

实验 1 报告

第 24 小组
孔静、谈清扬、张传奇

一、实验目的

学习 MIPS 指令，掌握简单的 MIPS 汇编程序编写；
熟悉并掌握交叉编译工具链的使用；
熟悉并掌握 ISE 系列工具的使用；
熟悉并掌握电路设计仿真的方法，并且能够进行初步的调试；
熟悉并掌握 FPGA 开发的流程；

二、实验任务

（一）设计

1、汇编代码设计

（1）工作内容

编写汇编程序，使之能在提供的 SoC 平台上运行，控制 8 个数码管来，显示 24 进制的数字时钟。

（2）工作目标

设计出能正确记录时间并显示的数字钟，要求显示时、分、秒、百分秒，一共 8 位，同时需要足够精确，至少一分钟累计误差不超过 1 秒。

（二）实现

首先在 start.s 将时钟信号记录，并存储，然后转换为数码管要显示出来的数字。我们简单粗暴的将显示的数字钟（时、分、秒、百分秒）共八位分别存储在一个块内存中，因为数据不多，并没有写循环，而是对每一个进位情况都分别处理。

（三）验证

在 FPGA 板上进行测试，并根据钟表时间确认一分钟的误差。

三、实验设计

（一）设计方案

1、总体设计思路

在开始先按照时间精度进行划分。认定一段程序应该运行在 10ms，也就是百分秒，时钟显示的最小精度。显示出此时相应的 8 个数字，就能完成数字钟的功能。然后对这 10ms 进行拆分，拆成 10 个 1ms，前面 8ms 每个 1ms 用来显示数字，最后的 2ms 用来进行运算得出下 10ms 应该显示的数字。

显示的数字对应的各个数码管的控制是一个 8 位的数据，对应到每个数码管中，预先将这些写好存放在代码的 data 段之中，此处每个数字如果在每一级的个位的话，需要多显示出 1 个点，所以一共用了 20 byte 来存放。同时在内存中再预留好空间为这 8 位数字存储。

开始设计 10 次循环的控制条件，这里我打算借用片选控制的变量来进行循环的控制。从 0x1 开始，每次左移，一直移到 0x400，正好是循环的次数。

然后开始设计 1ms 时长的程序，每次循环开始。读取内存中存好的数字，以这个作为偏移量，在 dsram 内找到我们要的显示数据，加载到 confreg 模块的对应位置(cr02)中，然后将片选信号也写入 confreg 模块的对应位置(cr01)中去，然后等待到 1ms。这里不必担心循环到 9 和 10 次的时候写入 cr02 中的未初始化数据，因为此时片选为 0xff，不会有任何显示。

在循环第 10 次的时候，开始将记录时间+10ms 这里因为我们采用了 8 个数字单独存储的策略，所以从百分秒位开始，直接+1。然后开始判断进位，如果进位则清零，并用寄存器记录进位信息。然后开始计算十分秒位，需要注意的是分和秒的十位数，需要 6 的时候就算进位。另一个是在小时的个位数和十位数需要注意，当个位为 4 的时候做标记，当十位数为 2 的时候检测标记，如果符合则两位都清零。如此运算，就能够得到下一个百分秒之后的时间数字。

现在需要记录 1ms 的方式，在每次循环开始的时候将 c0_count 寄存器清零，然后在每次执行完正常的程序之后，进入等待循环，直到等到 c0_count 寄存器超过记录好数值，再重新开始下段程序。

（二）验证方案

1、总体验证思路

在 ISE 的仿真环境下，进行功能的验证，即要求能够正常输出片选信号和

控制信号，预期能正常显示即可。

在 FPGA 环境下，进行精度的确认，即验证我们的设计是否足够精确。

2、验证环境

仿真验证中需要我们利用 godson system 这个预先配置好的模块来进行测试，模拟了 flash 模块，和对整个系统的复位信号等。我们需要监测仿真的 NUM_CS_n 还有 NUM_A_G 信号。这就对应了我们设计中的写入显示部分，也同时可以检查是否有错误的输出。这两个信号能够确实地反映到 confreg 中。同时要监测每次循环的时长，确认大约是 1ms，这是检测我们的等待 ms 循环正确。

FPGA 系统验证环节，直接由板上的复位信号开始，MIPS 核开始执行程序。我们需要直接检查数码管能否正常显示，对应之前的加载部分，精度是否符合要求，对应设计的计时等待部分。

3、验证计划

表 1. 验证计划表

编号	功能点描述	考核标准
1	正确显示	显示正常，顺序无误
2	计时误差小	每分钟小于 1s 的误差

四、实验实现

（一）实现交付说明

修改了 start.s 文件，放到 lab_1\simu\soft\func_led 目录下

（二）实现说明

start.s 文件就对应了之前说的所有设计模块。

五、实验测试

（一）测试过程

1、仿真环节

（1）问题

首先发现有些循环的时间过短，基本只是执行了基本的处理，而且每

2 次循环会发生一次这种现象。

解决：

检查在等待循环中到将 `slt` 这种判断条件的语句写到了 `beq` 分支延迟槽中，这样在信号寄存器未清零的情况下，会在第二次的时候直接退出等待循环。

(2) 问题

输出的 `NUM_A_G` 信号不正常。

解决：

自己在编写程序的时候没有处理好几个数据的关系，应当是读出对应数字之后，再按照对应数字去数据段寻找对应数码管控制数据，结果直接存储了对应数字。

2、FPGA 测试环节

(3) 问题

数字显示顺序反了。

解决：

反方向存储内存，先存高位的数据。另一种解决方法没有成功：将游标设成 7，每次-1，直到 0，这样读取，结果不能正常工作，原因不明。

(4) 问题

精度不够。

解决：

在几次测量之后，发现每 1 分钟会慢了 6 秒上下，误差很大，我此时感觉到 $\frac{1}{10}$ 这个比例很特殊。相当于程序每 10ms 的时候多等待了 1ms，因此我开始检测我的循环条件，发现一开始对循环次数的理解有问题，程序循环 10 次，从 0x1 开始到 0x800 但是实际上，最后一次不跳转也会执行下去，所以应该是 0x400.

(二) 测试结果

1、仿真测试

在修改过数据标号和跳转错误之后，仿真能符合验证标准

2、FPGA 测试

在修改过数据顺序之后，正常显示。修改循环次数之后，一分钟误差至多

5 个十分秒。

六、成员分工

(一) 张传奇

设计部分和调试验证部分和报告部分都参与了，约 6 个小时。

(二) 谈清扬

设计代码，以及调试验证，约 4 个小时。

(三) 孔静

设计部分，围观调试，和修饰报告，约 4 个小时。

七、实验总结

(一) 张传奇

熟悉了一遍 MIPS 下的汇编程序编写。同时感觉确实上板验证是最准确的验证，一些错误是没法用仿真发现的。

(二) 谈清扬

之前写程序都可以用现成的调试器或者集成开发环境监察结果，这一次的实验相当于需要通过查看仿真过程中波形的变化来间接检查程序的正确性，复杂度陡升，感觉心累/(T o T)/~~。了解到编写底层程序的复杂性，需要了解各种关于移动代码，地址标号变化的问题。

(三) 孔静

之前没怎么写过 MIPS 汇编，对代码指令之类的都不是很熟悉，这次边查边写，学习了一下 MIPS 汇编。另一方面，之前写 Verilog，没有用 FPGA 跑过，相对理想化，所以遇到了一些问题，在队友以及其他同学的帮助下解决了。还是要多加练习，才能熟练，o(n_n)o 笑。

八、参考文献

无。