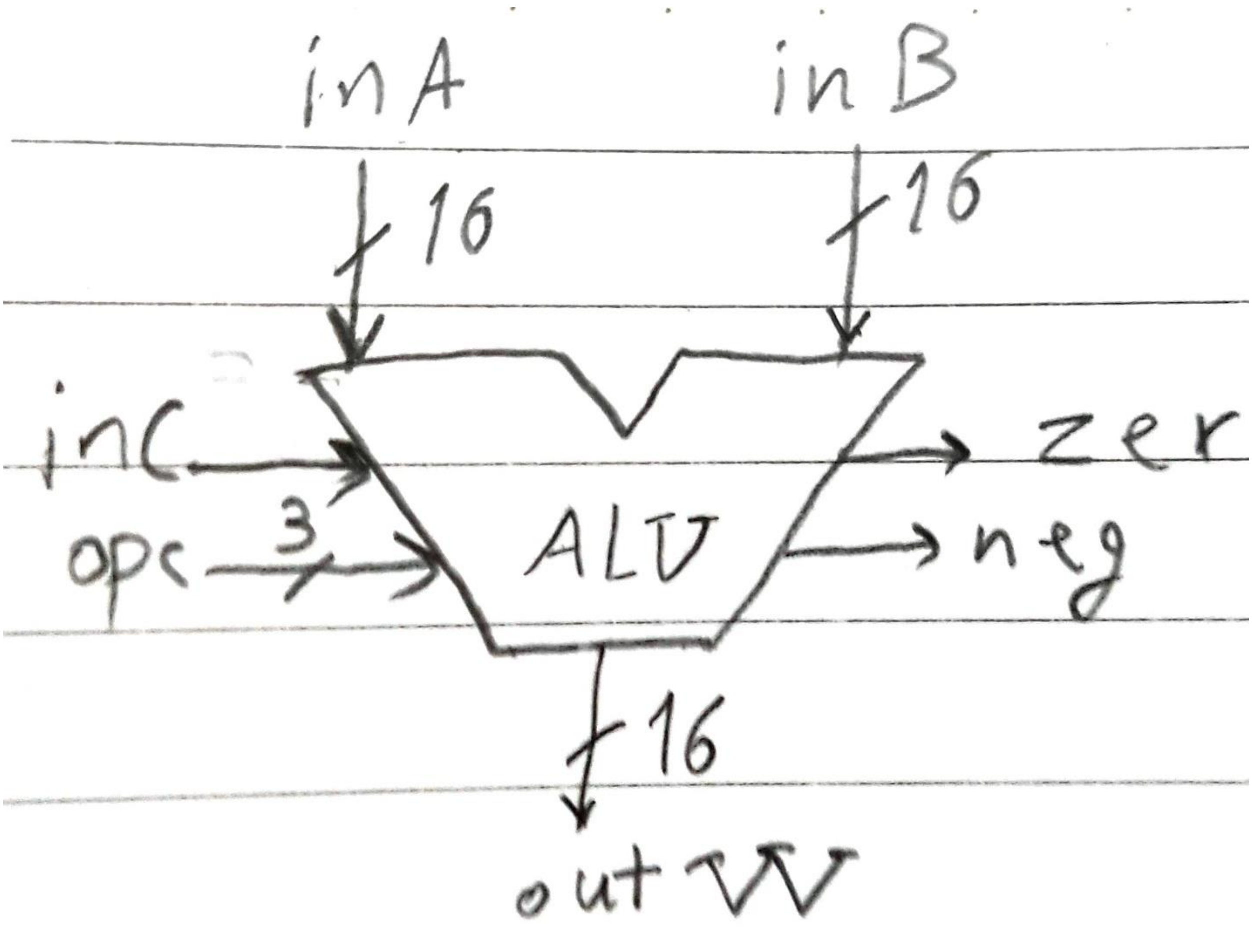
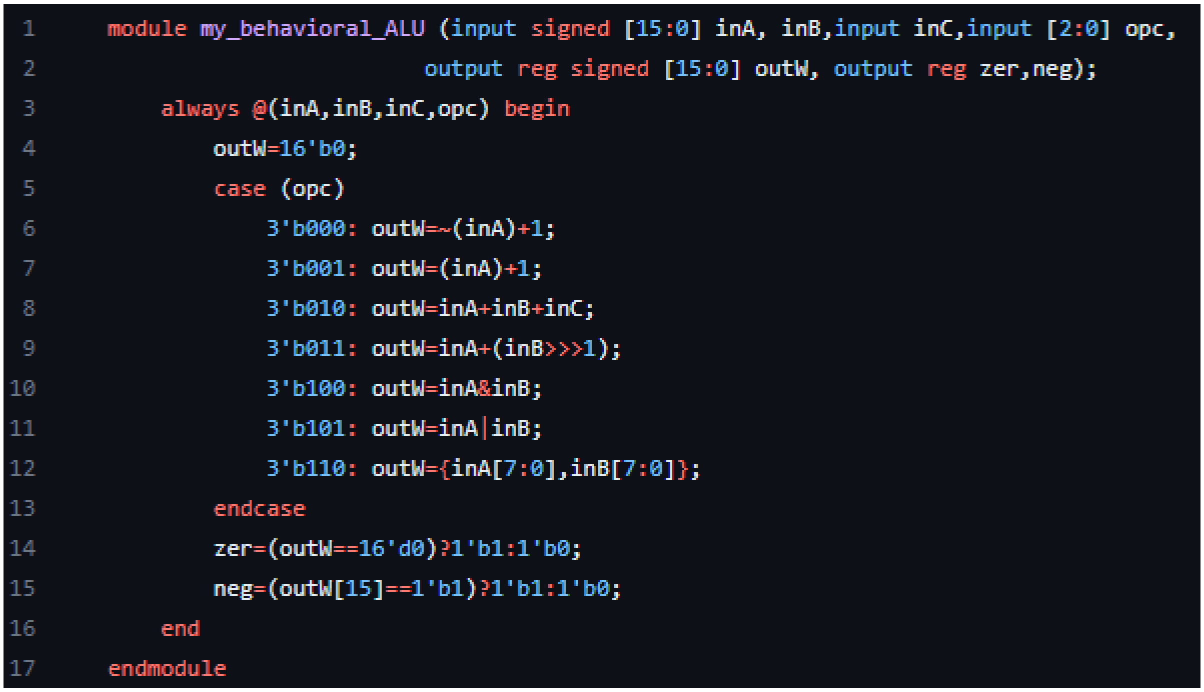
Report for computer assignment #3

