

"گزارشکار آزمایش ۲"

عنوان VHDL مقدماتی

مقدمه (توصیف جمع کننده، مقایسه کننده با استفاده از برخی قطعات تعریف شده).

هدف

الف- طراحی مدار مقایسه کننده تک بیتی و استفاده از گیت xor که با گیتهای nand پیاده سازی شده

ب- طراحی مقایسه کننده با استفاده از مالتی پلکسر 4x1 و حداقل تعداد گیت

ج- طراحی دیکودر ۲ به ۴ سپس طراحی یک مالتی پلکسر ۴ به یک با استفاده از دیکودر و حداقل تعداد

گیت or و and و طراحی یک جمع کننده با استفاده از مالتی پلکسر طراحی شده و گیت Nand

شرح آزمایش (

```
entity xor_gate is
  Port ( A : in  STD_LOGIC;
        B : in  STD_LOGIC;
        Q : out  STD_LOGIC);
end xor_gate;
```

```
architecture Behavioral of xor_gate is
```

```
  signal w0,w1,w2: std_logic ;
```

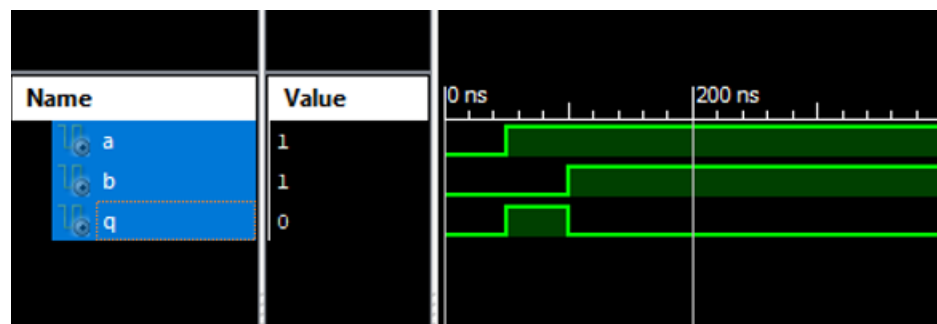
```
begin
  w0 <= A nand B;
  w1 <= w0 nand A;
  w2 <= w0 nand B;
  q <= w1 nand w2;
```

```
end Behavioral;
```

الف- طراحی مدار مقایسه کننده تک بیتی:

ابتدا باید گیت XOR را با گیت NAND توصیف کنیم :

خروجی به این صورت خواهد بود:



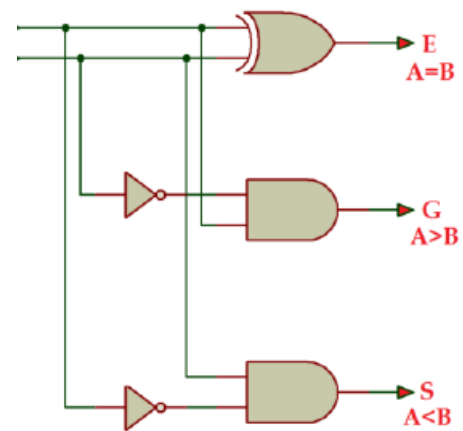
سپس با استفاده از گیت AND,NOT توصیف را تکمیل میکنیم:

```
entity comparator_lbit is
port(
A ,B : in std_logic ;
E,G,S : out std_logic
);
end comparator_lbit;

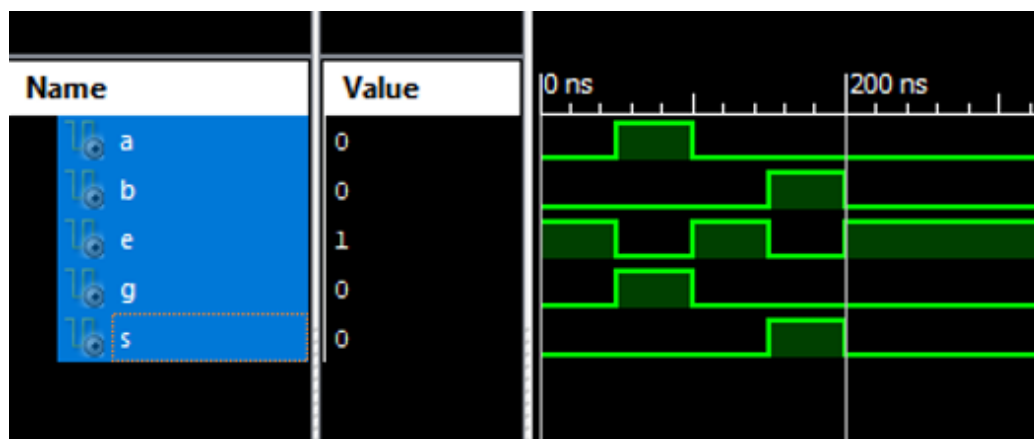
architecture Behavioral of comparator_lbit is

component xor_gate is
Port ( A : in  STD_LOGIC;
      B : in  STD_LOGIC;
      Q : out  STD_LOGIC);
end component ;

signal e0 , not_A, not_B : std_logic;
begin
not_A <= not A;
not_B <= not B;
instance0: xor_gate port map ( A => A ,B => B, Q => e0);
E <= not e0 ;
G <= not_b and A;
S <= not_A and B;
end Behavioral;
```



