KOTAKONDA ABITHA RAO
194238-B

Behavioural Modelling

Behavioural models in verilog contain procedural statements, which control the simulation and manipulate variables of the data types. These all statements are contained within the procedures. Each of the procedure has an activity flow associated with it.

During simulation of behavioural model, all the flows defined by the 'always' and 'initial' statements start together at simulation time 'zero'. the initial statements are executed once, and the always statements are executed repetively. The 'always' statement, because of its looping nature, is only useful when used in conjunction with some form of timing control.

KOTAKONDA ABITHA RAO
194238-B

3d] **Aim:-**

To design a synchronous up/down counter with async/sync load and clear in behavioural modelling.

Tools used:-

Xilinx Vivado 2019.1

Theory:-

→ In a synchronous counter all the 16 bits change all at once.
→ The load and clear inputs can be synchronous or asynchronous.
→ Asynchronous load is used to load the inputs with a value that we want at any point irrespective of the edge clock.
→ Synchronous load is used to load the inputs with a value that we want at a next clock edge.
→ If m=0, then it counts up.
→ If m=1, then it counts down
→ clr is used to clear the input when clr=1.
→ If clr=0, it doesn't clear.

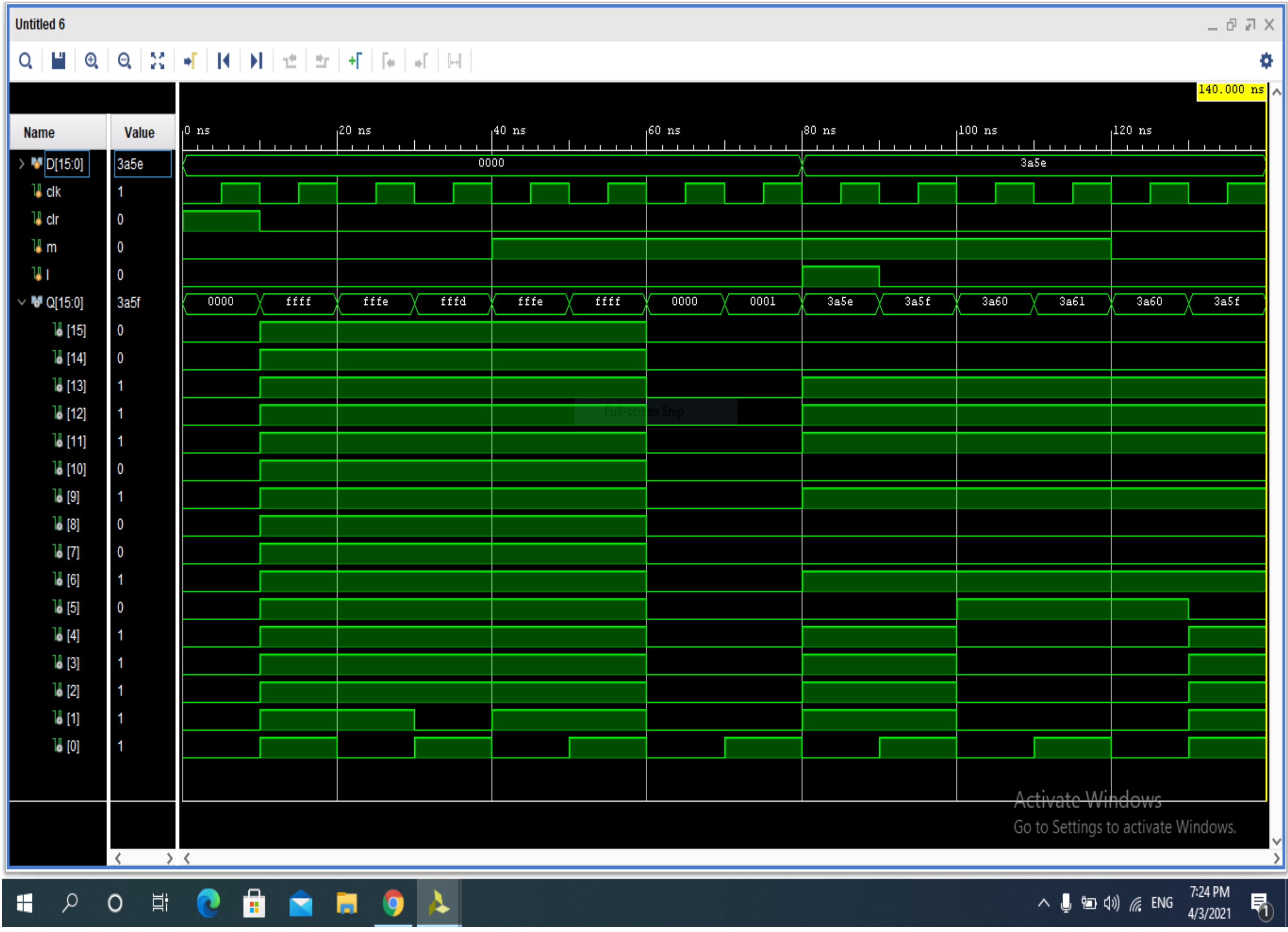
KOTAKONDA ABITHA RAO
194238-B

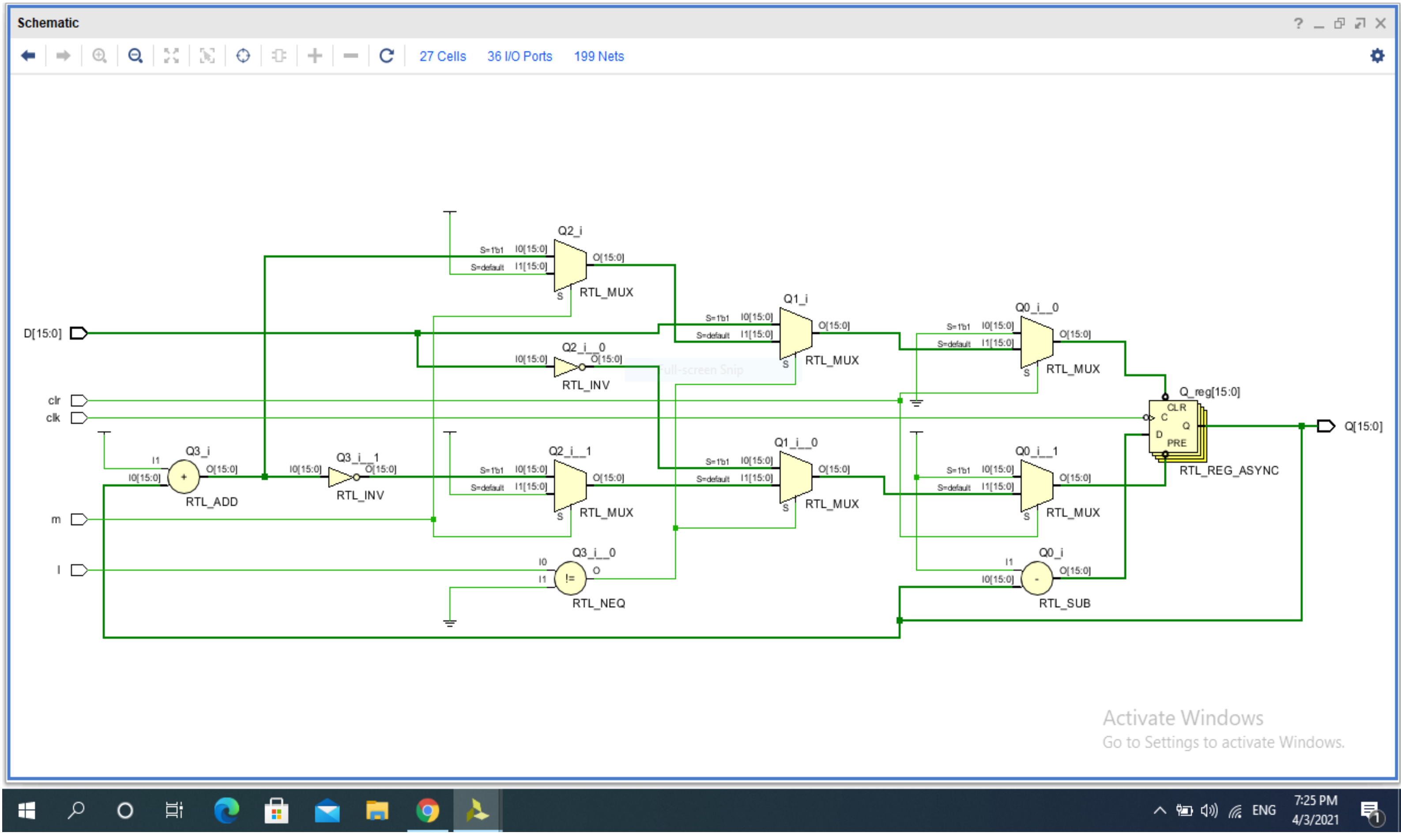
Code + Testbench:-

```
module  counter_16bit (Q, Clk, clr, D, m, l);
output reg [15:0] Q;
input clk, clr, m, l;
input [15:0] D;

always @(posedge clr, posedge l, posedge m , negedge
         clk)

begin
   if (clr)
        Q <= 16'b0000_0000_0000_0000;
   else if (l!=0)
        Q <= D;
   else if (m)
        Q<= Q +1;
   else
        Q<= Q-1;
end
endmodule


module  counter_16bit_tb();
reg  [15:0] D;
reg clk, clr, m, l;
wire [15:0] Q ;

counter_16bit  uut (Q, Clk, clr, D, m, l);

initial
begin
```

```
    clk = 1'b0;
    forever  #5 clk = ~clk;
end

initial
begin
    D = 16'b 0000_0000_0000_0000;
    clr = 1'b1; l = 1'b0; m = 1'b0;
    #10
    clr = 1'b0;
    #30
    m = 1'b1;
    #40
    l = 1'b1; D = 16'b 0011_1010_0101_1110;
    #10
    l = 1'b0;
    #30
    m = 1'b0;
    #20
    $stop;
end
endmodule
```

