Implementation of ALU on FPGA

Urvish Lakadiwala[1], Soham Hirapara[2], Raj Ramani[3], Niket chaudhary[4]

[1]Student of B.E., Electronics & Communication Engineering, SSASIT, Gujarat, India
[2] Student of B.E., Electronics & Communication Engineering, SSASIT, Gujarat, India
[3] Student of B.E., Electronics & Communication Engineering, SSASIT, Gujarat, India
[4] Student of B.E., Electronics & Communication Engineering, SSASIT, Gujarat, India

---***---

Abstract – *We have implement ALU on the FPGA by use of Xilinx Ise. FPGA (Field Programmable Gate Array) is configured by designer in field after manufacturing [1]. To program the ALU we use Verilog hardware descriptive language. For that purpose according to our need we have to design it. We have to design ALU with three flag, one output line, Reset line, two input and operation which we have to perform.*

Key Words: FPGA, Verilog, ALU

I. INTRODUCTION

An **arithmetic logic unit (ALU)** is used to perform arithmetic and logic operations. ALU is computational units used everywhere in field. It is core part of CPU and main part of the microcontroller. The basic controller have adder, subtractor, multiplier and divider on it. We implement two ALU.one is 1-bit ALU and another is 8-bit ALU.

A. SINGLE BIT(1 BIT) ALU

In this ALU we perform 6 different operation weather those operations are logical (AND operation, OR operation, NOT operation, NAND operation, NOR operation) or arithmetic (ADD operation, SUBSTRACTION operation, MULTIPLICATION operation, MODULO operation, INCREMENT operation, DECREMENT operation). Now in ALU we use 4 inputs. In this ALU number of related logic is 6, and unrelated logics are 0. Number of flip-flop (combination of several gates like OR, NOT, AND) used are 3, number of IOB (INPUT/OUTPUT BLOCK) flip-flops are 2. In this one bit ALU number of fan-out (Fan-out is a term that define the maximum number of digital input that the output of a single logic gate can feed) of Non-clock nets are 4.00.

B. 8 BIT ALU

In this ALU we perform 64 different operation weather those operations are logical (AND operation, OR operation, NOT operation, NAND operation, NOR operation) or arithmetic (ADD operation, SUBSTRACTION operation, MULTIPLICATION operation, MODULO operation, INCREMENT operation, DECREMENT operation). Now in ALU we use 4 inputs in this ALU number of related logic is 64, and unrelated logics are 0. Number of flip-flop (combination of several gates like OR, NOT, AND) used are 128, number of IOB (INPUT/OUTPUT BLOCK) flip-flops are 39. In this one bit ALU number of fan-out (Fan-out is a term that define the maximum number of digital input that the output of a single logic gate can feed) of Non-clock nets are 2.90.

II. DESIGN OF ALU

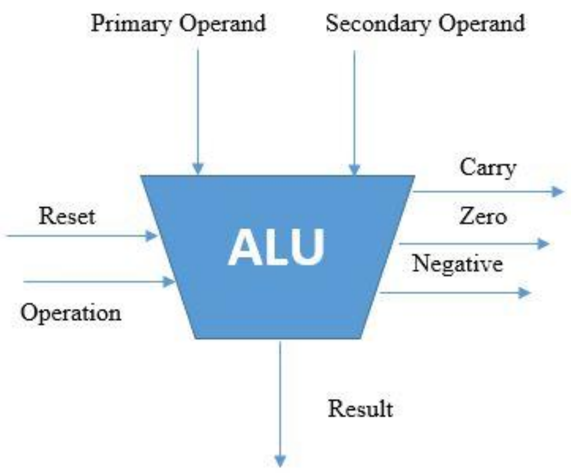


Figure 1:- Block diagram of ALU

For design of ALU we have two operand. I.e. primary operand and secondary operand. All operations are perform between these two operands. From this operation whether generate carry, zero or negative flag by ALU. Reset clear the ALU.

A. DESIGN 1 BIT ALU

For design of single bit ALU we have to define two single bit inputs for the operation. Which is operand A and operand B. These two are single bit operand. In operation whatever operation is occur the output is store at Result which is 1 bit. If there is a carry so carry flag is generated.

B. DESIGN OF 8 BIT ALU

In 8 bit ALU two 8 bit operand A (7:0) and B (7:0) and result store in Result (15:0). Here Result is 16 bit long because of multiplication operation which takes more than 8 bits for an output bus [2]. Flags (2:0) so 3 flag status are stored that. This way carry, zero and negative flags are generated.

