中央处理器

CPU的功能及主要寄存器

- 功能
  - 指令控制：控制程序的顺序执行
  - 操作控制：产生完成每条指令所需的控制命令
  - 时间控制：对各种操作加以时间上的控制
  - 数据加工：对数据进行算术运算和逻辑运算
  - 处理中断
  - 寄存器：存放指令和数据
- 组成结构
  - IR 译码部件：对指令的操作码进行译码
  - CU 控制部件：发出操作命令序列
  - ALU 运算部件：完成算术运算
  - 中断系统：处理异常情况和特殊请求
  - AC 累加器
- 主要寄存器
  - 通用寄存器
  - 专用寄存器
    - PC 程序计数器：存放正在执行的指令地址或即将执行的下条指令地址
    - IR 指令寄存器：存放从存储器中取出的指令
    - MAR 存储器地址寄存器：保存当前CPU所访问的主存单元的地址
    - MDR 存储器数据寄存器：暂时存放由主存储器读出的一条指令或一个数据字；反之，当向主存存入一条指令或一个数据字时，也暂时用它开放出存储器数据寄存器。
    - PSWR 状态标志寄存器：存放程序状态字，程序状态字的各位表征程序时机器运行的状态
      - 组成
        - 状态标志
        - 控制标志
      - 8086的状态标志寄存器有16位，但只用九个标志位，其中六个为状态标志，三个为控制标志
        - 状态标志
          - CF：进位标志 Carry Flag
          - ZF：零标志 Zero Flag
          - SF：符号标志 Sign Flag
          - OF：溢出标志 Overflow Flag
          - PF：奇偶标志 Parity Flag
          - AF：辅助进位标志 Auxiliary Carry Flag
        - 控制标志
          - TF：追踪标志 Trap Flag
          - DF：方向标志 Direction Flag
          - IF：中断允许标志 Interrupt enable Flag
- 控制器的功能
  - 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
- 运算器的功能
  - 执行算术运算
  - 执行逻辑运算
  - 进行逻辑测试
- 通用
  - 指令：PC -> MAR -> M主存储器 -> MDR -> IR -> 操作控制器
  - 数据：M主存储器 -> MDR -> ALU -> AC -> MDR -> M主存储器

指令执行过程

- 三个阶段
  - 取指令
    - (PC) -> MAR：将程序计数器（PC）中的内容送至存储器地址寄存器（MAR），并送地址总线（AB）
    - Read：由控制单元（CU）经控制总线（CB）向主存发送命令
    - M(MAR) -> MDR：从主存中取出的指令通过数据总线（DB）送到存储器数据寄存器（MDR）
    - (MDR) -> IR：将MDR的内容送至指令寄存器（IR）中
    - (PC) + 1 -> PC：将PC的内容递增，为取下一条指令做好准备
  - 分析取数
  - 执行
- 周期
  - 指令周期：取指令，分析指令到执行完成指令所需的全部时间
  - CPU周期（机器周期）：通常把一条指令划分为若干个机器周期，每个机器周期完成一个基本操作
  - 指令周期 = 1 * 机器周期
  - 指令周期 = 机器周期 + 时钟周期(脉冲周期)
- 指令的微操作序列
  - 加法指令 ADD R[1],0单元
    - 取址周期
      - (PC)—>MAR
      - Read
      - M(MAR)—>MDR
      - (MDR)—>IR
      - (PC) + 1—>PC
    - 取数周期
      - (R[1]) -> MAR
      - Read
      - M(MAR) -> MDR -> Y
    - 执行周期
      - (R[1]) + Y -> Z
      - (Z) -> MDR
      - Write
      - (MDR) -> M(MAR)
  - 转移指令 JC A
    - 取址周期
      - (PC)—>MAR
      - Read
      - M(MAR)—>MDR
      - (MDR)—>IR
      - (PC) + 1—>PC
    - 执行周期
      - (PC) -> Y (C = 1)
      - A(IR) + Y -> Z (C = 1)
      - (Z) -> PC (C = 1)

