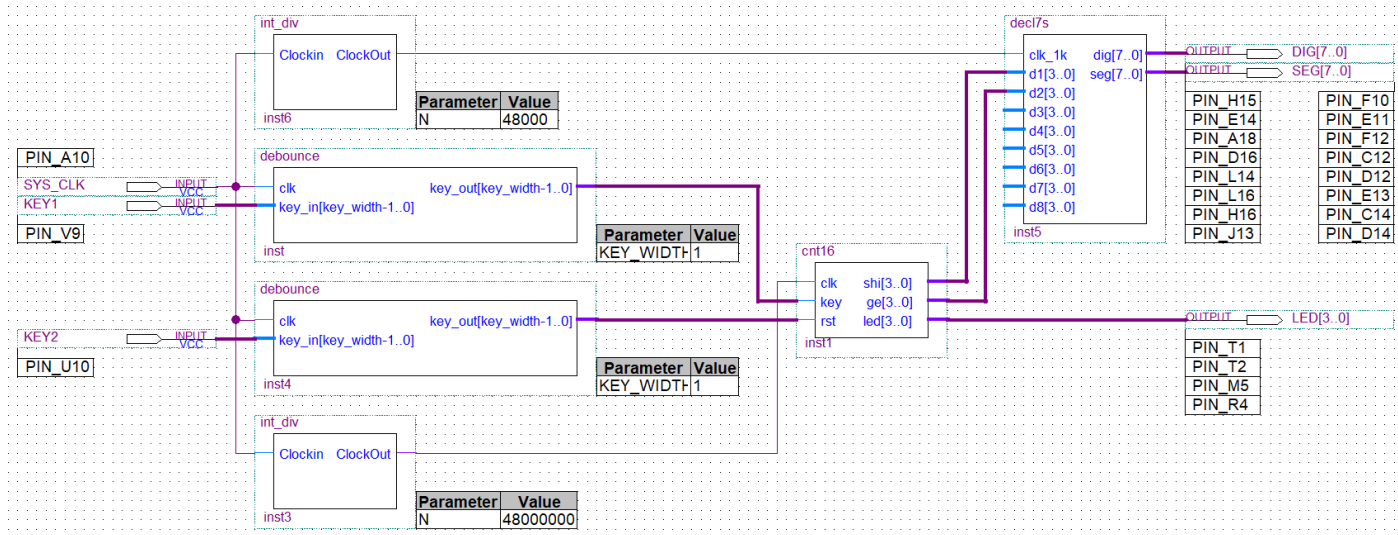


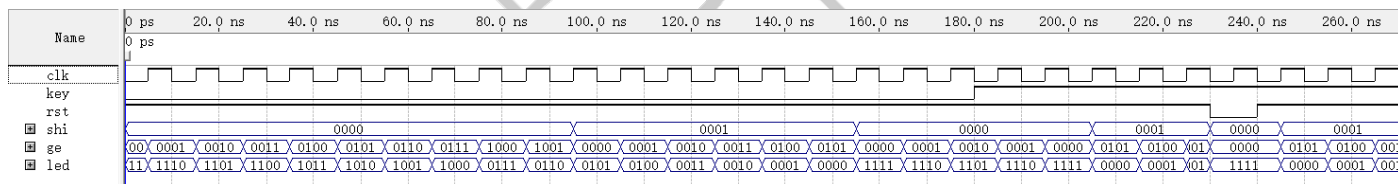
## 《可编程器件应用技术》预测题

### 预测一 设计一个模 16 计数器

要求：计数初值为 0，每过 1s 计数值进行一次加减，当按键 KEY1 向上拨时计数值加 1，当按键 KEY1 向下拨时计数值减 1。用 4 颗 LED 灯 (LED1-4) 表示计数的二进制（亮代表二进制“1”，灭代表二进制“0”，LED1 代表最高位，LED4 代表最低位），用 2 位七段数码管 (d1-2) 表示计数的十进制（d1 代表十位，d2 代表个位）。此外，每按下一次按键 KEY2，计数清零。



cnt16.vhd 模块仿真波形：



cnt16.vhd 模块代码[注意：代码中 LED3~0 对应管脚 LED0~3]：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY cnt16 IS
PORT(
    clk,key,rst    :    IN  STD_LOGIC;
    shi,ge         :    OUT STD_LOGIC_VECTOR(3 DOWNT0 0);
    led            :    OUT STD_LOGIC_VECTOR(3 DOWNT0 0)
);
END cnt16;
```

