**CPU**

一.CPU的组成

1.运算器



(1).算术逻辑运算单元ALU，它能完成各种算术运算和逻辑运算。最基本的算术运算主要包括加、减、乘、除等，逻辑运算主要包括与、或、非、异或及移位运算等.

(2).ALU一般有两个输入端，它能一次完成两个操作数的运算。另外在运算器中还会设置一些通用寄存器R，用于暂时存放运算中产生的中间结果。

2.控制器



1. .指令寄存器IR（Instruction Register）主要用于存放由PC指向的从存储器中取出的指令代码。指令寄存器一般为一个指令字长，它主要由两个字段组成，一是指令操作码OP字段，二是操作数或转移地址Addr字段。OP字段用于指出该指令是一条什么样的指令，如加法、移位等；Addr字段根据指令的不同功能有所不同，对于顺序指令，Addr字段用于指出操作数的类型及存放的位置或地址，而对于转移或转子指令，Addr字段则用于指出要转向的指令的地址。
2. .地址生成器AG（Address Generator）主要用于生成操作数在存储器中的单元地址，从而为取该操作数做好准备。
3. .地址寄存器AR（Address Register）主要有两个用途：一是用于存放由地址生成器按寻址方式进行计算得到的操作数在存储器中的地址；二是用于存放由当前转移或转子指令产生的要转向的指令的地址。
4. .控制信号产生部件就是根据指令译码的结果产生当前指令执行过程中所需的全部操作控制信号，这些控制信号在时序部件产生的时序下按照一定的顺序逐个产生，从而控制不同的部件完成相应的操作。
5. .指令译码器ID（Instruction Decoder）用于对指令寄存器中的OP字段进行译码，并将译码结果输出给控制信号产生部件
6. .程序计数器PC（Program Counter）主要存放下一条要执行的指令的地址
7. .时序部件CP用于产生时序信号

3.Cache

为了进一步提高CPU的运行效率，目前的CPU内部通常会内置高速缓冲存储器。内置Cache的容量和结构对CPU的性能影响较大，一般容量越大越好。在许多高性能处理器 内部，一级缓存通常设置为两个，一个指令Cache，一个数据Cache，以减少取指令和读操作数的访问冲突，这种结构也叫哈佛结构。

二.指令系统

1. .指令系统分类

CISC（复杂指令系统计算机）：CISC的设计思想是提供尽可能丰富的指令系统，为程序设计者提供最大的方便。其基本的指令系统通常有几百条指令。

RISC（精减指令系统计算机）：选取使用频率高的一些简单指令，指令条数少；指令长度固定，指令格式种类少；只有取数／存数指令访问存储器，其余指令的操作都在寄存器之间进行。

(2).指令的分类

a.数据传送类指令：主要包括取数指令、存数指令、传送指令、成组传送指令、字节交换指令、清累加器指令、堆栈操作指令等

b.算术运算类指令：包括二进制定点加、减、乘、除指令，浮点加、减、乘、除指令，求反、求补指令，算术移位指令，算术比较指令，十进制加、减运算指令等

c.逻辑运算类指令：包括逻辑加、逻辑乘、按位加、逻辑移位等

d.程序控制类指令：条件转移指令、无条件转移指令、转子程序指令、返回主程序指令、中断返回指令等

e.输入输出类指令：主要用来启动外围设备，检查测试外围设备的工作状态，并实现外部设备和CPU之间，或外围设备与外围设备之间的信息传送

f.特权指令：是指具有特殊权限的指令

g.处理器控制指令：状态寄存器置位、复位指令、测试指令、暂停指令，空操作指令，以及其他一些系统控制用的特殊指令

(3).指令格式



1. 操作码:操作码指出指令做什么。用操作码字段的不同编码来表示不同的指令，每一个编码代表一种指令。组成操作码字段的位数一般取决于计算机指令系统的规模。操作码的编码方法包括：定长操作码编码；变长操作码编码
2. 地址码：根据一条指令中有几个操作数地址，可将该指令称为几操作数指令或几地址指令。几种常见的指令格式：零地址指令，一地址指令，二地址指令，三地址指令

(4).指令的寻址方式

1. 顺序寻址方式:指令地址在内存中按顺序安排，当执行一段程序时，通常是一条指令接一条指令的顺序执行。
2. 跳跃寻址方式:当程序转移执指令地址字段中的形式地址D不是操作数的真正地址，而是操作数地址的指示器，D单元的内容才是操作数的有效地址。行的顺序时，指令的寻址就采取跳跃寻址方式。所谓跳跃，是指下条指令的地址码不是由程序计数器给出，而是由本条指令给出。程序跳跃后，按新的指令地址开始顺序执行。指令计数器的内容也必须相应改变，以便及时跟踪新的指令地址。