控制器

- 组成
  - 指令部件
    - PC 程序计数器
    - IR 指令寄存器
    - ID 指令译码器
    - 地址形成部件
  - 时序部件
    - 脉冲源
      - 信号
        - 频率
          - 高
          - 低
          - 持续
        - 类型
          - 上升沿
          - 下降沿
          - 瞬时
    - 节拍信号发生器
    - 控序控制逻辑
  - 微操作信号发生器
    - CU 控制单元：控制器的输入是机器指令代码、输出是微操作控制信号，因此微操作信号发生器是控制器的核心
  - 中断控制逻辑
- 实现方法
  - 组合逻辑型
    - 这种控制器称为常规控制器或硬布线控制器
    - 速度快
  - 存储逻辑型
    - 这种控制器称为微程序控制器
    - 速度慢
  - 组合逻辑和存储逻辑结合型：这种控制器称为PLA控制器
- 控制方式
  - 同步控制方式：又称为全一种控制方式或中央控制方式
  - 异步控制方式：又称为分散控制方式或局部控制方式
  - 联合控制方式：同步控制和异步控制相结合的方式

微程序控制器基本概念

- 微命令和微操作
  - 微命令是控制计算机各部件完成某个基本微操作的命令
  - 微操作是计算机中最基本的、不可再分解的操作
  - 微命令是微操作的控制信号，微操作是微命令的操作过程
  - 微命令和微操作是互为表之分
- 微指令、微地址
  - 微指令是控制序列器中的一个单元的内容，即控制字，它是若干个微命令的集合
  - 存放控制字的控制存储器的单元地址称为微地址
  - 微指令格式：操作控制字段 + 顺序控制字段
    - 操作控制字段：该步操作所需的各种操作控制信号
    - 顺序控制字段：下一条要执行的微指令地址
  - 类型
    - 垂直型
      - 单行性
    - 水平型
      - 并行性
- 微周期：从控制存储器中读取一条微指令开始执行完这条微命令所需的全部时间称为微周期
- 微程序：一条机器指令对应于一段微程序
- 微命令 -> 微指令 -> 微令(微程序) -> 程序
  - 微程序存放在控制存储器CM中
  - 程序存放在主存或磁存中
- 微程序控制器的组成
  - CM 控制存储器：核心部件，用来存放微程序
  - uIR 微指令寄存器：存放从CM取出的正在执行的微指令
  - 微地址形成部件：产生初始微地址和后继微地址
  - uMAR 微地址寄存器：接受微地址形成部件送来的微地址，为在CM中查取微指令作准备

微指令编码法

- 直接控制法(不译码法)
  - 结构简单、并行性强、操作速度快
  - 微指令字长长
- 最短编码法
  - 微命令的总数为N，操作控制字段的长度为L，则：2^L >= N
  - 微指令字长最短
  - 译码最复杂
  - 不能充分利用控制器硬件资源的并行性
- 字段编码法
  - 将操作控制字段分为若干个小段，每段内采用最短编码法，段与段之间采用直接控制法
  - 字段直接编码法：在完全分散在第一字段
  - 字段间接编码法
    - 静态编码
    - 动态编码：设置解释位或解释字段
    - 如：解释位C：A
  - 分段原则
    - 在同时使用微命令分在同一段内，两段性的微命令分在不同段内
    - 每个小段还要留出一个空位状态和第一位空位，表示本字段不发出任何微命令

指令流水法

- 概念
  - 把指令的处理过程分为几个子过程，如取指令、指令译码、取操作数和执行
  - 每个子过程由独立的功能部件完成
  - 在同一时间段，由不同的功能部件执行不同指令的不同子过程
- 实现

  - 如子过程数 m = 4，任务数 n = 4
  - 执行一条指令的时间为m个时间 = m * t1 = 4 * t
- 流水线瓶颈
  - 最长小段
    - IF 取指令
    - ID 指令译码/读寄存器
    - EX 执行/计算有效地址计算
    - MEM 存储器访问
    - WB 结果写回寄存器
  - 结构相关（资源相关）
    - 不同指令同一时刻争用同一功能部件
    - 解决访存冲突方案：停顿
  - 数据相关
    - 流水线中的指令访问数据操作，改变了对操作数的读写访问顺序
    - 解决数据冲突方案：前推法
  - 控制相关（指令相关）
    - 由转移指令引起的
- 超标量流水线
  - 流水线技术
    - 理想的流水线，每个周期平均执行的指令条数小于等于1
    - 指令吞吞行度（ILP）≤1
  - 超标量流水线是通过重复设置多个功能部件
  - 假设每个时钟周期发射n条指令，则有 1< ILP < n