آزمایشگاه مدار منطقی

Amirkabir University of Technology
(Tehran Polytechnic)

نویسنده:

علیرضا طباطبائیان(9723052)

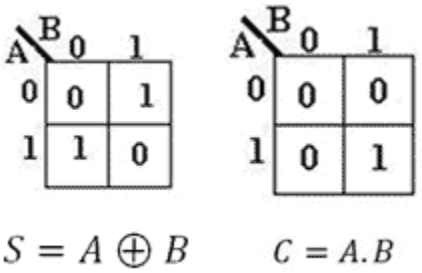
استاد آزمایشگاه:
دکتر علیرضا
طاهری نوید

نیم جمع کننده:

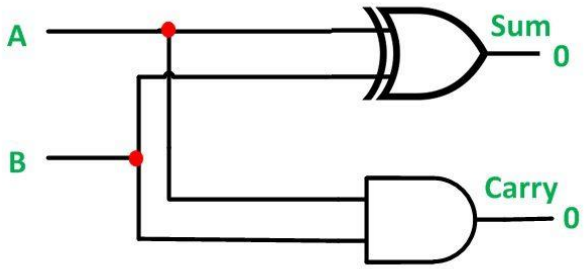
در ابتدا یک پروژه جدید ساخته و یک نیم جمع کننده طراحی میکنیم.

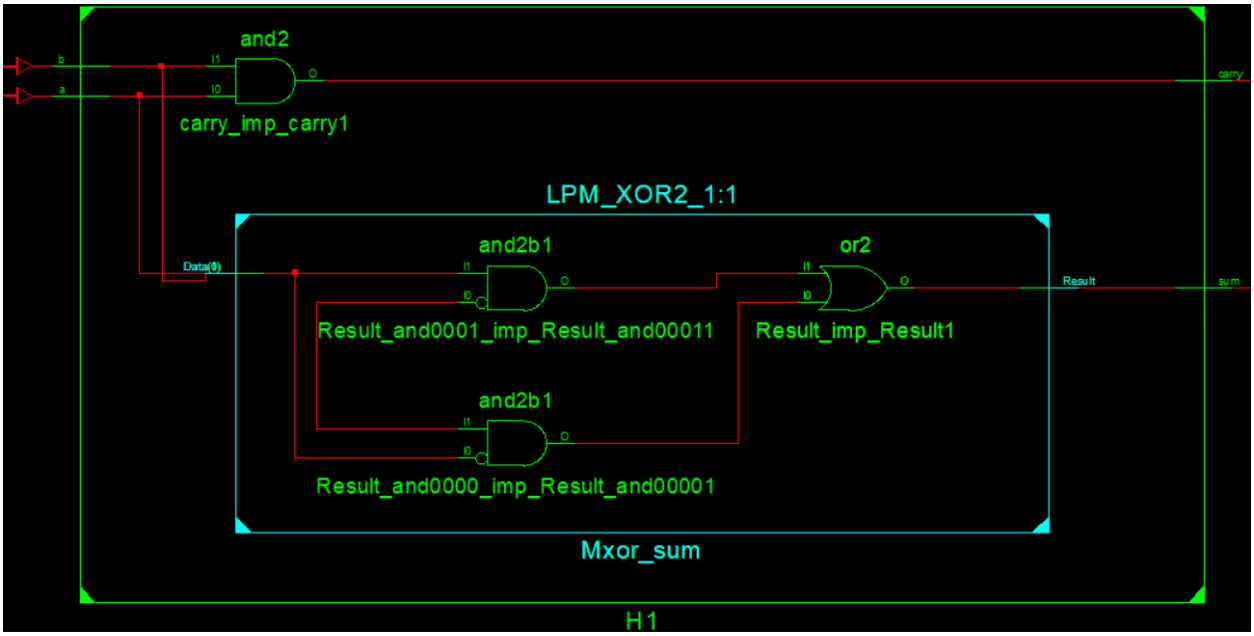
برای طراحی به 2 ورودی و دو خروجی نیاز داریم.

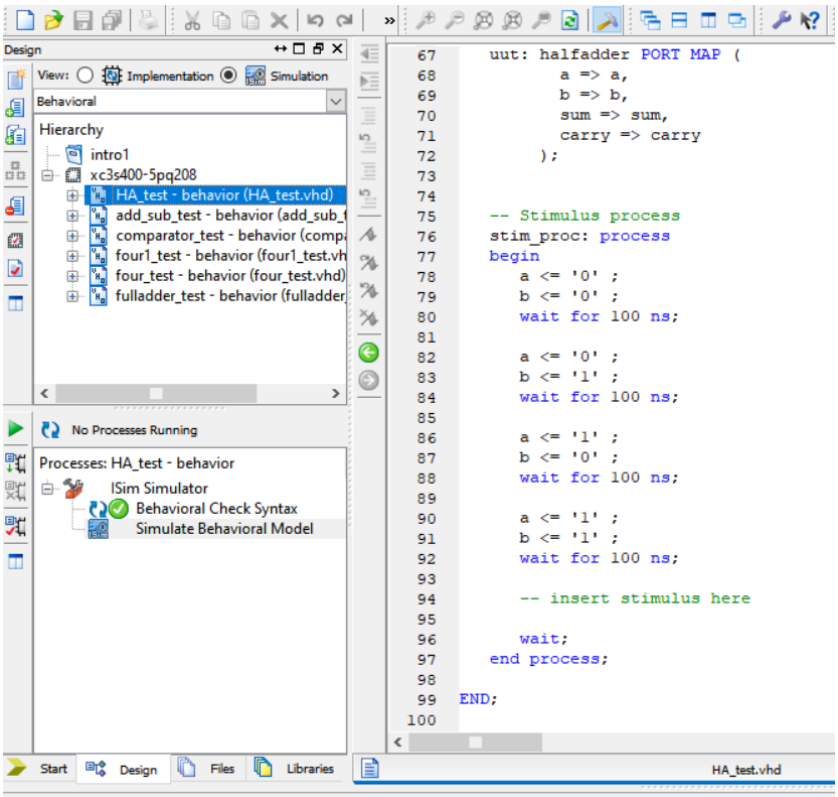
IN1	IN2	SUM	CARRY
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

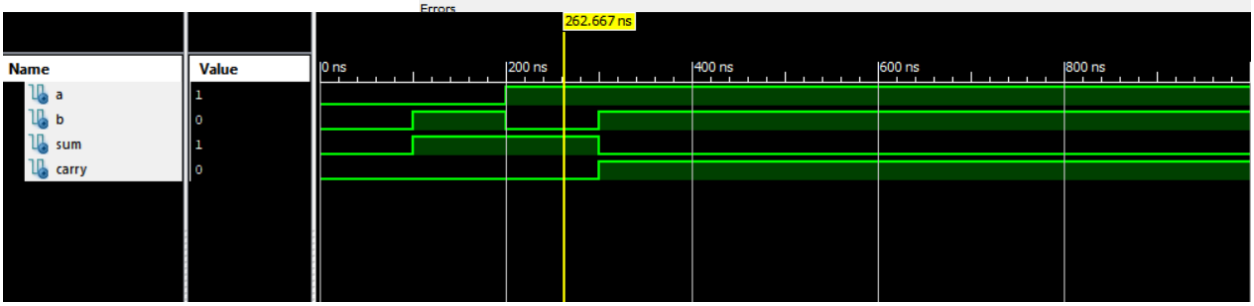


$S = A \oplus B$ $C = A.B$




and2
b
a
I1
I0
O
carry
carry_imp_carry1
LPM_XOR2_1:1
and2b1
or2
Data(0)
Result
sum
Result_and0001_imp_Result_and00011
Result_imp_Result1
and2b1
Result_and0000_imp_Result_and00001
Mxor_sum
H1


