

## ۱. واحد Parity detector

پرتی زوج را در اینجا تشخیص خواهیم داد بدین منظور باید تک تک بیت های رشته مورد نظر با یک دیگر XOR شوند .

کد نوشته شده :

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.NUMERIC_STD.ALL;
4
5  entity Parity_8bit is
6      Port ( data : in  STD_LOGIC_VECTOR (7 downto 0);
7            P : out  STD_LOGIC;
8            en : in  STD_LOGIC);
9  end Parity_8bit;
10
11  architecture Behavioral of Parity_8bit is
12
13
14  begin
15
16  process(data ,en)
17  begin
18      P<='Z' ;
19      if en='1' then
20          P<=(data(0)xor data(1))xor (data(2)xor data(3)) xor (data(4)xor data(5))xor (data(6)xor data(7)) ;
21      end if ;
22  end process ;
23  end Behavioral;
24
25
26
```

کد شبیه سازی :

```

1
2  LIBRARY ieee;
3  USE ieee.std_logic_1164.ALL;
4  USE ieee.numeric_std.ALL;
5
6  ENTITY Parity_test IS
7  END Parity_test;
8
9  ARCHITECTURE behavior OF Parity_test IS
10
11     -- Component Declaration for the Unit Under Test (UUT)
12
13     COMPONENT Parity_8bit
14     PORT(
15         data : IN  std_logic_vector(7 downto 0);
16         P : OUT  std_logic;
17         en : IN  std_logic
18     );
19     END COMPONENT;
20
21
22     --Inputs
23     signal data : std_logic_vector(7 downto 0) := (others => '0');
24     signal en : std_logic := '0';
25
26     --Outputs
27     signal P : std_logic;
28     -- No clocks detected in port list. Replace <clock> below with
29     -- appropriate port name

```

```

31
32  BEGIN
33
34     -- Instantiate the Unit Under Test (UUT)
35     uut: Parity_8bit PORT MAP (
36         data => data,
37         P => P,
38         en => en
39     );
40
41
42     -- Stimulus process
43     stim_proc: process
44     begin
45         -- hold reset state for 100 ns.
46         wait for 100 ns;
47         en<='1' ;
48         data<="10101010" ;
49
50         wait for 100 ns;
51         en<='0' ;
52
53         wait for 100 ns;
54         en<='1' ;
55         data<="10101110" ;
56

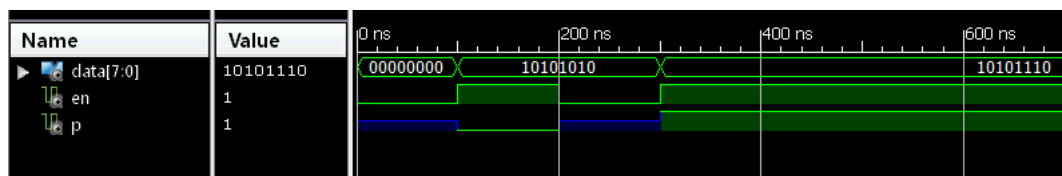
```

```

56
57         -- insert stimulus here
58
59         wait;
60     end process;
61
62  END;
63

```

خروجی مشاهده شده :



در ابتدا EN=0 می باشد و خروجی Z می باشد. سپس فعالساز، ۱ می شود و برای عددی که تعداد زوج ۱ دارد مقدار ۰ در خروجی را مشاهده می کنیم. سپس فعالساز قطع می شود و خروجی مدار های امیدانس می شود. سپس با اعمال عددی که تعداد فرد ۱ دارد مشاهده می شود خروجی برابر با ۱ می گردد.

۲. حافظه ۱ کیلو بیتی

حافظه ۱ کیلو بیتی با خط داده ۸ بیتی دارای ۱۲۸ سطر خواهد بود که به معنای ۷ خط برای آدرس دهی می باشد. ابتدا با تعریف یک نوع داده آرایه ای که دارای ۱۲۸ سطر ۸ بیتی هست این حافظه را ایجاد می کنیم. اگر فعالساز نوشتن ۱ باشد Din به سطری که پایه آدرس اشاره می کند لچ خواهد شد. اگر فعالساز ۰ باشد Dout برابر مقدار لچ شده ای که خط آدرس به آن اشاره می کند خواهد بود.

کد نوشته شده :

