

计算机硬件技术基础简明教程

计算机系统基本组成原理

冯诺依曼结构特点：

- 1.由运算器、控制器、存储器、输入设备、输出设备组成；
- 2.数据和程序以二进制代码不加区分地存放在存储器中，存放位置由二进制地址码指定；
- 3.控制器根据存储器中指令序列（程序）工作，由指令地址计数器控制指令的执行，控制器有判断能力；

Pentium 系列 PC 硬件组织结构：南北桥结构、中心结构；

微机系统组成部分：

- 1.微处理器（MPU）
包含控制器、运算器、内部总线与缓冲器；
- 2.内存储器
存放必需的程序和数据；
- 3.I/O 设备接口
- 4.总线
数据总线：传续数据信息，双向；
地址总线：传送 CPU 发送的地址信息，单向；
控制总线：传送控制信号、时序信号、状态信息等，整体双向；

机器数：符号位一同数字化的数字；

有符号数：

计算机采用不同码制时，运算器和控制器结构不同；

原码： $-(2^{n-1}-1) \sim (2^{n-1}-1)$

$$[X_o] = \begin{cases} 0X_1X_2 \cdots X_{n-1}, X \geq 0 \\ 1X_1X_2 \cdots X_{n-1}, X < 0 \end{cases}$$

反码：负数除符号位外按位取反；

$-(2^{n-1}-1) \sim (2^{n-1}-1)$

$$[X_o] = \begin{cases} 0X_1X_2 \cdots X_{n-1}, X \geq 0 \\ 1\overline{X_1}\overline{X_2} \cdots \overline{X_{n-1}}, X < 0 \end{cases}$$

补码：负数的原码除符号位外，各位按位取反加 1；

$-2^{n-1} \sim (2^{n-1}-1)$

$$[X_o] = \begin{cases} 0X_1X_2 \cdots X_{n-1}, X \geq 0 \\ 1\overline{X_1}\overline{X_2} \cdots \overline{X_{n-1}} + 1, X < 0 \end{cases}$$

$$X_o = [[X_o]_c]_c$$

$$X_c \pm Y_c = [X_o \pm Y_o]_c$$

定点数：小数点位置固定的表示方法；

浮点数：

$$J_f | J | S_f | S$$

S 最高位为 1；

无符号数：

- 1.位数不等的二进制码；
- 2.BCD 码；
压缩 BCD：每位 BCD 用 4 位二进制表示；
非压缩 BCD：每位 BCD 用 1 个字节表示，高字节为 0000B；
- 3.ASCII 码；

计算机工作过程：

1.取指令

根据 PC 中的值从存储器读出现行指令；

将指令送到 IR；

PC 自动加 1；

2.分析指令

将 IR 中的指令操作码译码；

3.执行指令

取出操作数，执行操作；

字长：计算机内部一次可处理的二进制数码位数；

运算速度：每秒钟能执行的指令条数；

微处理器

中国芯片：

方舟芯片：北京方舟科技公司；

龙芯芯片：中科院计算机所；

星光芯片：中星微电子公司；

微处理器中计算机技术：

流水线技术：将每条指令分解为多个步骤，实现多指令并行处理；

不改变指令执行三阶段；

高速缓冲存储器技术：CPU 与内存间存储空间较小，存取速度高的存储器；

虚拟存储器技术：在内存和外存间建立的虚拟存储器；

虚地址空间：程序可用空间；

实地址空间：CPU 可访问内存空间；

多媒体扩展技术：一条指令同时完成对多重数据的计算；

动态执行技术：分析程序的数据流，选择指令执行顺序；

8086/8088 处理器结构特点：

指令流水线：EU 与 BIU 为两个独立功能部件，并行工作；

存储器分段管理：将 1MB 物理存储空间分为若干逻辑段，由段基址和段内偏移地址表示；

Pentium 微处理器：

内部寄存器：

通用寄存器：

EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP

扩展为 32 位，可拆解为 16 位/8 位访问；

指令指针寄存器：

EIP：保存下一条待预取指令相对于代码段基址的偏移量；

标志寄存器：

EFLAGS：32 位，对 PSW 扩展；

段寄存器：

CS, SS, DS, ES, and extra FS, GS

工作方式：

1.实地址方式

2.保护虚拟地址方式

3.虚拟 8086 方式

4.系统管理方式

操作数寻址方式：

1.立即数寻址：出现在指令中的立即数作为一个操作数用；

2.寄存器寻址：操作数存放于寄存器中；

3.直接寻址：操作数存放于存储器中，有效地址由指令直接给出；

4.寄存器间接寻址：操作数存放于存储器中，有效地址存放在规定寄存器中；

16 位：SI, DI, BP, BX; (BP 默认 SS 为段基址，其他默认 DS)

