آزمایشگاه مدار منطقی

نویسنده:

علیرضا طباطبائیان(9723052)

Amirkabir University of Technology
(Tehran Polytechnic)

استاد آزمایشگاه:
دکتر علیرضا
طاهری نوید

THE PRESENT IS THEIRS, THE FUTURE, FOR

WHICH I REALLY WORK, IS MINE.

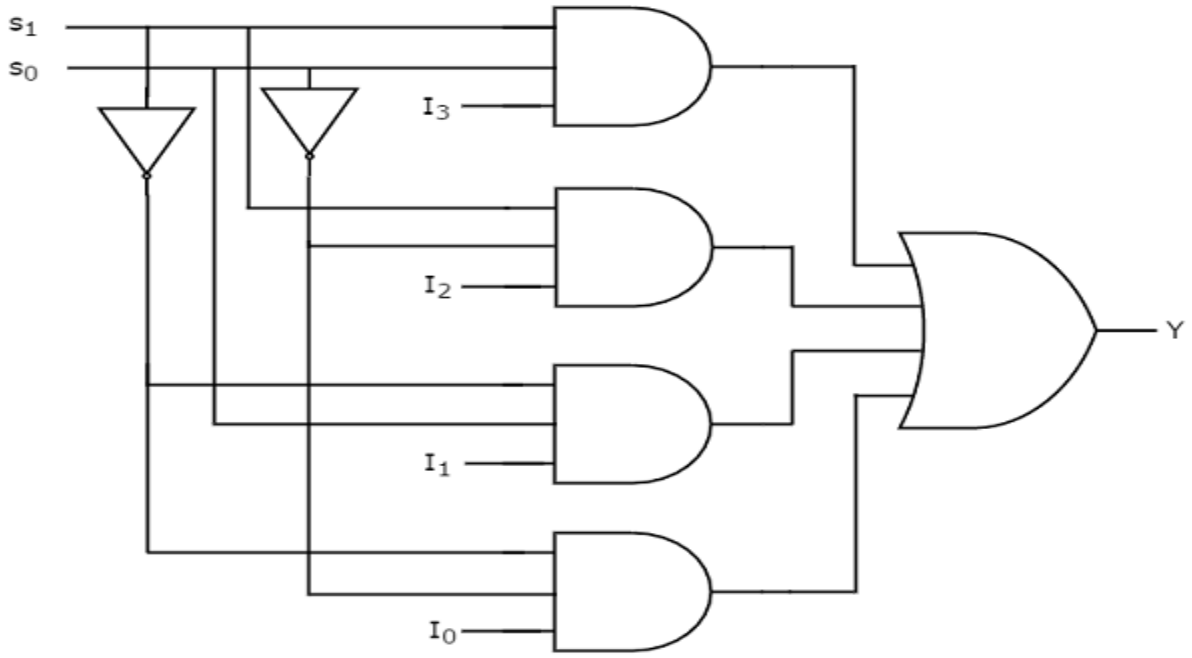
مالتی پلکسر 4 به 1(البته در کلاس طراحی گردید): یک آی سی که 4 پایه ورودی ، یک پایه خروجی و دو پایه انتخابی دارد و با توجه به اینکه کدام پایه های انتخابی روشن باشند، یکی از ورودی ها به خروجی انتقال پیدا میکند.

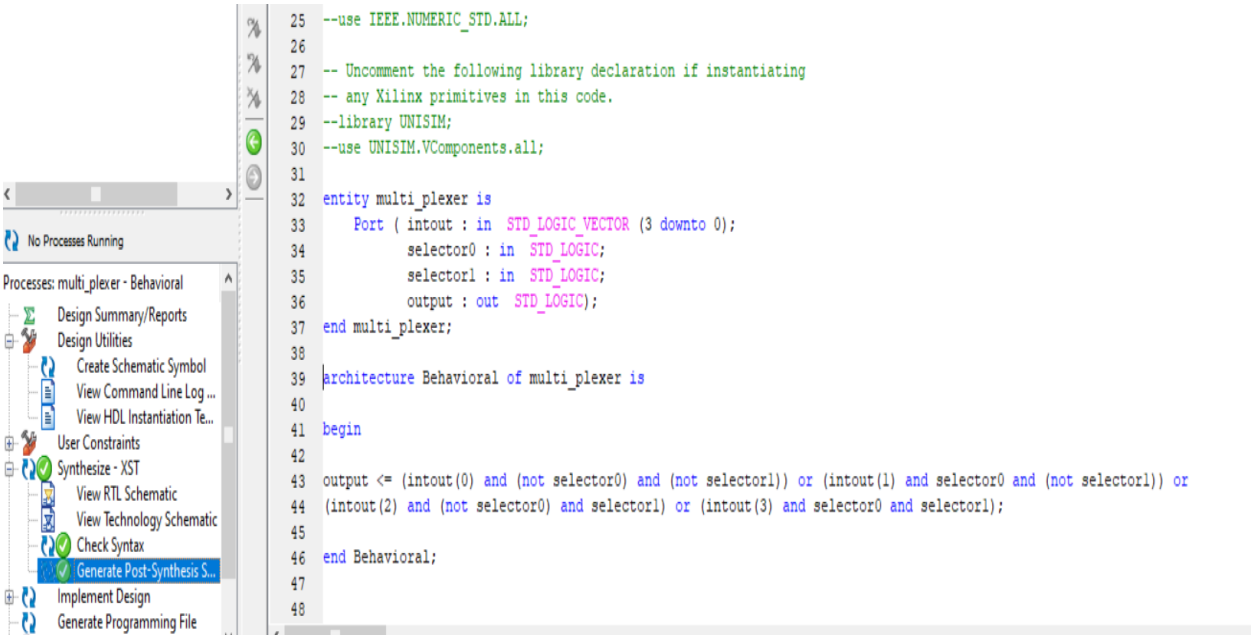
Truth Table

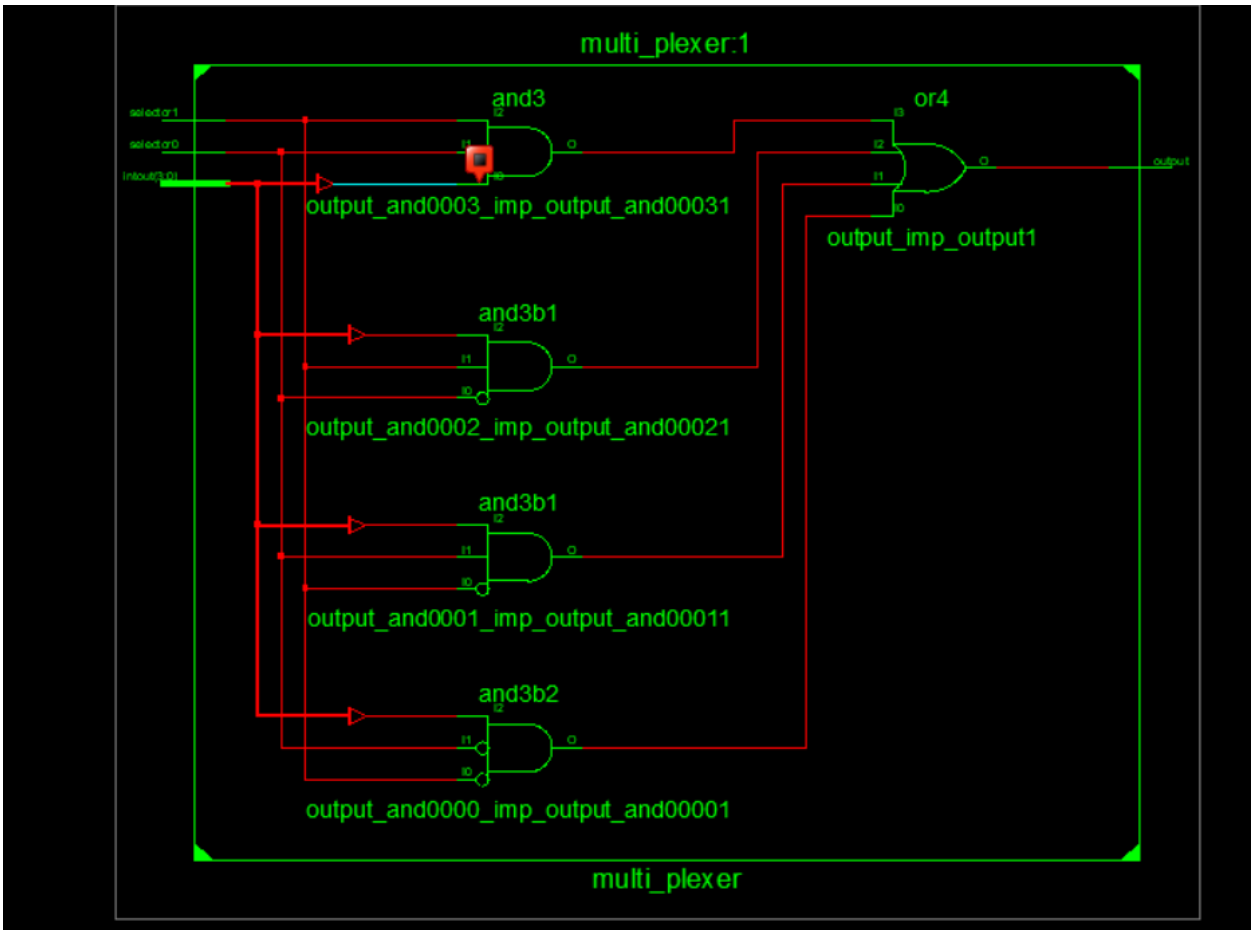
S0	S1	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