خروجی به این صورت خواهد بود:



ب- طراحی مقایسه کننده

ابتدا یک مالتی پلکسر 4X1 طراحی میکنیم:

```
entity Mux_4x1 is
    Port ( input : in  STD_LOGIC_VECTOR (3 downto 0);
          output : out  STD_LOGIC;
          sel : in  STD_LOGIC_VECTOR (1 downto 0));
end Mux_4x1;

architecture Behavioral of Mux_4x1 is

    signal sel_not :std_logic_vector(1 downto 0);
    signal result :std_logic_vector(3 downto 0);

begin

    sel_not(0) <= not sel(0);
    sel_not(1) <= not sel(1);

    result(0) <= sel_not(1) and sel_not(0) and input(0);
    result(1) <= sel_not(1) and sel(0) and input(1);
    result(2) <= sel(1) and sel_not(0) and input(2);
    result(3) <= sel(1) and sel(0) and input(3);
    output <= result(0) or result(1) or result(2) or result(3);

end Behavioral;
```

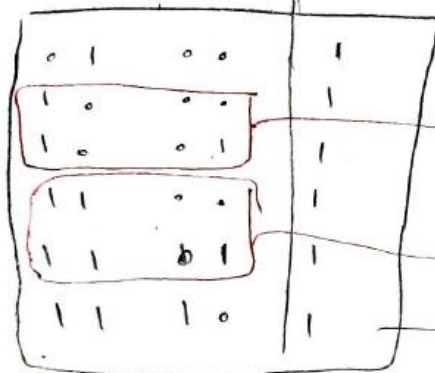
خروجی ماکس به این صورت خواهد بود:



سپس مدار مدنظر برای توصیف مقایسه کننده را رسم میکنیم:

comparator by mux:

a_1	a_0	b_1	b_0	$A > B \rightarrow 1$
1	0	0	0	1
1	0	0	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0



$$a_1 a_0' b_1'$$

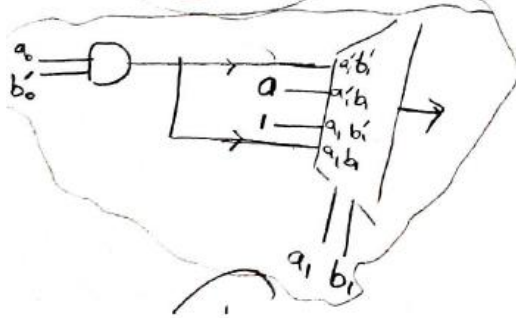
$$a_1 a_0' b_1' b_0'$$

$$a_1 a_0' b_1' b_0$$

$$a_1 a_0 b_1 b_0'$$

$$a_1 b_1' (a_0 + a_0') = a_1 b_1'$$

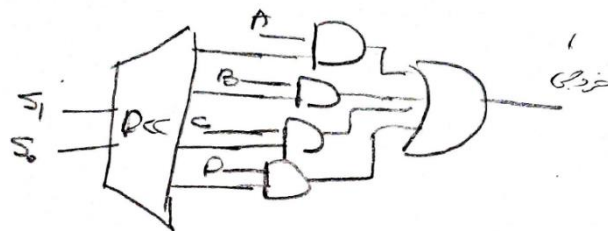
$$a_1' a_0' b_1' b_0' + a_1 a_0' b_1' + a_1 a_0' b_1' b_0 + a_1 a_0 b_1 b_0'$$