Conclusion:-
We have written the verilog code for a 16-bit counter with sync/async load and clear. We have also obtained the waveform for the testbench and the RTL has been generated.


Untitled 6
Name
Value
D[15:0] 3a5e
clk 1
clr 0
m 0
l 0
Q[15:0] 3a5f
[15] 0
[14] 0
[13] 1
[12] 1
[11] 1
[10] 0
[9] 1
[8] 0
[7] 0
[6] 1
[5] 0
[4] 1
[3] 1
[2] 1
[1] 1
[0] 1
140.000 ns
0 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
0000
3a5e
0000 ffff fffe fffd fffe ffff 0000 0001 3a5e 3a5f 3a60 3a61 3a60 3a5f
Activate Windows
Go to Settings to activate Windows.
7:24 PM
4/3/2021
ENG


Schematic
27 Cells
36 I/O Ports
199 Nets
D[15:0]
clr
clk
m
l
Q[15:0]
Q2_i
Q1_i
Q0_i__0
Q2_i__0
Q3_i
Q3_i__1
Q2_i__1
Q1_i__0
Q0_i__1
Q3_i__0
Q0_i
Q_reg[15:0]
RTL_MUX
RTL_INV
RTL_ADD
RTL_NEQ
RTL_SUB
RTL_REG_ASYNC
S=1'b1
S=default
I0[15:0]
I1[15:0]
O[15:0]
CLR
PRE
C
D
Q
Activate Windows
Go to Settings to activate Windows.
7:25 PM
4/3/2021
ENG

KOTAKONDA ABITHA RAO
194238-B

3e] <u>Aim:-</u>

To design a 16-bit universal shift register using behavioural modelling

<u>Tools Used:-</u>

Xilinx Vivad 2019.2

<u>Theory:-</u>

→ A universal shift register is used for:-

i) serial to-serial convertor
ii) Parallel-to-serial convertor
iii) serial-to-parallel convertor
iv) parallel data transfer
v) serial-to-serial data transfer
vi) Micro-controllers for i/o expansion.

