Implementation of Power Efficient Novel Multiplexer Based Arithmetic Logic Unit

Sakshi Rajput

Asst. Professor (ECE), Maharaja Surajmal Institute of Technology
Affiliated to GGSIPU, Delhi, India

Abstract— **ALU (Arithmetic logic unit) is a critical component of a microprocessor and is the core component of central processing unit.**
Furthermore, it is the heart of the instruction execution portion of every computer [1] .ALUs comprise the combinational logic that implements logic operations, such as AND, OR, and arithmetic operations, such as ADD and SUBTRACT. In this an 8-bit ALU using 16:1 multiplexer is designed that is more efficient, less power consuming, less surface area and faster as compared to the conventional ALU.

Keywords—*ALU (Arithmetic Logic Unit), Multiplexer, XOR, AND, verilog.*

1. Introduction

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip. These signal processing applications not only demand great computation capacity but also consume considerable amount of energy. While performance and Area remain to be the two major design tools, power consumption has become a critical concern in today's VLSI system design [2].

A novel design of ALU using Multiplexers for all the purpose of addition, subtraction, comparison, shifting etc. is designed in this work. An algorithm is devised in order to felicitate the writing of codes in Verilog. The Verilog coding synthesis issues play a vital role in the speed-area optimality because RTL schematic depends heavily on how it has coded in Verilog [3].

A novel MUX (Multiplexer) - ALU has better performance with the less area and power optimization on the bases of macro cells which is the basic building block of any electronic device.

2. Conventional Arithmetic Unit

In the conventional 8-bit ALU, there are two inputs a & b of eight bits each, and a signal 's' is used to select the operation which is to be performed, and an output register that stores the data obtained after the required operation. The execution happens serially according to the select line and the output is displayed accordingly. [4]

The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing applications. The key arithmetic operations in such applications are multiplication, addition, division and subtraction.

An arithmetic and 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 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.

The arithmetic and logic unit performs both the arithmetic and the logic operations on the data supplied to it. Although many functions can be performed by ALU, the basic arithmetic operations – addition, subtraction, multiplication, and division – continue to be "bread and butter" operations. The other basic operations like shifting, logical multiplication and logical addition are also performed by ALU.

A critical component of the microprocessor and the core component of central processing unit is ALU. ALU comprises the combinational logic that implements logic operations such as AND and OR, and arithmetic operations such as Addition, Subtraction, and Multiplication [7].

Figure 1. displays the general block diagram of an 8-bit ALU that is connected with the output LEDs and seven segment displays through a decoder and a display switch. This is the implementation of ALU on the FPGA kit.

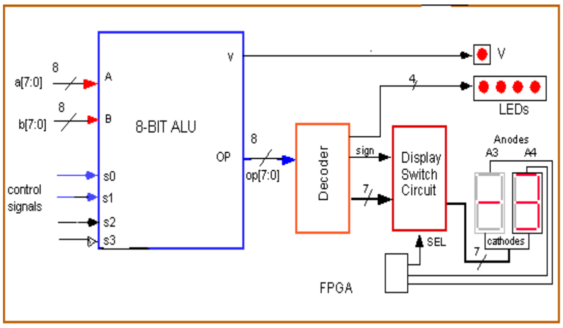


Figure 1: Block diagram of 8-bit ALU

The functionalities of the ALU include AND, OR, addition, subtraction, less than, NOR, XOR, left shift and right shift operations as shown in the flow diagram in figure 2.

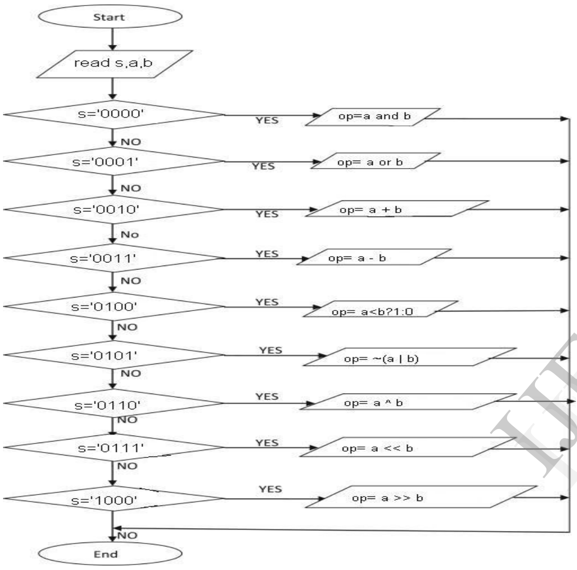


Figure 2: Flow diagram of 8-bit ALU[7]

