آنوشا شریعتی 9923041

تکلیف 7 کد نویسی در سطح رفتاری و ساختاری

الف)

ورودی های مدار:

1. ورودی Punlock که نقیض متغیر t_i است یعنی وقتی 0 است t_i یک میشود و وقتی 1 است t_i 0 است. از مقدار t_i برای ریست کردن فلیپ فلاپ T استفاده میشود به این صورت که برای کار کردن باید Punlock همیشه 1 باشد و برای ریست شدن باید به آن مقدار 0 بدهیم.
2. ورودی sel که در صورت 0 بودن به این معناست که کلید باز است و ورودی sipo با مقاومتی به زمین وصل میشود. در صورت 1 شدن به این معناست که کلید وصل شده و مقدار vcc به ورودی sipo وصل میشود.
3. ورودی clk که کلاک کانتر24 بیتی را تامین میکند.
4. ورودی preset که نقیض متغیر res است یعنی وقتی 0 است res 1 میشود و وقتی 1 است res 0 است. با 1 شدن متغیر res کانتر 2و16و24 بیتی، فلیپ فلاپ D و T و sipoها ریست میشوند پس مقدار preset برای کار کردن مدار باید همیشه1 باشد و وقتی میخواهیم مدار ریست شود 0 بدهیم.
5. ورودی s_i ورودی sipo دوم را مشخص میکند.
6. ورودی data_clk به پایه کلاک sipo دوم داده میشود.

خروجی مدار:

1. متغیر y خروجی مدار را نشان میدهد که برابر با مقدار k یعنی خروجی فلیپ فلاپ Tاست.

نحوه کارکرد مدار:

با آمدن لبه بالارونده کلاک کانتر 24 بیتی شروع به کار میکند. وقتی کلید زده میشود ورودی ساپیو به وی سی سی وصل شده و در هر کلاک مقدار 1 را شیفت میدهد. قبل از این که 8 کلاک طی شود ورودی فلیپ فلاپ دی صفر است و کانتر 2 فعال نمیشود. بعد از کلاک هشتم کانتر 2 فعال میشود. در این حین ورودی های مختلف مالتی پلکسر عبور داده میشوند. خروج وای در صورتی یک میشود که فرکانس دیتا کلاک با یکی از فرکانس های ورودی مالتی پلکسر برابر شود. که باعث میشود حاصل کامپرتور یک شود.

ب) کد نوشته شده در سطح رفتاری:

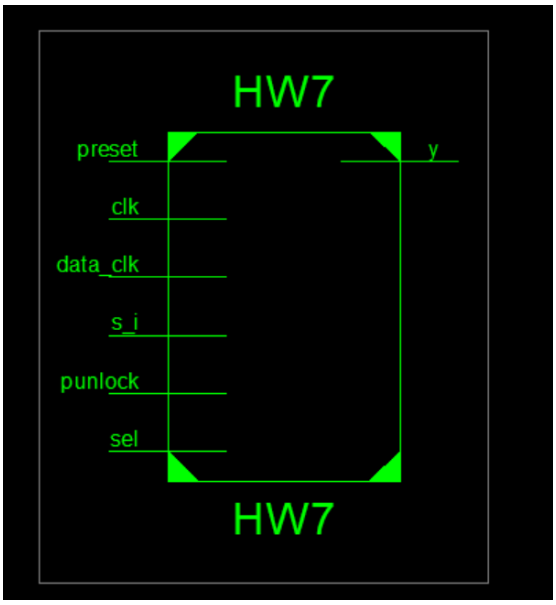
- از خط 13 تا 36 سیگنال های مورد نیاز در طول برنامه را تعریف کردیم.

```
1
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  use IEEE.STD_LOGIC_unsigned.ALL;
5
6  entity HW7 is
7      Port ( preset , clk , data_clk , s_i , punlock , sel : in  STD_LOGIC;
8             y : out  STD_LOGIC);
9  end HW7;
10
11 architecture Behavioral of HW7 is
12 --sipo
13 signal res : STD_LOGIC;                                  --not preset
14 signal t_i : STD_LOGIC;                                  --not punlock
15 signal q : STD_LOGIC_VECTOR(23 downto 0);                --output of cnt24
16 signal sipo1 : STD_LOGIC_VECTOR(7 downto 0);             --output of sipo1
17 --dff
18 signal idff : STD_LOGIC;                                 --input of dff
19 signal dff : STD_LOGIC;                                  --Output of dff
20 --cnt2
21 signal clk2 : STD_LOGIC;                                 --clk for cnt2
22 signal x : STD_LOGIC_VECTOR(1 downto 0);                 --output of cnt2
23 --mux1,2
24 signal mux1 : STD_LOGIC;                                 --output of mux1
25 signal mux2 : STD_LOGIC;                                 --output of mux2
26 signal k : STD_LOGIC;                                    --selector of mux2
27
28 --cnt3
29 signal cnt3 : STD_LOGIC_VECTOR(15 downto 0);             --output of cnt16
30 signal res2 : STD_LOGIC;                                 --reset of cnt16
31 --sipo2
32 signal sipo2 : STD_LOGIC_VECTOR(15 downto 0);            --output of sipo2
33 --comp
34 signal comp : STD_LOGIC;                                 --output of comparetor
35 --tff
36 signal res3 : STD_LOGIC;                                 --reset of fft
37
38 begin
39
```