III. OPCODE OF ALU

Several operation are done in ALU. We have to implement microcontroller so for that ALU get opcode for operation. According to opcode ALU doing operation. There are 4 bit opcode we have taken, so total 16 instruction should be created. Here we take total 13 instruction and remaining free for future update. Move is simple pass-through [1] instruction in which one operand is copied to result. Which is simple register transfers, remaining immediate and memory operation also perform in these instructions. All opcode and operation are shown in Table1. [3]

Table 1:- Opcode Table

Opcode	Instruction	Operation
0000	MOV	Result = A
0001	ADD	Result = A+B
0010	SUB	Result = A-B
0011	MUL	Result = A*B
0100	AND	Result = A&&B
0101	OR	Result = A||B
0110	LSR	Result = A>>x
0111	LSL	Result = A<<x
1000	ASR	Result = A>>x
1001	ASL	Result = A<<x
1010	COMP	Result = (~A)
1011	NEG	Result = 2's complement(A)
1100	HALT	Stop Execution

IV. IMPLEMENTATION OF ALU ON FPGA

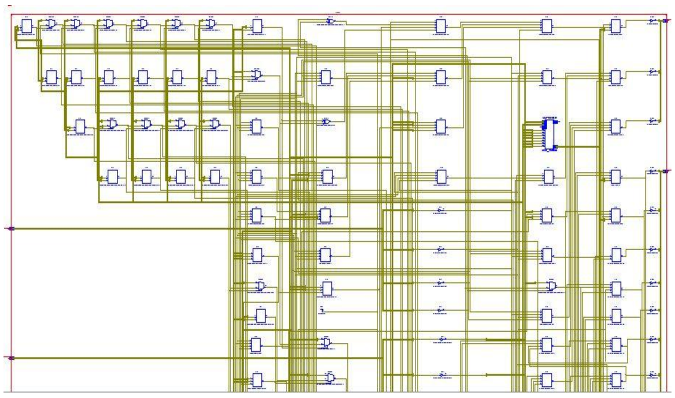


Figure 2:- Technology View of ALU

Verilog coding of ALU is compiled in Xilinx ise. First check 1 bit ALU. If this succeed than same thing done by 8 bit ALU. In behavioral model we check our logic with same output which we have define for that. If that code working properly with no errors than download that on FPGA. Before that you have to do post synthesis in which real time behavior of device is shown which is indicated in figure 3. There is a technology view of ALU indicate how many gates, multiplexers and flip flops are used to builds one ALU, Which is shown in figure 2.

V. SIMULATION

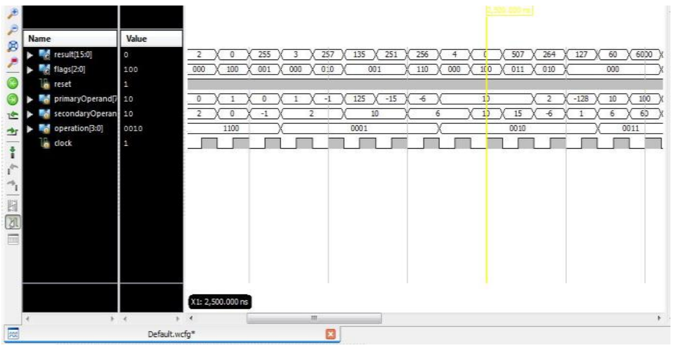
Example: A=10, B=10

Opcode=0010

Result=A-B

Result=0

Which is mention in the diming diagram of ALU which is shown in figure 3.



Name	Value
result[15:0]	0
flags[2:0]	100
reset	1
primaryOperand[7	10
secondaryOperan	10
operation[3:0]	0010
clock	1

Figure 3:- Timing Diagram of ALU

VI. CONCLUSION

This project help us to know how to implement ALU on the build microcontroller on FPGA with a Xilinx ise. After doing programmed ALU we have to check and result match with correct one, which indicate that ALU is successfully built.

ACKNOWLEDGEMENT

I especially thanks to Prof. Bhavesh Asodariya and Prof. Nirmal Bhatt of our collage for giving me guide about the design, without help of this guidance we cannot complete this type of work.

REFERENCES

[1] Margaret Rouse, "Definition of FPGA" Sep. 2005 http://whatis.techtarget.com/definition/field-programmable-gate-array-FPGA.

[2] Mr.Steven Bell, "Microprocessor Final Design", Oklahoma Christian University , Dec - 2010.

[3] Mr. Samir Palnitkar, "Verilog HDL," Pearson Publication, 2013.

BIOGRAPHIES

Raj Ramani
Study in B.E. , Electronics & Communication Engineering , Shree Swami Atmanand Saraswati Institute of Technology

Soham Hirapara
Study in B.E. , Electronics & Communication Engineering , Shree Swami Atmanand Saraswati Institute of Technology

Urvish Lakadiwala
Study in B.E. , Electronics & Communication Engineering , Shree Swami Atmanand Saraswati Institute of Technology

Niket chaudhary
Study in B.E. , Electronics & Communication Engineering , Shree Swami Atmanand Saraswati Institute of Technology