```
ARCHITECTURE behav OF cnt16 IS
    SIGNAL cs,cg : STD_LOGIC_VECTOR(3 DOWNT0 0);
BEGIN
    P1:PROCESS(clk,rst)
    BEGIN
        IF rst = '0' THEN
            cs <= X"0";
            cg <= X"0";
        ELSIF clk'EVENT AND clk = '1' THEN
            IF key = '0' THEN
                IF cg < X"9" THEN
                    cg <= cg + X"1";
                ELSE
                    cg <= X"0";
                END IF;
            ELSE
                IF cg > X"0" THEN
                    cg <= cg - X"1";
                ELSE
                    cg <= X"9";
                END IF;
            END IF;
        END IF;
    END PROCESS;
    shi <= cg(2 downto 0);
    ge  <= cg(3);
    led <= cs;
```

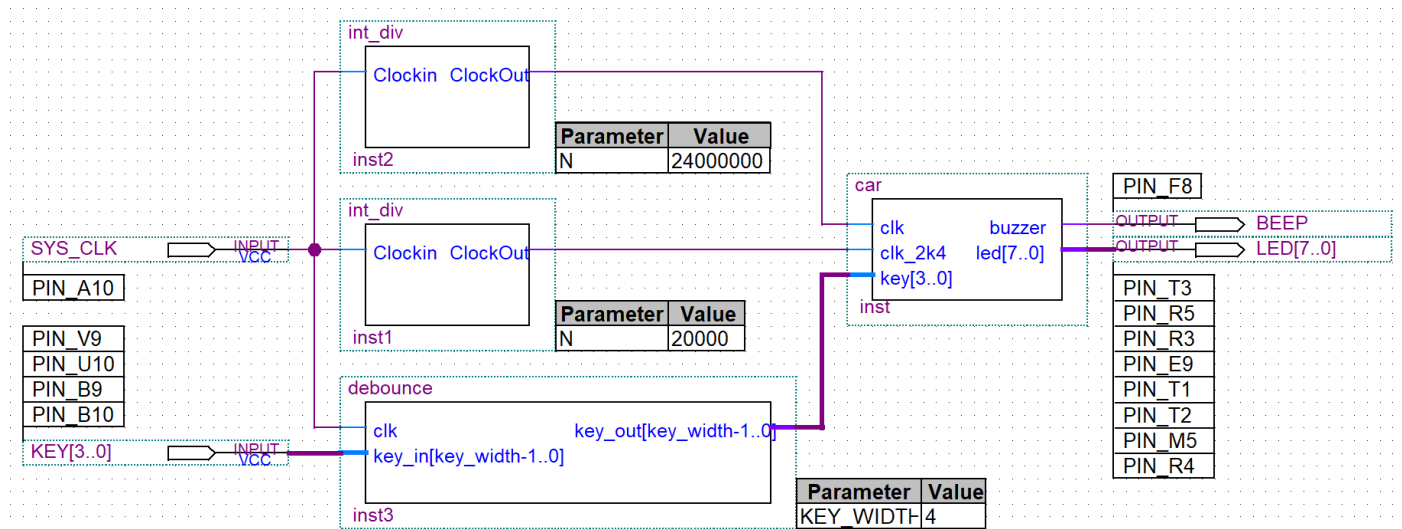
```