Babak Hosseini Mohtasham 810101408

1.

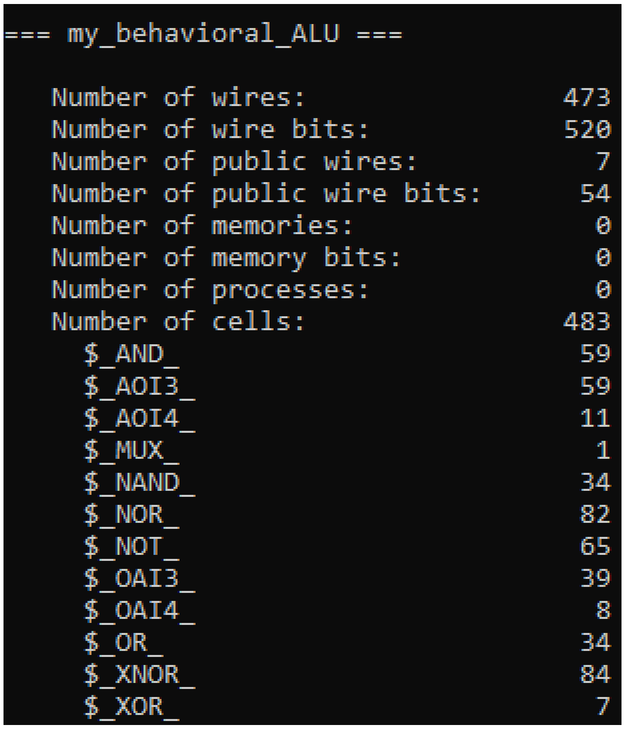
a)





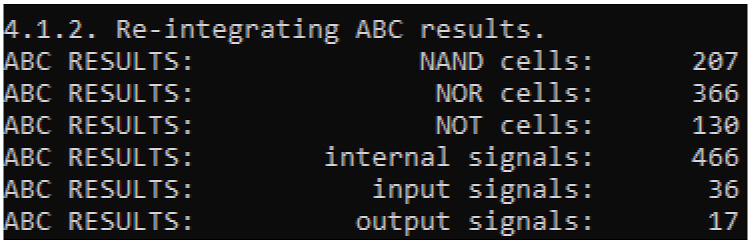
```
module my_behavioral_ALU (input signed [15:0] inA, inB,input inC,input [2:0] opc,
                        output reg signed [15:0] outW, output reg zer,neg);
    always @(inA,inB,inC,opc) begin
        outW=16'b0;
        case (opc)
            3'b000: outW=~(inA)+1;
            3'b001: outW=(inA)+1;
            3'b010: outW=inA+inB+inC;
            3'b011: outW=inA+(inB>>>1);
            3'b100: outW=inA&inB;
            3'b101: outW=inA|inB;
            3'b110: outW={inA[7:0],inB[7:0]};
        endcase
        zer=(outW==16'd0)?1'b1:1'b0;
        neg=(outW[15]==1'b1)?1'b1:1'b0;
    end
endmodule
```

c)



```
=== my_behavioral_ALU ===

   Number of wires:                473
   Number of wire bits:            520
   Number of public wires:           7
   Number of public wire bits:      54
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                483
     $_AND_                         59
     $_AOI3_                        59
     $_AOI4_                        11
     $_MUX_                          1
     $_NAND_                        34
     $_NOR_                         82
     $_NOT_                         65
     $_OAI3_                        39
     $_OAI4_                         8
     $_OR_                          34
     $_XNOR_                        84
     $_XOR_                          7
```

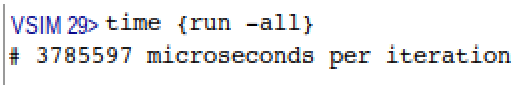
yosis’ internal library



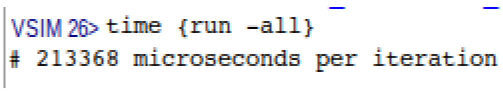
```
4.1.2. Re-integrating ABC results.
ABC RESULTS:               NAND cells:      207
ABC RESULTS:                NOR cells:      366
ABC RESULTS:                NOT cells:      130
ABC RESULTS:         internal signals:      466
ABC RESULTS:            input signals:       36
ABC RESULTS:           output signals:       17
```

mycells.lib

Number of cells=703

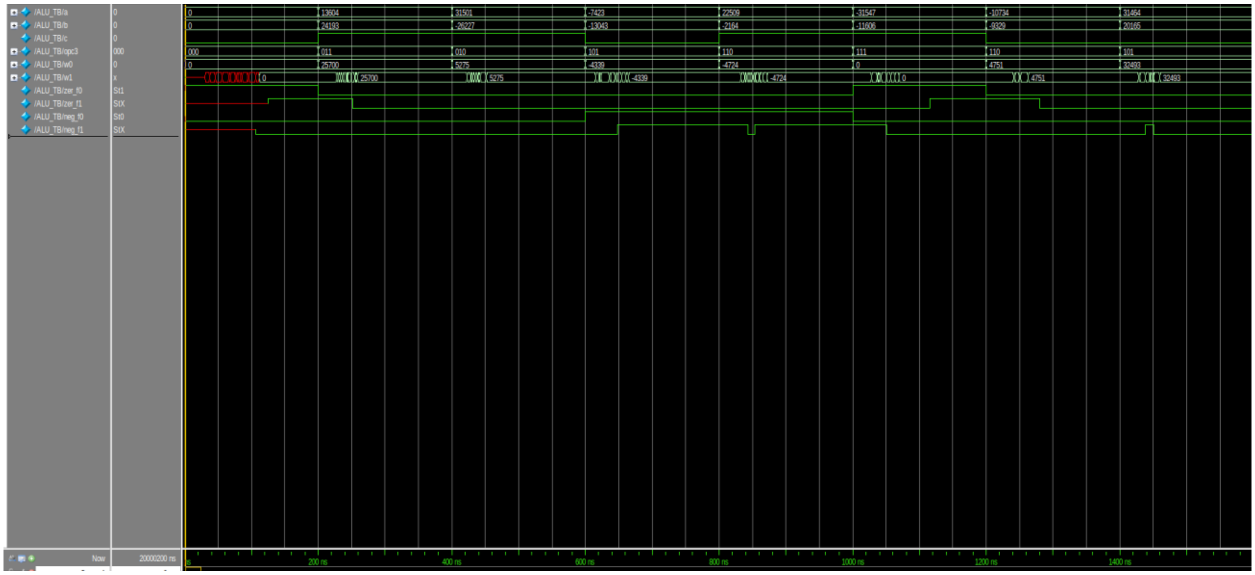


```
VSIM 29> time {run -all}
# 3785597 microseconds per iteration
```



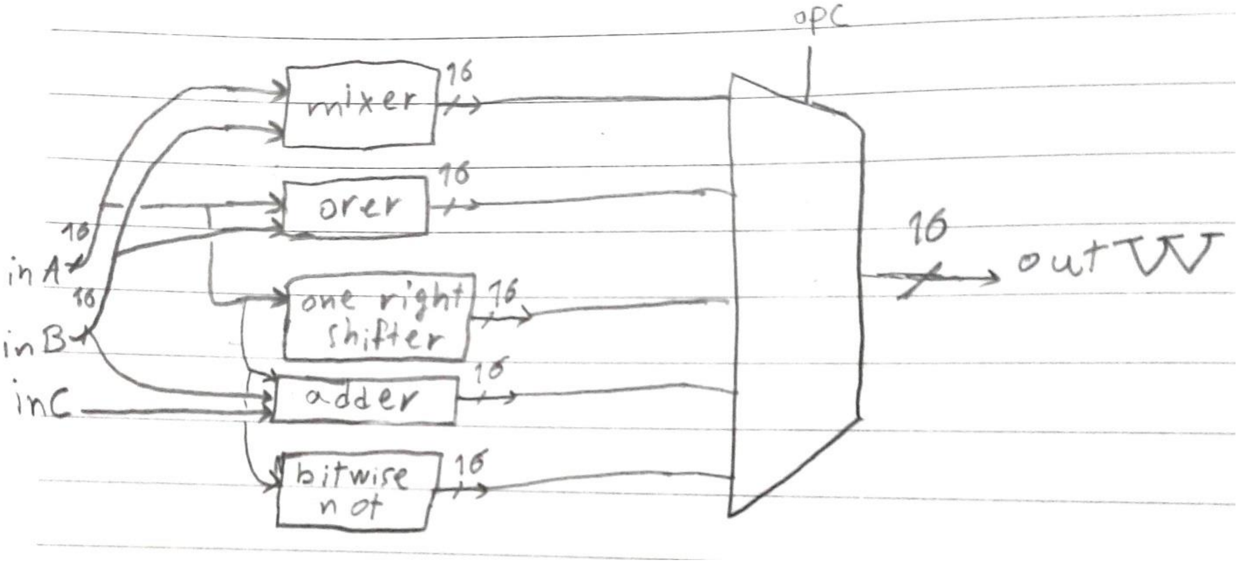
```
VSIM 26> time {run -all}
# 213368 microseconds per iteration
```