3. Novel MUX based ALU

In this Arithmetic Logic Unit, a design of an 8 bit ALU in Verilog, using multiplexers is proposed. And it has a Multiplexer with 9 inputs for each operation like addition, subtraction etc. and one output.

In this technique the macro cell usage, Pterms (product terms or output of AND gates), functional block inputs, clock delays parameter values are comparatively lesser than the conventional ALU.

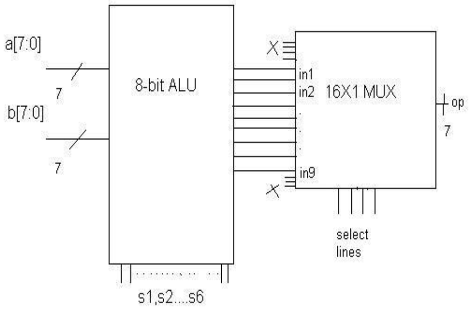


Figure 3: Block diagram of novel 8-bit ALU

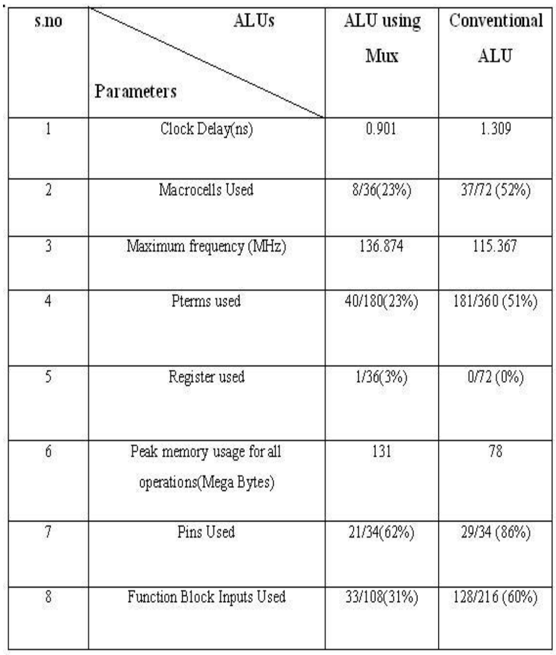
In figure 3, 8-bit ALU outputs are connected to the 9 inputs of the 16x1 Multiplexer. The remaining seven pins of the MUX have don't care values. The 16x1 multiplexer the n selects the appropriate input through select lines to give the desired output.

4. Comparison of Conventional and Novel ALU

Table 1 shows the comparison of the conventional ALU and the ALU using MUX based of the different characteristics like speed of operation, efficiency and power consumption etc.

In conventional ALU maximum frequency is 115.367MHz whereas in case of novel ALU it is 136.874MHz. Number of Pterms is reduced from 181/360 to 40/180.

Table 1: Comparison Table



s.no	ALUs / Parameters	ALU using Mux	Conventional ALU
1	Clock Delay(ns)	0.901	1.309
2	Macrocells Used	8/36(23%)	37/72 (52%)
3	Maximum frequency (MHz)	136.874	115.367
4	Pterms used	40/180(23%)	181/360 (51%)
5	Register used	1/36(3%)	0/72 (0%)
6	Peak memory usage for all operations(Mega Bytes)	131	78
7	Pins Used	21/34(62%)	29/34 (86%)
8	Function Block Inputs Used	33/108(31%)	128/216 (60%)

On the basis of the collected data and simulation result it has been analyzed that proposed MUX-ALU accepts the inputs and corresponding signal for the operation. Multiplexer uses the registers to stores the intermediate result which is further can be use for selecting the corresponding output signal on the same data with having comparatively less delay than the conventional ALU while in case of conventional ALU it calculates the result each time for the selected operation and then assign it to the output signal.

