

中央处理器

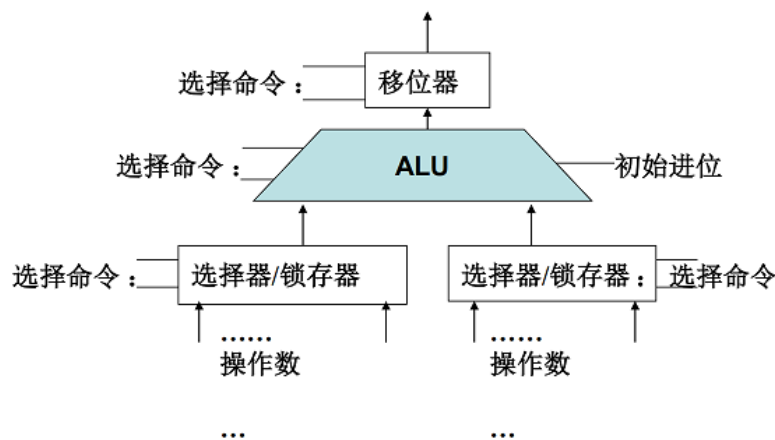
笔记本： 计算机组成原理

创建时间： 2021/6/26 9:07

更新时间： 2021/6/29 12:30

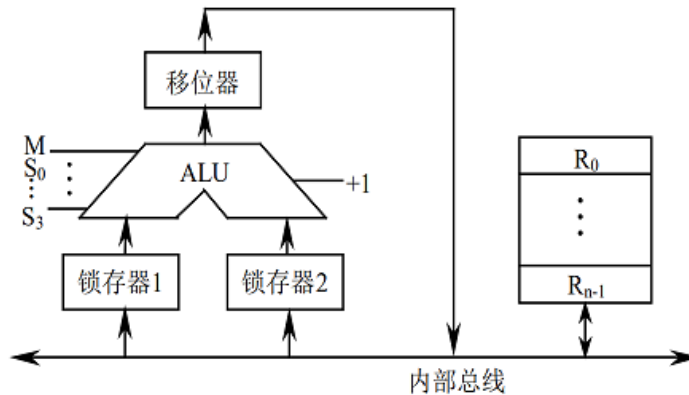
作者： 134exetj717

- 我们每天都使用电脑，那我们知道电脑的核心是什么吗？
 - (CPU) 中央处理器，是整个计算机的核心。
 - 它包括：运算器和控制器。
- 那么CPU的功能有哪些？
 - 指令控制：产生下一条指令在内存中的地址。
 - 操作控制：产生各种操作信号送往相应部件，以控制完成指令所要求的动作。
 - 时序控制：对各操作信号实施时间上的控制，以保证计算机有条不紊地连续自动工作。
- 简而言之，CPU的功能是：
 - 对指令流和数据流在时间与空间上实施正确的控制。
- 了解了其功能以后，我们可能会好奇，CPU的具体组成有哪些呢？
 - 控制器
 - 从主存中取出一条指令，并指出下一条指令在主存中的位置；
 - 对指令进行译码或测试，产生相应的操作控制信号，以便启动规定的动作；
 - 指挥并控制CPU、主存和输入输出设备之间的数据流动方向。
 - 运算器
 - 运算部件由算术逻辑单元 (ALU)、累加寄存器、数据缓冲寄存器和状态条件寄存器组成。
 - 执行算术运算和逻辑运算；
 - 同时能够暂时存储计算得到的结果。



- 寄存器：
 - 用来暂时保存运算和控制过程中的中间结果、最终结果以及控制、状态信息的。
 - 通用寄存器
 - 用来存放原始数据和运算结果。
 - 当然，有的还可以作为变址寄存器、计数器、地址指针等
 - 专用寄存器
 - 专门用来完成某一种特殊功能
 - (1) 程序计数器 (PC)
 - 又称指令计数器：用来存放正在执行的指令地址或者接着要执行的下一条指令地址
 - (2) 指令寄存器 (IR)

