https://blog.csdn.net/sandalphon4869/article/details/103685019?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522162513048316780265424127%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request\_id=162513048316780265424127&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~blog~first\_rank\_v2~rank\_v29-9-103685019.pc\_v2\_rank\_blog\_default&utm\_term=%E8%AE%A1%E7%BB%84&spm=1018.2226.3001.4450

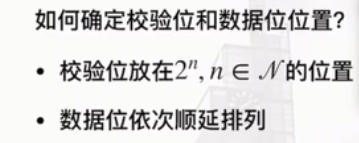
1. CIP/IPC/MIPS/FLOPS
2. 优化求加速比，两个定理
3. IEEE-754（分单/双精度）表示
4. 浮点数规格化表示与非规格化表示的范围
5. 注意：移码运算所得的结果是实际结果的补码
6. 1字节（DB）=8位，1字（DW）=2字节；1双字（DD）=2字=4字节
7. 海明码（汉明码，缺点是只能修正一位）

<https://space.bilibili.com/197506412/video>

1. 确定校验位数（不需要记忆，可以根据（2）分析）



1. 确定校验位的位置（依次插入数据中）



1. 确定每个校验位的值



举例说明，P1可以覆盖：01，11，101，111，1001，1011……

1. 计算每一位的校验值（自行决定奇偶校验）
2. 循环冗余检验（CRC。可以视为一种信息摘要，可用于校验）

①在待编码的数据后面加上(n-1)位0，其中n为生成多项式编码位数

②用上面加完0后的数做除法运算（异或），直到除尽，剩下的就是余数（FCS）

③把余数加在之前待编码的数据（未加0）后面，即为编码（接收端除以生成多项式，只有可以除尽时，才算编码正确）



1. 运算方法

（1）定点乘法运算

①一位原码乘法运算

