Question No. 1 (05 Marks) Design an ALU datapath that performs the following operations in parallel on two 16-bit signed inputs A and B and assigns the value of one of the outputs to 16-bit C. The selection of operation is based on a 2-bit selection line. Code the design in Verilog HDL. Write test vectors to verify the implementation.

1. C = A + B
2. C = A - B
3. C = A & B
4. C = A | B

Solution:

CODE:

// Code your design here

module alu(a,b,op,c);

input [15:0]a,b;

input[1:0]op;

output reg[15:0]c;

always@(\*)

begin

case(op)

2'b00:c=a+b;

2'b01:c=a-b;

2'b10:c=a&b;

2'b11:c=a|b;

default:c=0;

endcase

end

endmodule

TEST BENCH:

// Code your testbench here

// or browse Examples

module alu\_tb;

reg[15:0]a,b;

reg[1:0]op;

wire[15:0]c;

alu alu\_tb(a,b,op,c);

initial

begin

$monitor(a,b,op,c);

//$dumpfile("dump.vcd");

op=2'b00;

a=-16;

b=17;

#5

op=2'b01;

a=16;

b=17;

#5

op=2'b10;

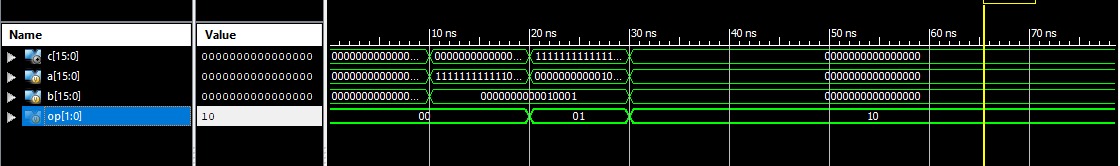
a=0;

b=0;

end

endmodule

OUTPUT:



Question No. 2

Write RTL Verilog code for the following design, also give the test bench for testing the system

CODE:

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 23:29:51 04/23/2021

// Design Name:

// Module Name: NEW

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module fir\_filter (

input clk,

input signed [15:0] data\_in,

output reg signed [15:0] data\_out

);

parameter signed [15:0] b0 = 16'b1101110110111011;

parameter signed [15:0] b1 = 16'b1110101010001110;

parameter signed [15:0] b2 = 16'b0011001111011011;

parameter signed [15:0] b3 = 16'b0110100000001000;

parameter signed [15:0] b4 = 16'b0110100000001000;

parameter signed [15:0] b5 = 16'b0011001111011011;

parameter signed [15:0] b6 = 16'b1110101010001110;

parameter signed [15:0] b7 = 16'b1101110110111011;

reg signed [15:0] xn [0:7];

wire signed [39:0] yn;

always @(posedge clk)

begin

xn[0] <= data\_in;

xn[1] <= xn[0];

xn[2] <= xn[1];

xn[3] <= xn[2];

xn[4] <= xn[3];

xn[5] <= xn[4];

xn[6] <= xn[5];

xn[7] <= xn[6];

data\_out <= yn[30:15];

end

assign yn = xn[0] \* b0 + xn[1] \* b1 + xn[2] \* b2 +

xn[3] \* b3 + xn[4] \* b4 + xn[5] \* b5 +

xn[6] \* b6 + xn[7] \* b7;

endmodule

TEST BENCH:

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 23:39:23 04/23/2021

// Design Name: fir\_filter

// Module Name: E:/Semester 6/DSD-LAB/Assignmen\_2/test\_fir\_filter.v

// Project Name: Assignmen\_2

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: fir\_filter

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

module test\_fir\_filter;

// Inputs

reg clk;

reg [15:0] data\_in;

// Outputs

wire [15:0] data\_out;

// Instantiate the Unit Under Test (UUT)

fir\_filter uut (

.clk(clk),

.data\_in(data\_in),

.data\_out(data\_out)

);

initial begin

// Initialize Inputs

clk = 0;

data\_in = 0;

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

end

endmodule

OUTPUT:

