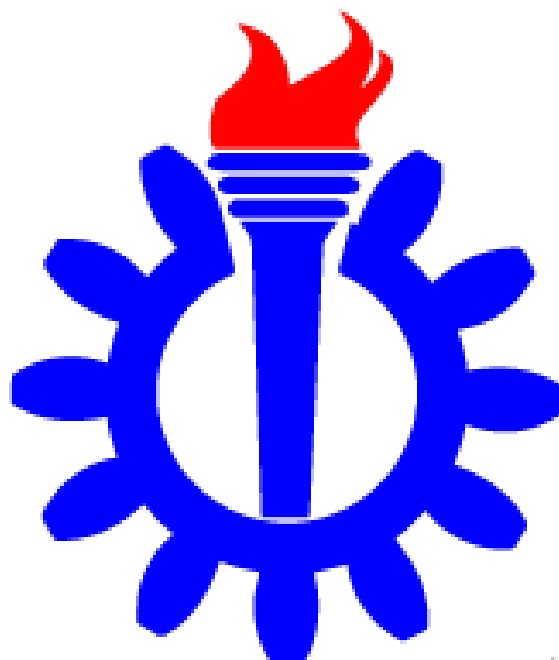


بسم الله الرحمن الرحيم



دانشگاه علم و صنعت ایران

گزارش کار دوم

استاد: دکتر معبتر

مصدر عرفان زارچ زردینر

مصدر رضا صاحب زاده

امیر حسین ملکوتی

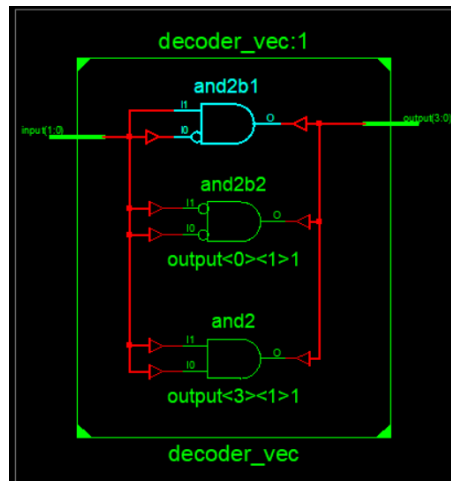
## هدف آزمایش:

ساخت دیکودر ۴ به ۱ و مالتی پلکسر ۴ به ۱ و ۱ به ۴ هست.

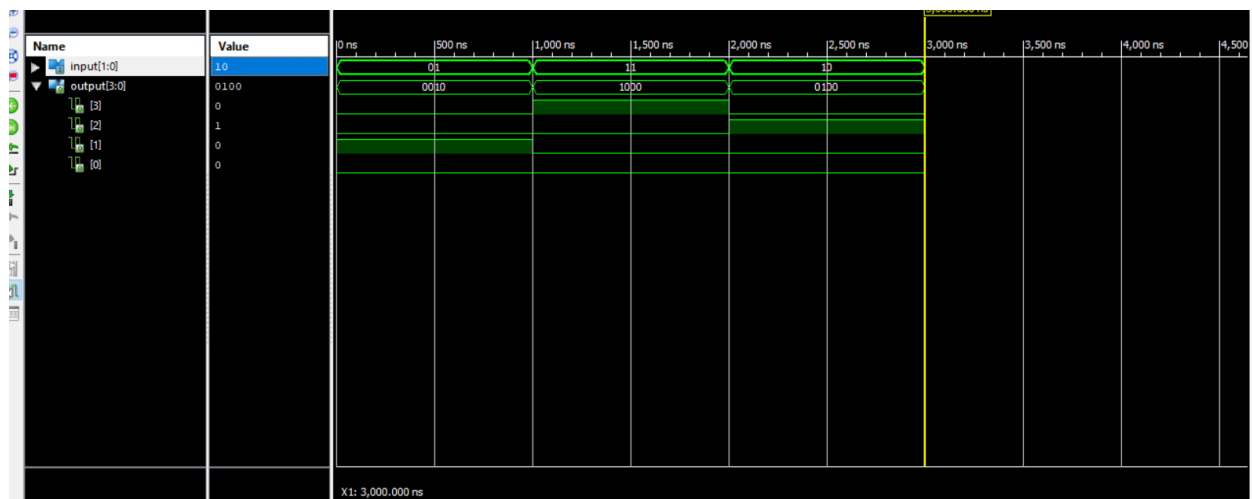
## روش کار:

ابتدا با استفاده از دستور **when**؛ بر حسب ورودی داده شده خروجی رو مقدار دهی می کنیم.

