

نویسنده:

علیرضا طباطبائی (9723052)

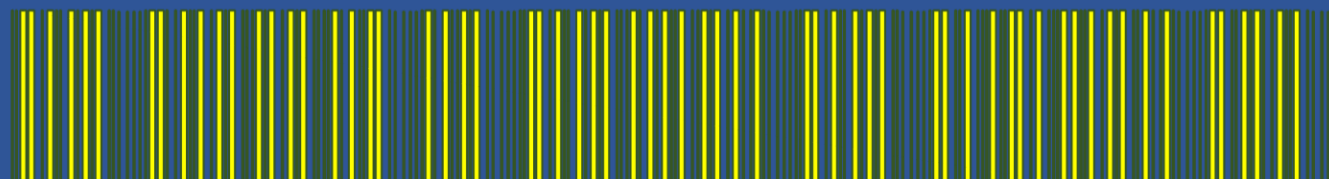


Amirkabir University of Technology  
(Tehran Polytechnic)

استاد آزمایشگاه:

دکتر علیرضا

ظاهری نوید



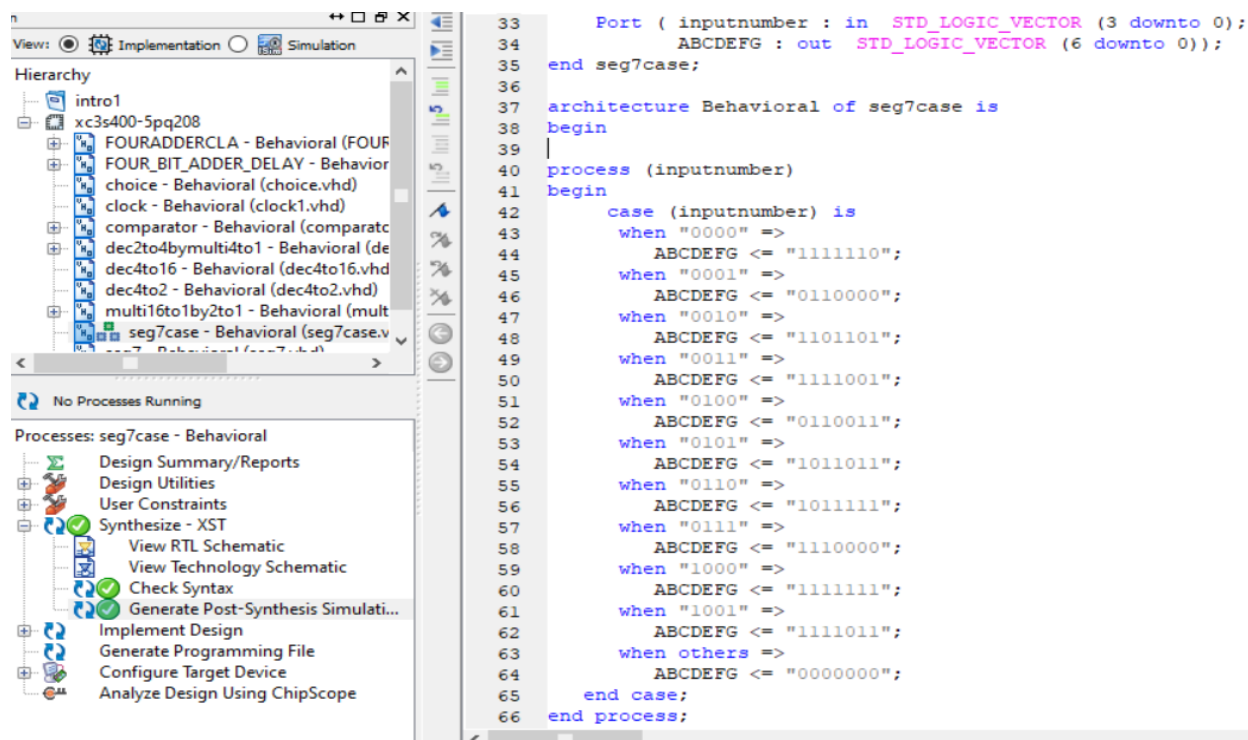
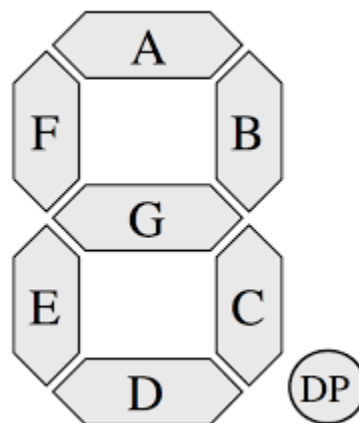
THE PRESENT IS THEIRS, THE FUTURE, FOR