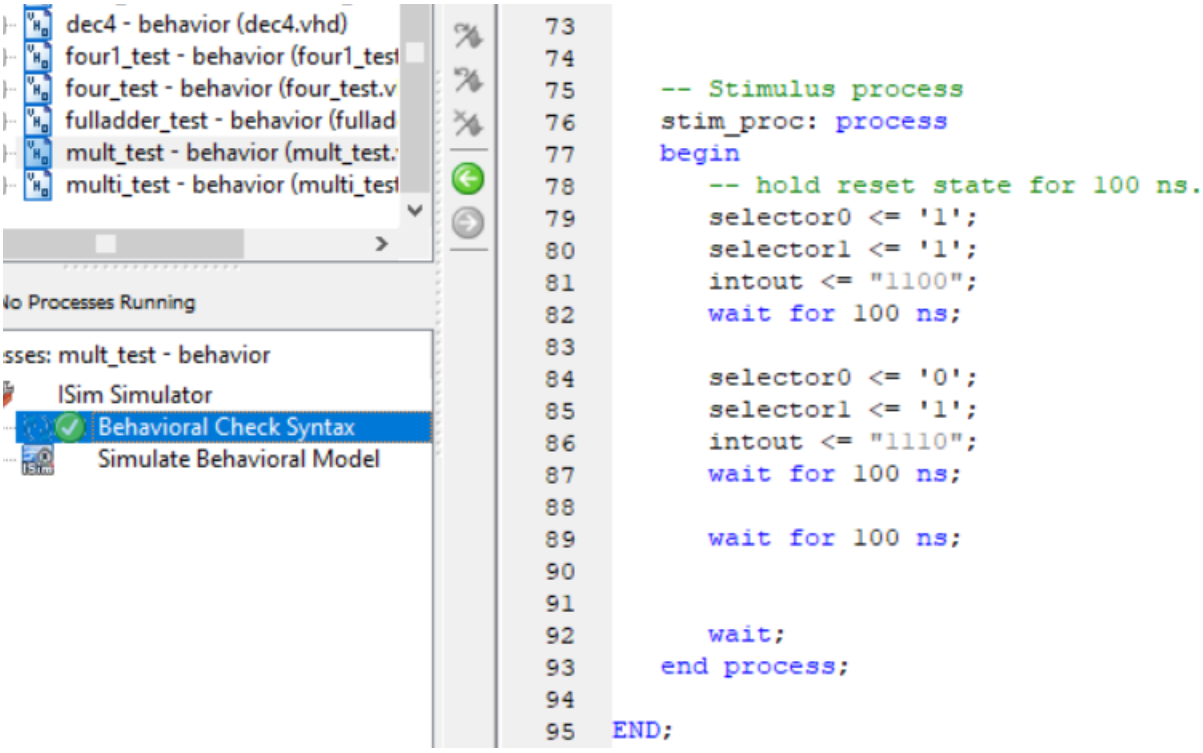
So, final equation,

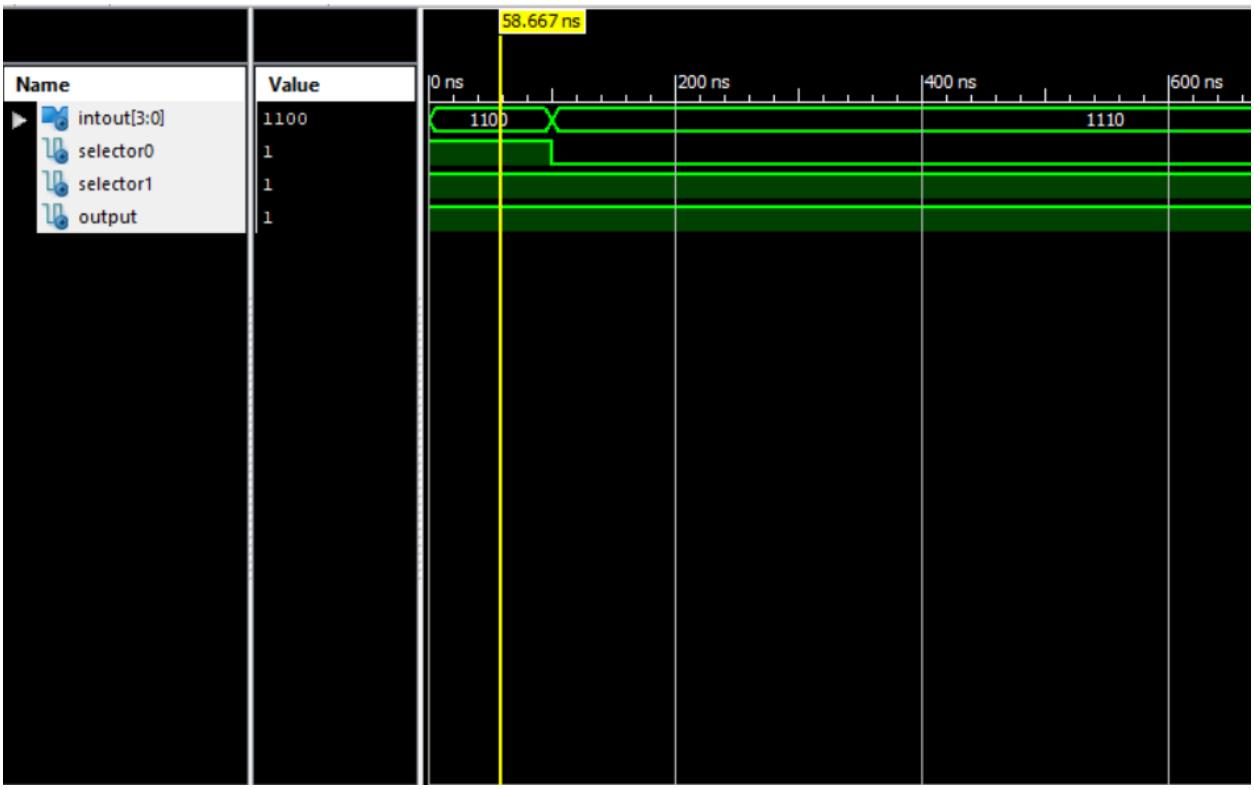
$Y = S0'.S1'.I0 + S0'.S1.I1 + S0.S1'.I2 + S0.S1.I3$




No Processes Running
Processes: multi_plexer - Behavioral
Design Summary/Reports
Design Utilities
Create Schematic Symbol
View Command Line Log ...
View HDL Instantiation Te...
User Constraints
Synthesize - XST
View RTL Schematic
View Technology Schematic
Check Syntax
Generate Post-Synthesis S...
Implement Design
Generate Programming File
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity multi_plexer is
33 Port (intout : in STD_LOGIC_VECTOR (3 downto 0);
34 selector0 : in STD_LOGIC;
35 selector1 : in STD_LOGIC;
36 output : out STD_LOGIC);
37 end multi_plexer;
38
39 architecture Behavioral of multi_plexer is
40
41 begin
42
43 output <= (intout(0) and (not selector0) and (not selector1)) or (intout(1) and selector0 and (not selector1)) or
44 (intout(2) and (not selector0) and selector1) or (intout(3) and selector0 and selector1);
45
46 end Behavioral;
47
48

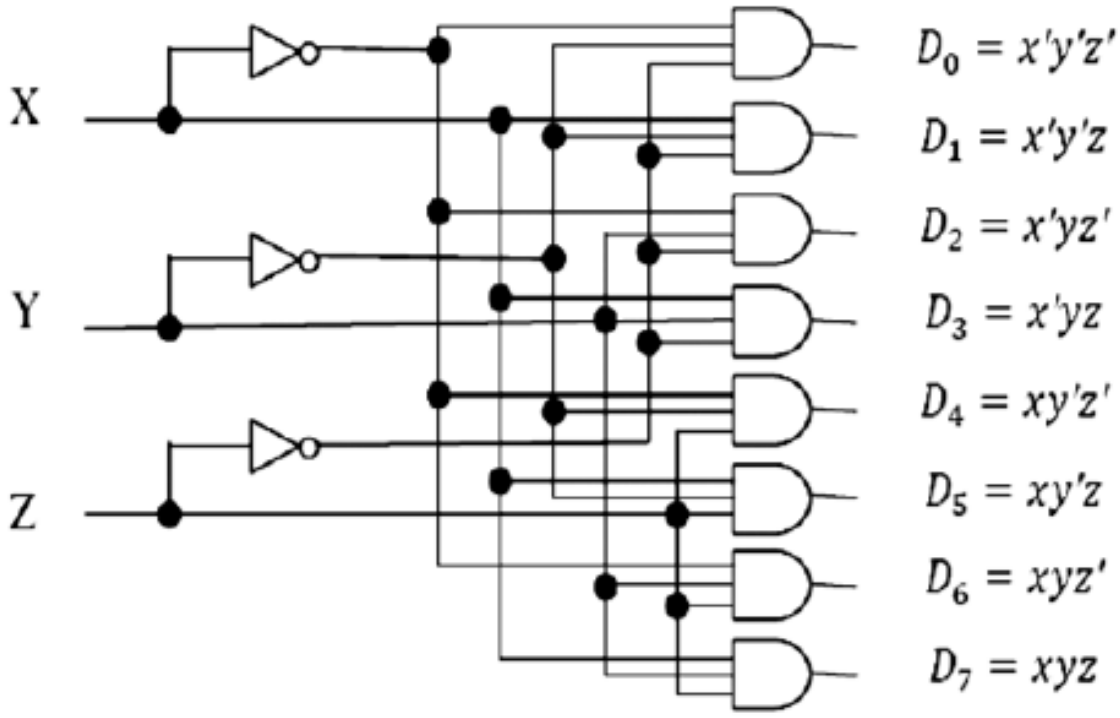

multi_plexer:1
and3
or4
output_and0003_imp_output_and00031
output_imp_output1
and3b1
output_and0002_imp_output_and00021
and3b1
output_and0001_imp_output_and00011
and3b2
output_and0000_imp_output_and00001
multi_plexer

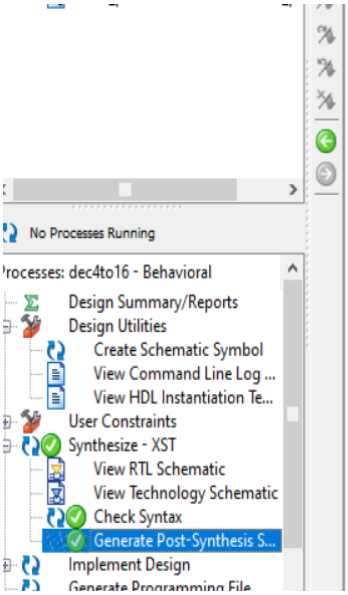

dec4 - behavior (dec4.vhd)
four1_test - behavior (four1_test
four_test - behavior (four_test.v
fulladder_test - behavior (fullad
mult_test - behavior (mult_test.
multi_test - behavior (multi_test
lo Processes Running
sses: mult_test - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
73
74
75 -- Stimulus process
76 stim_proc: process
77 begin
78 -- hold reset state for 100 ns.
79 selector0 <= '1';
80 selector1 <= '1';
81 intout <= "1100";
82 wait for 100 ns;
83
84 selector0 <= '0';
85 selector1 <= '1';
86 intout <= "1110";
87 wait for 100 ns;
88
89 wait for 100 ns;
90
91
92 wait;
93 end process;
94
95 END;


58.667 ns
Name
Value
0 ns
200 ns
400 ns
600 ns
intout[3:0]
1100
1100
1110
selector0
1
selector1
1
output
1