→ It can be used in the SISO, PIPO, SIPO and PISO cases

→ D_par is used as parallel input in case of parallel in

→ MSB_in is used as serial input in case of shift right

→ LSB_in is used as parallel input in case of parallel in

→ The code and testbench for a 16 bit universal shift register is given in the next page.

KOTAKONDA ABITHA RAO
194238-13

Code + Testbench:-

```
module register-16bit(
input MSB-in, LSB-in, clk, clr,
input [15:0] D-par,
input [1:0] s,
output reg [15:0] A-par);

always @ (negedge clk, posedge clr)
begin
    if (clr)
        A-par<= 16'b0000_0000-0000_0000;
    else
        begin
            case (s)
            2'b00: A-par <= A-par;
            2'b01: A-par <= {MSB-in, A-par [15:1]};
            2'b10 ; A-par<= {A-par[14:0], LSB-in};
            2'b11 : A-par <= D-par;
            endcase
        end
end
endmodule


module register-16bit-tb();
reg MSB-in, LSB-in, clk, clr;
reg [15:0] D-par;
reg [1:0] s;
wire [15:0] A-par;
```

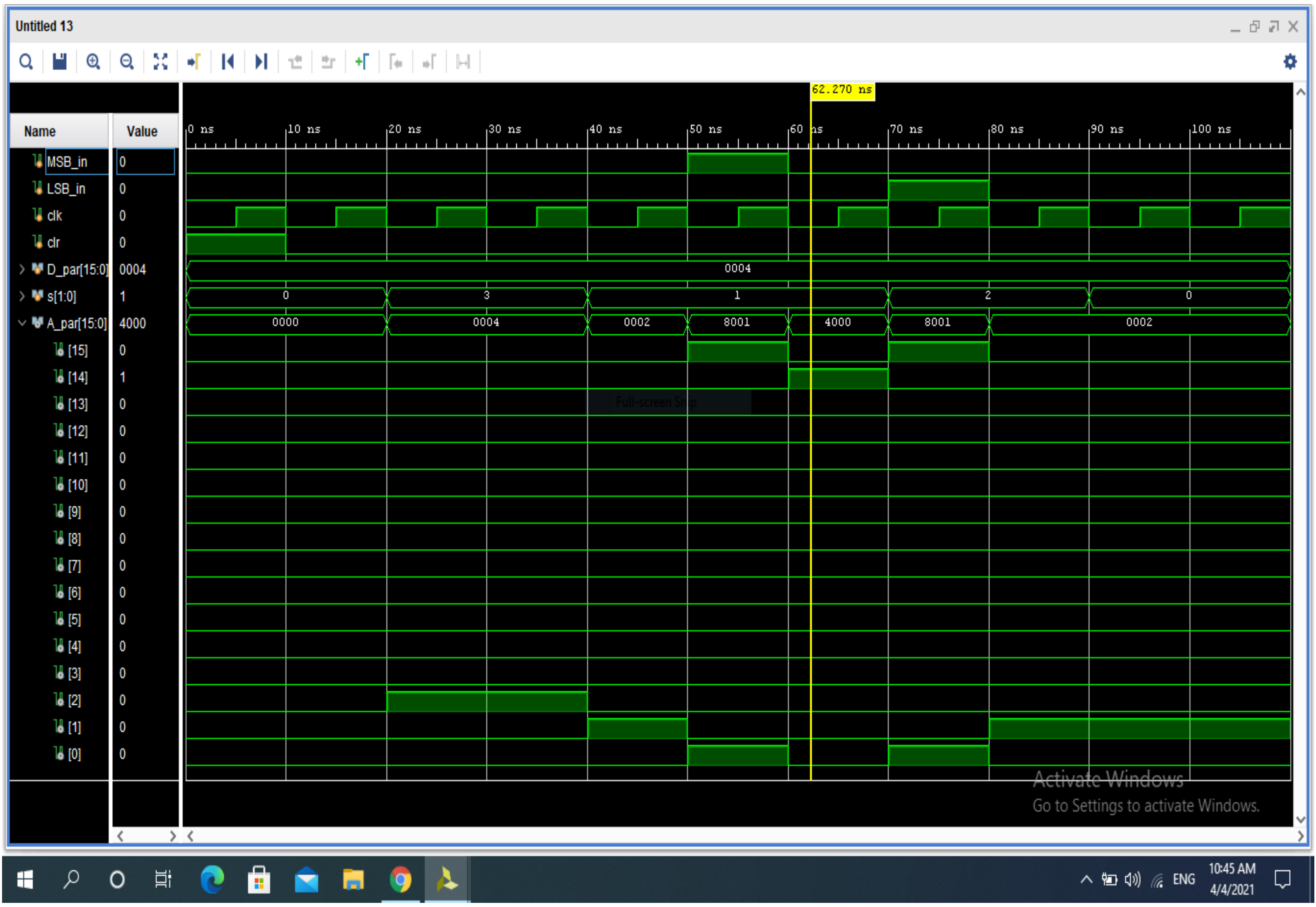
KOTAKONDA ABITHA RAO
194238 - B

```
register_16bit   uut (.MSB_in(MSB_in), .LSB_in(LSB_in),
                      .clk(clk), .clr(clr), .D_par(D_par),
                      .s(s), .A_par (A_par));

initial
begin
clk = 1'b0;
forever #5 clk = ~clk;
end

initial
begin
MSB_in = 1'b0; LSB_in = 1'b0;
clr = 1'b1; s = 2'b00; D_par = 16'b0000_0000_0000_0000;
#10
clr = 1'b0;
#10
s = 2'b11;
#20
MSB_in = 1'b0; s = 2'b01;
#10
MSB_in = 1'b1;
#10
MSB_in = 1'b0;
#10
LSB_in = 1'b1; s = 2'b10;
#10
LSB_in = 1'b0;
#10
s = 2'b00;
#20
```

