PES
UNIVERSITY
CELEBRATING 50 YEARS

B.TECH. (CSE)
III SEMESTER

UE23CS251A – DIGITAL DESIGN & COMPUTER ORGANIZATION LABORATORY

DDCO HACKATHON

TEAM NO - 05

SUBMITTED BY

NAME	SRN
1) ABHISHEK P	PES2UG23AM002
2) ADYANTH S	PES2UG23AM007
3) ADYAA G	PES2UG23AM006
4) TEJASHWINI P S	PES2UG24AM806

QUESTION 1:

Problem Statement:

1. **Design a iverilog code for ALU, register file and concatenate Alu –Reg for**

- **8-bit processor which will take 2-bit op code to perform only logical (and, or, nand, nor) operation**
- **Memory should have 8 registers of each size of eight bit and save the result in one of the register and perform read operation to know the output.**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ELECTRONIC CITY CAMPUS, BENGALURU – 560100, KARNATAKA, INDIA

Module code:

```
module AND_block (
input [7:0] inputA, inputB,
output [7:0] and_output
);
assign and_output = inputA & inputB;
endmodule

module OR_block (
input [7:0] inputA, inputB,
output [7:0] or_output
);
assign or_output = inputA | inputB;
endmodule

module NAND_block (
input [7:0] inputA, inputB,
output [7:0] nand_output
);
assign nand_output = ~(inputA & inputB);
endmodule

module NOR_block (
input [7:0] inputA, inputB,
output [7:0] nor_output
);
assign nor_output = ~(inputA | inputB);
endmodule
```

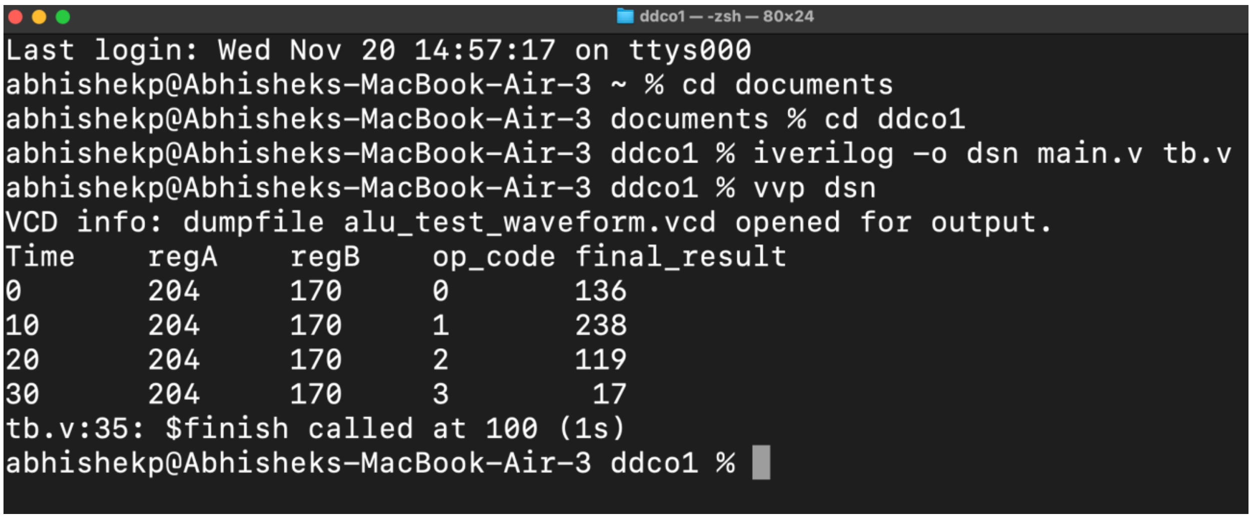
```
module ALU_8bit_unit (

input [7:0] inputA, inputB,

input [1:0] operation,

output [7:0] alu_result

);

wire [7:0] and_res, or_res, nand_res, nor_res;

AND_block and_instance (

.inputA(inputA), .inputB(inputB), .and_output(and_res)

);

OR_block or_instance (

.inputA(inputA), .inputB(inputB), .or_output(or_res)

);

NAND_block nand_instance (

.inputA(inputA), .inputB(inputB), .nand_output(nand_res)

);

NOR_block nor_instance (

.inputA(inputA), .inputB(inputB), .nor_output(nor_res)

);

assign alu_result = (operation == 2'b00) ? and_res :

(operation == 2'b01) ? or_res :

(operation == 2'b10) ? nand_res :

nor_res;

endmodule
```