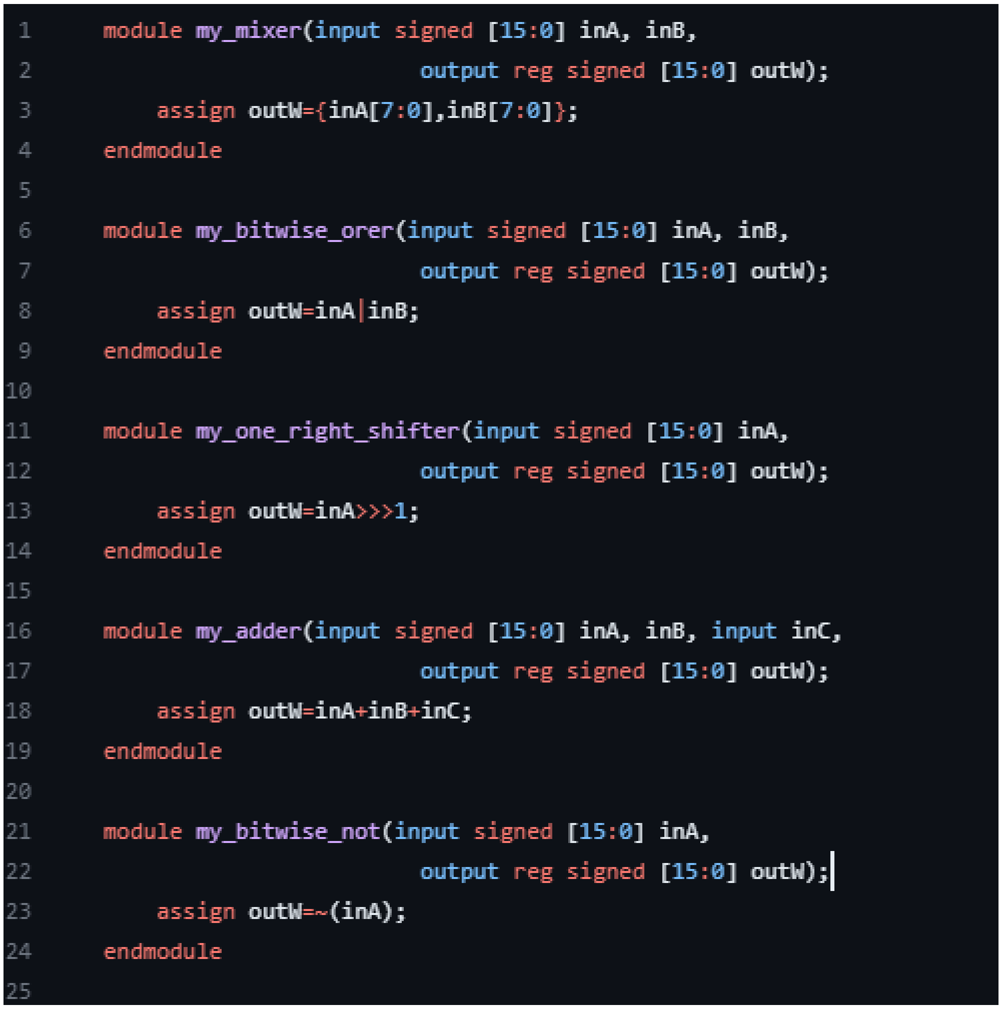
Pre-synthesis Post-synthesis



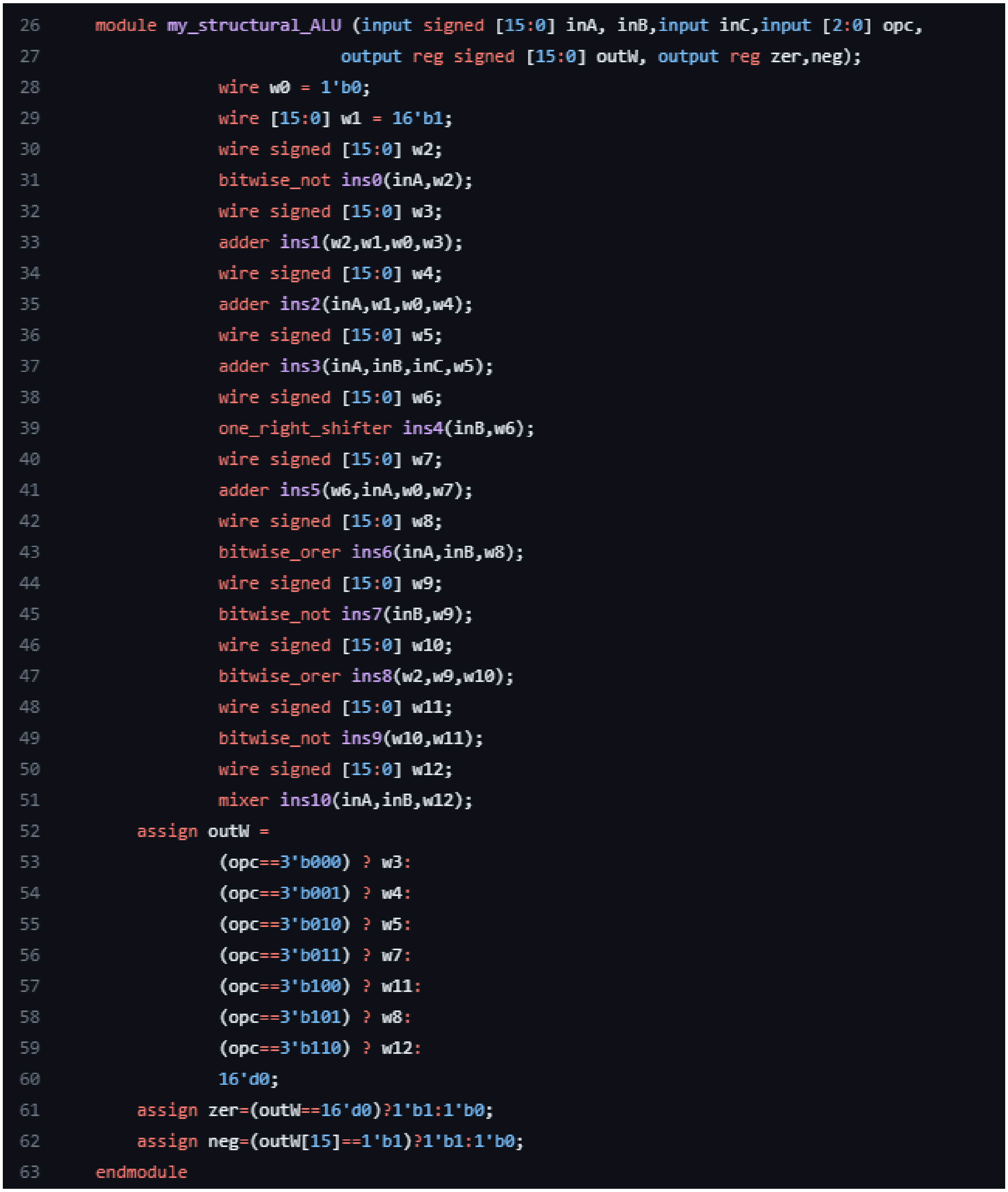
2.

