

نویسنده:

علیرضا طباطبائی (9723052)

آزمایشگاه مدار منطقی



Amirkabir University of Technology
(Tehran Polytechnic)

استاد آزمایشگاه:

دکتر علیرضا

ظاهری نوید



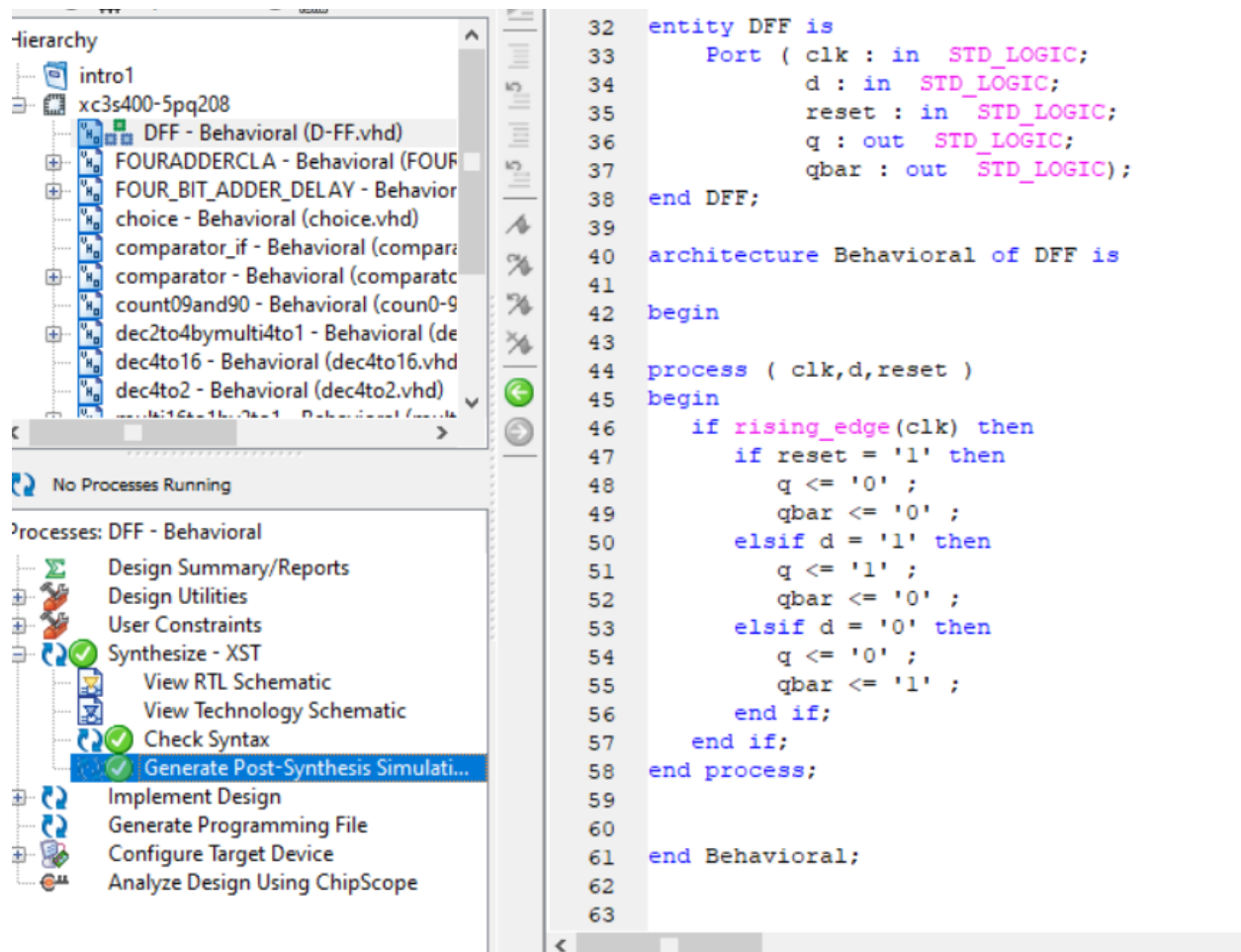
THE PRESENT IS THEIRS, THE FUTURE, FOR



WHICH I REALLY WORK, IS MINE.

