

第2章 微处理器与总线

主要内容:

- 2.1概述
- 2.2 8088/8086 微处理器 (60%)
- 2.3 系统总线
- 2.4 80386CPU的工作模式及结构特点
- Pentium / Core i微处理器(Extra)

2.1 微处理器概述

- ■微处理器的功能:
 - ■能够进行算术和逻辑运算
 - ■能对指令进行译码、寄存并执行指令所规定的操作
 - ■具有与存储器和I/O接口进行数据通信的能力
 - ■少量数据的暂存
 - ■提供系统所需的定时和控制信号
 - ■能够响应输入输出设备发出的中断请求

2.1.1 运算器

▶运算器由算术逻辑单元、通用或专用寄存器组及内部总线三个部分组成

ALU的内部包括负责加、减、乘、除运算的加法器,以及实现与、或、非、异或等逻辑运算的逻辑运算功能部件。其结构示意图如图2-1所示。



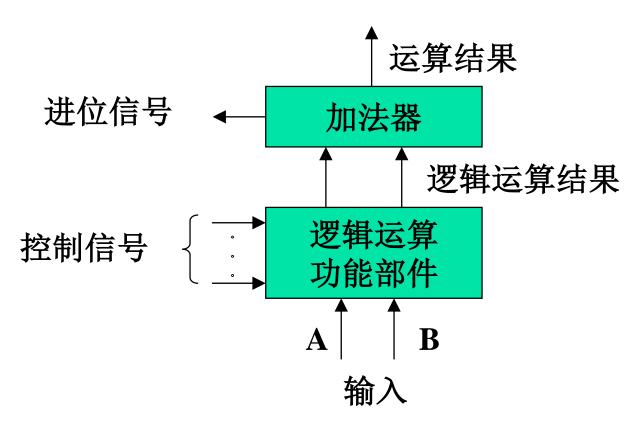


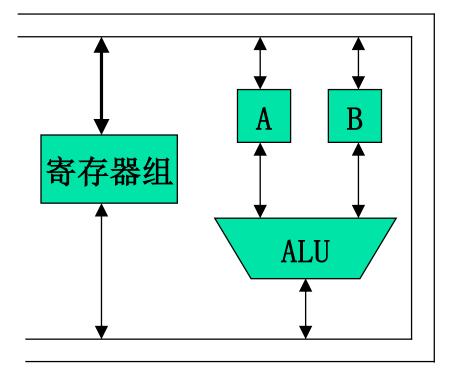
图2-1 ALU结构示意图

运算器的结构

- ■CPU内部总线: CPU内部用于传送数据和指令的传送通道。
- ■根据其内部总线数量的不同分为:
 - □单总线结构运算器
 - □双总线结构运算器
 - □三总线结构运算器

单总线结构运算器

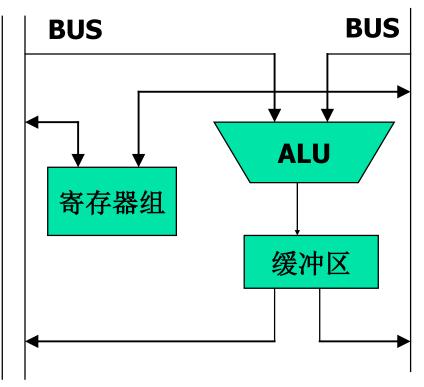
- 所有的部件都通过一条内部 总线传递信息,任何时刻只 有一组数据从源部件传送到 目标部件。



(1)单总线结构

双总线结构运算器

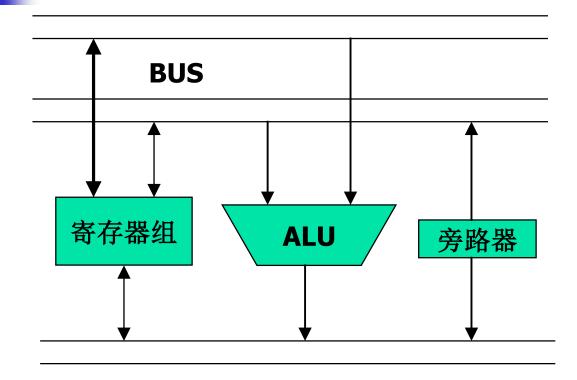




(2)双总线结构

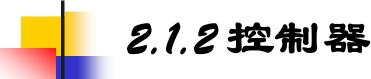
- ■双总线结构是在运算器 内部用两条总线来传送 操作数。
- ■参加运算的两个操作数可同时通过两条总线送至ALU进行运算,运算的结果经缓冲器再通过任意一条总线传送到通任意一条总线传送到通用寄存器。
- ■这种结构的运算器的处理速度比单总线结构的快。

三总线结构运算器



(3)三总线结构

- ■用两条总线来 传送操作数,一 条专门用于传送 运算结果。
- ■在传送运算结 里在传送运算通过 是的同众总数 是外面操作数 是外面操作数 是外面操作数 是的操作数 是的操作数 是的操作数 是的操作。 是一步完成。



■ 指令控制

控制器要能根据指令所在的地址按顺序或在遇到转 移指令时按照转移地址取出指令,分析指令,传送 必要的操作数,并在指令执行结束后存放运算结果, 要保证计算机中指令流的正常工作。

■ 时序控制

■ 指令的执行是在时钟信号的严格控制下进行的。一 条指令的执行时间称为指令周期,时序信号由控制 器产生,使系统按一定的时序关系进行工作。

■ 操作控制

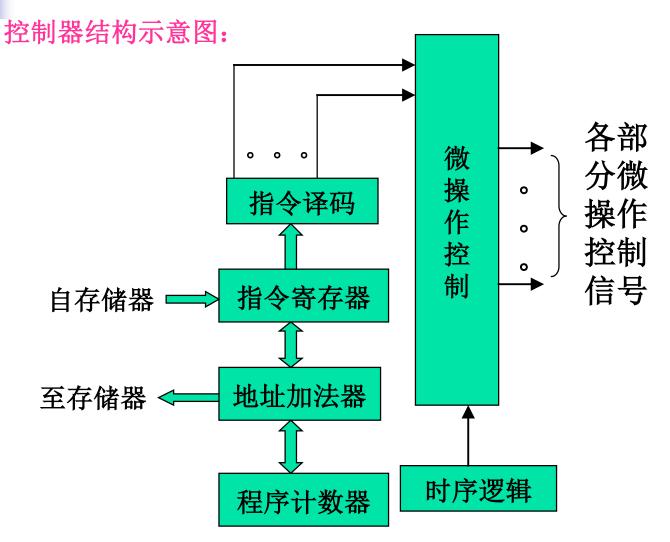
根据指令流程,确定在指令周期的各个节拍中要产生的微操作控制信号,以有效地完成各条指令的操作过程。



≥ 控制器的内部组成:

- ▶程序计数器PC
- ▶指令寄存器IR
- ▶指令译码器ID
- ▶时序控制部件
- ▶微操作控制部件
 - •微操作控制部件用于产生与各条指令相对应的微操作。
 - •根据当前正在执行的指令,在指令的各机器周期的各个节拍内产生相应的微操作控制信号,从而控制整个系统各部件的工作。





