微机原理课程总结-宁毓伟

宁毓伟 | 今天创建

微机原理对于我来说是一门比较陌生的课程，因为我原来就读的计算机学院对硬件课程部分进行了大量的删减，去掉了模拟电子电路课程，同时将数字逻辑电路课程的重心放在了理论部分，并且没有安排硬件课程设计。计算机组成原理实验也仅仅采用模拟器完成，并且着眼于CPU的设计和构造，省略了很多更加底层的问题。

微机原理刚开始的时候，我就感受到了我在硬件知识方面的欠缺。还记得课程刚开始的时候，我当时对于面包版的内部连线构造还没有任何了解，对于如何通过程序控制使单片机在特定引脚输出高电平并使二极管亮起来也没有非常清晰的思路。但是幸好我的队友以及种子班内的同学们都十分热心，愿意花时间给我这个硬件小白讲述硬件基本知识。在他们的帮助下，我逐渐对硬件开发有了一定的了解：单片机内存在各种控制寄存器，用于控制各个功能模块，只需要向控制寄存器中相应的bit写入特定的值，我们就可以让其对应的模块完成相应的功能。同时单片机中还存在一些通用寄存器，用于存储变量的值。这些硬件开发的知识在我现在看来是十分平常的，但对于十天之前的我却是闻所未闻的知识点。

当我掌握了硬件开发的基本知识后，我曾有些自大的认为微机原理已经难不倒我了，但是实验过程中却还是遇到了许多问题。在第一个实验：闪灯实验的过程中，我就立即碰到了一个问题——我在剪切板中保存的程序代码无法通过简单的点击右键复制到putty中并下载到单片机上。但是在Linux版本的putty中直接使用右键并不能将程序复制上去，经过使用Linux环境的同学漫长的尝试事件后，最终发现使用鼠标中键才可以将程序复制到终端上。

在接下来的实验中，我也遇到了许许多多的问题，但我对硬件开发的理解也是一直在增长。

在实验二中，我对单片机内的各种时钟频率有了更清晰的认识。实验二刚开始时，我对于机器周期、指令周期、时钟周期等一系列周期的理解非常混乱。在实验二的过程中，我认识到单片机中时钟信号的实现是由机器内部的振荡器通过震荡电路产生的信号，不同的振荡器产生的时钟频率不相同，同一个振荡器在不同的温度等条件下震荡频率也不相同。

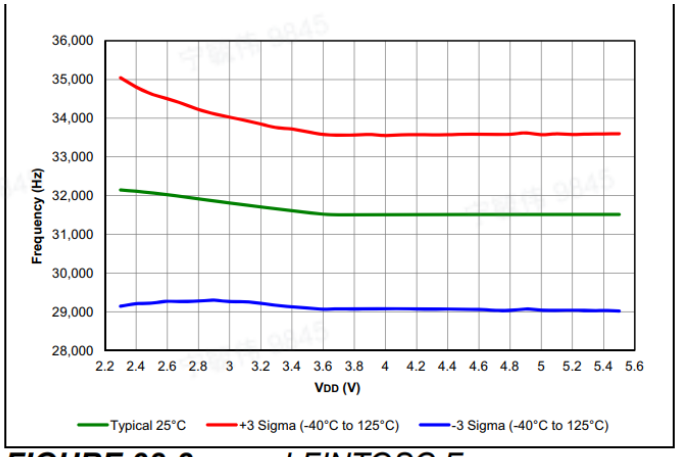

FIGURE 38-8: *LFINTOSC Frequency, PIC16F18854 Only.*

LFINTOSC的频率随着温度的变化而发生改变

同时我还认识到，本次实验的框架中机器周期设为1mHz，并且指令周期为机器周期的四倍250kHz。深入理解了时钟周期、指令周期之后，我便能够使用程序精确的控制计时器，并使用轮询的方式进行计数器溢出的判断，并通过nop指令延时补偿，得到精确的500ms计时，这真的是太酷啦😙😙😙。

```
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
```

精确的500ms仿真结果

实验三是在实验二的基础上进行的中断闪灯实验，将实验二中轮询计数器溢出标志的部分改为计数器溢出中断，并在中断处理程序中完成二极管电平的反转。由于我之前学习的x86汇编课程中有对中断理论的讲解，因此这部分实验对于我来说并不算特别难，只需要按照数据手册对中断控制相关的寄存器进行相应的赋值即可。

