

Opcode	Function
000	$\text{outW} = \text{inB} + \text{inA} + \text{inC}$
001	$\text{outW} = \text{inB} + \text{inA} \times 2.0$ (Sign Extend)
010	$\text{outW} = \text{inB} + 1$
011	$\text{outW} = \text{inB} \div 1.5$
100	$\text{outW} = \text{inB} \& \text{inA}$ (Bitwise)
101	$\text{outW} = \text{inB}   \text{inA}$ (Bitwise)
110	$\text{outW} = \sim \text{inB}$ (Bitwise)
111	No operation

16'd0 is given for no-operation instead of 16'bz since Yosys does not support tri-state logic.

1-

```

`timescale 1ns/1ns
module ALU1 (input signed [15:0] inB,inA,input [2:0] opc,input inC,output zer,neg,output logic signed [15:0] w);
    assign zer=~(|w);
    assign neg=w[15];
    always @(inB,inA,opc,inC) begin
        w=16'd0;
        case(opc)
            3'd0:w=inB+inA+inC;
            3'd1:w=inB+(inA>>>1);
            3'd2:w=inB+1;
            3'd3:w=inB+(inB>>>1);
            3'd4:w=inB&inA;
            3'd5:w=inB|inA;
            3'd6:w=~inB;
            default:w=16'd0;
        endcase
    end
endmodule

```

## Yosys synthesis:

```
4.1.2. Re-integrating ABC results.
ABC RESULTS:          NAND cells:      211
ABC RESULTS:          NOR cells:       457
ABC RESULTS:          NOT cells:       169
ABC RESULTS:          internal signals:  484
ABC RESULTS:          input signals:    36
ABC RESULTS:          output signals:    17
Removing temp directory.
```

