

实验 3 报告

学号：2016K8009929011

姓名：段江飞

一、实验任务（10%）

本实验要求实现一个电子表，电子表初始是 23:59:55，具有设置和功能，能够在计时模式设置模式转换，在设置模式的时候，能够设置时分秒。

二、实验设计（30%）

（一）软件实现

1、计时功能

首先初始化寄存器，利用 t0-t5 分别表示电子表的时分秒的个位和十位，然后设置 count 和 compare 寄存器，compare 设置为 1s 对应的时钟周期数，是 2500 万，开中断，开中断的时候把 ERL 清 0，IE 置 1。当 count 和 compare 相等时会进入时钟中断，在时钟中断中，先给时钟加 1s，然后不修改 count 寄存器，给 compare 寄存器加上一秒对应的时钟周期数，然后返回。

2、进位和显示功能

进位设置成一个单独的函数，有 add_1s, add_1m, add_1h 三个函数入口，分别对应加 1 秒，加 1 分，加 1 小时功能，完成进位之后，会进入 show_time 函数，将 t0-t5 组合起来，存入 numdata 中，在数码管还是哪个显示。

3、设置功能

首先，在 soc 代码中将 confreg 的 btn_key_r[0] 和 int_n_i[4] 连起来，当按下矩阵键盘上 0 号按键时会进入中断，通过 cause 寄存器判断是时钟中断还是硬件中断，若是硬件中断则进入设置模式，在设置模式中，对按键进行轮询，首先轮询是否又按了一次设置键，若是，则退出设置模式；否则继续轮询矩阵键盘上的 1,2,3 号按键，分别对应加 1s, 1m 和 1h，在 4 个按键上不断轮询，若按了加 1 键，加 1 过后，则从设置键开始重新轮询。

连续按键功能实现：在查询到按某个按键的时候，经 count 寄存器和设置的五分之一秒对应的时钟周期比较，如果按下的时间超过该值，对按键对应的时分秒进行加 1，然后清空 count，继续判断按键按的时间。

退出设置模式时，要清除中断位，然后重置 count 和 compare 寄存器，eret 返回计时。

（二）硬件实现

在龙芯的 ls132r_top 中声明一个 wire 类型位宽 6 的 int_n_i 变量，和 interface 的 int_n_i 连接，在 soc_lite_top 中连接按键和 int_n_i。

```
assign cpu.int_n_i = {1'b1, ~confreg.btn_key_r[0], 4'b1111};
```

（三）验证

1、仿真验证

在 vivado 上仿真，将 1s 对应的时钟周期调为 1000，然后修改 testbench，强制按键，进行测试。

2、上板验证

数码管初始为 23:59:55，然后观察进位满足，在计时过程中按设置键和连续加 1 也都满足，通过上板验证。

三、实验过程（60%）

（一）实验流水账

1、10 月 3 日晚上 11 点到凌晨 1 点 40

进行试验设计，写了进位和显示函数。

2、10 月 4 日上午 10 点到晚上 9 点

完成试验，解决 bug，并进行上板验证。

（二）错误记录

1、错误 1

（1）错误现象

Count 一直累加，但是不进入计时的加 1s，数码管的数值不变。

（2）分析定位过程

通过看波形图发现 count 和 compare 相等的时候，没有进入时钟中断，然后把 status 和 cause 寄存器拿过来看，发现 status 的 ERL 位为 1，是关中断状态。

（3）错误原因

CPU 在复位的时候，会将 ERL 为置为 1，我没有注意到这个，需要将 ERL 位置 0，IE 置为 1 才是开中断。

（4）修正效果

程序开始正确执行，然后开始进行漫长的上板测试，之后又添加连续按键功能。

2、错误 2

（1）错误现象

精确计时问题。

（2）分析定位过程

Count 寄存器每次清 0，计时存在累积误差。

（3）错误原因

每次计时不需对 count 要清 0，只需要修改 compare 寄存器，每次给 compare 寄存器加上 1s 对应的时钟周期数，即便溢出了，修改 compare 也清楚了中断位，接下来，count 会加到溢出之后和 compare 相等才触发中断。

（4）修正效果

上板测试通过，能连续按键。

四、实验总结

上板测试好慢，感觉查询和数拍数很难实现精确计时。

国科大B62009H计算机体系结构研讨课17-18秋季