- 存放从存储器中取的指令
- (3) 存储器数据寄存器 (MDR)
 - 用来暂时存放由主存储器读出的一条指令或一个数据字；
 - 反之，当主存写入一条指令或一个数据字时，也暂时将他们存放在存储器数据寄存器中。
- (4) 存储器地址寄存器 (MAR)
 - 保存当前CPU所访问的主存单元的地址
- (5) 程序状态字寄存器 (PSW)
 - 又称状态标志寄存器：存放程序状态字
 - 状态字一般包括：进位标志 (C)，零标志 (Z) 等。
- CPU内部的数据通路
 - 单组内总线数据通路结构



- CPU与主存间的信息交换（两种结构类型）
 - 将存储器作为一个独立部件，通过系统总线和CPU连接

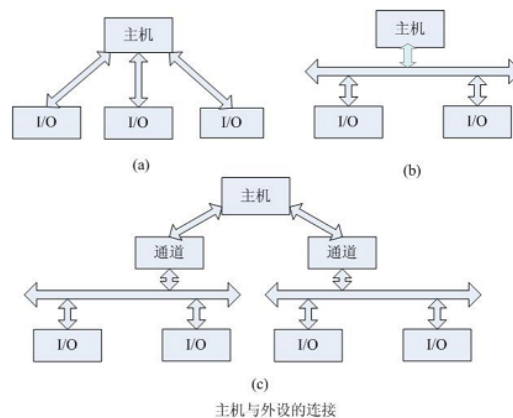


- 将主存储器当作主机的一部分，用一组专用的存储总线将其和CPU相连接

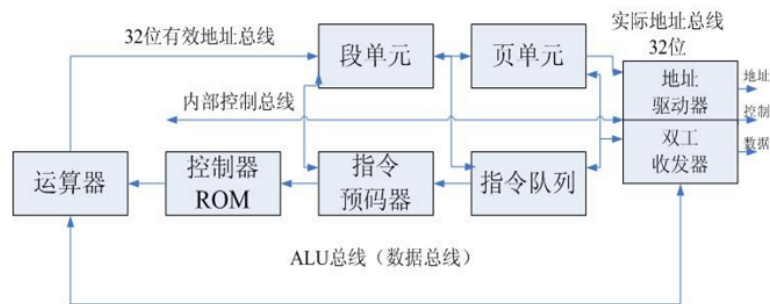


- 主机与I/O接口间的数据通路

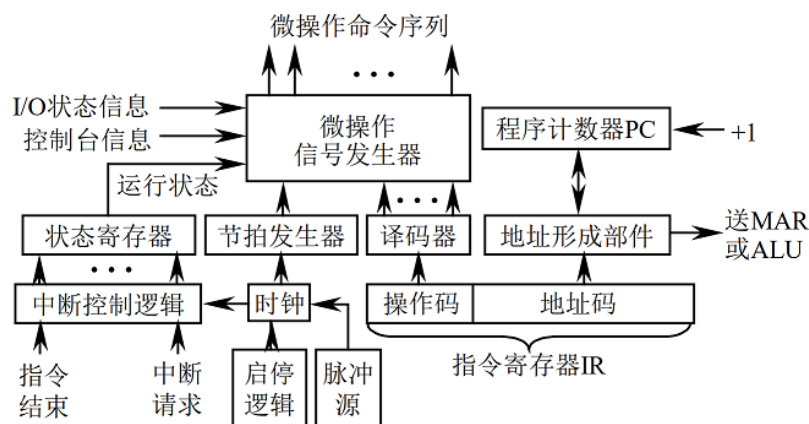
- 辐射型
- 总线型
- 通道型



- 多组内总线结构

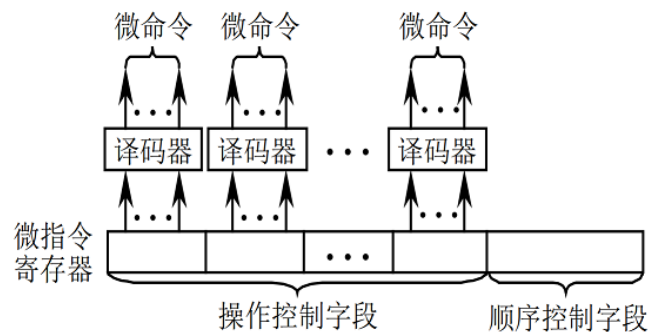


- 了解了CPU的基本组成以后，我们需要知道，其实控制器远非刚才描述的那么简单，那么控制器是怎样的呢？
 - 控制器的基本组成

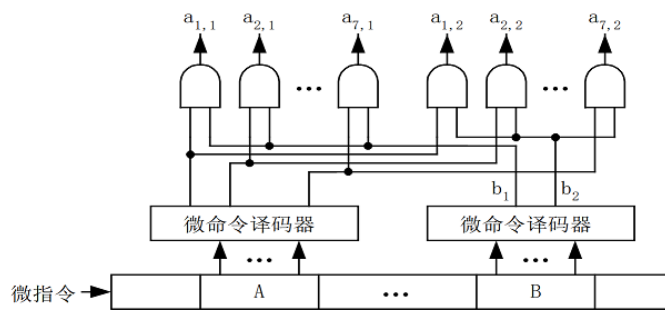