Design
View: Implementation Simulation
Behavioral
Hierarchy
intro1
xc3s400-5pq208
HA_test - behavior (HA_test.vhd)
add_sub_test - behavior (add_sub_
comparator_test - behavior (compa
four1_test - behavior (four1_test.vh
four_test - behavior (four_test.vhd)
fulladder_test - behavior (fulladder
No Processes Running
Processes: HA_test - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
Start
Design
Files
Libraries
Errors
HA_test.vhd
67 uut: halfadder PORT MAP (
68 a => a,
69 b => b,
70 sum => sum,
71 carry => carry
72);
73
74
75 -- Stimulus process
76 stim_proc: process
77 begin
78 a <= '0' ;
79 b <= '0' ;
80 wait for 100 ns;
81
82 a <= '0' ;
83 b <= '1' ;
84 wait for 100 ns;
85
86 a <= '1' ;
87 b <= '0' ;
88 wait for 100 ns;
89
90 a <= '1' ;
91 b <= '1' ;
92 wait for 100 ns;
93
94 -- insert stimulus here
95
96 wait;
97 end process;
98
99 END;
100

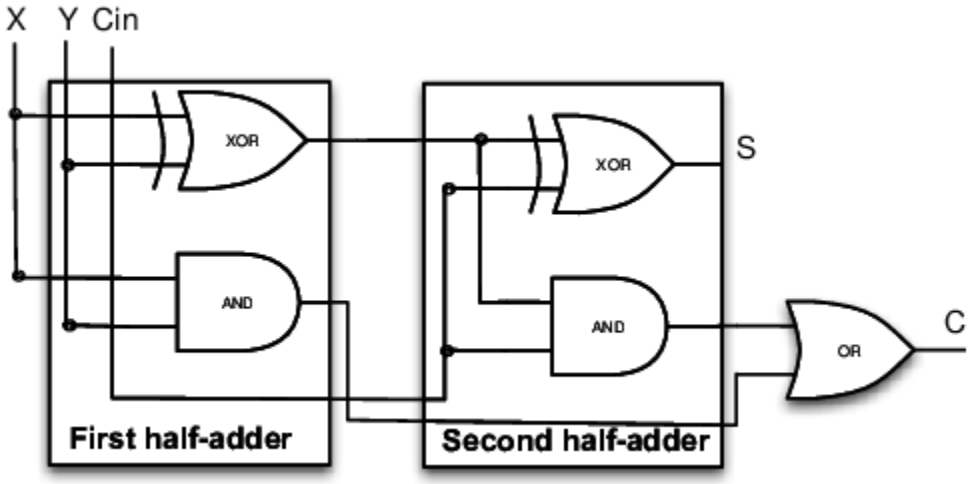

262.667 ns
Name
Value
a
b
sum
carry
1
0
1
0
0 ns
200 ns
400 ns
600 ns
800 ns

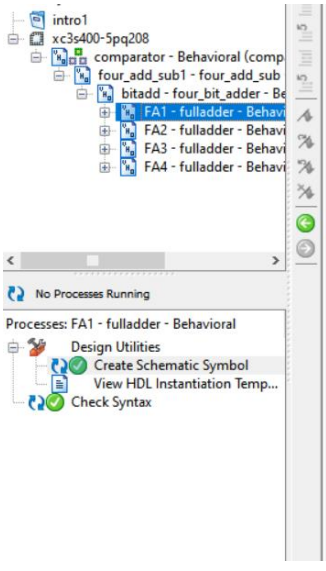
ساخت تمام جمع کننده با نیم جمع کننده:

3 ورودی و 2 خروجی مورد نیاز است.

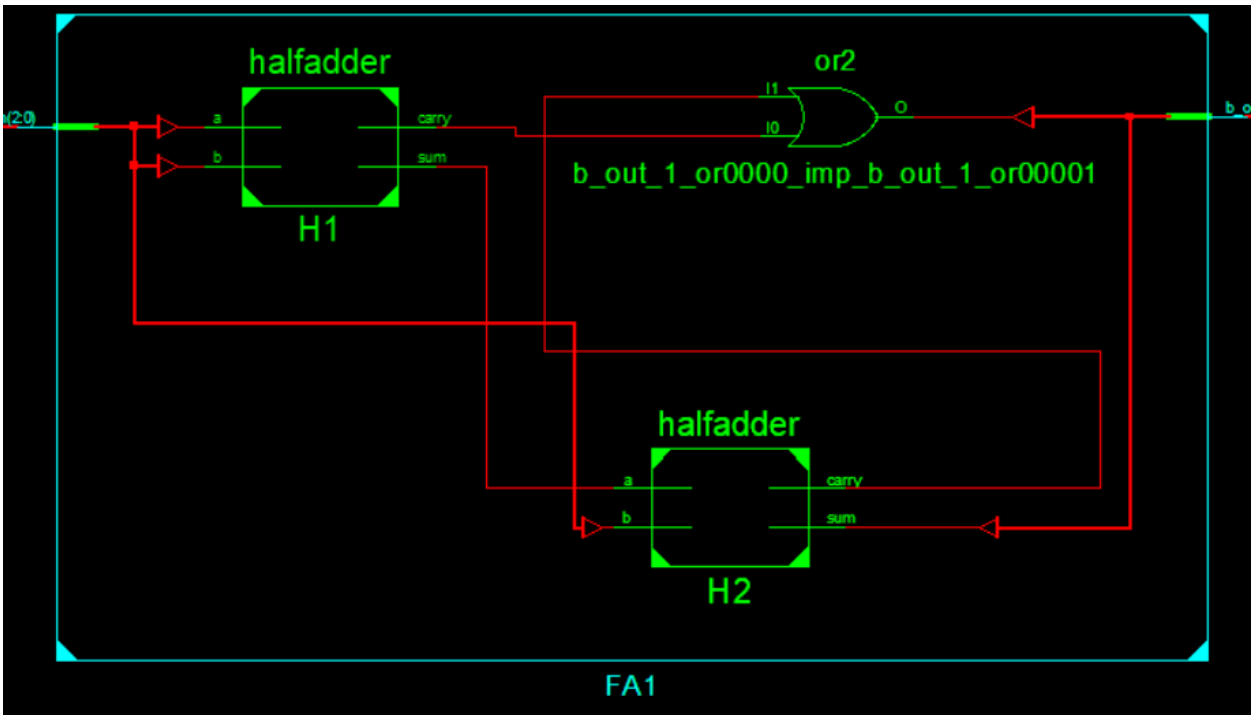
همچنین نیاز به استفاده از signal و port map میباشد زیرا میخواهیم 2 آی سی را به هم متصل کنیم.