a)



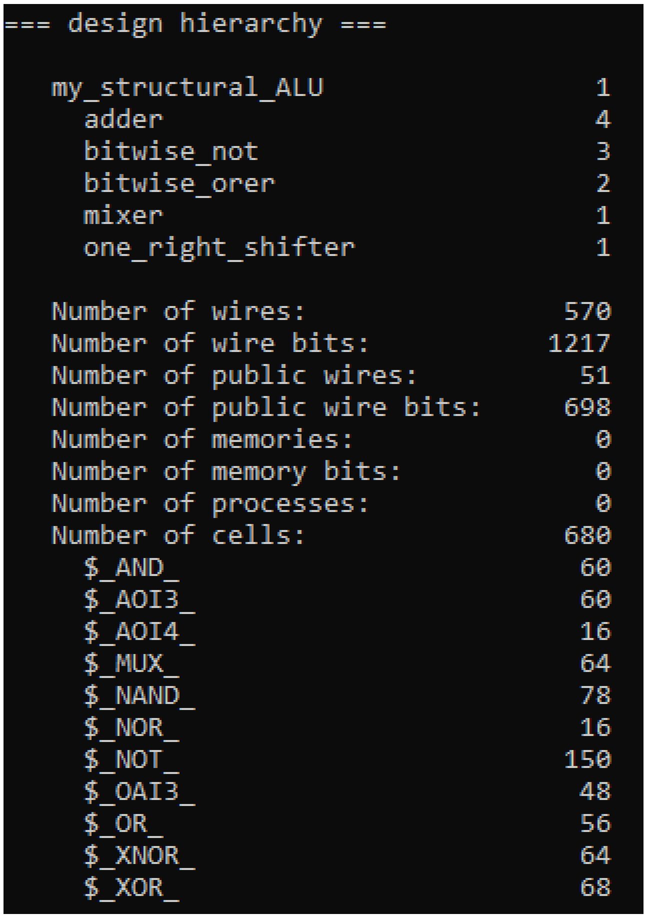


```
module my_mixer(input signed [15:0] inA, inB,
                        output reg signed [15:0] outW);
    assign outW={inA[7:0],inB[7:0]};
endmodule

module my_bitwise_orer(input signed [15:0] inA, inB,
                        output reg signed [15:0] outW);
    assign outW=inA|inB;
endmodule

module my_one_right_shifter(input signed [15:0] inA,
                        output reg signed [15:0] outW);
    assign outW=inA>>>1;
endmodule

module my_adder(input signed [15:0] inA, inB, input inC,
                        output reg signed [15:0] outW);
    assign outW=inA+inB+inC;
endmodule

module my_bitwise_not(input signed [15:0] inA,
                        output reg signed [15:0] outW);
    assign outW=~(inA);
endmodule
```



