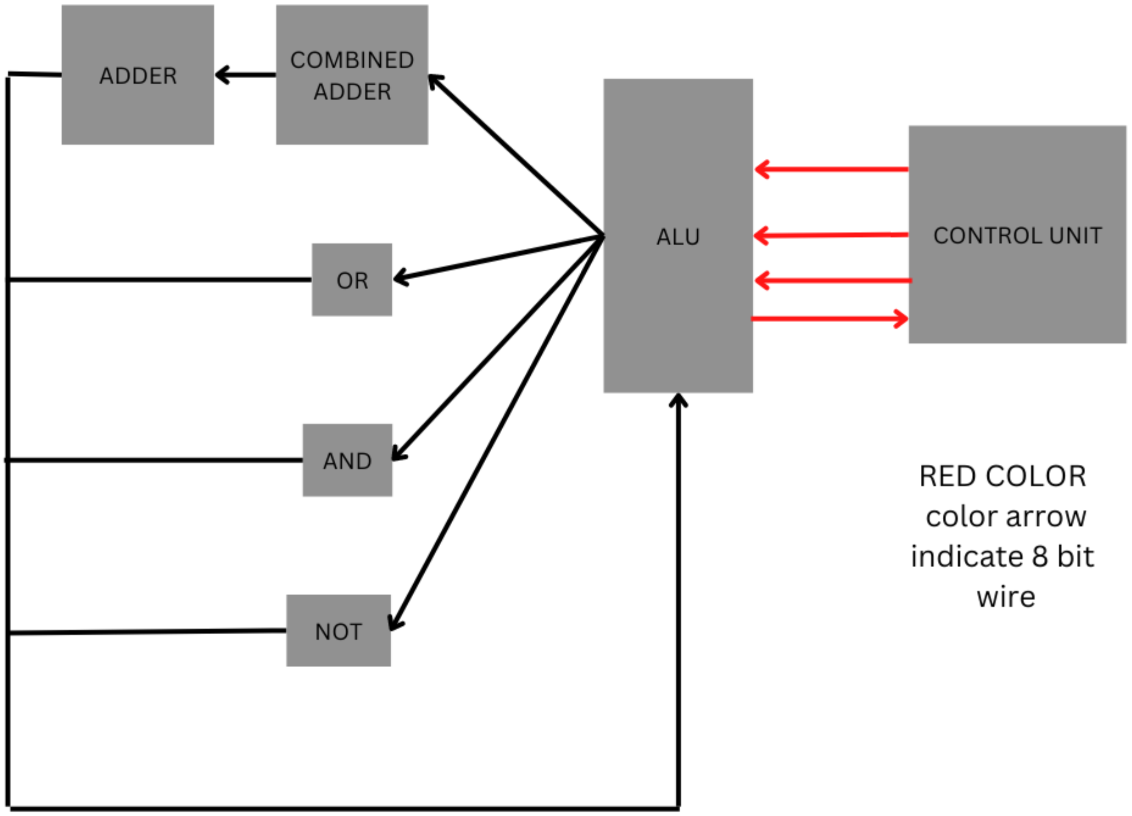
CPU DESIGN REPORT

Athish Pranav D CS21B011

Circuit Diagram



Explanation

Control Unit:

- This takes 19-bit opcode as input and returns 8-bit output.
- It is having inbuilt decoder circuit which takes input as the first 3 bit of the opcode and gives an 8-bit output which is used for selecting the operation.
- The next 8-bit corresponds to operand1, and last 8 corresponds to operand2.

Arithmetic and Logic Unit:

- The inputs to this module are the operands and the select bus.
- This module calls the Combined Adder module, NOT, AND, OR module by passing the select bus bits as input and temporarily receives the output from all these modules
- These outputs are passed into an 8-bit OR gates.
- Only one output will be at most non zero and rest all outputs will be 0.
- Hence the output of OR is the Result of the operation selected.

Combined Adder

- The inputs to this module are the operands, first 4-bit of the select bus (which corresponds to operations involving adders).
- Then we implement a mux which is 4*4 configuration and selects the operand2 from (original operand2, 2’s compliment of operand2, 00000001,11111111) based on the operation requested.
- Then the selected operand2 and original operand1 are passed to the 8-bit full adder.
- And if none of these operations are selected then they are ANDed with 00000000, else 11111111.

Note:

Only one 8-bit Full Adder is called in the entire CPU implementation thus only one Adder circuit is used.

AND

- This is an 8-bit AND module which gets 2 8-bit operands and on select bit.
- If operation is selected then the output of the AND is ANDed with 11111111, else 00000000.

NOT

- This is an 8-bit NOT module which gets 1 8-bit operand1 and on select bit.
- If operation is selected then the output of the NOT is ANDed with 11111111, else 00000000.

