Project on Arithmetic Logic Unit (ALU)

Submitted for requirements for the practical coursework

of

M Tech (VDN)

by

Shivam Vaish (2302102034)

Akash Pandey (2302102035)

Sushil Chaudhary (2302102027)

Shivam Chaudhary (2302102032)

DISCIPLINE OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

13th Oct

INDIAN INSTITUTE OF TECHNOLOGY INDORE

I hereby certify that the work which is being presented in the report entitled **Project on Arithmetic Logic Unit(ALU)** submitted for requirements of the practical coursework of **M Tech (VDN)** and submitted in the Discipline of Electrical Engineering**, Indian Institute of Technology Indore**, is an authentic record of our group work carried out during the time period from July2023 to Oct 2023 under the supervision of Dr. Santosh Kumar Vishwakarma. The matter presented in this report has been cited properly wherever necessary.

(Shivam Vaish)
(Akash Pandey)
(Sushil Chaudhary)
(Shivam Chaudhary)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

(Dr. Santosh Kumar Vishvakarma)

ACKNOWLEDGEMENTS

First of all, I would like to thank my Supervisor Dr. Santosh Kumar Vishwakarma whose constant motivation, guidance and suggestion helped me to accomplish the task. I would also like to thank him for providing Vivado software resources associated with Nanoscale Devices, VLSI Circuit and System Design (NSDCS) Lab. I Would Like To Thank NSDCS Phd Scholar Mr. Mukul and all other Phd scholars for guidance and support and fellow MTech VDN students for constant motivation towards achieving the task.

SYNOPSIS

Arithmetic Logic Unit (ALU), which is capable of performing logical operations (e.g. AND, OR, Ex-OR, Invert etc.) and arithmetic operations (e.g. Addition, Subtraction etc.). The control unit supplies the data required by the ALU from memory, or from input devices, and directs the ALU to perform a specific operation based on the instruction fetched from the memory. ALU is the “calculator” portion of the computer.

In this project we’ve implemented 16 arithmetic functions using Verilog programming. There are two inputs A & B of 1 byte size and one output Y also of size 1 byte. Function to be implemented is selected using case statement where we ‘ve allotted a unique id for each function. Simulation for the different inputs has been performed to test the functionality of the ALU.

Further, synthesis of the ALU design has also been performed in which various reports such as timing, power, utilization and area reports are made.

TABLE OF CONTENTS

List of Figures

List of tables

TABLE OF CONTENTS

1.INTRODUNCTION

An arithmetic logic unit (ALU) is a key component of a computer's central processor unit. The ALU performs all arithmetic and logic operations that must be performed on instruction words. The ALU is split into two parts in some microprocessor architectures: the AU and the LU.

It can execute all arithmetic and logic operations, including Boolean comparisons, such as subtraction, addition, and shifting (XOR, OR, AND, and NOT operations). Binary numbers can also perform bitwise and mathematical operations. AU (arithmetic unit) and LU (logic unit) are two types of arithmetic logic units. The ALU's operands and code instruct it on which operations to perform based on the incoming data. When the ALU has finished processing the data, it sends the result to the computer memory.