$$AND = \text{mux}, \text{mux} = \oplus$$

comparator

$$MUX \rightarrow S_1' S_0' A + S_1' S_0 B + \dots$$



سپس مدار رسم شده را توصیف میکنیم:

```
entity comparator_2bit is
    Port ( a1 : in  STD_LOGIC;
          a0 : in  STD_LOGIC;
          b1 : in  STD_LOGIC;
          b0 : in  STD_LOGIC;
          output : out  STD_LOGIC);
end comparator_2bit;

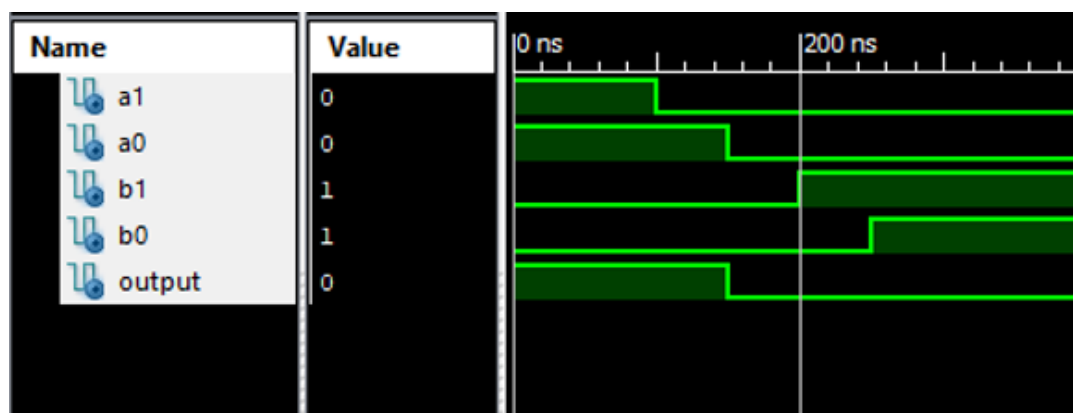
architecture Behavioral of comparator_2bit is

    component Mux_4x1 is
        Port ( input : in  STD_LOGIC_VECTOR (3 downto 0);
              output : out  STD_LOGIC;
              sel : in  STD_LOGIC_VECTOR (1 downto 0));
    end component ;

    signal and_1 , b0_not : std_logic ;

begin
    b0_not <= not b0;
    and_1 <= a0 and b0_not;
    instance0 : Mux_4x1 port map( input(0) =>and_1 , input(1) => '0' , input(2) => '1',input(3) =>and_1,
                                   output =>output , sel(0) =>b1 , sel(1) =>a1 );
end Behavioral;
```

خروجی به صورت زیر خواهد بود:



ج-

۱) طراحی مالتی پلکسر 4X1 با استفاده از دیکودر 2X4:

ابتدا یک دیکودر ۲ به ۴ طراحی و توصیف میکنیم:

```
entity Decoder_2x4 is
  Port ( input : in  STD_LOGIC_VECTOR (1 downto 0);
        output : out STD_LOGIC_VECTOR (3 downto 0));
end Decoder_2x4;

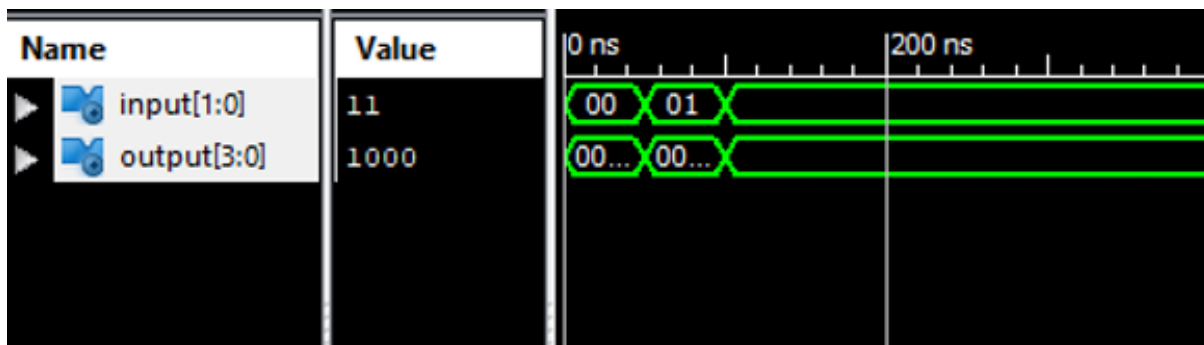
architecture Behavioral of Decoder_2x4 is
  signal w :std_logic_vector (1 downto 0);
begin

  w(0) <= not input(0);
  w(1) <= not input(1);

  output(0) <= w(0) and w(1);
  output(1) <= input(0) and w(1) ;
  output(2) <= w(0) and input(1);
  output(3) <=input(0) and input(1);

end Behavioral;
```

