

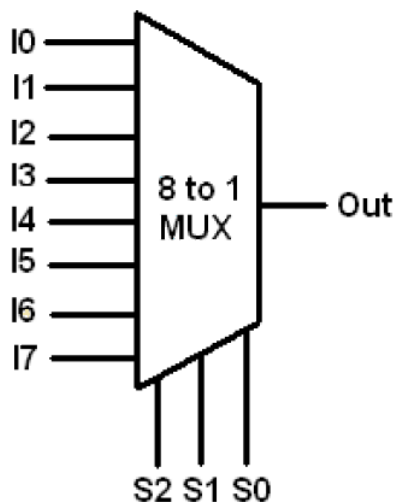
گزارش کار آزمایشگاه معماری

یاسمین مدنی-زهرا مومنی نژاد



مالتی پلکسر

مالتی پلکسر ۸ به ۱ یک دستگاه دیجیتالی است که با استفاده از خط انتخاب سه بیتی یکی از هشت خط ورودی به خط خروجی می برد.



شکل زیر بلوک دیاگرام این مالتی پلکسر را نمایش می دهد.

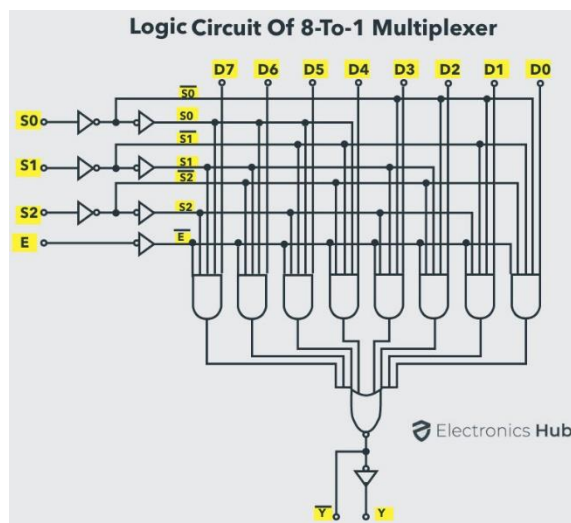
جدول درستی این وسیله نیز به صورت زیر نمایش داده می شود.

S0	S1	S2	D0	D1	D2	D3	D4	D5	D6	D7	Y
0	0	0	0	X	X	X	X	X	X	X	0
0	0	0	1	X	X	X	X	X	X	X	1
0	0	1	X	0	X	X	X	X	X	X	0
0	0	1	X	1	X	X	X	X	X	X	1
0	1	0	X	X	0	X	X	X	X	X	0
0	1	0	X	X	1	X	X	X	X	X	1
0	1	1	X	X	X	0	X	X	X	X	0
0	1	1	X	X	X	1	X	X	X	X	1
1	0	0	X	X	X	X	0	X	X	X	0
1	0	0	X	X	X	X	1	X	X	X	1
1	0	1	X	X	X	X	X	0	X	X	0
1	0	1	X	X	X	X	X	1	X	X	1
1	1	0	X	X	X	X	X	X	0	X	0
1	1	0	X	X	X	X	X	X	1	X	1
1	1	1	X	X	X	X	X	X	X	0	0
1	1	1	X	X	X	X	X	X	X	1	1

از جدول درستی آن می توان عبارت جبری را برای آن به این صورت به دست آورد.

$$Y = \overline{S_0} \overline{S_1} \overline{S_2} D_0 + \overline{S_0} \overline{S_1} \overline{S_2} D_1 + \overline{S_0} \overline{S_1} S_2 D_2 + \overline{S_0} S_1 \overline{S_2} D_3 + \overline{S_0} S_1 S_2 D_4 + S_0 \overline{S_1} \overline{S_2} D_5 + S_0 \overline{S_1} S_2 D_6 + S_0 S_1 \overline{S_2} D_7$$

مدار داخلی را با استفاده از عبارت بالا و گیت های منطقی به صورت زیر طراحی می کنیم.



