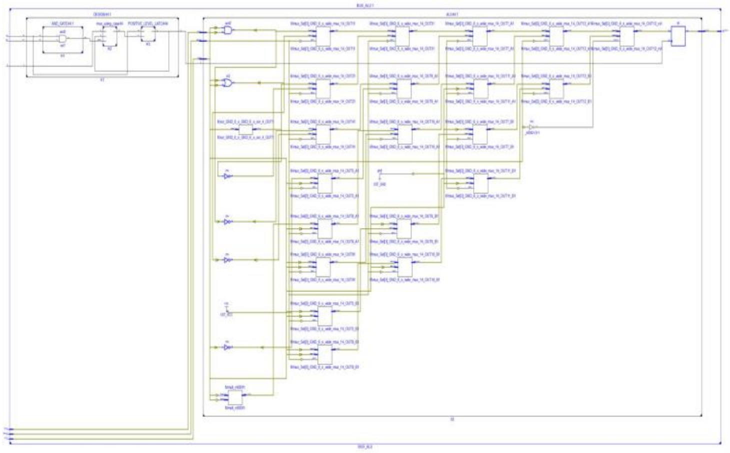


Figure 5.3: RTL Schematic of ALU

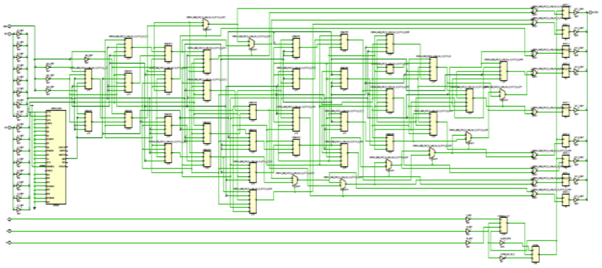
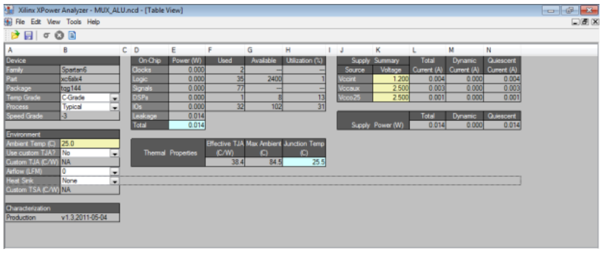


Figure 5.4: RTL Schematic of ALU


Xilinx XPower Analyzer - MUX_ALU.ncd - [Table View]
File Edit View Tools Help

Device	
Family	Spartan6
Part	xc6slx4
Package	tqg144
Temp Grade	C-Grade
Process	Typical
Speed Grade	-3

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0
Heat Sink	None
Custom TSA (C/W)	NA

Characterization	
Production	v1.3.2011-05-04

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	2	---	---
Logic	0.000	35	2400	1
Signals	0.000	77	---	---
DSPs	0.000	1	8	13
IOs	0.000	32	102	31
Leakage	0.014			
Total	0.014			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	38.4	84.5	25.5

Supply Summary Source	Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.200	0.004	0.000	0.004
Vccaux	2.500	0.003	0.000	0.003
Vcco25	2.500	0.001	0.000	0.001

	Total	Dynamic	Quiescent
Supply Power (W)	0.014	0.000	0.014

Figure 5.5: Xilinx XPOWER ANALYSIS

Table 4: Comparison of existing ALU & proposed ALU

Parameter Which has to be compared	Existing ALU	Proposed ALU
Gate count	22	10
Garbage outputs	12	8
Constant input	10	4
Quantum cost	53	29

Table 5: Comparison of Existing Control Unit & Proposed Control Unit

Parameter Which has to be compared	Existing control unit	Proposed control unit
Gate count	15	9
Garbage outputs	8	6
Constant input	8	3
Quantum cost	41	23

Table 6: ALU power consumption

S. No.	Design	No. of Transistors	Power (mW)
1	ALU With GDI Technique	96	3.994
2	ALU With 2T XOR Module	96	2.308
3	ALU With 3T NXOR Module	88	2.250
4	ALU With MUX	75	1.450

VII. CONCLUSION

In this paper, 8 -bit MUX based ALU design using at 45nm technology. The proposed MUX based ALU designs are verified using Xilinx & Spartan-6 FPGA software. Both proposed designs are analyzed and compared in terms of number of gate count, Ancilla Inputs, No. of Arithmetic and Logic Operations, Constant Inputs. In 8- bit ALU performs better as compare to existing design and consumes low power. 8s-bit ALU using Modified Gate input. Swing degradation can be overcome by modified gate diffusion input. This modified gate diffusion input logic style allows reducing power consumption, delay and area of digital circuits. It is therefore seen that the MUX based ALU design are much faster than the conventional method. The algorithms of MUX based ALU design are much more efficient than of conventional method. With MUX based technique, low power ALU design is possible and in next stage low power processor design is also possible with MUX based techniques.