②两位原码乘法运算

③一位补码乘法运算（Booth法）

（2）定点除法运算

①恢复余数法

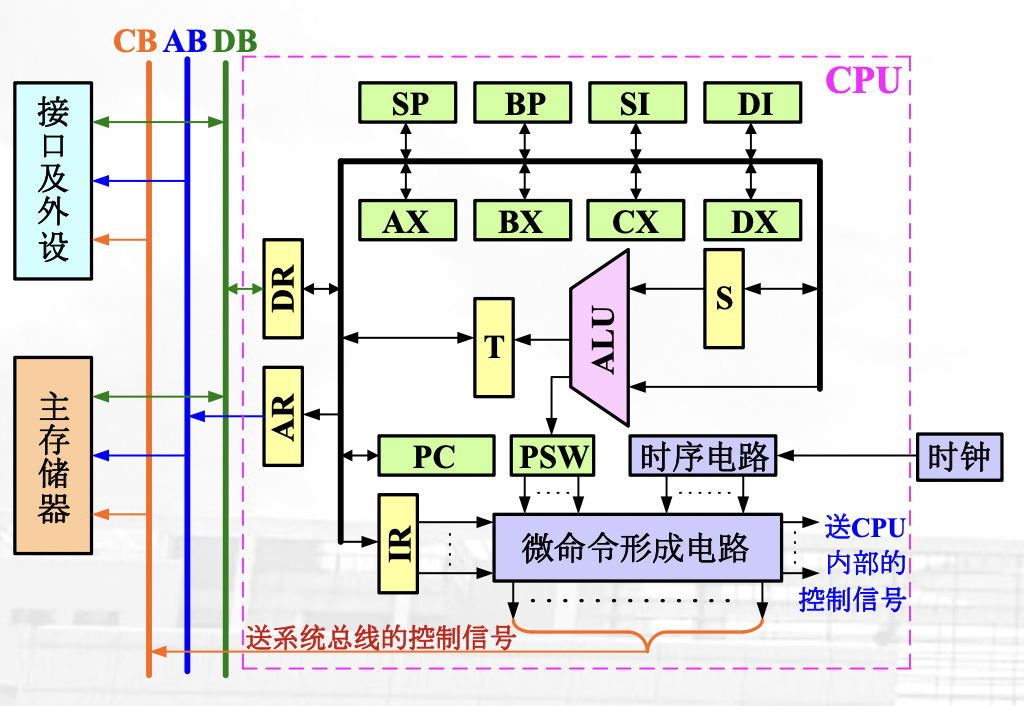
②原码加减交替法

③补码加减交替法

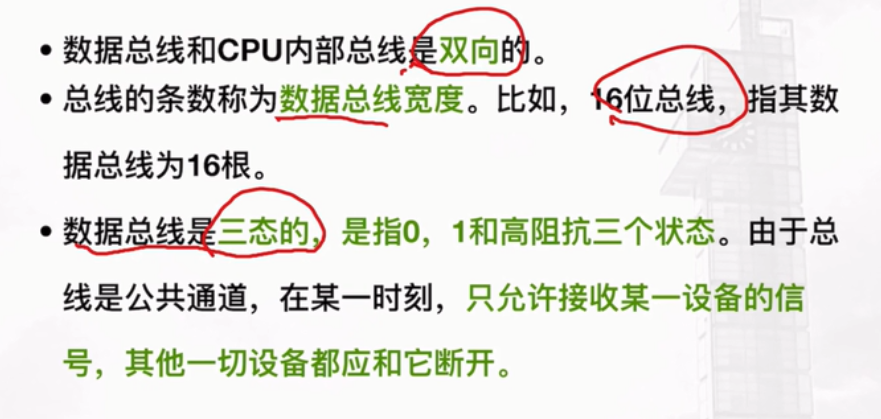
1. 浮点加减运算

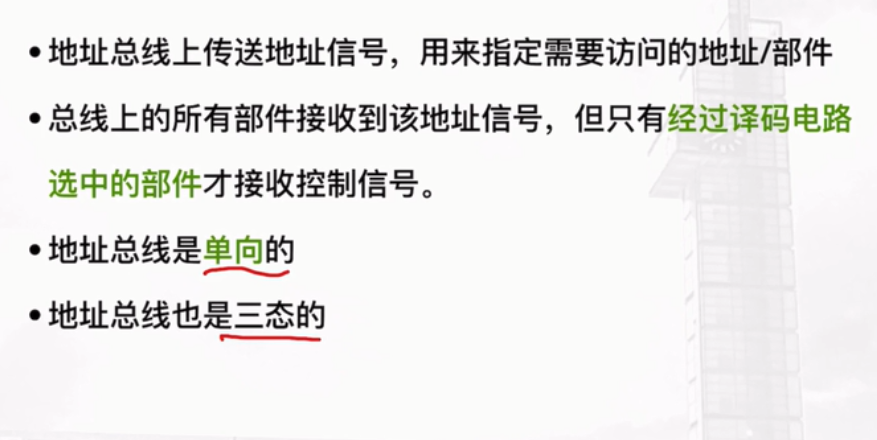
小阶对大阶，注意结果的规格化和舍入

1. CPU执行指令的3(4)流程循环：取指令→译码→执行（→存储）





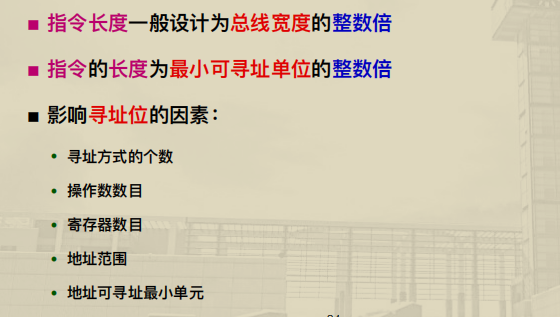




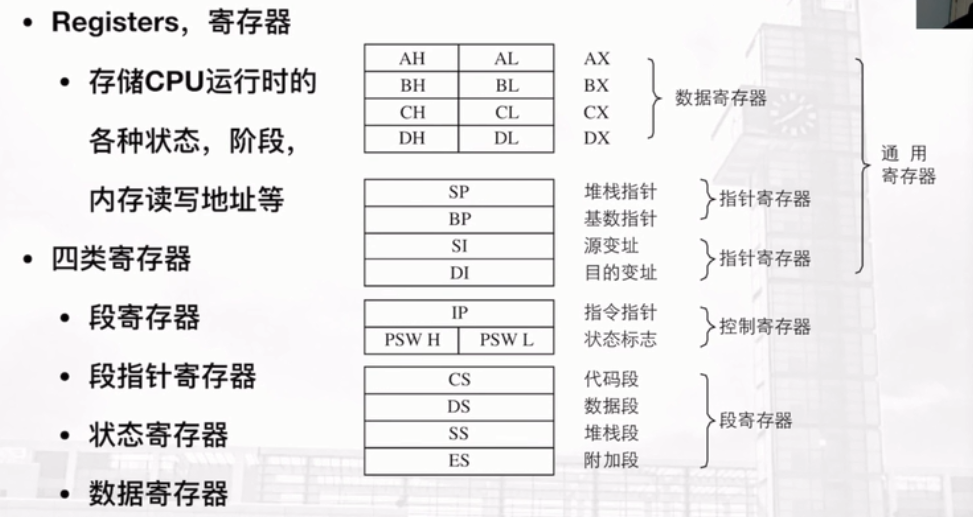
1. 指令格式

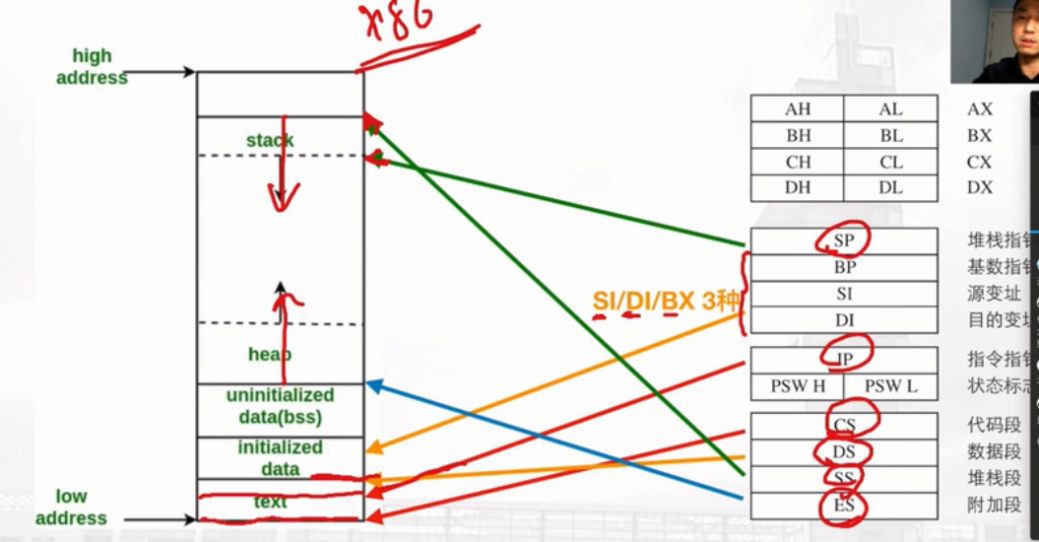


1. 指令长度

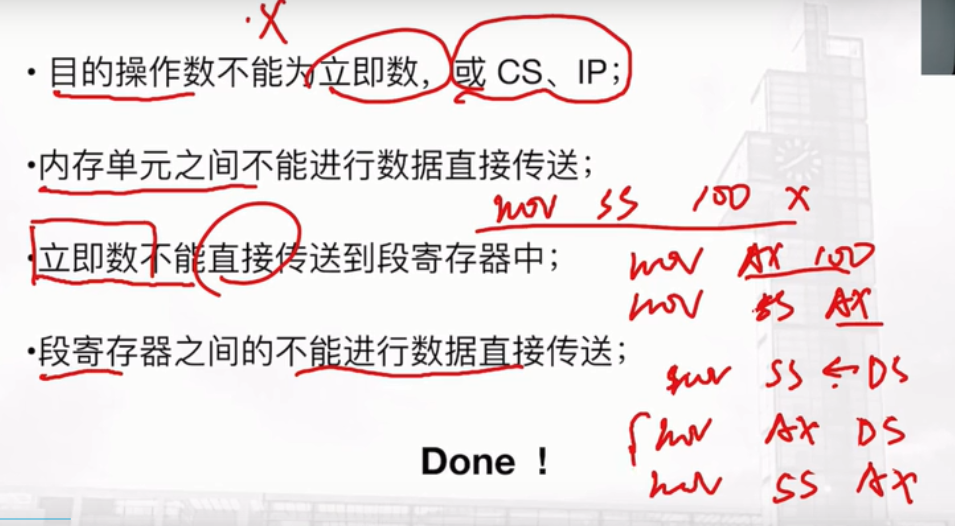
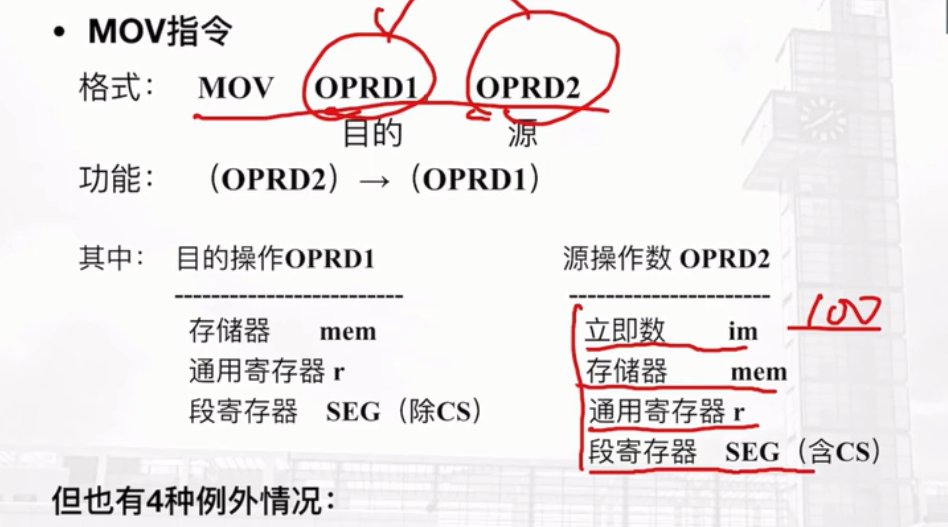