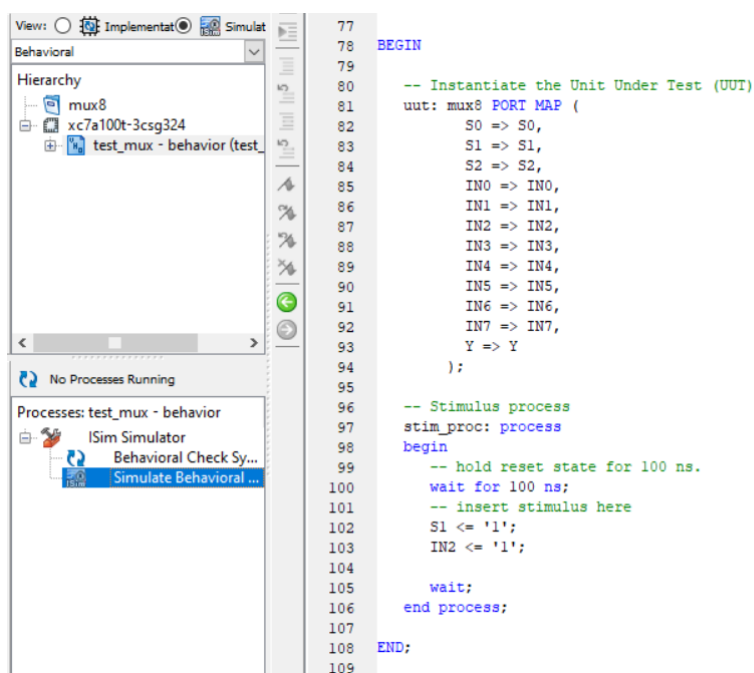
کد مدار را به صورت رفتار مدار پیاده سازی می کنیم. تصویر زیر کد پیاده سازی شده در VHDL را نمایش می دهد.

```

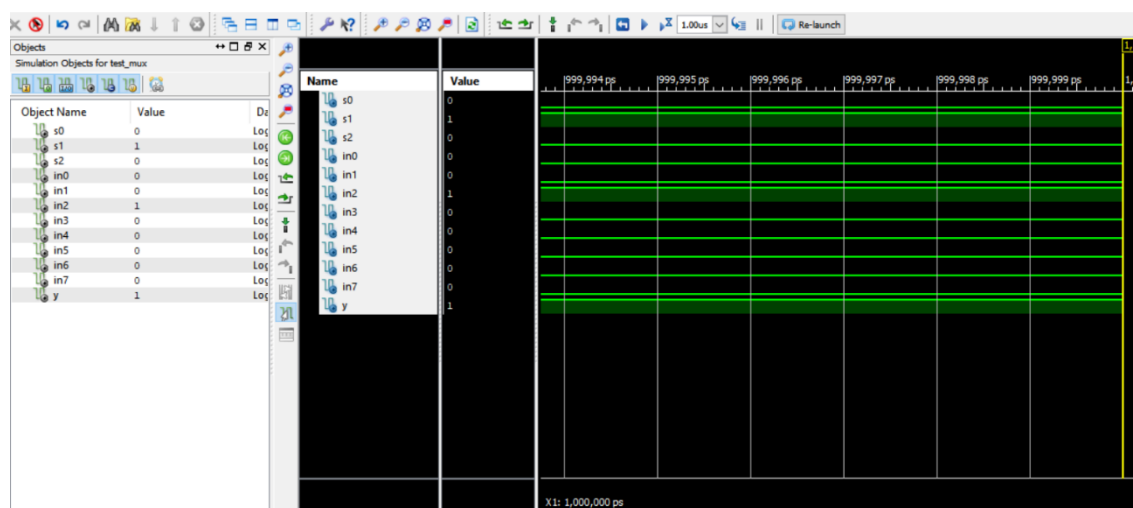
31
32 entity mux8 is
33     Port ( S0 : in  STD_LOGIC;
34           S1 : in  STD_LOGIC;
35           S2 : in  STD_LOGIC;
36           IN0 : in  STD_LOGIC;
37           IN1 : in  STD_LOGIC;
38           IN2 : in  STD_LOGIC;
39           IN3 : in  STD_LOGIC;
40           IN4 : in  STD_LOGIC;
41           IN5 : in  STD_LOGIC;
42           IN6 : in  STD_LOGIC;
43           IN7 : in  STD_LOGIC;
44           Y : out  STD_LOGIC);
45 end mux8;
46
47 architecture Behavioral of mux8 is
48     Signal a0,a1,a2,a3,a4,a5,a6,a7:STD_LOGIC;
49     begin
50         a0 <= IN0 and (not s0) and (not s1) and (not s2);
51         a1 <= IN1 and (not s0) and (not s1) and s2;
52         a2 <= IN2 and (not s0) and s1 and (not s2);
53         a3 <= IN3 and (not s0) and s1 and s2;
54         a4 <= IN4 and s0 and (not s1) and (not s2);
55         a5 <= IN5 and s0 and (not s1) and s2;
56         a6 <= IN6 and s0 and s1 and (not s2);
57         a7 <= IN7 and s0 and s1 and s2;
58
59         Y <= a0 or a1 or a2 or a3 or a4 or a5 or a6 or a7;
60
61     end Behavioral;
62
63

```

برای سنجش درستی و تست مدار برای آن تست بنچی به صورت زیر نوشتیم.



