1. **计算机组成：一条指令被执行的过程**

指令寄存器是临时存放从内存中取出来的程序指令的寄存器。

通用寄存器用来传送和暂存数据的，还可以参与算数逻辑运算。

程序计数器用来存放下一条指令所在单元的地址。

状态寄存器是运算器的一部分，用来存放两类信息，1、当前指令执行结果的状态信息（如有无进位，有无溢出等），2、存放控制信息（如是否允许中断）

当执行一条指令，计算机首先要将指令所在的内存地址从程序计数器中取出来，从而知道它在内存什么地方放着，然后把取出的地址交给地址总线，然后地址总线将这条指令取出来，取出来这条指令后就将这条指令放到指令寄存器中，然后程序计数器+1指向下一条指令的地址。当指令寄存器中的指令执行完之后，再根据程序计数器的地址再取一条指令出来放到指令寄存器中如此往复循环执行。

1. **浮点数所能表示的数值范围：**



尾数的位数越多它表示的精度越高。

阶码的位数越多它表示的数值范围越大。

最大正数：，最小负数 【记忆该公式】

M表示尾数位数，R表示阶码位数

1. **Cache到主存的映射**

全相联地址映射：主存的任意一块可以映象到Cache中的任意一块。

直接相联映射：主存中一块只能映象到Cache的一个待定的块中。

组相联的映射：各区中的某一块只能存入缓存的同组号的空间内，但组内各块地址之间则可以任意存放。即从主存的组到Cache的组之间采用直接映象方式，在两个对应的组内部采用全相联映象方式。

1. **移位操作指令**

移位操作指令是一组经常使用的指令，属于汇编语言逻辑指令中的一部分，它包括移位指令（含算术移位指令、逻辑移位指令），循环移位指令（包带进位的循环移位指令），双精度移位指令三大类。

* 算术左移SAL把目的操作数的低位向高位移，空出的低位补0。
* 算术右移SAR把目的操作数的高位向低位移，空出的高位用最高位（符号位）填补。
* 算术移位指令常用于带符号数\*2或/2，左移相当于\*2，右移相当于/2。
* 逻辑左移/右移指令只有它们的移位方向不同，移位后空出的位都补0。
* 逻辑移位指令常用于无符号数\*2或/2。

循环左移/右移指令只是移位方向不同，它们移出的位不仅要进入CF（进位标志），而且还要填补空出的位。可以理解为蛇咬尾巴型循环。

1. **并行总线与串行总线**

并行总线，速度快，传输距离短。串行总线，速度相对来说慢，但是传输距离很长