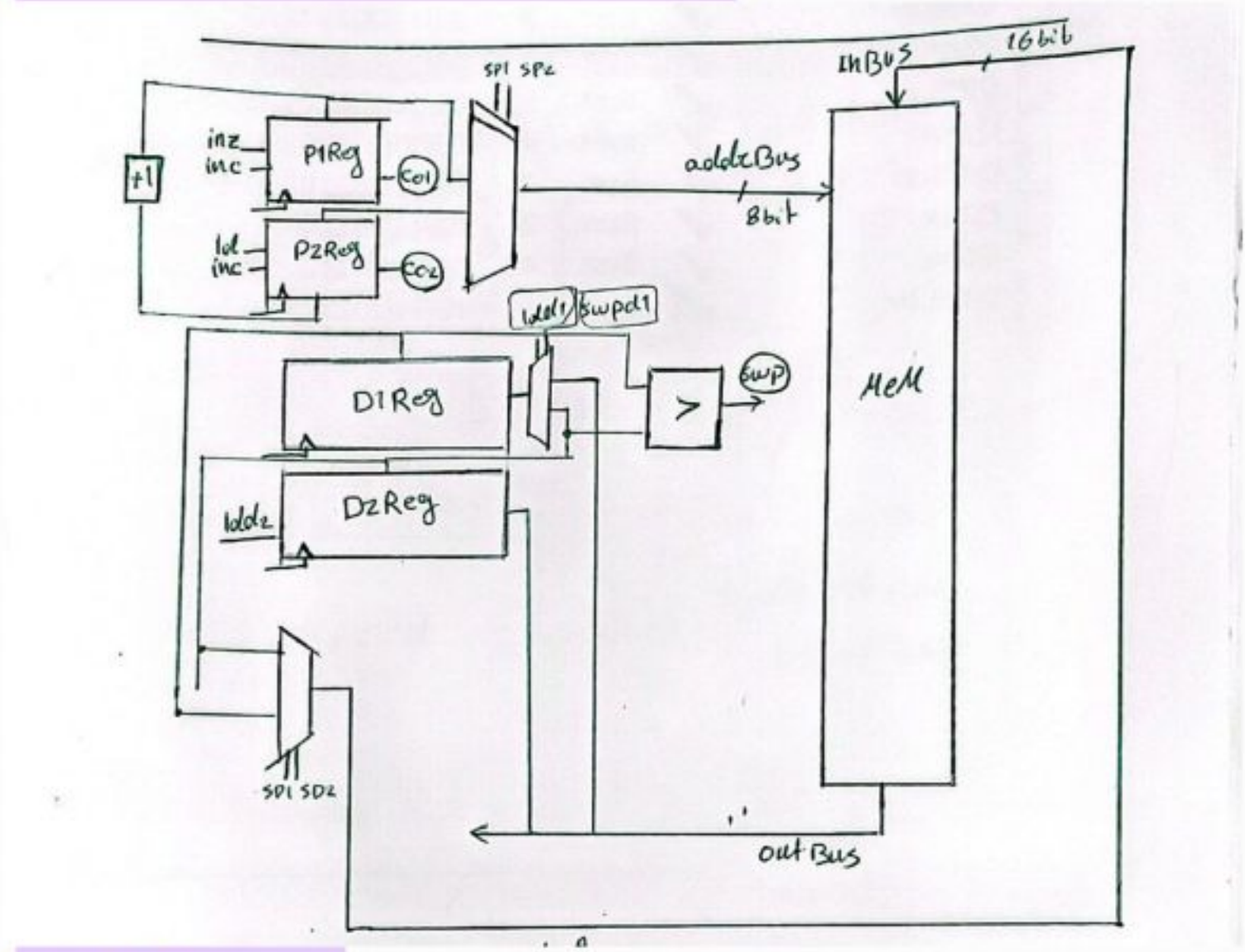
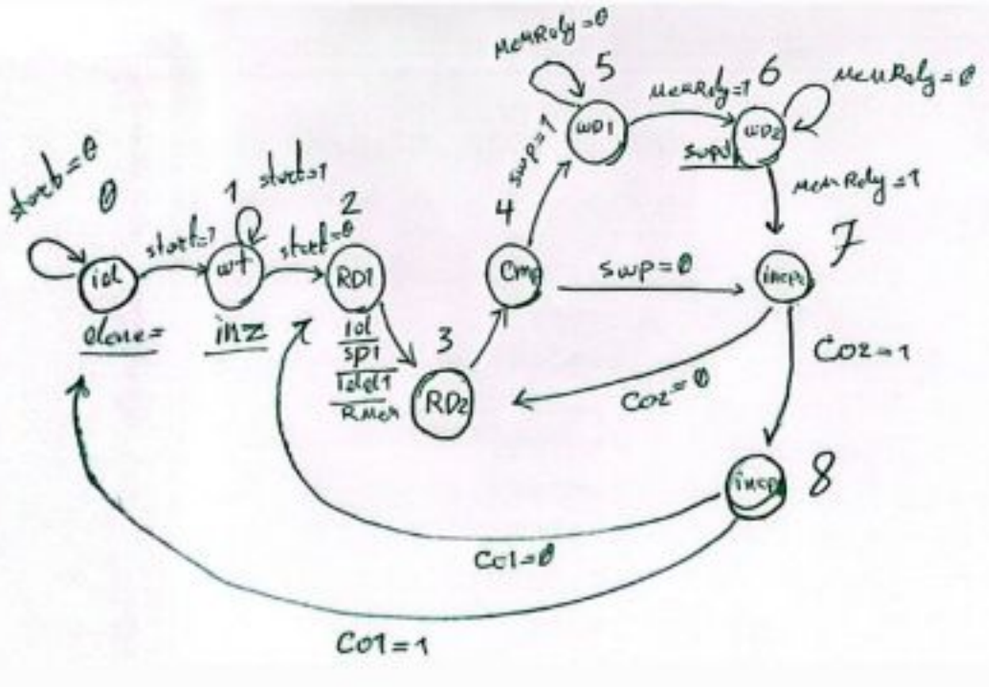
a. design