OR

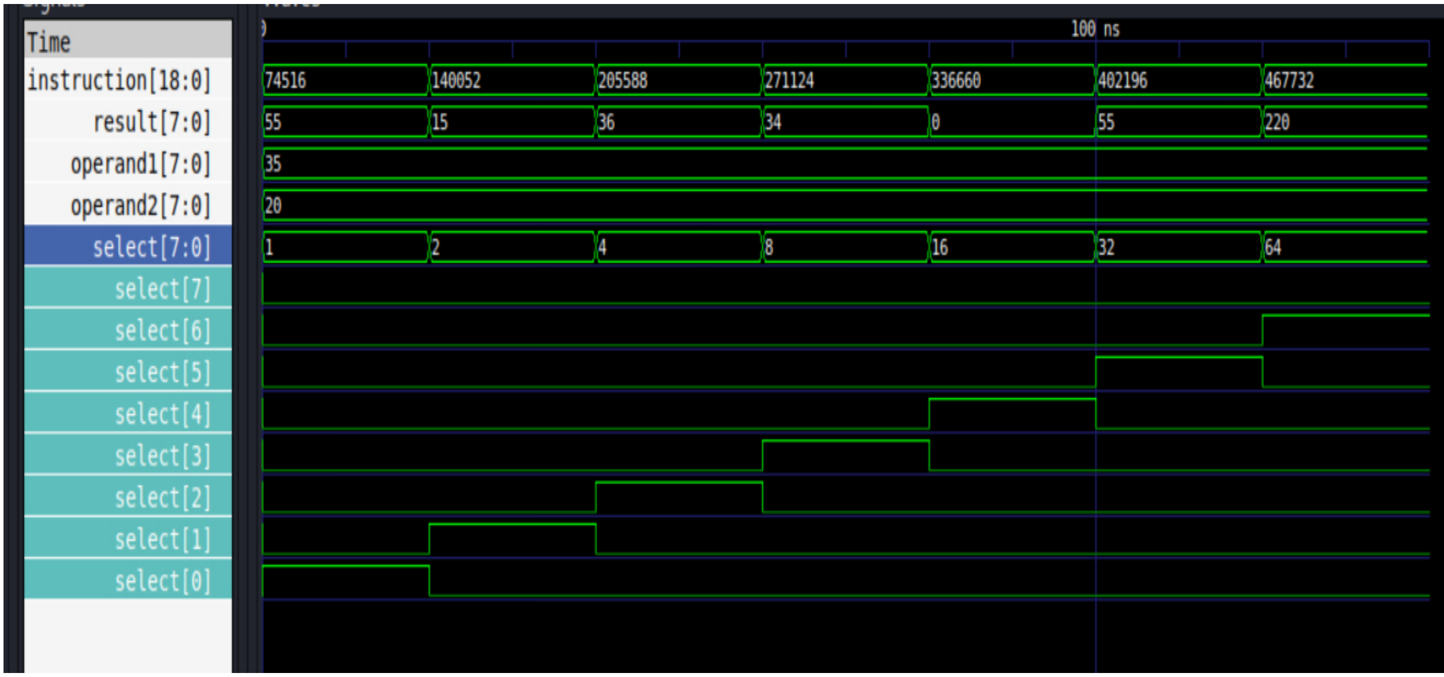
- This is an 8-bit OR module which gets 2 8-bit operands and on select bit.
- If operation is selected then the output of the OR is ANDed with 11111111, else 00000000.

The series of AND gates (AND with select bits present in each of the modules) followed by passing its output to OR gates (present in the ALU) is the MUX but it was not modulated in my code rather implemented directly.

ASSIGN

- This module takes 1-bit as input and gives 8-bit wire as output.
- This module creates an 8-bit wire which has same value as the input bit.

The output of the given testbench.



The output of each module is given below

Combined Adder Module:

The output of this module is 0 when none of the first 4 operations are selected.

Else the output is the result of the operation.

When instruction is 001 the operand2 is selected and passed along with operand1 to the adder.

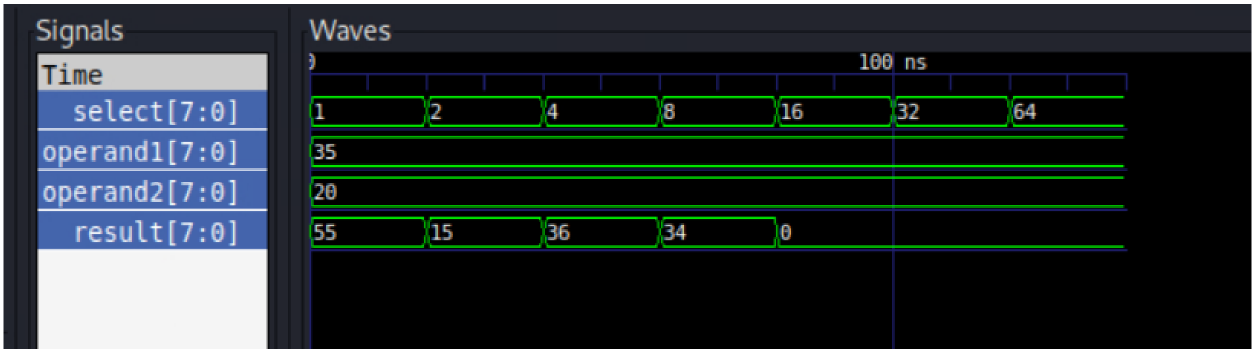
When instruction is 010 the compliment of the operand2 is selected and passed along with operand1 with carry in as 1 to the adder.