1. 优秀的堆设计可以避免碎片化
2. 寄存器（8086）





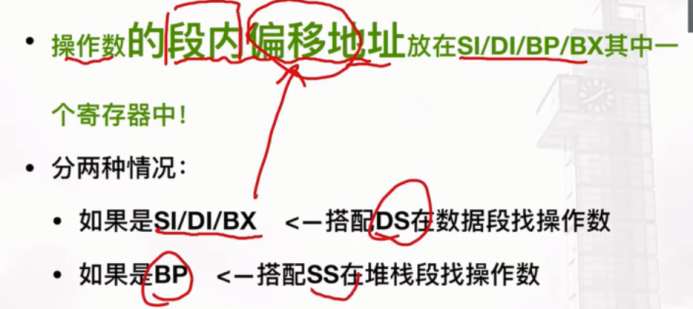
1. 寻址

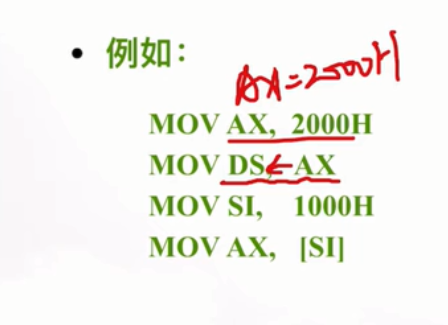


1. 寻址方式

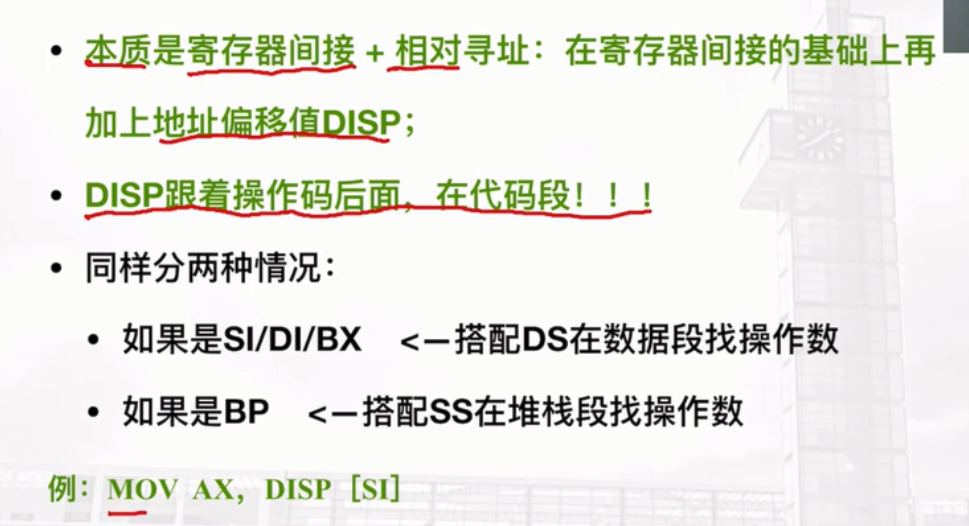
<https://blog.csdn.net/sandalphon4869/article/details/90682122>