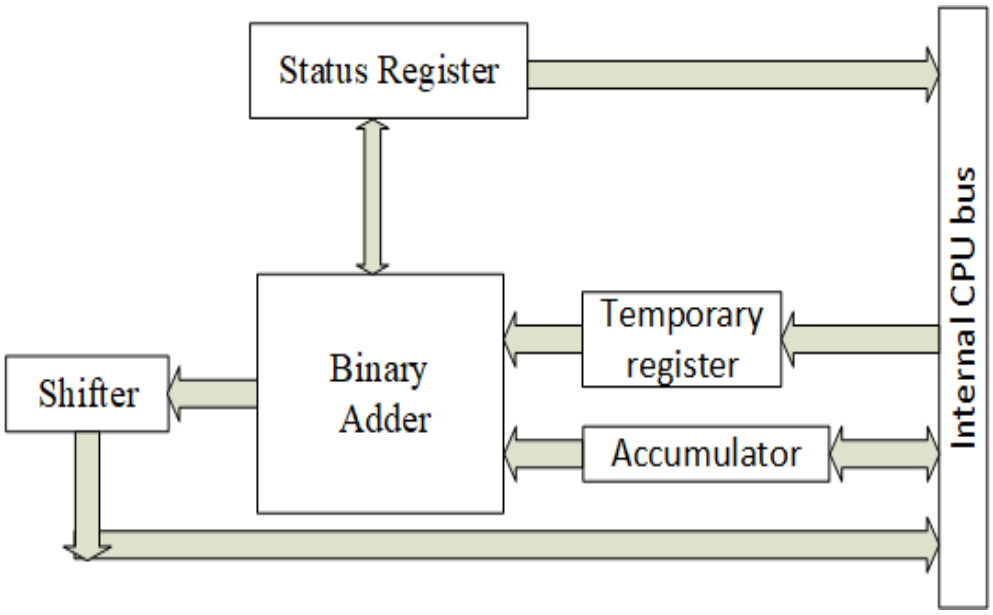

Fig1: ALU block diagram

2. ARITHMETIC LOGIC UNIT (ALU)- THEORY

An arithmetic logic unit(ALU) is a major component of the central processing unit of a computer system. It does all processes related to arithmetic and logic operations that need to be done on instruction words. In some microprocessor architectures, the ALU can also be divided into the arithmetic unit (AU) and the logic unit (LU).

Arithmetic Logic Unit (ALU), which is capable of performing logical operations (e.g. AND, OR, Ex-OR, Invert etc.) and arithmetic operations (e.g. Addition, Subtraction etc.). The control unit supplies the data required by the ALU from memory, or from input devices, and directs the ALU to perform a specific operation based on the instruction fetched from the memory. ALU is the “calculator” portion of the computer.

An ALU can be designed by engineers to calculate many different operations. When the operations become more and more complex, then the ALU will also become more and more expensive and also takes up more space in the CPU and dissipates more heat. That is why engineers make the ALU powerful enough to ensure that the CPU is also powerful and fast, but not so complex as to become prohibitive in terms of cost and other disadvantages.

The arithmetic logic unit is that part of the CPU that handles all the calculations the CPU may need. Most of these operations are logical in nature. Depending on how the ALU is designed, it can make the CPU more powerful, but it also consumes more energy and creates more heat. Therefore, there must be a balance between how powerful and complex the ALU is and how expensive the whole unit becomes. This is why faster CPUs are more expensive, consume more power and dissipate more heat.

Different operation as carried out by ALU can be categorized as follows –

logical operations − These include operations like AND, OR, NOT, XOR, NOR, NAND, etc.

Bit-Shifting Operations − This pertains to shifting the positions of the bits by a certain number of places either towards the right or left, which is considered a multiplication or division operation.

Arithmetic operations − This refers to bit addition and subtraction. Multiplication and subtraction can also be done by repetitive additions and subtractions respectively.