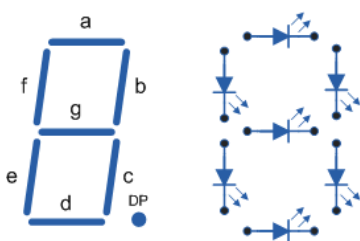
در ادامه خروجی نمودار شبیه سازی را مشاهده می کنیم.



تبدیل BCD به سون سگمنت

آی‌سی‌های دیکدر (Decoder) قطعاتی هستند که یک کد باینری در ورودی را به یک کد باینری دیگر در خروجی تبدیل می‌کنند. یکی از پرکاربردترین این آی‌سی‌ها، دیکدر تبدیل کد BCD به کد «نمایشگر هفت قسمتی» یا سون سگمنت است.

عموماً نمایشگرهای سون سگمنت از هفت LED رنگی که به آنها سگمنت گفته می‌شود، تشکیل شده‌اند. این هفت LED در کنار یکدیگر قرار گرفته و یک آی‌سی سون سگمنت را می‌سازند. برای نمایش هر یک از اعداد از 0 تا 9 و کاراکترهای هگزادسیمال لازم است که ترکیب خاصی از این LED ها روشن شوند.



به طور کلی در مدارهای الکترونیکی دو دسته نمایشگر دیجیتالی سون سگمنت وجود دارد.

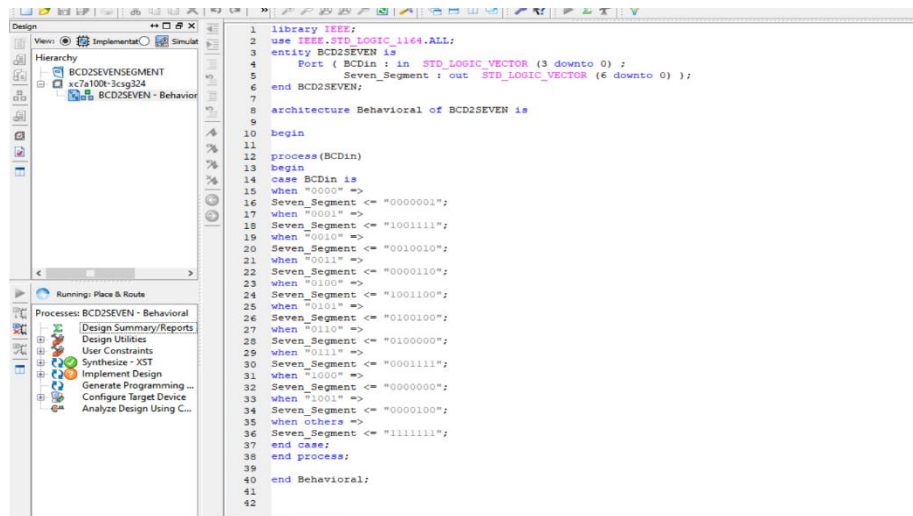
- نمایشگر کاتد مشترک که برای روشن شدن هر کدام از سگمنت‌ها، تنها لازم است که به پایه‌ی ورودی آن سگمنت ۱ منطقی اعمال کنیم.
- نمایشگر آند مشترک که برای روشن شدن هر کدام از سگمنت‌ها، تنها لازم است که به پایه‌ی ورودی آن سگمنت ۰ منطقی اعمال کنیم.

جدول درستی مربوط به هر کدام از اعداد برای نمایش روی سون سگمنت به شکل زیر است.

Digit	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

با استفاده از این جدول می‌توان رفتار مدار را تعریف کرد.

در تصویر زیر، کد مبدل، با فرض آنکه نمایشگر آند مشترک هستند پیاده سازی شده است.