2.2 8088 8086 微处理器

- 8088/8086 CPU的特点
- 8088 CPU的外部引脚及其功能
- 8088/8086的功能结构
- 8088/8086的存储器结构
- <u>8088/8086的工作时序</u>

2.2.1 8088 8086 CPU的特点

- 1
- 8086 CPU是Intel系列的16位微处理器,它有16根数据线和20根地址线,所以可寻址的地址空间是2²⁰=1MB。
- 8088 CPU是准16位微处理器,它的内部寄存器、内部运算部件以及内部操作都是按16位设计的,但对外的数据总线只有8位,在处理一个16位数据时,8088需要两步操作,因而称8088是准16位微处理器。

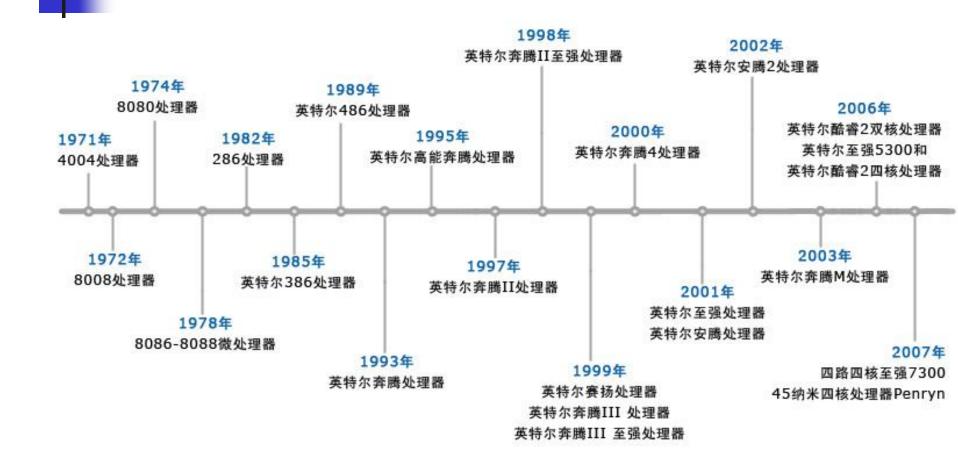


- 8088的指令暂存队列为4个字节,而8086的指令暂存队列为6个字节;
- 8088的外部引脚与8086的外部引脚稍有不同。









GND-1 A ₁₄ -2 A ₁₃ -3 A ₁₂ -4 A ₁₁ -5 A ₁₀ -6 A ₉ -7 A ₈ -8 AD ₇ -9 BOSS AD ₆ -10 AD ₅ -11 AD ₄ -12 AD ₃ -13 AD ₂ -14 AD ₁ -15 AD ₀ -16 NMI-17 INTR-18 CLK-19 GND-20.	40 — V _{CC} 39 — A ₁₅ 38 — A ₁₆ /S ₃ 37 — A ₁₇ /S ₄ 36 — A ₁₈ /S ₅ 35 — A ₁₉ /S ₆ 34 — #SS ₀ 33 — MN/#MX 32 — #RD 31 — HOLD (RQ/GT ₀) 30 — HLDA (RQ/GT ₁) 29 — #WR (LOCK) 28 — IO/#M (S ₂) 27 — DT/#R (S ₁) 26 — #DEN (S ₀) 27 — ALE (QS ₀) 24 — #INTA (QS ₁) 23 — #TEST 22 — READY 21 — RESET	GND — 1 AD ₁₄ — 2 AD ₁₃ — 3 AD ₁₂ — 4 AD ₁₁ — 5 AD ₁₀ — 6 AD ₉ — 7 AD ₈ — 8 AD ₇ — 9 AD ₆ — 10 AD ₅ — 11 AD ₄ — 12 AD ₃ — 13 AD ₂ — 14 AD ₁ — 15 AD ₀ — 16 NMI — 17 INTR — 18 CLK — 19 GND — 20.	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
--	---	---	---



- 8088/8086的功能特征:
 - 指令采用并行流水线处理方式;
 - 对内存空间实行分段管理;
 - 每段64KB, 用段基地址和段内偏移实现对1MB内存空间的寻址
 - 支持多处理器系统;
 - 片内无浮点运算部件,浮点运算由数学协处理器 8087支持(或用软件模拟)。

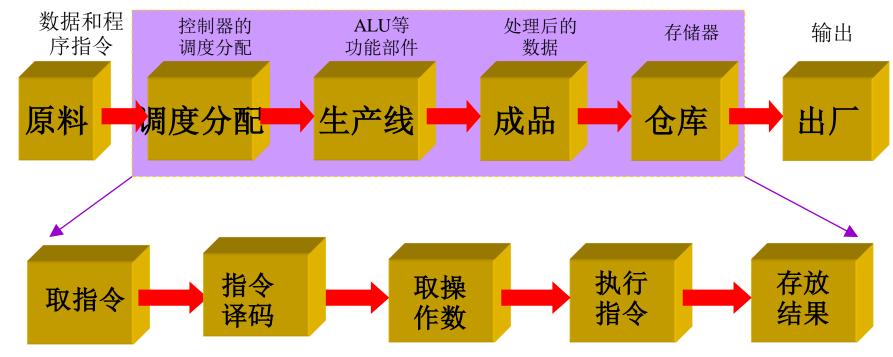


■ 8088/8086的<u>指令流水线</u>

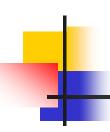
- 指令的一般执行过程:
 - 取指令→指令译码→读取操作
- 串行处理方式存在的问题:
 - CPU访问存储器(存取数据或指令)时要等待总线操作的完成;
 - CPU执行指令时总线、存储器等部件处于空闲状态。
- 解决办法:
 - 让总线空闲时预取指令,使CPU需要指令时就能立刻得到。

指令流水线

- > CPU执行一条指令的过程类似于工厂生产流水线
- ,被分解为多个小的步骤,称为指令流水线。





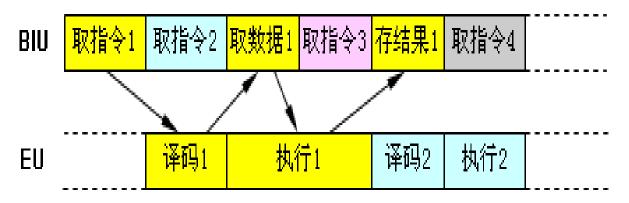


- ·执行单元EU:负责分析指令和执行指令
- ·总线接口单元BIU:负责取指令、取操作数和写结果。
- 验当这两个单元并行工作时,在大多数情况下, 取指令与指令译码/执行都可重叠地进行。因为 EU要执行的指令总是被BIU从存储器中已经" 预取"出来,所以大多数情况下取指令的时间被 "省掉"了,从而加快了程序的运行速度。指令 的这种处理方式被形象地称为指令流水线(此流 水线有两个工位,简称两级流水线)操作。