Combinational part design



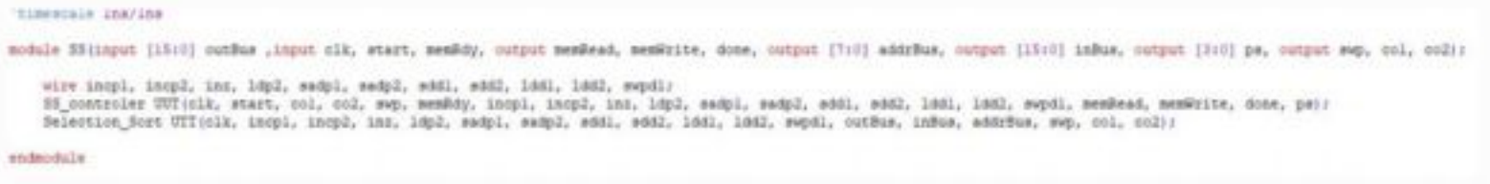
FSM design



b. verilog

Verilog Description of SS

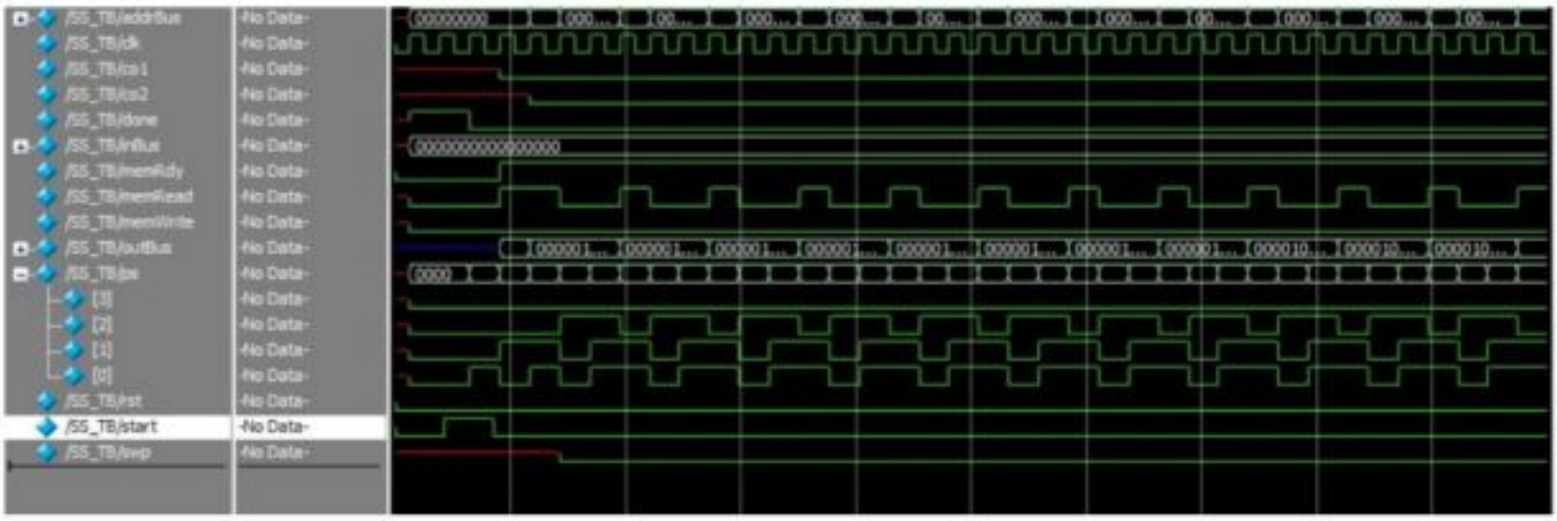
```
module Selection_Sort(input clk, incp1, incp2, inz, ldp2, sadp1,
    wire [7:0] ns_p1, ns_p2;
    reg [7:0] ps_p1, ps_p2;
    wire [15:0] ns_d1, ns_d2;
    reg [15:0] ps_d1, ps_d2;
    wire [7:0] p1_next, p2_next;

    always@(posedge clk) begin
        ps_d1<=ns_d1;
        ps_d2<=ns_d2;
        ps_p1<=ns_p1;
        ps_p2<=ns_p2;
    end

    assign {co1 , p1_next} =  ps_p1 + 8'd1;
    assign {co2 , p2_next} =  ps_p2 + 8'd1;
    assign ns_p1 = inz ? 8'd0 : incp1 ? p1_next : ps_p1;
    assign ns_p2 = ldp2 ? p1_next : incp2 ? p2_next : ps_p2;
    assign ns_d1 = ldd1 ? data_in : swpd1 ? ps_d2 : ps_d1;
    assign ns_d2 = ldd2 ? data_in : ps_d2;
    assign swp = (ps_d1 > ps_d2) ? 1'b1 : 1'b0;
    assign data_out = sdd1 ? ps_d1 : sdd2 ? ps_d2 : 16'd0;
    assign address = sadp1 ? ps_p1 : sadp2 ? ps_p2 : 16'd0;

endmodule
```



```
always@(ps, start, co1, co2, swp, memRdy) begin
    ns=4'd0;
        case(ps)
            0: ns = start ? ps + 4'd1 : ps;
            1: ns = start ? ps : ps + 4'd1;
            2: ns = memRdy ? ps + 4'd1 : ps;
            3: ns = memRdy ? ps + 4'd1 : ps;
            4: ns = swp ? ps + 4'd1 : 4'd7;
            5: ns = memRdy ? ps + 4'd1 : ps;
            6: ns = memRdy ? ps + 4'd1 : ps;
            7: ns = co2 ? ps + 4'd1 : 4'd3;
            8: ns = co1 ?  4'd0 : 4'd2;
            default: ns = 4'd0;
        endcase
end

always@(posedge clk) begin

    ps<=ns;
end

assign done = (ps==4'd0) ? 1'b1 : 1'b0;
assign inz = (ps==4'd1) ? 1'b1 : 1'b0;
assign ldp2 = (ps==4'd2) ? 1'b1 : 1'b0;
assign sadp1 = (ps==4'd2 || ps==4'd6) ? 1'b1 : 1'b0;
assign ldd1 = (ps==4'd2) ? 1'b1 : 1'b0;
assign read_mem = (ps==4'd2 || ps==4'd3) ? 1'b1 : 1'b0;
assign ldd2 = (ps==4'd3) ? 1'b1 : 1'b0;
assign sadp2 = (ps==4'd3 || ps==4'd5) ? 1'b1 : 1'b0;
assign write_mem = (ps==4'd5 || ps==4'd6) ? 1'b1 : 1'b0;
assign sdd1 = (ps==4'd5) ? 1'b1 : 1'b0;
assign sdd2 = (ps==4'd6) ? 1'b1 : 1'b0;
assign swpd1 = (ps==4'd6) ? 1'b1 : 1'b0;
assign incp2 = (ps==4'd7) ? 1'b1 : 1'b0;
assign incp1 = (ps==4'd8) ? 1'b1 : 1'b0;
```

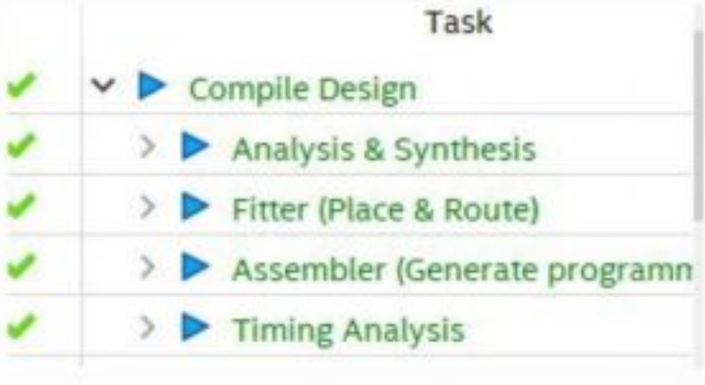
Test Bench

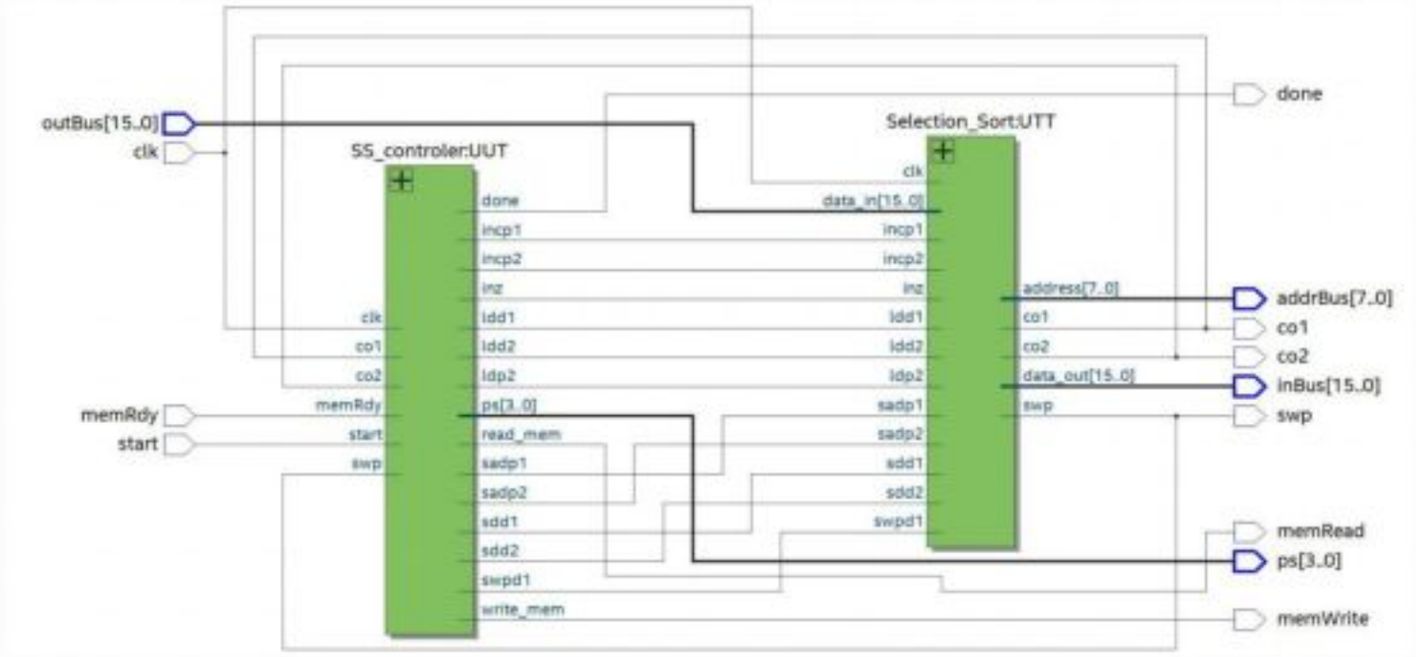
```
timescale 1ns/1ns
module SS_TB();

  reg clk = 0;
  reg start = 0;
  wire rst = 0;
  wire memRead, memWrite, memRdy, done, swp;
  wire [7:0] addrBus;
  wire [15:0] inBus;
  wire [15:0] outBus;
  wire [3:0] ps;

  SS UTT(outBus , clk, start, memRdy, memRead, memWrite, done, addrBus, inBus, ps, swp, co1, co2);
  Memory UUT(clk, rst, memRead, memWrite, addrBus, inBus, memRdy, outBus);

  initial repeat(1000) # 13 clk = ~clk;
    initial begin
            #43 start = 1'b1;
            #43 start = 1'b0;
            #10000 $stop;
    end

endmodule
```

