

微机原理笔记

计算机系统概论

计算机五大部件

运算器、控制器、存储器、输入设备、输出设备

现代计算机的发展

- 第一代：电子管计算机
- 第二代：晶体管计算机
- 第三代：集成电路计算机
- 第四代：大规模和超大规模集成电路计算机

微型计算机的层次

微型计算机系统 $\left\{ \begin{array}{l} \text{微型计算机} \left\{ \begin{array}{l} \text{微处理器} \\ \text{总线、存储器(RAM、ROM)、输入输出设备} \end{array} \right. \\ \text{外围设备、电源、辅助电路、软件} \end{array} \right.$

微处理器概述

4个基本功能: **程序控制、操作控制、时间控制、数据加工**

程序由指令构成, 指令由操作码、操作数构成

例如:

```
mov AX, 1234H;
```

mov 是操作码, AX是目的操作数, 1234H是源操作数

性能指标

- 字长: 由数据总线控制, 计算机一次传送或处理的二进制位数。
(字节: 通用基本单元的长度, 由8个二进制位 (bit) 组成)
- 指令数: *CISC*(复杂指令计算机系统)、*RISC*(精简指令系统计算机)
- 指令执行时间
基本指令指令执行时间 = 寄存器加法指令执行时间
平均指令执行时间 = $\sum \text{所有指令执行时间} / \text{所有指令数}$
- 最大存储空间: 8086有20根地址线, 最大存储空间为1MB

8086是16位计算机, 即一次可以处理两个字节 (16位二进制数)。

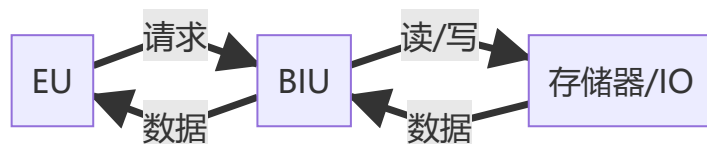
8086的寄存器是16位的, 一个存储单元8位, 存放的数据用两位十六进制数表示/8位二进制数表示。

8086在一个总线周期内可以同时访问两个存储单元

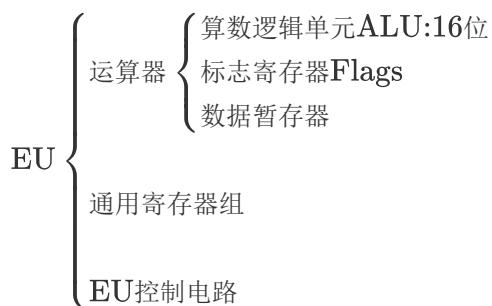
内部结构

- *BIU*: 执行部件
- *EU*: 总线接口单元

EU只负责分析和执行指令，不和外部总线打交道。BIU是专门和总线打交道的部件。EU从BIU读取指令、执行指令。执行结果数据或者执行所需数据，由EU向BIU发出请求，BIU根据EU的请求对IO口或者存储器写或读。



EU由运算器、通用寄存器组、EU控制电路组成：



ALU的两个功能：

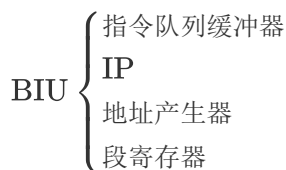
- 进行所有的算术运算和逻辑运算
- 产生偏移地址，在BIU的地址产生器中与CS/DS/SS/ES*16相加，产生物理地址

通用寄存器组：

包含八个16位寄存器

- 四个数据寄存器：AX、BX、CX、DX
- 四个地址指针变化寄存器：SP、BP；SI、DI

BIU由指令队列缓冲器、IP(指令指针寄存器)、地址产生器、段寄存器组成



指令队列缓冲器

- 遵循FIFO，由六个8位寄存器组成，最多存放6个字节指令（因为6个八位寄存器）
- EU从指令队列缓冲器中读取指令，当队列中只要出现一个空字节时，BIU立即读取指令补上。一段程序执行完后(发生跳转)，会清空队列缓冲器，填入新地址的指令
- 特点：取指令与执行指令重叠并行

IP

- 保存EU下一条指令的地址
- 自动修正

地址产生器Σ

为什么需要地址产生器？因为8086有20根地址线，最大可寻址范围为1MB，但只有16根地址线，CPU内部存放地址信息的寄存器只有16位，所以需要产生20位物理地址的单元。这个单元就是Σ。要产生一个数据的物理地址，先由ALU计算出偏移地址，然后在地址产生器中与段寄存器中的段地址*16相加（左移54位），得到物理地址。

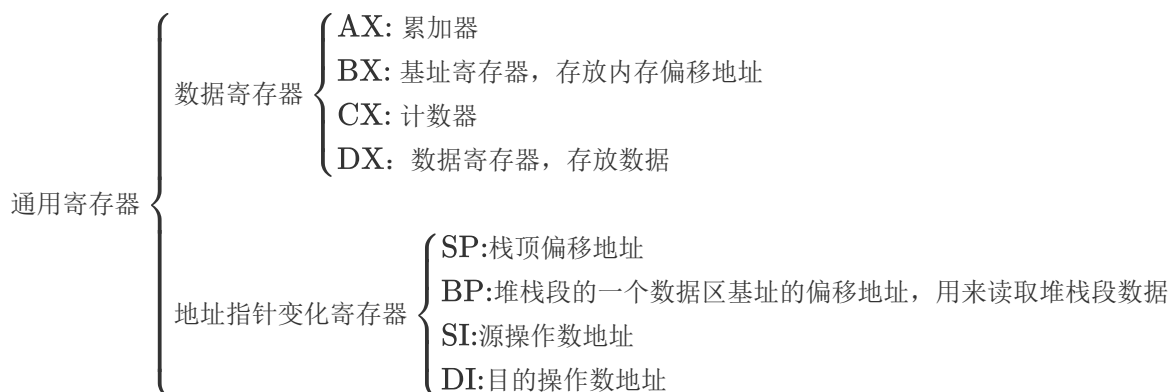
段寄存器

- DS：数据段
- CS：代码段
- SS：堆栈段
- ES：附加段

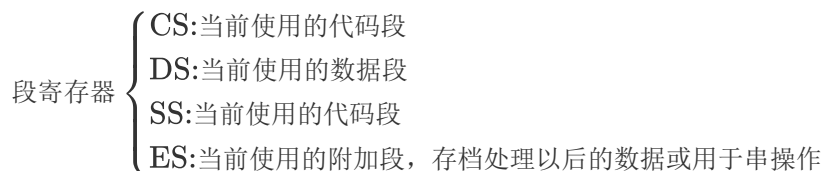
8086寄存器结构

8086有14个寄存器，8个通用寄存器，4个段寄存器，2个控制寄存器

通用寄存器



段寄存器



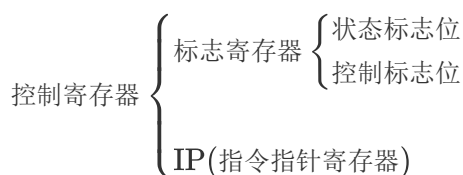
SS:SP / BP

CS:IP

DS:BX / SI / DI

ES:SI / DI

控制寄存器



状态标志位(6个)

- CF：进位/借位标志，运算结果最高位上出现进位/借位时为1，Carry

- AF: 低8位的最高位产生了进位/借位, Auxiliary (辅助的)
- OF: 溢出时为1, overflow
- ZF: 运算结果为零时为1, zero
- SF: 带符号运算结果为负数(最高位为1)时为1, sign
- PF: 运算结果有偶数个1时(结果为偶数)为1, Parity

控制标志位(3个)

- IF: IF=1, 允许中断, Interrupt
- DF: DF=1, 以地址递减的顺序对数据串处理, Direction
- TF: TF=1, 8086单步执行指令(debug), Trap

溢出只针对有符号数, 进位只针对运算过程。对于有符号数, 不存在溢出; 对于无符号数, 运算过程存在进位

8位二进制数的范围为:

有符号数: -128 ~ +127

无符号数: 0 ~ 255

进位标志位表示无符号数运算结果是否超出范围(0 ~ 255), 结果仍然正确。

溢出标志位表示有符号数运算结果是否超出范围(-128 ~ 127), 结果不正确

例: FFH + 01H

- FFH和01H为无符号数

$$1111\ 1111 + 1 = 1\ 0000\ 0000$$

有进位, 结果为0
- FFH和01H是有符号数

则FFH和01H都是补码, 先转化为原码

FFH => 81H (-1), 01H => 01H (1)

对于运算结果: -1+1 = 0, 无溢出

对于运算过程: 1111 1111 + 0000 0001 = 1 0000 0000, 有进位, 结果为0

PS: 运算过程要用补码进行计算, 不能用原码。

8086存储器结构体系

存储器的组织结构

8086的1MB空间使用情况

地址	使用状况
F0000H - FFFFFH	ROM
C8000H - EFFFFH	外设
A0000H - C7FFFH	显示器
00400H - 9FFFFH	内存 (640KB)
00000H - 003FFFH	中断向量

FFFF0H为BIOS入口(复位), 复位时, CS:IP为FFFFH:0000H

8086存放字数据时采用小端方式：低字节存放低地址，高字节存放高地址。低字节存放在偶数地址为规则存放，读/写一个字（16位）需要一个总线周期。低字节存放在高地址为不规则存放，读/写一个字（16位）需要两个总线周期。存放地址数据，低地址存放偏移地址，高地址存放段地址。

8086在组织1MB时，将存储器分为高位和低位两个部分，各512KB，分别叫做高位库和低位库。高位库和数据线 $AD_{15} - AD_8$ 相连，低位库和 $AD_7 - AD_0$ 相连。控制线 \overline{BHE} 和地址线 A_0 用于库的选择，对8086数据传输的作用如下：

\overline{BHE}	A_0	用到的引脚	操作
1	0	$AD_7 \sim AD_0$	字节，偶地址（从偶地址读/写一个字节）
0	1	$AD_{15} \sim AD_8$	字节，奇地址
0	0	$AD_{15} \sim AD_0$	字，偶地址
1	1	\	无效

A_0 是地址的最低位， $A_0 = 0$ ，地址是偶数，自然是从偶地址读写， $A_0 = 1$ ，地址是奇数，自然是从奇地址读写。 \overline{BHE} （Bus High Enable），总线高位使能，低电平有效。

从奇地址读写一个字需要两个总线周期，所以 \overline{BHE} 和 A_0 要变化一次。第一个总线周期 $\overline{BHE} = 0$ 、 $A_0 = 1$ ，用总线的高八位去读/写数据的低八位，第二个总线周期 $\overline{BHE} = 1$ 、 $A_0 = 0$ ，用总线的低八位去读/写数据的高八位。这是一个字节交换过程，对数据来说，依然是第一次读/写低八位，第二次读/写高八位。

地址分段管理

为什么需要分段管理？因为8086有20根地址线，最大可寻址范围为1MB，但只有16根地址线，CPU内部存放地址信息的寄存器只有16位，不能存放物理地址PA，所以需要引入分段管理。

分段管理就是把内存空间分成若干份，段首地址能被16整除（二进制的最后四位为0），每份64KB，地址段内的存储单元（1Byte）相对于段首的地址偏移量为偏移地址。

物理地址可由下面公式计算出

$$\text{物理地址(PA)} = \text{段地址} \times 16(\text{左移4位}) + \text{偏移地址}$$

物理地址用五位十六进制数（20位二进制数）表示，**逻辑地址用 段地址(4位):偏移地址(4位)** 表示。

因为段地址二进制的最后四位为0（十六进制最后一位为0），所以1MB空间可以分为 2^{16} （ 16^4 ，65536）个段。每个段可以重叠也可以分离，一个存储单元可以有不同的逻辑地址，但只有一个物理地址。在指令中只能使用逻辑地址，不能使用物理地址。

堆栈

一段特殊的存储单元，遵循 FILO (先进后出)

目的(作用)：存放不用或暂时需要保护的数据。

双端RAM、指令队列缓冲器遵循 FIFO (先进先出)。

8086中，堆栈必须按字操作(16位)，压栈先压高8位。

- eg:

push AL;	<input type="checkbox"/>
push AX;	<input checked="" type="checkbox"/>

堆栈段中存取数据的地址 ($SS : SP$) 由堆栈段寄存器 SS 和堆栈指针寄存器 SP 规定：

- SS 提供段首地址

- SP 提供偏移地址，存放栈顶地址

初始化时， SP 指向栈底 +1

1. 压栈

```
push AX;
```

(1) $SP \leftarrow (SP) - 2$

(2) 压栈

例：push AX； 其中AX = 1234H，SP = 1200H

1. $(SP) = (SP) - 2$ ； (SP) 变为 11FEH

2. 压栈

$((SP)) = (11FEH) = 34H$

2. 出栈

```
pop AX;
```

(1) 出栈

(2) $SP \leftarrow (SP) + 2$

8086外部引脚

8086引脚按特性可分为4类

8086外部引脚 { 地址/数据复用总线16根
地址/状态复用总线4根
控制总线
电源线

地址/数据复用总线

$AD_0 \sim AD_{15}$ ，访问存储器/IO时，首先发出地址信号，锁存器将地址信号锁存后，再传输数据。

地址/状态复用总线

A_{19}/S_6 、 A_{18}/S_5 、 A_{17}/S_4 、 A_{16}/S_3

RESET

四个时钟周期的高电平

时钟

时钟周期：CPU最基本的时间计量单位，由主频决定。一个时钟周期又称一个状态 T

总线周期：四个时钟周期组成，分别为 T_1, T_2, T_3, T_4

存储器结构

CPU寄存器
cache
RAM、ROM
外存