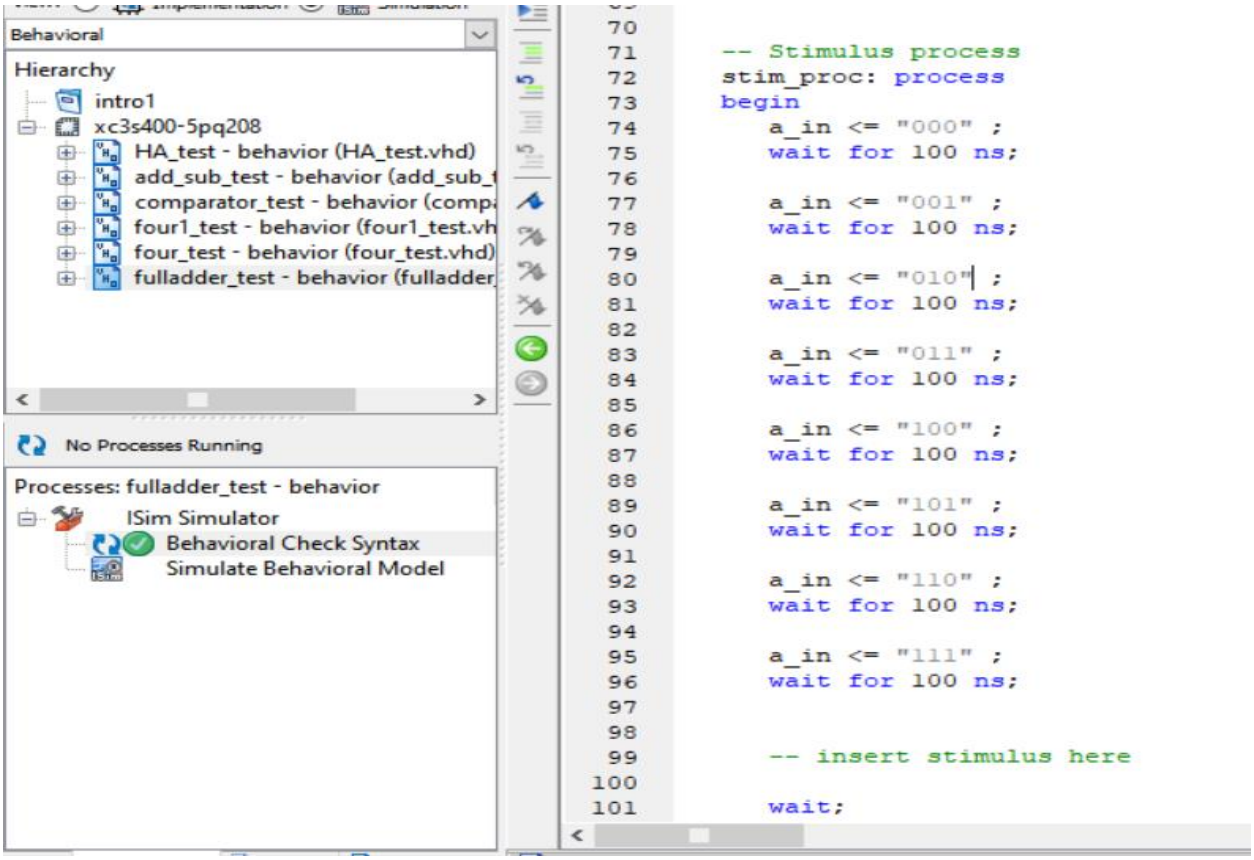
همچنین برای استفاده از یک آی سی در مداری دیگر، نیاز به استفاده از component میباشد که از بخش تست بنچ آن آی سی کپی میشود.



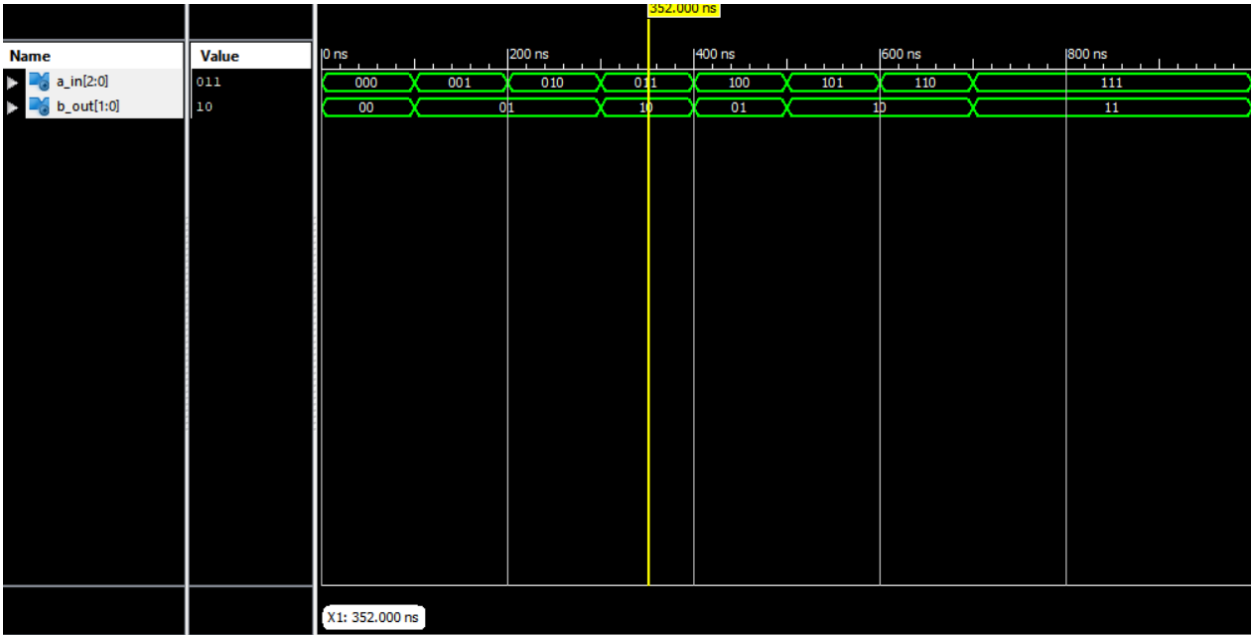



```
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity fulladder is
33      Port ( a_in : in  STD_LOGIC_VECTOR (2 downto 0);
34             b_out : out  STD_LOGIC_VECTOR (1 downto 0));
35  end fulladder;
36
37  architecture Behavioral of fulladder is
38
39      COMPONENT halfadder
40      PORT(
41           a : IN  std_logic;
42           b : IN  std_logic;
43           sum : OUT  std_logic;
44           carry : OUT  std_logic
45          );
46      END COMPONENT;
47
48      signal w1,w2,w3 : std_logic :='0';
49
50  begin
51
52     H1 : halfadder port map( a => a_in(0) , b => a_in(1) , sum => w1 , carry => w2 );
53     H2 : halfadder port map( a => w1      , b => a_in(2) , sum => b_out(0) , carry => w3 );
54     b_out(1) <= w2 or w3;
55
56
57  end Behavioral;
58
59
```

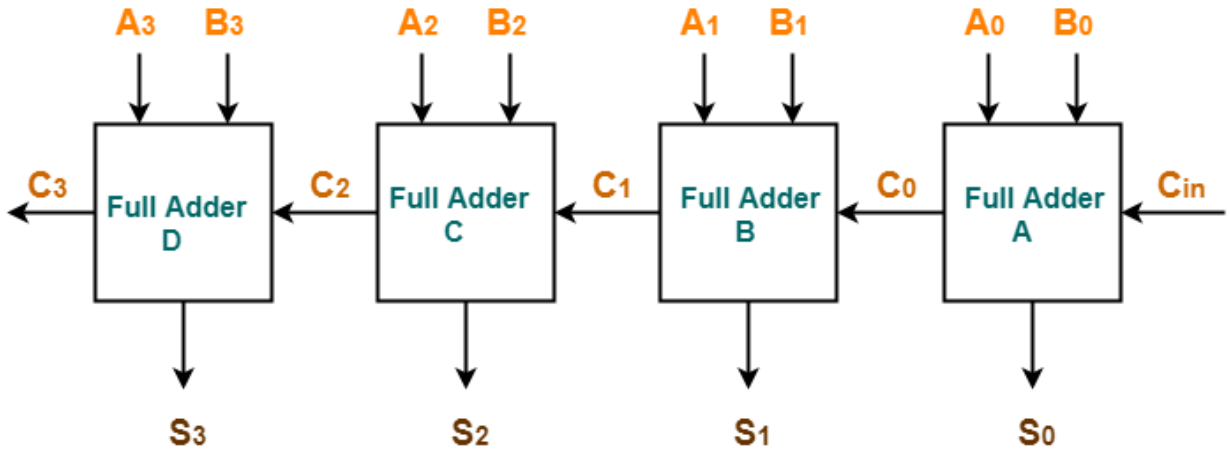




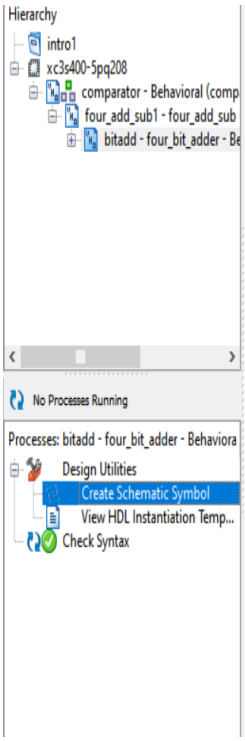

```
70
71      -- Stimulus process
72      stim_proc: process
73      begin
74          a_in <= "000" ;
75          wait for 100 ns;
76
77          a_in <= "001" ;
78          wait for 100 ns;
79
80          a_in <= "010" ;
81          wait for 100 ns;
82
83          a_in <= "011" ;
84          wait for 100 ns;
85
86          a_in <= "100" ;
87          wait for 100 ns;
88
89          a_in <= "101" ;
90          wait for 100 ns;
91
92          a_in <= "110" ;
93          wait for 100 ns;
94
95          a_in <= "111" ;
96          wait for 100 ns;
97
98
99          -- insert stimulus here
100
101         wait;
```



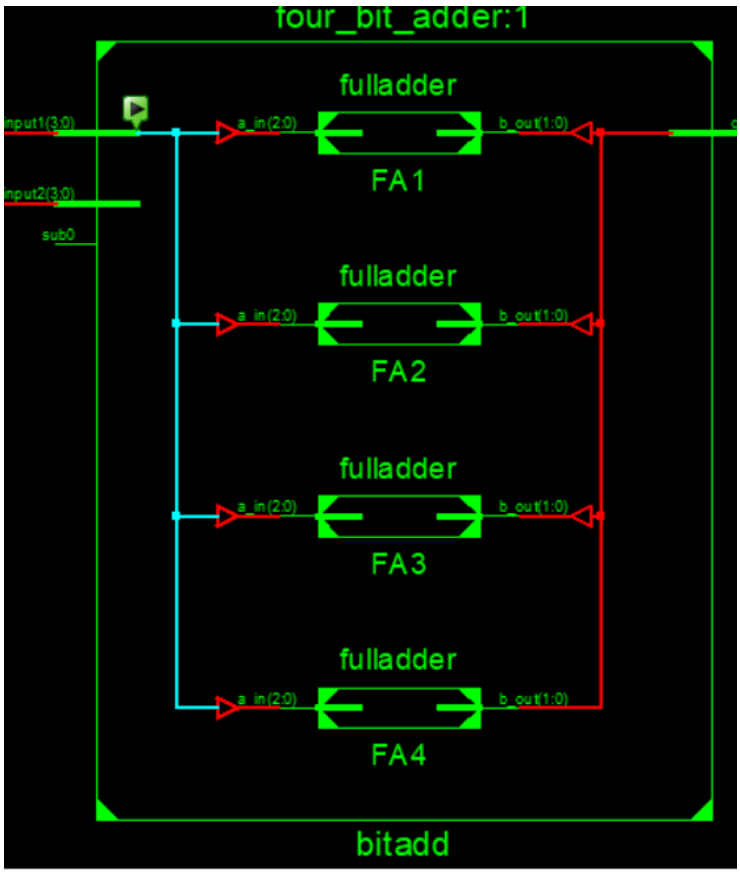
مدار بعدی جمع کننده 4 بیتی: مقدار sub را برابر صفر میگذاریم تا به جمع کننده تبدیل شود.

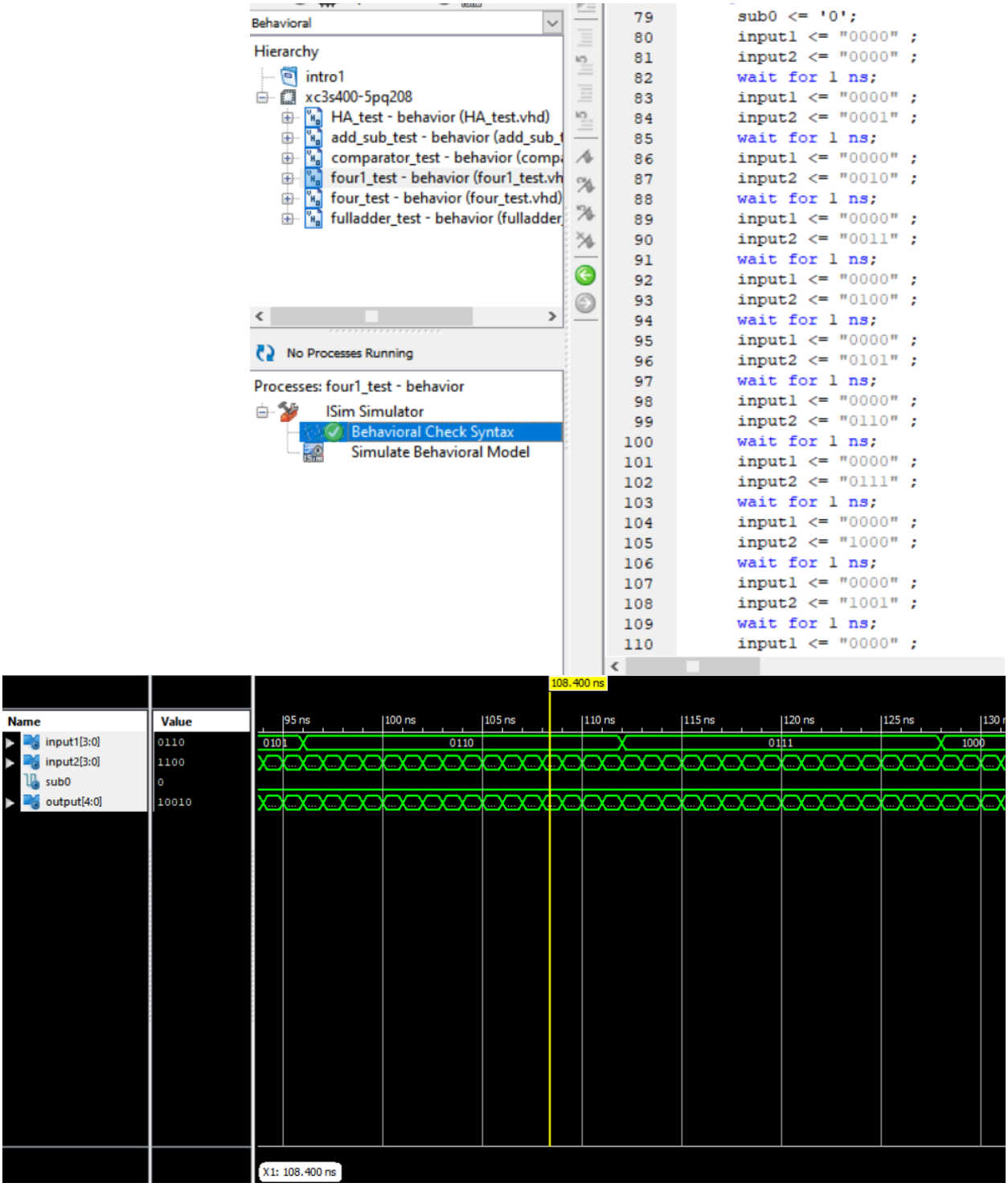


4-bit Ripple Carry Adder

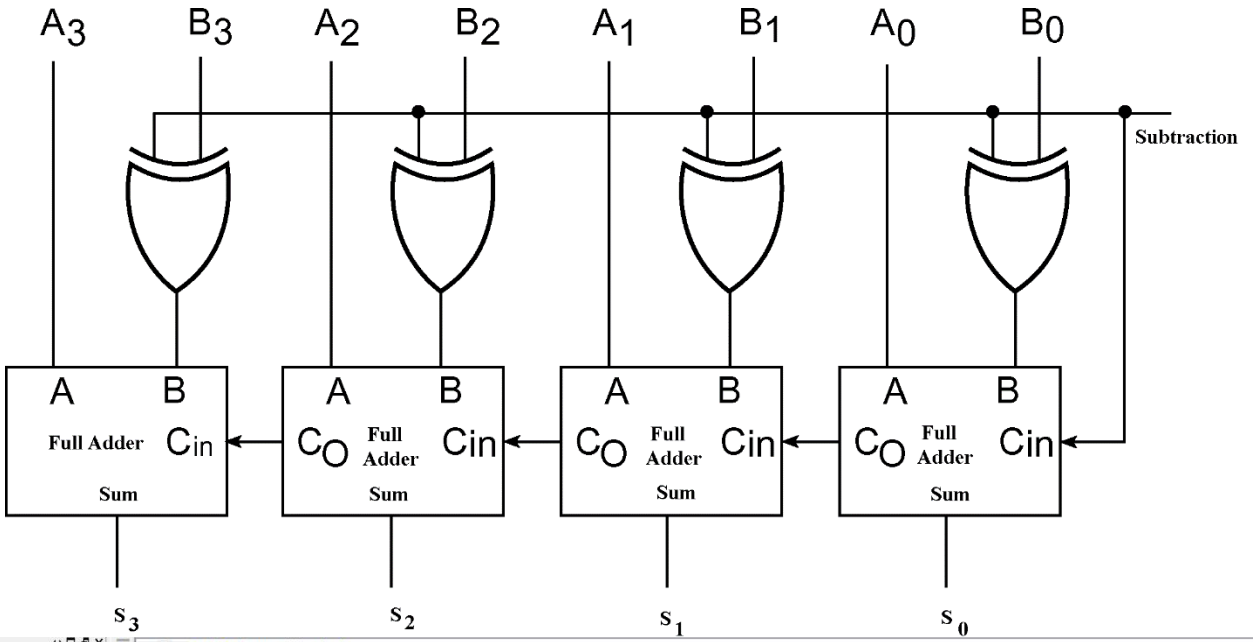


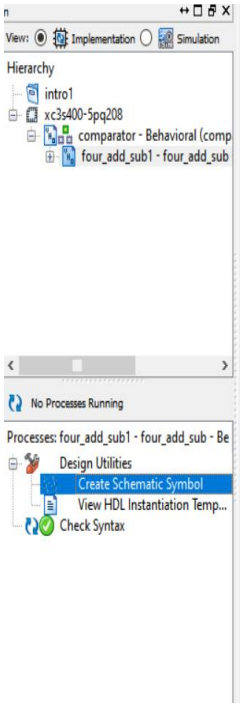

```
32  entity four_bit_adder is
33      Port ( input1 : in  STD_LOGIC_VECTOR (3 downto 0);
34             input2 : in  STD_LOGIC_VECTOR (3 downto 0);
35             sub0   : in  std_logic;
36             output : out  STD_LOGIC_VECTOR (4 downto 0));
37  end four_bit_adder;
38
39  architecture Behavioral of four_bit_adder is
40
41
42      COMPONENT fulladder
43      PORT(
44           a_in : IN  std_logic_vector(2 downto 0);
45           b_out : OUT  std_logic_vector(1 downto 0)
46          );
47      END COMPONENT;
48
49  signal w1,w2,w3 : std_logic :='0';
50
51  begin
52
53  FA1 : fulladder port map( a_in(0) => input1(0) , a_in(1) => input2(0) , a_in(2) => sub0  , b_out(0) => output(0) , b_out(1) => w1 );
54  FA2 : fulladder port map( a_in(0) => input1(1) , a_in(1) => input2(1) , a_in(2) => w1  , b_out(0) => output(1) , b_out(1) => w2 );
55  FA3 : fulladder port map( a_in(0) => input1(2) , a_in(1) => input2(2) , a_in(2) => w2  , b_out(0) => output(2) , b_out(1) => w3 );
56  FA4 : fulladder port map( a_in(0) => input1(3) , a_in(1) => input2(3) , a_in(2) => w3  , b_out(0) => output(3) , b_out(1) => output(4) );
57
58
59
60  end Behavioral;
61
```