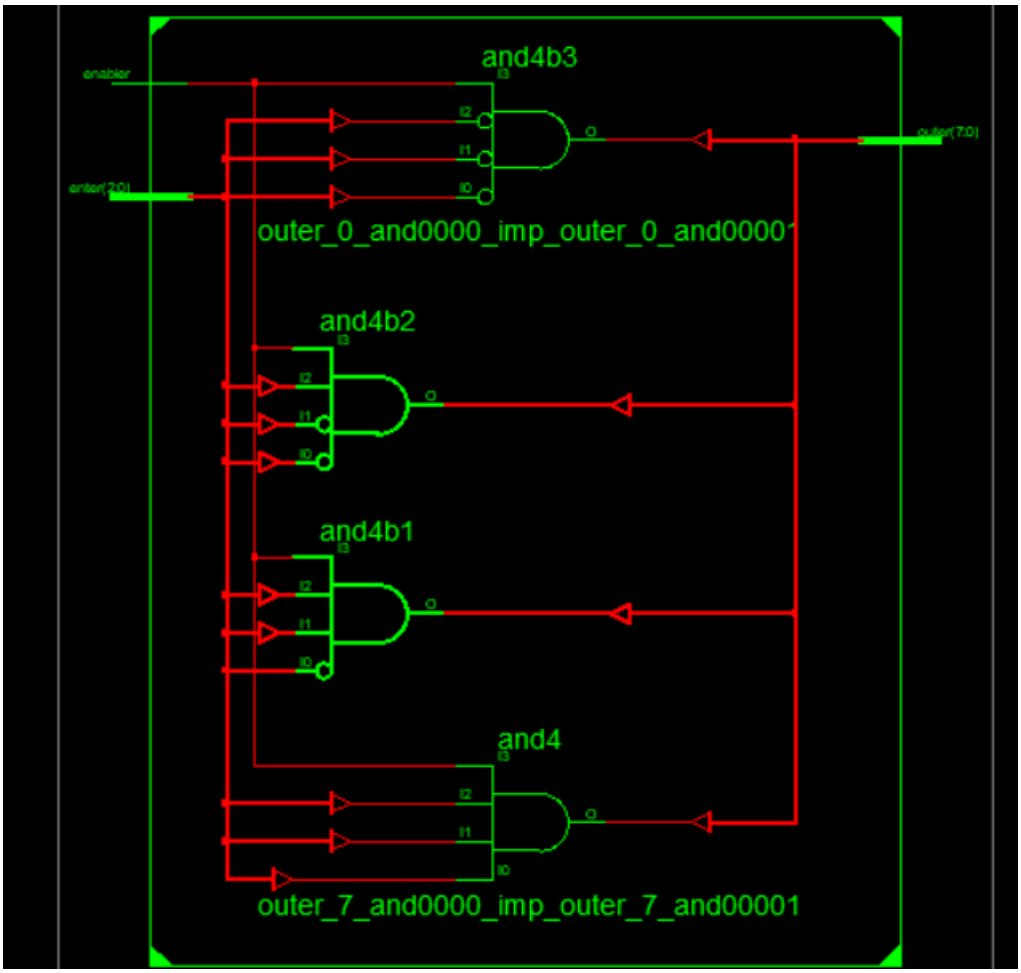
دیکدر 3 به 8 : یک آی سی که با توجه به اینکه کدام ورودی ها روشن است، یکی از خروجی ها روشن میشود(البته یک enabaler نیز به آن اضافه میکنیم)

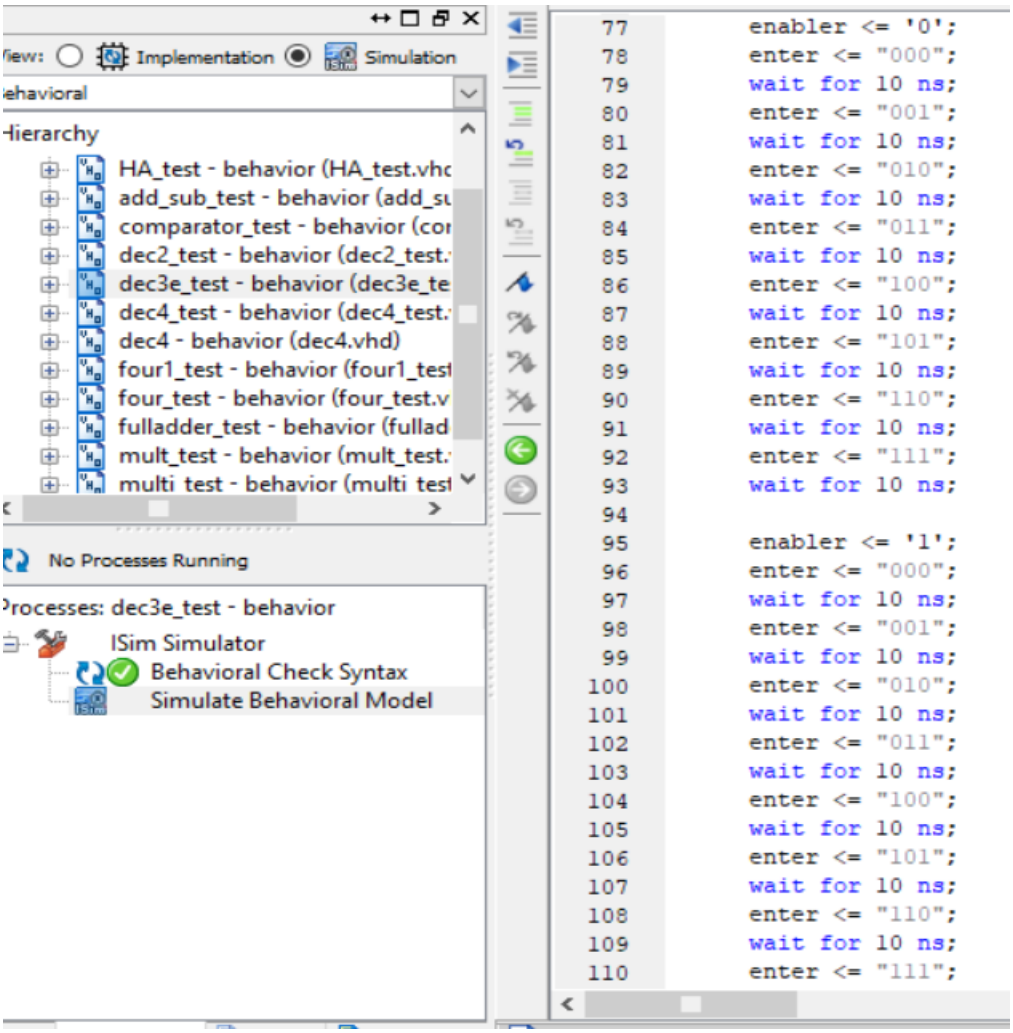
A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