نتیجه به صورت زیر خواهد بود:



حال باید مالتی پلکسر چهار به یک را با استفاده از دیکودر طراحی شده توصیف کنیم :

```
entity Mux_4x1_byDecoder is
  Port ( input : in  STD_LOGIC_VECTOR (3 downto 0);
        sel : in  STD_LOGIC_VECTOR (1 downto 0);
        output : out  STD_LOGIC);
end Mux_4x1_byDecoder;

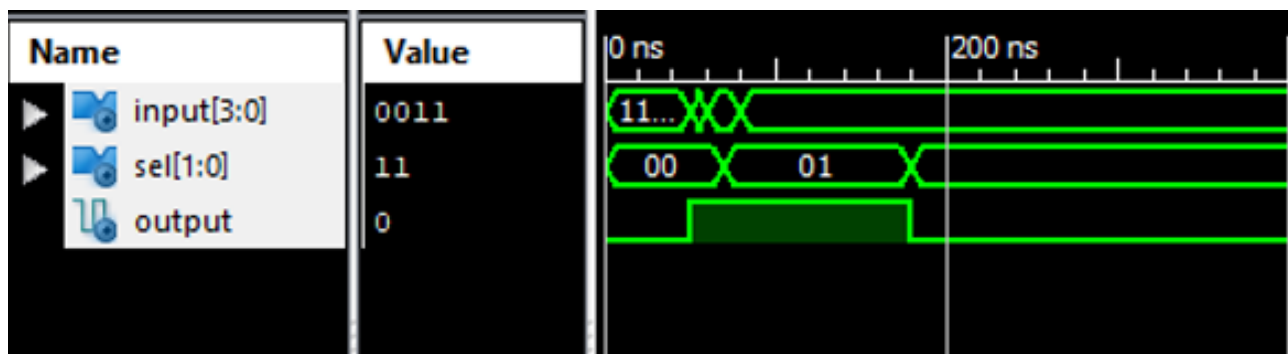
architecture Behavioral of Mux_4x1_byDecoder is
  component Decoder_2x4 is
    Port ( input : in  STD_LOGIC_VECTOR (1 downto 0);
          output : out  STD_LOGIC_VECTOR (3 downto 0));
  end component ;

  signal w0, w1,w2,w3,and_0,and_1,and_2,and_3 : std_logic ;
begin
  instance0 : Decoder_2x4 port map ( input(0) =>sel(0) , input(1) =>sel(1) , output(0) => w0 ,output(1) => w1 ,
                                     output(2) => w2 , output(3) => w3 );

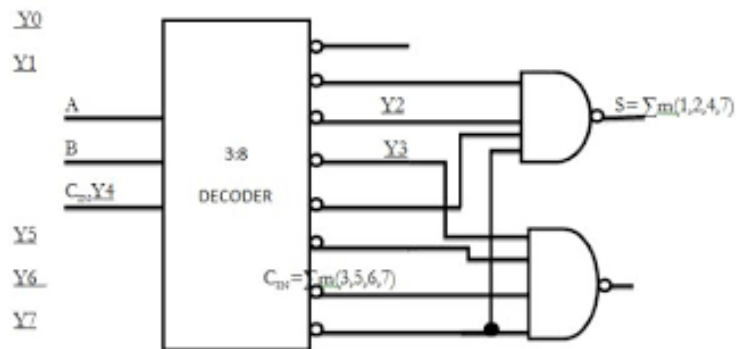
  and_0 <= w0 and input(0);
  and_1 <= w1 and input(1);
  and_2 <= w2 and input(2);
  and_3 <= w3 and input(3);

  output <= and_1 or and_2 or and_3 or and_0 ;
end Behavioral;
```