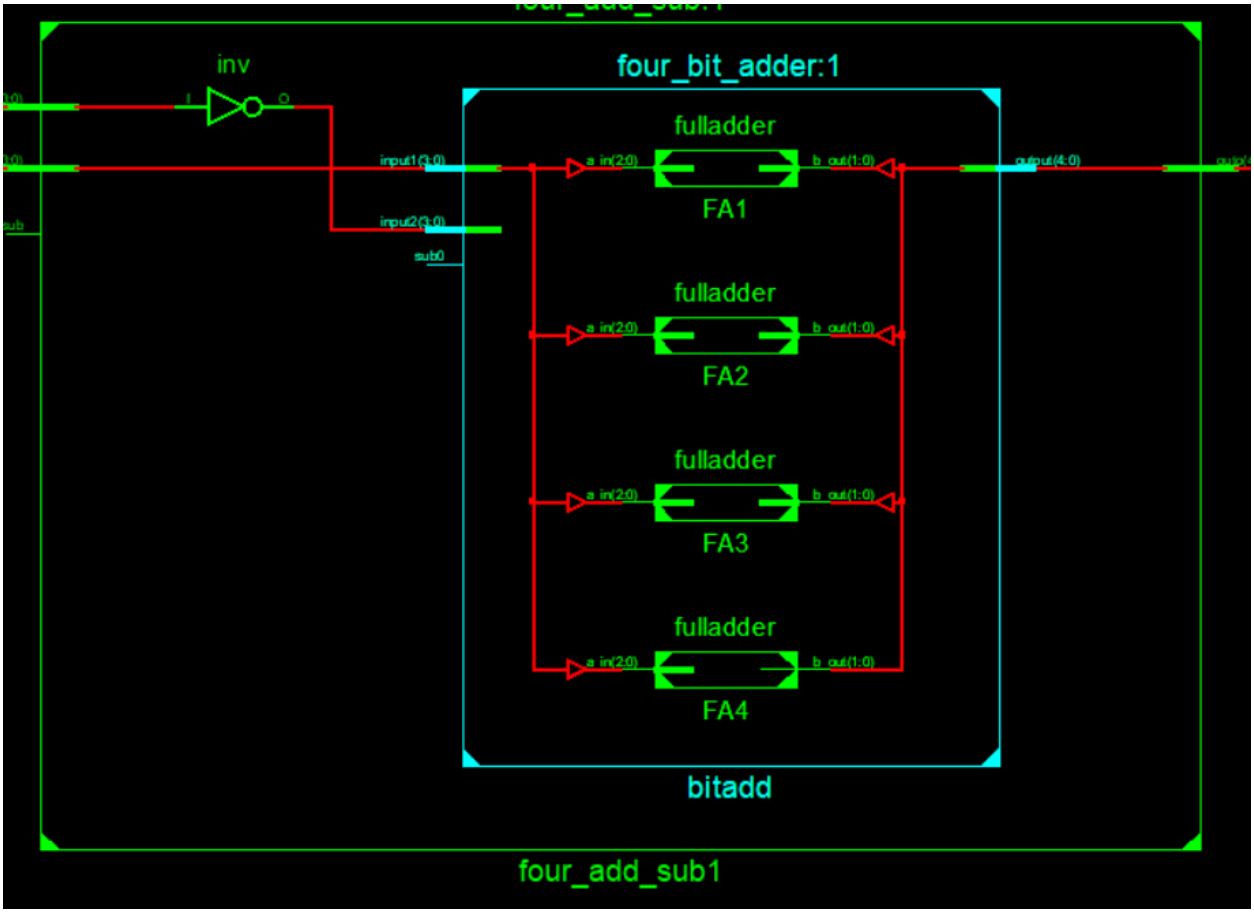

Behavioral
Hierarchy
intro1
xc3s400-5pq208
HA_test - behavior (HA_test.vhd)
add_sub_test - behavior (add_sub_t
comparator_test - behavior (compa
four1_test - behavior (four1_test.vh
four_test - behavior (four_test.vhd)
fulladder_test - behavior (fulladder
No Processes Running
Processes: four1_test - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
79 sub0 <= '0';
80 input1 <= "0000" ;
81 input2 <= "0000" ;
82 wait for 1 ns;
83 input1 <= "0000" ;
84 input2 <= "0001" ;
85 wait for 1 ns;
86 input1 <= "0000" ;
87 input2 <= "0010" ;
88 wait for 1 ns;
89 input1 <= "0000" ;
90 input2 <= "0011" ;
91 wait for 1 ns;
92 input1 <= "0000" ;
93 input2 <= "0100" ;
94 wait for 1 ns;
95 input1 <= "0000" ;
96 input2 <= "0101" ;
97 wait for 1 ns;
98 input1 <= "0000" ;
99 input2 <= "0110" ;
100 wait for 1 ns;
101 input1 <= "0000" ;
102 input2 <= "0111" ;
103 wait for 1 ns;
104 input1 <= "0000" ;
105 input2 <= "1000" ;
106 wait for 1 ns;
107 input1 <= "0000" ;
108 input2 <= "1001" ;
109 wait for 1 ns;
110 input1 <= "0000" ;
108.400 ns
Name
Value
input1[3:0]
0110
input2[3:0]
1100
sub0
0
output[4:0]
10010
95 ns
100 ns
105 ns
110 ns
115 ns
120 ns
125 ns
0101
0110
0111
1000
X1: 108.400 ns

جمع کننده و تفاضل گیر: مقدار sub را اگر 0 بگذاریم جمع کننده و اگر 1 بگذاریم تفریق کننده است.

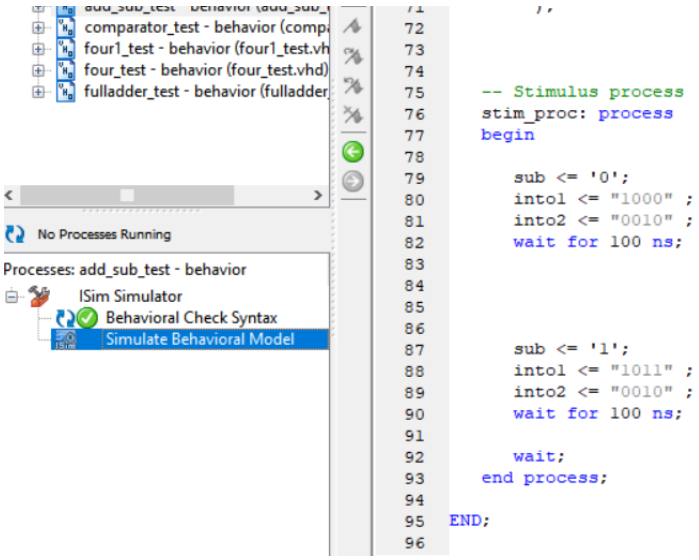




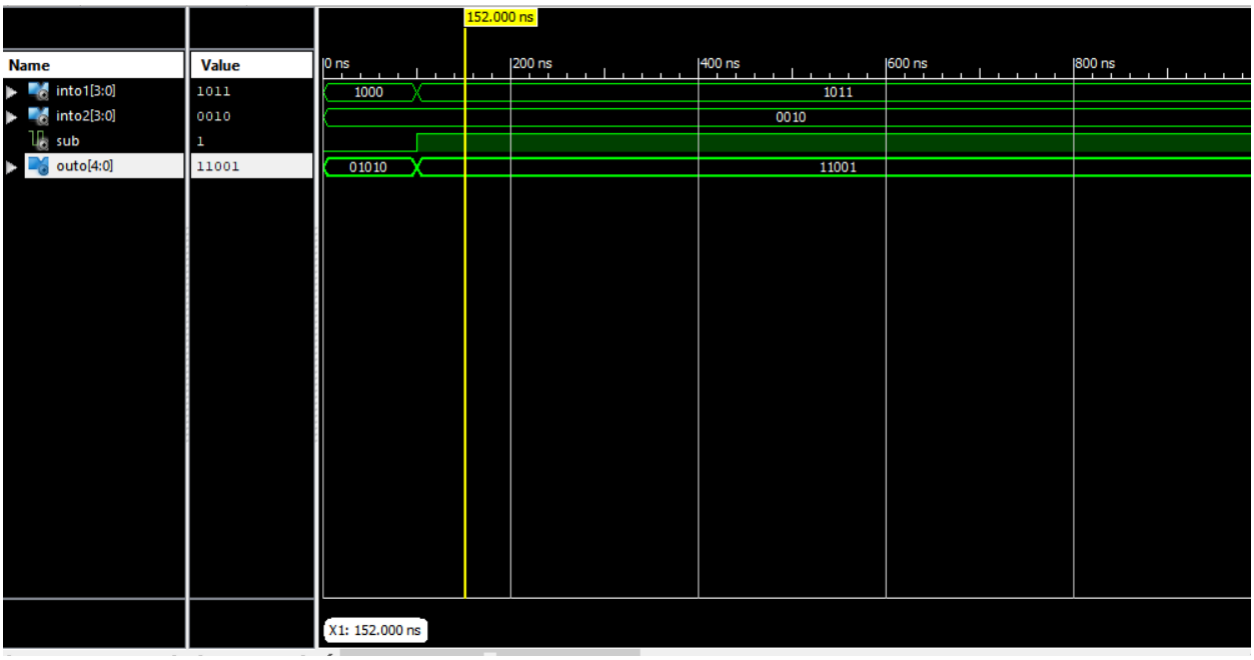

```
32  entity four_add_sub is
33      Port ( into1 : in  STD_LOGIC_VECTOR (3 downto 0);
34             into2 : in  STD_LOGIC_VECTOR (3 downto 0);
35             sub   : in  std_logic ;
36             outo  : out STD_LOGIC_VECTOR (4 downto 0));
37  end four_add_sub;
38
39  architecture Behavioral of four_add_sub is
40
41    COMPONENT four_bit_adder
42     PORT(
43          input1 : IN  std_logic_vector(3 downto 0);
44          input2 : IN  std_logic_vector(3 downto 0);
45          sub0 : IN  std_logic;
46          output : OUT  std_logic_vector(4 downto 0)
47         );
48     END COMPONENT;
49
50     signal w0,w1,w2,w3,w4 : std_logic :='0';
51
52  begin
53
54  w0 <= into2(0) xor sub ;
55  w1 <= into2(1) xor sub ;
56  w2 <= into2(2) xor sub ;
57  w3 <= into2(3) xor sub ;
58
59  bitadd : four_bit_adder port map( input1(0) => into1(0) , input1(1) => into1(1) , input1(2) => into1(2) , input1(3) => into1(3) , sub0 => sub ,
60                                    input2(0) => w0 , input2(1) => w1 , input2(2) => w2 , input2(3) => w3 ,
61                                   output(0) => outo(0) , output(1) => outo(1) ,output(2) => outo(2) ,output(3) => outo(3) ,output(4) => w4 );
62
63
64  end Behavioral;
```