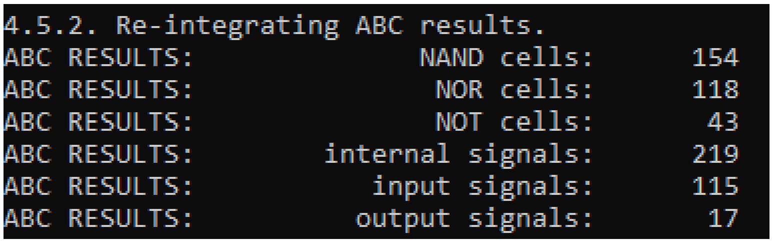
```
module my_structural_ALU (input signed [15:0] inA, inB,input inC,input [2:0] opc,
                        output reg signed [15:0] outW, output reg zer,neg);
            wire w0 = 1'b0;
            wire [15:0] w1 = 16'b1;
            wire signed [15:0] w2;
            bitwise_not ins0(inA,w2);
            wire signed [15:0] w3;
            adder ins1(w2,w1,w0,w3);
            wire signed [15:0] w4;
            adder ins2(inA,w1,w0,w4);
            wire signed [15:0] w5;
            adder ins3(inA,inB,inC,w5);
            wire signed [15:0] w6;
            one_right_shifter ins4(inB,w6);
            wire signed [15:0] w7;
            adder ins5(w6,inA,w0,w7);
            wire signed [15:0] w8;
            bitwise_orer ins6(inA,inB,w8);
            wire signed [15:0] w9;
            bitwise_not ins7(inB,w9);
            wire signed [15:0] w10;
            bitwise_orer ins8(w2,w9,w10);
            wire signed [15:0] w11;
            bitwise_not ins9(w10,w11);
            wire signed [15:0] w12;
            mixer ins10(inA,inB,w12);
    assign outW =
            (opc==3'b000) ? w3:
            (opc==3'b001) ? w4:
            (opc==3'b010) ? w5:
            (opc==3'b011) ? w7:
            (opc==3'b100) ? w11:
            (opc==3'b101) ? w8:
            (opc==3'b110) ? w12:
            16'd0;
    assign zer=(outW==16'd0)?1'b1:1'b0;
    assign neg=(outW[15]==1'b1)?1'b1:1'b0;
endmodule
```

c)

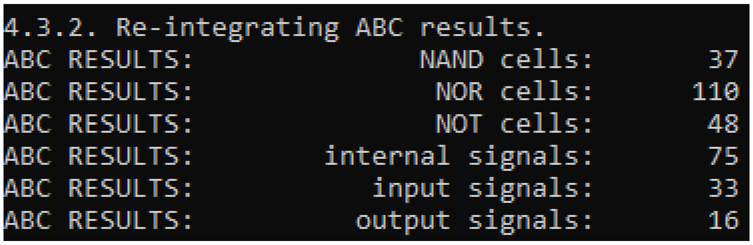


```
=== design hierarchy ===

   my_structural_ALU                 1
     adder                           4
     bitwise_not                     3
     bitwise_orer                    2
     mixer                           1
     one_right_shifter               1

   Number of wires:                570
   Number of wire bits:           1217
   Number of public wires:          51
   Number of public wire bits:     698
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                680
     $_AND_                         60
     $_AOI3_                        60
     $_AOI4_                        16
     $_MUX_                         64
     $_NAND_                        78
     $_NOR_                         16
     $_NOT_                        150
     $_OAI3_                        48
     $_OR_                          56
     $_XNOR_                        64
     $_XOR_                         68
```

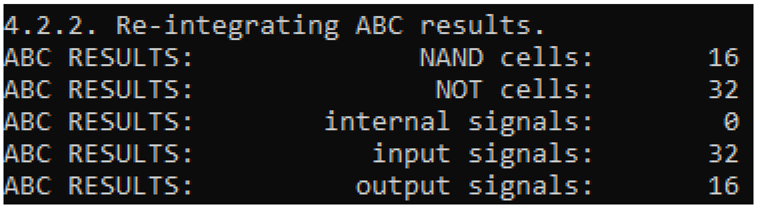
yosis' internal library



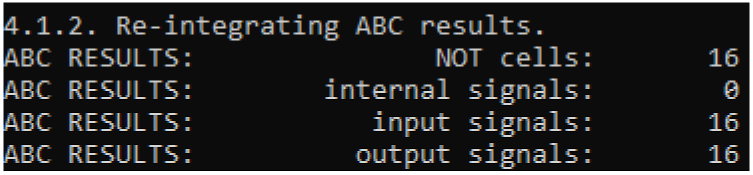
```
4.5.2. Re-integrating ABC results.
ABC RESULTS:              NAND cells:      154
ABC RESULTS:               NOR cells:      118
ABC RESULTS:               NOT cells:       43
ABC RESULTS:        internal signals:      219
ABC RESULTS:           input signals:      115
ABC RESULTS:          output signals:       17
```



```
4.3.2. Re-integrating ABC results.
ABC RESULTS:              NAND cells:       37
ABC RESULTS:               NOR cells:      110
ABC RESULTS:               NOT cells:       48
ABC RESULTS:        internal signals:       75
ABC RESULTS:           input signals:       33
ABC RESULTS:          output signals:       16
```