寄存器间接寻址：注意偏移地址所在的段

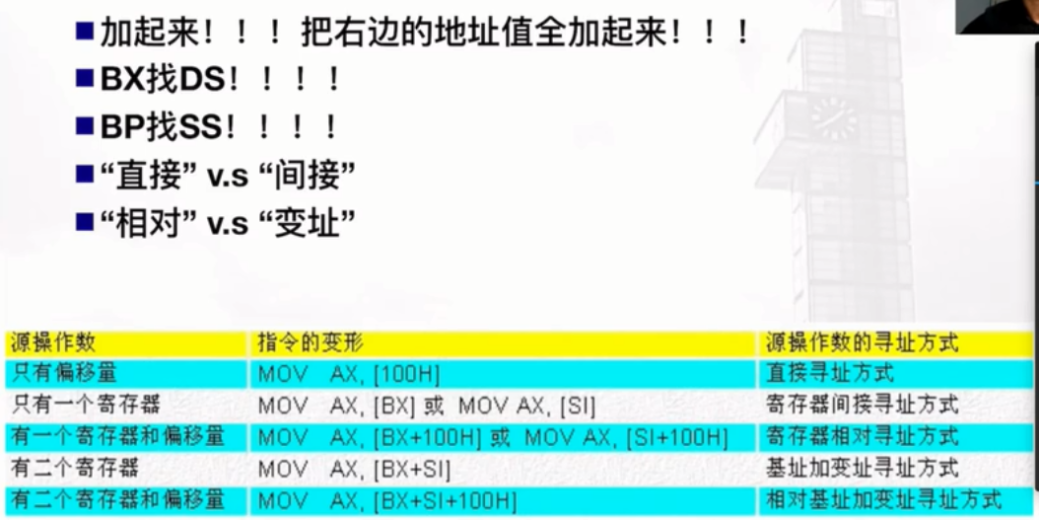




特殊：

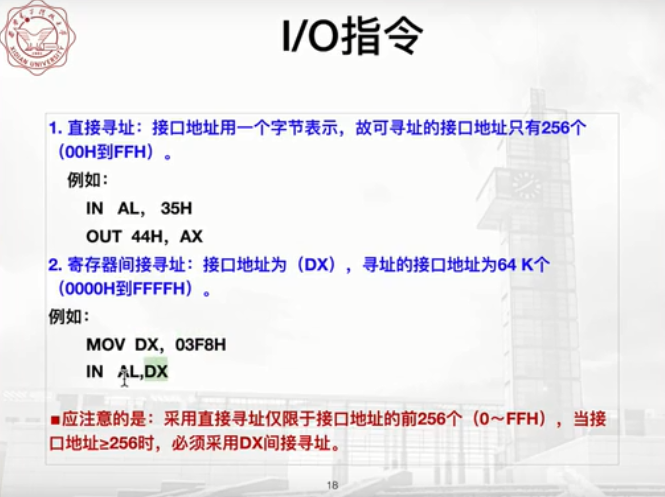


小总结

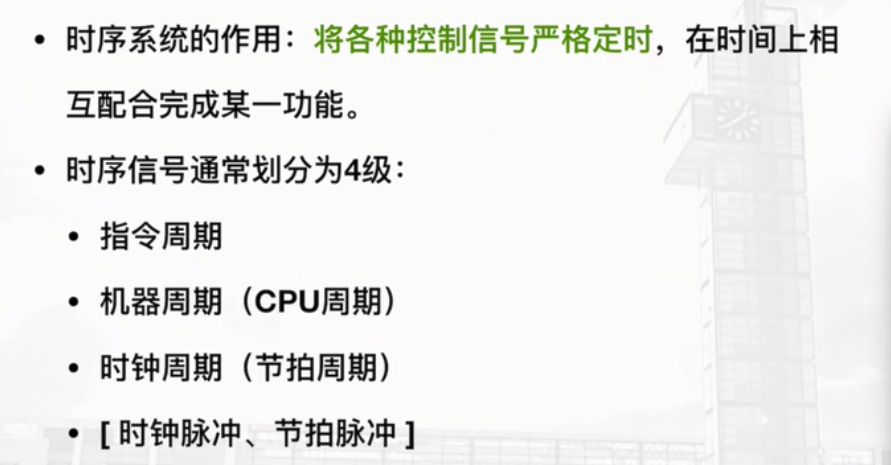


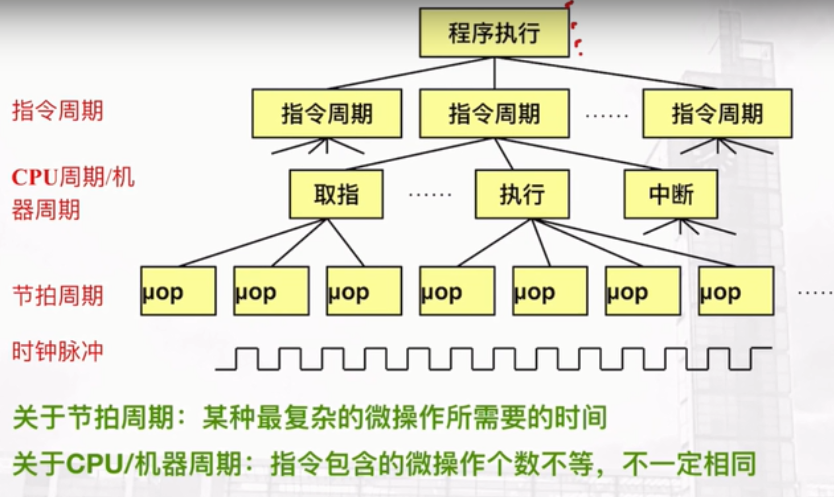
1. SEG取所在的段首地址；OFFSET取段内偏移地址

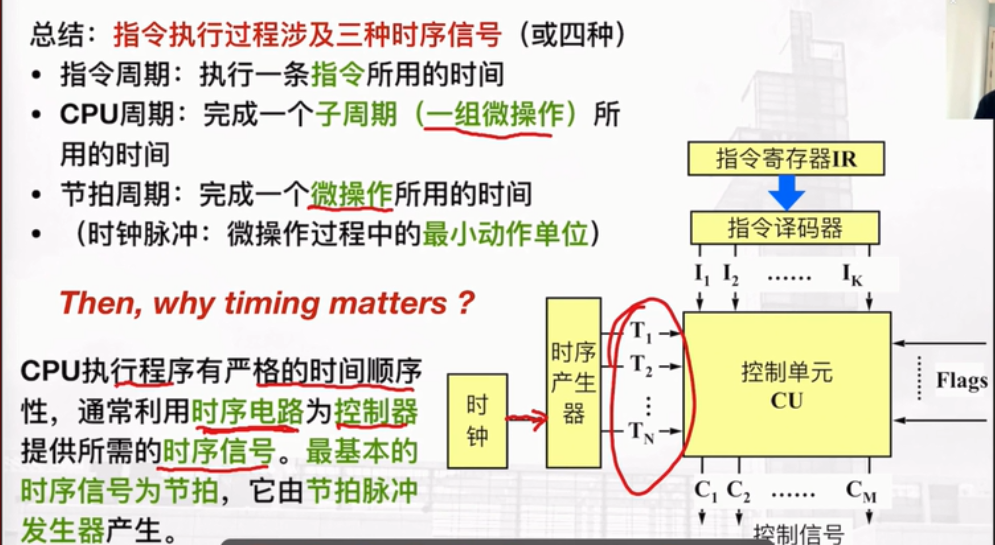
18、



19、时序系统



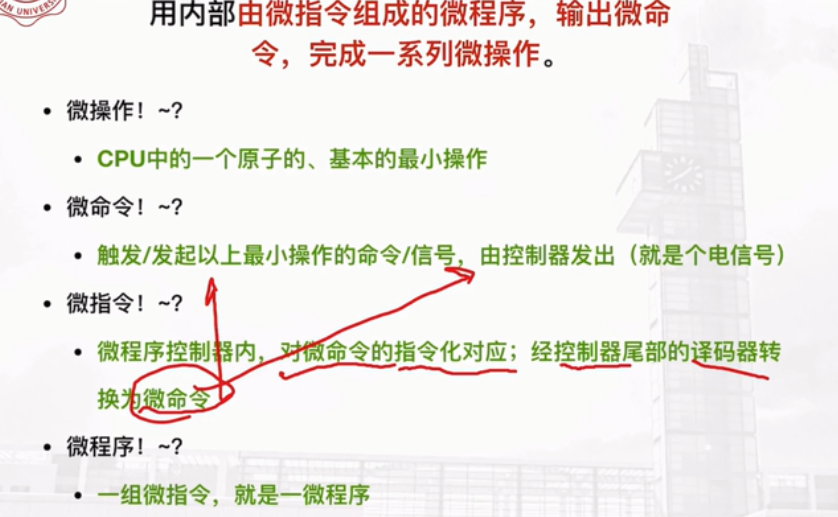