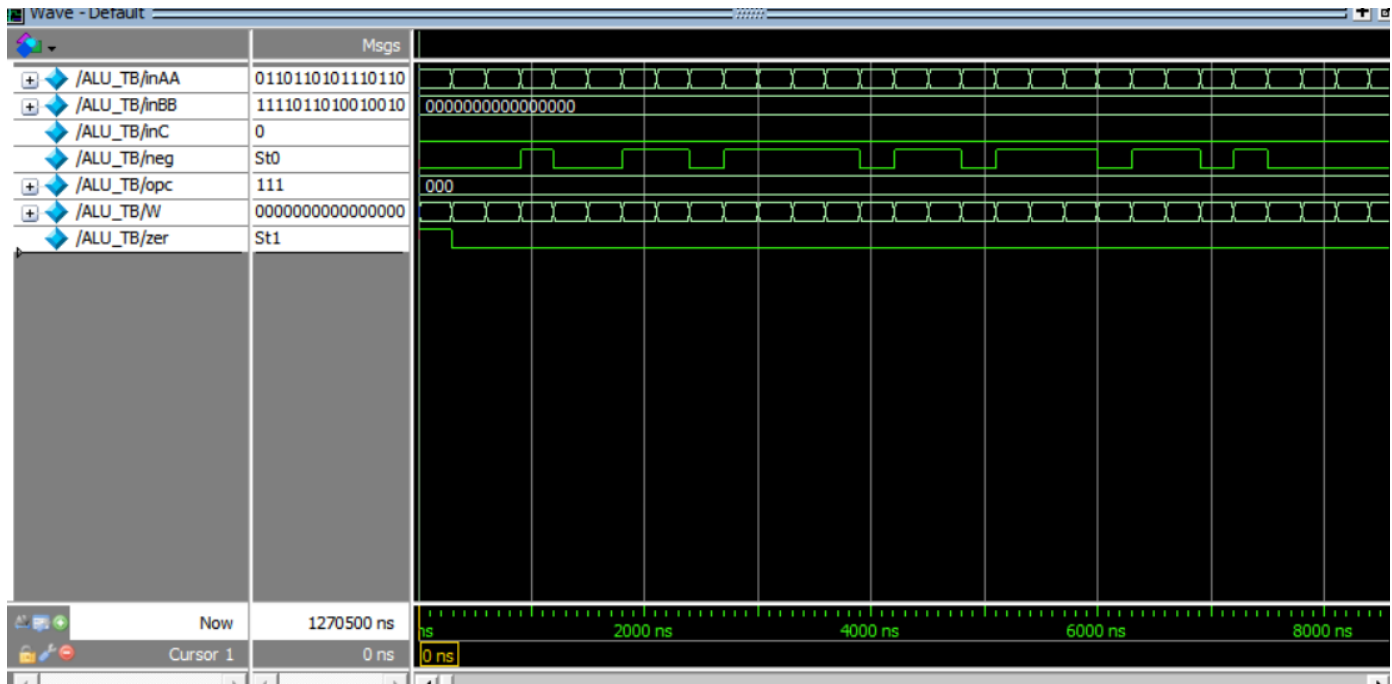
Test bench code:

```
module ALU_TB();
  wire [15:0]w;
  reg [15:0]inAA = 16'b0;
  reg [15:0]inBB = 16'b0;
  reg inC = 1'b0;
  reg [2:0]opc = 3'b0;
  wire neg ;
  wire zer;
  myALU1 CUT( .B(inBB) , .A(inAA) , .f(opc) , .c(inC) , .z(zer) , .n(neg), .w(w) );