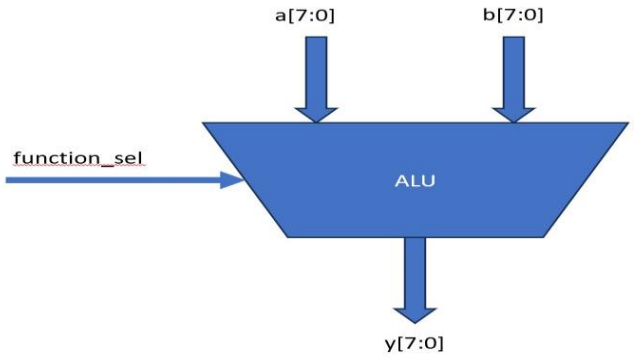

Fig2: ALU symbol

Processor Ops

1. IDLE: Performs no operation
2. ADD: y= a+b
3. SUB: y= a-b
4. MUL: y=a*b
5. DIV: y=a/b
6. COMPARE: y=(a>=<b)
7. CLEAR: y=0
8. READ: Read value stored in y
9. WRITE: y= 7
10. AND: y=a&b
11. OR: y=a|b
12. NOT: y=~a
13. NOR: y=(~(a|b))
14. XOR: y= a^b
15. XNOR: y= (~(a^b))
16. SET: y= 1
17. SWAP: Swap values of a and b
18. INC: y= a+4
19. DEC: y= b-4
20. REVERSE: Inverse bits of y
21. LEFT SHIFT: Left shift 2 bits of y
22. RIGHT SHIFT:Right shift 2 bits of y