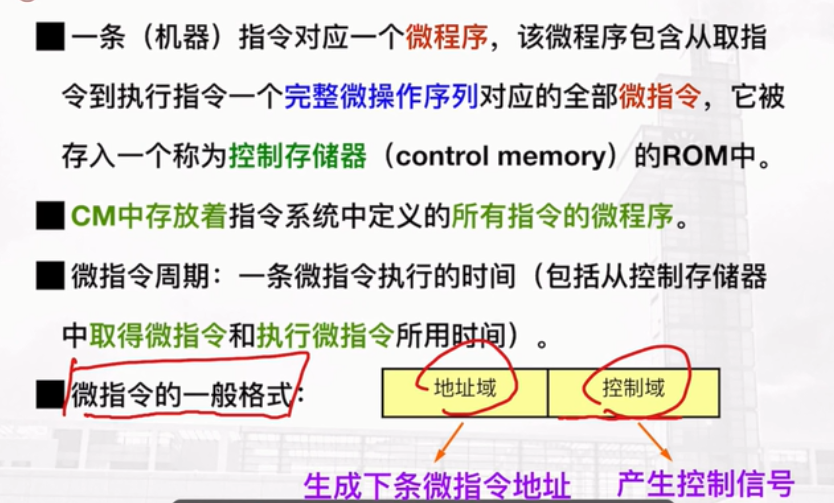


20、CU的作用

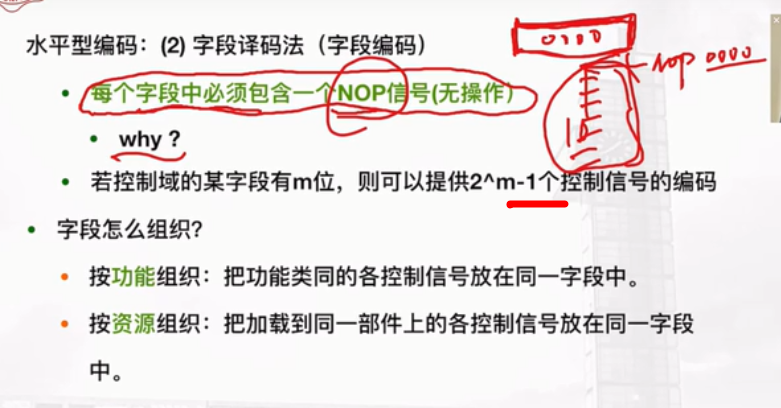


21、微指令

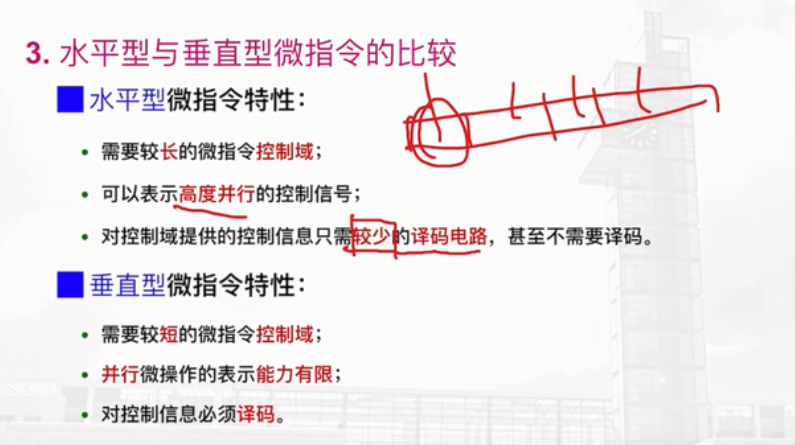




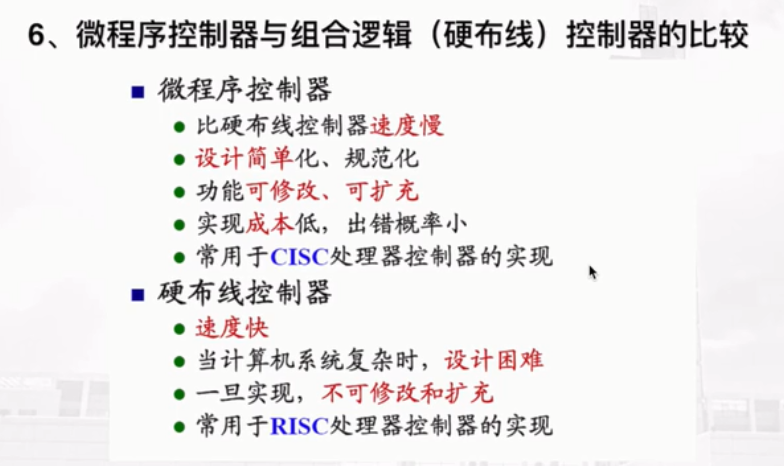
22、控制域重点



23、水平型微指令与垂直型微指令



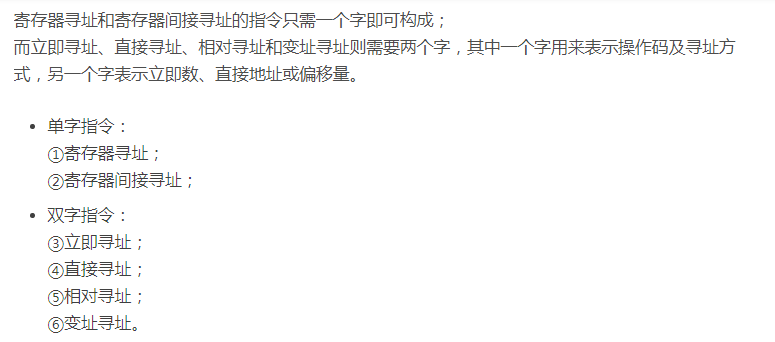
24、微程序控制器与组合逻辑控制器比较



25、操作码

0XX：双操作数；0E：单操作数指令；OF：转移指令；OFF：无操作数指令

26、寻址



1. 更多待补充