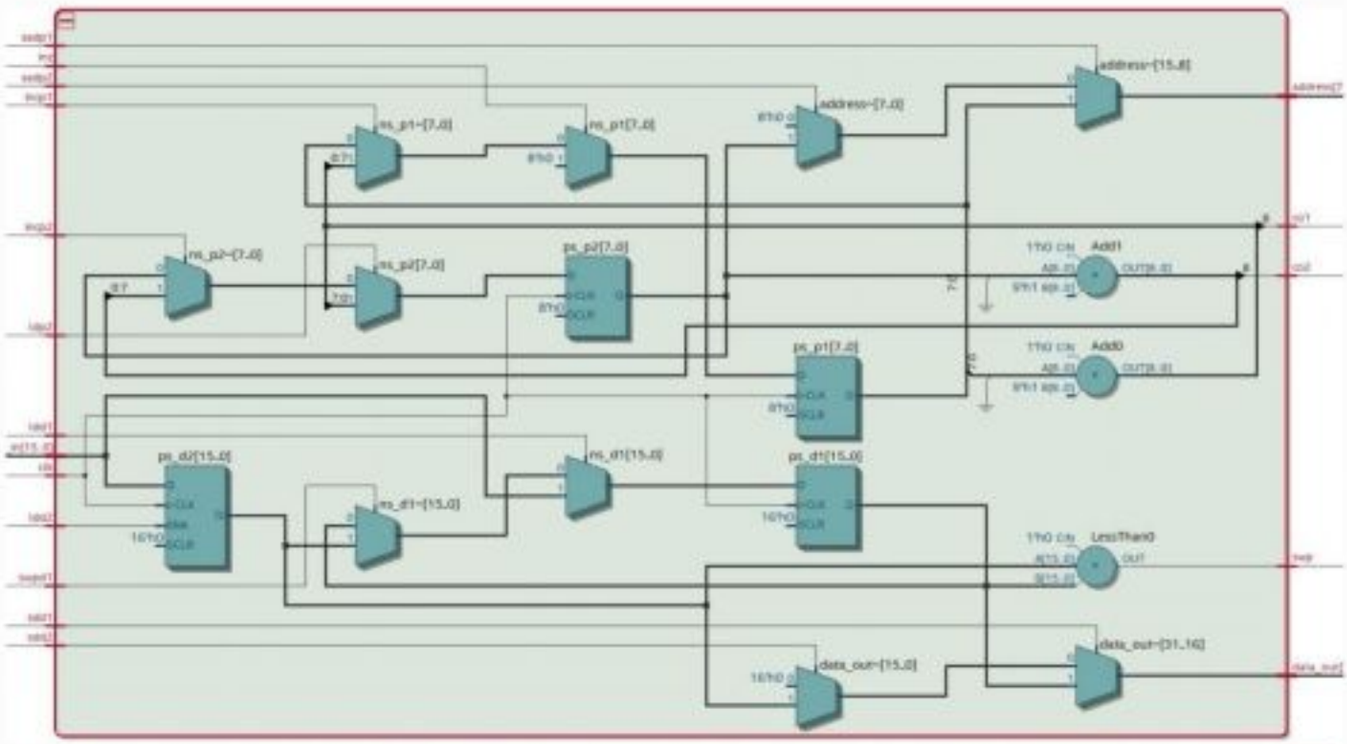


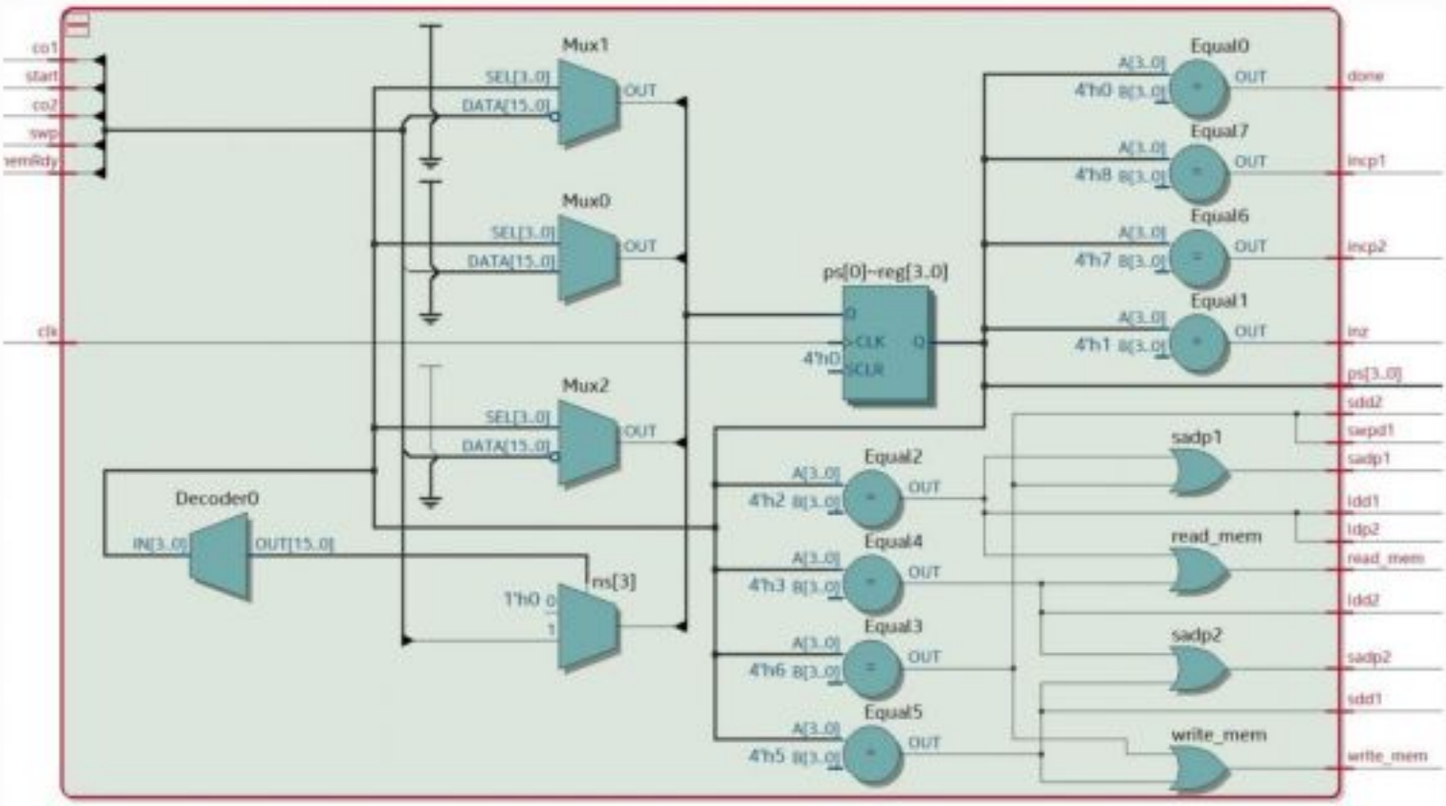
c. Quartus synthesize

RTL viewer

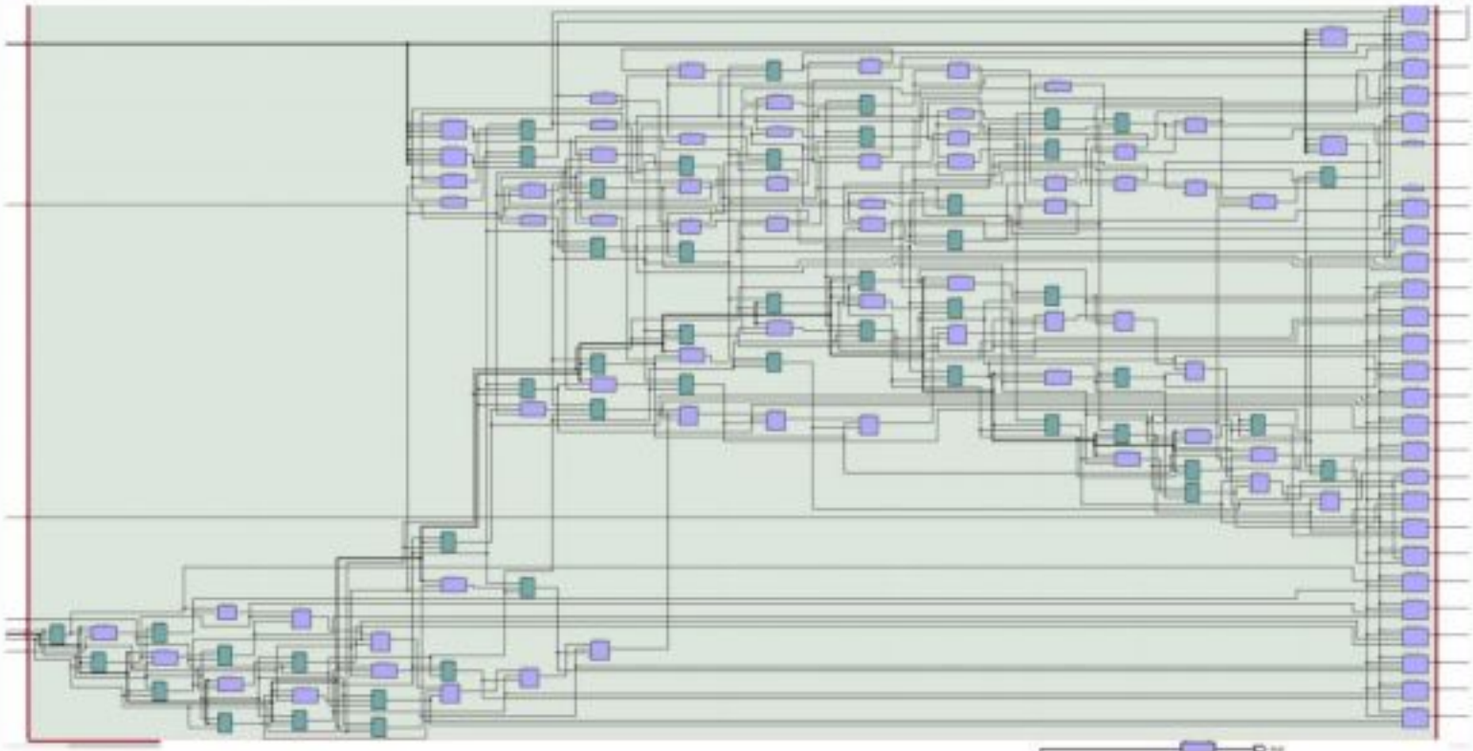


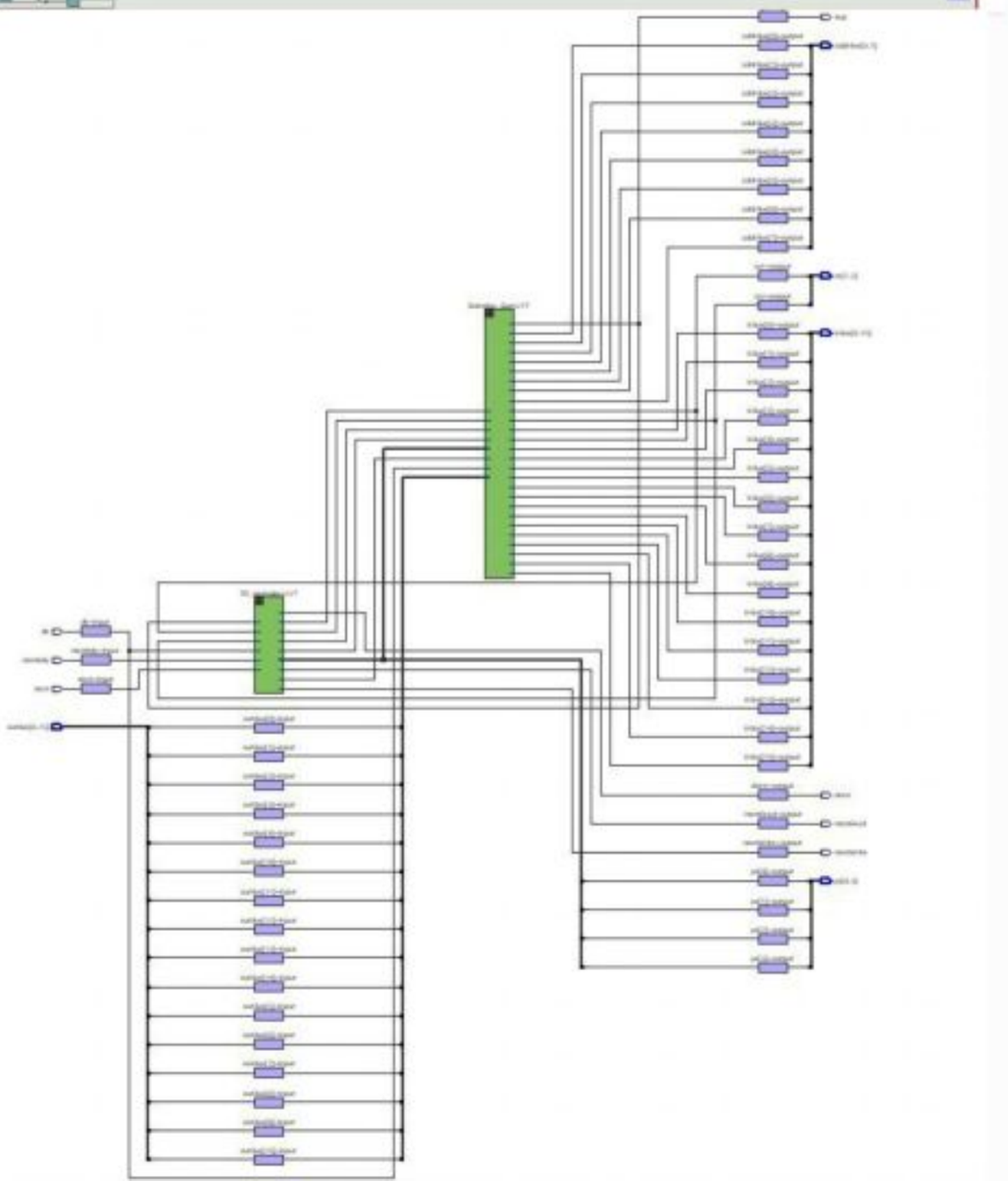

outBus[15..0]
clk
SS_controler:UUT
done
incp1
incp2
inz
clk
idd1
co1
idd2
co2
idp2
memRdy
ps[3..0]
start
read_mem
swp
sadp1
sadp2
sdd1
sdd2
swpd1
write_mem
Selection_Sort:UTT
clk
data_in[15..0]
incp1
incp2
inz
address[7..0]
idd1
co1
idd2
co2
idp2
data_out[15..0]
sadp1
swp
sadp2
sdd1
sdd2
swpd1
done
addrBus[7..0]
co1
co2
inBus[15..0]
swp
memRead
ps[3..0]
memWrite
memRdy
start