串行处理 取指令1 译码1 取数据1 执行1 存结果1 取指令2 译码2 执行2 …

并行流水处理







- 指令流水线的操作过程
 - 当指令队列空时,EU就等待,直到有指令为止。只要 有一条指令,EU就从指令队列取出指令并执行之;
 - 当EU从指令队列中取走指令,指令队列出现空字节时, BIU就自动执行取指操作,从内存中取出后续的指令 代码放入队列中,直到填满队列为止;
 - 若EU需要进行存储器或I/0接口的数据存取操作,则 BIU在执行完当前的取指操作后,才会执行EU所要求 的数据存取操作;
 - 当执行转移指令时,BIU就清除指令队列,从新地址 取出指令。新取得的第一条指令直接送EU执行,随后 取得的指令填入队列。



- 内存的分段管理技术:
 - 将内存分为4个段并设置地址段寄存器,以实现对 1MB空间的寻址
 - 将内存地址空间分为多个逻辑段,每个逻辑段最大 为64K个单元,段内每个单元的地址码长度为16位。
- 支持多处理器系统
 - 具有最小和最大两种工作模式以及内置的多任务处理能力,可通过模式选择引脚进行选择。
 - 最小模式称为单处理器模式,系统控制总线的信号由8088CPU直接产生。
 - 最大模式称为多处理器模式,CPU支持系统总线上的 多个处理器,由总线控制器提供所有总线控制信号 和命令信号。

2.2.2 8088 CPU的外部引脚及其功能

- 8088和8086都是具有40条引出线的集成电路芯片,采用双列直插式封装。
- 为了减少芯片的引线,8088/8086的许多引脚具有双重定义和功能,采用分时复用方式工作(即在不同时刻,信号含义不同)。



- 8088/8086的最大和最小两种工作模式可以通过MN/#MX引脚来选择:
 - ◆当MN/#MX=1时,工作在最小模式。此时,构成的系统中只允许有一个处理器,所有控制信号直接由8088/8086提供。
 - ◆当MN/#MX=0时,工作在最大模式。此时,构成的系统中除了有8088/8086之外,还可以接另外的处理器(如8087数学协处理器)。在最大模式下,CPU并不直接向外界提供全部控制信号,而由S₀、S₁、S₂通过8288总线控制器提供。引脚图中括号内的信号用于最大模式。



$$A_{16} \sim A_{19} / S_3 \sim S_6$$

地址/状态复用引脚(输出、三态)。当8088输出地址时,这4个引脚上送出地址的最高4位 A_{16} \sim A_{19} 。

当8088输出状态时,这4个引脚送出状态信号S₃~S₆,其中S₆始终为逻辑0,S₅指示中断允许标志位IF的状态,S₄、S₃的指示CPU当前正在使用的段寄存器,其编码见下表。

S4	S 3	当前使用的段寄存器
0	0	ES
0	1	SS
1	0	CS或未使用任何段寄存器
1	1	DS

1

$A_8 \sim A_{15}$

地址(输出、三态)。CPU访问存储器或I/O接口时, 从这些引脚送出地址信号A₈~A₁₅。

$AD_0 \sim AD_7$

地址/数据复用引脚(双向、三态)。当ALE=1时,这些引脚上传输的是地址,而不是数据。

IO/#M

IO/存储器控制(输出、三态)。低电平表示CPU当前访问的是存储器,高电平表示访问的是I/O接口。

#WR

写控制(输出、三态)。低电平表示CPU正在对存储器或I/O接口进行写操作。

1

DT/#R

数据传送方向控制(输出、三态)。用于确定数据传送的方向。高电平时表示CPU向存储器或I/O接口发送数据;低电平表示CPU从存储器或I/O接口接收数据。此信号常用于控制总线收发器的传送方向。

#DEN

数据允许(输出、三态)。低电平表示数据总线上有有效数据。它在每次访问内存或I/O接口以及在中断响应期间有效。它常用作数据总线驱动器的片选信号。

ALE

地址锁存允许(输出、三态)。高电平表示地址线上有效地址。它常作为锁存控制信号将A₀~A₁₉锁存到地址锁存器。



#RD

读控制(输出、三态)。低电平表示CPU正在对存储器或I/O接口进行读操作。

READY

"准备就绪"(输入、高电平有效)。它是被访问的内存或I/O接口发出的响应信号,高电平表示存储器或I/O设备已准备好,可以进行数据传送。若存储器或或I/O设备没准备好,则可将此引脚拉低。CPU在总线周期的T3采样READY引脚,若为低电平,CPU将自动插入1个或多个等待周期Tw。直到READY变为高电平后,CPU才脱离等待状态,继续执行后续操作。

INTR

可屏蔽中断请求(输入、高电平有效)。CPU在每条指令的最后一个周期对此引脚进行采样,以决定是否进入中断响应周期。此信号可用软件屏蔽。



NMI

非屏蔽中断请求(输入、上升沿有效)。此信号不能用软件屏蔽。若此信号有效,CPU在当前指令执行结束后就进入NMI中断过程。

RESET

系统复位(输入、高电平有效)。为使CPU完成内部复位过程,该信号至少要保持4个时钟周期。复位后CPU内部寄存器的状态如下表所示。当RESET返回低电平时,CPU将重新启动。

#INTA

中断响应(输出、低电平有效)。CPU对INTR信号的响应。在响应过程中,CPU在此引脚连续送出两个负脉冲,可用作外部中断源的中断向量码的读选通信号。

复位后的内部寄存器状态:

内部寄存器	复位后的状态
CS	FFFFH
DS	0
SS	0
ES	0
IP	0
FLAGS	0
其他寄存器	0
指令队列	空



HOLD

总线保持请求(输入、高电平有效)。当某一总线主控设备要占用系统总线时,通过此引脚向CPU提出请求。

HLDA

总线保持响应(输出、高电平有效)。CPU对HOLD请求信号的响应:所有三态引脚变为高阻态,同时使此引脚变为高电平,表示处理器已放弃对总线的控制。当CPU检测到HOLD信号无效后,就立即使此引脚变为低电平,同时恢复对总线的控制。

