

# 基于Verilog和FPGA的多功能秒表设计

刘书畅 518021910789

## 实验目的

1. 初步掌握利用Verilog硬件描述语言进行逻辑功能设计的原理和方法
2. 理解和掌握运用大规模可编程逻辑器件进行逻辑设计的原理和方法
3. 理解硬件实现方法中的并行性，联系软件实现方法中的并发性
4. 理解硬件和软件是相辅相成、并在设计和应用方法上的优势互补的特点
5. 本实验学习积累的Verilog硬件描述语言和对FPGA/CPLD的编程操作，是进行后续《计算机组成原理》部分课程实验，设计实现计算机逻辑的基础

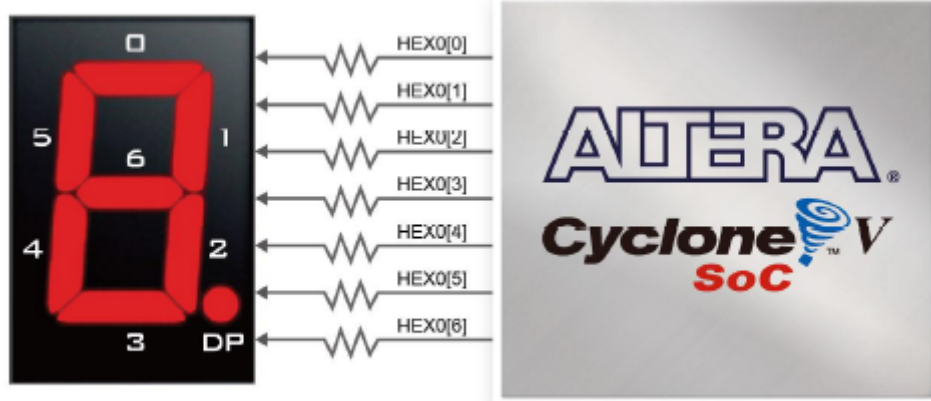
## 设计思路

由于这是本人第一次接触相关实验，所以采取先减后繁，逐步迭代的设计思路。先实现简单的控制数码管显示，并亮起相应数字，再实现时间计数功能。当完成基础功能后逐步加上按键功能，LED灯功能。最后增加按键消抖。

### 1. 数码管显示

一个数字由七根数码管组成，所以需要7个bit控制。使用case语句，进行不同数字到数码管亮暗的转换。1表示暗，0表示亮。

数码管的不同管的亮暗由不同bit控制



### 2. 时间计数功能设计

因为DE1-SOC板上的时钟为50MHz，所以需要500000次20ns之后，才是10ms。

而要求设计的秒表需要精确到10ms，所以我们使用变量 `counter_50M` 进行计数，每次时钟上升沿将计数器加一，当计数器的值为500000时，说明时间累积10ms，相应增加 `msecond_display_low` 的值。

为了数码管同时改变值，和为之后的暂停显示功能做准备，我们区分计数寄存器和显示寄存器，`xxx_counter_high/low`用于计数，`xxx_display_high/low`用于显示，若需要显示当前时间，则把计数值赋值给显示值。

不同时间位之间的进制不同。`msecond_counter_low`满10，`msecond_counter_high`加一；`msecond_counter_high`满10，`second_counter_low`加一；`second_counter_low`满10，`second_counter_high`加一；`second_counter_high`满6，`minute_counter_low`加一；`minute_counter_low`满10，`minute_counter_high`加一。

### 3. 按键功能设计

按照需求，我们需要设计三个按键，分别为“计时复位”、“计数/暂停”、“显示暂停/显示继续”。我们采用 `KEY0` 控制计时复位，用 `KEY1` 控制计数/暂停，用 `KEY2` 控制显示暂停/显示继续。

按键按下信号为0。

使用 `reset_time` 表示是否需要复位，若为1则将 `xxx_counter_high/low` 和计数器 `counter_50M` 均置为0，并赋值给 `xxx_display_high/low`。