3. SCHEMATIC AND SIMULATION:

3.1 ALU SCHEMATIC:

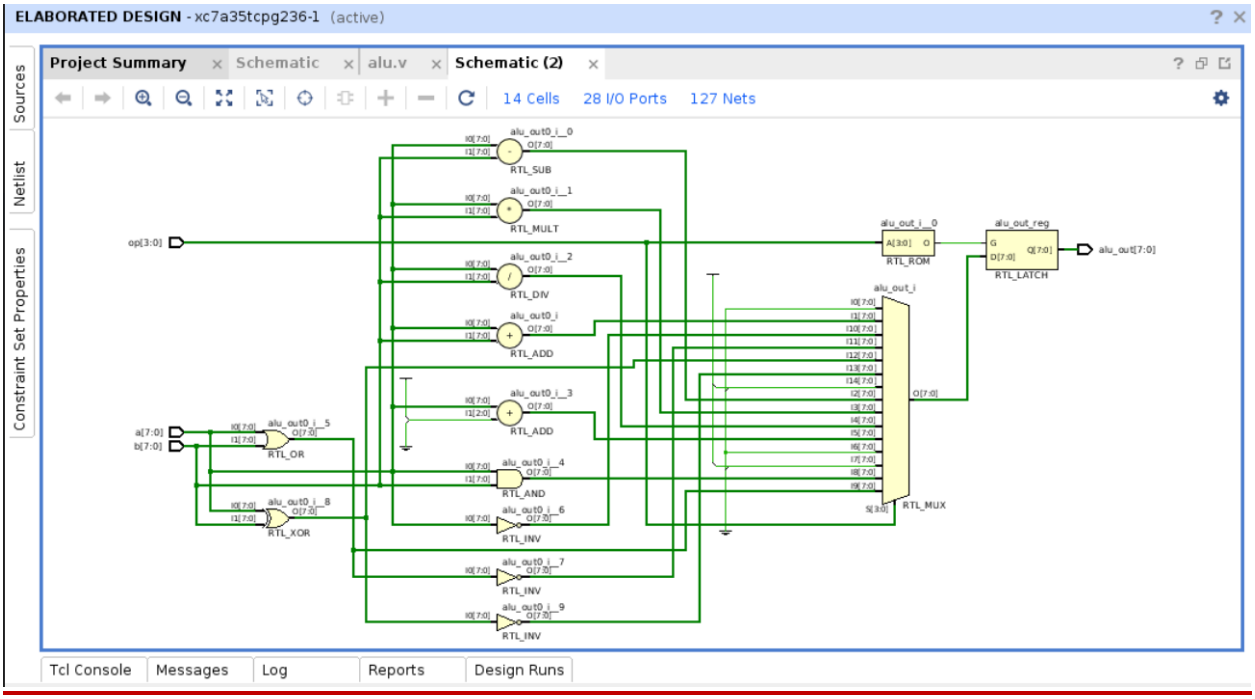

Fig3: ALU schematic

3.2 ALU SIMULATION:

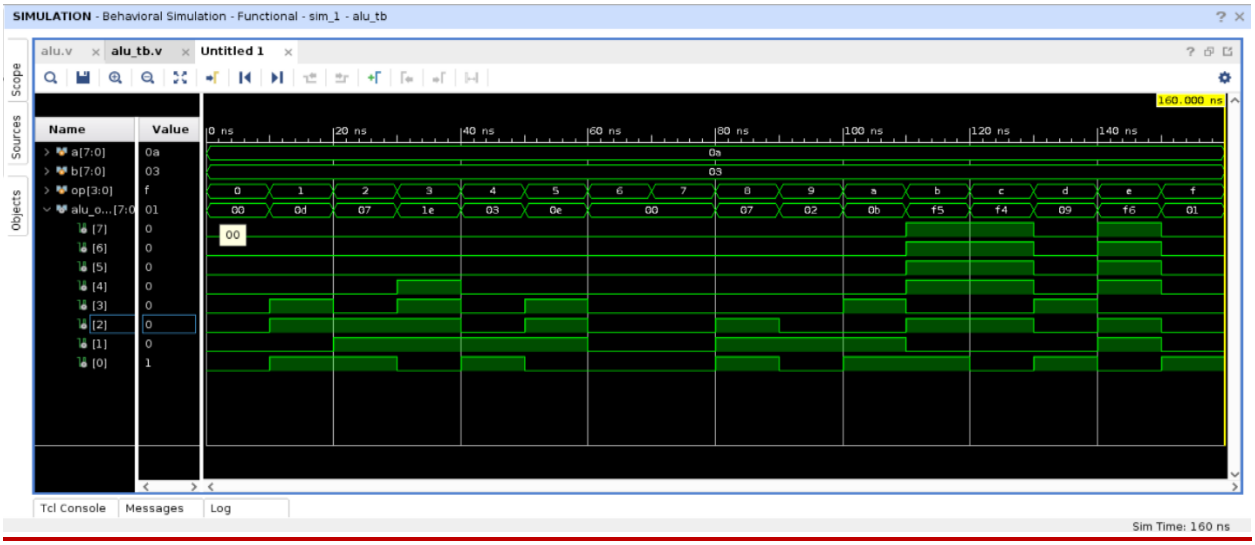

Fig4:- ALU simulation for different logic operations

Result: Verified all the logic function performed by the ALU in above simulation. All the testcases are provided in table below.

Inputs		Arithmetic logic operation	Output
A[7:0]	**B[7:0]**		
00001010	00000011	Idle	00
00001010	00000011	Add	0d
00001010	00000011	Subtraction	07
00001010	00000011	Multiplication	1e
00001010	00000011	Division	03
00001010	00000011	Increment	0e
00001010	00000011	Clear	00
00001010	00000011	Read	00
00001010	00000011	Write	07
00001010	00000011	And	02
00001010	00000011	Or	0b
00001010	00000011	Not	f5
00001010	00000011	Nor	f4
00001010	00000011	Xor	09
00001010	00000011	Xnor	f6
00001010	00000011	Set	01