VIII. FUTURE SCOPE

In this work, we implement our design on 45nm Spartan-6 FPGA. There is open scope to implement this design on 40nm Virtex-6 FPGA and 28nm Artix-7 FPGA to achieve energy efficiency in our design on higher level. This target design is 8-bit ALU, so there is scope to design this ALU in form of 16-bit ALU or 32-bit ALU or even higher 64-bit ALU for 64-bit architecture. There is lots of future objective in this ALU architecture. Future Objective of this Research area is followings:

1. To devolve architecture which is reduce the timing complexity issues.
2. To devolve architecture which is reduce the area related problems and which is suitable for all portable device.
3. To devolve architecture which is reduces power and consumption and work on low power.
4. To devolve architecture which is reduce energy consumption and which will make justice with all portable devices which are demands low energy consumption.
5. To devolve architecture which use some approximation concept and also some power reduction approach.

Applications

There are many applications of Arithmetic and Logical unit. These is used in many filed like multimedia, signal processing, general purpose. There is some applications which are demands ALU and those are:

1. MP3 Decoding/Encoding.
2. Video Decoding/Encoding.
3. Image Filtering.
4. Adaptive Difference Pulse Code Modulation.
5. Image and video processing.
6. General Purpose Processor.
7. Application Specific Processor.

So these are the different applications where we use Arithmetic logical unit.

Acknowledgments

The authors would like to thank the direct and implied help received at various stages while developing the project. It takes a great privilege to thank Prof. Ravi Prakash Dubey, Vice Chancellor, Dr. C.V. Raman University; Dr. Manish Upadhyay, Principal, Dr. C.V. Raman Institute of Science & Technology (CVRIST); Mr. Saurabh Mitra (HOD,ECE Department, CVRIST) for giving a great opportunity to carry out this research and heartfelt thanks to the mentor Mr. Ravish Gupta (Astt. Professor, ECE Department, CVRIST) whose generous support at all stages of this research helped us achieve the desired objectives.

REFERENCES

[1]. Saeid Moslehpour, Srikrishna Karatalapu" VLSI and SPICE modeling of ALU" Journal of Communication and Computer, ISSN 1548-7709, USA , Oct. 2009, Volume 6, No.10 (Serial No.59)

[2]. H.T. Bui, Y. Wang and Y. Jiang, "Design and analysis of low-power 10-transistor full adders using novel XOR-XNOR gates", IEEE Trans. On Circuits and Systems II: Analog and Digital Signal Processing, vol. 49, pp. 25 –30, Jan. 2002.

[3]. J.M. Wang, S.C. Fang and W.C. Fang, "New efficient designs for XOR and XNOR functions on transistor level", IEEE J. of Solid State Circuits, vol. 29, pp. 780-786, July 1994.

[4]. A.M. Shams and M.A. Bayoumi, "A novel high-performance CMOS 1-bit full-adder cell", IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 47, pp. 478 –481, May 2000. 105

[5]. D. Radhakrishnan, "Low-voltage low-power CMOS full adder", IEE Proceedings-Circuits, Devices and Systems, vol.148, pp. 19 - 24, Feb. 2001.

[6]. K. Suzuki, M. Yamashina, J. Goto, Y. Inoue, T. Koseki, Y. Horiuchi, T. Hamatake, K. Kumagai, T. Enomoto and H. Yamada, "A 2.4- ns, 16-bit, 0.5- ìm CMOS arithmetic logic unit for microprogrammable video signal processor LSIs", Proc. of the IEEE Custom Integrated Circuits Conference, vol. 9, pp.12.4.1 -12.4.4, May 1993.

[7]. A. Srivastava and C. Srinivasan, " ALU Design using Reconfigurable CMOS Logic", Proc. of the 45th IEEE 2002 Midwest Symposium on Circuits and Systems, vol.2, pp. 663-666, Aug. 2002.

[8]. R.R. Harrison, J.A. Bragg, P. Hasler, B.A. Minch and S.P. Deweerth, "A CMOS programmable analog memory-cell array using floating- gate circuits" IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 48, pp. 4 –11, Jan. 2001.

[9]. Y. Berg, T.S. Lande and O. Naess, "Programming floating-gate circuits with UV-activated conductances", IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 48, no. 1, pp. 12 –19, Jan. 2001.