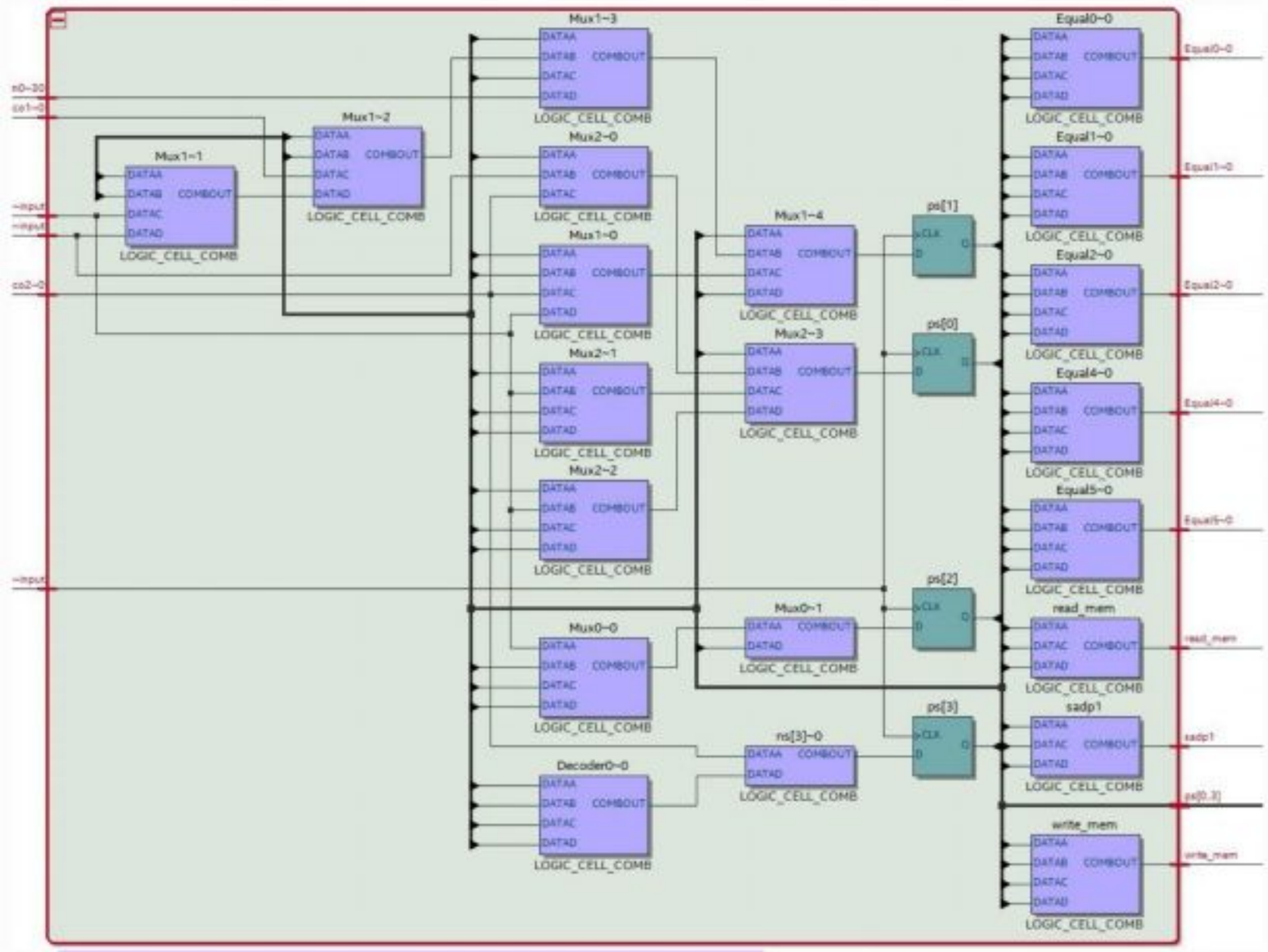

address~[15..8]
address~[7..0]
ns_p1~[7..0]
ns_p1[7..0]
ns_p2~[7..0]
ns_p2[7..0]
ps_p2[7..0]
ps_p1[7..0]
Add1
Add0
ps_d2[15..0]
ns_d1~[15..0]
ns_d1[15..0]
ps_d1[15..0]
LessThan0
data_out~[15..0]
data_out~[31..16]


co1
start
co2
swp
memRdy
clk
Mux1
Mux0
Mux2
SEL[3..0]
DATA[15..0]
OUT
ps[0]~reg[3..0]
4'h0
CLK
SCLR
Q
Decoder0
IN[3..0]
OUT[15..0]
ns[3]
1'h0
Equal0
Equal7
Equal6
Equal1
Equal2
Equal4
Equal3
Equal5
A[3..0]
B[3..0]
4'h0
4'h8
4'h7
4'h1
4'h2
4'h3
4'h6
4'h5
done
incp1
incp2
inz
ps[3..0]
sdd2
swpd1
sadp1
idd1
idp2
read_mem
idd2
sadp2
sdd1
write_mem