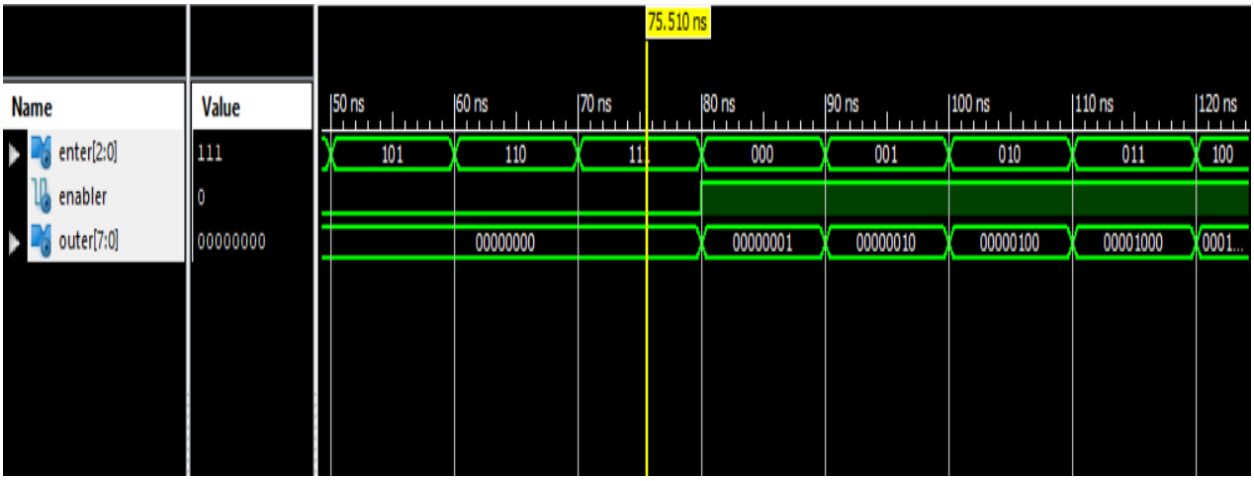




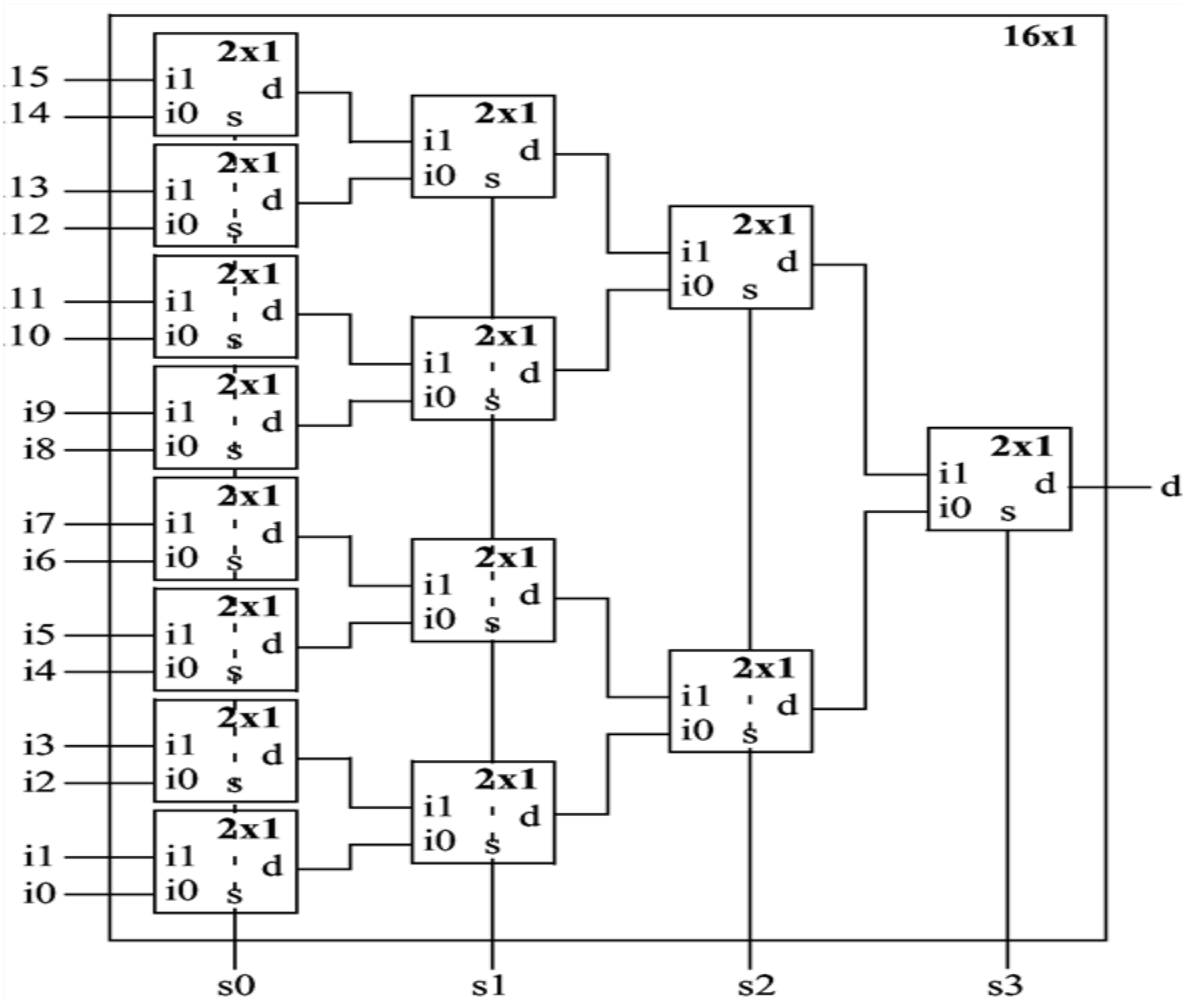

```
31
32  entity dec4to16 is
33      Port ( enter : in  STD_LOGIC_VECTOR (2 downto 0);
34             outer : out  STD_LOGIC_VECTOR (7 downto 0);
35             enabler : in std_logic);
36  end dec4to16;
37
38  architecture Behavioral of dec4to16 is
39
40  begin
41
42  outer(0) <= (not enter(0)) and (not enter(1)) and (not enter(2)) and enabler;
43  outer(1) <= (enter(0))     and (not enter(1)) and (not enter(2)) and enabler;
44  outer(2) <= (not enter(0)) and (enter(1))     and (not enter(2)) and enabler;
45  outer(3) <= (enter(0))     and (enter(1))     and (not enter(2)) and enabler;
46  outer(4) <= (not enter(0)) and (not enter(1)) and (enter(2))     and enabler;
47  outer(5) <= (enter(0))     and (not enter(1)) and (enter(2))     and enabler;
48  outer(6) <= (not enter(0)) and (enter(1))     and (enter(2))     and enabler;
49  outer(7) <= (enter(0))     and (enter(1))     and (enter(2))     and enabler;
50
51
52  end Behavioral;
53
54
```

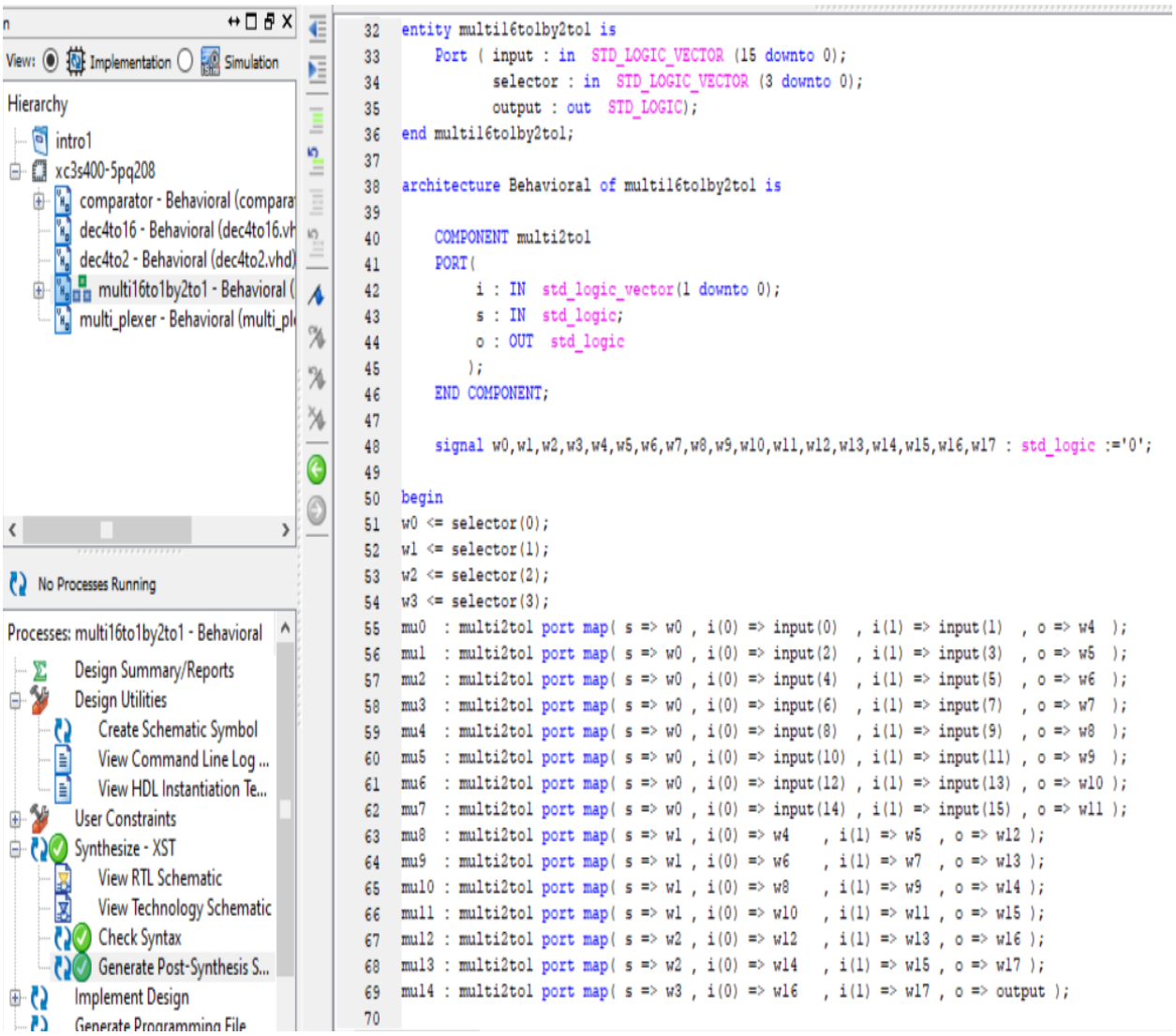



View: Implementation Simulation
Behavioral
Hierarchy
HA_test - behavior (HA_test.vhd
add_sub_test - behavior (add_su
comparator_test - behavior (cor
dec2_test - behavior (dec2_test.
dec3e_test - behavior (dec3e_te
dec4_test - behavior (dec4_test.
dec4 - behavior (dec4.vhd)
four1_test - behavior (four1_test
four_test - behavior (four_test.v
fulladder_test - behavior (fullad
mult_test - behavior (mult_test.
multi_test - behavior (multi_test
No Processes Running
Processes: dec3e_test - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
77 enabler <= '0';
78 enter <= "000";
79 wait for 10 ns;
80 enter <= "001";
81 wait for 10 ns;
82 enter <= "010";
83 wait for 10 ns;
84 enter <= "011";
85 wait for 10 ns;
86 enter <= "100";
87 wait for 10 ns;
88 enter <= "101";
89 wait for 10 ns;
90 enter <= "110";
91 wait for 10 ns;
92 enter <= "111";
93 wait for 10 ns;
94
95 enabler <= '1';
96 enter <= "000";
97 wait for 10 ns;
98 enter <= "001";
99 wait for 10 ns;
100 enter <= "010";
101 wait for 10 ns;
102 enter <= "011";
103 wait for 10 ns;
104 enter <= "100";
105 wait for 10 ns;
106 enter <= "101";
107 wait for 10 ns;
108 enter <= "110";
109 wait for 10 ns;
110 enter <= "111";

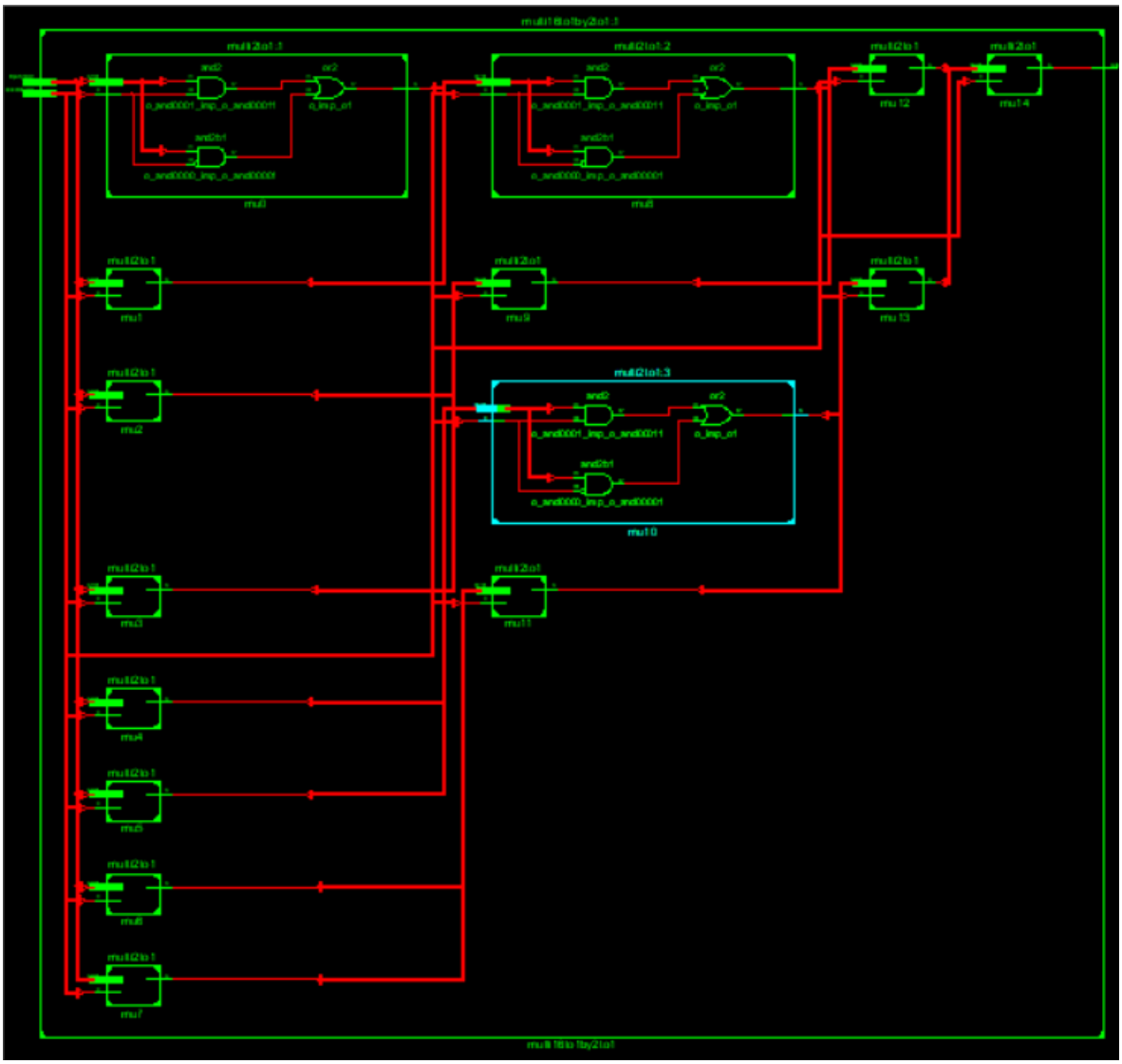

75.510 ns
Name
Value
enter[2:0]
111
enabler
0
outer[7:0]
00000000
50 ns
60 ns
70 ns
80 ns
90 ns
100 ns
110 ns
120 ns
101
110
111
000
001
010
011
100
00000000
00000001
00000010
00000100
00001000