#SSO

系统状态信号输出。它与IO/#M和DT/#R信号 决定了最小模式下当前总线周期的状态。三者 的状态组合所表示的处理器操作见下表。

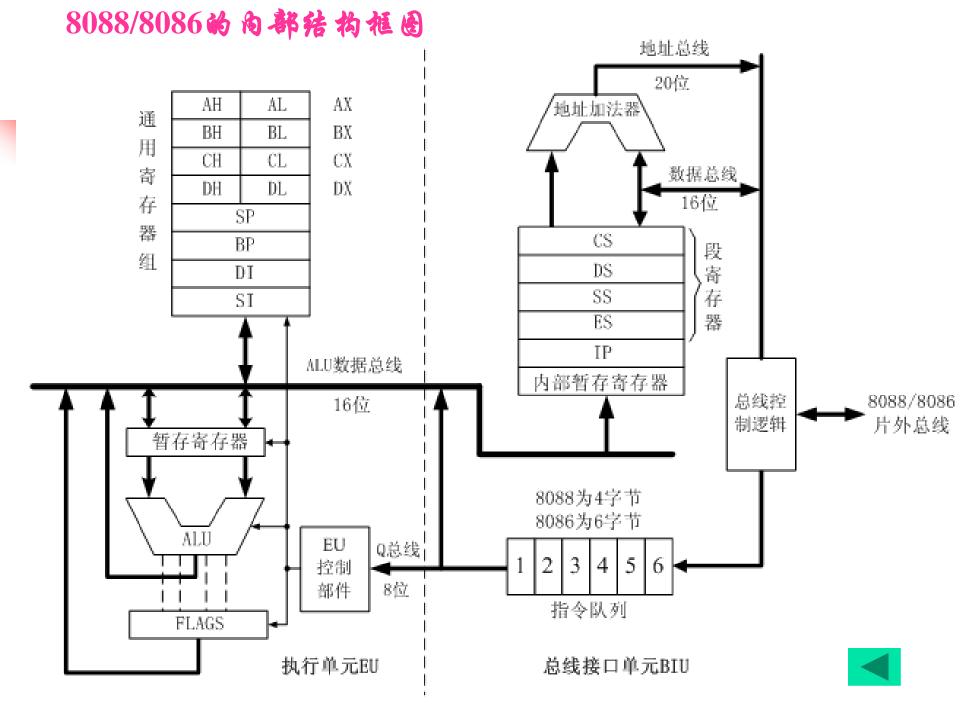
10/#M	DT/#R	#880	操作
1	0	0	发出中断响应信号 <mark>。</mark>
1	0	1	读I/0端口
1	1	0	写I/0端口
1	1	1	暂停
0	0	0	取指令
0	0	1	读内存
0	1	0	写内存
0	1	1	无操作

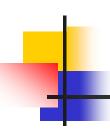
80863 脚	8088对应引脚	说明
AD ₈ ∼AD ₁₅	A ₈ ∼A ₁₅	地址/数据复用引脚(双向、三态)。8086的外部数据总线宽度为16位,这8个引脚在数据有效期间提供了数据总线的高8位。
#BHE/S ₇	#SS ₀	总线高位允许/状态复用引脚(输出)。低电平时,允许数据总线的高8位有效。状态位S7始终为逻辑1。
M/#10	IO/#M	存储器/IO控制。低电平表示CPU当前访问的是 I/O接口,高电平表示访问的是存储器。



2.2.3 8088 8086的功能结构

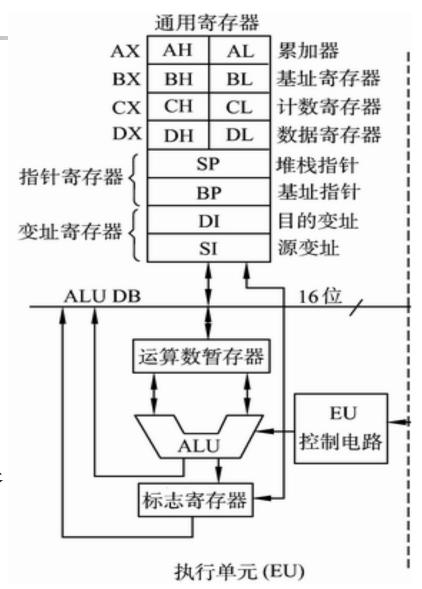
- 8088/8086的内部结构从功能上分成两个单元
 - ■总线接口单元BIU
 - 管理8088与系统总线的接口
 - 负责CPU对存储器和外设进行访问
 - 执行单元EU
 - 负责指令的译码、执行和数据的运算
- 两个单元相互独立,分别完成各自操作,还可以并行执行,实现指令预取(指令读取和执行的流水线操作)





■执行单元(EU)

- □ 算术逻辑单元
- □ 数据暂存寄存器
- □ 标志寄存器
- □ EU控制器
- □通用数据寄存器组
- □ 地址指针和变址寄存器



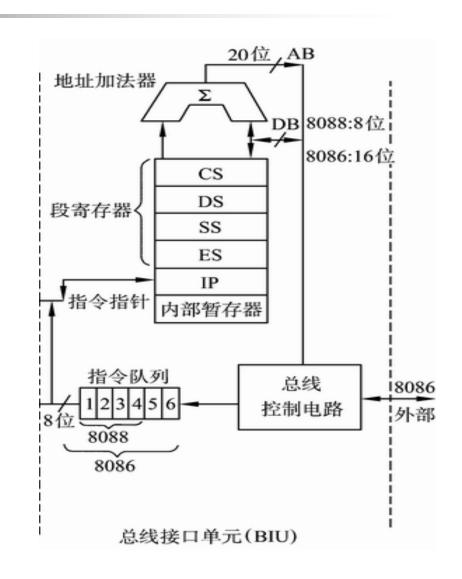
- 4
- 算术逻辑单元
 - ALU (Arithmetic Logic Unit)
 - 处理各种数据
- 数据暂存寄存器
 - 暂存数据
- ■标志寄存器
 - Flag Register
 - 存放运算结果的状态特征和设置控制标志
- EU控制器
 - 程序计数器: 下一条要取出的指令
 - 指令译码器:译码和分析
 - 从队列中取指令、译码、产生控制信号等



- ■通用数据寄存器组
 - 存放数据(数据、中间结果、地址)
- ■地址指针和变址寄存器
 - SP (Stack Pointer) BP (Base Pointer)
 - SI (Source Index) DI (Destination Index)
 - 存放地址信息(偏移地址)



- □指令队列缓冲器
- □总线控制逻辑
- □地址加法器
- □段寄存器
- □ 指令指针寄存器(=? 第一章)





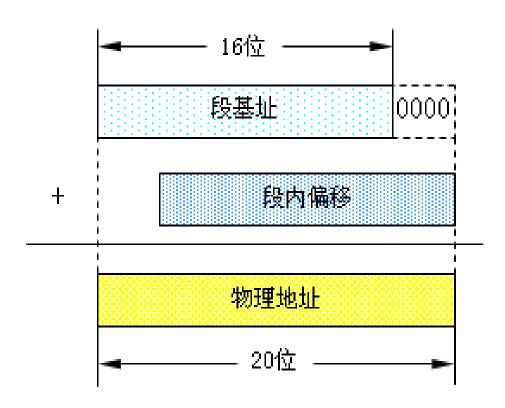
- 6字节指令队列
 - 暂时存放BIU从存储器中预取的指令
- 总线控制逻辑
 - 以控制逻辑方法实现分时传送信息
- 20位物理地址加法器
 - 物理地址=段地址×16d+偏移地址
- 16位段寄存器
 - CS (Code Segment) (与IP联用)
 - DS (Data Segment) (与BX、SI、DI联用)
 - SS (Stack Segment) (与BP、SP联用)
 - ES (Extended Segment) (与DI联用)
- 16位指令指针寄存器
 - IP (Instruction Pointer)