使用 `start` 表示是否开始计时。只有 `start` 为1时才改变 `xxx_counter_high/low` 的值。

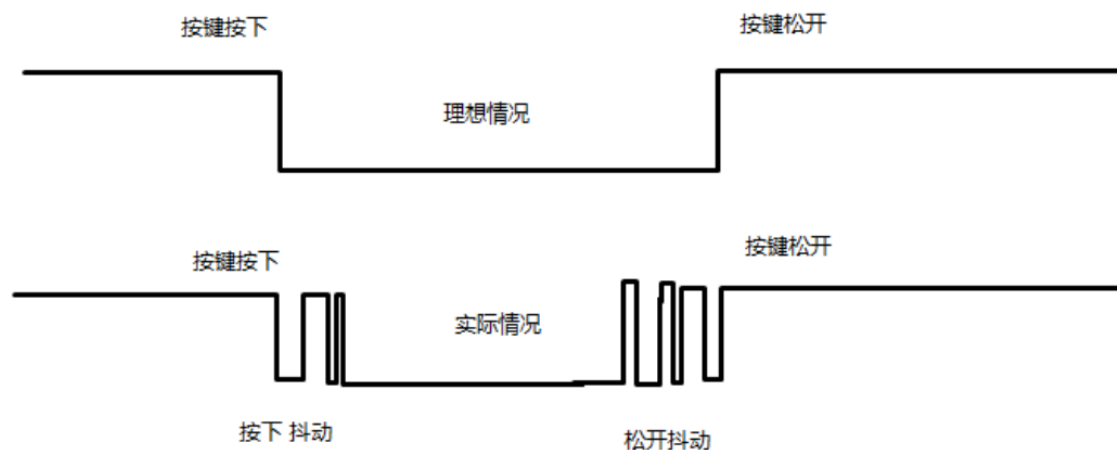
使用 `display` 表示是否开始显示。只有 `display` 的值为1时，才将 `xxx_counter_high/low` 赋值给 `xxx_display_high/low`。

### 4. LED灯设计

`LED0` 表示“计时复位”按键是否按下，`LED1` 表示“计数/暂停”按键是否按下，`LED2` 表示“显示暂停/显示继续”按键是否按下。按键按下LED灯为暗。

### 5. 按键消抖

按键按下的理想状态波形应该如图所示，但由于人手的不精确以及材料的形变回弹等等原因，往往不能达到理想情况。



按键消抖的总体思路为忽略抖动时的信号变化。

此处我进行按键消抖的思路为，当按键按下时开始计数，当持续有 8'b11111111 次，按键信号为按下时，表示按键真的按下，修改相应 `reset_time`, `start`, `display` 的值。当按键松开时开始计数，当持续有 8'b11111111 次，按键信号为松开时，表示按键真的松开。

由于 `reset` 操作较开始和显示简单，所以松开按键不做计数。

## 代码说明

一下对关键代码进行说明

### 1. 时间计数

每次时钟上升沿增加计数器 `counter_50M` 的值，当 `if(counter_50M == 500000)` 时，说明累积 10ms，逐级判断进位。

```
if(start)
begin
counter_50M = counter_50M + 1;
if(counter_50M == 500000)
begin
counter_50M = 0;
msecond_counter_low = msecond_counter_low + 1;
if(msecond_counter_low == 10)
begin
msecond_counter_high = msecond_counter_high + 1;
msecond_counter_low = 0;

if(msecond_counter_high == 10)
begin
msecond_counter_high = 0;
second_counter_low = second_counter_low + 1;

if(second_counter_low == 10)
begin
second_counter_low = 0;
second_counter_high = second_counter_high + 1;

if(second_counter_high == 6)
begin
minute_counter_low = minute_counter_low + 1;
second_counter_high = 0;

if(minute_counter_low == 10)
begin
minute_counter_low = 0;
minute_counter_high = minute_counter_high + 1;

if(minute_counter_high == 6)
minute_counter_high = 0;
end
end
end
end
end
end
end
```

```
end
```

## 2. 按键消抖

此处以“计数/暂停”的按键消抖为例。

当按键完成按下并松开这一动作时，才视为一次成功按键操作，转换 `start` 状态。

`start_time` 变量用于按键消抖状态转换，`start_time` 值为1时进行按下计数，为0时进行松开计数。

```
always @(posedge CLOCK_50)
begin
    if (start_time && !key_start_pause)
    begin
        counter_start = counter_start + 1;
        if (counter_start == 8'b11111111)
        begin
            counter_start = 0;
            start_time = ~start_time;
        end
    end else if (!start_time && key_start_pause)
    begin
        counter_start = counter_start + 1;
        if (counter_start == 8'b11111111)
        begin
            counter_start = 0;
            start_time = ~start_time;

            start = !start;
        end
    end else
    begin
        counter_start = 0;
    end
end
```

## 3. 时序设置

```
create_clock -name "CLOCK_50" -period 20.000ns [get_ports {CLOCK_50}]
derive_clock_uncertainty
```

## 4. 引脚分配

完成代码编写后需要将对应原件映射到对应的引脚上，根据《DE1-SoC User Manual 》

LED灯的映射如下：

Table 3-8 Pin Assignment of LEDs

<i>Signal Name</i>	<i>FPGA Pin No.</i>	<i>Description</i>	<i>I/O Standard</i>
LEDR[0]	PIN_V16	LED [0]	3.3V
LEDR[1]	PIN_W16	LED [1]	3.3V
LEDR[2]	PIN_V17	LED [2]	3.3V
LEDR[3]	PIN_V18	LED [3]	3.3V
LEDR[4]	PIN_W17	LED [4]	3.3V
LEDR[5]	PIN_W19	LED [5]	3.3V
LEDR[6]	PIN_Y19	LED [6]	3.3V
LEDR[7]	PIN_W20	LED [7]	3.3V
LEDR[8]	PIN_W21	LED [8]	3.3V
LEDR[9]	PIN_Y21	LED [9]	3.3V