ELSE
    cs <= cs + X"1";
    cg <= X"0";
END IF;
IF cs = X"1" AND cg = X"5" THEN
    cs <= X"0";
    cg <= X"0";
END IF;
ELSIF key = '1' THEN
    IF cg > X"0" THEN
        cg <= cg - X"1";
    ELSE
        cs <= cs - X"1";
        cg <= X"9";
    END IF;
    IF cg = X"0" AND cg = X"0" THEN
        cs <= X"1";
        cg <= X"5";
    END IF;
END IF;
END IF;
END PROCESS;
P2:PROCESS(cs,cg)
    VARIABLE temp : STD_LOGIC_VECTOR(7 DOWNT0 0);
BEGIN
    temp := cs & cg;
    CASE temp IS
        WHEN X"15" => led <= NOT "1111";
        WHEN X"14" => led <= NOT "1110";
        WHEN X"13" => led <= NOT "1101";
        WHEN X"12" => led <= NOT "1100";
        WHEN X"11" => led <= NOT "1011";
        WHEN X"10" => led <= NOT "1010";
        WHEN X"09" => led <= NOT "1001";
        WHEN X"08" => led <= NOT "1000";
        WHEN X"07" => led <= NOT "0111";
        WHEN X"06" => led <= NOT "0110";
        WHEN X"05" => led <= NOT "0101";
        WHEN X"04" => led <= NOT "0100";
        WHEN X"03" => led <= NOT "0011";
        WHEN X"02" => led <= NOT "0010";
        WHEN X"01" => led <= NOT "0001";
        WHEN X"00" => led <= NOT "0000";
        WHEN OTHERS => led <= NOT "0000";
    END CASE;
END PROCESS;
shi <= cs;
ge <= cg;
END behav;