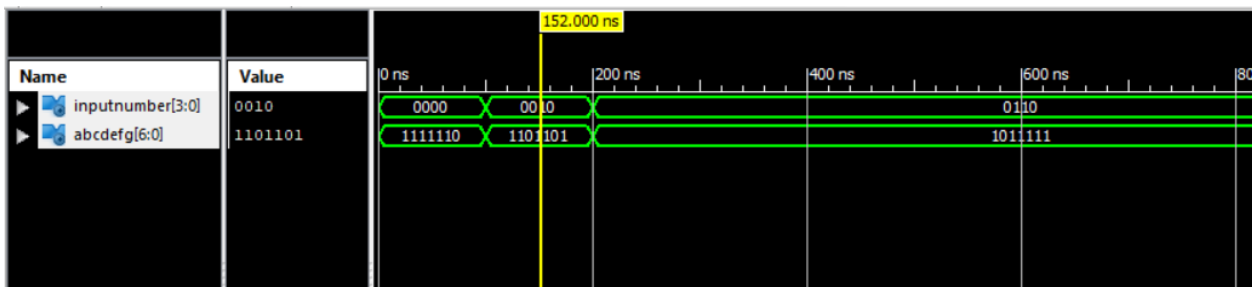
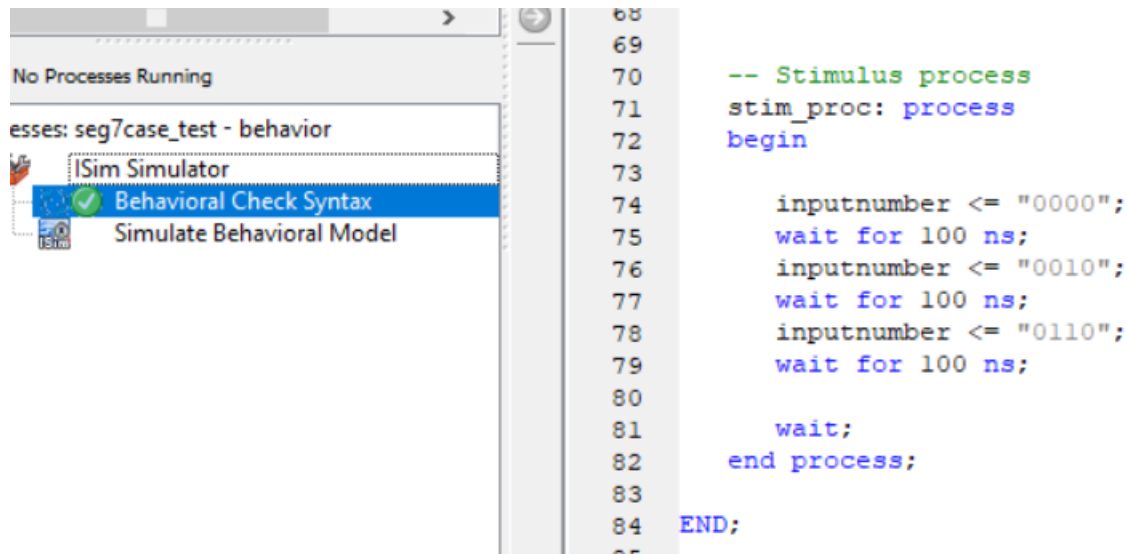
WHICH I REALLY WORK, IS MINE.

سون سگمنت با استفاده از دستورات شرطی:

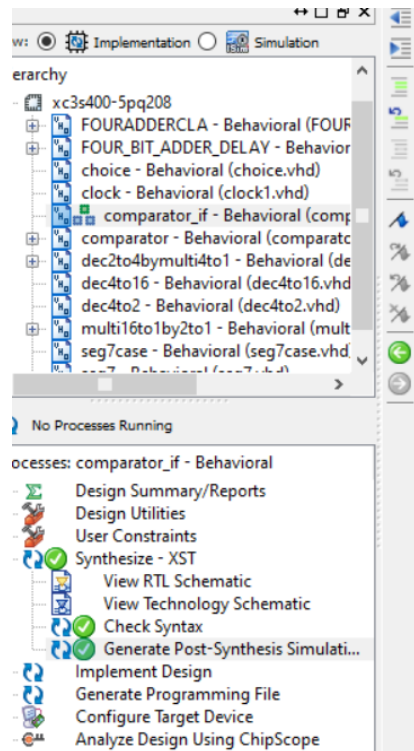
با استفاده از دستور CASE در داخل PROCESS میتوان طبق شکل زیر، LED ها را روشن کرد. (هر عددی در case باشد، چک میشود که آن خروجی ها روشن شوند)



```
33 Port ( inputnumber : in STD_LOGIC_VECTOR (3 downto 0);
34       ABCDEFG : out STD_LOGIC_VECTOR (6 downto 0));
35 end seg7case;
36
37 architecture Behavioral of seg7case is
38 begin
39 |
40 process (inputnumber)
41 begin
42     case (inputnumber) is
43     when "0000" =>
44         ABCDEFG <= "1111110";
45     when "0001" =>
46         ABCDEFG <= "0110000";
47     when "0010" =>
48         ABCDEFG <= "1101101";
49     when "0011" =>
50         ABCDEFG <= "1111001";
51     when "0100" =>
52         ABCDEFG <= "0110011";
53     when "0101" =>
54         ABCDEFG <= "1011011";
55     when "0110" =>
56         ABCDEFG <= "1011111";
57     when "0111" =>
58         ABCDEFG <= "1110000";
59     when "1000" =>
60         ABCDEFG <= "1111111";
61     when "1001" =>
62         ABCDEFG <= "1111011";
63     when others =>
64         ABCDEFG <= "0000000";
65     end case;
66 end process;
```



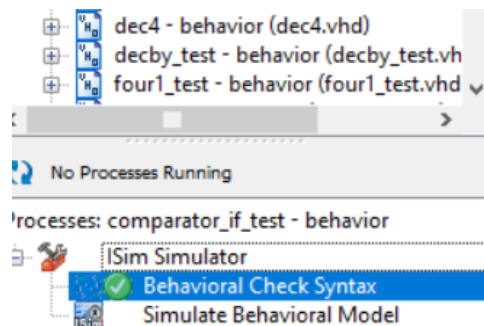
مقایسه دو عدد 8 بیتی: در این مقایسه از کتابخانه unsigned و تابع else if استفاده شده است و تابع مقایسه در درون خود کتابخانه موجود میباشد(البته میتوان از تفاضل دو عدد هم استفاده کرد ولی سخت و پیچیده تر میشود)



```

28 -- Uncomment the following library declaration if instantiating
29 -- any Xilinx primitives in this code.
30 --library UNISIM;
31 --use UNISIM.VComponents.all;
32
33 entity comparator_if is
34     Port ( inputnumber1 : in  STD_LOGIC_VECTOR (7 downto 0);
35           inputnumber2 : in  STD_LOGIC_VECTOR (7 downto 0);
36           EQUAL : out  STD_LOGIC;
37           LESS : out  STD_LOGIC;
38           GREATER : out  STD_LOGIC);
39 end comparator_if;
40
41 architecture Behavioral of comparator_if is
42
43 begin
44
45 process (inputnumber1,inputnumber2)
46 begin
47     if inputnumber1 > inputnumber2 then
48         GREATER <= '1';
49         LESS <= '0';
50         EQUAL <= '0';
51     elsif inputnumber1 < inputnumber2 then
52         GREATER <= '0';
53         LESS <= '1';
54         EQUAL <= '0';
55     else
56         GREATER <= '0';
57         LESS <= '0';
58         EQUAL <= '1';
59     end if;
60 end process;
61

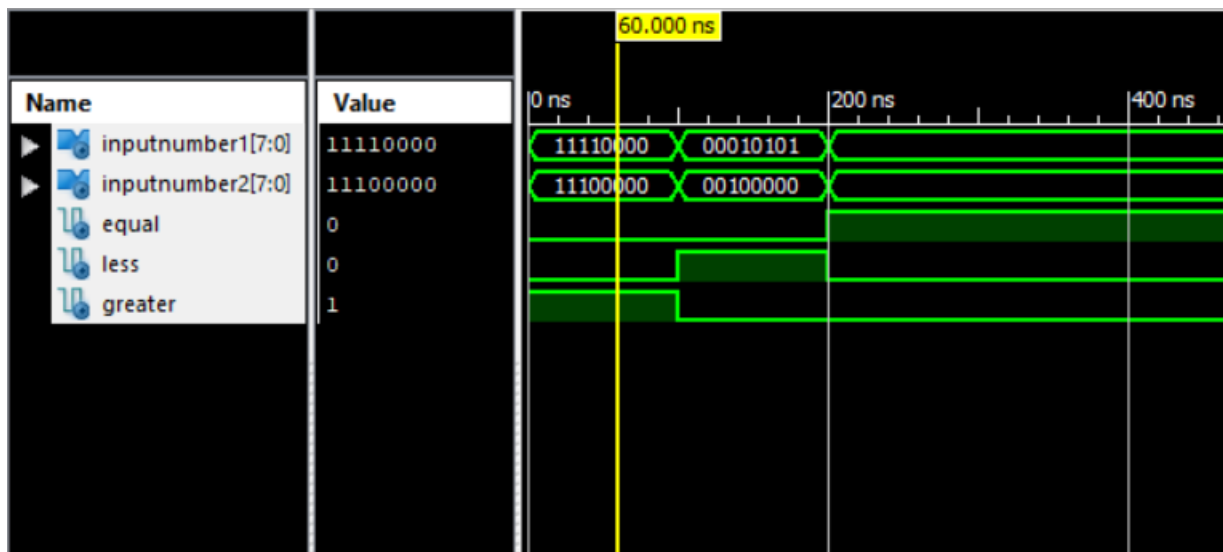
```



```

79
80 -- Stimulus process
81 stim_proc: process
82 begin
83
84     inputnumber1 <= "11110000";
85     inputnumber2 <= "11100000";
86     wait for 100 ns;
87     inputnumber1 <= "00010101";
88     inputnumber2 <= "00100000";
89     wait for 100 ns;
90     inputnumber1 <= "11110000";
91     inputnumber2 <= "11110000";
92     wait for 100 ns;
93
94
95     wait;
96 end process;
97
98 END;
99

```

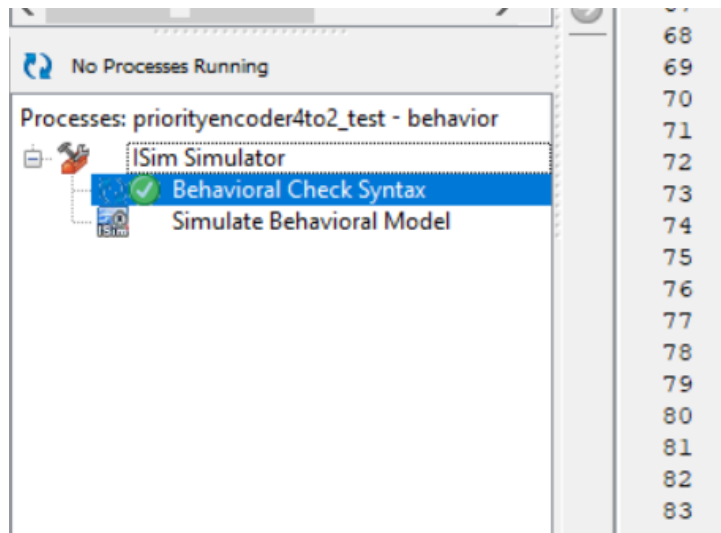


ساخت PRIORITY ENCODER : با استفاده از if else میتوان priority را تعیین کرد.

```

32
33 entity priorityencoder4to2 is
34     Port ( inputnumber : in  STD_LOGIC_VECTOR (3 downto 0);
35           outputnumber : out STD_LOGIC_VECTOR (1 downto 0));
36 end priorityencoder4to2;
37
38 architecture Behavioral of priorityencoder4to2 is
39
40 begin
41
42 process (inputnumber)
43 begin
44     if inputnumber(3)='1' then
45         outputnumber <= "11";
46     elsif inputnumber(2)='1' then
47         outputnumber <= "10";
48     elsif inputnumber(1)='1' then
49         outputnumber <= "01";
50     elsif inputnumber(0)='1' then
51         outputnumber <= "00";
52     end if;
53
54 end process;
55
56
57 end Behavioral;

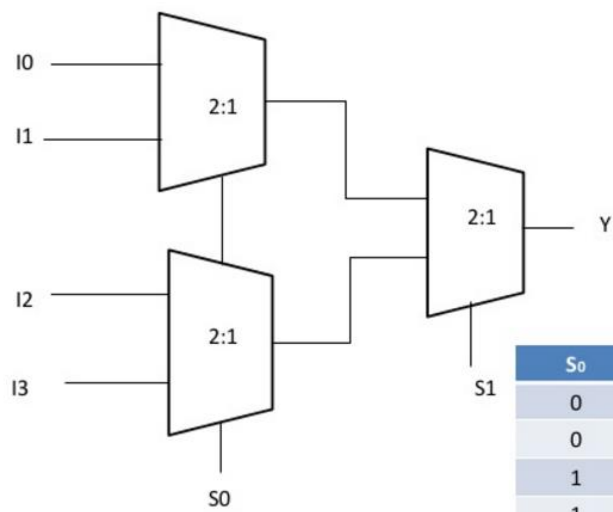
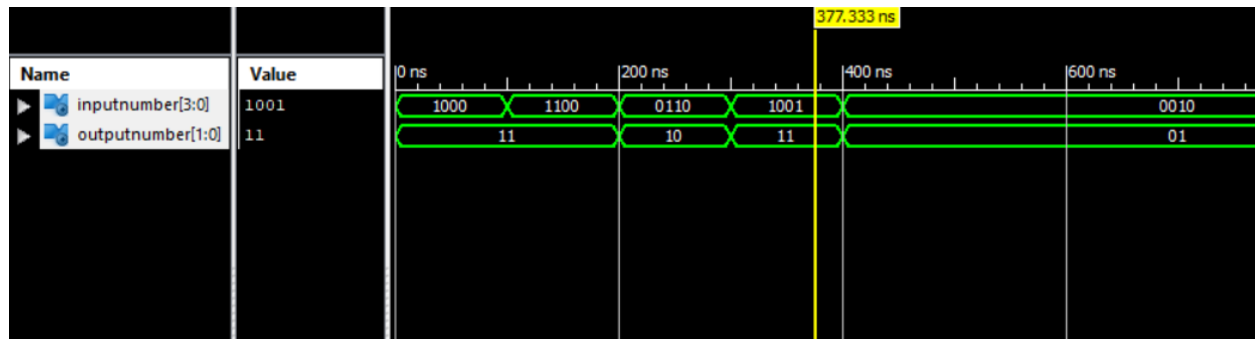
```



```

68
69  -- Stimulus process
70  stim_proc: process
71  begin
72
73      inputnumber <= "1000";
74      wait for 100 ns;
75      inputnumber <= "1100";
76      wait for 100 ns;
77      inputnumber <= "0110";
78      wait for 100 ns;
79      inputnumber <= "1001";
80      wait for 100 ns;
81      inputnumber <= "0010";
82      wait for 100 ns;
83

```



S <sub>0</sub>
0
0
1
1

ساخت مالتی 4 به 1 با 2 به 1:

choice - Behavioral (choice.vhd)  
clock - Behavioral (clock1.vhd)  
comparator\_if - Behavioral (comparat  
comparator - Behavioral (comparatc  
dec2to4bymulti4to1 - Behavioral (de  
dec4to16 - Behavioral (dec4to16.vhd)  
dec4to2 - Behavioral (dec4to2.vhd)  
multi16to1by2to1 - Behavioral (mult  
multi4to1by2to1byselectwhen - E  
priorityencoder4to2 - Behavioral (p  
seg7case - Behavioral (seg7case.vhd)  
seg7 - Behavioral (seg7.vhd)