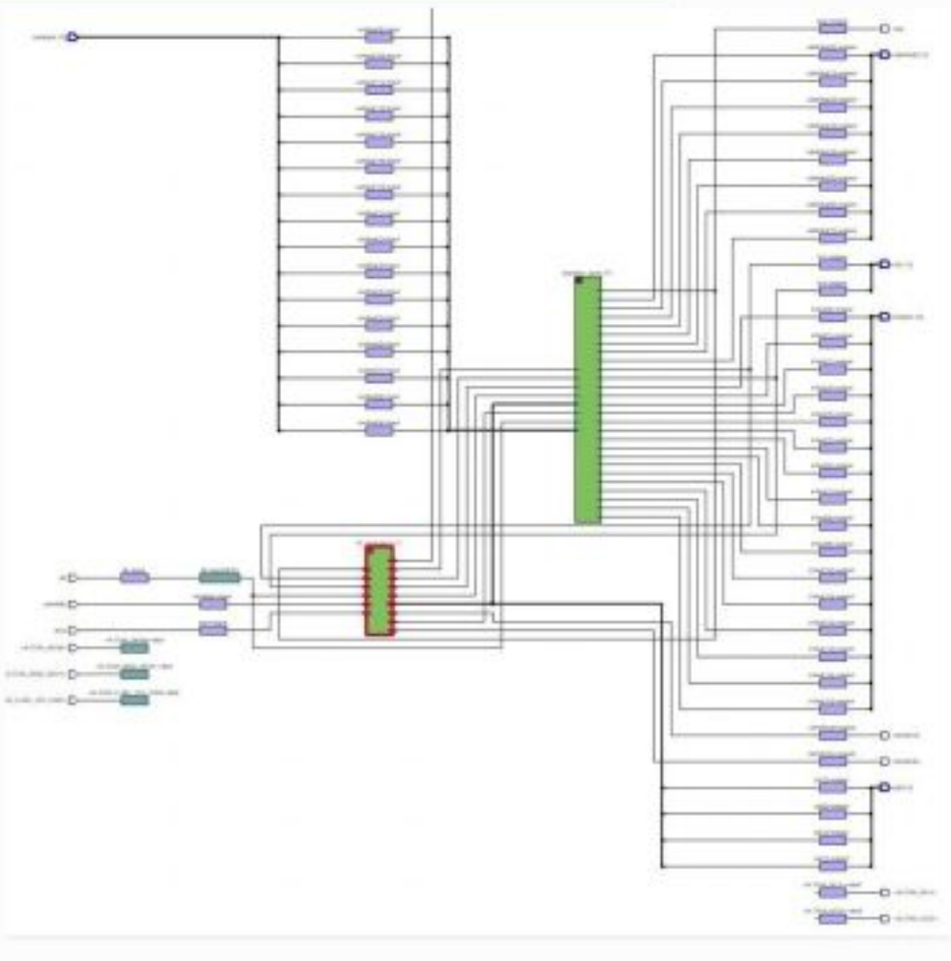
Map Technology viwer PM



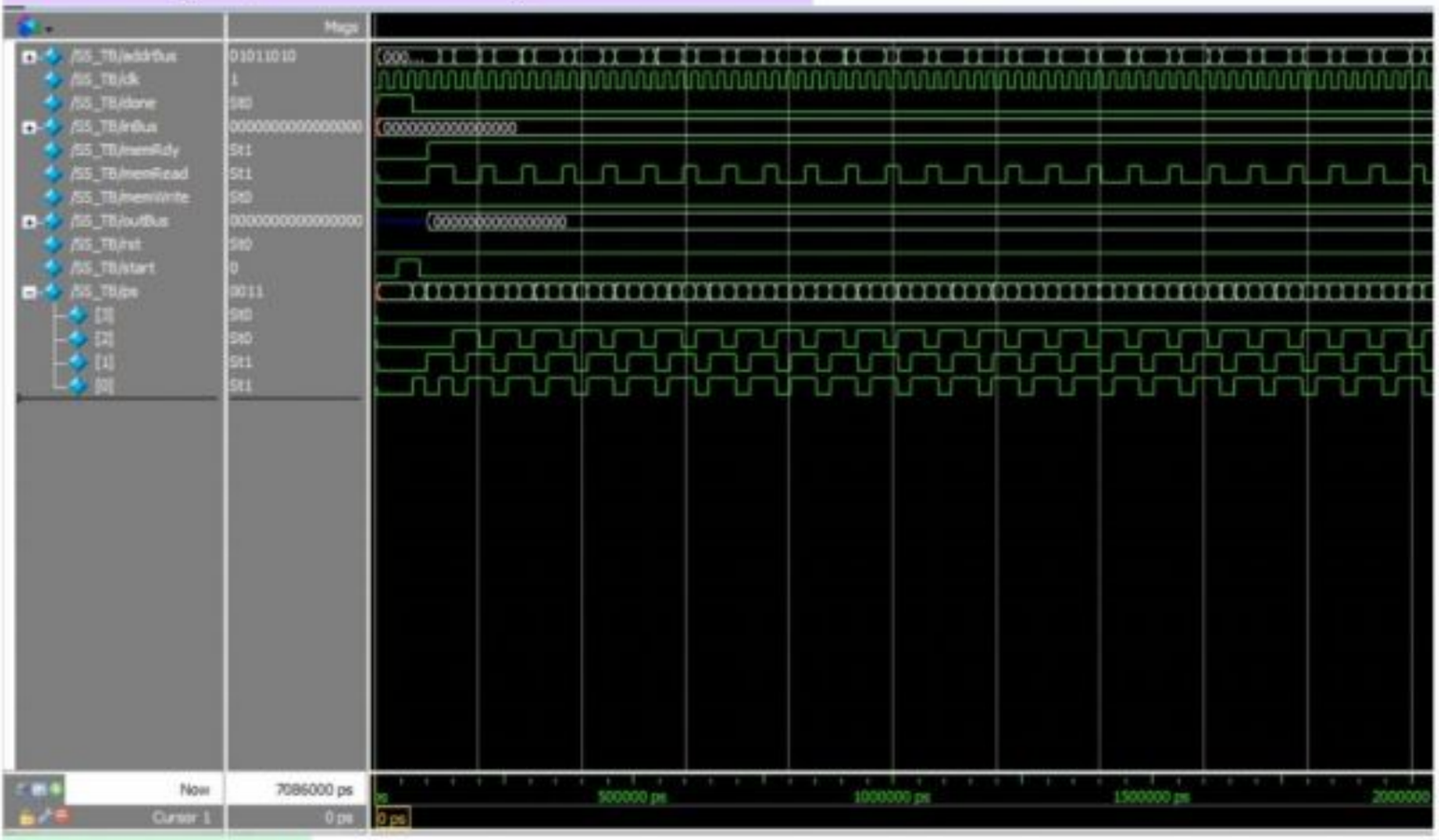




Map technology viewer PF

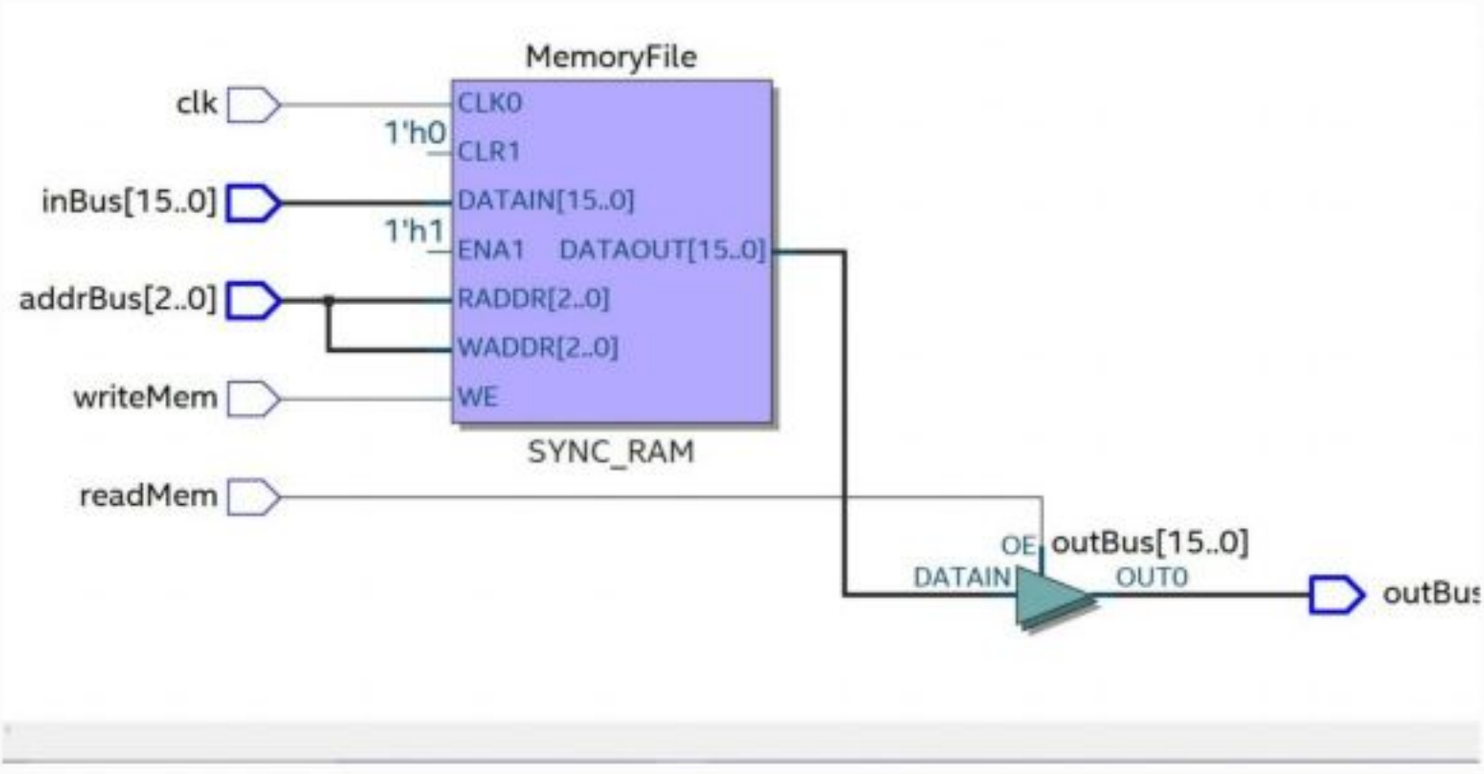


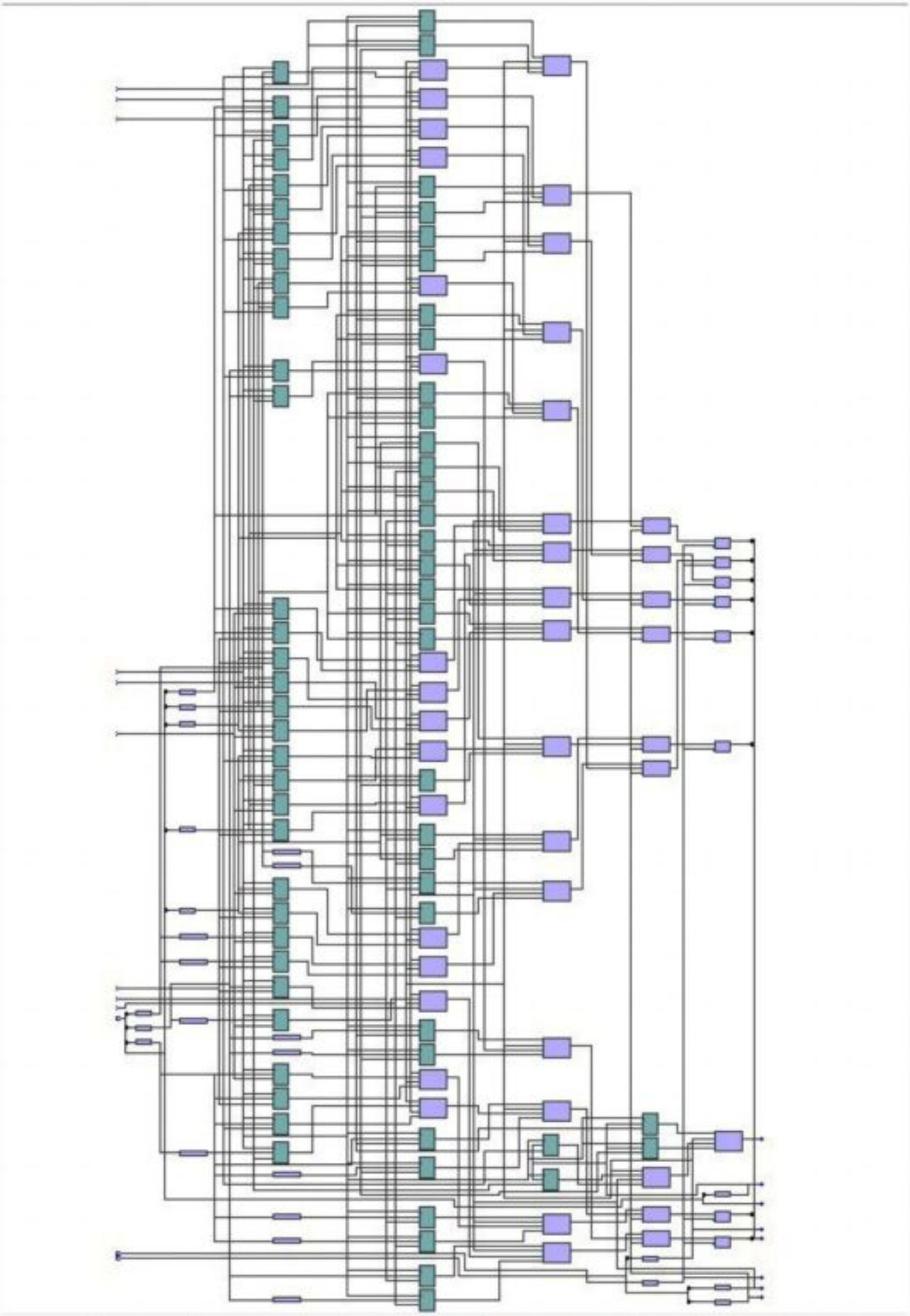
Justify quartus synthesize

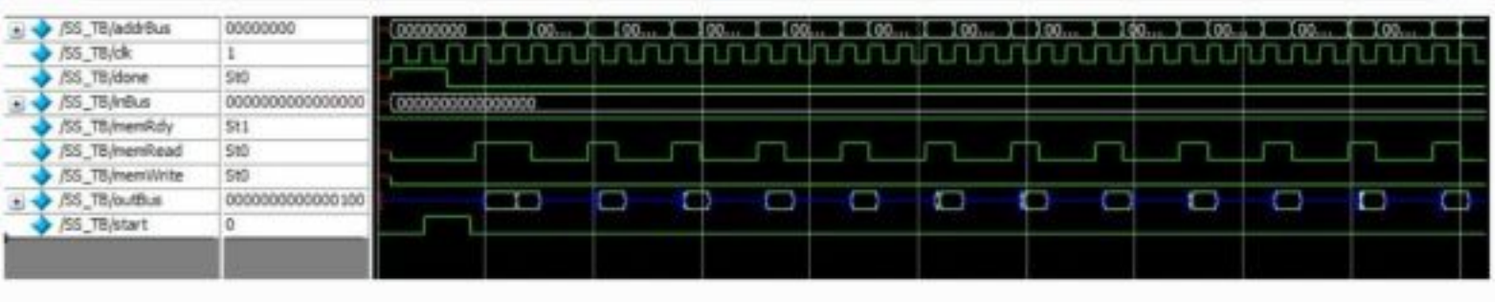