■ 指令预取队列的存在使EU和BIU两个部分可同时进行工作,从而

提高了CPU的效率 降低了对存储器存取速度的要求





8088/8086的寄存器结构

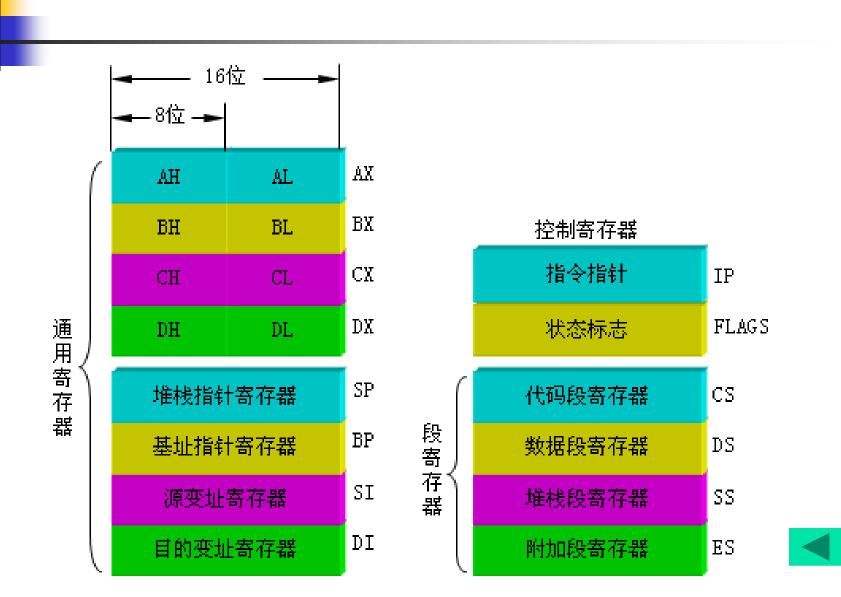
- 8088/8086的寄存器组有
 - 8个通用寄存器
 - 4个段寄存器
 - 1个标志寄存器
 - 1个指令指针寄存器

他们均为16位!



- 汇编语言程序员看到的处理器,就是寄存器
- 所以,一定要熟悉这些寄存器的名称和作用







1. 通用寄存器

■ 8088有8个通用的16位寄存器

(1) 数据寄存器: AX BX CX DX

(2) 变址寄存器: SI DI

(3) 指针寄存器: BP SP

■ 4个数据寄存器还可以分成高8位和低8位两个 独立的寄存器,这样又形成8个通用的8位寄存器

AX: AH AL BX: BH BL

CX: CH CL DX: DH DL



(1) 数据寄存器

- AX称为累加器(Accumulator)
 使用频度最高。用于算术、逻辑运算以及与外设传送信息等
- BX称为基址寄存器 (Base address Register) 常用做存放存储器地址
- CX称为计数器 (Counter) 作为循环和串操作等指令中的隐含计数器
- DX称为数据寄存器(Data register) 常用来存放双字长数据的高16位,或存放外设端口 地址

(2) 指针寄存器

- 指针寄存器用于寻址内存堆栈内的数据
 - SP为堆栈指针寄存器(Stack Pointer),指示堆栈 段栈顶的位置(偏移地址)
 - 为基址指针寄存器 (Base Pointer),表示数据 在堆栈模械的基地垛)是主存中一个特殊的

4

(3) 变址寄存器

- 16位变址寄存器SI和DI
- ■常用于存储器变址寻址方式时提供地址
 - SI是源地址寄存器 (Source Index)
 - DI是目的地址寄存器 (Destination Index)
- 在串操作类指令中, SI、DI还有较特殊的用法



2. 段寄存器

用于存放相应逻辑段的段基地址

CS: 代码段寄存器:

DS: 数据段寄存器

ES: 附加段寄存器

SS: 堆栈段寄存器:

存放指令代码

存放操作数

指示堆栈区域的位 置

3. 指令指针寄存器

- IP (Instruction Pointer) 为指令指针寄存器, 指示主存储器指令的位置
- 随着指令的执行, IP将自动修改以指示下一条 指令所在的存储器位置
- IP寄存器是一个专用寄存器
- IP寄存器与CS段寄存器联合使用以确定下一条 指令的存储单元地址

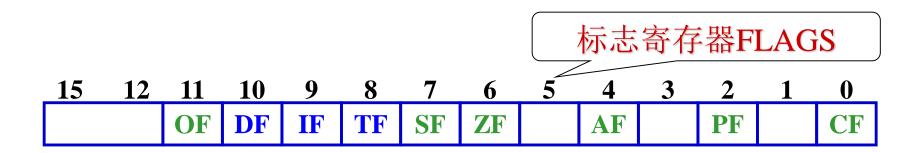
4. 标志寄存器

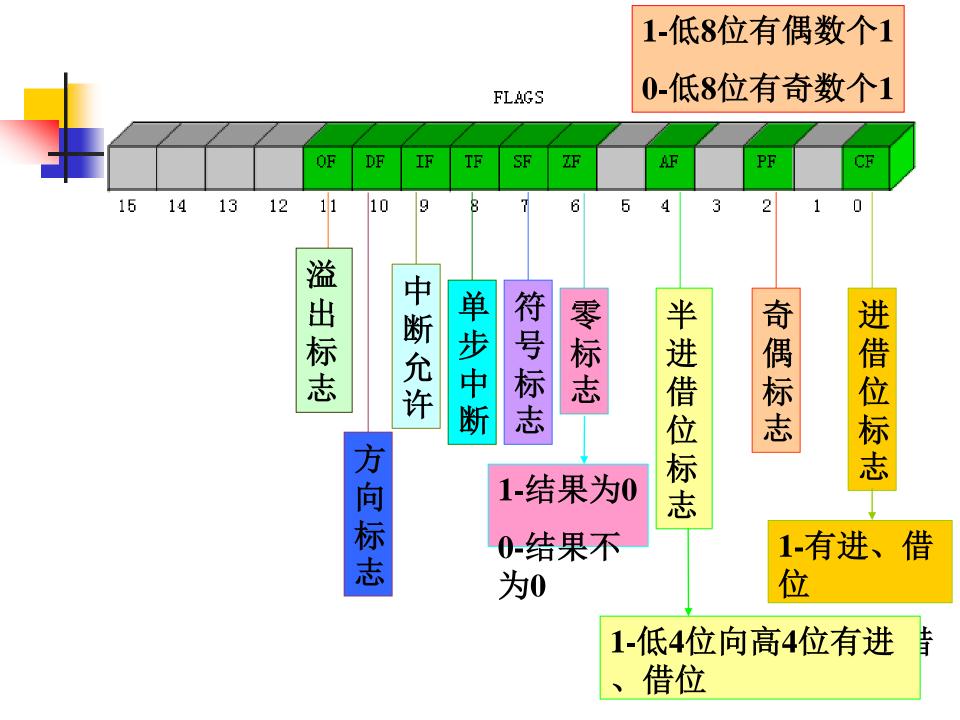
- 标志 (Flag) 用于反映指令执行结果或控制指令执行形式
- 8088处理器的各种标志形成了一个16位的标志寄存器FLAGS(程序状态字PSW寄存器)

● 程序设计需要利用标志的状态



- 状态标志--用来记录程序运行结果的状态信息,许多指令的执行都将相应地设置它 CF ZF SF PF OF AF
- 控制标志 一可由程序根据需要用指令设置,用于控制处理器执行指令的方式
 DF IF TF





进位标志CF(Carry Flag)