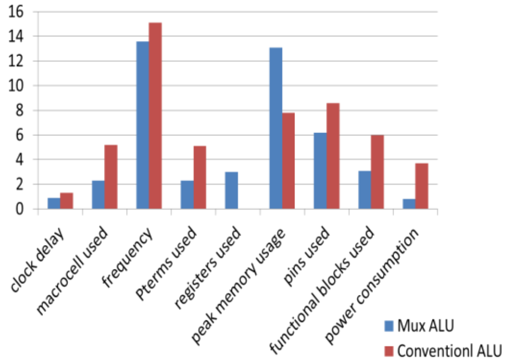


Figure 4: Comparison chart for two ALU

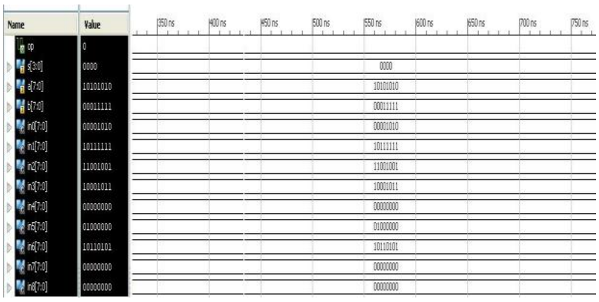
As can be seen in the comparison table 1, that the proposed ALU having less delay with less surface area on the base of macro cells used and it can also be easily visualized in the corresponding bar graph in fig 4 which shows that clock delay is less in case of proposed ALU and also the macro cells which is the bases of the electronic circuitry which completely defines the comparison in b/w the two ALUs. Also the other factors or parameters which greatly affect the results have been shown.

5. Implementation Results

The conventional ALU and modified MUX based ALU has been designed in Verilog HDL synthesized in Xilinx ISE design suit 12.1b and the design is simulated on ModelSim PE 10.1b tool. [3]

Simulation is the imitation of the operation of a real-world process or system over time. The act of simulating something first requires that a model be developed; this model represents the key characteristics of the selected physical or abstract system or process.

In the simulation result shown in Figure 5 multiplexer uses the registers to stores the intermediate result which is further can be use for selecting the corresponding output signal on the same data with having comparatively less delay than the conventional ALU while in case of conventional ALU it calculates the result each time for the selected operation and then assign it to the output signal.



Name	Value
op	0
s[3:0]	0000
a[7:0]	10101010
b[7:0]	00011111
in0[7:0]	00001010
in1[7:0]	10111111
in2[7:0]	11001001
in3[7:0]	10001011
in4[7:0]	00000000
in5[7:0]	01000000
in6[7:0]	10110101
in7[7:0]	00000000
in8[7:0]	00000000

Figure 5: Simulation result of 8-bit Novel ALU

The analysis shows the great affecting results on power consumption in low power mode in comparison data table given in table 3, which is a huge difference in macro cells usage.

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 Mr. S.P. Singh, Chairman, Maharaja Surajmal Institute of Technology(MSIT); Dr. Navin Kapur ,Director, MSIT; Dr. V. Kumar (HOD,ECE Deptt. ,MSIT) for giving a great opportunity to carry out this research and heartfelt thanks to the mentor Ms. Sakshi Rajput (Astt. Professor, ECE Deptt, MSIT) whose generous support at all stages of this research helped us achieve the desired objectives.

References

[1] Parth Sarthy Mohanty, “Design and implementation of low power fast multipliers”, thesis report NIT Rourkela, 2009.

[2] N. Nityalakshmi, “VLSI implementation of low power multiplier”, (ICCCE 2012), 12 & 13 April, 2012

[3] B. Ram Kumar, H.M. Kittur, and P. M. Kannan, “ASIC implementation of modified faster carry save adder,” *Eur. J. Sci. Res., vol. 42, no. 1, pp. 53–58, 2010.*

[4] K.S.Rao, M.Reddy, A.V.Babu and P.Srinivasulu, “Energy and area efficient carry select adder,” *IJERA, vol. 34, issue 2, pp. 436-440, Mar-Apr 2012.*