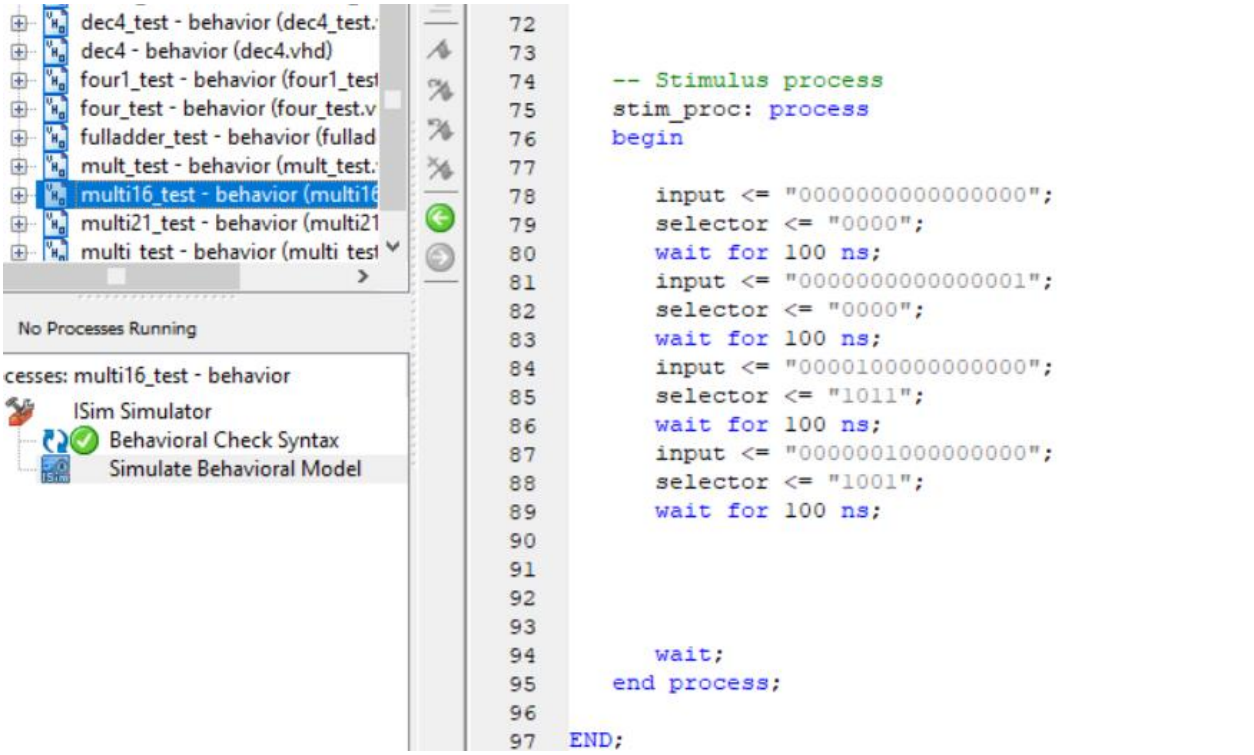
طراحی مالتی پلکسر 16 به 1 توسط 2 به 1 : طبیعتا چون 16 ورودی و 4 پایه کنترلی داریم، پس ابتدا به 8 مالتی پلکسر نیاز داریم و سپس برای انتخاب از آن 8 خروجی، به 4 مالتی پلکسر نیاز است و همینطور ادامه میدهیم تا به 1 مالتی پلکسر برسیم.

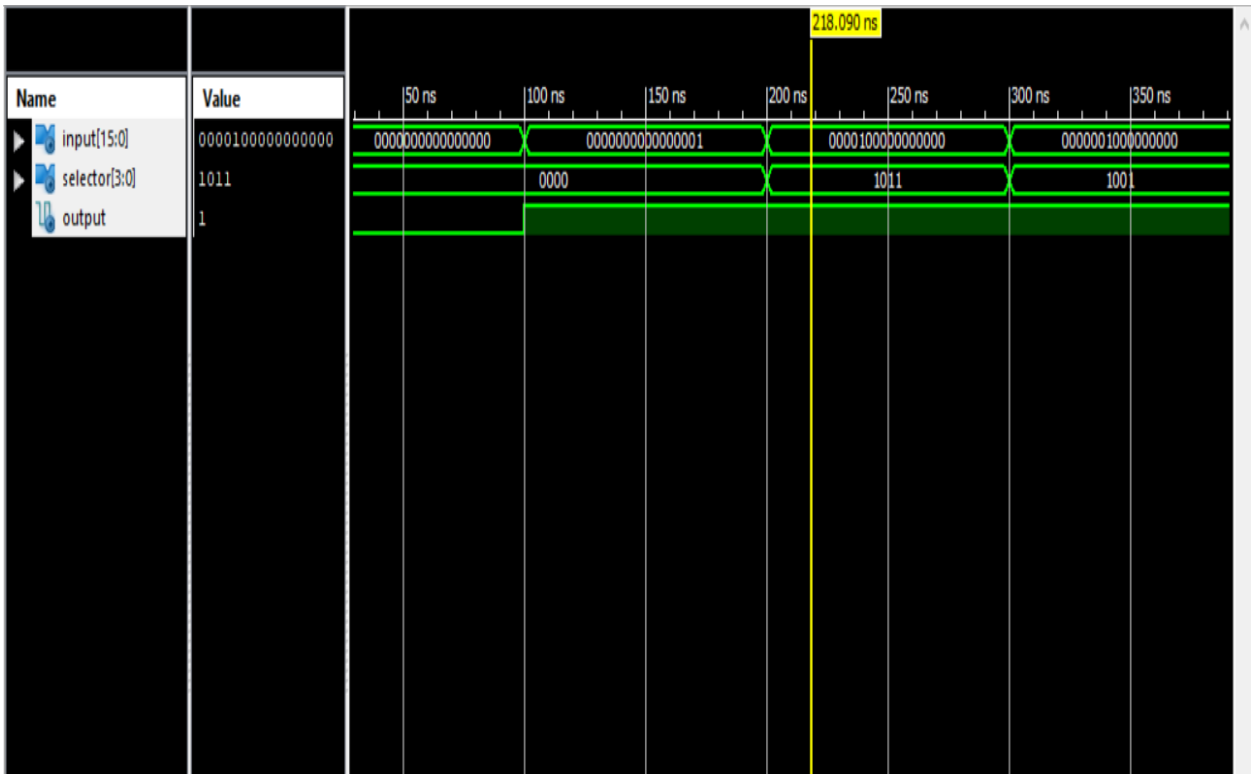





```
32  entity multil6tolby2tol is
33      Port ( input : in  STD_LOGIC_VECTOR (15 downto 0);
34             selector : in  STD_LOGIC_VECTOR (3 downto 0);
35             output : out  STD_LOGIC);
36  end multil6tolby2tol;
37
38  architecture Behavioral of multil6tolby2tol is
39
40      COMPONENT multi2tol
41      PORT(
42           i : IN  std_logic_vector(1 downto 0);
43           s : IN  std_logic;
44           o : OUT  std_logic
45          );
46      END COMPONENT;
47
48      signal w0,w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13,w14,w15,w16,w17 : std_logic :='0';
49
50  begin
51  w0 <= selector(0);
52  w1 <= selector(1);
53  w2 <= selector(2);
54  w3 <= selector(3);
55  mu0  : multi2tol port map( s => w0 , i(0) => input(0)  , i(1) => input(1)  , o => w4  );
56  mu1  : multi2tol port map( s => w0 , i(0) => input(2)  , i(1) => input(3)  , o => w5  );
57  mu2  : multi2tol port map( s => w0 , i(0) => input(4)  , i(1) => input(5)  , o => w6  );
58  mu3  : multi2tol port map( s => w0 , i(0) => input(6)  , i(1) => input(7)  , o => w7  );
59  mu4  : multi2tol port map( s => w0 , i(0) => input(8)  , i(1) => input(9)  , o => w8  );
60  mu5  : multi2tol port map( s => w0 , i(0) => input(10) , i(1) => input(11) , o => w9  );
61  mu6  : multi2tol port map( s => w0 , i(0) => input(12) , i(1) => input(13) , o => w10 );
62  mu7  : multi2tol port map( s => w0 , i(0) => input(14) , i(1) => input(15) , o => w11 );
63  mu8  : multi2tol port map( s => w1 , i(0) => w4    , i(1) => w5  , o => w12 );
64  mu9  : multi2tol port map( s => w1 , i(0) => w6    , i(1) => w7  , o => w13 );
65  mu10 : multi2tol port map( s => w1 , i(0) => w8    , i(1) => w9  , o => w14 );
66  mu11 : multi2tol port map( s => w1 , i(0) => w10   , i(1) => w11 , o => w15 );
67  mu12 : multi2tol port map( s => w2 , i(0) => w12   , i(1) => w13 , o => w16 );
68  mu13 : multi2tol port map( s => w2 , i(0) => w14   , i(1) => w15 , o => w17 );
69  mu14 : multi2tol port map( s => w3 , i(0) => w16   , i(1) => w17 , o => output );
70
```


mult16to1by2to1_1
mult2to1_1
mux0
mult2to1_2
mux8
mux12
mux14
mux1
mux9
mux13
mux2
mult2to1_3
mux10
mux3
mux11
mux4
mux5
mux6
mux7
mult16to1by2to1