سیگنال و گیت ساخته شده آن را که پردازش شده در زیر می بینید:



شکل دیکودر ۴ به ۱



سیگنال دیکودر ۴ به ۲

```

Layout  Help
[Icons]
library ieee;
use ieee.std_logic_1164.all;
entity decoder_vec is
    port(
        input: in std_logic_vector(1 downto 0);
        output: out std_logic_vector(3 downto 0)
    );
end entity decoder_vec;

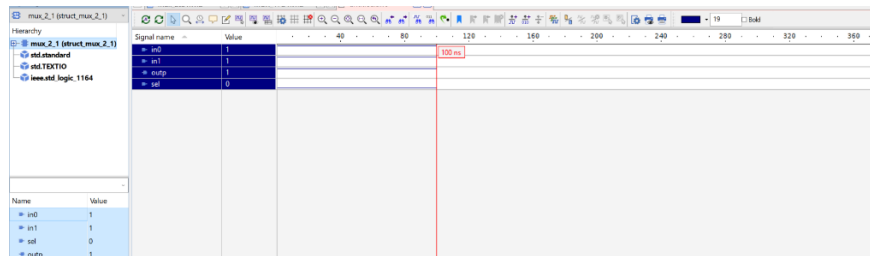
architecture behaviorial of decoder_vec is
begin
    output(0) <= '1' when input = "00" else '0';
    output(1) <= '1' when input = "01" else '0';
    output(2) <= '1' when input = "10" else '0';
    output(3) <= '1' when input = "11" else '0';

end behaviorial;

```

کد دیکودر ۴ به ۲

برای ساخت mux ها می شود از گیت های and و or استفاده کرد و به شکل زیر ساختار و کد یک دیکودر را ساخت و سیگنال های مد نظر های آن را تست کرد



سیگنال مالتی پلکسر به ۱

**mux\_2\_1 (struct\_mux\_2\_1)**

- std.standard
- std.TEXTIO
- ieee.std\_logic\_1164

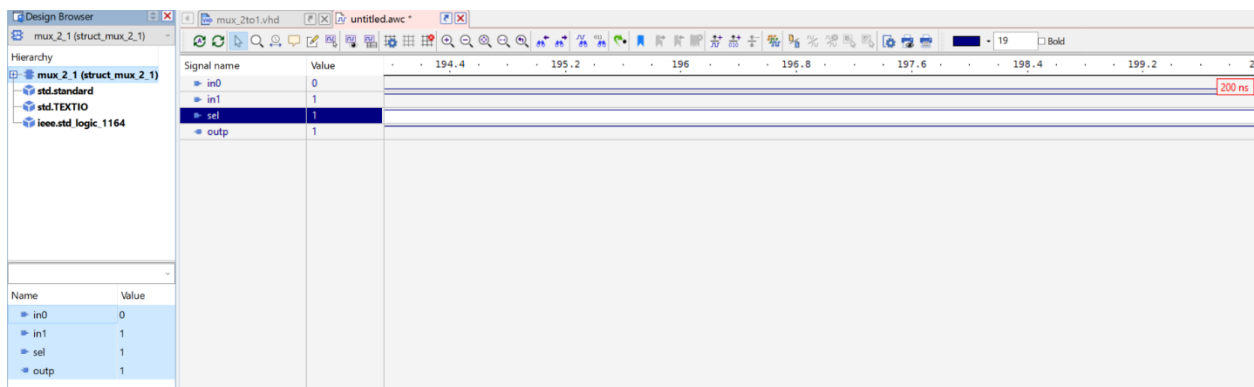
Name	Value
in0	0
in1	1
sel	1
outp	1

```

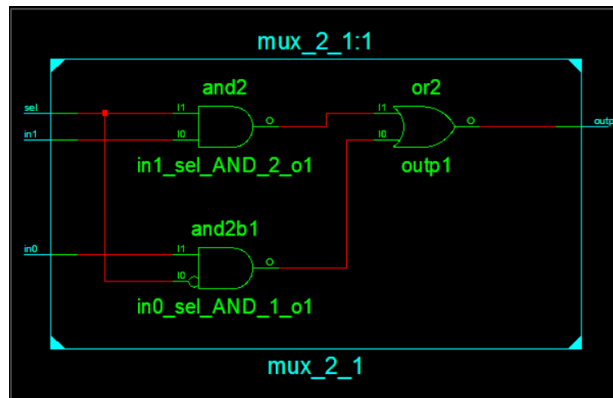
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 entity mux_2_1 is
4     port (
5         in0 , in1 :in std_logic;
6         sel :in std_logic;
7         outp:out std_logic
8     ) ;
9 end mux_2_1;
10 architecture struct_mux_2_1 of mux_2_1 is
11
12 begin
13     outp <= (in0 and not sel) or (in1 and sel);
14 end struct_mux_2_1;