32 位：均可；(ESP, EBP 默认 SS，其他默认 DS)

5.基址寻址：

$$EA = [Base\ reg] + disp$$

位移量为常数，跟随在操作码后；

BX 以 DS 为默认段寄存器，BP 以 SS 为默认段寄存器；

6.变址寻址：

$$EA = [Chg\ reg] + disp$$

仅 SI, DI 可作为变址寄存器，DS 默认为段基址寄存器；

7.比例变址寻址：

$$EA = [Chg\ reg] \times ratio + disp$$

只适用于 32 位寻址；

8.基址加变址寻址：

$$EA = [Base\ reg] + [Chg\ reg]$$

基址、变址寄存器规定与之前相同；

9.基址加比例变址寻址：

$$EA = [Chg\ reg] \times ratio + [Base\ reg]$$

仅有 32 位寻址情况；

10.带位移的基址加变址寻址：

$$EA = [Chg\ reg] + [Base\ reg] + disp$$

基址、变址寄存器规定与之前相同；

11.带位移的基址加比例变址寻址：

$$EA = [Chg\ reg] \times ratio + [Base\ reg] + disp$$

仅有 32 位寻址情况；

Pentium 指令系统与汇编语言编程

汇编语言开发过程：

- 1.编辑源程序；
- 2.汇编源程序；
- 3.连接目标程序；
- 4.调试并执行程序；

程序基本结构形式：顺序、分支、循环；

分支程序结构形式：不完全、完全、多分支结构；

循环结构程序部分：循环体、循环修改、循环控制；

DOS 功能调用：

01H：输入一个字符，ASCII 码送入 AL，同时显示在屏上；

02H：将 DL 中字符输出（ASCII）；

08H：不带显示的 01H；

09H：将一个以 '\$' 结束的字符串输出至显示器，DS：DX 指向字符串首地址；

0AH：键盘输入的字符串存入缓冲区，同时显示字符串，enter 结束；

DS：DX 指向缓冲区首地址，第一字节为输入字符长度，第二字节为输入字符数；

存储器

MOS 型存储器分类：只读存储器（ROM）、随机存取存储器（RAM），顺序存取存储器（SAM）；

存储器芯片基本组成：

- 1.存储体；
- 2.地址寄存/译码电路；
- 3.数据寄存器；
- 4.控制逻辑；

存储器性能指标：

存储容量、存取速度、可靠性、功耗；

译码方法：

- 1.线选法：高位地址线中每一根单独直接控制一个芯片的片选；
- 2.局部译码法：对高位地址总线中的一部分进行译码，产生片选控制信号；
- 3.全译码法：将全部高位地址总线进行译码；

I/O 接口与总线技术

接口电路基本结构：I/O 数据缓冲存储器、寄存器地址译码器、I/O 控制逻辑；

端口编址方式：

- 1.存储器映像编址方式：
 - 对 I/O 操作与对存储器操作相同，接口程序设计方便、灵活；
 - 微机系统只需要一套 I/O 控制逻辑，简化逻辑设计；
 - 占用部分存储器地址空间，可用内存空间减少；
 - 译码电路复杂，寻址操作时间长；
- 2.隔离 I/O 编址方式：
 - I/O 端口地址不占用地址空间；
 - I/O 端口地址译码器简单，寻址速度快；
 - 程序编写结构清晰；
 - 程序设计灵活性差；
 - 控制逻辑复杂性增加；

