Name: Adithya M SRN: PES1UG20CS621 Section: K

alu.v

```
module alu (input wire [1:0] op, input wire [15:0] i0, i1,output wire [15:0] o, output
wire cout);
	wire [14:0]c;

	alu_slice a0(op[0],i0[0],i1[0],op,o[0],c[0]);
	alu_slice a1(c[0],i0[1],i1[1],op,o[1],c[1]);
	alu_slice a2(c[1],i0[2],i1[2],op,o[2],c[2]);
	alu_slice a3(c[2],i0[3],i1[3],op,o[3],c[3]);
	alu_slice a4(c[3],i0[4],i1[4],op,o[4],c[4]);
	alu_slice a5(c[4],i0[5],i1[5],op,o[5],c[5]);
	alu_slice a6(c[5],i0[6],i1[6],op,o[6],c[6]);
	alu_slice a7(c[6],i0[7],i1[7],op,o[7],c[7]);
	alu_slice a8(c[7],i0[8],i1[8],op,o[8],c[8]);
	alu_slice a9(c[8],i0[9],i1[9],op,o[9],c[9]);
	alu_slice a10(c[9],i0[10],i1[10],op,o[10],c[10]);
	alu_slice a11(c[10],i0[11],i1[11],op,o[11],c[11]);
	alu_slice a12(c[11],i0[12],i1[12],op,o[12],c[12]);
	alu_slice a13(c[12],i0[13],i1[13],op,o[13],c[13]);
	alu_slice a14(c[13],i0[14],i1[14],op,o[14],c[14]);
	alu_slice a15(c[14],i0[15],i1[15],op,o[15],cout);
endmodule
```

alu_slice.v

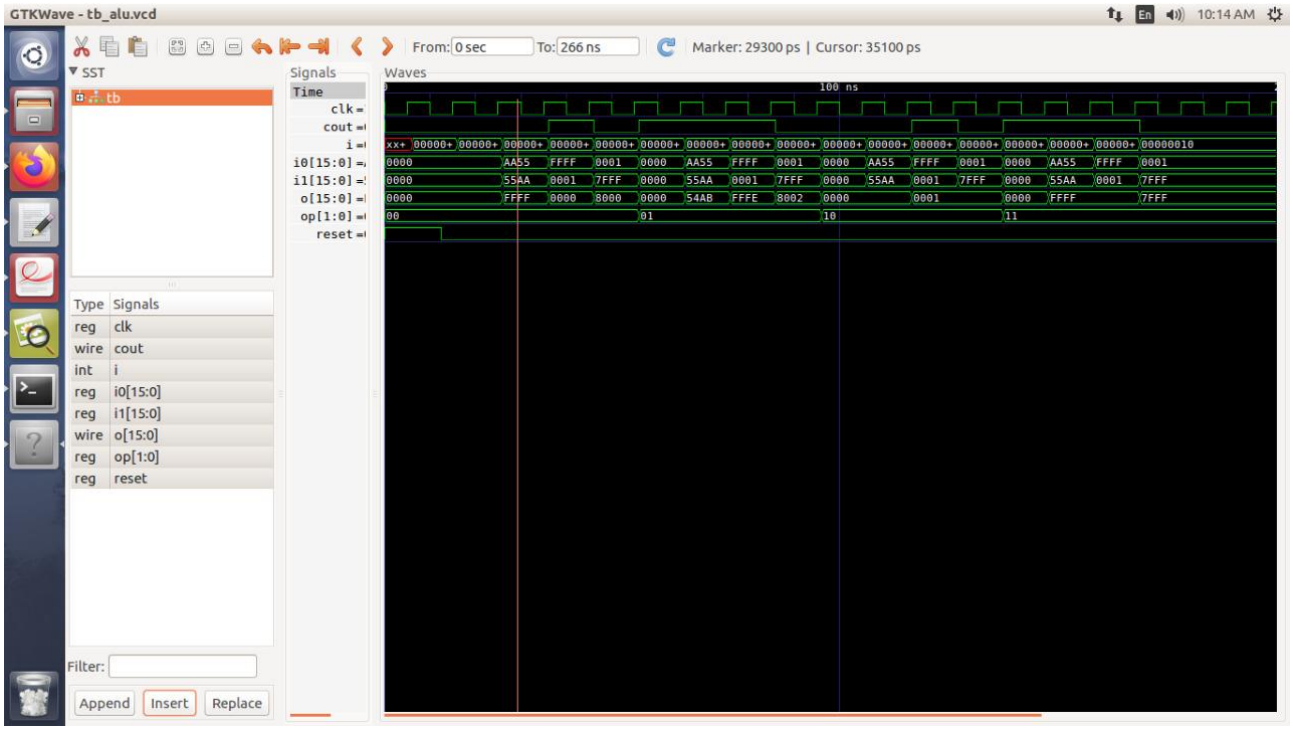
```
module full_adder(input wire a, b, cin,output wire cout, sum);
	assign sum = a^b^cin;
	assign cout = (b&cin)|(a&cin)|(a&b);
endmodule

module xor2 (input wire i0, i1, output wire o);
  assign o = i0 ^ i1;
endmodule

module and2 (input wire i0, i1, output wire o);
  assign o = i0 & i1;
endmodule

module or2 (input wire i0, i1, output wire o);
  assign o = i0 | i1;
endmodule
```

```
module mux2 (input wire i0, i1, j, output wire o);
  assign o = (j==0)?i0:i1;
endmodule

module alu_slice(input wire cin,i0,i1,input wire [1:0]op,output wire o,cout);
        wire temp,o1,o2,o3,o4;

        xor2 x(i1,op[0],temp);
        full_adder f(i0,temp,cin,cout,o1);

        and2 a(i0,i1,o2);
        or2 orr(i0,i1,o3);

        mux2 m1(o2,o3,op[0],o4);
        mux2 m2(o1,o4,op[1],o);
endmodule
```

