Name: - Sravan Chittupalli

VJTI ID: 181060018

EXPERIMENT -1

AIM

Using either VERILOG / VHDL design a 8-bit ALU which can perform the following functions:

1. Addition of 8-bit numbers
2. Subtraction of 8-bit numbers
3. AND operation on 8-bit numbers
4. OR operation on 8-bit numbers
5. NOT operation on 8-bit number
6. Increment of an 8-bit number
7. Decrement of 8-bit number
8. XOR operation on 8-bit number

SOFTWARE

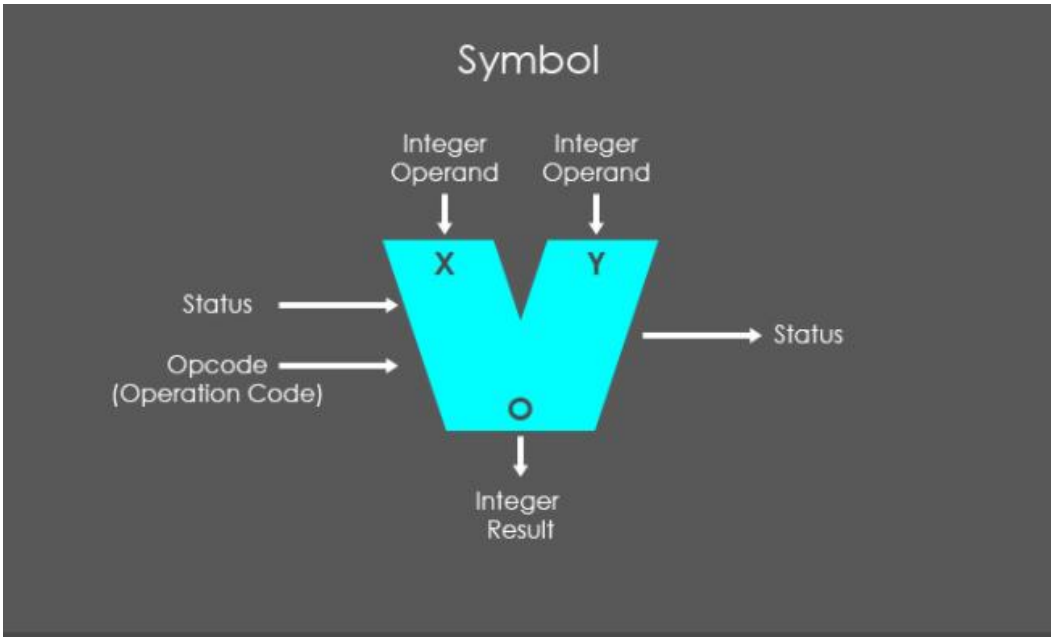
- QUARTUS PRIME

ALU

- ### Introduction

  ALU performs arithmetic and logic operations. It is also one of the most fundamental units of many computing circuits, including the central processing unit (CPU) of computers, and the graphic processing unit (GPU) of video cards.

  In this repo you can find independent modules that can perform 8-bit arithmetic operations like ADDITION, SUBTRACTION and logic operations like AND, OR, XOR. In the end all combine to form an ALU.