Extra part

Synthesize in LE mode

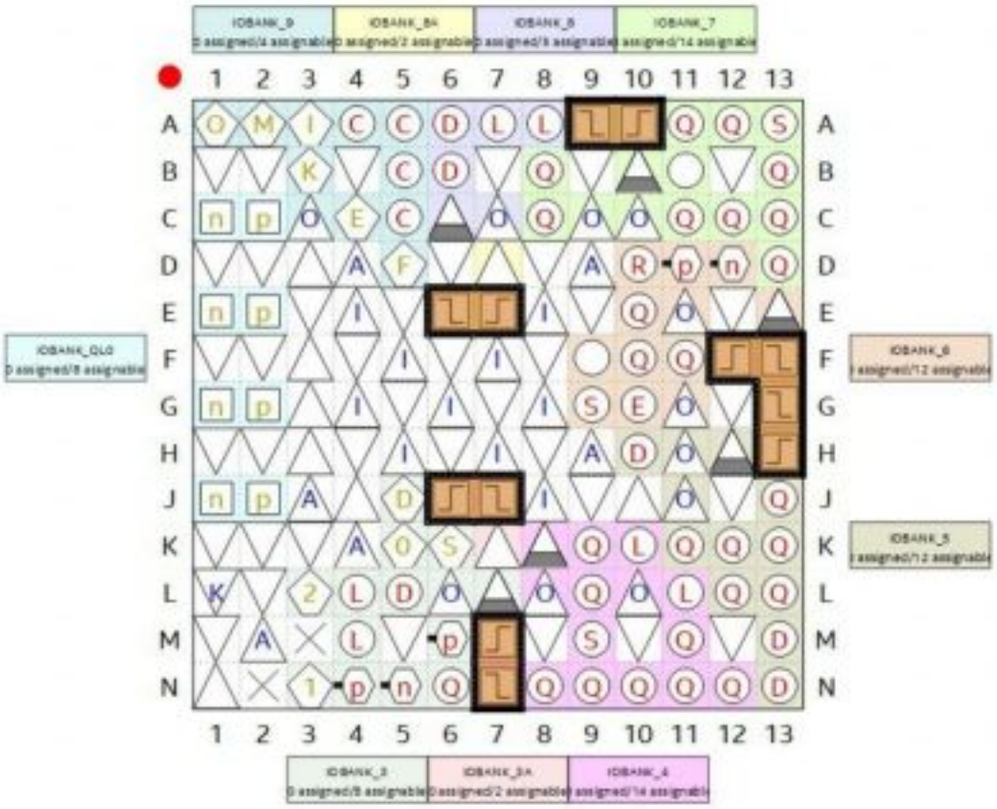





/SS_TB/addrBus
00000000
/SS_TB/clk
1
/SS_TB/done
St0
/SS_TB/inBus
0000000000000000
/SS_TB/memRdy
St1
/SS_TB/memRead
St0
/SS_TB/memWrite
St0
/SS_TB/outBus
0000000000000100
/SS_TB/start
0