■ 当运算结果的最高有效位有进位(加法)或借位(减法)时,进位标志置1,即CF=1; 否则CF=0

```
3AH + 7CH=B6H, 没有进位: CF = 0
```

AAH + 7CH= (1) 26H, 有进位: CF = 1

-

奇偶标志PF (Parity Flag)

- 当运算结果最低字节中"1"的个数为零 或偶数时, PF=1; 否则PF=0
- PF标志仅反映最低8位中"1"的个数是 偶或奇,即使是进行16位字操作

3AH+7CH=B6H=10110110B 结果中有5个"1",是奇数: PF=0

辅助进位标志AF(Auxiliary Carry Flag)

■ 运算时 D_3 位(低半字节)有进位或借位时,AF=1,否则AF=0

这个标志主要由处理器内部使用, 用于十进制算术运算调整指令中, 用户一般不必关心

3AH+7CH=B6H, D3有进位: AF=1

零标志ZF (Zero Flag)

■ 若运算结果为0,则ZF=1;否则ZF=0

●注意: ZF为1表示的结果是0

3AH+7CH=B6H,结果不是零:ZF=0 84H+7CH=(1)00H,结果是零:ZF=1

符号标志SF (Sign Flag)

- 运算结果最高位为1,则SF=1; 否则 SF=0
- ●有符号数据用最高有效位表示数据的符号 所以,最高有效位就是符号标志的状态

3AH+7CH=B6H, 最高位D₇=1: SF=1 84H+7CH=(1)00H, 最高位D₇=0: SF=0

溢出标志OF (Overflow Flag)

■ 若算术运算的结果有溢出,则OF=1; 否则 OF=0

3AH + 7CH=B6H, 没有溢出: OF=0

AAH + 7CH=(1) 26H, 有溢出: OF=1

溢出和进位的区别

- 溢出标志OF和进位标志CF是两个意义不同的标志
- 进位标志表示无符号数运算结果是否超出 范围,运算结果仍然正确
- 溢出标志表示有符号数运算结果是否超出范围,运算结果已经不正确



4

溢出和进位的对比

例1: 3AH+7CH=B6H

无符号数运算: 58+124=182

范围内, 无进位

有符号数运算: 58+124=182

范围外,有溢出

例2: AAH+7CH= (1) 26H

无符号数运算: 170+124=294

范围外,有进位

有符号数运算: -86+124=28

范围内,无溢出

溢出和进位的应用场合

- 处理器对两个操作数进行运算时,按照无符号数求得结果,并相应设置进位标志CF;同时,根据是否超出有符号数的范围设置溢出标志OF
- 应该利用哪个标志,则由程序员来决定。也就是说,如果将参加运算的操作数认为是无符号数,就应该关心进位;认为是有符号数,则要注意是否溢出



溢出的判断

- 判断运算结果是否溢出有一个简单的规则:
 - 只有当两个相同符号数相加(包括不同符号数相减),而运算结果的符号与原数据符号相反时,产生溢出;因为,此时的运算结果显然不正确
 - 其他情况下,则不会产生溢出

3AH+7CH=B6H溢出AAH+7CH无溢出3AH-7CH无溢出AAH-7CH=2DH溢出

陷阱标志TF (Trap Flag)

- 用于控制处理器进入单步操作方式:
 - 设置TF=0,处理器正常工作;
 - 设置TF=1,处理器单步执行指令
- ▶ 单步执行指令——处理器在每条指令执行结束时,便产生一个编号为1的内部中断
- > 这种内部中断称为单步中断
- > 所以TF也称为单步标志
- 利用单步中断可对程序进行逐条指令的调试
- 这种逐条指令调试程序的方法就是单步调试

中断允许标志IF(Interrupt-enable Flag)

- 控制可屏蔽中断是否可以被处理器响应:
 - 设置IF=1,则允许中断;
 - 设置IF=0,则禁止中断

- ▶CLI指令复位中断标志: IF=0
- ▶STI指令置位中断标志: IF=1

方向标志DF (Direction Flag)

- 用于串操作指令中,控制地址的变化方向:
 - 设置DF=0,存储器地址自动增加;
 - 设置DF=1,存储器地址自动减少

- ▶CLD指令复位方向标志: DF=0
- ▶STD指令置位方向标志: DF=1

例3: 8088/8086 ALU执行完8234H + 9BCDH加法操作后,各状态标志位的内容是什么?

解: 把上述16进制数加法写成二进制数加法

1000 0010 0011 0100

+ 1001 1011 1100 1101

1 0001 1110 0000 0001

最高位有进位, 所以CF=1;

低8位中包含1个"1",1的个数为奇数,所以PF=0;

bit3向bit4有进位,所以AF=1;

结果不为零,所以ZF=0:

结果的符号位为0, 所以SF=0;

结果超出了16位有符号数的范围,所以OF=1。

(从两个负数相加结果为正数也可以看出结果溢出)

2.2.4 8088 8086的存储器结构

- 存储器是计算机存储信息的地方。
- 掌握数据存储格式,以及存储器的分段管理对以后的汇编程序设计非常重要

■ 寄存器、存储器(主存)、外存(包括硬盘、光 盘、磁带等存储介质)有什么区别?

答案



- 寄存器是微处理器 (CPU) 内部暂存数据的存储 单元,以名称表示,例如: AX, BX........等
- 存储器也就是平时所说的主存,也叫内存,可 直接与CPU进行数据交换。主存利用地址区别
- 外存主要指用来长久保存数据的外部存储介质,常见的有硬盘、光盘、磁带、U盘等。外存的数据只能通过主存间接地与CPU交换数据
- 程序及其数据可以长久存放在外存,在运行需要时才进入主存

1. 数据的存储格式

- 计算机中信息的单位
 - 二进制位Bit: 存储一位二进制数: 0或1
 - 字节Byte: 8个二进制位, D₇~D₀
 - 字Word: 16位, 2个字节, D₁₅~D₀
 - 双字DWord: 32位, 4个字节, D₃₁~D₀
- 最低有效位LSB: 数据的最低位,D₀位
- 最高有效位MSB:数据的最高位,对应字节、字、双字分别指 D_7 、 D_{15} 、 D_{31} 位





存储单元及其存储内容

- 每个存储单元都有一个编号;被称为存储器 地址
- 每个存储单元存放一个字节的内容

0002H单元存放有一个数据34H 表达为 [0002H]=34H



多字节数据存放方式

- 80x86处理器采用"低对低、高对高"的存储形式,被称为"小端方式Little Endian"【Linux, Windows】
- ●相对应还存在"大端方式Big
- Endian" [???].

