实验 1 报告

第 9 小组
康楠，何博，唐铎月

一、实验目的

① 如何通过软件编程，实现计时这个功能？

②“当前时间”在内存&寄存器中要如何存放，在过了单位时间后，如何更新这个时间？

③ 如何将“当前时间”翻译为时钟上的 8 个数字？时钟刷新要在什么时刻进行？

二、实验任务

设计：

① 保存时间：设置一个计数器，每过百分秒加一，表示当前时间。

② 显示信号：将计数器里的数字”翻译“为对应数码管上的 8 个数字，在将对应数字”翻译“为控制数码管上 ABCDEFG 和 DP 开关的信号。

③ 精确计时：使用循环，精确控制一段代码在特定的时间内完成(比如 1ms，2ms)

实现：

整体时序：在实现了 1），2），3)功能之后，整个程序的时序就明了了。首先将计数器置为 0，然后进入一个名为 LOOP 的循环(每个循环代表一个百分秒)。循环开始为将计数器的数值翻译为控制信号，这个过程利用精确计时控制为 2ms。之后对于 8 个数码管，每个数码管在 1ms 内，将控制信号存入相应内存地址，8 个数码管恰好使用 8ms 的时间。10ms 过后，数字时钟更新，计数器加一，从 LOOP 循环继续循环执行。

验证：

模拟产生波形以后,就可以通过查看 CSn 和 NUM_A_G 的值以及时间刻度来验证计时是否正确，准确。如果仿真通过，就可以上板验证了。

如果在 FPGA 上计时成功，就可以通过手机等设备计时验证时间显示的精确程度。要求每分钟误差不超过一秒。

三、实验设计

流程图如下：

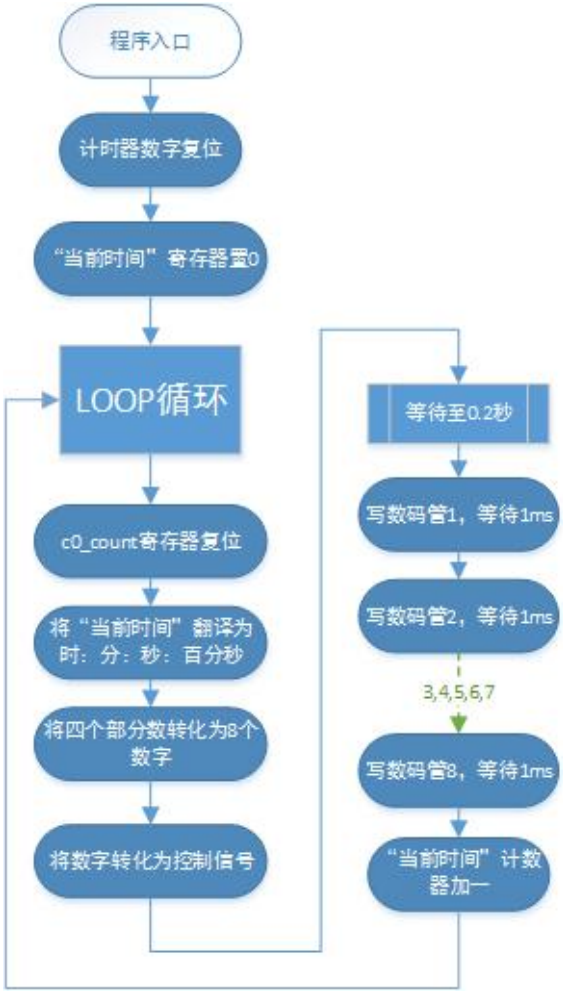

图 1 流程图

（一）设计方案

1、总体设计思路

（1）保存当前时间

当前时间(计数器)由一个特定的寄存器保存。我们的小时采用了 24 进制，因此每次计数器加 1，都需要模 864000 (24 小时)，使得它能循环计数。当然，也可以用 slt+bnez 实现。

（2）当前时间转化为控制信号

由于计数器保存的数字单位为百分秒，首先，需要用过取模求得各部分的数字 (时:分:秒:百分秒),之后，将四部分的数细化为 8 个数字。比如，假设计数器存了数字 3629816, 首先模 100×60×60,结果为 **10**，余数为 29816。再模 100×60，结果为 **4**，余数为 5816。再模 100，结果为 58，余 16。所以结果就是 10 时 04 分 58 秒 16。之后，对每个部分模 10，取余数和商，就得到数码管上需要显示的 8 个数字，从