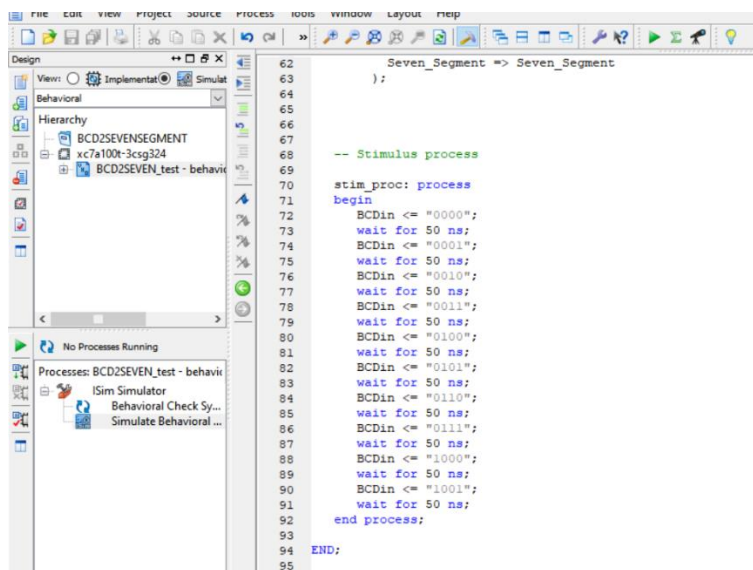


```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 entity BCD2SEVEN is
4     Port ( BCDin : in  STD_LOGIC_VECTOR (3 downto 0) ;
5           Seven_Segment : out  STD_LOGIC_VECTOR (6 downto 0) );
6 end BCD2SEVEN;
7
8 architecture Behavioral of BCD2SEVEN is
9
10 begin
11
12     process(BCDin)
13     begin
14         case BCDin is
15             when "0000" =>
16                 Seven_Segment <= "0000001";
17             when "0001" =>
18                 Seven_Segment <= "1001111";
19             when "0010" =>
20                 Seven_Segment <= "0010010";
21             when "0011" =>
22                 Seven_Segment <= "0000110";
23             when "0100" =>
24                 Seven_Segment <= "1001100";
25             when "0101" =>
26                 Seven_Segment <= "0100100";
27             when "0110" =>
28                 Seven_Segment <= "0100000";
29             when "0111" =>
30                 Seven_Segment <= "0001111";
31             when "1000" =>
32                 Seven_Segment <= "0000000";
33             when "1001" =>
34                 Seven_Segment <= "0000100";
35             when Others =>
36                 Seven_Segment <= "1111111";
37         end case;
38     end process;
39
40 end Behavioral;
41
42

```

برای تست مدار طراحی شده از تست بنچ استفاده کرده ایم که کد تست مربوطه را در شکل زیر می توان مشاهده کرد.

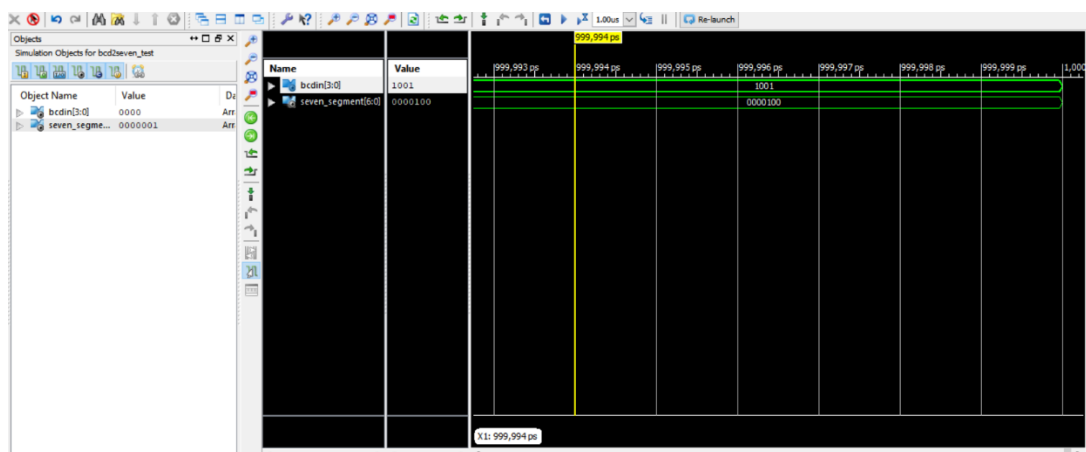


```

62     Seven_Segment => Seven_Segment
63 );
64
65
66
67
68 -- Stimulus process
69
70 stim_proc: process
71 begin
72     BCDin <= "0000";
73     wait for 50 ns;
74     BCDin <= "0001";
75     wait for 50 ns;
76     BCDin <= "0010";
77     wait for 50 ns;
78     BCDin <= "0011";
79     wait for 50 ns;
80     BCDin <= "0100";
81     wait for 50 ns;
82     BCDin <= "0101";
83     wait for 50 ns;
84     BCDin <= "0110";
85     wait for 50 ns;
86     BCDin <= "0111";
87     wait for 50 ns;
88     BCDin <= "1000";
89     wait for 50 ns;
90     BCDin <= "1001";
91     wait for 50 ns;
92 end process;
93
94 END;
95

```

نمودار شبیه سازی مدار در تصویر زیر به نمایش گذاشته شده است.



به علاوه کد هر دو قسمت به همراه تست ها در فایللی جداگانه به این فایل پیوست شده است.