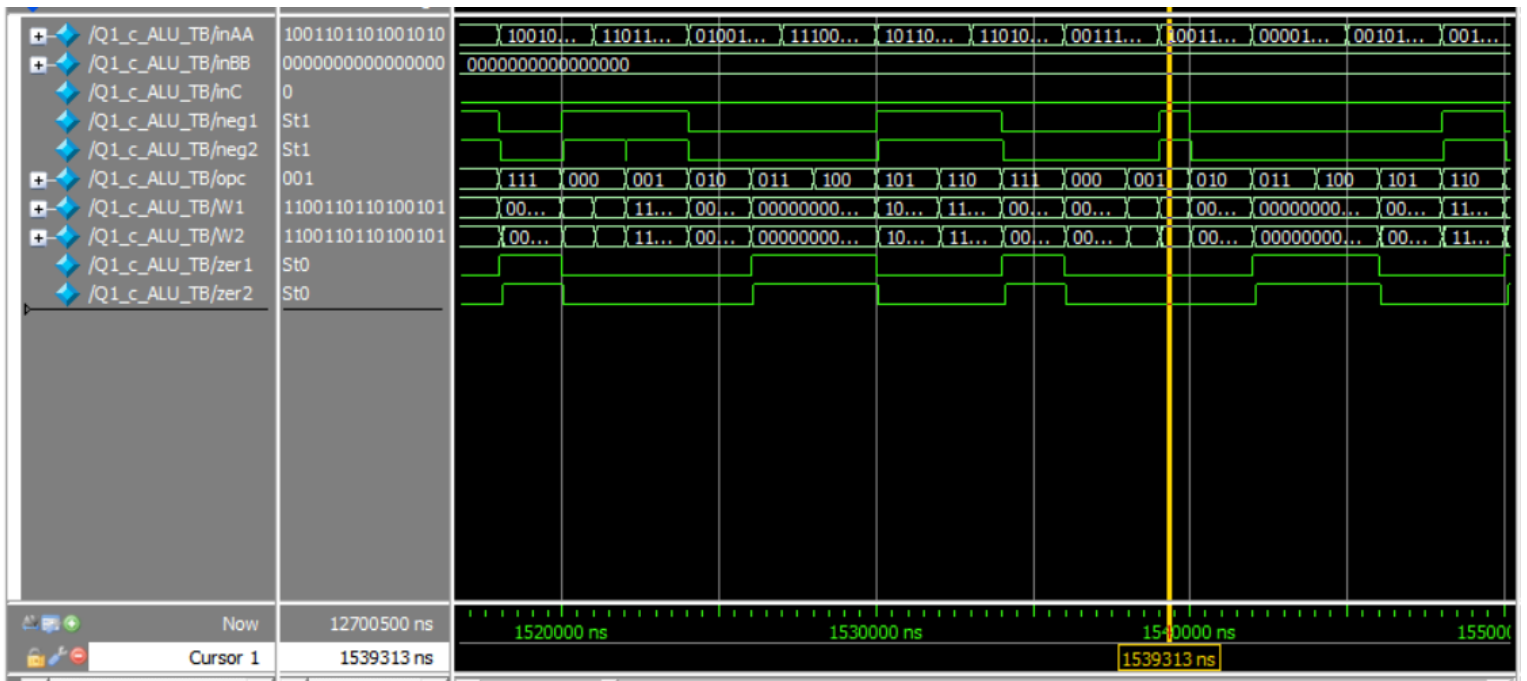
  initial repeat(18)#200 opc = opc + 1'b1;

  initial begin
    repeat(20) #300 inAA = $random;
    repeat(20) #200 inBB = $random;
    repeat(9) #300 inC = $random;
    #500 $stop;
  end

endmodule
```



1-c



1-d)

در بخش a چون با `always` و `case` داشتیم از دلیلی و سخت افزار بی خبر بودیم پس نمی توان از این لحاظ با بخش سی مقایسه کرد همچنین با توجه به حرف بالا و اینکه بخش سی دلیلی در خود داشت جواب ها با اینکه یکسان است اما برا خروجی یوسیس مدت زمان بیشتری طول می کشد تا نمایش داده شود.

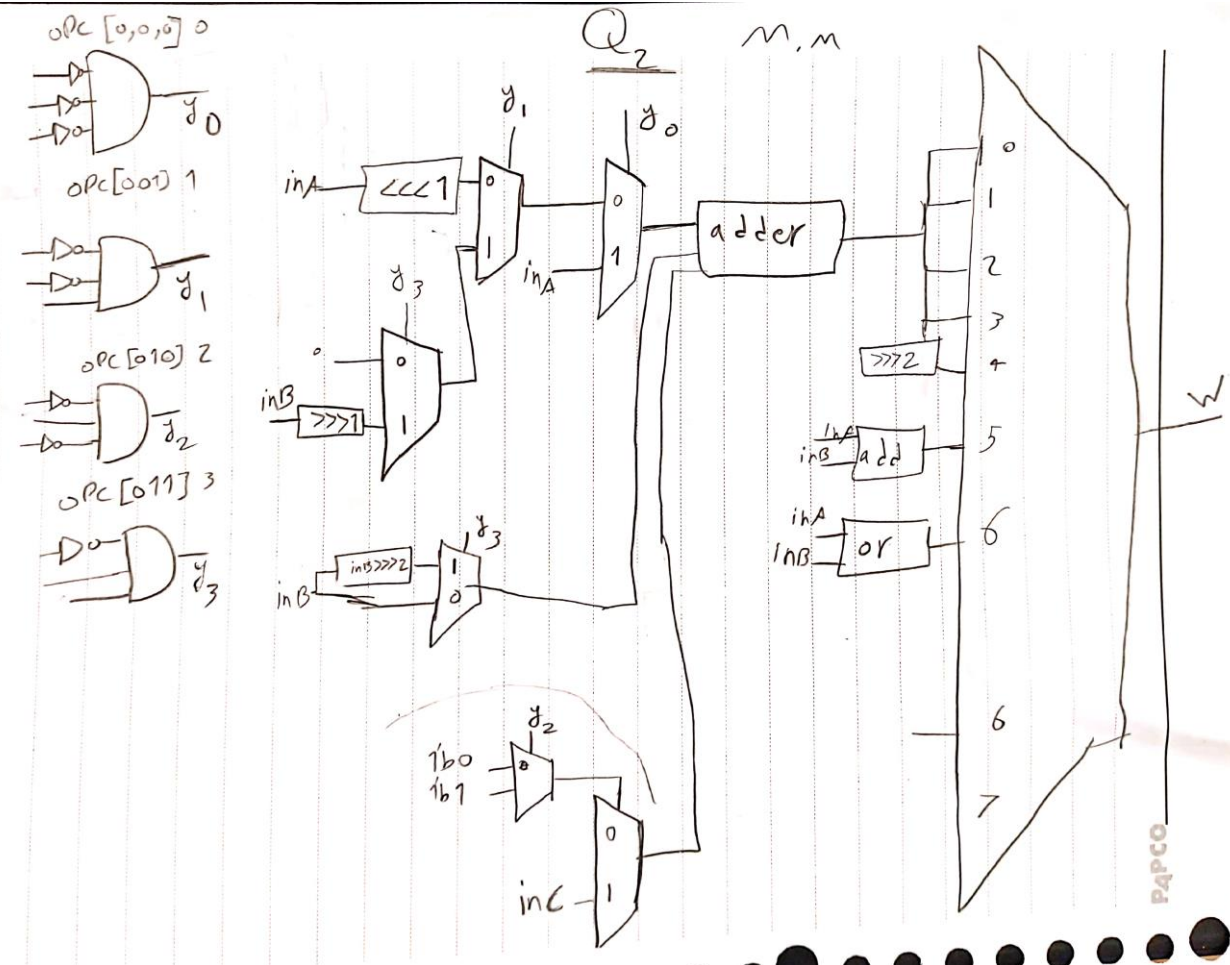
```
-----
# Memory Statistics
#   mem: size after elab (VSZ)          142.29 Mb
#   mem: size during sim (VSZ)         174.45 Mb
# Elaboration Time
#   elab: wall time                     0.14 s
#   elab: cpu time                      0.11 s
# Simulation Time
#   sim: wall time                      0.02 s
#   sim: cpu time                       0.00 s
# Tcl Command Time
#   cmd: wall time                      52.73 s
#   cmd: cpu time                       0.25 s
# Total Time
#   total: wall time                    52.89 s
#   total: cpu time                     0.36 s
#
```