数码管的映射如下：

Table 3-9 Pin Assignment of 7-segment Displays

<i>Signal Name</i>	<i>FPGA Pin No.</i>	<i>Description</i>	<i>I/O Standard</i>
HEX0[0]	PIN_AE26	Seven Segment Digit 0[0]	3.3V
HEX0[1]	PIN_AE27	Seven Segment Digit 0[1]	3.3V
HEX0[2]	PIN_AE28	Seven Segment Digit 0[2]	3.3V
HEX0[3]	PIN_AG27	Seven Segment Digit 0[3]	3.3V
HEX0[4]	PIN_AF28	Seven Segment Digit 0[4]	3.3V
HEX0[5]	PIN_AG28	Seven Segment Digit 0[5]	3.3V
HEX0[6]	PIN_AH28	Seven Segment Digit 0[6]	3.3V
HEX1[0]	PIN_AJ29	Seven Segment Digit 1[0]	3.3V
HEX1[1]	PIN_AH29	Seven Segment Digit 1[1]	3.3V
HEX1[2]	PIN_AH30	Seven Segment Digit 1[2]	3.3V
HEX1[3]	PIN_AG30	Seven Segment Digit 1[3]	3.3V
HEX1[4]	PIN_AF29	Seven Segment Digit 1[4]	3.3V
HEX1[5]	PIN_AF30	Seven Segment Digit 1[5]	3.3V
HEX1[6]	PIN_AD27	Seven Segment Digit 1[6]	3.3V
HEX2[0]	PIN_AB23	Seven Segment Digit 2[0]	3.3V
HEX2[1]	PIN_AE29	Seven Segment Digit 2[1]	3.3V
HEX2[2]	PIN_AD29	Seven Segment Digit 2[2]	3.3V
HEX2[3]	PIN_AC28	Seven Segment Digit 2[3]	3.3V
HEX2[4]	PIN_AD30	Seven Segment Digit 2[4]	3.3V
HEX2[5]	PIN_AC29	Seven Segment Digit 2[5]	3.3V
HEX2[6]	PIN_AC30	Seven Segment Digit 2[6]	3.3V
HEX3[0]	PIN_AD26	Seven Segment Digit 3[0]	3.3V
HEX3[1]	PIN_AC27	Seven Segment Digit 3[1]	3.3V
HEX3[2]	PIN_AD25	Seven Segment Digit 3[2]	3.3V
HEX3[3]	PIN_AC25	Seven Segment Digit 3[3]	3.3V
HEX3[4]	PIN_AB28	Seven Segment Digit 3[4]	3.3V
HEX3[5]	PIN_AB25	Seven Segment Digit 3[5]	3.3V
HEX3[6]	PIN_AB22	Seven Segment Digit 3[6]	3.3V
HEX4[0]	PIN_AA24	Seven Segment Digit 4[0]	3.3V
HEX4[1]	PIN_Y23	Seven Segment Digit 4[1]	3.3V
HEX4[2]	PIN_Y24	Seven Segment Digit 4[2]	3.3V
HEX4[3]	PIN_W22	Seven Segment Digit 4[3]	3.3V
HEX4[4]	PIN_W24	Seven Segment Digit 4[4]	3.3V
HEX4[5]	PIN_V23	Seven Segment Digit 4[5]	3.3V
HEX4[6]	PIN_W25	Seven Segment Digit 4[6]	3.3V
HEX5[0]	PIN_V25	Seven Segment Digit 5[0]	3.3V
HEX5[1]	PIN_AA28	Seven Segment Digit 5[1]	3.3V
HEX5[2]	PIN_Y27	Seven Segment Digit 5[2]	3.3V
HEX5[3]	PIN_AB27	Seven Segment Digit 5[3]	3.3V
HEX5[4]	PIN_AB26	Seven Segment Digit 5[4]	3.3V
HEX5[5]	PIN_AA26	Seven Segment Digit 5[5]	3.3V
HEX5[6]	PIN_AA25	Seven Segment Digit 5[6]	3.3V

按键的映射如下：

**Table 3-7 Pin Assignment of Push-buttons**

<i>Signal Name</i>	<i>FPGA Pin No.</i>	<i>Description</i>	<i>I/O Standard</i>
KEY[0]	PIN_AA14	Push-button[0]	3.3V
KEY[1]	PIN_AA15	Push-button[1]	3.3V
KEY[2]	PIN_W15	Push-button[2]	3.3V
KEY[3]	PIN_Y16	Push-button[3]	3.3V

**参考文献:**

1. 《软件学院《数字系统设计》课程实验指导书（2018级）交大》
2. 《DE1-SoC User Manual 》
3. 《My\_First\_Fpga》