Table1: Testcases description

4. SYNTHESIS REPORTS:

4.1 POWER REPORT

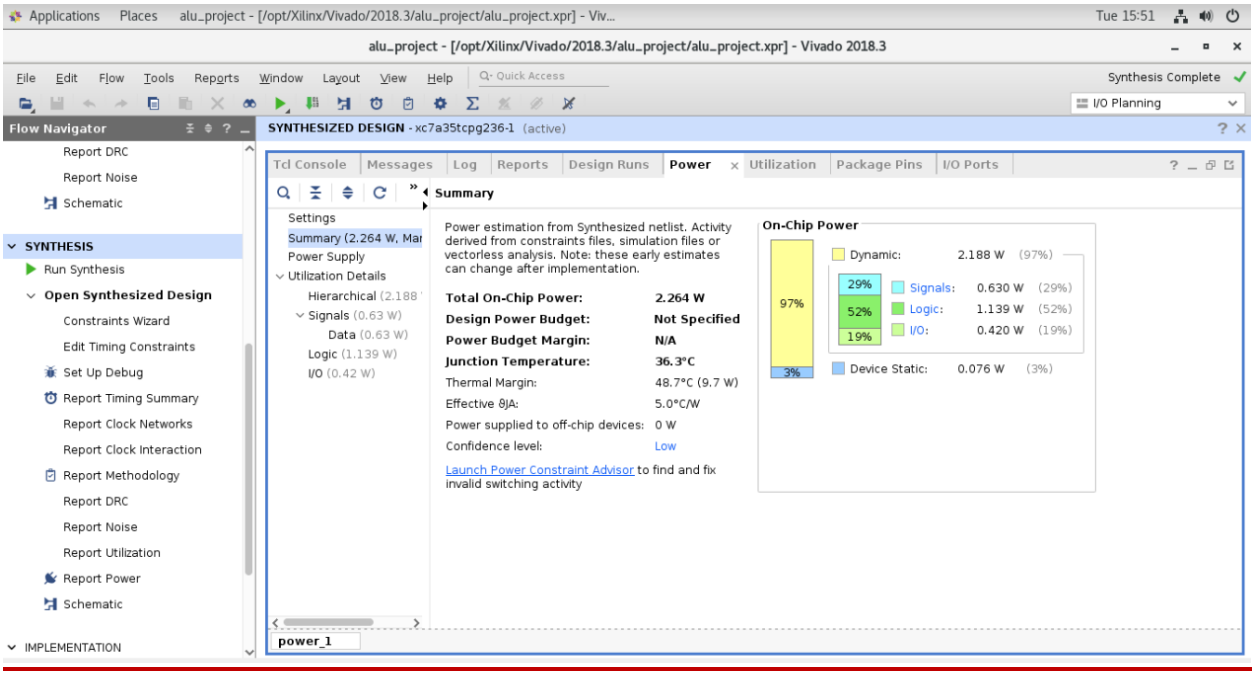

Fig5: Power report

4.2 TIMING REPORT

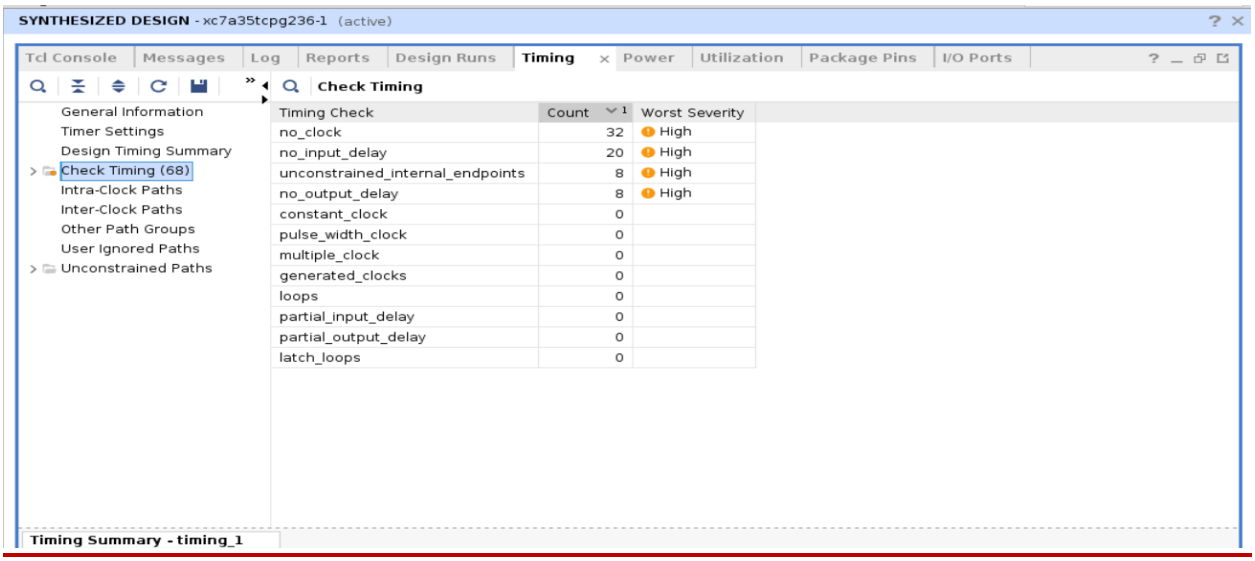

Fig6: Timing report

4.3 UTILIZATION REPORT

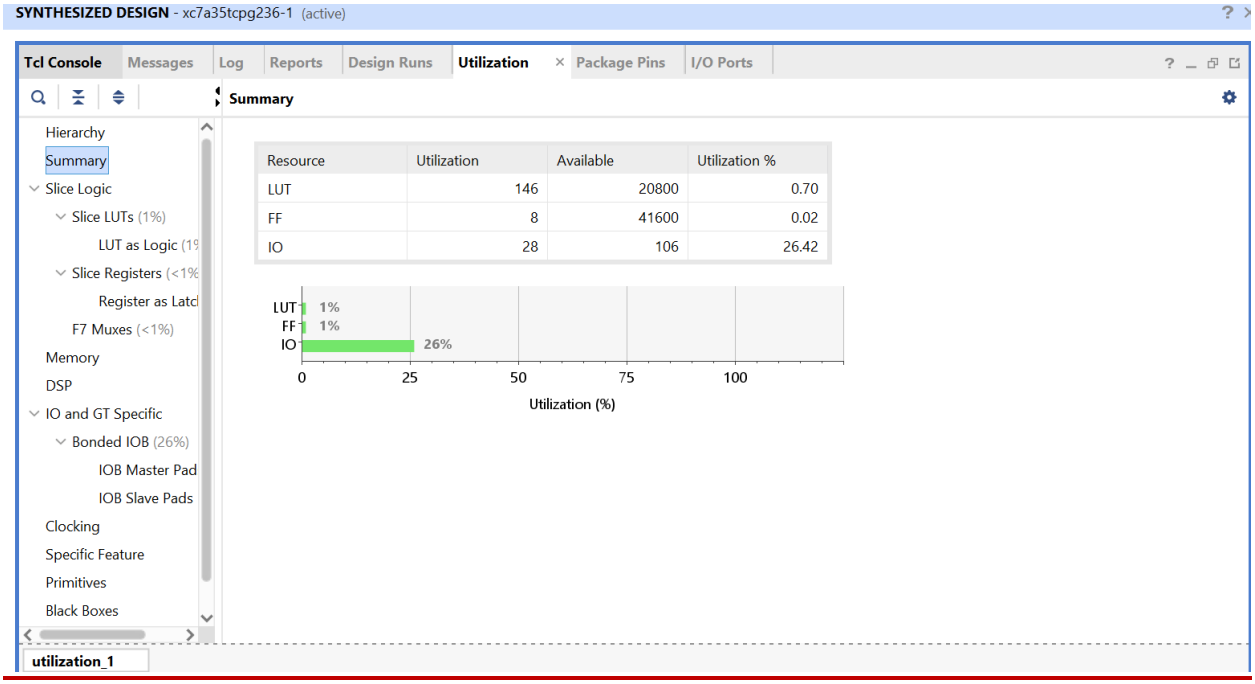

Fig7: Utilization report

5. I/O PIN MAPPING

We have mapped input, output & op variables pins in the basys 3 board for hardware implementation.