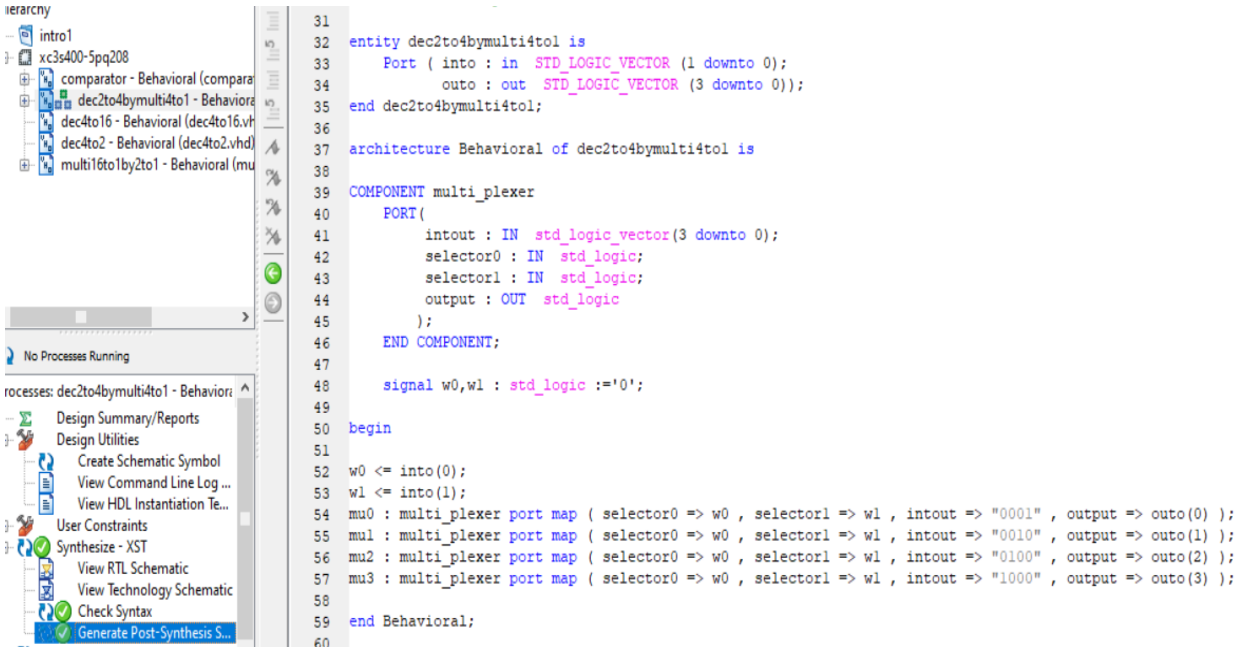

dec4_test - behavior (dec4_test.
dec4 - behavior (dec4.vhd)
four1_test - behavior (four1_test
four_test - behavior (four_test.v
fulladder_test - behavior (fullad
mult_test - behavior (mult_test.
multi16_test - behavior (multi16
multi21_test - behavior (multi21
multi_test - behavior (multi_test
No Processes Running
cesses: multi16_test - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
72
73
74 -- Stimulus process
75 stim_proc: process
76 begin
77
78 input <= "0000000000000000";
79 selector <= "0000";
80 wait for 100 ns;
81 input <= "0000000000000001";
82 selector <= "0000";
83 wait for 100 ns;
84 input <= "0000100000000000";
85 selector <= "1011";
86 wait for 100 ns;
87 input <= "0000001000000000";
88 selector <= "1001";
89 wait for 100 ns;
90
91
92
93
94 wait;
95 end process;
96
97 END;


218.090 ns
Name
Value
50 ns
100 ns
150 ns
200 ns
250 ns
300 ns
350 ns
input[15:0]
0000100000000000
0000000000000000
0000000000000001
0000100000000000
0000010000000000
selector[3:0]
1011
0000
1011
1001
output
1

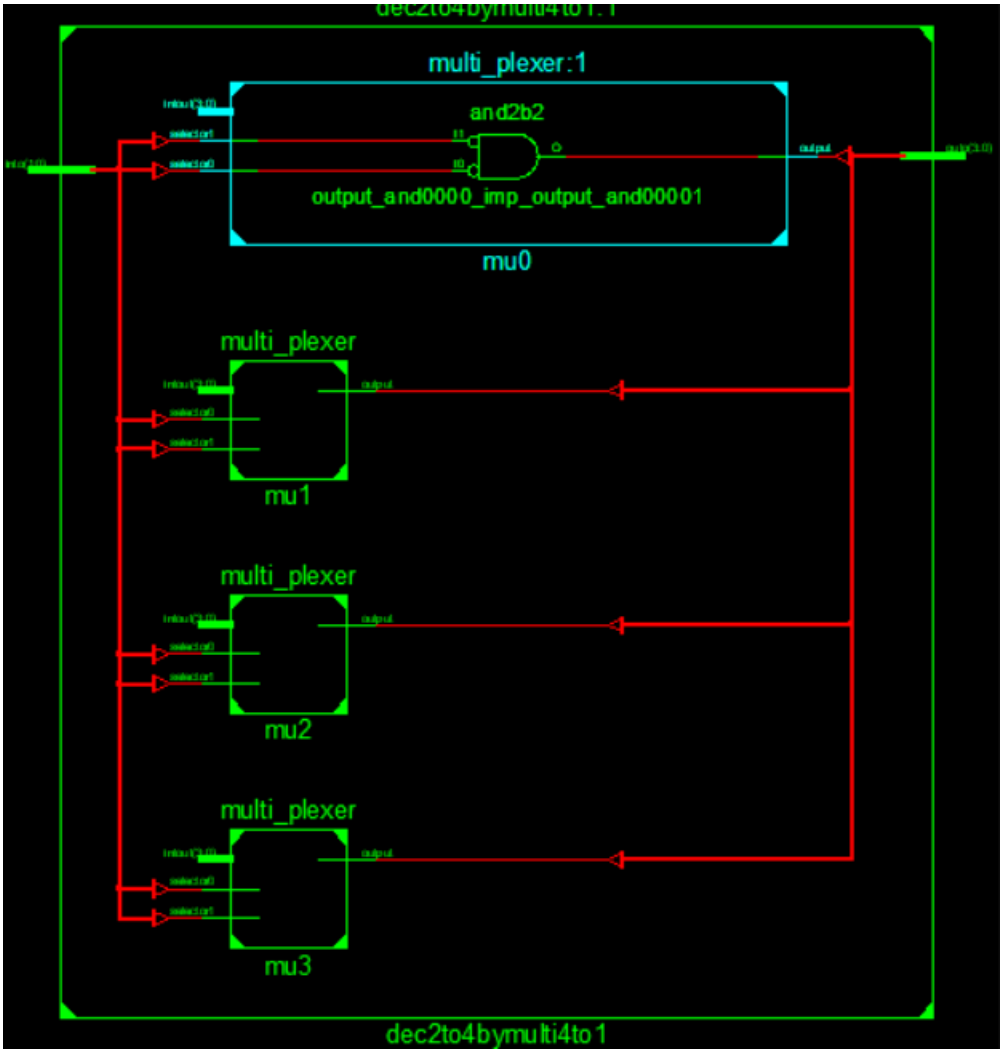
ساخت دکودر 2 در 4 بوسیله مالتی پلکسر چهار در یک:

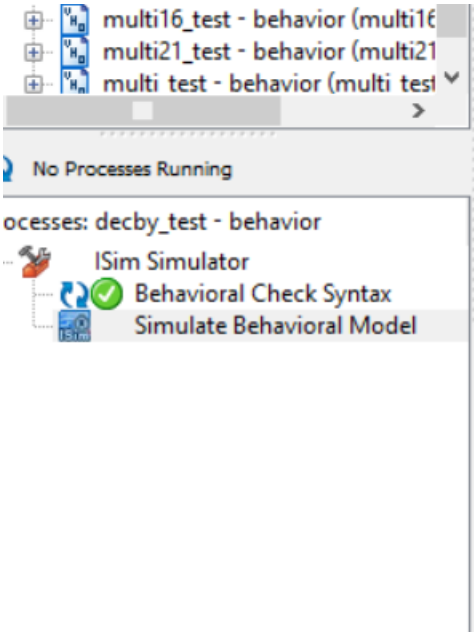
جهت انجام این کار، چهار مالتی پلکسر با 2 پایه کنترلی را بصورت موازی قرار داده و همه ورودی های اول کنترل آنها را به ورودی اول مدار و همه ورودی های دوم کنترل آنها را به ورودی دوم مدار وصل میکنیم و برای مالتی پلکسر اول، همه بیت های ورودی بجز اولی را صفر قرار داده و اولی را 1 قرار میدهیم. برای مالتی پلکسر دومی، همه بجز دومی را صفر میگذاریم. همین کار را تا مالتی پلکسر چهارم ادامه میدهیم.

اکنون یک دیکودر 2 در 4 داریم که ورودی های دیکودر، به پایه های کنترلی مالتی پلکسر و خروجی های دیکودر، از خروجی های مالتی پلکسر گرفته میشود.




```
31
32  entity dec2to4bymulti4to1 is
33      Port ( into : in  STD_LOGIC_VECTOR (1 downto 0);
34             outo : out  STD_LOGIC_VECTOR (3 downto 0));
35  end dec2to4bymulti4to1;
36
37  architecture Behavioral of dec2to4bymulti4to1 is
38
39  COMPONENT multi_plexer
40      PORT(
41           intout : IN  std_logic_vector(3 downto 0);
42           selector0 : IN  std_logic;
43           selector1 : IN  std_logic;
44           output : OUT  std_logic
45          );
46      END COMPONENT;
47
48      signal w0,w1 : std_logic :='0';
49
50  begin
51
52  w0 <= into(0);
53  w1 <= into(1);
54  mu0 : multi_plexer port map ( selector0 => w0 , selector1 => w1 , intout => "0001" , output => outo(0) );
55  mu1 : multi_plexer port map ( selector0 => w0 , selector1 => w1 , intout => "0010" , output => outo(1) );
56  mu2 : multi_plexer port map ( selector0 => w0 , selector1 => w1 , intout => "0100" , output => outo(2) );
57  mu3 : multi_plexer port map ( selector0 => w0 , selector1 => w1 , intout => "1000" , output => outo(3) );
58
59  end Behavioral;
60
```