Testbench code:

```
module tb_ALU_8bit_unit_test;

reg [7:0] regA;

reg [7:0] regB;

reg [1:0] op_code;

wire [7:0] final_result;

ALU_8bit_unit alu_instance (

.inputA(regA),

.inputB(regB),

.operation(op_code),

.alu_result(final_result)

);


initial begin

$dumpfile("alu_test_waveform.vcd");

$dumpvars(0, tb_ALU_8bit_unit_test);

end


initial begin

$display("Time\tregA\tregB\top_code\tfinal_result");

$monitor("%0t\t%d\t%d\t%d\t%d", $time, regA, regB, op_code, final_result);


regA = 8'b11001100; regB = 8'b10101010; op_code = 2'b00;

#10;


regA = 8'b11001100; regB = 8'b10101010; op_code = 2'b01;

#10;
```

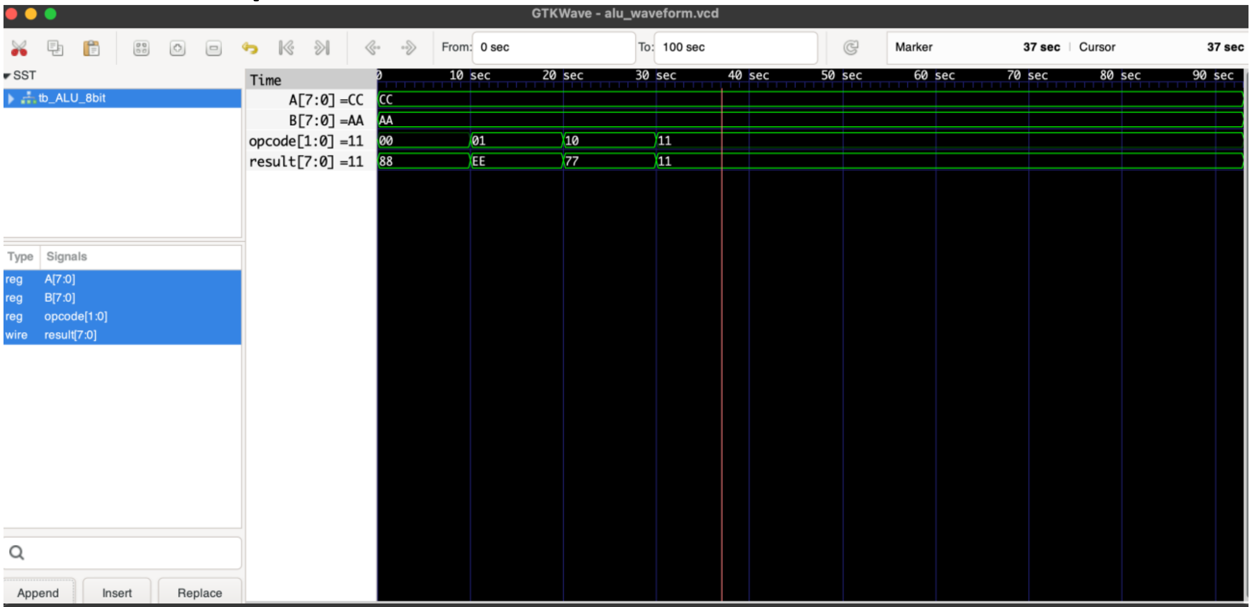
```
regA = 8'b11001100; regB = 8'b10101010; op_code = 2'b10;

#10;


regA = 8'b11001100; regB = 8'b10101010; op_code = 2'b11;

#10;


#60; // Wait till 100 time units

$finish;

end

endmodule
```

Console output in decimal snapshot:

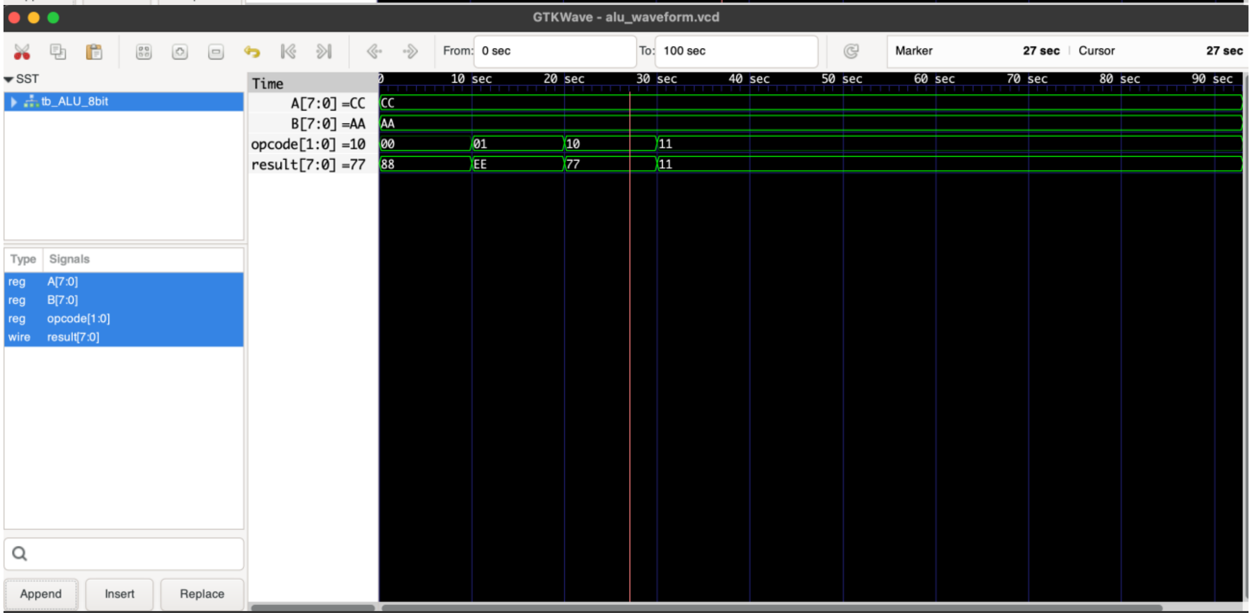


```
Last login: Wed Nov 20 14:57:17 on ttys000
abhishekp@Abhisheks-MacBook-Air-3 ~ % cd documents
abhishekp@Abhisheks-MacBook-Air-3 documents % cd ddco1
abhishekp@Abhisheks-MacBook-Air-3 ddco1 % iverilog -o dsn main.v tb.v
abhishekp@Abhisheks-MacBook-Air-3 ddco1 % vvp dsn
VCD info: dumpfile alu_test_waveform.vcd opened for output.
Time    regA    regB    op_code final_result
0       204     170     0       136
10      204     170     1       238
20      204     170     2       119
30      204     170     3        17
tb.v:35: $finish called at 100 (1s)
abhishekp@Abhisheks-MacBook-Air-3 ddco1 %
```

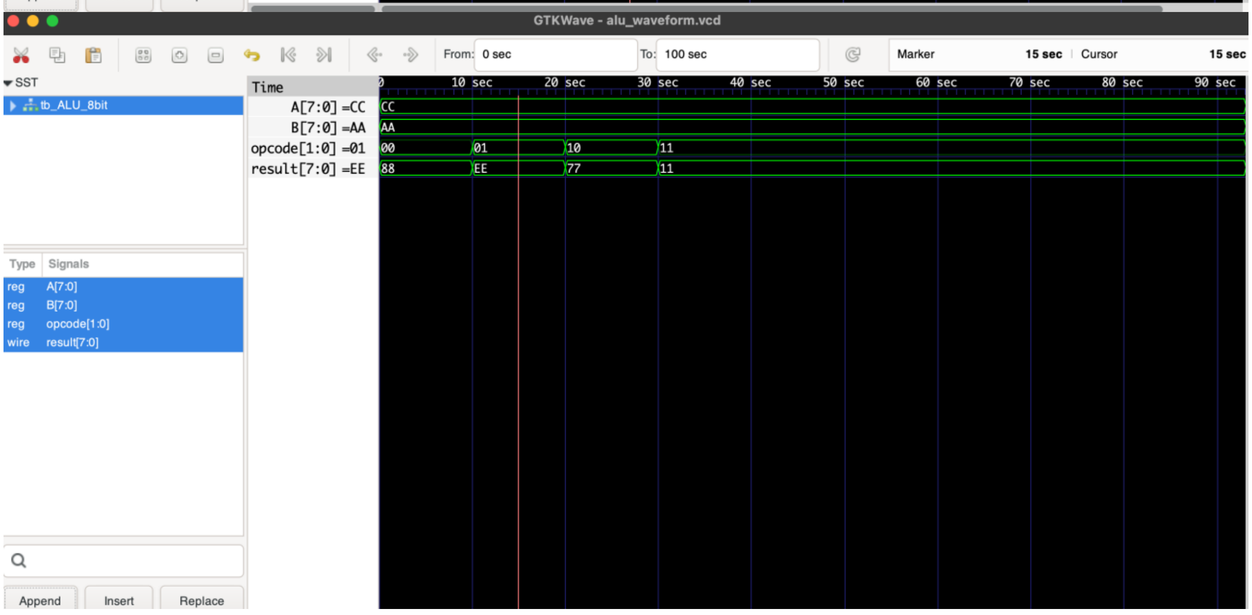
GTK wave snapshot:



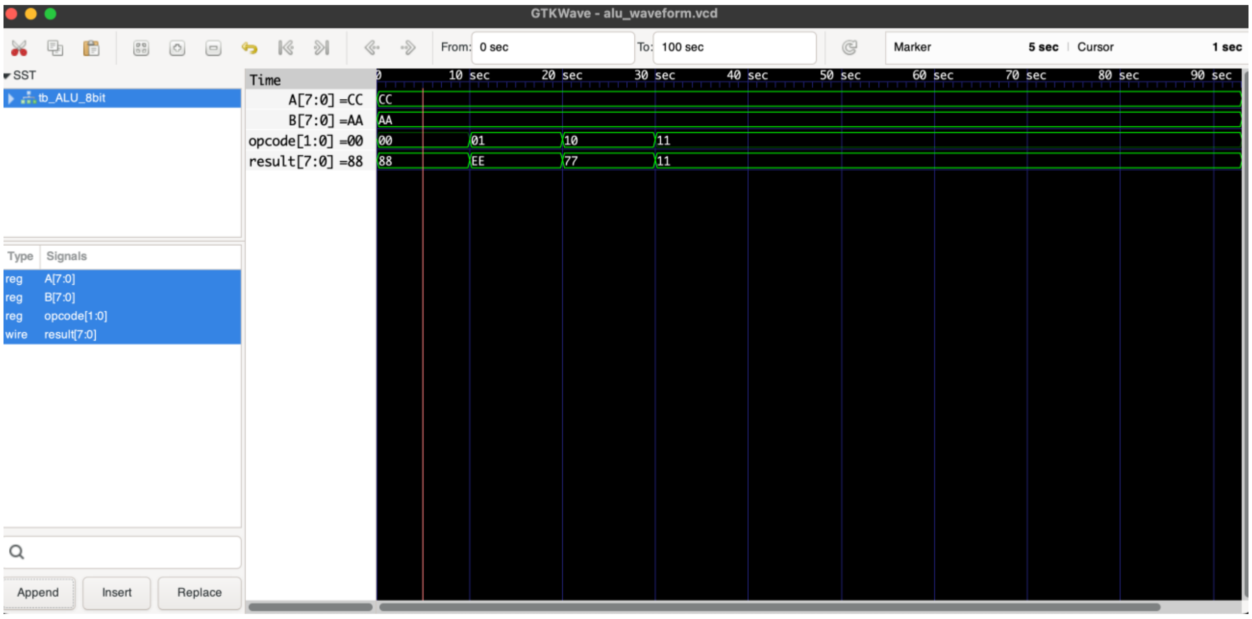
```
GTKWave - alu_waveform.vcd
From: 0 sec   To: 100 sec   Marker 37 sec | Cursor 37 sec

SST
  tb_ALU_8bit

Type  Signals
reg   A[7:0]
reg   B[7:0]
reg   opcode[1:0]
wire  result[7:0]

Time              0      10 sec   20 sec   30 sec   40 sec   50 sec   60 sec   70 sec   80 sec   90 sec
A[7:0] =CC        CC
B[7:0] =AA        AA
opcode[1:0] =11   00     01       10       11
result[7:0] =11   88     EE       77       11
```



```
GTKWave - alu_waveform.vcd
From: 0 sec   To: 100 sec   Marker 27 sec | Cursor 27 sec

SST
  tb_ALU_8bit

Type  Signals
reg   A[7:0]
reg   B[7:0]
reg   opcode[1:0]
wire  result[7:0]

Time              0      10 sec   20 sec   30 sec   40 sec   50 sec   60 sec   70 sec   80 sec   90 sec
A[7:0] =CC        CC
B[7:0] =AA        AA
opcode[1:0] =10   00     01       10       11
result[7:0] =77   88     EE       77       11
```



```
GTKWave - alu_waveform.vcd
From: 0 sec   To: 100 sec   Marker 15 sec | Cursor 15 sec

SST
  tb_ALU_8bit

Type  Signals
reg   A[7:0]
reg   B[7:0]
reg   opcode[1:0]
wire  result[7:0]

Time              0      10 sec   20 sec   30 sec   40 sec   50 sec   60 sec   70 sec   80 sec   90 sec
A[7:0] =CC        CC
B[7:0] =AA        AA
opcode[1:0] =01   00     01       10       11
result[7:0] =EE   88     EE       77       11
```



Type	Signals
reg	A[7:0]
reg	B[7:0]
reg	opcode[1:0]
wire	result[7:0]

```
Time              0      10 sec   20 sec   30 sec   40 sec   50 sec   60 sec   70 sec   80 sec   90 sec
A[7:0] =CC        CC
B[7:0] =AA        AA
opcode[1:0] =00   00     01       10       11
result[7:0] =88   88     EE       77       11
```