

实验四：按键控制的状态机设计

设计要求：

- 1、 按键控制数码管显示，当按下 S2 时，Q4 数码管显示 1；再次按下 S2，Q4 数码管显示 2；第三次按下 S2，Q4 数码管显示 3；第四次按下 S2 时，Q4 数码管显示 4.第五次按下 S2 时 Q4 又从 1 开始显示，如此反复。Q4 使能对应 IO 口为 168（低电平有效），按键 S2 对应 IO 口为 125（按下为低电平）。
- 2、 八段 led 对应 IO 口为 A~H (144,158,162,160,159,156,163,161)。
- 3、 检测按键是否按下需要消抖，消抖程序请同学们参考附录程序，参考程序是利用四个按键控制四个 LED 灯的亮灭。请同学们在读懂参考程序的基础上将其改编为符合要求 1 的程序。
- 4、 参考程序是普通的控制程序，本实验要求同学们利用状态机来编写程序。
- 5、 顶层模块命名方式，key_state_(班级)_(班级序号)。

附：八段 led 数码管编码表（1~4）

	A	B	C	D	E	F	G	H
1	1	0	0	1	1	1	1	0
2	0	0	1	0	0	1	0	0
3	0	0	0	0	1	1	0	0
4	1	0	0	1	1	0	0	0

参考程序

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
USE IEEE.std_logic_unsigned.all;
ENTITY key_read IS
PORT (clk : IN std_logic;
      sw1 ,sw2 ,sw3, sw4 :IN std_logic;
      sw1_led ,sw2_led,sw3_led,sw4_led :BUFFER std_logic );
END ENTITY ;
ARCHITECTURE ked_dec OF key_read IS
signal key_rst :std_logic_vector(3 downto 0);
signal key_rst_an :std_logic_vector(3 downto 0);
signal key_rst_r :std_logic_vector(3 downto 0);
signal low_sw : std_logic_vector(3 downto 0);
signal low_sw_r :std_logic_vector(3 downto 0);
signal low_sw_an : std_logic_vector(3 downto 0);
signal cnt : std_logic_vector(19 downto 0);
BEGIN
PROCESS (clk)
BEGIN
    IF clk'event and clk = '1' THEN
        key_rst <= sw4&sw3&sw2&sw1 ;
```

```

        END IF ;
    END PROCESS ;
    PROCESS (clk)
    BEGIN
        IF clk'event and clk = '1' THEN
            key_rst_r <= key_rst ;
        END IF ;
    END PROCESS ;
    key_rst_an <= key_rst_r AND NOT key_rst ;
    PROCESS (clk )
    BEGIN
        IF clk'event and clk = '1' THEN
            IF key_rst_an = "0000" THEN
                cnt <= cnt + '1' ;
            ELSE cnt <= (OTHERS => '0') ;
            END IF ;
        END IF ;
    END PROCESS ;
    PROCESS (clk)
    BEGIN
        IF clk'event and clk = '1' THEN
            IF cnt = "11111111111111111111" THEN
                low_sw <= sw4&sw3&sw2&sw1 ;
            ELSE NULL ;
            END IF ;
        END IF ;
    END PROCESS ;
    PROCESS (clk )
    BEGIN
        IF clk'event and clk = '1' THEN
            low_sw_r <= low_sw ;
        END IF ;
    END PROCESS ;
    low_sw_an <= low_sw_r AND NOT low_sw ;
    PROCESS (clk)
    BEGIN
        IF low_sw_an(3)='1' THEN sw4_led <= NOT sw4_led ;
        ELSIF low_sw_an(2)='1' THEN sw3_led <= NOT sw3_led ;
        ELSIF low_sw_an(1)='1' THEN sw2_led <= NOT sw2_led ;
        ELSIF low_sw_an(0)='1' THEN sw1_led <= NOT sw1_led ;
        ELSE NULL ;
        END IF ;
    END PROCESS ;
    END ked_dec ;

```