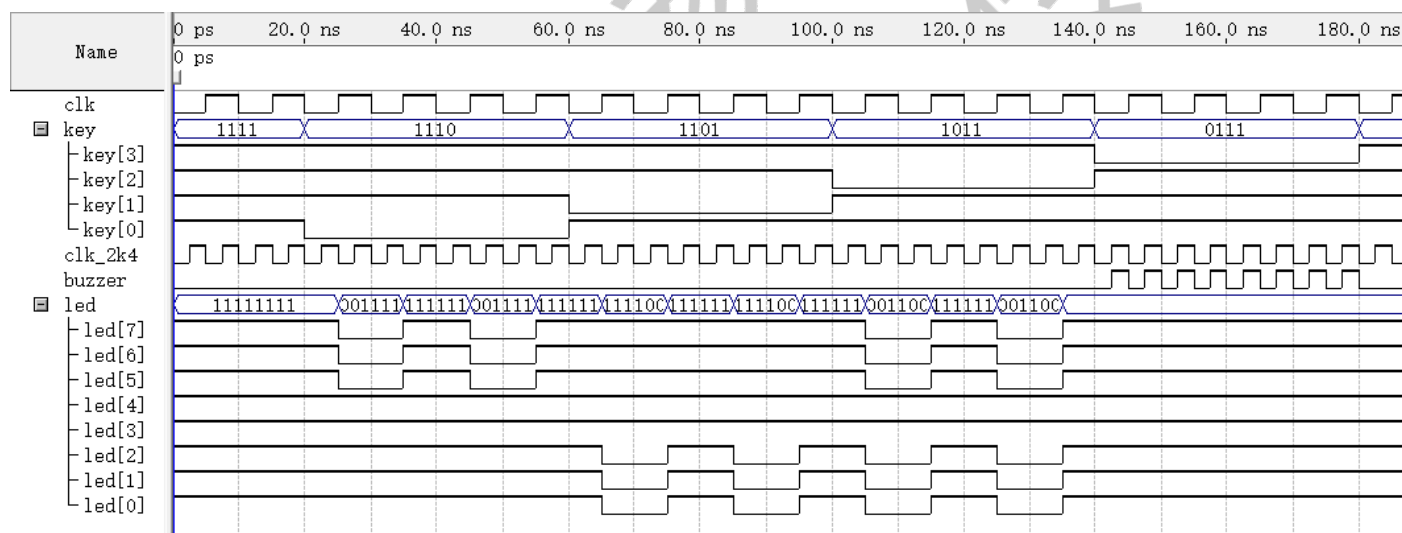
```

## 预测二 设计一个汽车尾灯

要求：按键 KEY1-3 全部向下拨代表正常行驶，不闪灯；按键 KEY1 向上拨代表左转，左侧 3 颗 LED 灯 (LED1-3) 闪；按键 KEY2 向上拨代表右转，右侧 3 颗 LED 灯 (LED6-8) 闪；按键 KEY3 向上拨代表双闪，左右两侧共 6 颗 LED 灯 (LED1-3、LED6-8) 闪；其中，双闪的优先级最高，闪烁的周期为 1s (0.5s 亮，0.5s 灭)。此外，按键 KEY4 按下代表按下喇叭，此时蜂鸣器发出频率为 2400Hz 的声音。



car.vhd 模块仿真波形：

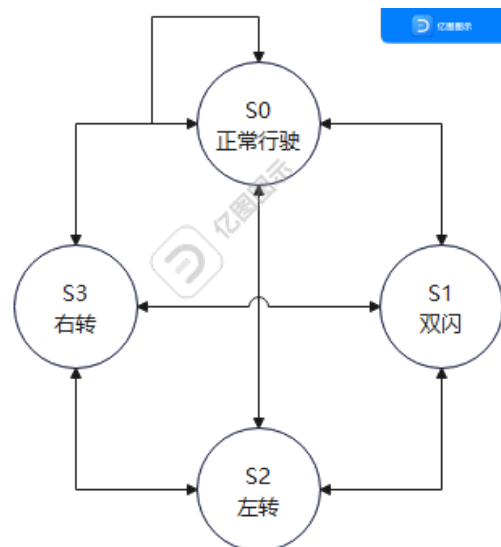


car.vhd 模块代码[注意：代码中 LED7~0 对应管脚 LED0~7]：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
ENTITY car IS
PORT(
    clk      : IN  STD_LOGIC;
    clk_2k4  : IN  STD_LOGIC;
    key      : IN  STD_LOGIC_VECTOR(3 DOWNTO 0);
    buzzer   : OUT STD_LOGIC;
    led      : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
);
END car;
```

```
ARCHITECTURE behav OF car IS
    TYPE state IS (s0,s1,s2,s3);
    SIGNAL current_state,next_state : state;
BEGIN
    P1:PROCESS(clk)
```



```

BEGIN
  IF clk'EVENT AND clk = '1' THEN
    current_state <= next_state;
  END IF;
END PROCESS;
P2:PROCESS(current_state,key)
BEGIN
  CASE current_state IS
    WHEN s0 => led <= "11111111";
      IF key(2) = '0' THEN
        next_state <= s1;
      ELSIF key(0) = '0' THEN
        next_state <= s2;
      ELSIF key(1) = '0' THEN
        next_state <= s3;
      ELSE
        next_state <= s0;
      END IF;
    WHEN s1 => led <= "00011000";
      IF key(2) = '0' THEN
        next_state <= s0;
      ELSIF key(0) = '0' THEN
        next_state <= s2;
      ELSIF key(1) = '0' THEN
        next_state <= s3;
      ELSE
        next_state <= s0;
      END IF;
    WHEN s2 => led <= "00011111";
      IF key(2) = '0' THEN
        next_state <= s1;
      ELSIF key(0) = '0' THEN
        next_state <= s0;
      ELSIF key(1) = '0' THEN
        next_state <= s3;
      ELSE
        next_state <= s0;
      END IF;
    WHEN s3 => led <= "11111000";
      IF key(2) = '0' THEN
        next_state <= s1;
      ELSIF key(0) = '0' THEN
        next_state <= s2;
      ELSIF key(1) = '0' THEN
        next_state <= s0;
      ELSE
        next_state <= s0;
      END IF;
  END CASE;
END PROCESS;
P3:PROCESS(clk_2k4,key)
BEGIN
  IF key(3) = '0' THEN
    buzzer <= clk_2k4;
  ELSE
    buzzer <= '0';
  END IF;
END PROCESS;
END behav;

```