خروجی بصورت زیر خواهد بود :



۲) طراحی جمع کننده با استفاده از دیکودر و گیت NAND



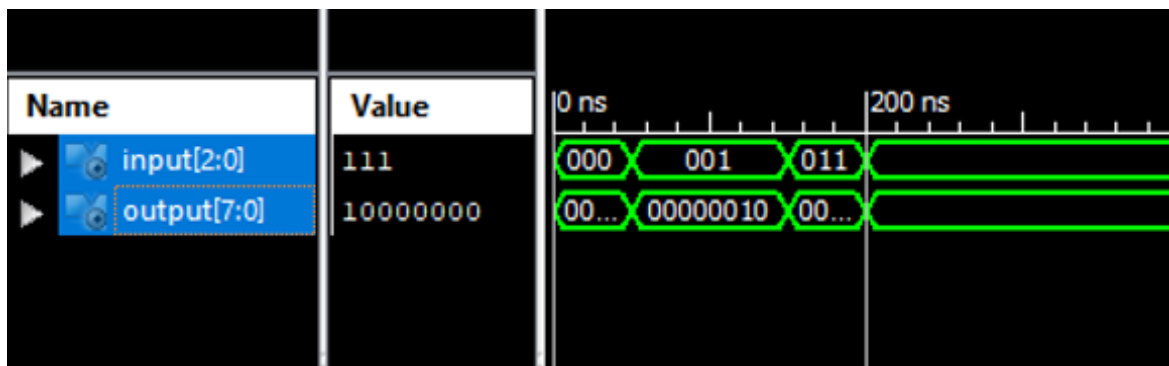
با توجه به مدار بالا ابتدا باید یک دیکودر ۳ به ۸ طراحی کنیم:

```
entity decoder_3to8 is
    Port ( input : in  STD_LOGIC_VECTOR (2 downto 0);
          output : out STD_LOGIC_VECTOR (7 downto 0));
end decoder_3to8;

architecture Behavioral of decoder_3to8 is
    signal w :std_logic_vector (2 downto 0);
begin
    w(0) <= not input(0);
    w(1) <= not input(1);
    w(2) <= not input(2);
    output(0) <= w(0) and w(1) and w(2);
    output(1) <= input(0) and w(1) and w(2);
    output(2) <= w(0) and input(1) and w(2);
    output(3) <=input(0) and input(1) and w(2);
    output(4) <= w(0) and w(1) and input (2);
    output(5) <= input(0) and w(1) and input (2);
    output(6) <= w(0) and input(1) and input (2);
    output(7) <= input(0) and input(1) and input (2);

end Behavioral;
```

خروجی :



سپس با استفاده از دیکودر توصیف شده و گیت NAND جمع کننده را توصیف میکنیم:

```
entity adder is
    Port ( A : in  STD_LOGIC;
          B : in  STD_LOGIC;
          Cin : in  STD_LOGIC;
          S : out  STD_LOGIC;
          Cout : out  STD_LOGIC);
end adder;

architecture Behavioral of adder is
    component decoder_3to8 is
        Port ( input : in  STD_LOGIC_VECTOR (2 downto 0);
              output : out  STD_LOGIC_VECTOR (7 downto 0));
    end component;
    signal m : std_logic_vector(7 downto 0);
    signal not_m : std_logic_vector(7 downto 0);
    signal s0,c0 :std_logic ;
begin
    instance0 : decoder_3to8 port map (input(0) => A, input(1) =>B , input(2) =>Cin,
        output(0)=>m(0),output(1)=>m(1),output(2)=>m(2),output(3)=>m(3),output(4)=>m(4),output(5)=>m(5),output(6)=>m(6),output(7)
        not_m(0)<= not m(0);    not_m(1)<= not m(1);
        not_m(2)<= not m(2);    not_m(3)<= not m(3);
        not_m(4)<= not m(4);    not_m(5)<= not m(5);
        not_m(6)<= not m(6);    not_m(7)<= not m(7);
        s0<= not_m(1) and not_m(2) and not_m(4) and not_m(7) ;
        S <= not s0 ;
        c0 <= not_m(3) and not_m(5) and not_m(6) and not_m(7) ;
        Cout <= not c0 ;
```

خروجی نهایی بصورت زیر خواهد بود:

