**知识点: 指令类型+指令系统设计+指令系统改进+实例**

1. 了解常用的四种指令类型即可
2. 指令系统设计是包括什么设计和设计？理解其功能设计分别用软件和硬件实现时候的特点。
3. 指令系统设计的基本要求，知道这个要求的概念即可。指令格式设计中，通常CISC使用的什么编码格式？RISC采用什么编码格式？
4. CISC指令系统的特点和不足？RISC指令系统的特点，即设计师遵循的选择。理解8/2原则是RISC实现的依据。
5. 理解和体会7.7节 MIPS指令系统的案例在本课本中的知识体系逻辑，知识逻辑如下：本教材以MIPS这个CPU为例，首先进行其指令系统的设计，然后在第8和第9章节中针对设计的此指令系统进行具体实现。MIPS在这里是RISC指令集结构，实现的时候往往**基于硬件连接电路的方式去实现**，如何实现，**可先行预习8.4章节内容**。但是，这种通过硬件逻辑电路实现CPU指令实现的方式存在不足，电路可能会很复杂，因此，往往在实现的时候，还通过会采用**微程序控制的方式**来实现，这种微程序控制的方式就是我**们第9章的内容。**
6. 熟悉MIPS这个案例中的指令系统，围绕如下方面：
7. 可供指令使用的寄存器类型。因为指令以及数据要进入寄存器的。
8. 数据表示的方法。
9. 寻址的方式，体会这里内存是按字节编址的，具体地址为64位。数据在内存存放的时候讲究边界对齐，以便好读写数据。
10. MIPS指令设计为32位，共有3种指令格式，见图7.14.

基于上述几点共识，熟悉MIPS的三种指令格式，针对这三类指令

格式，MIPS所设计的具体指令有哪些，理解其操作方式，也体会通用寄存器和浮点寄存器的不同，一般通用寄存器是放的整数，浮点寄存器是小数。具体要求：看懂表7.3，表7.4，表7.5即可。

**常见问题：**

* 1. 关于p185页表7.3中，其中关于装入半字的方式为何是48+8+8

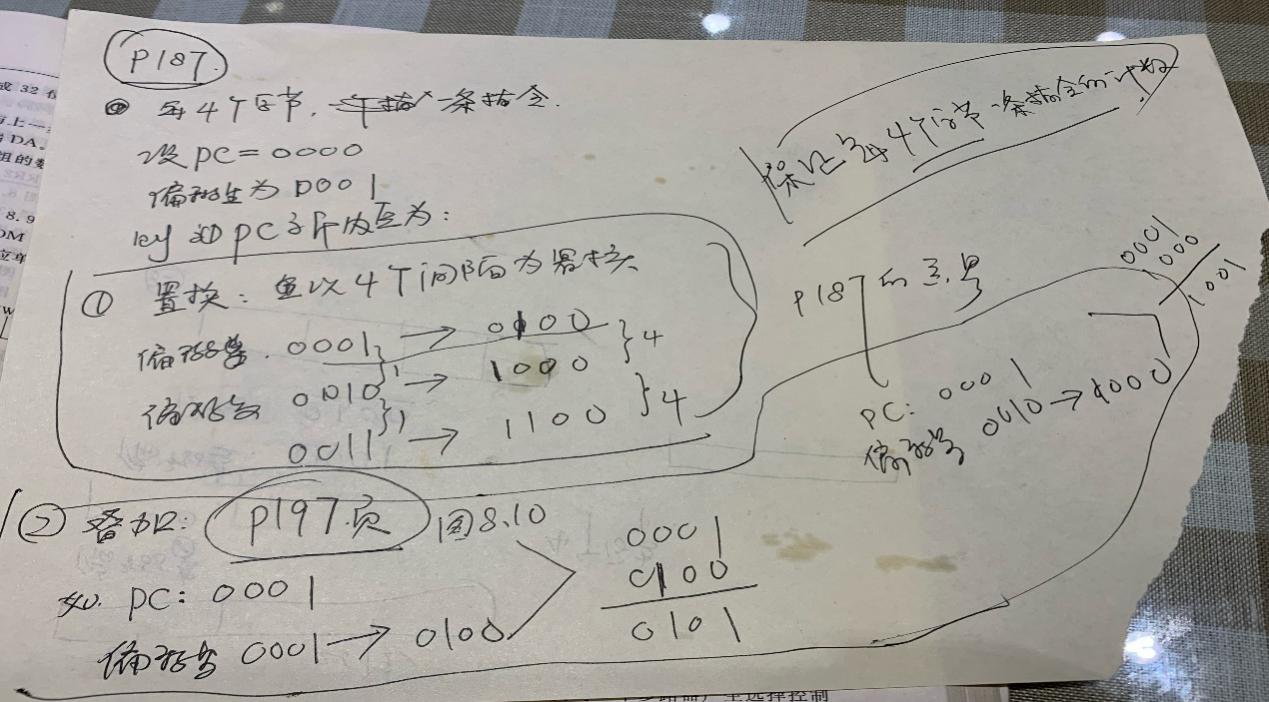
这种方式，而非48+16这种方式。

我的理解为：如果是边界对齐的，其实也可直接写48+16的这种表示方式，相当于连续读取16位，然后扩展了48位的符号位，构成64位的数据放到寄存器中。如果不是边界对齐的方式的话，可能会存在这种一个字节一个字节读取的方式。这么来说的话，表7.3的装入双字、装入字类似于这样的指令在解释其功能的具体操作过程的时候，也应该写成一个字节一个字节读取的方式。如果对齐的话，那就无所谓了，连续读取就可以了，如同书上现在的表示的一般。课本上目前都默认是边界对齐的，所以没有关系了，怎么表示这个功能的解释，都可以的。

2）编址和寻址是同样的意思吗？总是看到“按字节寻址”“按字节或字编址”这些语句，感觉有点混淆了。  
 **可如下理解**：这两个是有差别。编地址，是站在内存的角度来说 ，是以一个字节给一个地址，还是以一个字给予一个地址。而寻址是站在指令的角度来说的，这个指令能识别多少个地址，也就是会有多少个编码。至于，每个地址对应的内存是多大，它不关心的，一个地址对应一个字节也可以，对应一个字也可以。

3）为什么要左移两位？

在这个案例中，每条指令是32位的，即4个字节。PC转到下一条指令，表现在PC内容上，其实是PC+4，这里的4就是4个字节。在MIPS指令设计中，J类指令中的立即数，如取a，则跳转到a，那么，当这个数变为a+1时，应该跳转到哪里呢？所以，才会有了左移两位的含义。具体，可以如下枚举体会体会。这个左移2位的意思在后面第八章如P197还会出现，道理同此。



**预习内容：**

8.1-8.4节，重点是看看8.4内容，看起来可能比较晦涩，大致了解一下，能理解到什么程度先体会一下 ，以便下节课学习。