Figure 2pre

```
VSIM 120> simstats
# Memory Statistics
#   mem: size after elab (VSZ)          142.27 Mb
#   mem: size during sim (VSZ)         174.46 Mb
# Elaboration Time
#   elab: wall time                     0.19 s
#   elab: cpu time                      0.20 s
# Simulation Time
#   sim: wall time                      0.16 s
#   sim: cpu time                       0.16 s
# Tcl Command Time
#   cmd: wall time                      42.88 s
#   cmd: cpu time                       0.27 s
# Total Time
#   total: wall time                    43.23 s
#   total: cpu time                     0.63 s
#
```

Figure 1post

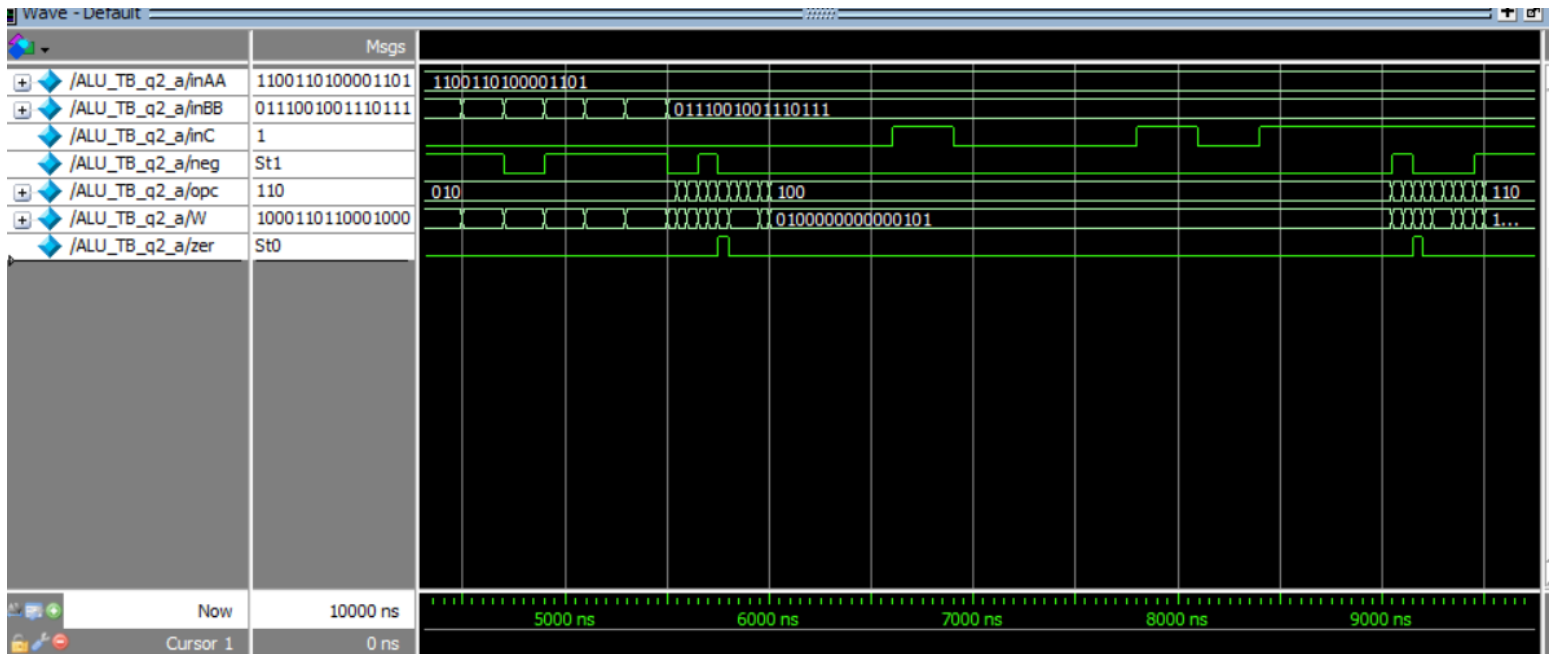
## Design



Code:

```
1  `timescale 1ns/1ns
2  module myALU2 (input signed [15:0] inB, inA, input [2:0] opc , input inC,
3  [15:0] output zer,neg, output logic signed [15:0] w);
4  wire [15:0]temp1 , temp2 ;
5  assign neg = w[15];
6  assign zer = ~|w;
7
8  assign temp1 = (opc == 3'b000) ? inA :
9  (opc == 3'b001) ? inA<<<1 :
10 (opc == 3'b011) ? inB>>>1 : 16'd1;
11
12
13
14 assign temp2 = inB + ((~|opc)&inC);
15
16
17 always @(inA,inB,inC,opc,temp2) begin
18     w = 16'b0;
19     case (opc)
20         3'b000 : w = temp2;
21         3'b001 : w = temp2;
22         3'b010 : w = temp2;
23         3'b011 : w = temp2>>>2;
24         3'b100 : w = inA & inB;
25         3'b101 : w = inA | inB;
26         3'b110 : w = ~inB;
27         default : w = 16'b0;
28     endcase
29 end
30
31 endmodule
```

Test bench:

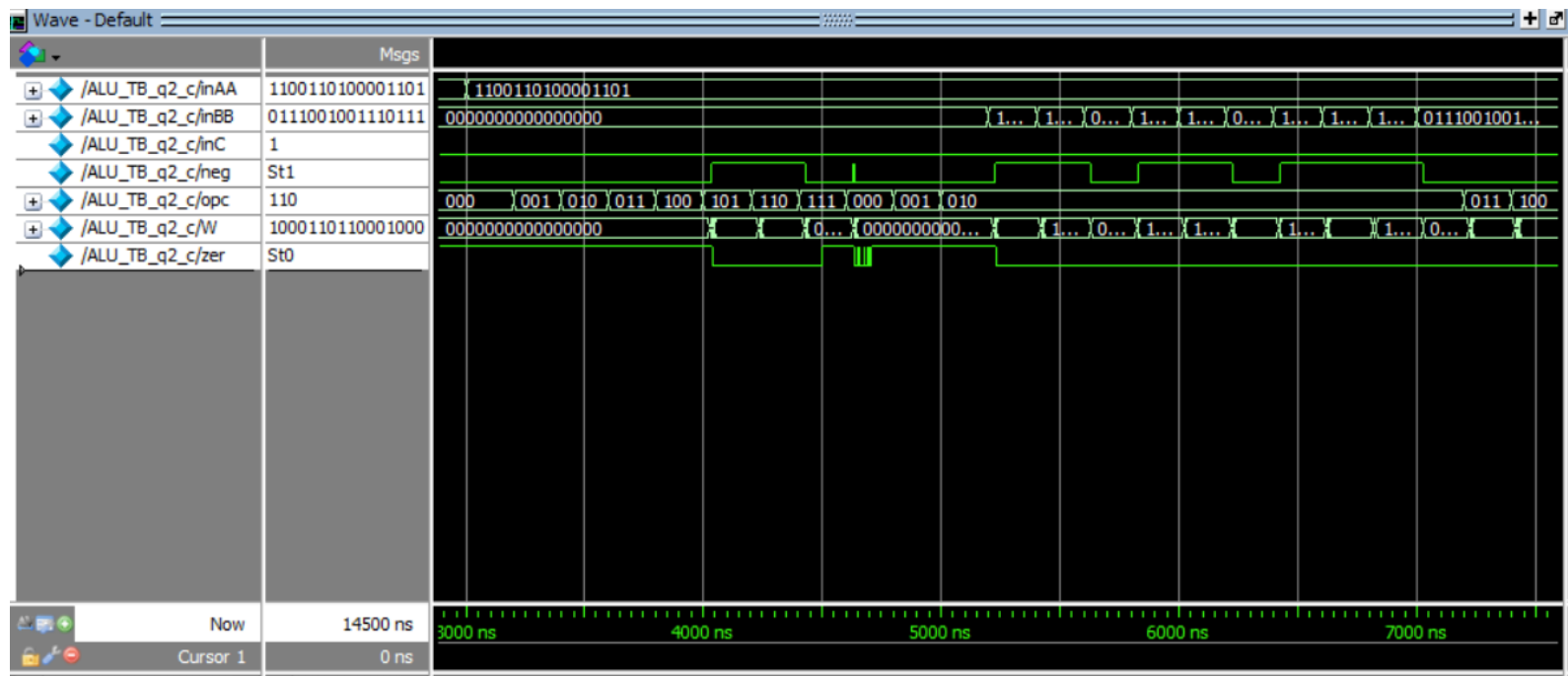


Yosys:

```
4.1.2. Re-integrating ABC results.
ABC RESULTS:      NAND cells:      120
ABC RESULTS:      NOR cells:       158
ABC RESULTS:      NOT cells:        49
ABC RESULTS:      internal signals: 186
ABC RESULTS:      input signals:    36
ABC RESULTS:      output signals:   17
Removing temp directory.
```



2-c



2-d

```
VSIM 10> simstats
# Memory Statistics
#   mem: size after elab (VSZ)                142.22 Mb
#   mem: size during sim (VSZ)               173.98 Mb
# Elaboration Time
#   elab: wall time                          0.31 s
#   elab: cpu time                          0.17 s
# Simulation Time
#   sim: wall time                          1.32 s
#   sim: cpu time                          1.28 s
# Tcl Command Time
#   cmd: wall time                         18.66 s
#   cmd: cpu time                          0.20 s
# Total Time
#   total: wall time                       20.28 s
#   total: cpu time                        1.66 s
#
```

Figure 3 pre

```
VSIM 21> simstats
# Memory Statistics
#   mem: size after elab (VSZ)                142.30 Mb
#   mem: size during sim (VSZ)               173.93 Mb
# Elaboration Time
#   elab: wall time                          0.29 s
#   elab: cpu time                          0.17 s
# Simulation Time
#   sim: wall time                          1.34 s
#   sim: cpu time                          1.31 s
# Tcl Command Time
#   cmd: wall time                         14.25 s
#   cmd: cpu time                          0.27 s
# Total Time
#   total: wall time                       15.88 s
#   total: cpu time                        1.75 s
#
```

Figure 4 post

همانند سوال یک:

در بخش a چون با `always` و `case` داشتیم از دیلی و سخت افزار بی خبر بودیم پس نمی توان از این لحاظ با بخش سی مقایسه کرد همچنین با توجه به حرف بالا و اینکه بخش سی دیلی در خود داشت جواب ها با اینکه یکسان است اما برا خروجی یوسیس مدت زمان بیشتری طول می کشد تا نمایش داده شود.

تفاوت با سوال یک:

چون تعداد cell ها خیلی کمتر است این دفعه زمان کمتری طول می کشد (ساختار بهینه)

در سوال یک ما از `always` استفاده کردیم و بعد با یوسیس گسترشش دادیم و با نند و نور ساختیم اما مشکلی داشت این بود که تعداد بیش از حد گیت به کار برده بود و بعضی جا ها می توانست با تعداد کمتری گیت آن را درست کند  
به همین دلیل در سوال دو تا جایی می شد که تونستیم اشتراکات رو بیرون آوردیم و تعداد گیت ها کم شد و نتیجتاً دلیلی هم کمتر شد.

\*از ۷ تا اپراتور استفاده می کنیم و از حالت `opc=7` استفاده نمی کنیم به این دلیل که اگر موقعی خروجی `w` مهم نبود بتوان از اون استفاده کرد