

Fudan University

復旦大學

可編程器件與硬件描述語言

作業報告

姓名：名字

學號：123456789

2021 年 11 月 1 日

1 优先编码器

设计一个 BCD 的优先编码器电路，输入为 10 个开关的状态，要求输出开关对应的编码。输出编码用 4 位表示，第一个开关为 0 时，输出为 0000 时，第二个开关为 0 时，输出为 0001 时，..... 第 10 个开关为 0 时，输出为 1001。第 10 个开关的优先级最高。当没有按键按下时，输出信号 E 为 1。有按键按下时，输出信号 E 为 0。

1.1 顶层说明

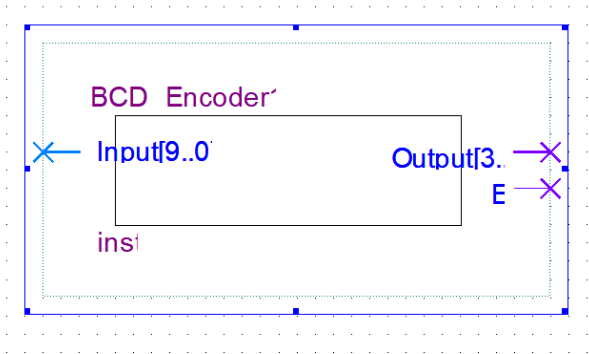


图 1: 元件示意图

Input[9:0] 为十个开关，其中 Input(9) 的优先级最高，Output[3:0] 为四位输出，E 表示是否有按键被按下。

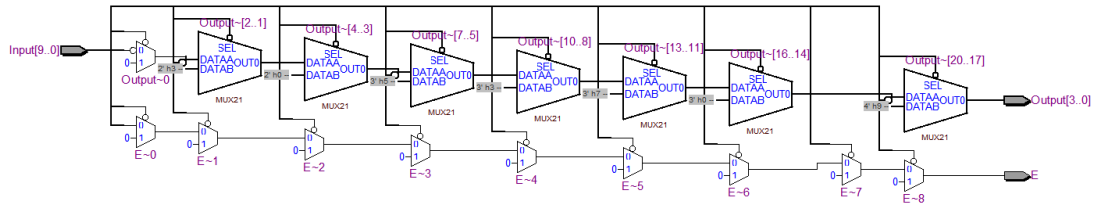


图 2: RTL 图

1.2 每个宏单元如何实现 (对所写代码的说明)

代码如下：

```

1  -----
2  -- 10-bit Priority encoder
3  -- by Zhang Ke, 2021/10/30 9:26PM
4  -----
5
6  library ieee;
7  use ieee.std_logic_1164.all;
8
9  -----
10
11 entity BCD_Encoder10 is
12     port (
13         Input  : in std_logic_vector(9 downto 0);
14         Output : out std_logic_vector(3 downto 0);
15         E      : out std_logic
16     );
17 end BCD_Encoder10;
18
19 -----
20
21 architecture rtl of BCD_Encoder10 is
22
23 begin
24     process (Input)
25     begin
26         if (Input(9) = '0') then
27             Output <= "1001";
28             E      <= '0';
29         elsif (Input(8) = '0') then
30             Output <= "1000";

```

```
31     E    <= '0';
32 elsif (Input(7) = '0') then
33     Output <= "0111";
34     E    <= '0';
35 elsif (Input(6) = '0') then
36     Output <= "0110";
37     E    <= '0';
38 elsif (Input(5) = '0') then
39     Output <= "0101";
40     E    <= '0';
41 elsif (Input(4) = '0') then
42     Output <= "0100";
43     E    <= '0';
44 elsif (Input(3) = '0') then
45     Output <= "0011";
46     E    <= '0';
47 elsif (Input(2) = '0') then
48     Output <= "0010";
49     E    <= '0';
50 elsif (Input(1) = '0') then
51     Output <= "0001";
52     E    <= '0';
53 elsif (Input(0) = '0') then
54     Output <= "0000";
55     E    <= '0';
56 else
57     Output <= "XXXX";
58     E    <= '1';
59 end if;
60 end process;
61
```

```
62 end rtl;
```