- از خط 40 تا آخر برنامه را به صورت ماژولار برای هر قطعه نوشتیم.

```
40 t_i <= not(punlock);
41 res <= not(preset);
42
43 --cnt24
44 process(clk,res)
45    begin
46    if(res = '1')then
47       q <= (others => '0');
48    elsif(clk' event and clk = '1') then
49       q <= q + 1;
50    end if;
51 end process;
52
53 --sipo1
54 process(q(15),res)
55    begin
56    if(res='1') then
57       sipo1 <= (others => '0');
58    elsif(q(15)' event and q(15)= '1')then
59       sipo1 <= sel & sipo1(7 downto 1);
60    end if;
61 end process;
62
63 idff <= sipo1(0) and sipo1(1) and sipo1(2) and sipo1(3) and sipo1(4) and sipo1(5) and sipo1(6) and sipo1(7);
64
65 --dff
66 process(q(15),res)
67   begin
68   if(res='1') then
69       dff <= '0';
70   elsif(rising_edge(q(15))) then
71       dff <= idff;
72   end if;
73 end process;
74
75 clk2 <= (idff and not(dff));
```

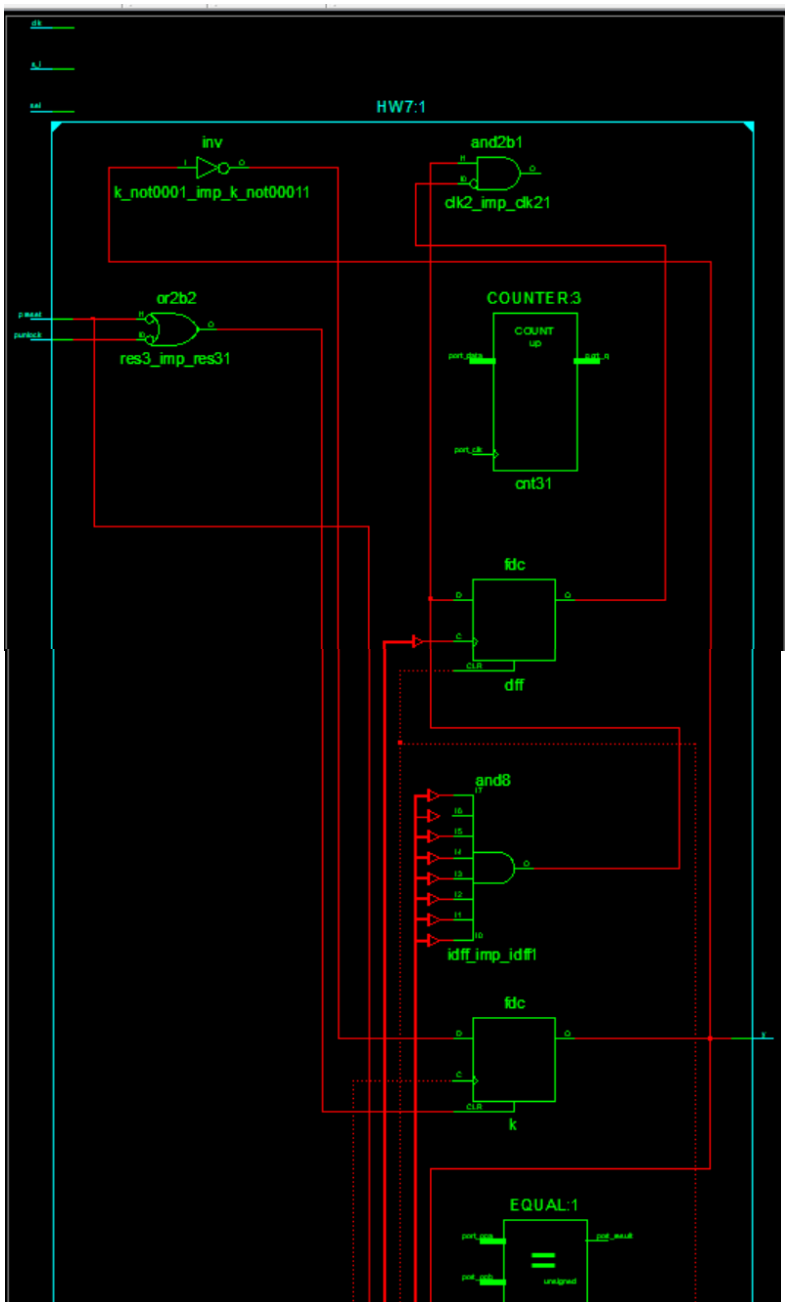
```
77  --cnt2
78  process(clk2,res)
79    begin
80    if(res = '1')then
81        x <=(others => '0');
82    elsif(clk2' event and clk2 = '1') then
83        x <= x+1;
84    end if;
85  end process;
86
87  --mux1
88  mux1 <= q(20) when x = "00" else
89          q(15) when x="01" else
90          q(10) when x="10" else
91           q(5) when x="11";
92
93  --mux2
94  mux2 <= mux1 when k='0' else
95           '0' when k='1';
96
97  res2 <= res or comp;
99  --cnt3
100 process(mux2, res2)
101   begin
102   if(res2 = '1')then
103       cnt3<= (others=> '0');
104   elsif(mux2' event and mux2='1')then
105       cnt3 <= cnt3 + 1;
106   end if;
107 end process;
108
109 --sipo2
110 process(data_clk,res)
111 begin
112   if(res='1')then
113       sipo2 <= (others => '0');
114   elsif(data_clk'event and data_clk='1')then
115       sipo2 <= s_i & sipo2 (15 downto 1);
116   end if;
117 end process;
118
119 -- Comparator
120 comp<='1' when ( sipo2=cnt3 ) else '0';
121
122 res3 <= (t_i or res);
119 -- Comparator
120 comp<='1' when ( sipo2=cnt3 ) else '0';
121
122 res3 <= (t_i or res);
123
124 --TFF
125 process(comp,res3)
126   begin
127   if(res3 = '1')then
128       k <= '0';
129   elsif(comp' event and comp='1') then
130       k <= not(k);
131   end if;
132 end process;
133
134 y <= k;
135
136 end Behavioral;
```