Name	Direction	Board Part Pin	Board Part Interface	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco
All ports (28)									
a (8)	IN					☑	14	LVCMOS33*	3.300
a[7]	IN				W13	☑	14	LVCMOS33*	3.300
a[6]	IN				W14	☑	14	LVCMOS33*	3.300
a[5]	IN				V15	☑	14	LVCMOS33*	3.300
a[4]	IN				W15	☑	14	LVCMOS33*	3.300
a[3]	IN				W17	☑	14	LVCMOS33*	3.300
a[2]	IN				W16	☑	14	LVCMOS33*	3.300
a[1]	IN				V16	☑	14	LVCMOS33*	3.300
a[0]	IN				V17	☑	14	LVCMOS33*	3.300
alu_out (8)	OUT					☑	14	LVCMOS33*	3.300
alu_ou...	OUT				V14	☑	14	LVCMOS33*	3.300
alu_ou...	OUT				U14	☑	14	LVCMOS33*	3.300
alu_ou...	OUT				U15	☑	14	LVCMOS33*	3.300
alu_ou...	OUT				W18	☑	14	LVCMOS33*	3.300
alu_ou...	OUT				V19	☑	14	LVCMOS33*	3.300
alu_ou...	OUT				U19	☑	14	LVCMOS33*	3.300
alu_ou...	OUT				E19	☑	14	LVCMOS33*	3.300
alu_ou...	OUT				U16	☑	14	LVCMOS33*	3.300
b (8)	IN					☑	34	LVCMOS33*	3.300
b[7]	IN				R2	☑	34	LVCMOS33*	3.300
b[6]	IN				T1	☑	34	LVCMOS33*	3.300
b[5]	IN				U1	☑	34	LVCMOS33*	3.300
b[4]	IN				W2	☑	34	LVCMOS33*	3.300
b[3]	IN				R3	☑	34	LVCMOS33*	3.300
b[2]	IN				T2	☑	34	LVCMOS33*	3.300
b[1]	IN				T3	☑	34	LVCMOS33*	3.300
b[0]	IN				V2	☑	34	LVCMOS33*	3.300
op (4)	IN					☑	35	LVCMOS33*	3.300
op[3]	IN				L1	☑	35	LVCMOS33*	3.300
op[2]	IN				P1	☑	35	LVCMOS33*	3.300
op[1]	IN				N3	☑	35	LVCMOS33*	3.300
op[0]	IN				P3	☑	35	LVCMOS33*	3.300

Fig8: I/O pin mapping

6. FPGA VALIDATION

We've validated our ALU design on basys 3 FPGA board.