(5).操作数寻址方式

1. 立即寻址:指令的地址字段指出的不是操作数的地址，而是操作数本身。
2. 直接寻址:在指令格式的地址字段中直接指出操作数在内存的地址。
3. 间接寻址:指令地址字段中的形式地址D不是操作数的真正地址，而是操作数地址的指示器，D单元的内容才是操作数的有效地址。
4. 寄存器寻址:指令中给出的操作数地址不是内存的地址单元号，而是通用寄存器的编号。
5. 寄存器间接寻址:指令格式中的寄存器内容不是操作数，而是操作数的地址，该地址指明的操作数在内存中。
6. 相对寻址:把程序计数器PC的内容加上指令格式中的形式地址D而形成操作数的有效地址。
7. 变址寻址:把CPU中某个变址寄存器的内容与偏移量D相加来形成操作数有效地址。
8. 基址寻址:将CPU中基址寄存器的内容加上指令格式中的形式地址而形成操作数的有效地址。
9. 段寻址:一种为了扩大寻址范围而采用的技术，在Intel 8088/8086等处理器中使用。
10. 复合寻址:把两种或两种以上的基本寻址方式组合在一起的寻址方式。

(6).堆栈寻址

堆栈(Stack)是一种按特殊顺序进行存取的存储区。存放在主存储器中的数据，只要给出地址就能立即从存储器中读出，而不管这个数据是先写入还是后写入的，而堆栈的存取顺序是“进先后出”(Last In First Out)；堆栈的设置通常有两种方式：硬堆栈和软堆栈。硬堆栈是利用CPU中的一组专门的寄存器来组成，硬堆栈的容量有限。软堆栈是通过执行相关指令，把主存储器中的一段存储区定义为堆栈，并利用一个通用寄存器作为堆栈指针SP(Stack Pointer)，并设置SP的初值。目前大多数计算机中都支持软堆栈，软堆栈设置灵活，可满足较大容量的要求，可以用对存储器寻址的指令来对堆栈中的数据进行访问。

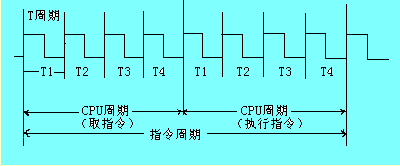
1. 指令周期

(1).指令周期:CPU从内存取出一条指令并执行这条指令的时间总和。

(2).CPU周期:又称总线周期、机器周期，是指CPU进行一次总线操作所需要的时间。包括CPU访存、访I/O和中断响应等。

(3).时钟周期:又称为T周期，是指CPU工作时钟的一个周期时间。

一个指令周期由若干个总线周期组成；一个总线周期由若干个T周期组成。



LDA 00H:它需要两个CPU 周期，其中取指令阶段需要一个CPU周期，执行指令阶段需要一个CPU周期;

ADD [E0H]:它需要三个CPU 周期：取指令周期、取操作数周期和执行加法周期;

STR [F0H]:需两个CPU 周期：取指令周期和存数周期;

JMP 11H:它由两个CPU周期组成：取指令周期和执行跳转周期

1. CPU设计方法
2. 硬布线设计法（又称组合逻辑设计法），采用组合逻辑思想设计；

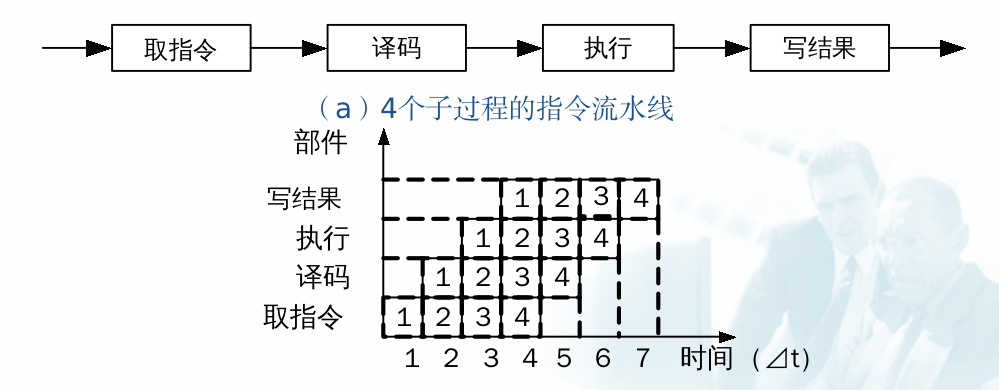
将操作控制信号产生器用组合逻辑网络设计实现，它根据指令译码的结果，在一定的时序控制下由组合逻辑电路产生所有指令执行时所需的全部控制信号。

1. 微程序设计法，采用存储逻辑思想设计

把每条指令执行过程中应产生的操作控制信号编成二进制微码，事先存放在一个只读存储器中，一条机器指令对应一段这种微码。执行指令时，则将该指令对应的这段微码取出，由相应位产生操作控制信号。

五.并行技术

1.时间重叠：多个处理过程在时间上相互错开，轮流重叠地使用同一套硬件设备的各个部分，以加快硬件周转而赢得速度。



1. 资源重复：根据“以数量取胜”的原则来实现并行，其付出的代价是在空间上通过重复地设置资源，尤其是硬件资源，以提高计算机系统的性能。
2. 资源共享:是一种软件方法的并行，它使多个任务按一定时间顺序轮流使用同一套硬件设备。资源共享的实质就是用单处理机模拟多处理机的功能，形成所谓虚拟机的概念。