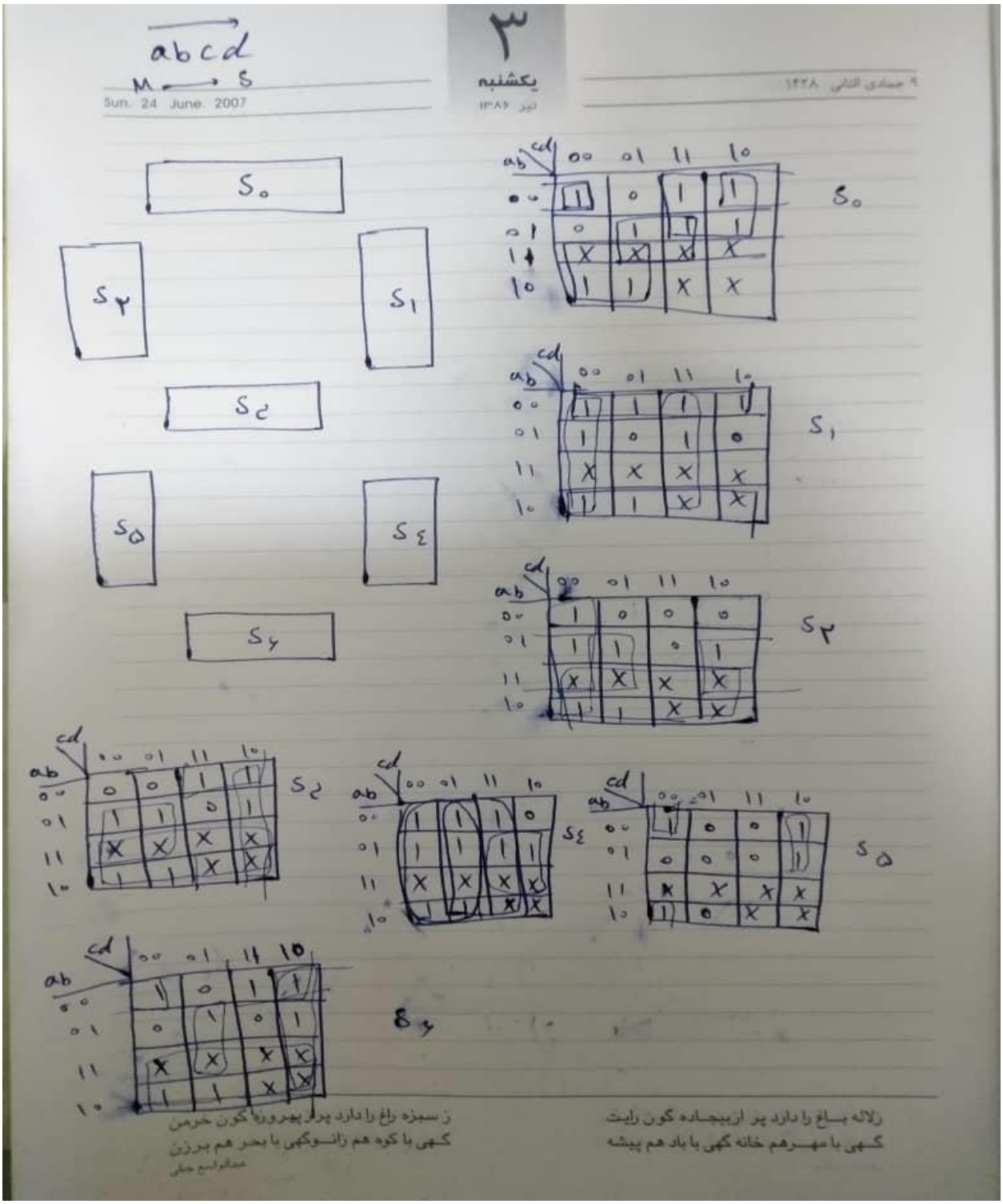


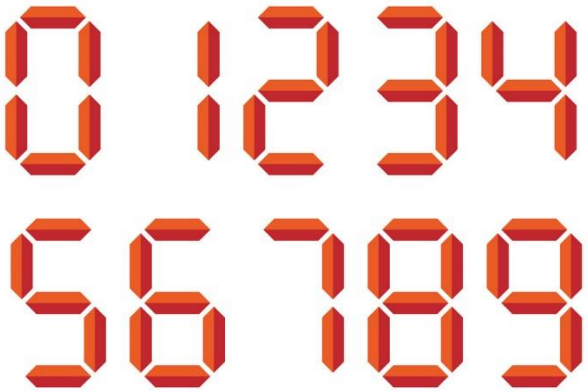



```
68
69
70      -- Stimulus process
71      stim_proc: process
72      begin
73
74         into <= "00";
75         wait for 100 ns;
76         into <= "01";
77         wait for 100 ns;
78         into <= "10";
79         wait for 100 ns;
80         into <= "11";
81         wait for 100 ns;
82
83         wait;
84      end process;
85
86  END;
```

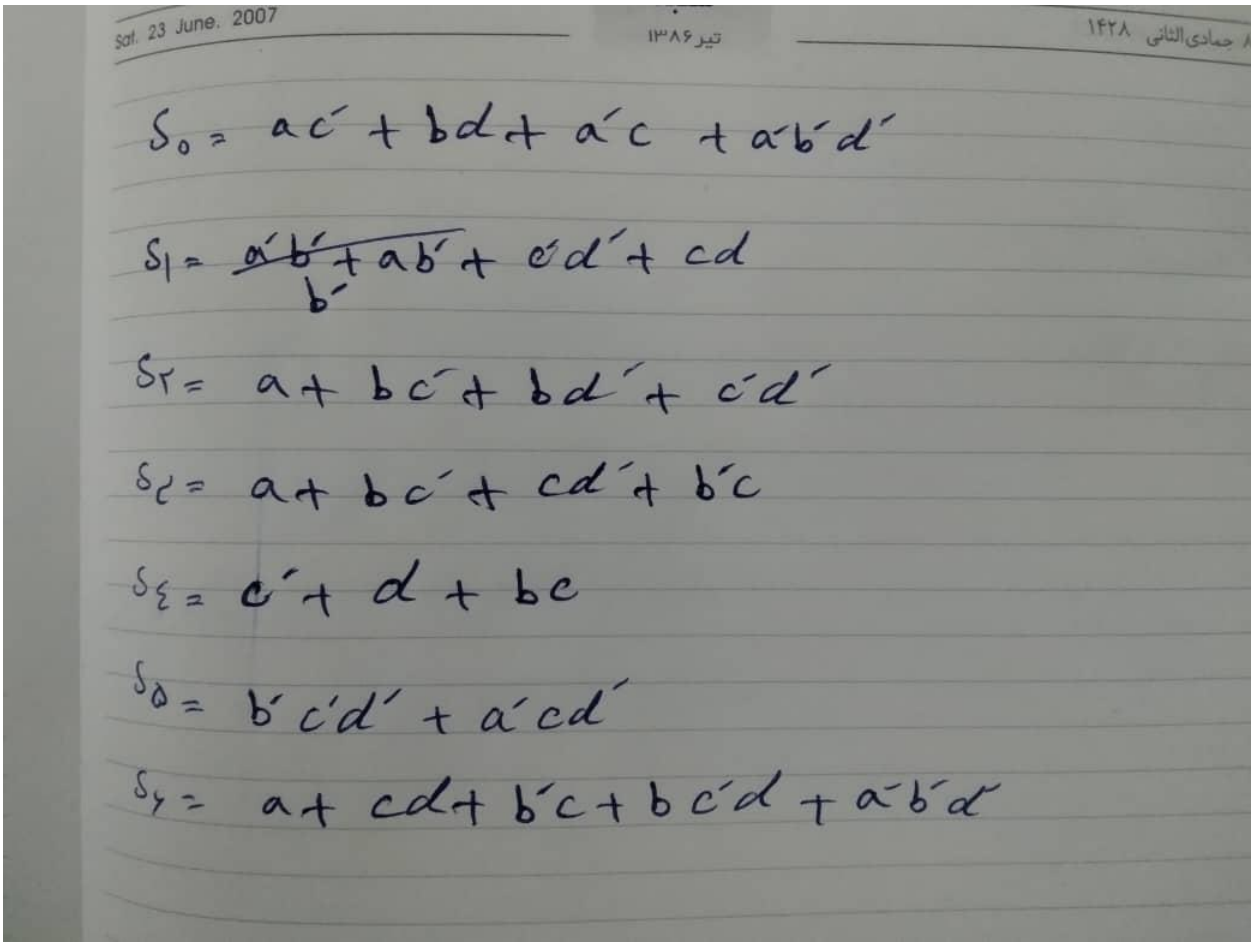

262.667 ns
Name
Value
0 ns
200 ns
400 ns
into[1:0]
10
00
01
10
outo[3:0]
0100
0001
0010
0100

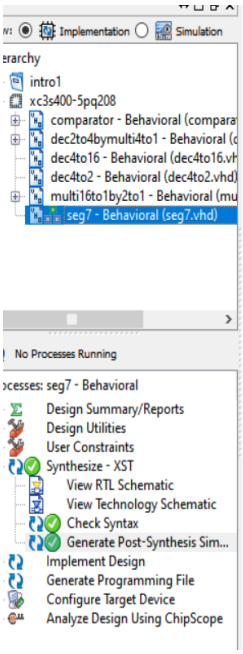
طراحی دیکدر BCD به 7SEGMENT :