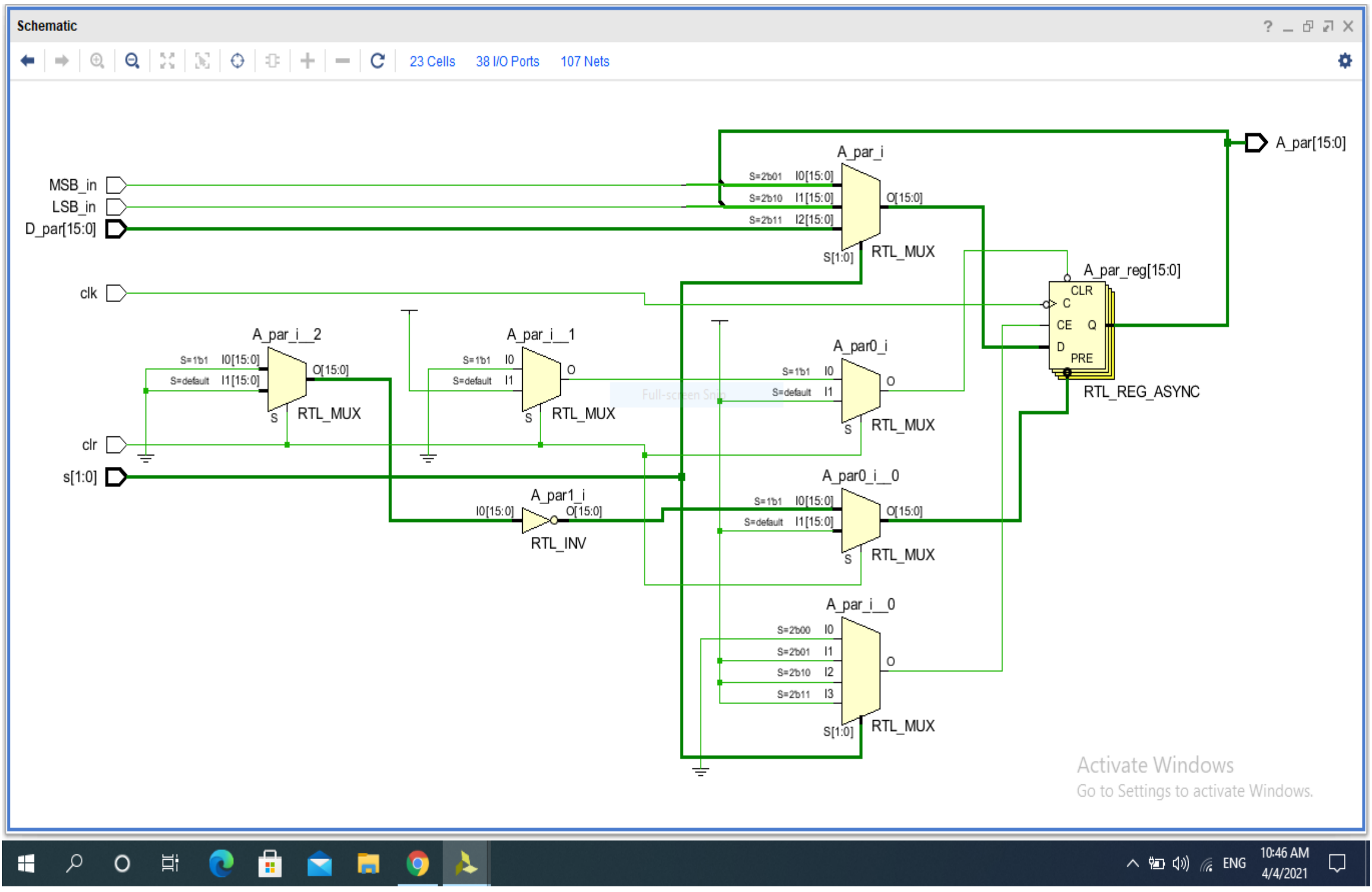
```
$stop;
end
endmodule
```

Conclusion:-

We have written the verilog code for 16bit universal shift register using behavioural modelling.
We have obtained the waveform as expected for the given testbench. RTL schematics has also been generated.



Untitled 13
62.270 ns
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns
100 ns
Name
Value
MSB_in 0
LSB_in 0
clk 0
clr 0
D_par[15:0] 0004
s[1:0] 1
A_par[15:0] 4000
[15] 0
[14] 1
[13] 0
[12] 0
[11] 0
[10] 0
[9] 0
[8] 0
[7] 0
[6] 0
[5] 0
[4] 0
[3] 0
[2] 0
[1] 0
[0] 0
0004
0
3
1
2
0
0000
0004
0002
8001
4000
8001
0002
Activate Windows
Go to Settings to activate Windows.
10:45 AM
4/4/2021
ENG



Schematic
23 Cells
38 I/O Ports
107 Nets
MSB_in
LSB_in
D_par[15:0]
clk
clr
s[1:0]
A_par[15:0]
A_par_i
S=2'b01 I0[15:0]
S=2'b10 I1[15:0]
S=2'b11 I2[15:0]
O[15:0]
S[1:0]
RTL_MUX
A_par_reg[15:0]
CLR
C
CE
Q
D
PRE
RTL_REG_ASYNC
A_par_i__2
S=1'b1 I0[15:0]
S=default I1[15:0]
O[15:0]
S
RTL_MUX
A_par_i__1
S=1'b1 I0
S=default I1
O
S
RTL_MUX
A_par0_i
S=1'b1 I0
S=default I1
O
S
RTL_MUX
A_par1_i
I0[15:0]
O[15:0]
RTL_INV
A_par0_i__0
S=1'b1 I0[15:0]
S=default I1[15:0]
O[15:0]
S
RTL_MUX
A_par_i__0
S=2'b00 I0
S=2'b01 I1
S=2'b10 I2
S=2'b11 I3
O
S[1:0]
RTL_MUX
Activate Windows
Go to Settings to activate Windows.
10:46 AM
4/4/2021
ENG