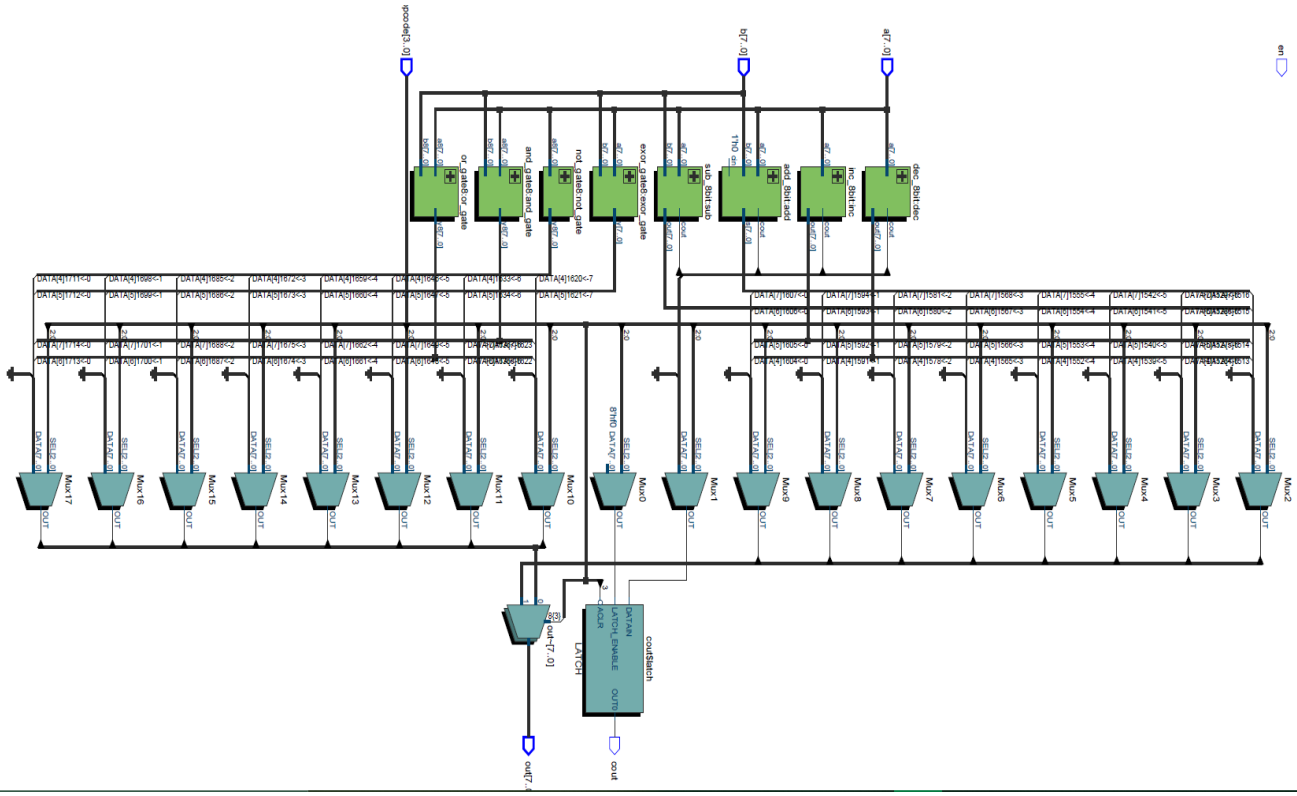
Pin description

Pins	Description
en	enables the ALU
A7 - A0	8-bit Integer operand A
B7 - B0	8-bit Integer operand B
opcode 4-bit	operational code that selects the opearation (refer opcode table)
out7 - out0	8- bit output
cout	carry output

Opcodes

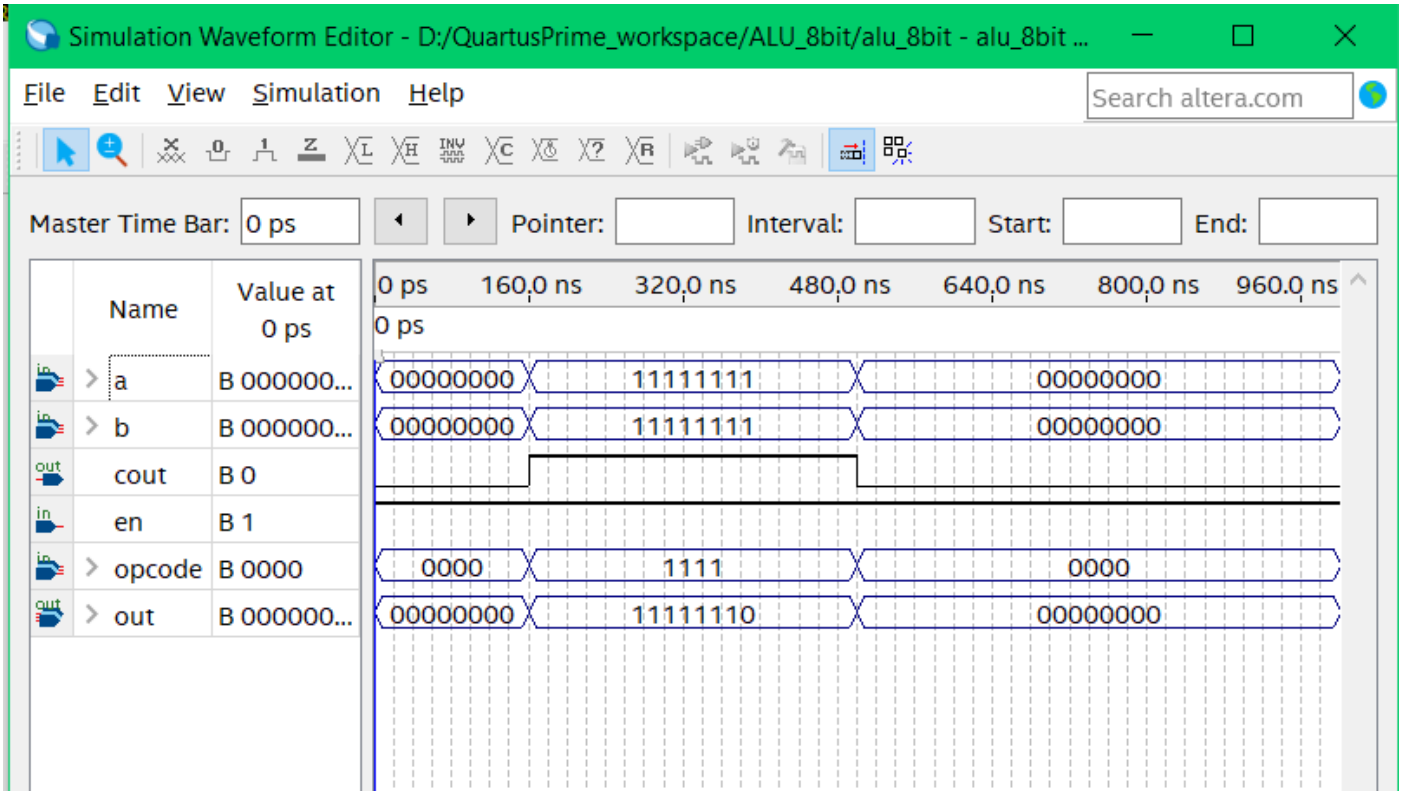
opcode	Operation
1111	A + B
1110	A - B
1101	A + 1
1100	A - 1
0111	A & B
0110	A | B
0101	A ^ B
0100	~(A)

Internal View Of ALU



Output

Using waveform file



Using Testbench

Opcode: 0000	case: default	example: NONE	output: 00000000
Opcode: 1111	case: A + B	example: 11111111 + 11111111	output: 11111110
Opcode: 0110	case: A | B	example: 11001100 | 00110011	output: 11111111
Opcode: 0100	case: ~A	example: ~10101010	output: 01010101
Opcode: 1110	case: A – B	example: 0000000 – 00000001	output: 11111111
Opcode: 0111	case: A & B	example: 11001100 & 00110011	output: 00000000

Conclusion:

Implemented 8-bit ALU in Verilog using basic gates like AND, OR, NOT, NAND, XOR. Also written a testbench code and generated output in Modelsim-Altera.