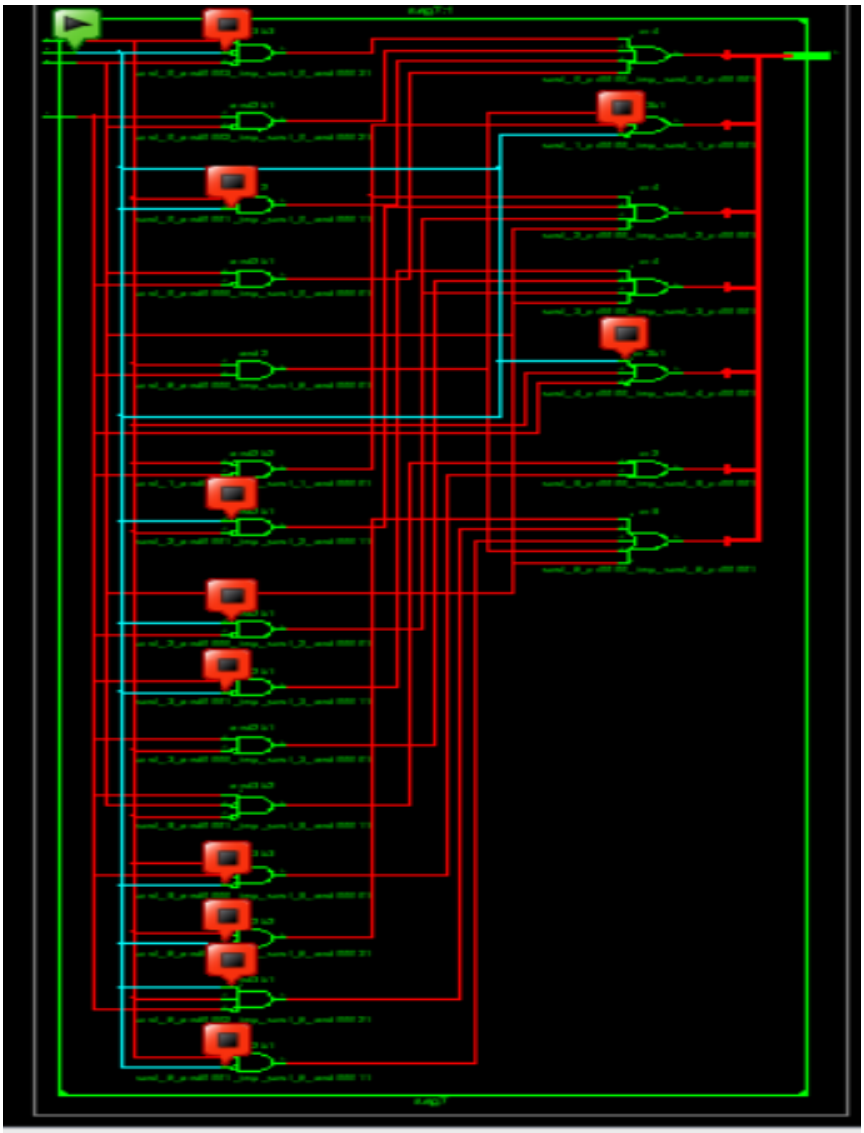


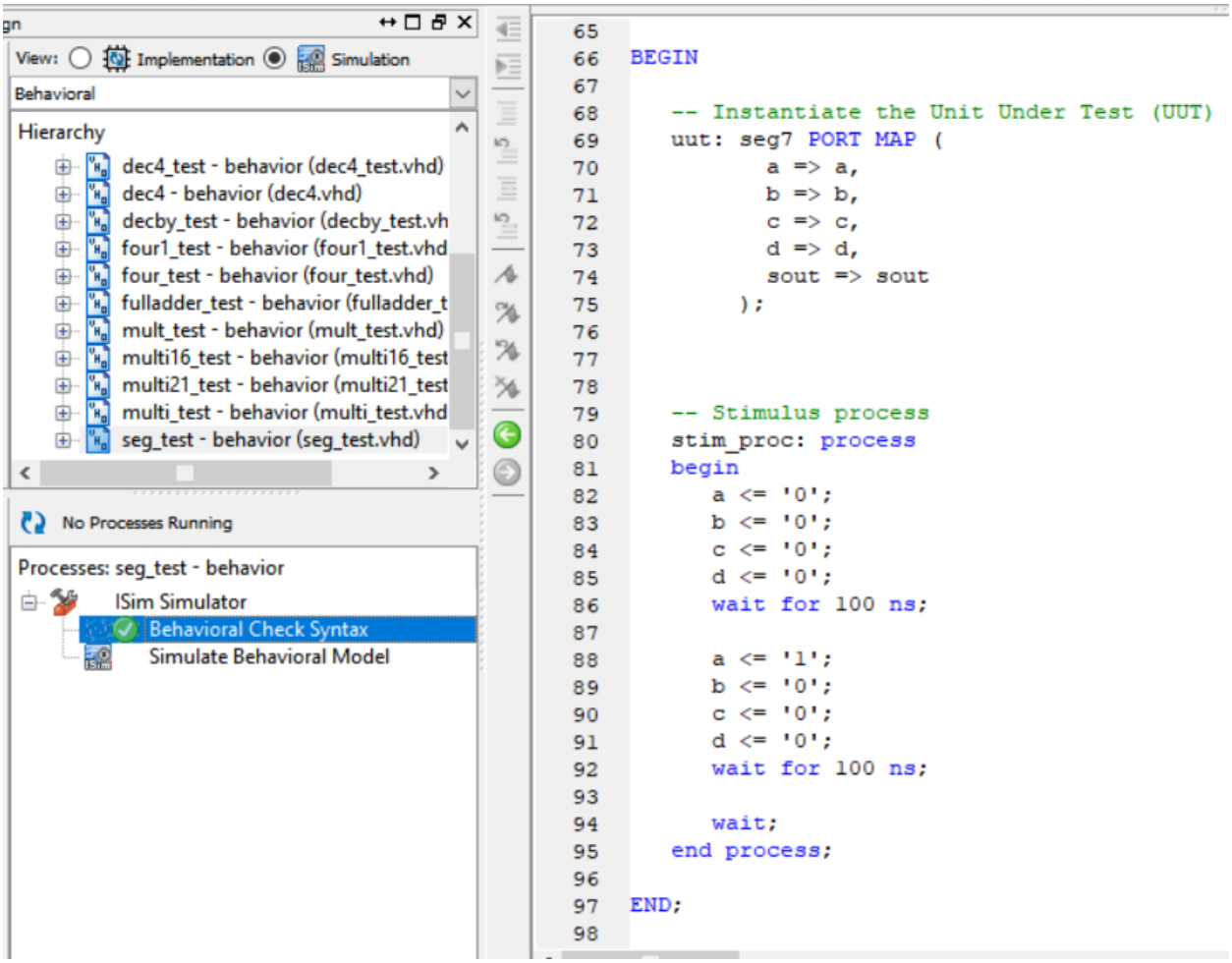
VectorStock®
VectorStock.com/23824162

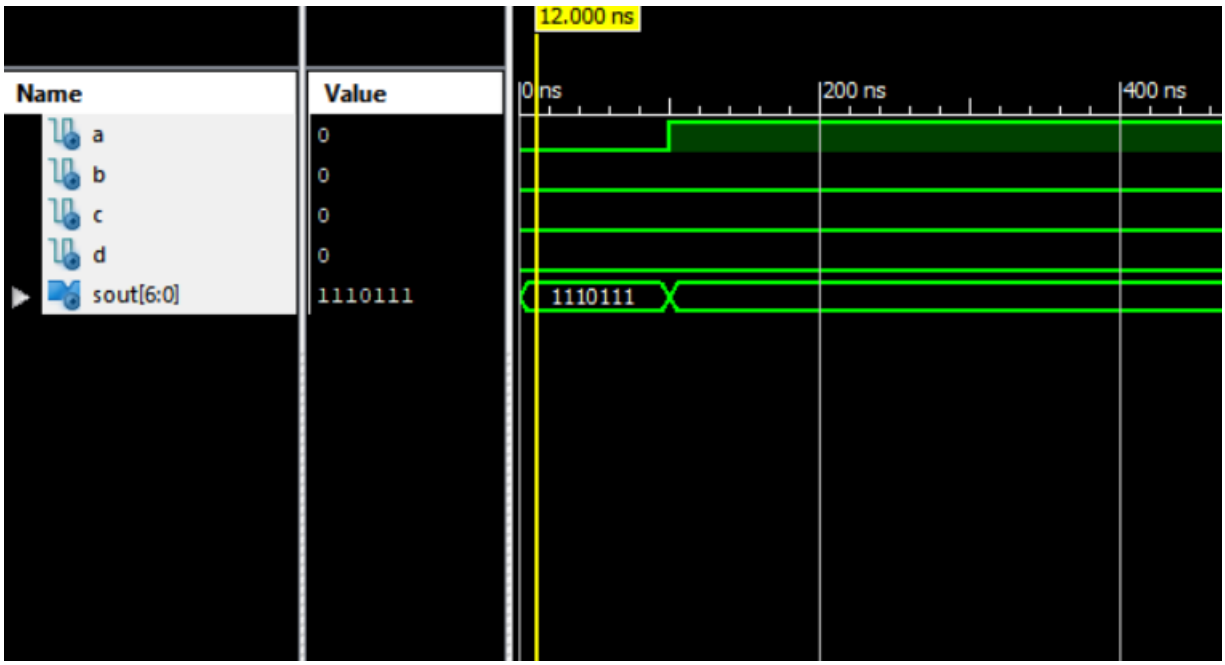





```
21  use IEEE.STD_LOGIC_1164.ALL;
22
23  -- Uncomment the following library declaration if using
24  -- arithmetic functions with Signed or Unsigned values
25  --use IEEE.NUMERIC_STD.ALL;
26
27  -- Uncomment the following library declaration if instantiating
28  -- any Xilinx primitives in this code.
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity seg7 is
33      Port ( a : in  STD_LOGIC ;
34             b : in  STD_LOGIC ;
35             c : in  STD_LOGIC ;
36             d : in  STD_LOGIC ;
37             sout : out  STD_LOGIC_VECTOR (6 downto 0));
38  end seg7;
39
40  architecture Behavioral of seg7 is
41
42  begin
43
44  sout(0) <= (a and (not c)) or (b and d) or ((not a) and c) or ((not a) and (not b) and (not d));
45  sout(1) <= (not b) or ((not c) and (not d)) or (c and d);
46  sout(2) <=  a or (b and (not c)) or (b and (not d)) or ((not c) and (not d)) ;
47  sout(3) <=  a or (b and (not c)) or (c and (not d)) or (c and (not b));
48  sout(4) <=  (not c) or d or (b and c);
49  sout(5) <= ((not b) and (not c) and (not d)) or ((not a) and c and (not d));
50  sout(6) <=  a or (c and d) or ((not b) and d) or (b and (not c) and d) or ((not a) and (not b) and (not d));
51
52  end Behavioral;
53
54
```




View: Implementation Simulation
Behavioral
Hierarchy
dec4_test - behavior (dec4_test.vhd)
dec4 - behavior (dec4.vhd)
decby_test - behavior (decby_test.vh
four1_test - behavior (four1_test.vhd
four_test - behavior (four_test.vhd)
fulladder_test - behavior (fulladder_t
mult_test - behavior (mult_test.vhd)
multi16_test - behavior (multi16_test
multi21_test - behavior (multi21_test
multi_test - behavior (multi_test.vhd
seg_test - behavior (seg_test.vhd)
No Processes Running
Processes: seg_test - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
65
66 BEGIN
67
68 -- Instantiate the Unit Under Test (UUT)
69 uut: seg7 PORT MAP (
70 a => a,
71 b => b,
72 c => c,
73 d => d,
74 sout => sout
75);
76
77
78
79 -- Stimulus process
80 stim_proc: process
81 begin
82 a <= '0';
83 b <= '0';
84 c <= '0';
85 d <= '0';
86 wait for 100 ns;
87
88 a <= '1';
89 b <= '0';
90 c <= '0';
91 d <= '0';
92 wait for 100 ns;
93
94 wait;
95 end process;
96
97 END;
98


Name
Value
a
b
c
d
sout[6:0]
0
0
0
0
1110111
12.000 ns
0 ns
200 ns
400 ns
1110111