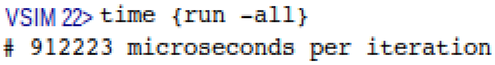
```
4.2.2. Re-integrating ABC results.
ABC RESULTS:               NAND cells:       16
ABC RESULTS:                NOT cells:       32
ABC RESULTS:         internal signals:        0
ABC RESULTS:            input signals:       32
ABC RESULTS:           output signals:       16
```



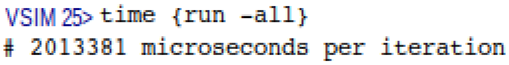
```
4.1.2. Re-integrating ABC results.
ABC RESULTS:                NOT cells:       16
ABC RESULTS:         internal signals:        0
ABC RESULTS:            input signals:       16
ABC RESULTS:           output signals:       16
```

mycells.lib

Number of cells=574

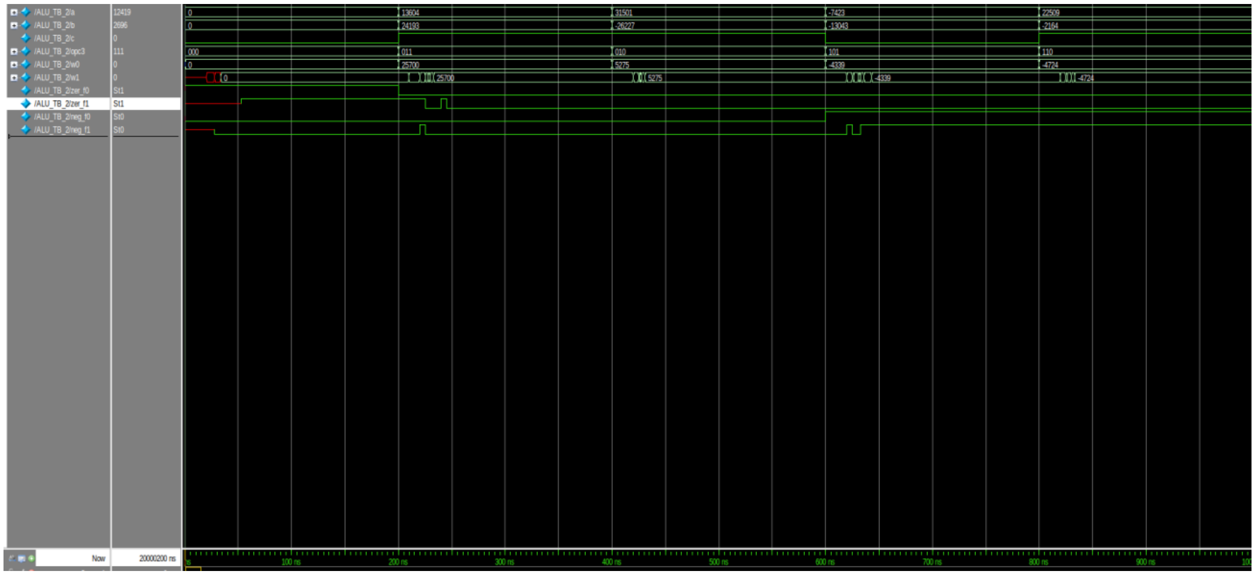


```
VSIM 22> time {run -all}
# 912223 microseconds per iteration
```



```
VSIM 25> time {run -all}
# 2013381 microseconds per iteration
```

Pre-synthesis   Post-synthesis



Yosis' internal library consists of many different components compared to the mycells.lib library which only has not, nand and nor. We can see the part a of q1 has 483 cells while part c has 703 cells. Also in q2 there are 680 cells for part a and 574 cells for part c. I think due to sharing several components (sharing adder for opc 0,1,2,3, sharing not for opc 0 and 5 and sharing or for opc 4 and 5) in q3 the are far fewer components used in q3 with mycells.lib than q1 mycells.lib, though yosis with its own library has used the least number of cells in q1.

Due to existence of time delays in mycells.lib library the waveform diagrams of part cs have delays unlike the assign statements used in part as.