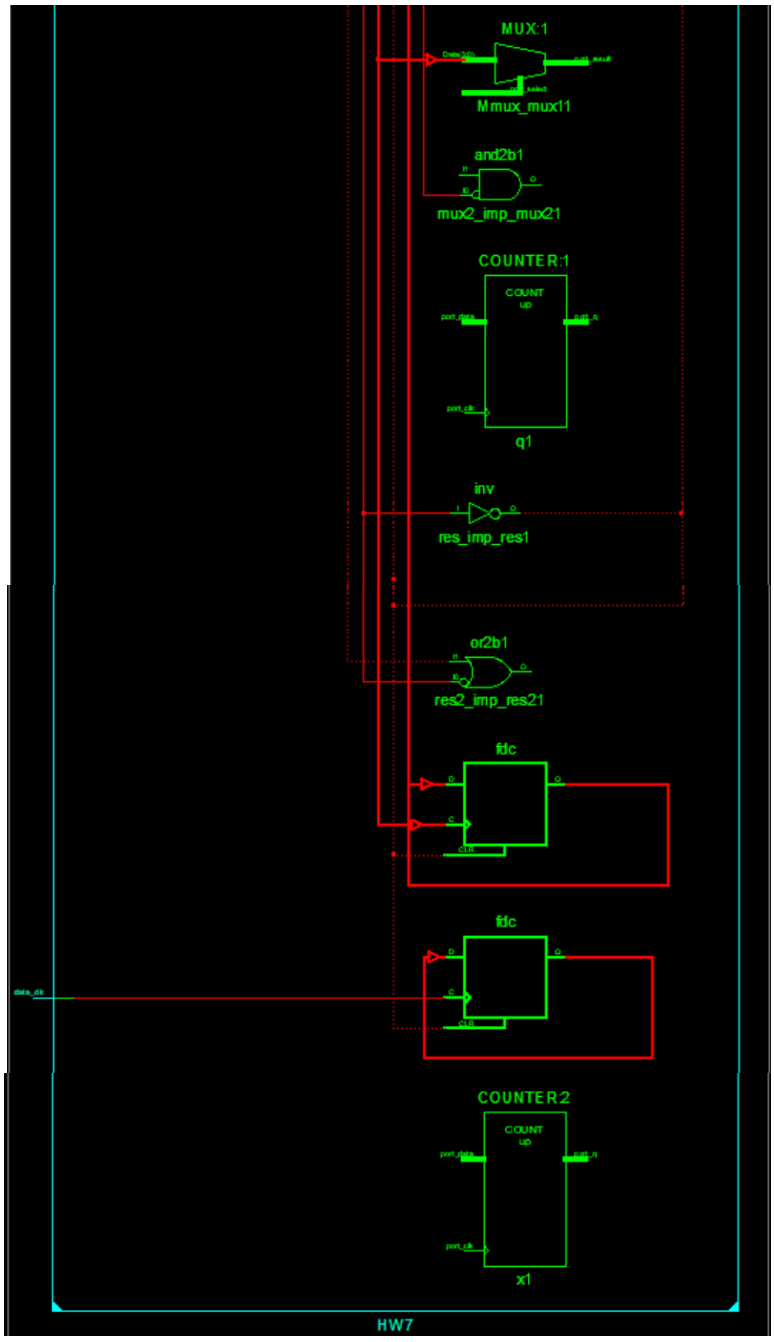


RTL شماتیک مدار طراحی شده:

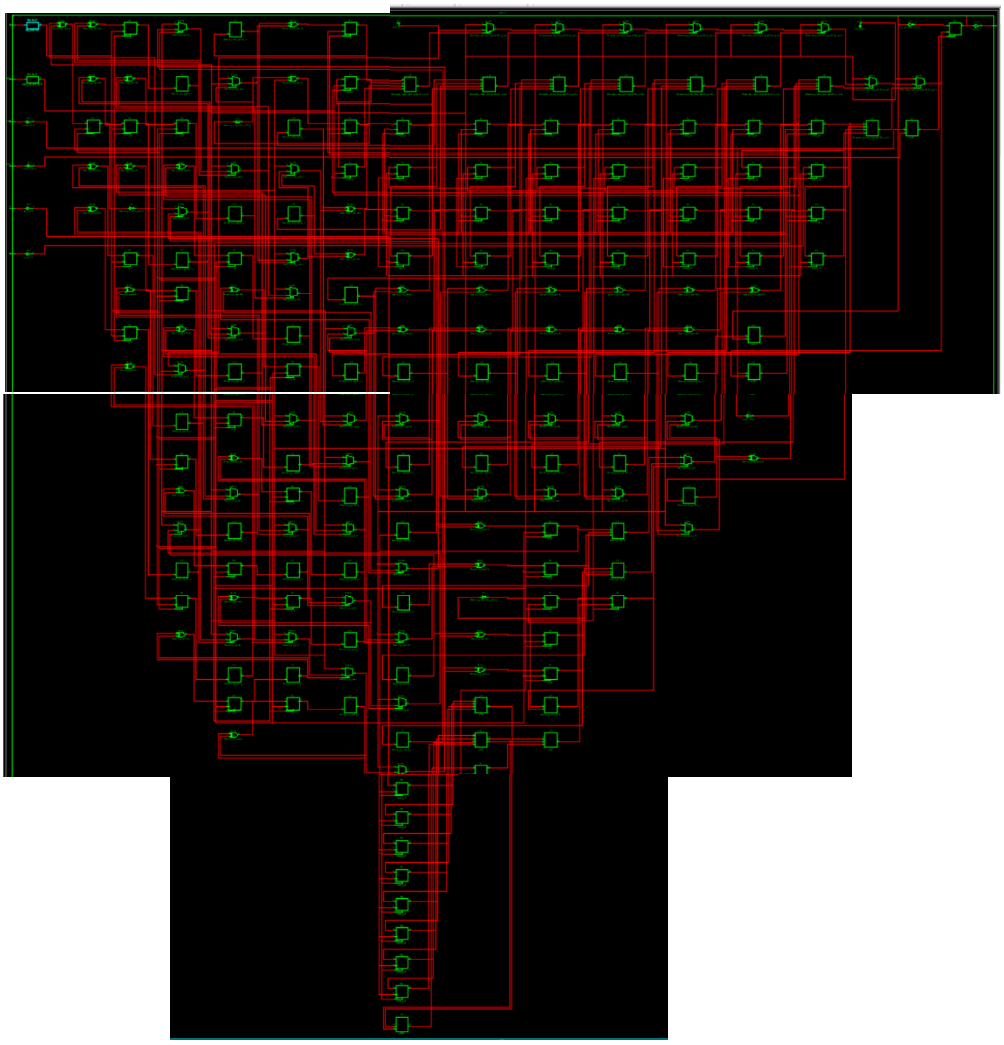
همینطور که مشاهده میشود مدار دارای 6 ورودی و یک خروجی که در قسمتاول گزارش ذکر شده است میباشد.

اگر روی این ماژول کلیک کنیم مدار با جزئیات به صورت زیر نمایش داده میشود.




MUX:1
Mmux_mux11
and2b1
mux2_imp_mux21
COUNTER:1
COUNT
up
q1
inv
res_imp_res1
or2b1
res2_imp_res21
fdc
fdc
COUNTER2
COUNT
up
x1
HW7

TECHNOLOGY MAP:



د) شبیه سازی در Isim:

ورودی رو به رو را به مدار میدهیم تا خروجی را برای ما شبیه سازی کند.

```
62    stim_proc: process
63    begin
64       punlock<='1';
65       sel<='1';
66       preset<='1';
67       clk<='0';
68       s_i<='1';
69       data_clk<='0';
70       wait for 100 ns;
71       punlock<='1';
72       sel<='1';
73       preset<='1';
74       clk<='1';
75       s_i<='0';
76       data_clk<='0';
77       wait for 100 ns;
78       punlock<='1';
79       sel<='0';
80       preset<='1';
81       clk<='0';
82       s_i<='1';
83       data_clk<='1';
84       wait for 100 ns;
```