```

کد مالتی پلکسر به ۱



سیگنال مالتی پلکسر به ۱ دوم



شکل مالتی پلکسر ۲ به ۱

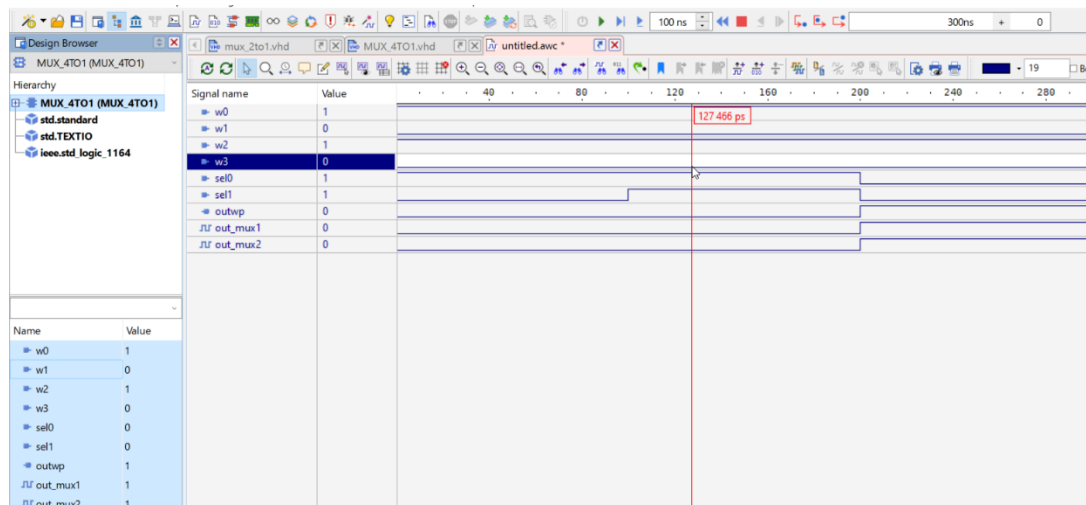
برای ساخت mux ۴ به ۱ هم از mux ۲ به ۱ ایجاد می شود و بدین سان با استفاده از ۳ تا mux ما یک mux ۴ به ۱ می سازیم.

```

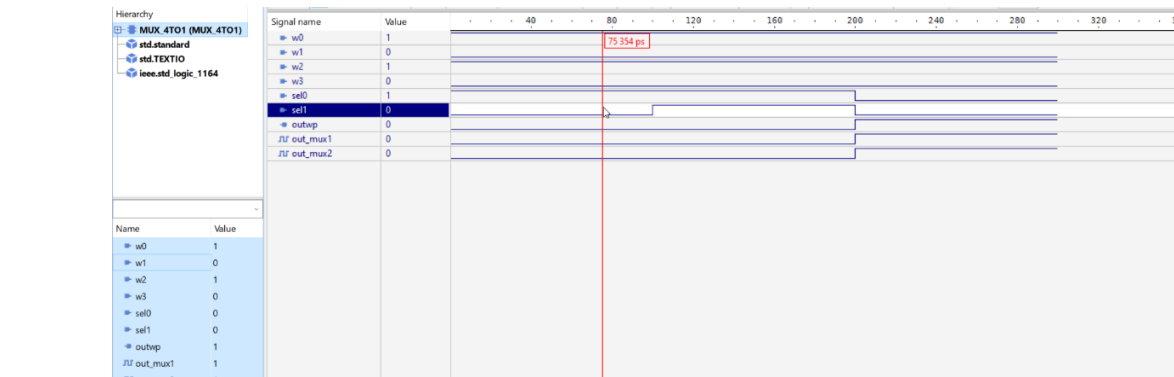
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 entity MUX_4TO1 is
4     port(
5         w0, w1, w2, w3:in std_logic;
6         sel0, sel1:in std_logic;
7         outwp : out std_logic
8     );
9 end MUX_4TO1;
10
11 architecture MUX_4TO1 of MUX_4TO1 is
12     component mux_2_1 is
13         port (
14             in0 , in1 :in std_logic;
15             sel :in std_logic;
16             outp:out std_logic
17         );
18     end component;
19     signal out_mux1 , out_mux2 :std_logic;
20
21 begin
22     mux_ins1: mux_2_1 port map (in0 => w0 , in1 => w1 , outp=> out_mux1 , sel => sel0);
23     mux_ins2: mux_2_1 port map (in0 => w2 , in1 => w3 , outp => out_mux2 , sel => sel0);
24     mux_ins3: mux_2_1 port map (in0 => out_mux1 , in1 => out_mux2 , outp => outwp , sel=> sel1);
25
26 end MUX_4TO1;

```

کد mux ۴ به ۱



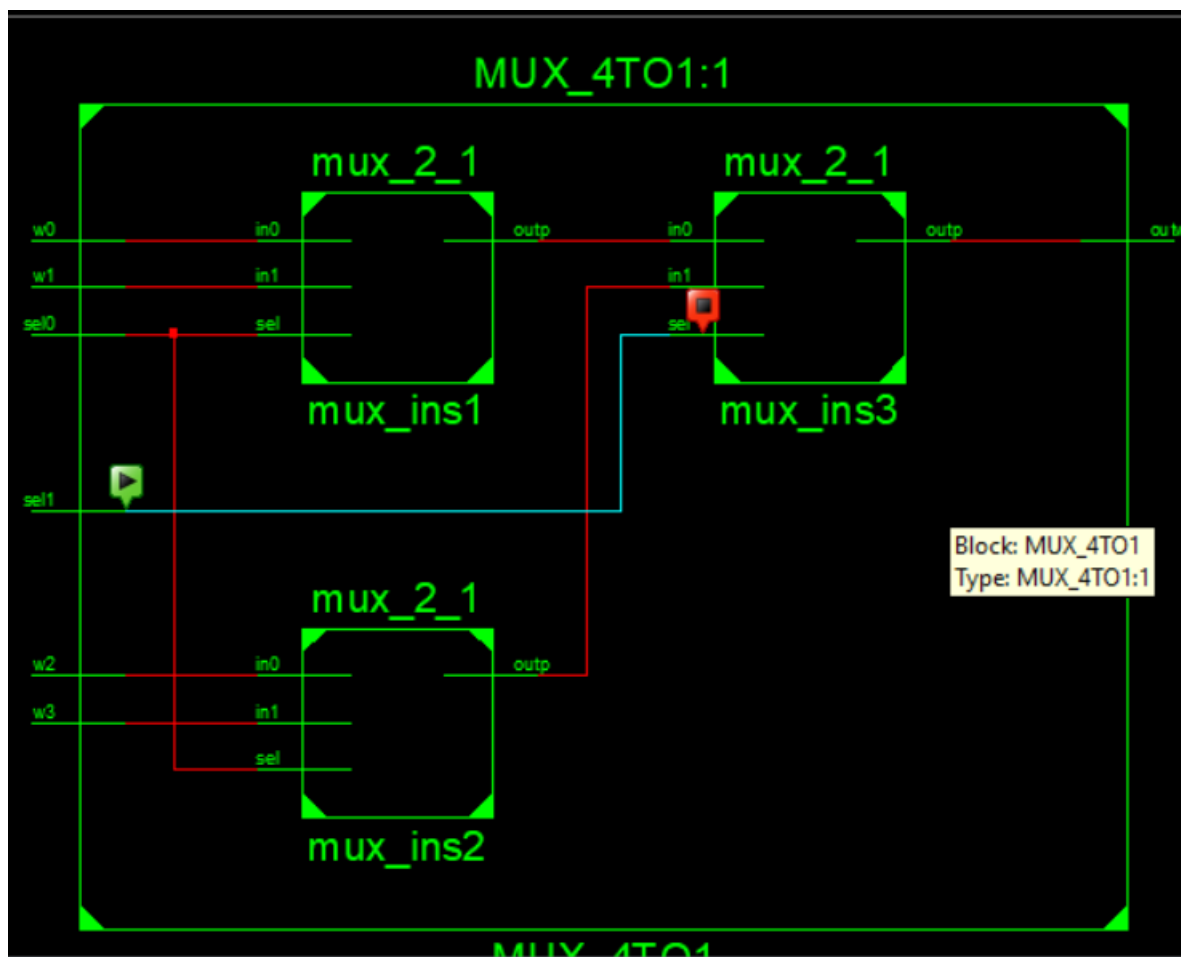
سیگنال اول



سیگنال ۲



سیگنال ۳



شکل نمای مالتی پلکسر ۴ به ۱