左到右分别是：1 0 0 4 5 8 1 6，分别将 8 个数字存入 **t0-t7** 8 个寄存器中。（实际上存入一个寄存器就已足够，但考虑到实现方式简洁，便使用了 8 个寄存器）。之后，对每个数字使用分支语句，将对应的控制信号写回 **t0-t7**。

（3）精确计时

上述部分实现后，需要等待到 2ms,这时使用计时功能。首先 mfc0 v0,c0_count,之后如果 v0 小于一个特定值就一直循环，直到达到了这个特定值才跳出，这样就实现了计时功能。现在我们要实现 2ms 的计时，那么这个特定值就设为 33000。注：c0_count 在 LOOP 循环起始位置就已置位。之后的计时功能实现同理。

（4）刷新数码管

剩余 8ms 的时间内，每个毫秒我们只需做两件事，第一件事就是将对应寄存器的控制信号值写入内存，之后就是使用 3)的方法循环等待直到 1ms 结束。

（二）验证方案

1、总体验证思路

仿真验证：验证 CPU 是否能够正常运转，CSn 以及 NUM_A_G 端口显示的数字是否符合预期，计时是否精确。

FPGA 系统验证：数字显示是否正确，时钟显示是否正常，对比手机，手表等验证计时是否有偏差。

2、验证环境

所需工具：FPGA 实验设备，手表或手机等对照计时设备。

3、验证计划

仿真验证：

编号	功能点描述	考核标准
1	CPU 指令正常执行	PC 寄存器能有序更新
2	CSn，NUM_A_G 端口符合预期	在特定时间内他们能做出相应计时反应
3	计时精确	每次循环所用时间恰好为 10ms

FPGA 系统验证:

编号	功能点描述	考核标准
1	数字显示正确	数码管显示 0-9 的数字，而不是其他图案
2	数字进位正确	100 个百分秒进一秒，60 秒进 1 分钟等
3	计时准确	计时足够长时间，如 10 分钟，平均每分钟误差不超过 1 秒

四、实验实现

（一）实现交付说明

附件 start.s

（二）实现说明

代码中有若干标号：

LOOP: 每次计时 10ms 的整个过程。

GET_HOUR: 求得小时对应两个数码管的对应**数字**

GET_MINUTE，同上

GET_SECOND，同上

GET_10MS，同上

CHANGE_t：将 GET_** 求得数字转化为相应控制信号。

SWITCH:转化，由 CHANGE_t 调用

time2ms,time3ms,...,time10ms: 在各个时间点精确计时。

五、实验测试

（一）测试过程

仿真测试：

① CPU 在执行一段程序后突然 PC 卡在一个位置再也不继续执行。

解决方法：sw 的地址设置不对，更改后便得以解决。

② 时间表示不对，如 12 秒，应该显示 1 2，但实际显示 2 1.

解决方法：在对两位数，如 12 除以 10 后，mfhi，mflo 的位置用反了，导致个位十位数字的显示交换位置。

③ 显示时间时一直只显示最低两位：00:00:00:xx.

解决方法：当前时间保存在 a2 寄存器内，在分析时：分：秒：百分秒的过程中对 a2 做了修改，最终 a2 只保存了百分秒的数字，因此最终 a2 的数值为百分秒数。因此，在对 a2 做除法之前，将 a2 保存在 t8 寄存器内，在循环末尾恢复。