tb_alu.v

```
`timescale 1 ns / 100 ps
`define TESTVECS 16

module tb;
  reg clk, reset;
  reg [1:0] op; reg [15:0] i0, i1;
  wire [15:0] o; wire cout;
  reg [33:0] test_vecs [0:(`TESTVECS-1)];
  integer i;
  initial begin $dumpfile("dump.vcd"); $dumpvars(0,tb); end
  initial begin reset = 1'b1; #12.5 reset = 1'b0; end
  initial clk = 1'b0; always #5 clk =~ clk;
  initial begin
    test_vecs[0][33:32] = 2'b00; test_vecs[0][31:16] = 16'h0000;test_vecs[0][15:0] =
16'h0000;
    test_vecs[1][33:32] = 2'b00; test_vecs[1][31:16] = 16'haa55;test_vecs[1][15:0] =
16'h55aa;
    test_vecs[2][33:32] = 2'b00; test_vecs[2][31:16] = 16'hffff;test_vecs[2][15:0] =
16'h0001;
    test_vecs[3][33:32] = 2'b00; test_vecs[3][31:16] = 16'h0001;test_vecs[3][15:0] =
16'h7fff;
    test_vecs[4][33:32] = 2'b01; test_vecs[4][31:16] = 16'h0000;test_vecs[4][15:0] =
16'h0000;
```

```
    test_vecs[5][33:32] = 2'b01; test_vecs[5][31:16] = 16'haa55;test_vecs[5][15:0] =
16'h55aa;
    test_vecs[6][33:32] = 2'b01; test_vecs[6][31:16] = 16'hffff;test_vecs[6][15:0] =
16'h0001;
    test_vecs[7][33:32] = 2'b01; test_vecs[7][31:16] = 16'h0001;test_vecs[7][15:0] =
16'h7fff;
    test_vecs[8][33:32] = 2'b10; test_vecs[8][31:16] = 16'h0000;test_vecs[8][15:0] =
16'h0000;
    test_vecs[9][33:32] = 2'b10; test_vecs[9][31:16] = 16'haa55;test_vecs[9][15:0] =
16'h55aa;
    test_vecs[10][33:32] = 2'b10; test_vecs[10][31:16] = 16'hffff;test_vecs[10][15:0] =
16'h0001;
    test_vecs[11][33:32] = 2'b10; test_vecs[11][31:16] = 16'h0001;test_vecs[11][15:0]
= 16'h7fff;
    test_vecs[12][33:32] = 2'b11; test_vecs[12][31:16] = 16'h0000;test_vecs[12][15:0]
= 16'h0000;
    test_vecs[13][33:32] = 2'b11; test_vecs[13][31:16] = 16'haa55;test_vecs[13][15:0]
= 16'h55aa;
    test_vecs[14][33:32] = 2'b11; test_vecs[14][31:16] = 16'hffff;test_vecs[14][15:0] =
16'h0001;
    test_vecs[15][33:32] = 2'b11; test_vecs[15][31:16] = 16'h0001;test_vecs[15][15:0]
= 16'h7fff;
  end
  initial {op, i0, i1} = 0;
  alu alu_0 (op, i0, i1, o, cout);
  initial begin
    #6 for(i=0;i<`TESTVECS;i=i+1)
      begin #10 {op, i0, i1}=test_vecs[i]; end
    #100 $finish;
  end
endmodule
```


GTKWave - tb_alu.vcd
10:14 AM
From: 0 sec
To: 266 ns
Marker: 29300 ps | Cursor: 35100 ps
SST
tb
Type Signals
reg clk
wire cout
int i
reg i0[15:0]
reg i1[15:0]
wire o[15:0]
reg op[1:0]
reg reset
Filter:
Append
Insert
Replace
Signals
Time
clk
cout
i
i0[15:0]
i1[15:0]
o[15:0]
op[1:0]
reset
Waves
100 ns