

# 目录

# 1 绪论

## 2 单核处理器 8086/8088

单核处理器是多核处理器的基础

从 8086 开始 Intel CPU 设计采用了向后兼容，也称向下兼容  
单核的 8086 CPU 成为其后 Intel CPU 的基石

### 2.1 8086/8088 处理器功能特性

1. 第一次将流水线思想引进微处理器：指令级流水
2. 存储器分段管理机制引入处理器，扩大寻址能力
3. 只有整数运算指令。配套数值协处理器 8087、输入输出协处理器 8089，具备较强大的计算能力和 I/O 处理能力

8086 CPU 有三个版本 8086、8086-2、8086-1。仅时钟频率不同，依次为 5Mhz、8Mhz、10Mhz。具有以下功能特性：

1. 直接主存寻址能力 1MB
2. 体系结构针对强大的汇编语言和有效的高级语言设计
3. 14 个 16 位寄存器
4. 24 种操作数寻址方式
5. 操作数类型：位、字节、字和块
6. 8、16 位无符号和带符号二进制和十进制运算，包括乘除

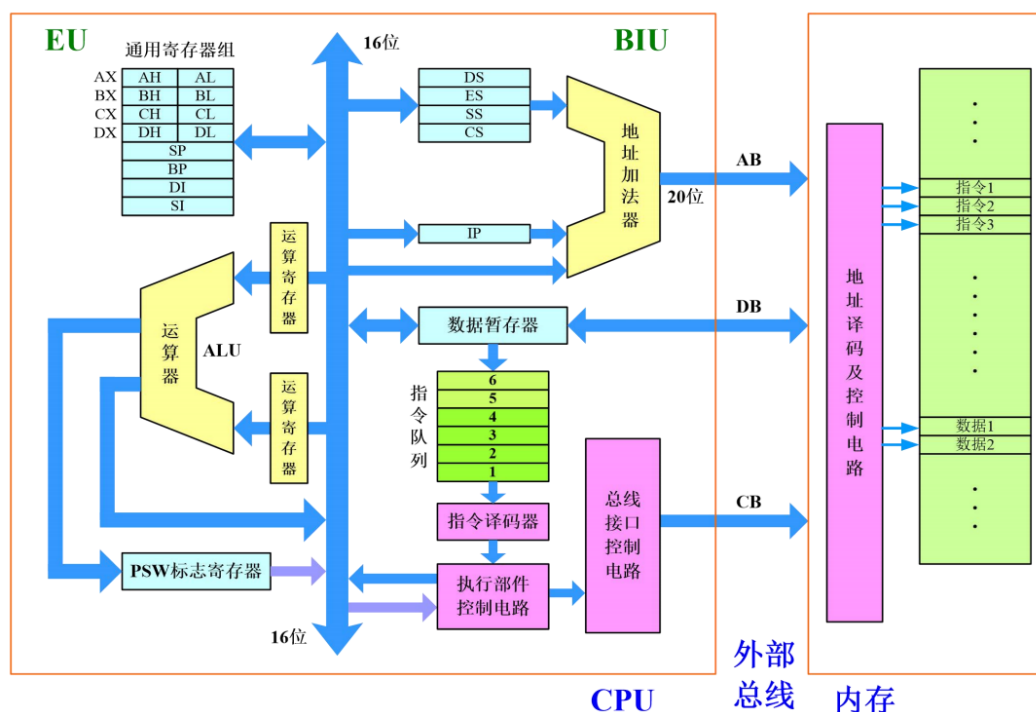


图 1: 8086 处理器体系结构

右侧的 BIU 是总线接口单元 (Bus Interface Unit) 负责与存储器、I/O 接口传递数据，具体完成：

1. 从内存中取指令、送到指令队列
2. 配合 EU 从指定的内存单元或 I/O 端口取数据
3. 将 EU 的操作结果送到指定的内存单元或 I/O 端口

左侧执行单元 EU 负责指令的执行 (算术、逻辑、移位运算，有效地址计算，控制命令、……)，包括 ALU，通用寄存器组和状态寄存器，进行 8/16 位运算。

取指令和执行指令在时间上重叠，BIU 在指令队列缓冲器有 2 个以上空字节时就不断从主存连续地址单元中取得指令送入指令队列缓冲器中，EU 则不断从指令队列中取出指令译码执行。仅当一下情况发生时，两者并行工作的状态打破：

1. 6 个字节的指令队列缓冲器满，且 EU 没有主存或 I/O 访问请求时，BIU 进入空闲状态
2. EU 执行访存或 I/O 指令时，需要对主存或 I/O 设备进行读写数据操作，则在 BIU 执行取指周期后，暂停取指操作，在下一总线执行 EU 所要求的主存或 I/O 读写操作，之后再继续 BIU 的取指操作。
3. 在 EU 执行转移、调用、返回等程序跳转指令时，因之前读入指令队列缓冲器的指令已无效，则 BIU 在清空缓冲器的同时，根据 EU 提供的跳转地址，重新获取跳转后的程序指令。