进入到实验四，代码量逐渐开始多了起来，因此程序的编写规范变得相当的重要。在实验四的过程中，我将数码管的阴极引脚以及阳极引脚全部使用宏定义进行表示，同时还定义了一些有用的常量，例如数码管各个位对应的阴极选择信号、数码管可显示的数字总数以及显示在数码管上数字的数量。这些宏定义在实验七：串口实验中被证实是非常有用的，因为我们在实验四中将数码管的控制引脚设为了端口B，而端口B在实验七中需要腾出B7与B6进行串口通信。由于上述宏定义的存在，我们对代码的改动仅需要将宏定义中的数码管控制信号改为另一个端口即可，对于连线的改动也只需要将端口B上连接的线移动到另一个端口即可。

在实验四中，我也遇到了许多问题。刚写完程序时，我发现自己的数码管流水灯有一段时间会全部熄灭，而我的程序中将阴极信号全部拉高的逻辑仅出现在main函数中，而main函数仅会运行一

次并结束。我根据已有的知识思考原因，并初步认为这是因为单片机在执行完main函数后程序计数器并不会停止递增，而是会一直加满溢出后重新执行main函数。但是后来发现我这个想法是错误的，因为有同学告诉我这个框架内设置了一个看门狗，并在一段时间后会重新执行我们写的程序。仔细想来，我会产生错误推断的原因是因为我对烧写到单片机上的程序还没有一个全面的了解，对于框架上写好的配置字也没有细究。

在实验五：按键实验的过程中，我对于按键检测的代码迟迟下不了手，这是因为我并不想使用最简单的逻辑对按键一个接一个的检测，这样会使得我的代码堆积成一堆包含各种字面量的小山丘，破坏代码整体的规范以及美观（美观的代码对我来说真的很重要）。因此我在实验开始之前对整体的轮询过程进行了仔细的思考，并发现可以用三态门的移位操作简单的改变引脚的输入输出模式，同时还可以使用掩码变量的与运算获取对应引脚的电压值，从而判断按键是否被按下。在理论课上，老师要求我们使用流程图来描述整个实验的过程，但是我个人并不太习惯流程图的方式，反倒更喜欢以伪代码的方式进行思路的呈现（可能是因为我看代码比较多，对流程图的方式不算特别习惯）。因此我首先对整个轮询过程进行了伪代码的编写。在编写伪代码的过程中，我也尽量注意伪代码尽量贴近XC8汇编代码，因为这样一来我就可以将伪代码轻松的转换成汇编代码。由于前期的伪代码的设计十分完善，在转成汇编代码的过程中几乎不存在思路上的问题，反倒是在对于单片机的控制上出现了一些细节上的问题。例如我在实验的过程中发现数码管的显示有残影（其实是老师提醒我的）。由于我在编写代码的过程中注意各个模块之间尽量没有任何耦合现象的存在了，对于数码管残影问题我也仅针对数码管显示模块的逻辑进行修改就解决了问题。

在实验六中，我将之前编写的汇编代码全部移植为C语言代码。得益于编写汇编代码的时候注意编程的规范，整个移植的过程都相当的顺利。在测量单片机的电压时，我首先查看ADC的电路框图，并使用高亮标记出需要用到的输入输出信号，接着在PDF文档中搜索相应的选择信号进行选择。但是这样一来还是有所疏忽：我忽略了对FVR_buffer电压信号的使能信号的设置，因为在ADC的电路框图中并没有明确指出要设置其选择信号。看来读数据手册仅仅查看框图还是远远不够的。

在实验七中，我们不仅仅着眼于单块板子上程序的编写，还需要考虑多个机器之间从串口通信。由于在前面的实验中已经积累了一些开发经验，在实验七的过程中我感觉还是比较顺利的，我还使用了~~花里胡哨的~~颜色控制字符串对输出在终端中的简易示波器信号进了上色🥰，同时采用了控制光标位置的方式来代替空格的打印，提高了示波器的刷新率。

总的来说，我在整个微机原理实验过程中学到了非常多的硬件开发知识，同时也加深了和种子班中的小伙伴们的关系。在整个实验过程中，我和班上的同学相互帮助（更多的时候是我被帮助hhh），共同撑过了十几天的学习过程，这门课程学习也让我对下个学期在种子班内的生活更加向往。