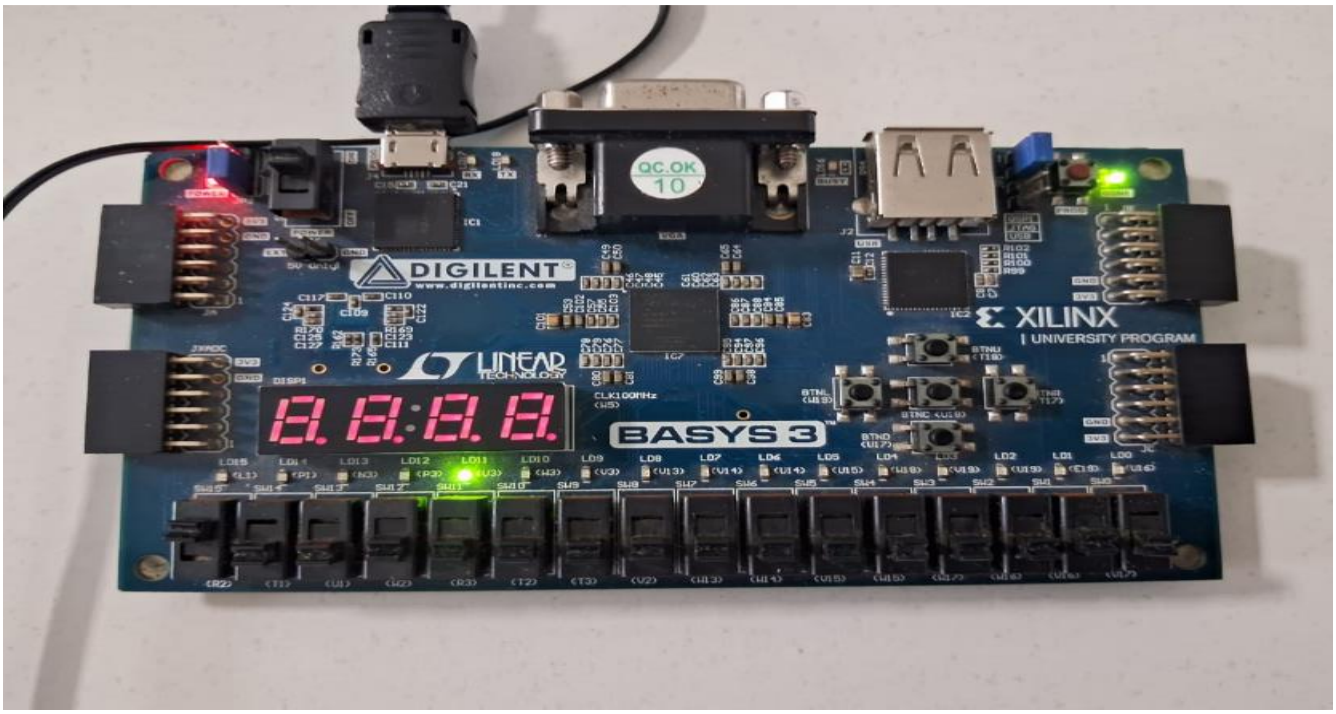

Fig 9: FPGA board

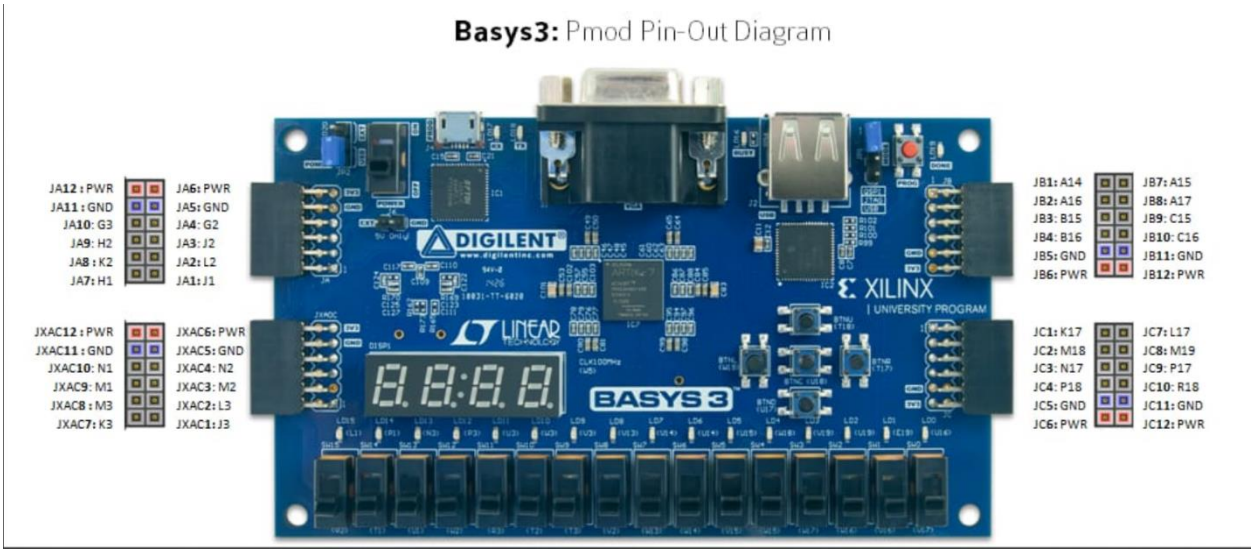

Fig 10: Basys 3 pins

7.Conclusion

In this report, we have implemented 16 different functions by ALU design and simulated output of the design for different inputs in testbench. Further, we generated various synthesis reports to analyze the performance of the ALU design. Here, are some key results in tabular form.

Different reports	**Value**
Total on-chip power	2.264 W
Dynamic power	2.188 W
Static power	0.076 W
Total maximum delay	16 ns

Table 2: Different report and values