فلیپ فلاپ D :

ورودی را به خروجی انتقال میدهد و در Q قرار میدهد. $\text{not } q$ نیز Qbar است.



The screenshot displays the Xilinx ISE environment. On the left, the 'Hierarchy' panel shows a project named 'xc3s400-5pq208' with a component 'DFF - Behavioral (D-FF.vhd)' selected. Below it, the 'Processes' panel lists various design steps, with 'Generate Post-Synthesis Simulation' highlighted. On the right, the VHDL code for the 'DFF' entity is shown, defining its ports and behavioral logic.

```
32 entity DFF is
33     Port ( clk : in  STD_LOGIC;
34           d : in  STD_LOGIC;
35           reset : in  STD_LOGIC;
36           q : out  STD_LOGIC;
37           qbar : out  STD_LOGIC);
38 end DFF;
39
40 architecture Behavioral of DFF is
41
42 begin
43
44 process ( clk,d,reset )
45 begin
46     if rising_edge(clk) then
47         if reset = '1' then
48             q <= '0' ;
49             qbar <= '0' ;
50         elsif d = '1' then
51             q <= '1' ;
52             qbar <= '0' ;
53         elsif d = '0' then
54             q <= '0' ;
55             qbar <= '1' ;
56         end if;
57     end if;
58 end process;
59
60
61 end Behavioral;
62
63
```

No Processes Running

Processes: DFF_test - behavior

ISim Simulator

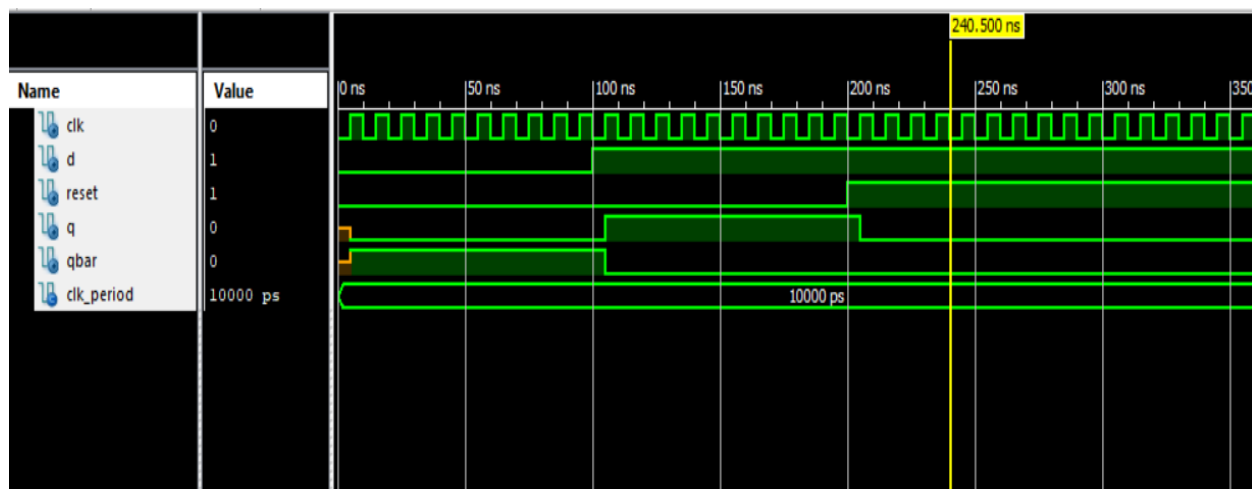
Behavioral Check Syntax

Simulate Behavioral Model

```

85
86 -- Stimulus process
87 stim_proc: process
88 begin
89     d <= '0';
90     wait for 100 ns;
91     d <= '1';
92     wait for 100 ns;
93     reset <= '1';
94     wait for 100 ns;
95
96     wait for clk_period*10;
97
98     wait;
99 end process;
100
101 END;