When instruction is 011 – 00000001 is selected and passed along with operand1 to the adder.

When instruction 100 – 11111111 is selected and passed along with operand1 to the adder.

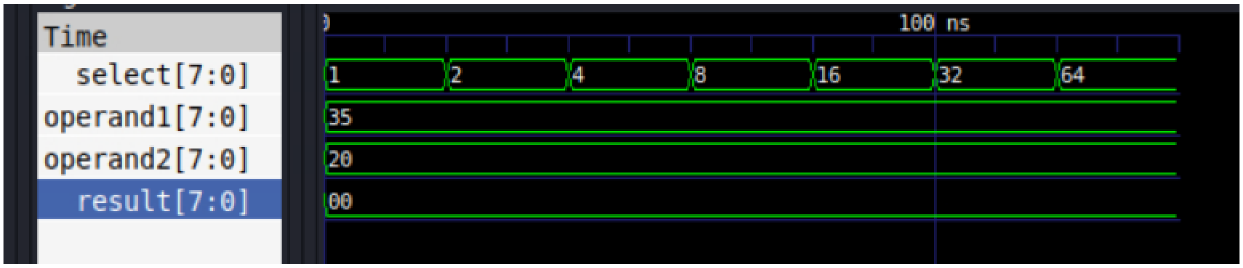
If either of this instruction happens the output of the adder is ANDed with 11111111 and it is the output.

Else the output of the adder is ANDed with 00000000 and it is the output.



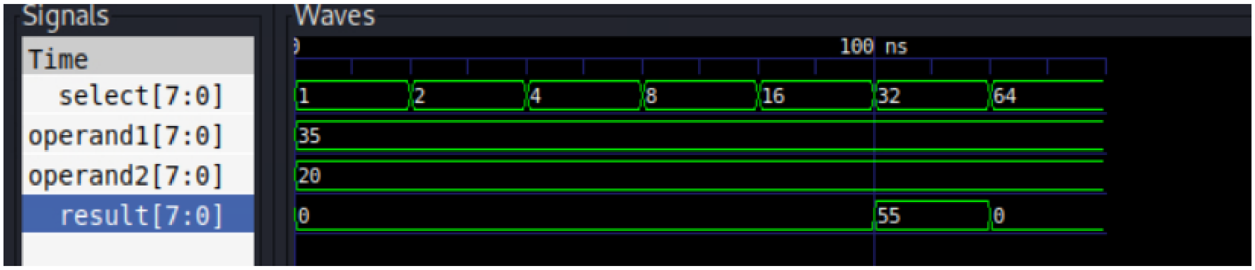
AND Module:

The output of this module is AND of the 2 operands only if the instruction code is 101 (this is passed a one bit select where it is high when this is the instruction else it is low) else the output is 0.



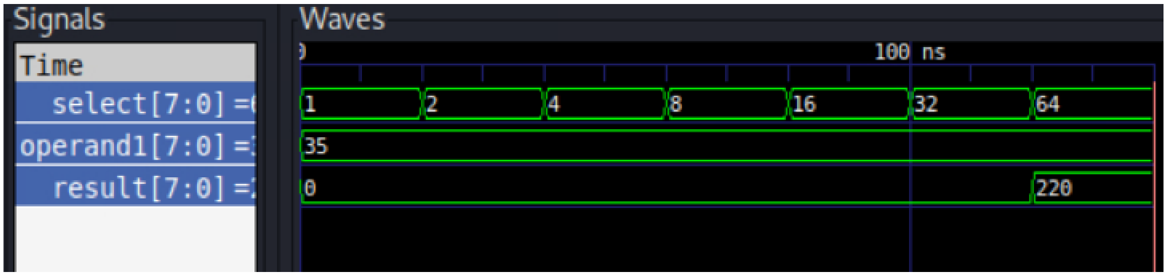
OR Module:

The output of this module is OR of the 2 operands only when the instruction is 110(this is passed a one bit select where it is high when this is the instruction else it is low) else the output is 0.



NOT Module:

The output of this module is NOT of the operand only when the instruction is 111 (this is passed a one bit select where it is high when this is the instruction else it is low) else the output is 0.


Signals
Waves
Time
select[7:0] =
operand1[7:0] =
result[7:0] =
100 ns
1
2
4
8
16
32
64
35
0
220