No Processes Running

Processes: multi4to1by2to1byselectwhen - Behav

- Design Summary/Reports
- Design Utilities
- User Constraints
- Synthesize - XST
- View RTL Schematic
- View Technology Schematic
- Check Syntax
- Generate Post-Synthesis Simulati...
- Implement Design
- Generate Programming File
- Configure Target Device
- Analyze Design Using ChipScope

```

30 --use UNISIM.VCOMPONENTS.ALL;
31
32 entity multi4tolby2tolbyselectwhen is
33     Port ( inputnumber : in  STD_LOGIC_VECTOR (3 downto 0);
34           selector : in  STD_LOGIC_VECTOR (1 downto 0);
35           outputnumber : out STD_LOGIC);
36 end multi4tolby2tolbyselectwhen;
37
38 architecture Behavioral of multi4tolby2tolbyselectwhen is
39
40     signal w0,w1 : std_logic := '0';
41
42 begin
43
44     with selector(0) select
45         w0 <= inputnumber(0) when '0',
46             inputnumber(1) when '1',
47             inputnumber(1) when others;
48
49     with selector(0) select
50         w1 <= inputnumber(2) when '0',
51             inputnumber(3) when '1',
52             inputnumber(3) when others;
53
54     with selector(1) select
55         outputnumber <= w0 when '0',
56             w1 when '1',
57             w1 when others;
58
59

```

multi\_test - Behavioral (multi\_test.vhd)  
priorityencoder4to2\_test - behavior (

No Processes Running

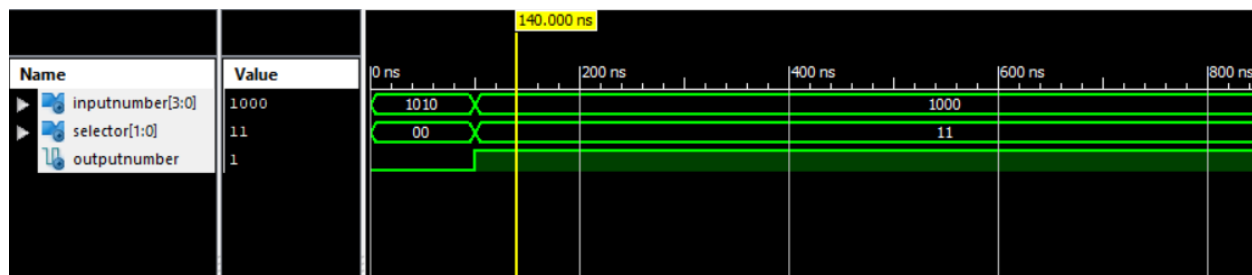
Processes: multi4to1by2to1byselectwhen\_test - be

- ISim Simulator
- Behavioral Check Syntax
- Simulate Behavioral Model

```

69
70
71
72
73 -- Stimulus process
74 stim_proc: process
75 begin
76
77     inputnumber <="1010";
78     selector <="00";
79     wait for 100 ns;
80     inputnumber <="1000";
81     selector <="11";
82     wait for 100 ns;
83
84     wait;
85 end process;
86
87 END;
88

```





## امتیازی:

درباره for loop: برای اجرای حلقه ای به تعداد دفعات مشخص است مثلاً

میگوییم:

```
For I in 0 to 3 loop
```

```
statement ;
```

```
End loop;
```

به این معناست که مقدار i از 0 تا 3 خود به خود زیاد میشود و عملاً این حلقه 4 بار تکرار میشود. میتوان از همین i به عنوان اندیس نیز استفاده کرد و حتی در دستورات شرطی if هم کاربرد دارد.

عبارت for generate: یک سری دستور هستند که برای تکرار هستند و بصورت موازی همه دستوراتش اجرا میشود

برای if generate هم یک سری دستور شرطی بصورت موازی اجرا میشوند.

تفاوت اینها با قبلی در این است که قبلی ها در process باید باشند ولی اینها نیازی به نوشته شدن در process ندارند. و همچنین دستورات قبلی ها بصورت سری اجرا میشوند ولی اینها بصورت موازی اجرا میشوند.