```

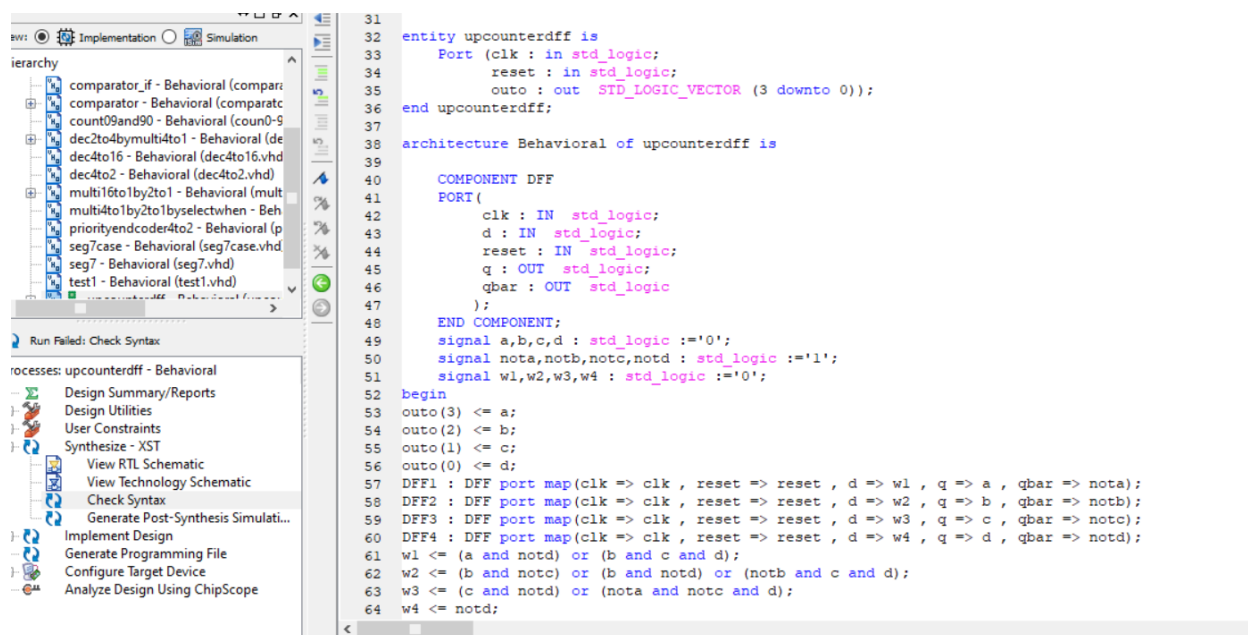


سوال 2:

استاد من کد زیر رو برای این سوال زدم ولی اصلا نمیدونم چرا ارور

میده!!! با کامپوننت و حتی if هم زدم ولی بازم ارور داد...

چند ساعته دارم روش فکر میکنم ولی هیچ تغییری در ارور ایجاد نشد!



```
31
32 entity upcounterdff is
33     Port (clk : in std_logic;
34           reset : in std_logic;
35           outo : out STD_LOGIC_VECTOR (3 downto 0));
36 end upcounterdff;
37
38 architecture Behavioral of upcounterdff is
39
40     COMPONENT DFF
41     PORT (
42         clk : IN std_logic;
43         d : IN std_logic;
44         reset : IN std_logic;
45         q : OUT std_logic;
46         qbar : OUT std_logic
47     );
48 END COMPONENT;
49 signal a,b,c,d : std_logic := '0';
50 signal nota,notb,notc,notd : std_logic := '1';
51 signal w1,w2,w3,w4 : std_logic := '0';
52 begin
53     outo(3) <= a;
54     outo(2) <= b;
55     outo(1) <= c;
56     outo(0) <= d;
57     DFF1 : DFF port map(clk => clk , reset => reset , d => w1 , q => a , qbar => nota);
58     DFF2 : DFF port map(clk => clk , reset => reset , d => w2 , q => b , qbar => notb);
59     DFF3 : DFF port map(clk => clk , reset => reset , d => w3 , q => c , qbar => notc);
60     DFF4 : DFF port map(clk => clk , reset => reset , d => w4 , q => d , qbar => notd);
61     w1 <= (a and notd) or (b and c and d);
62     w2 <= (b and notc) or (b and notd) or (notb and c and d);
63     w3 <= (c and notd) or (nota and notc and d);
64     w4 <= notd;
```