[10]. P. Hasler, "Continuous-time feedback in floating-gate MOS circuits", IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vo l. 48, no. 1, pp. 56 –64, Jan. 2001.

[11]. J.M. Rabaey, "Digital Integrated Circuits", Pearson Education, pp. 559-596, 2005.

[12]. Suzuki, M. et al., "A 1.5-ns 32-b CMOS ALU in double pass-transistor logic" IEEE Journal of Solid-State Circuits, Vol 28, pp 1145-1151, 1993.

[13]. Arvind Rajput, Anil Goyal, "DESIGN AND COMPARISON OF LOW POWER & HIGH SPEED 4-BIT ALU", Proc. of 2nd National Conference on Challenges & Opportunities in Information Technology (COIT-2008), RIMT-IET, Mandi Gobindgarh. March 29, 2008.

[14]. A. Srivastava and C. Srinivasan, " ALU Design using Reconfigurable CMOS Logic", Proc. of the 45th IEEE 2002 Midwest Symposium on Circuits and Systems, vol.2, pp. 663-666, Aug. 2002.

[15]. J. Wakerly, "Digital Design", 3rd Edition, Prentice Hall, Upper Saddle River, 2000.M. Mano and C. Kime, "Logic and Computer Design Fundamentals", 2nd Edition, Prentice Hall, Upper Saddle River, NJ, 2001

[16]. D. Radhakrishnan, "Low-voltage low-power CMOS full adder", IEE Proceedings- Circuits, Devices and Systems, vol.148, pp. 19 - 24, Feb. 2001.

[17]. N. Weste and K. Eshragian, Principles of CMOS VLSI Design: A Systems Perspective, Pearson/ Addison - Wesley Publishers, 2005.

[18]. Chandrakasan, A., and Brodersen, Low Power Digital Design, Kluwer Academic Publishers, R., 1995

[19]. J.P. Hayes, "Computer Architecture and Organization, Computer Organization", McGraw Hill, pp. 230-250, 1988.

[20]. V.G. Mobdzija, "A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach", IEEE Transactions on Computers, vol. 45, No, 3, pp. 294-305, 1996.

[21]. Suzuki, M. et al., "A 1.5-ns 32-b CMOS ALU in double pass-transistor logic" IEEE Journal of Solid-State Circuits, Vol 28, pp 1145-1151, 1993.

[22]. Preeti Takhar ,Priyanka Rajpal , Rahul Borthakur , Sakshi Agarwal" DESIGNING OF 8 BIT ALU AND IMPLEMENTING ON XILINX VERTEX 4 FPGA", April 2011

[23]. T. Esther Rani, M.A. Rani, R. Rao, "AREA optimized low power arithmetic and logic unit", *IEEE International Conference on Electronics Computer Technology*, pp. 224-228, April 2011.

[24]. Gangadhar Reddy Ramireddy, "A Novel Power-Aware and High Performance Full Adder Cell for Ultra low Power Design", *IEEE International Conference on Circuit Power and Computing Technologies*, pp. 1121-1126, 2014.

[25]. JVR Ravindra, Gangadhar Reddy Ramireddy, Harikrishna Kamatham, "Design of Ultra Low Power Full Adder using Modified Branch Based Logic Style", *IEEE European Modelling Symposium*, pp. 691-696, 2013.

[26]. Rajesh Parihar, Nidhi Tiwari, Aditya Mandloi, Binod Kumar, "An Implementation of 1-Bit Low Power Full Adder Based on Multiplexer and Pass Transistor Logic", *IEEE International Conference on Information Communication and Embedded System*, pp. 101-103, 2014.

[27]. L. Dhulipalla, A. Deepak, "Design and implementation Of 4-bit ALU using FINFETS for nano scaletechnology", *IEEE International Conference on Nanoscience Engineering and Technology*, pp. 190-195, November 2011.

[28]. A. Srivastava, C. Srinivasan, "ALU Design Using Reconfigurable CMOS Logic", *IEEE 45th midwest symposium on circuit and system*, vol. 2, pp. 663-666, August 2002.

[29]. Ravi Tiwari, Khemraj Deshmukh, "Design and analysis of low power 11-transistor full adder", *IJAREEIE*, vol. 3, no. 6, pp. 10301-10307, June 2014.

[30]. Pooja Vaishnav, Mr. Vishal Moyal, "Performance Analysis Of 8-Bit ALU For Power In 32 Nm Scale", *IJERT*, vol. 1, no. 8, pp. 1-3, October 2012.