工作原理：

1. DB 线中取数据到数据暂存器里
2. 指令队列有 2+ 空字节，则 BIU 自动取指，放到指令队列中
3. EU 总是从队列前部取指令执行
4. 指令要访问 M 或 I/O，EU 会请求 BIU 完成

优点：实现了预取指令，取指令和执行指令可以并行，加速程序运行。由于有指令队列，BIU 和 EU 可以并行工作，一边取指令、一边执行指令，构成指令级流水线。

## 2.2 寄存器、主存和 I/O 结构

### 2.2.1 寄存器

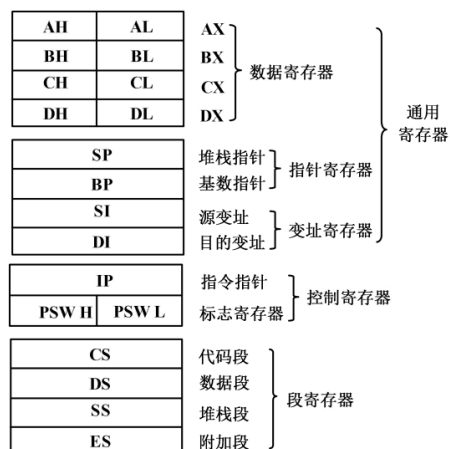


图 2: 寄存器结构

首先是数据寄存器：

1. 通用寄存器组均为 16 位，但是可以拆分成高位和低位转化成 8 位
2. AX 用于字乘法、字除法和字节乘法、字节除法以及字 I/O
3. BX 有时作为地址寄存器保存段内偏移地址
4. CX 有时作为隐含计数器串操作计数循环计数 CL 移位次数
5. DX 有时作为地址寄存器保存 I/O 地址

指针寄存器

1. SP 堆栈指针寄存器，用于存放主存堆栈区的偏移地址，指示当前操作位置
2. BP 基数指针寄存器，用于存放主存的基本偏移地址

变址寄存器

1. SI 源变址寄存器指向源操作数

2. DI 目的变址寄存器指向目的操作数
3. SI、DI 具有自动修改内容的功能

### 控制寄存器

1. IP 指令指针寄存器，用来指示指令所在存储单元的段内偏移地址。系统将欲执行程序的首地址加载至 CS 和 IP 中，转移类指令用跳转的目标地址修改 IP（或 CS 和 IP）。当以及 CS 和 IP 从主存取指令后 IP 自动加一，指向下一个读取的指令字节
2. PSW 是程序状态字，也称为状态寄存器或标志寄存器格式如下



图 3: 状态标志位的值

1. PSW 为 16 位，暂定 9 个标志位
2. C——进位标志。加减运算若在最高位出现进位或错位，则标志为 1，否则清 0。逻辑运算、位移和循环指令也影响该标志位
3. P——奇偶标志位。低 8 位中 1 的个数位偶数时，标志位置位 1，否则清 0
4. A——半加进位标志位。加减运算时，低四位向高思维进位或者借位，则置 1，否则清 0。用于 BCD 运算结果矫正
5. Z——零标志位。运算结果全为 0，则为 1，否则清 0
6. T——陷阱标志位（单步标志位）。为 1 则进入单步执行指令工作方式，产生单步中断，CPU 执行陷阱中断程序。将执行结果显示（寄存器、存储单元和 I/O 接口）
7. I——中断允许标志位。为 1 则可屏蔽中断请求，为 0 则不可
8. D——方向标志位。为 1 则 SI 和 DI 在串操作指令执行中自动减即从高到低，否则 SI 和 DI 自动增量。

9. O——溢出标志位。当带符号数运算超出 8 位或 16 位表示范围

#### 段寄存器

由于 8086 的设计，主存是按段划分，分为数据段、堆栈段、代码（程序）段、附加段。

1. 代码段寄存器 CS 指示程序区
2. 数据段寄存器 DS 指示数据区
3. 堆栈段寄存器 SS 指示堆栈区
4. 附加段寄存器 ES 指示数据区

1. SS:SP  $\rightarrow$  堆栈地址
2. CS:IP  $\rightarrow$  主存地址类似 PC
3. 首先找段地址寄存器 + 段内偏移地址
4. 地址加法器只做地址加法，输出为 I/O、存储器地址

## 2.2.2 主存结构

双体结构，既可以实现 16 位存储，也可以实现 8 位存储

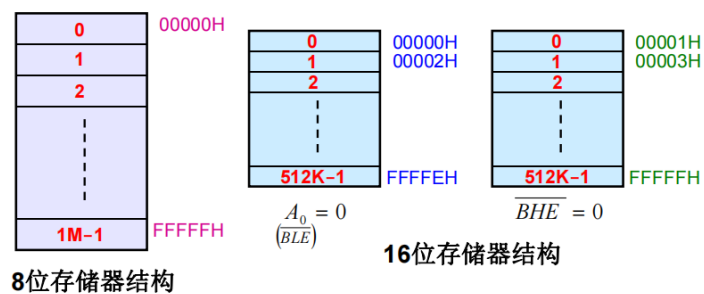


图 4: 双体结构

分段结构

1. 代码、数据量不大-> 使其处于同一段内 (64KB 范围内) -> 减少指令长度、提高运行速度
2. 内存分段为程序的浮动分配创造条件
3. 形成地址 6832H:1280H-> 物理地址 = 68320H+1280
4. 各个分段之间可以重叠