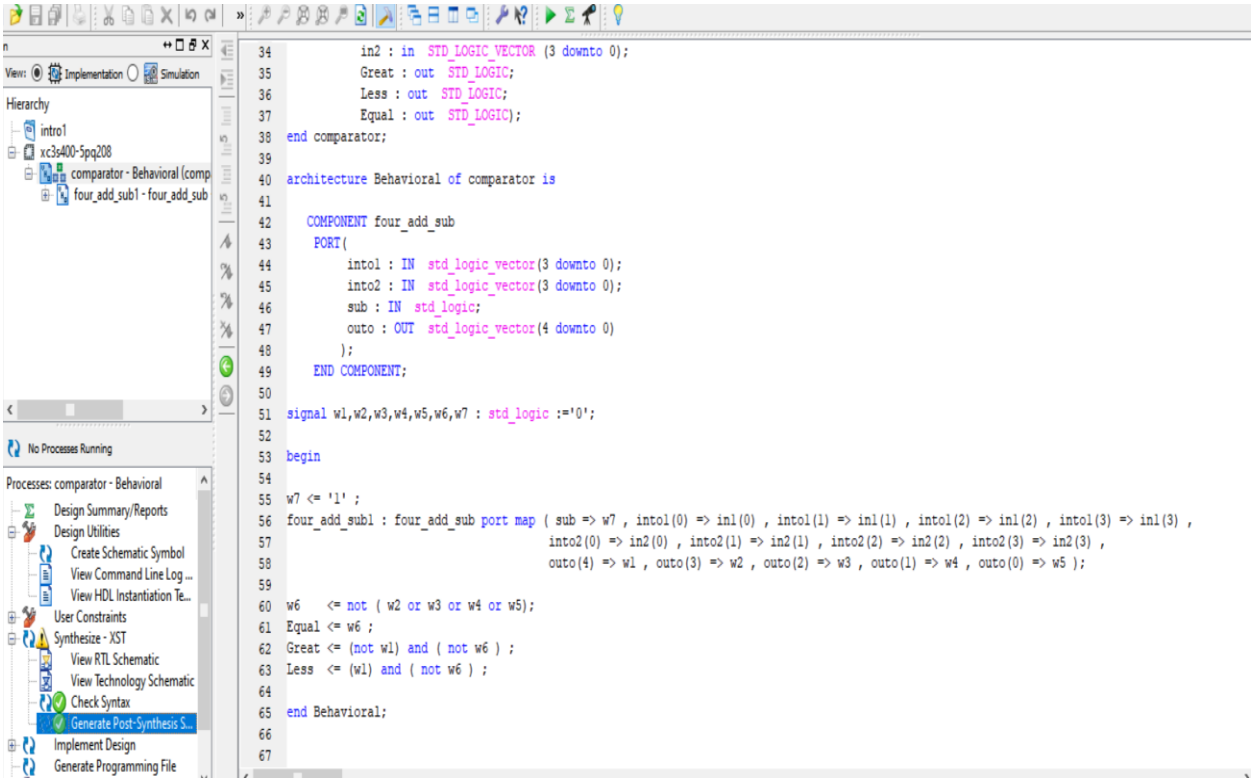
فقط دو ورودی میدهیم یکی برای جمع و یکی برای تفریق:



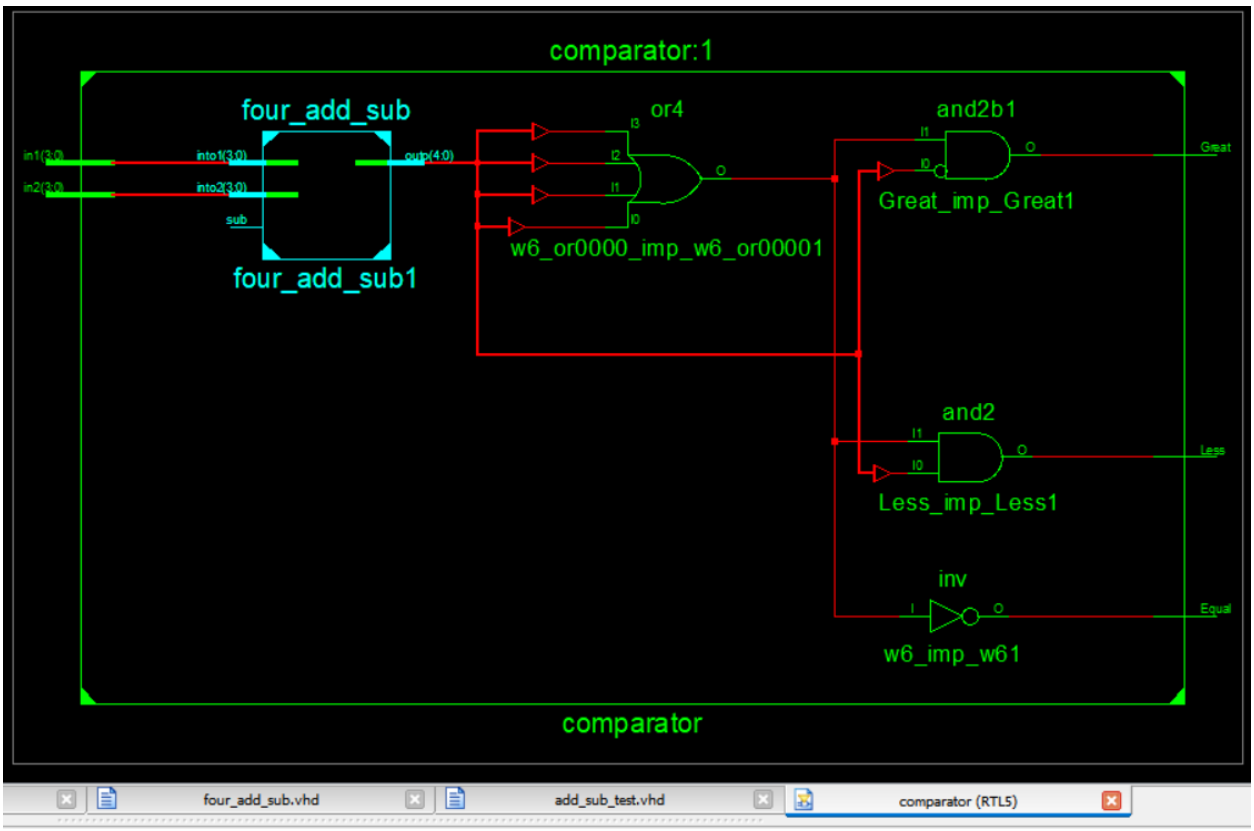

```
72
73
74
75      -- Stimulus process
76      stim_proc: process
77      begin
78
79          sub <= '0';
80          into1 <= "1000" ;
81          into2 <= "0010" ;
82          wait for 100 ns;
83
84
85
86
87          sub <= '1';
88          into1 <= "1011" ;
89          into2 <= "0010" ;
90          wait for 100 ns;
91
92          wait;
93      end process;
94
95  END;
96
```