FPGA 系统验证：

很顺利，一次通过。

（二）测试结果

仿真验证：PC 寄存器正常更新，数字正常更新，对 1ms 的计时做到了比较高的精度。

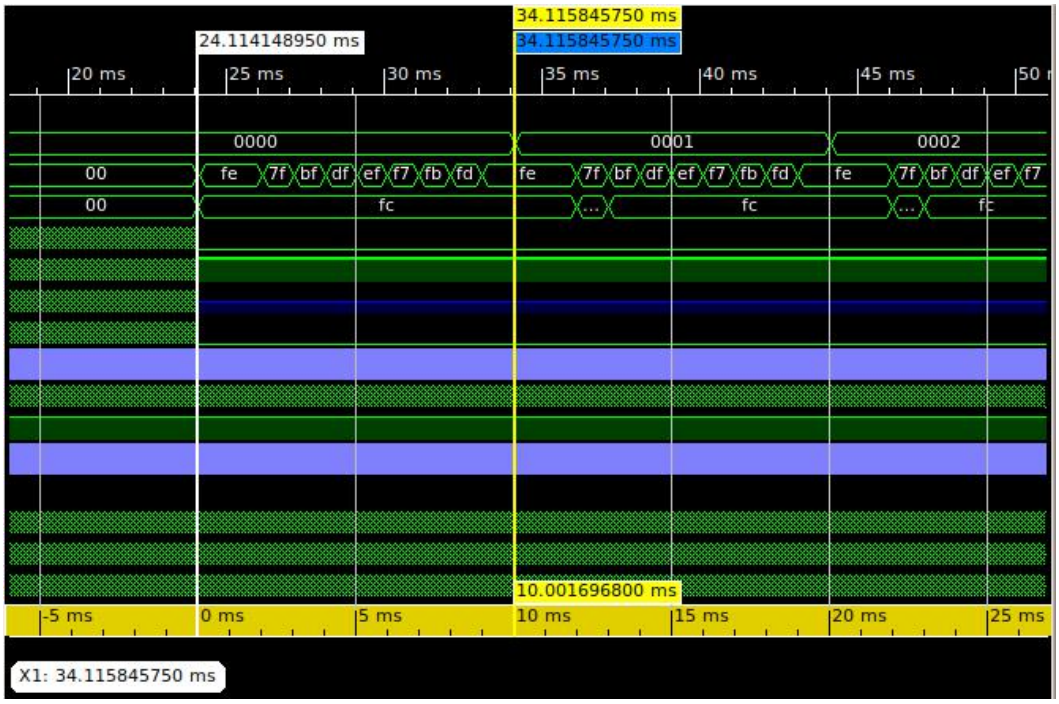

图 2 十毫秒计时

FPGA 系统验证：数字正常显示，正常进位。

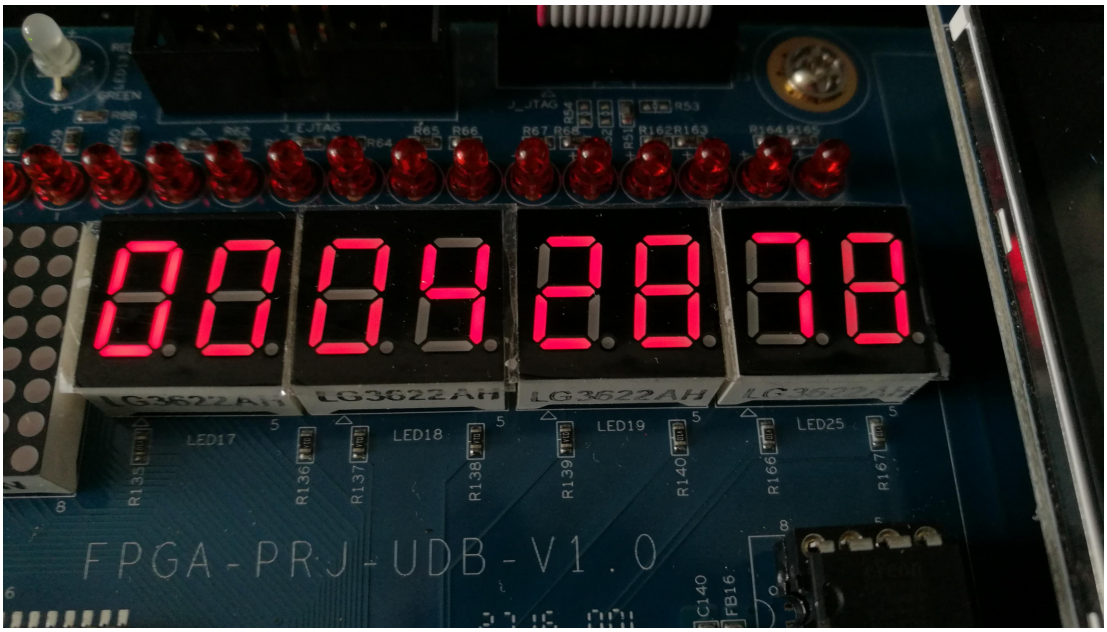

图 3 数字显示正常

下图为与手机计时的对比图（注：手机在 1 分钟时开始计时）

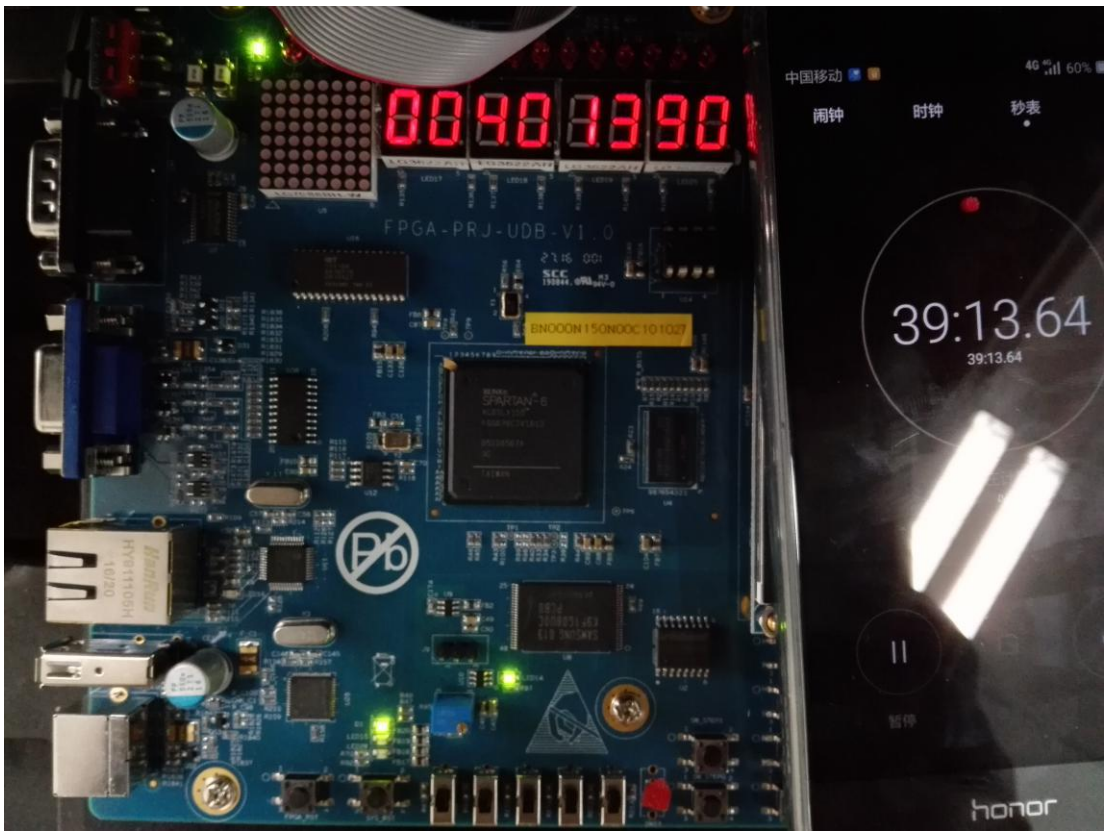

图 4 计时准确

六、成员分工

康楠：代码框架，时间控制及控制信号的转换、写入，FPGA 仿真，实验报告。

何博：设计的整体思路，时序，以及实现精确计时功能，FPGA 仿真。

唐铎月：

七、实验总结

（一）组员：康楠

由于这是第一次使用 MIPS 写汇编，与以前 X86 环境的汇编有很大不同，因此几乎花费了好几个小时的时间边写代码边摸索各个指令的使用方法，也犯过很多错误，比如 move 指令的寄存器顺序弄反、直接写 **move t0,0x1234** 之类的错误表达、立即数写了 32 位、使用 li 指令加载地址、使用 beq a0，a0,1f 来实现无条件跳转(不知道还有 b 指令存在...)、总是忘了考虑延迟槽、没有考虑虚实地址转换。此外，对 ISE 的使用过程也不太熟悉。

不过，最后还是学到了很多东西。比如学会了 div，mflo，mfhi 等指令，li,la 等宏

指令与 MIPS 指令的区别，ise 波形的查看，FPGA 系统验证的流程，wcfg 文件的加载。万事开头难，经过这次的实验，我对 MIPS 的编程有了一定的了解，相信对之后的任务会有很大帮助。

（二）组员：何博

因为这是第一次实验，在实验的过程中，由于对于 MIPS 指令不太熟悉，所以在实验过程中，要一边查询指令的格式和用法，一边写代码，效率较为低下。而且，当代码写完，调试的过程中，发现了很多 bug，要通过 ISE 的仿真波形信号来判断，这个过程也是比较耗时间和精力的。甚至调试 bug 的时间比写代码的时间还多。不过，通过这个实验，我对于此实验有了更加深刻的理解，对于 MIPS 汇编也逐渐熟悉了起来，特别是对于调试 bug 的过程有了更加深刻的印象，因为绝大多数的 bug 都不是设计结构上的缺陷，而是一些很小的错误，但是却不容易发现。希望以后的实验中，能够更加高效地调试 bug，减少调试 bug 的时间。

（三）组员：唐铎月

八、参考文献

无。