[0002H] = 1234H

0002H号"双字"单元的内容为:

[0002H] = 78561234H



2.存储器的分段管理

- 8088CPU有20条地址线
 - 最大可寻址空间为2²⁰=1MB
 - 寻址范围从0000H~FFFFFH
- 8088CPU将1MB空间分成许多逻辑段(Segment)
 - 每个段最大限制为64KB
 - 段地址的低4位为0000B
- 这样,一个存储单元除具有一个唯一的物理地址外,还具有多个逻辑地址

3.物理地址和逻辑地址

- 8088CPU存储系统中,对应每个物理存储单元都有一个唯一的20位编号,就是物理地址,从00000H ~ FFFFFH
- 分段后在用户编程时,采用逻辑地址,形式为

段基地址: 段内偏移地址



物理地址 14700H 逻辑地址 1460H:100H



- 段地址说明逻辑段在主存中的起始位置
- 8088规定段地址必须是模16地址: xxxx0H
- 省略低4位0000B, 段地址就可以用16位数据表示, 就能用16位段寄存器表达段地址
- 偏移地址说明主存单元距离段起始位置的偏移量
- 每段不超过64KB,偏移地址也可用16位数据表示

物理地址和逻辑地址的转换

将逻辑地址中的段地址左移4位,加上偏移地址就得到20位物理地址

_ 逻辑地址 1460:100、1380:F00 物理地址 14700H 14700H

举例

例4 如果要访问物理地址为12003H存储单元中的内容,请写出逻辑地址,并画出示意图。

解:物理地址12003H对应的逻辑地址有很多种分解方法,如:

 $12003H = 1200H \times 16 + 0003H$

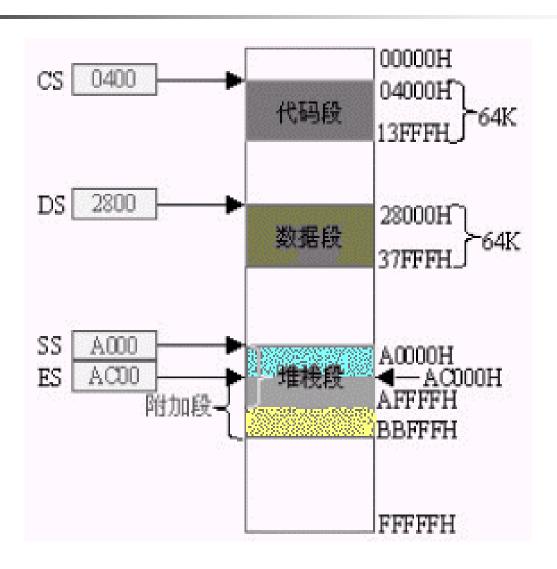
 $12003H = 1000H \times 16 + 2003H$

 $12003H = 1100H \times 16 + 1003H$

4. 段寄存器的使用

- 8088有4个16位段寄存器
 - CS (代码段) 指明代码段的起始地址
 - SS (堆栈段) 指明堆栈段的起始地址
 - DS (数据段) 指明数据段的起始地址
 - ES (附加段) 指明附加段的起始地址
- 每个段寄存器用来确定一个逻辑段的起始地址, 每种逻辑段均有各自的用途

段和段之间可以相互分离、连接、部分重叠或完全重叠。



代码段寄存器CS (Code Segment)

- 代码段用来存放程序的指令序列
 - 代码段寄存器CS存放代码段的段地址
 - ■指令指针寄存器IP指示下条指令的偏移地址
- 处理器利用CS: IP取得下一条要执行的指令

堆栈段寄存器SS(Stack Segment)

- 堆栈段确定堆栈所在的主存区域
 - 堆栈段寄存器SS存放堆栈段的段地址
 - 堆栈指针寄存器SP指示堆栈栈顶的偏移地址
- 处理器利用SS:SP操作堆栈顶的数据



数据段寄存器DS(Data Segment)

- 数据段存放运行程序所用的数据
 - ■数据段寄存器DS存放数据段的段地址
 - 各种主存寻址方式(有效地址EA)得到存储器中操作数的偏移地址
- 处理器利用DS:EA存取数据段中的数据



附加段寄存器ES (Extra Segment)

- 附加段是附加的数据段,也保存数据:
 - 附加段寄存器ES存放附加段的段地址
 - 各种主存寻址方式(有效地址EA)得到存储器中操作数的偏移地址
- 处理器利用ES:EA存取附加段中的数据
- 串操作指令将附加段作为其目的操作数的存放 区域



如何分配各个逻辑段

- 程序的指令序列必须安排在代码段
- 程序使用的堆栈一定在堆栈段
- 程序中的数据默认是安排在数据段,也经常安排在附加段,尤其是串操作的目的区必须是附加段
- 数据的存放比较灵活,实际上可以存放在 任何一种逻辑段中

演示



寄存器的总结

■ 8088有8个8位通用寄存器、8个16位通用寄存器。

■ 8088有6个状态标志和3个控制标志

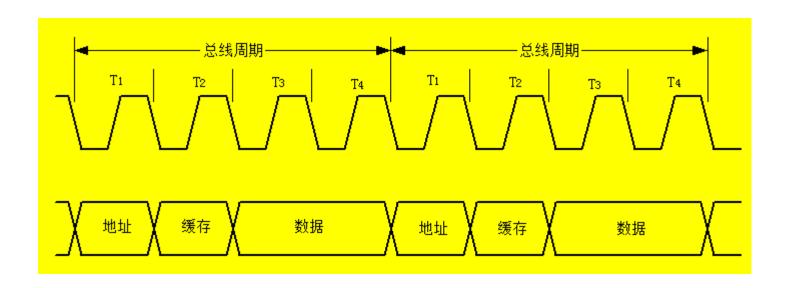
■ 8088将1MB存储空间分段管理,有4个段寄存器, 对应4种逻辑段

2.2.5 8088|8086 CPU的工作时序

- 时序的概念: CPU各引脚信号在时间上的关系
- 时钟周期:微处理器在运行过程中是在统一的时钟的协调下执行每一个操作的。每个时钟脉冲的持续时间称为一个时钟周期(用Ti表示)。
- 总线周期: CPU完成一次访问内存(或接口) 操作所需要的时间。

一个总线周期至少包括4个时钟周期

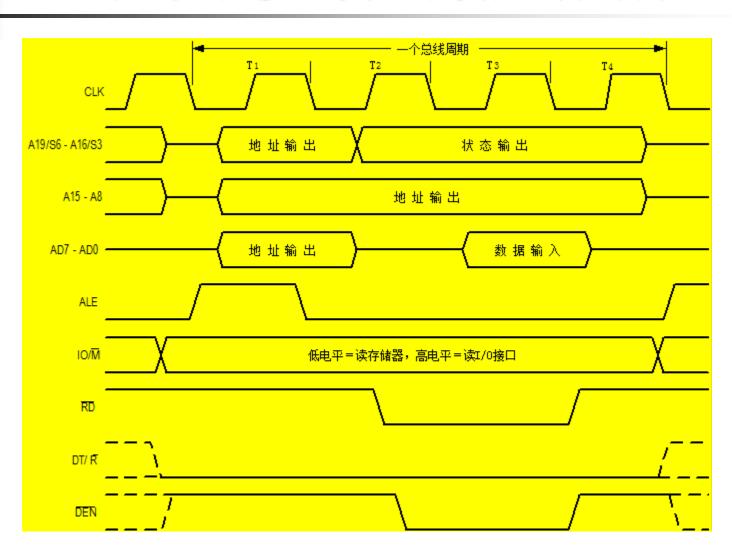
典型的总线周期



总线操作

- 读周期—总线读操作(取指令,读存储器,读**I/O**接口)
- 写周期—总线写操作(写存储器,写I/O接口)
- 中断响应周期—中断响应操作
- ■总线请求与总线响应