- 其中，指令部件的主要任务是完成取指令并分析指令，包括：
 - 程序计数器
 - 指令寄存器
 - 指令译码器：又称操作译码器，或者功能分析解释器，负责识别操作码是怎样的指令
 - 地址形成部件：根据指令的不同寻址方式，形成操作数的有效地址
- 硬件实现方法：控制器的核心是微操作信号发生器（控制单元CU）
 - 组合逻辑型控制器：常规控制器或硬连线控制器，速度快
 - 存储逻辑型控制器：把微操作信号代码化，使每条机器指令转化成为一段微程序并存入一个专门的存储器中，微操作控制信号由微指令产生，便于设计与扩充
 - 组合逻辑和存储逻辑结合型控制器
- 明白了硬件方面的组成，（时序系统与控制方式），那么CPU在工作过程中具体是如何运行的呢？
 - 我们从最常见的微型计算机说起，每个指令周期常采用的机器周期、节拍、工作脉冲三级时序系统。
 - 指令周期和机器周期
 - 指令周期是指：从取指令、分析取数到执行完该指令所需的全部时间。换句话说，指令周期=取值周期+取数周期+执行周期+中断周期等
 - 机器周期又分为CPU周期，由若干个时钟周期（节拍）组成
 - 节拍：
 - 统一节拍法：以最复杂的机器周期为准定出节拍数。称为定长CPU周期
 - 分散节拍法：按机器周期实际需要安排节拍数。称为不定长CPU周期
 - 延长节拍法：适当延长一两个节拍
 - 时钟周期插入法：一些微型计算机中，不设置节拍，直接使用时钟周期信号。可以在中间不断插入如等待始终周期
 - 控制方式
 - 同步控制方式：每项操作由统一的时序信号控制，可能导致简单指令剩余很多空闲时间，造成浪费