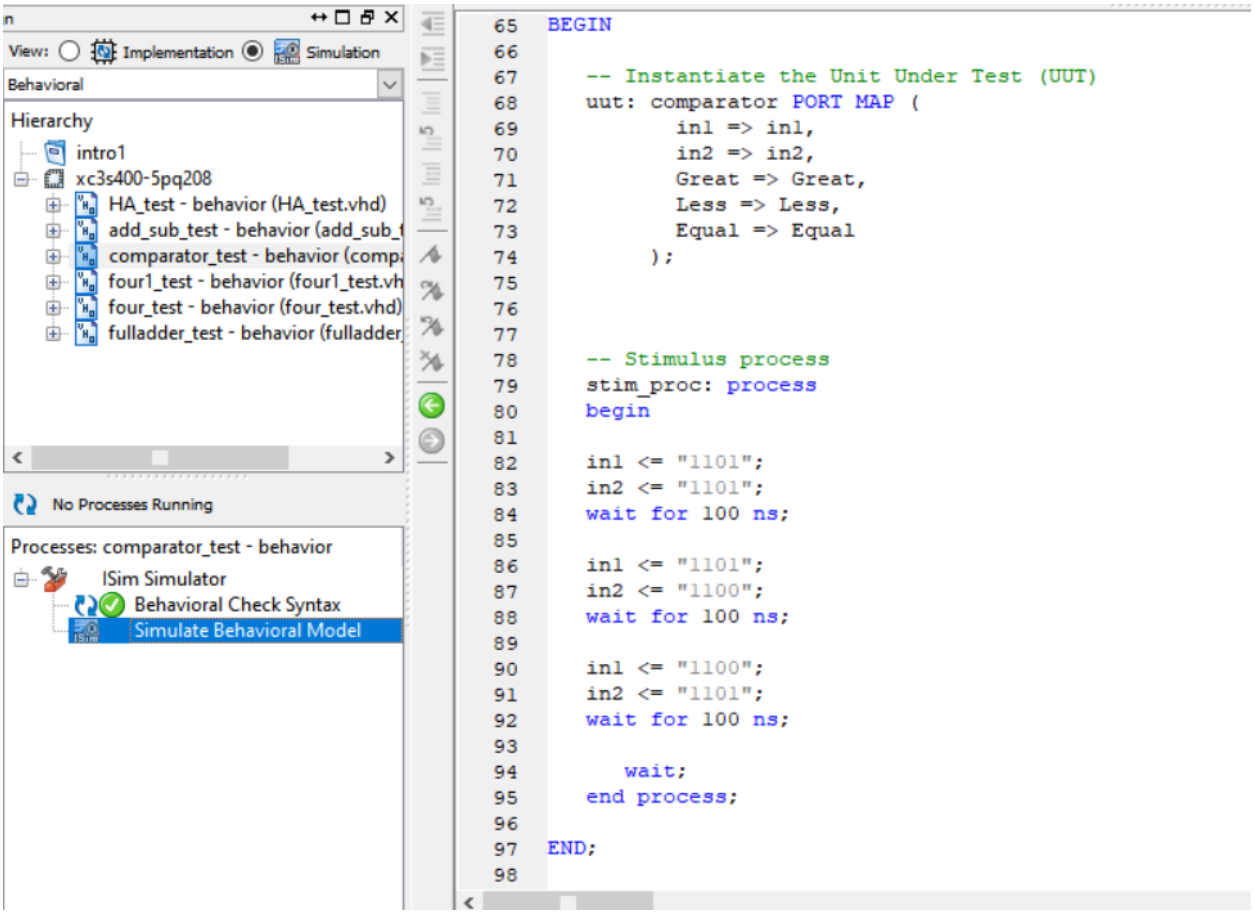


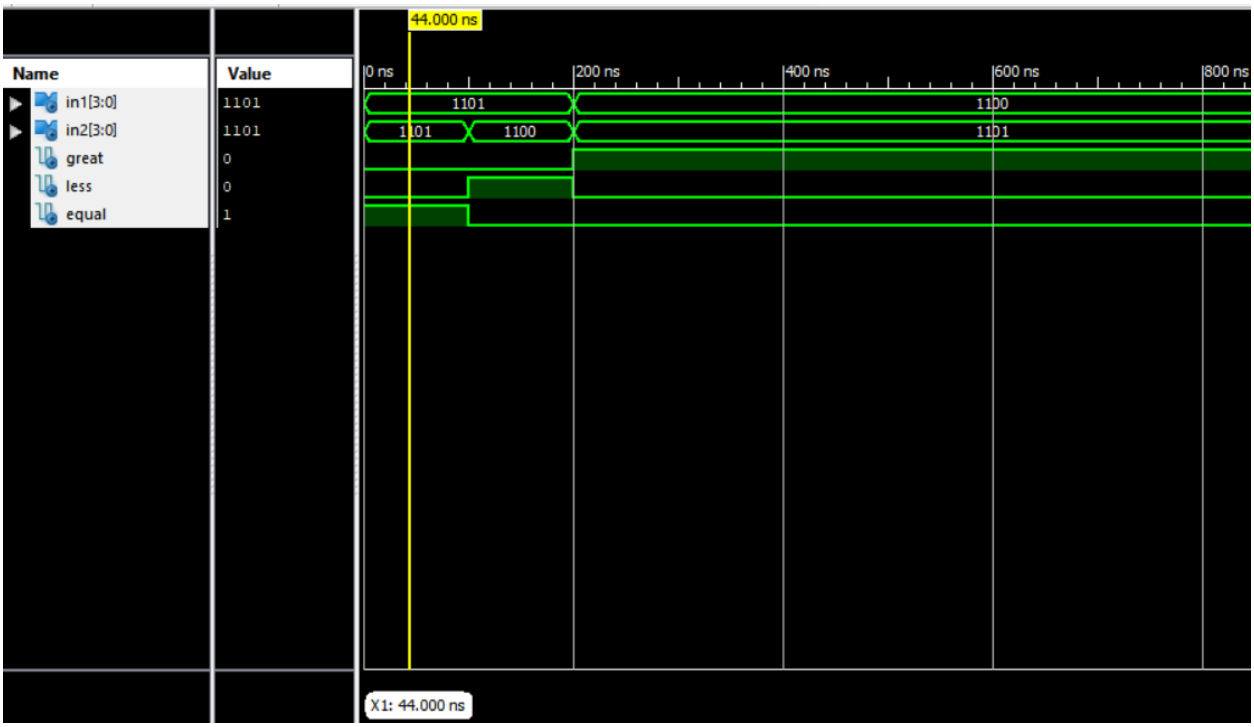
تمرین امتیازی:



```
34          in2 : in  STD_LOGIC_VECTOR (3 downto 0);
35          Great : out  STD_LOGIC;
36          Less : out  STD_LOGIC;
37          Equal : out  STD_LOGIC);
38  end comparator;
39
40  architecture Behavioral of comparator is
41
42    COMPONENT four_add_sub
43     PORT(
44          into1 : IN  std_logic_vector(3 downto 0);
45          into2 : IN  std_logic_vector(3 downto 0);
46          sub : IN  std_logic;
47          outo : OUT  std_logic_vector(4 downto 0)
48          );
49     END COMPONENT;
50
51  signal w1,w2,w3,w4,w5,w6,w7 : std_logic :='0';
52
53  begin
54
55  w7 <= '1' ;
56  four_add_sub1 : four_add_sub port map ( sub => w7 , into1(0) => in1(0) , into1(1) => in1(1) , into1(2) => in1(2) , into1(3) => in1(3) ,
57                                          into2(0) => in2(0) , into2(1) => in2(1) , into2(2) => in2(2) , into2(3) => in2(3) ,
58                                          outo(4) => w1 , outo(3) => w2 , outo(2) => w3 , outo(1) => w4 , outo(0) => w5 );
59
60  w6    <= not ( w2 or w3 or w4 or w5);
61  Equal <= w6 ;
62  Great <= (not w1) and ( not w6 ) ;
63  Less  <= (w1) and ( not w6 ) ;
64
65  end Behavioral;
66
67
```


comparator:1
four_add_sub
in1(3:0)
in2(3:0)
into1(3:0)
into2(3:0)
sub
outp(4:0)
four_add_sub1
or4
I3
I2
I1
I0
O
w6_or0000_imp_w6_or00001
and2b1
I1
I0
O
Great
Great_imp_Great1
and2
I1
I0
O
Less
Less_imp_Less1
inv
I
O
Equal
w6_imp_w61
comparator
four_add_sub.vhd
add_sub_test.vhd
comparator (RTL5)


View: Implementation Simulation
Behavioral
Hierarchy
intro1
xc3s400-5pq208
HA_test - behavior (HA_test.vhd)
add_sub_test - behavior (add_sub_t
comparator_test - behavior (compa
four1_test - behavior (four1_test.vh
four_test - behavior (four_test.vhd)
fulladder_test - behavior (fulladder
No Processes Running
Processes: comparator_test - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model
65 BEGIN
66
67 -- Instantiate the Unit Under Test (UUT)
68 uut: comparator PORT MAP (
69 in1 => in1,
70 in2 => in2,
71 Great => Great,
72 Less => Less,
73 Equal => Equal
74);
75
76
77
78 -- Stimulus process
79 stim_proc: process
80 begin
81
82 in1 <= "1101";
83 in2 <= "1101";
84 wait for 100 ns;
85
86 in1 <= "1101";
87 in2 <= "1100";
88 wait for 100 ns;
89
90 in1 <= "1100";
91 in2 <= "1101";
92 wait for 100 ns;
93
94 wait;
95 end process;
96
97 END;
98


44.000 ns
Name
Value
in1[3:0]
1101
in2[3:0]
1101
great
0
less
0
equal
1
0 ns
200 ns
400 ns
600 ns
800 ns
1101
1100
1101
1100
1101
X1: 44.000 ns