以上代码中，将元件输入 Input 定义为 10 位的标准逻辑向量类型，输出 E 和 Output 分别定义为标准逻辑位和 4 位的标准逻辑向量类型。具体 architecture 实现中使用 IF 语句进行状态判断并且输出。

1.3 时序仿真结果

编写 Test Bench 文件如下：

```
1 -----
2 -- Test Bench for 10-bit Priority encoder
3 -- by Zhang Ke, 2021/10/30 9:57PM
4 -----
5
6 library ieee;
7 use ieee.std_logic_1164.all;
8
9 -----
10
11 entity Q1_BCD_Encoder_tb is
12 end Q1_BCD_Encoder_tb;
13
14 -----
15
16 architecture tbbbehv of Q1_BCD_Encoder_tb is
17
18     component BCD_Encoder10
19     port (
20         Input  : in std_logic_vector(9 downto 0);
21         Output : out std_logic_vector(3 downto 0);
```

```

22     E      : out std_logic
23   );
24 end component;
25
26 signal Input  : std_logic_vector(9 downto 0) := "0000000000";
27 signal Output : std_logic_vector(3 downto 0);
28 signal E      : std_logic;
29 begin
30
31   U_BCD  : BCD_Encoder10 port map(Input, Output, E);
32   inputpc : process
33   begin
34     Input <= "0000000000"; wait for 20 ns; Input <= "1000000000"; wait for 20 ns;
35     Input <= "1100000000"; wait for 20 ns; Input <= "1110000000"; wait for 20 ns;
36     Input <= "1111000000"; wait for 20 ns; Input <= "1111100000"; wait for 20 ns;
37     Input <= "1111110000"; wait for 20 ns; Input <= "1111111000"; wait for 20 ns;
38     Input <= "1111111100"; wait for 20 ns; Input <= "1111111110"; wait for 20 ns;
39     Input <= "1111111111"; wait for 20 ns; Input <= "1001100000"; wait for 20 ns;
40     Input <= "1110100000"; wait for 20 ns; Input <= "1111100100"; wait for 20 ns;
41     Input <= "1111011100";
42     wait;
43   end process;
44
45 end architecture;

```

使用上述 Test Bench 文件进行仿真，波形图如下：

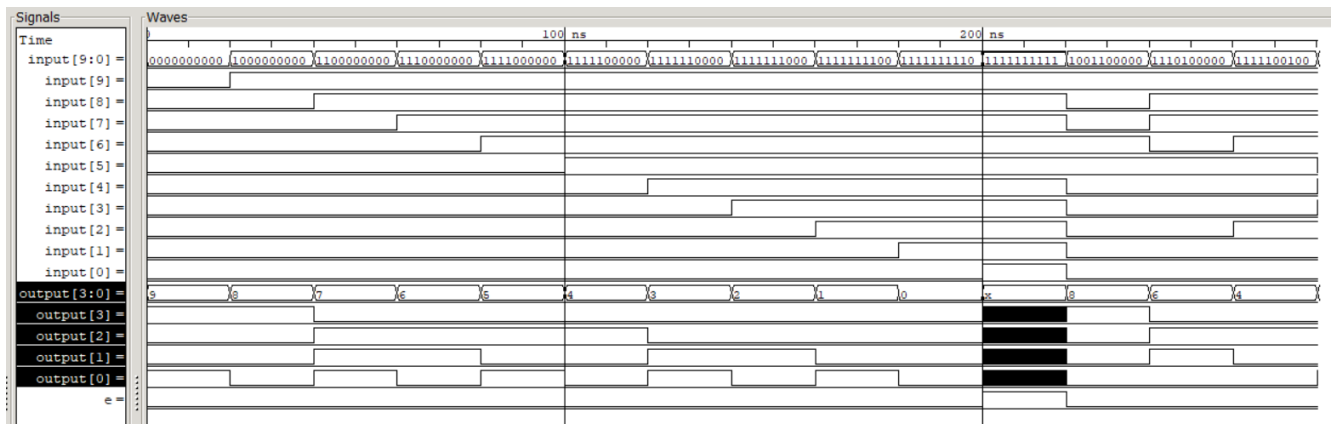


图 3: BCD 优先编码器时序仿真波形