I/O 同步控制方式：

- 1.程序查询式：
 - I/O 操作由 MPU 主动引发，I/O 只提供准备就绪状态，被动接收查询；
 - 所有 I/O 传送与程序的执行严格同步，数据传送可靠；
 - MPU 循环等待，使用效率低；
- 2.中断驱动式：
 - I/O 设备与 MPU 间数据传送由 MPU 响应 I/O 设备发出的中断请求实现；
 - 适于计算机任务繁忙，I/O 处理实时性要求高的系统；
 - 软件开发与调试较复杂；
- 3.直接存储器存取式：
 - I/O 设备与存储器直接交换信息；
 - 数据传送速度快，I/O 响应时间短，MPU 额外开销小；
 - 系统硬件复杂度增加，系统代价提高；
- 4.延时等待式：
 - 只适用于外部控制过程总时间固定、已知情况；

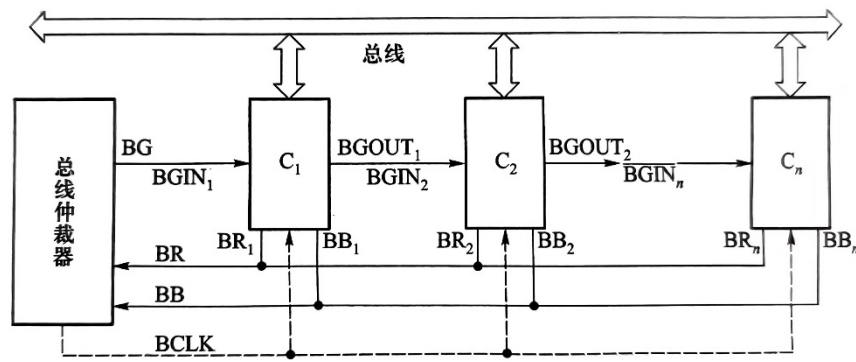
总线性能指标：带宽、位宽、工作频率；

总线操作周期：

- 1.总线请求与仲裁阶段；
- 2.寻址阶段；
- 3.传输阶段；
- 4.结束阶段；

总线仲裁控制：

1.菊花链仲裁:



仅使用 3 根控制线，易于扩充；
BG 信号需逐级传递，响应慢；
线路连好后，优先级无法改变；
能容纳设备数量受时钟频率限制；

2.并行仲裁:

每个主控器有自己独立的 BR, BG 线与仲裁器相连；
避免总线请求、总线允许信号传输延迟，响应快；
控制信号多，逻辑复杂，不易扩充；

3.并串行仲裁:

将主控设备分组，组间并，组内串；

总线握手控制:

1.同步总线握手:

主从模块间数据交换只受一个时钟源控制；
电路设计简单，总线传输速度快；
只能按最坏可能确定总线频带宽度，适应性差；

2.异步总线握手:

I/O，主控器等待受控器应答信号，结束总线动作；
数据传输可靠，适应性高；
传输速度慢。总线频带窄，总线周期长；

3.半同步总线握手:

通过 WAIT, READY 信号线延长总线周期；

中断与 DMA

中断： 外部： 非屏蔽中断（NMI）
可屏蔽中断
内部异常：失效、自陷、终止；

中断响应条件:

- 1.中断请求触发器置位；
- 2.中断屏蔽触发器处于非屏蔽状态；
- 3.MPU 内部中断开放；
- 4.无更高优先级中断请求正在发出/挂起/被响应；

多中断源服务判决:

1.程序查询式:

硬件简单，程序层次分明，易改变中断优先级；
中断源较多时，响应中断到进入 I/O 中断服务程序时间较长；

2.中断向量式:

中断响应速度较快，处理器利用率较高；

中断处理流程:

1.保护断点现场:

- (1) CPU 的标志寄存器内容，清除 IF 与 TF；
- (2) 将 CS, IP 压入堆栈；

- 2.中断服务：
 核心；
- 3.恢复现场：
 将断点信息从堆栈弹出；
- 4.中断返回：
- 5.开/关中断

DMAC 工作方式：

- 1.被动方式：作为系统中一个可编程 I/O 接口；
- 2.主动方式：作为总线主控器工作，独立于 MPU，数据传输由 DMAC 硬件控制完成；
 顺序传送：传送一个数据的 I/O 操作分两个 DMA 周期顺序进行；
 同时传送：传送一个数据的 I/O 操作在一个 DMA 周期同时进行；

DMA 操作过程：

- 1.DMAC 请求阶段；
- 2.DMAC 响应和传输阶段；
- 3.传送结束阶段；

微机系统中常用可编程接口芯片