Top View - Wire Bond

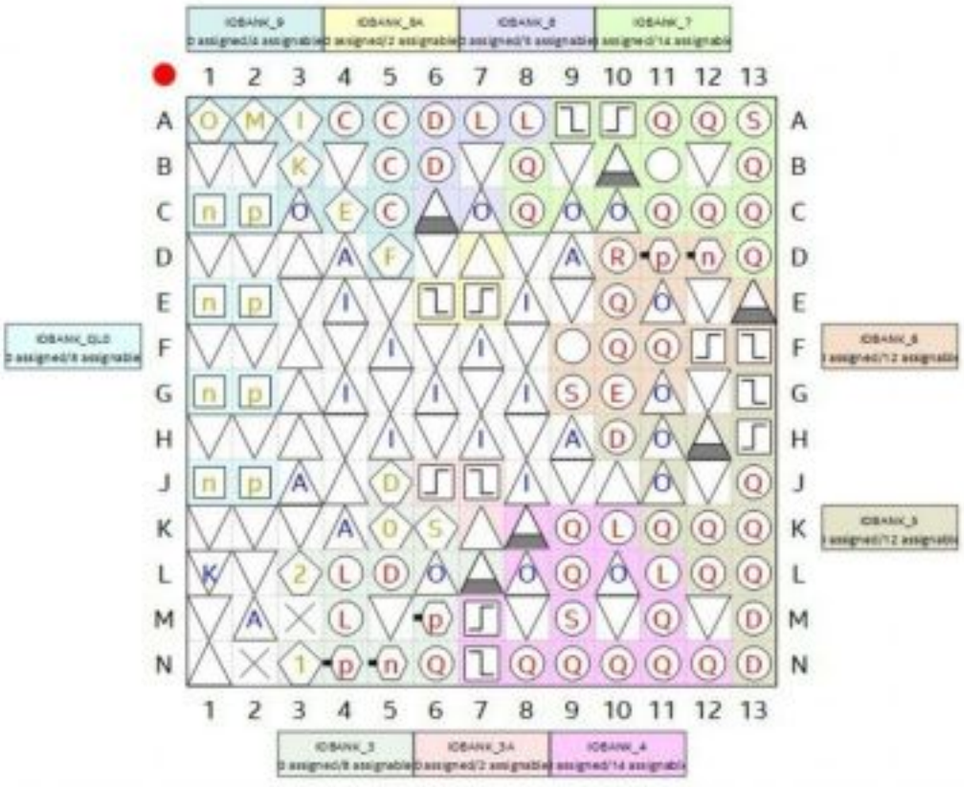
Cyclone IV GX - EP4CGX15BF14A7

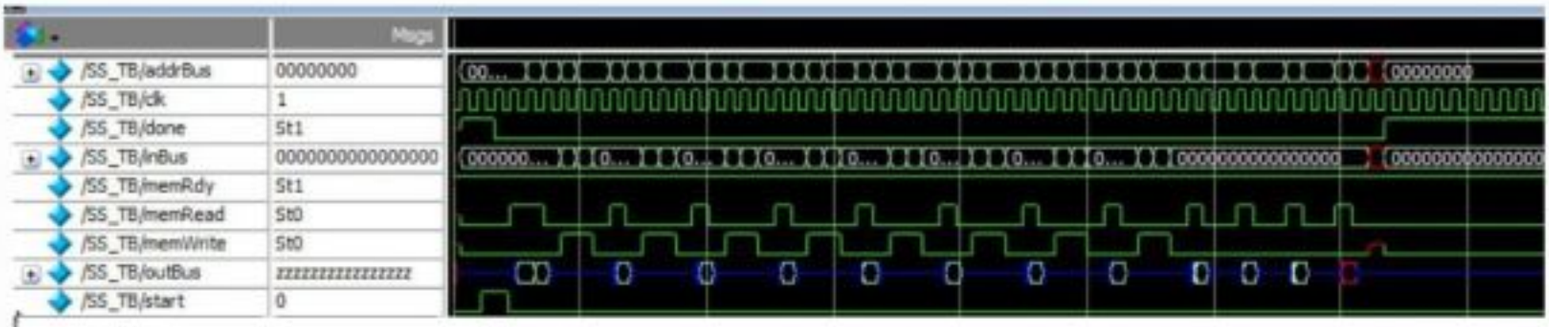


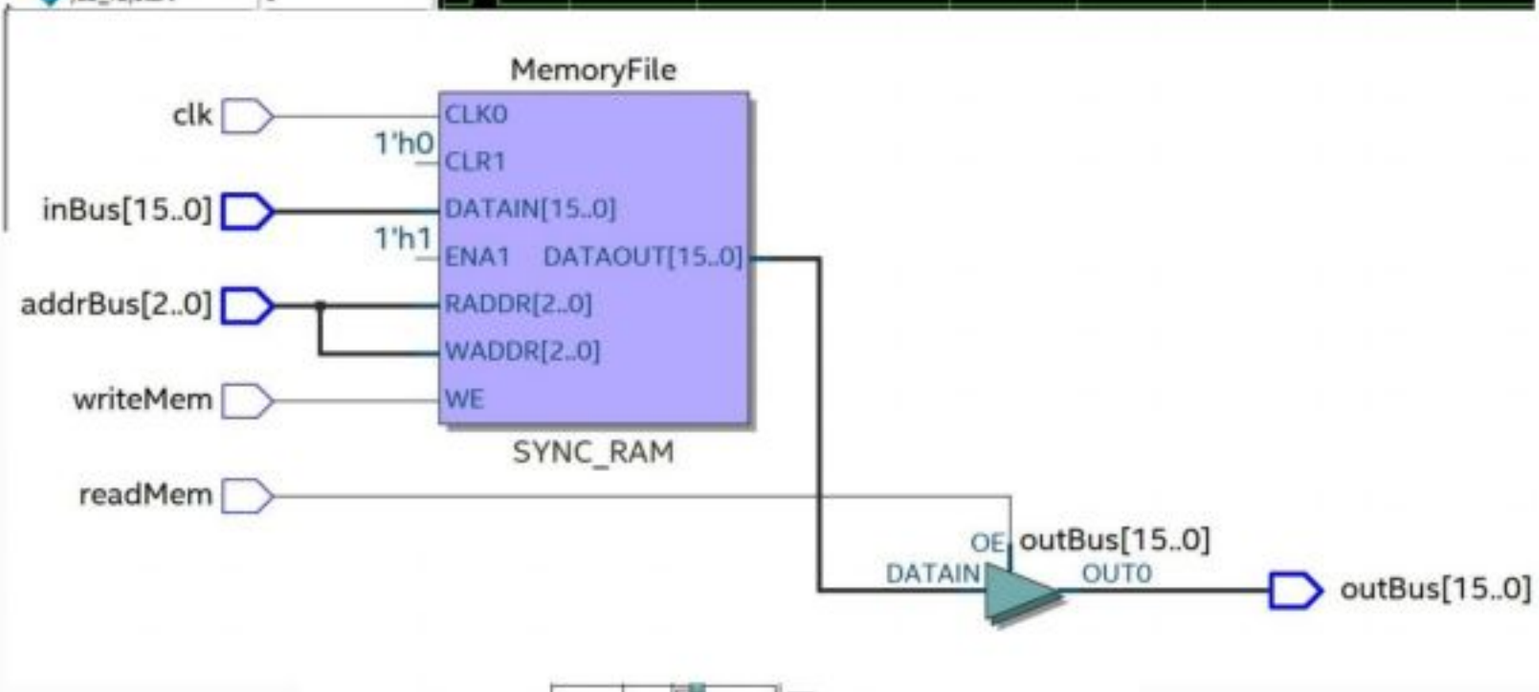
Synthesize in RAM mode

Top View - Wire Bond

Cyclone IV GX - EP4CGX15BF14A7




/SS_TB/addrBus
00000000
/SS_TB/clk
1
/SS_TB/done
St1
/SS_TB/inBus
0000000000000000
/SS_TB/memRdy
St1
/SS_TB/memRead
St0
/SS_TB/memWrite
St0
/SS_TB/outBus
zzzzzzzzzzzzzzzz
/SS_TB/start
0


MemoryFile
clk
1'h0
CLK0
CLR1
inBus[15..0]
1'h1
DATAIN[15..0]
ENA1
DATAOUT[15..0]
addrBus[2..0]
RADDR[2..0]
WADDR[2..0]
writeMem
WE
SYNC_RAM
readMem
OE
outBus[15..0]
DATAIN
OUT0
outBus[15..0]