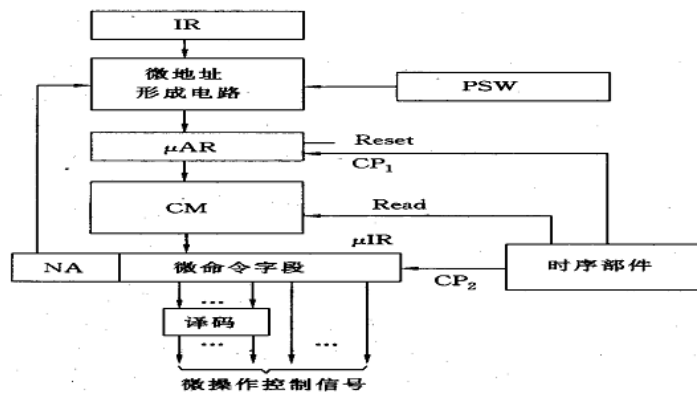
- 异步控制方式：根据实际情况占用时间，控制复杂。又称为分散控制方式、局部控制方式
 - 联合控制方式：前两者相结合
- 以上知识都涉及到了硬件问题，我们知道，操作系统以下的，就开始要运用到机器语言和微程序，此次，我们探讨微程序是如何控制的呢？（微程序控制原理）
 - 基本术语
 - 微命令和微操作：微命令是微操作的控制信号，微操作是微命令的操作过程
 - 微指令和微地址：微指令是控制存储器中的一个单元的内容，即控制字。微地址包括：垂直型和水平型（前者一条微指令一个微操作，后者并行）
 - 微周期：从控制存储器中读取一条微指令并执行相应的微命令所需的全部时间称为微周期。
 - 微程序：微指令的有序集合就是微程序。
 - 微指令编码法
 - 直接控制法：操作控制字段中的各位分别控制计算机
 - 最短编码法：若微命令总数为N，那么所需的位数是 $\log_2(N)$ ，但要通过微命令译码器译码才能得到需要的微命令
 - 字段编码法：段内采用最短编码法，段与段之间采用最短编码法
 - 字段直接编码法：译码器后直接连接对应微命令



- 字段间接编码法：译码器后多个字段的编码共同组成一个微命令



- 微程序控制器
 - 组成：
 - 控制存储器（CM）：存放微程序
 - 微指令寄存器（uIR）：存放从CM中取出的微指令
 - 微地址形成部件：产生初始微地址和后继微地址
 - 微地址寄存器（uMAR）：接收微地址形成部件送来的微地址



- 微程序入口地址的形成
 - 一级功能转换：所有操作码都用同一种编码
 - 二级功能转换：不同类型操作码用不同编码（位数不同，位置固定）
 - PLA电路实现功能转换：PLA电路将每条机器指令的操作码翻译成对应的微程序入口地址（位数、位置都不固定）
- 后继微地址的形成
 - 增量方式
 - 顺序执行，通常增量为1
 - 转移或转子：由微指令的顺序控制字段产生转移微地址
 - 断定方式：微程序设计者指定，或者由所规定的测试结果直接决定后继微地址的全部或部分值
- 学习了那么多关于CPU内部微观的结构以及工作原理，那么宏观的工作原理又是怎样的呢？（流水线技术）
 - 重叠控制：在取值、分析、执行上重叠，但是容易造成访问同一块内存
 - 先行控制：把分析、执行结合在一起，再进行重叠
 - 流水工作
 - 定义：将一个较复杂的处理过程分成m个复杂程度相当、处理时间大致相等的子过程，每个子过程由一个独立的功能部件来完成，处理对象在各子过程连成的线路上连续流动。
 - 处理级别分类：操作部件级、指令级、处理机级
 - 功能分类：单功能流水线、多功能流水线
 - 工作方式分类：静态流水线、动态流水线
 - 流水线结构：线性流水线、非线性流水线
 - 指令流水线的相关性：相邻或相近的两条指令可能会因为某种关联，导致后一条指令无法正常执行，使流水线断流
 - 结构相关：多条指令同一时刻争夺统一资源
 - 数据相关：后续指令需要用到前面指令运算的结果，但是这一结果还未产生
 - 控制相关：无法判定下一步是顺序执行还是转移执行
- 最后，了解了CPU这么多，我们知道决定CPU性能的主要技术参数有哪些吗？
 - 字长：单位时间内同时处理的二进制数据的位数
 - 内部工作频率：内频或主频
 - 外部工作频率：主板为CPU提供的基准时钟频率
 - 工作电压：CPU正常工作时所需的电压，越高越菜
 - 地址总线宽度：CPU可以访问的最大的物理地址空间
 - 数据总线宽度：决定了CPU与外部Cache、主存和输入输出设备之间进行一次数据传输的信息量