```

1
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4 use IEEE.NUMERIC_STD.ALL;
5
6 entity MEM128x8 is
7     Port ( Din : in  STD_LOGIC_VECTOR (7 downto 0);
8           Dout : out STD_LOGIC_VECTOR (7 downto 0);
9           Add : in  STD_LOGIC_VECTOR (6 downto 0);
10          Wr_EN : in  STD_LOGIC);
11 end MEM128x8;
12
13 architecture Behavioral of MEM128x8 is
14
15     type MEM128x8 is array (0 to 127) of STD_LOGIC_VECTOR (7 downto 0) ;
16     signal RAM1 : MEM128x8 :=(Others=>(others => '0'));
17
18 begin
19
20     process (Din,Add,Wr_EN)
21     begin
22
23         if Wr_EN='1' then
24
25             RAM1(to_integer(unsigned (Add))) <= Din ;
26
27         end if ;
28
29         Dout <= RAM1(to_integer(unsigned (Add)));
30
31     End Process;
32
33 end Behavioral;
34
35

```

```

1 |
2  LIBRARY ieee;
3  USE ieee.std_logic_1164.ALL;
4  USE ieee.numeric_std.ALL;
5
6  ENTITY TEST IS
7  END TEST;
8
9  ARCHITECTURE behavior OF TEST IS
10
11      -- Component Declaration for the Unit Under Test (UUT)
12
13      COMPONENT MEM128x8
14      PORT(
15          Din : IN  std_logic_vector(7 downto 0);
16          Dout : OUT std_logic_vector(7 downto 0);
17          Add : IN  std_logic_vector(6 downto 0);
18          Wr_EN : IN  std_logic
19      );
20      END COMPONENT;
21
22
23      --Inputs
24      signal Din : std_logic_vector(7 downto 0) := (others => '0');
25      signal Add : std_logic_vector(6 downto 0) := (others => '0');
26      signal Wr_EN : std_logic := '0';
27
28      --Outputs
29      signal Dout : std_logic_vector(7 downto 0);
30      -- No clocks detected in port list. Replace <clock> below with
31
32      -- appropriate port name
33
34  BEGIN
35
36      -- Instantiate the Unit Under Test (UUT)
37      uut: MEM128x8 PORT MAP (
38          Din => Din,
39          Dout => Dout,
40          Add => Add,
41          Wr_EN => Wr_EN
42      );
43
44      -- Stimulus process
45      stim_proc: process
46      begin
47          -- hold reset state for 100 ns.
48          wait for 100 ns;
49
50          Din<= "10101010" ;
51          Add<= "0000010" ;
52
53          wait for 100 ns;
54
55          Wr_EN <= '1' ;
56
57          wait for 100 ns;
58
59          Wr_EN <= '0' ;
60          Add<= "0000010" ;

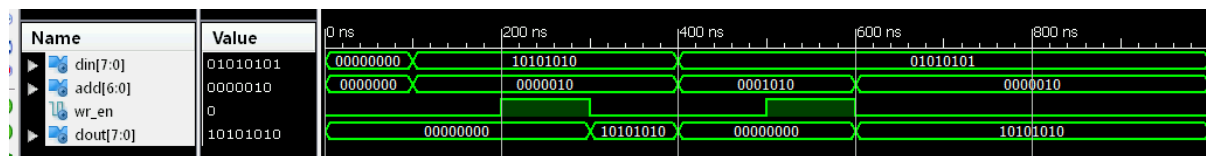
```

```

61
62     wait for 100 ns;
63
64     Add<= "0001010" ;
65
66     wait for 100 ns;
67
68     Din<= "01010101" ;
69
70
71
72     -- insert stimulus here
73
74     wait;
75 end process;
76
77 END;
78

```

خروجی :



مقدار Add و Din را مشخص می کنیم و پایه فعالساز را ۱۰۰ نانو ثانیه فعال می کنیم در نتیجه مقدار حافظه تغییر خواهد کرد سپس با غیر فعال شدن فعالساز چون ادرس هنوز تغییر نکرده است همین مقدار بر روی خروجی خواهد بود. سپس ادرس را تغییر می دهیم و چون ابتدا همه خانه ها صفر هستند خروجی صفر می شود ( خانه اشاره شده به آن صفر است) سپس یک مقدار درون آن می ریزیم ( جهت بررسی کار کردن همه لچ ها ) . در آخر با اشاره به ادرسی که در ابتدا مقدار دهی شده بود مقدار آن را در خروجی مشاهده می کنیم.