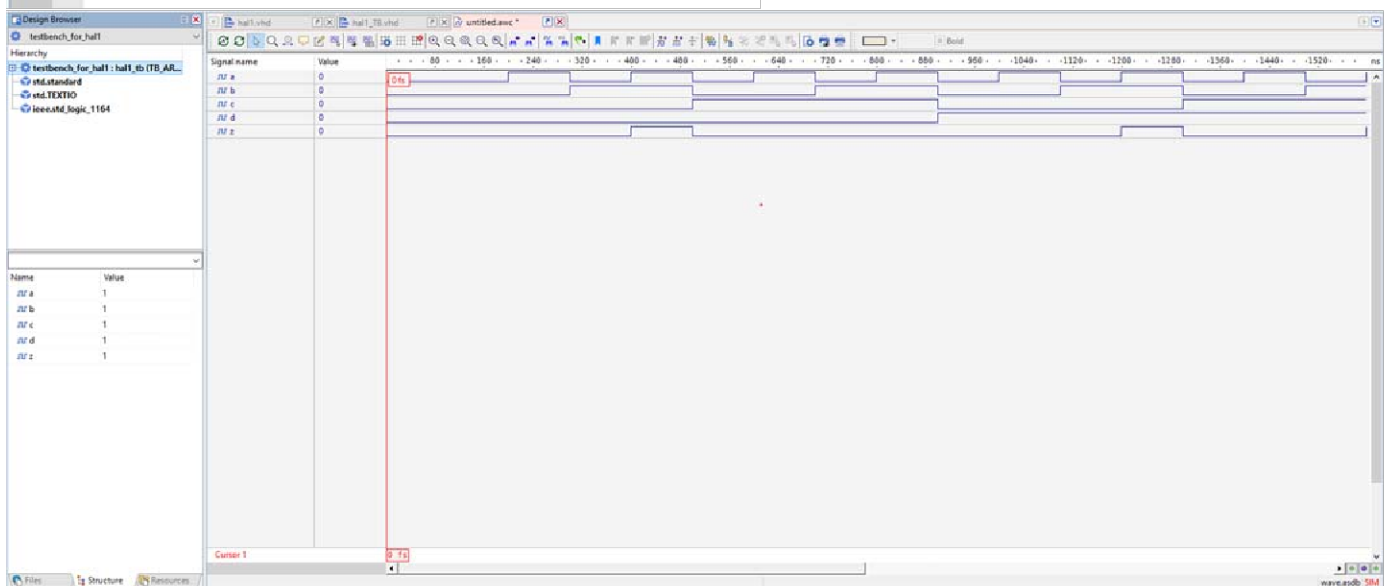


```

1
2
3 -- Title      : hall
4 -- Design     : test1
5 -- Author     : Mohammadreza Hendiani
6 -- Company    : MH.inc
7
8
9
10
11 library IEEE;
12 use IEEE.STD_LOGIC_1164.ALL;
13
14 entity hall is
15     Port ( a : in  STD_LOGIC;
16           b : in  STD_LOGIC;
17           c : in  STD_LOGIC;
18           d : in  STD_LOGIC;
19           z : out STD_LOGIC);
20 end hall;
21
22
23 architecture hall of hall is
24
25     signal t0 :STD_LOGIC;
26     signal t1 :STD_LOGIC;
27     signal t2 :STD_LOGIC;
28     signal t3 :STD_LOGIC;
29     signal t4 :STD_LOGIC;
30
31 begin
32
33     t0 <= b      or  c;
34     t1 <= not(c) or  d;
35     t2 <= not(d) or  b;
36     t3 <= t0     and t1;
37     t4 <= t2     and t3;
38     z  <= t4     and a;
39
40 end hall;
41

```

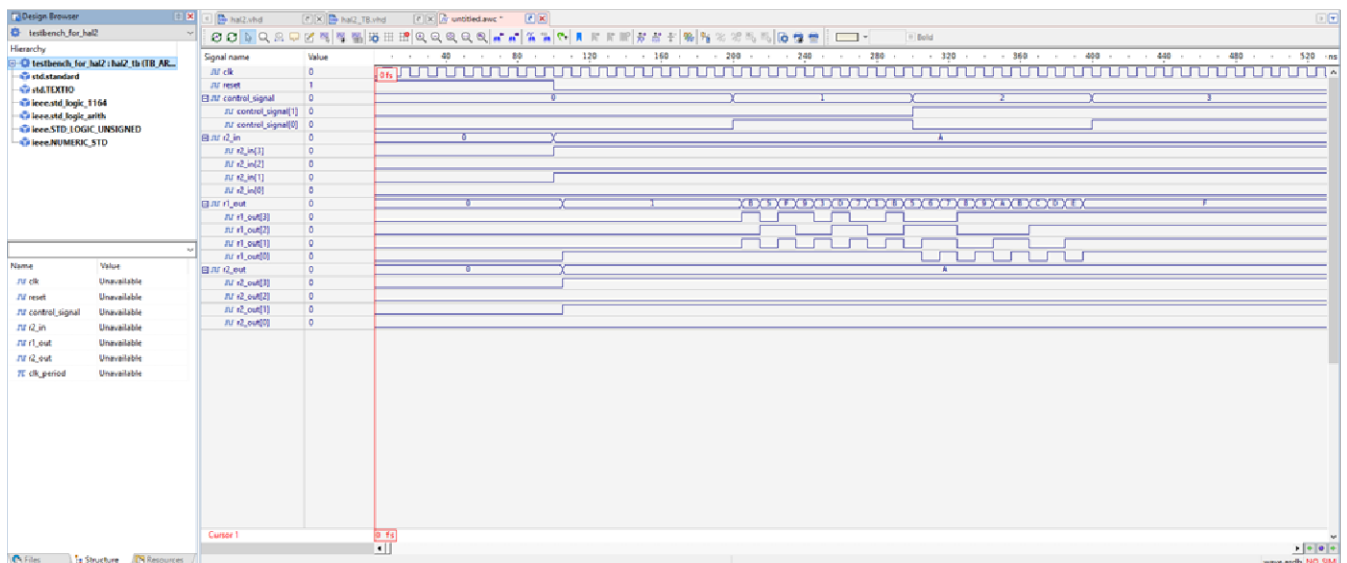


ورودی ها با همون اسمی ک توی صورت سوال هست A,B,C,D و خروجی زد. توی خط های 25 تا 29 یه تعداد سیگنال تعریف کردیم . تی صفر تا تی چهار برای اتصالات خروجی گیت هایی هست استفاده میکنیم اصل کار در خط 33 تا 38 انجام شده است به این صورت که مثلا تو خط 33 ما اومدیم اولین اور از بالا شروع کردیم بی رو با سی اور کردیم حالا فعلا خروجی رو تی صفر اسمش رو گذاشتیم یعنی توی سیگنال تی صفر ریختیم نات سی رو با دی اور کردیم بعنوان تی یک. نات دی رو با بی اور کردیم بعنوان تی دو الان سه تا اور که داشتیم تموم شد.

```

1
2
3
4 -- Title      : hal2
5 -- Design     : test2
6 -- Author      : Mohammadreza Hendifani
7 -- Company     : MH.inc
8
9
10 library IEEE;
11 use IEEE.STD_LOGIC_1164.ALL;
12 use IEEE.STD_LOGIC_UNSIGNED.ALL;
13 use IEEE.NUMERIC_STD.ALL;
14
15 entity hal2 is
16     Port ( clk : in  STD_LOGIC;
17           reset : in  STD_LOGIC;
18           control_signal : in  STD_LOGIC_VECTOR(1 downto 0);
19           r2_in : in  STD_LOGIC_VECTOR (3 downto 0);
20           r1_out : out STD_LOGIC_VECTOR (3 downto 0);
21           r2_out : out STD_LOGIC_VECTOR (3 downto 0));
22 end hal2;
23
24 architecture hal2 of hal2 is
25
26     signal r1 : STD_LOGIC_VECTOR(3 downto 0);
27     signal r2 : STD_LOGIC_VECTOR(3 downto 0);
28
29     signal r1_next : STD_LOGIC_VECTOR(3 downto 0);
30     signal add_out : STD_LOGIC_VECTOR(3 downto 0);
31     signal inc_out : STD_LOGIC_VECTOR(3 downto 0);
32
33 begin
34
35     add_out <= r1 + r2;
36     inc_out <= r1 + "0001";
37
38     r1_next <= "0001" when control_signal = "00" else
39     add_out when control_signal = "01" else
40     inc_out when control_signal = "10" else
41     r1;
42
43     process(clk,reset)
44     begin
45         if(reset = '1')then
46             r1 <= "0000";
47             r2 <= "0000";
48         elsif(rising_edge(clk))then
49             r1 <= r1_next;
50             r2 <= r2_in;
51         end if;
52     end process;
53
54     r1_out <= r1;
55     r2_out <= r2;
56
57 end hal2;
58