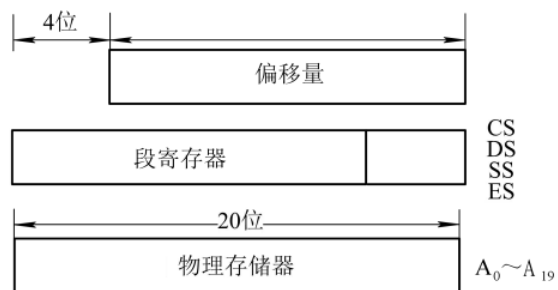


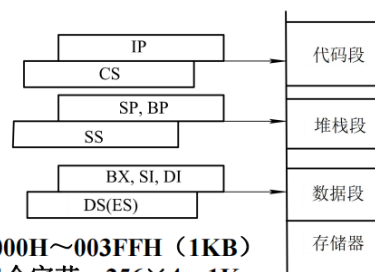
图 5: 物理地址形成



## 2. 主存结构

### ● 分段结构：

#### ◆ 段寄存器的使用



#### ◆ 特殊的主存区域

- 中断向量区：00000H~003FFH (1KB)  
每个中断向量占4个字节， $256 \times 4 = 1K$
- 显示缓冲区：  
B0000H~B0F9FH ( $25 \times 80 \times 2 = 4000$  字节)  
B8000H~BFFFFH (32KB)
- 启动区：FFFF0H~FFFFFH (16个字节)  
无条件转移指令

图 6: 分段结构

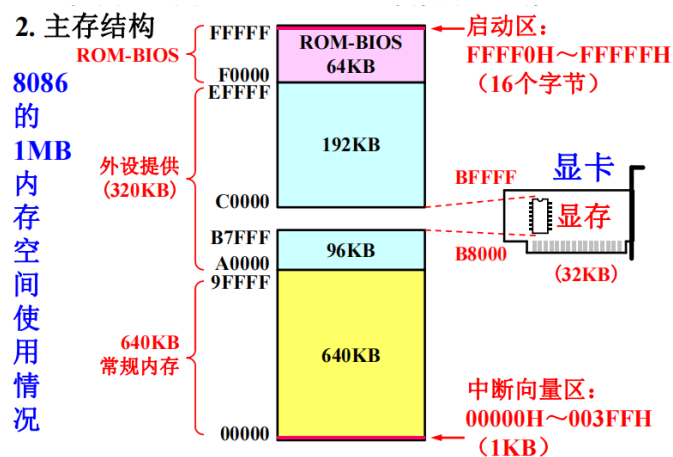


图 7: 主存结构

### 2.2.3 I/O 地址空间

1. I/O 地址空间独立于主存地址空间，两者采用不同的读写信号进行访问控制
2. I/O 地址空间包含 64k 个可单独寻址的 8 位 I/O 端口，编号 0 到 FFFFH，其中 I/O 端口地址 0F8H 0FFH 被保留

3. 16 位系统 I/O 也按地址的奇偶分为个体

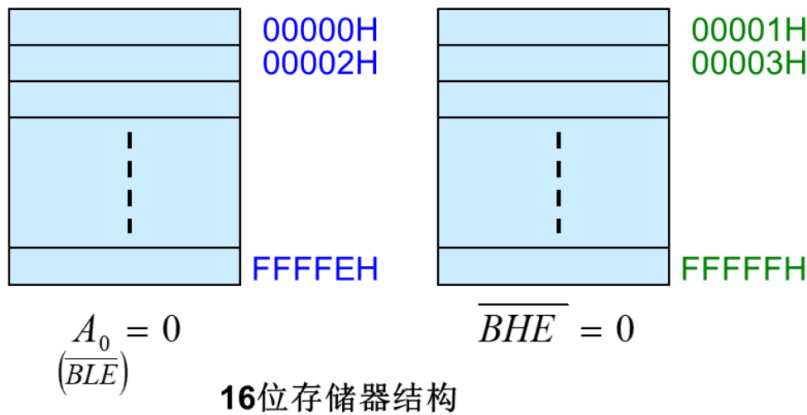


图 8: I/O 结构

2.2.4 处理器芯片引脚

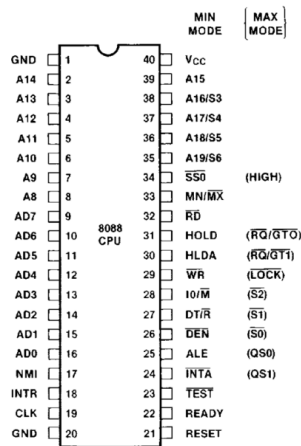


图 9: 8086 引脚图

8086 与 8088 对比:

- 1. 内部指令预期队列 6 字节 →4 字节

2. 8088: AD7 AD0 8086 AD15 AD0 (速度快)
3. 8088  $\bar{S}0$  8086  $\bar{B}E/S7$
4. 8088 I/O/ $\bar{M}$  8086  $\bar{M}/I/O$