```

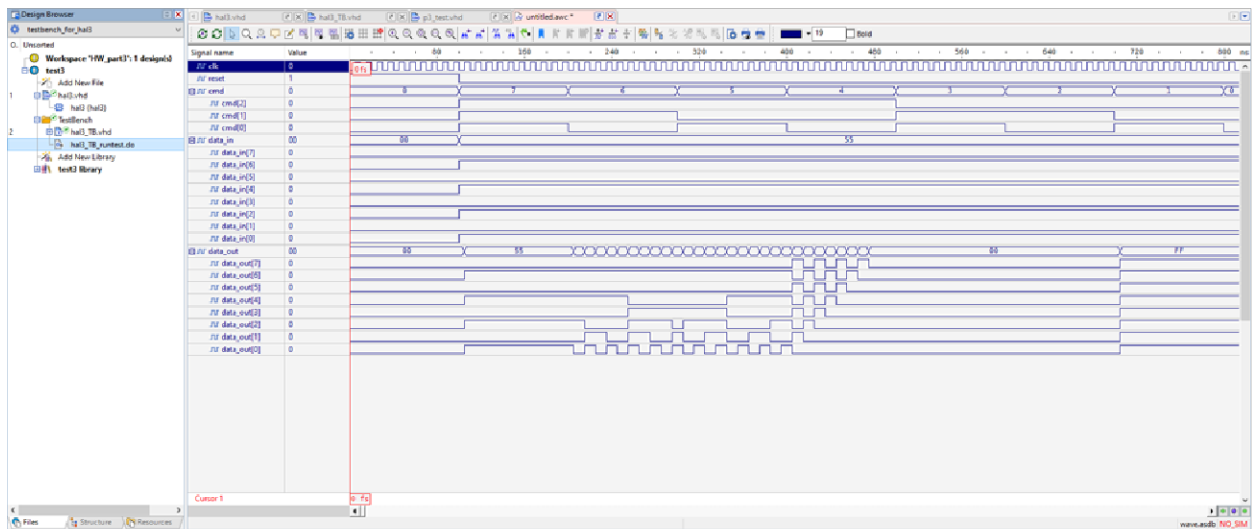


چهار تا ورودی داریم که در صورت سوال اومده دوتا خروجی در نظر گرفتیم در خط 26 و 27 که عملاً برامون حکم رجیستر رو دارن در خط 31 یه سری سیگنال در نظر گرفتیم

```

1  |-----|
2  |
3  |-- Title       : hal3
4  |-- Design      : test3
5  |-- Author       : Mohammadreza Hendiani
6  |-- Company      : MH.inc
7  |-----|
8  |
9  |library IEEE;
10 |use IEEE.STD_LOGIC_1164.ALL;
11 |use ieee.std_logic_unsigned.all;
12 |use ieee.numeric_std.all;
13 |
14 |
15 |entity hal3 is
16 |    Port ( clk : in  STD_LOGIC;
17 |          reset : in  STD_LOGIC;
18 |          cmd : in  STD_LOGIC_VECTOR (2 downto 0);
19 |          data_in : in  STD_LOGIC_VECTOR (7 downto 0);
20 |          data_out : out STD_LOGIC_VECTOR (7 downto 0));
21 |end hal3;
22 |
23 |architecture hal3 of hal3 is
24 |
25 |    signal data : STD_LOGIC_VECTOR(7 downto 0);
26 |
27 |begin
28 |
29 |    process(clk,reset)
30 |    begin
31 |        if(reset = '1')then
32 |            data <= "00000000";
33 |        elsif(rising_edge(clk))then
34 |            if(cmd = "000")then
35 |                data <= data;
36 |            elsif(cmd = "001")then
37 |                data <= "11111111";
38 |            elsif(cmd = "010")then
39 |                data <= "00000000";
40 |            elsif(cmd = "011")then
41 |                data <= ('0'&data(7 downto 1));
42 |            elsif(cmd = "100")then
43 |                data <= data(6 downto 0)&'0';
44 |            elsif(cmd = "101")then
45 |                data <= data + 1;
46 |            elsif(cmd = "110")then
47 |                data <= data - 1;
48 |            else
49 |                data <= data_in;
50 |            end if;
51 |        end if;
52 |    end process;
53 |    data_out <= data;
54 |
55 |end hal3;
56 |

```



یه سیگنال دیتا تعریف کردیم 8 بیت حکم رجیستر رو داره حالا خط 29 پراسس رو کلاک و ریست اکر ریست یک بود مقدار دیتا رو صفر میکنیم . در غیر این صورت اگر لبه ی بالا رونده کلاک بیاد حالا با توجه به سی ان بی تصمیم میگیریم ک چه کاری میخایم انجام بدیم ریست مون رو ها سنگ شد اگر سی ان صفر باشه هیچ اتفاقی نمی افتد هلد انجام میشه پس دیتا خودش اگر یک باشع دیتا همه بیت هاش یک