8088在最小模式下的读周期时序





最小模式下的时序操作小结

- 一个基本总线周期由T1~T4组成;
 - T1状态: ALE、 # M/IO、DT/ # R有效,分时复用 线上传送地址信息;
 - T2状态: # RD、# WR、# DEN信号有效。对读操作,数据线呈高阻;对写操作,直接出现输出数据;
 - T3状态: 在T3的前沿检测READY, 若有效,则读操作出现输入数据; 若READY无效, 持续其他各控制信号,加入若干个等待态Tw,并在每个Tw前沿继续检测READY,直至READY有效为止;
 - T4状态:接收或发送数据,将各控制信号驱动为无效,进入无源状态,为下一个总线周期做好准备。



总线请求与总线响应时序

- CPU在每个T的下降沿对HOLD符号进行采样。
- 当HOLD=1时,在T₄、T₁的下降沿发HLDA信号,释放总线控制权。(所有输出总线呈高阻状态)
- 当HOLD=0时,则在下一个T后,HLDA=0, CPU收回总线控制权。

2.3 系统总线

主要内容:

- 总线的基本概念和分类;
- ■总线的工作方式;
- ■常用系统总线标准。

2.3.1 概述

■总线:

是一组导线和相关的控制、驱动电路的集合。是计算机系统各部件之间传输地址、数据和控制信息的公共通道。

总线结构的优点

- 简化系统设计(模块化)
- ■提高兼容性
- 便于扩充升级
- 便于维修
- ■减低生产成本

总线分类

按相对

片内总线

CPU的位置

片外总线

按

层次结构

CPU总线: CPU ←→ 其他部件

系统总线: 主机←→I/O接口

外部总线: 微机←→外设

按 传送信息 地址总线 (AB)

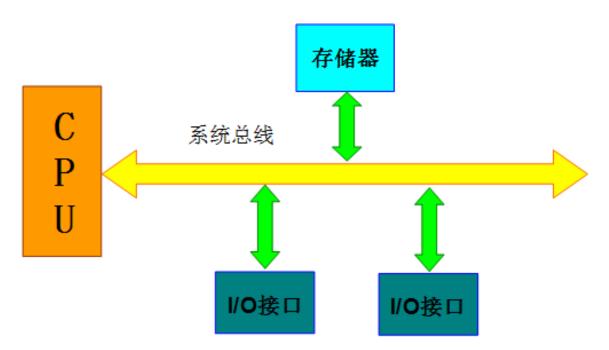
数据总线(DB)

控制总线(CB)

总线结构

■单总线结构

简单,但总线竞争严重





多总线结构 面向CPU的双总线结构

双总线结构

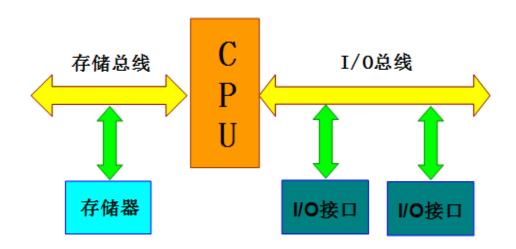
面向主存的双总线结构

多总线结构



■面向CPU的双总线结构

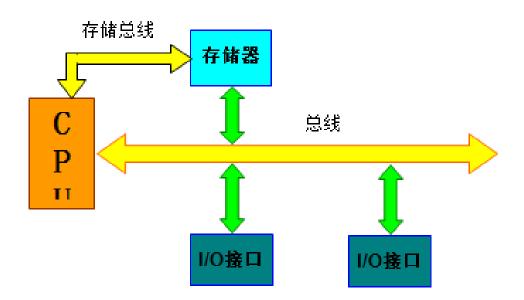
- 把需要很高带宽的主存储器用存储总线单独与CPU相 连
- 问题:外设到主存的数据传输必须通过CPU,传输效率低,无法实现DMA传输





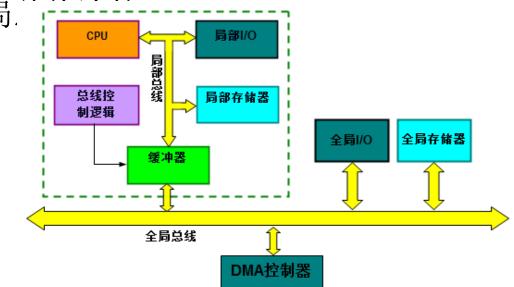
■面向主存的双总线结构

■ 主存储器即与CPU直接连接,又与系统总线连接, 较好地解决了上述问题

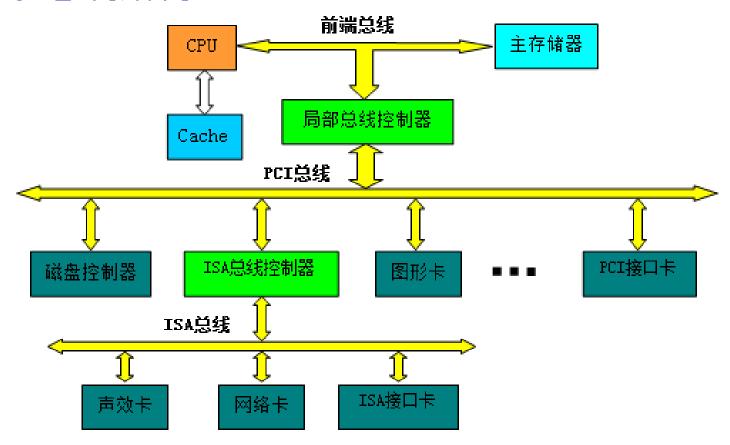


■ 双重总线结构

- CPU与高速的局部存储器和局部I/0接口通过高传输速率的局部总线连接



■多总线结构



总线的性能指标

- ■总线带宽BW
 - 总线的带宽指的是单位时间内总线上可传送的数据量,即我们常说的每秒钟传送多少字节。单位是字节/秒(B/s)或兆字节/秒(MB/s)。
- ■总线宽度W
 - ■总线的宽度指的是总线能同时传送的数据位数。
- 总线工作频率f

● 总线带宽的计算公式如下:
BW = (W/8) × f/每个存取周期的时钟数

2.3.2 总线技术

- 总线传输需要解决的问题:
 - 传输同步——协调通信双方的传输操作
 - 同步、异步、半同步
 - 总线仲裁——消除多个设备同时使用总线造成的冲 突现象
 - Master查询,Slave独立请求
 - 出错处理
 - 信号驱动
 - 采用三态输出电路或集电极开路输出电路来驱动总线。



总线的定时实现方式

- 同步传输——用公共的时钟统一各部件数据发送和接收的时机,要求收发双方处理速度要一致。
- 异步传输——用控制和状态信号协调各部件数据发送和接收的时机,收发双方处理速度可以不一致。
- 半同步——用公共的时钟协调控制和状态信号的产生时机(即控制和状态信号与时钟是同步的),但数据发送和接收的时机仍不固定。



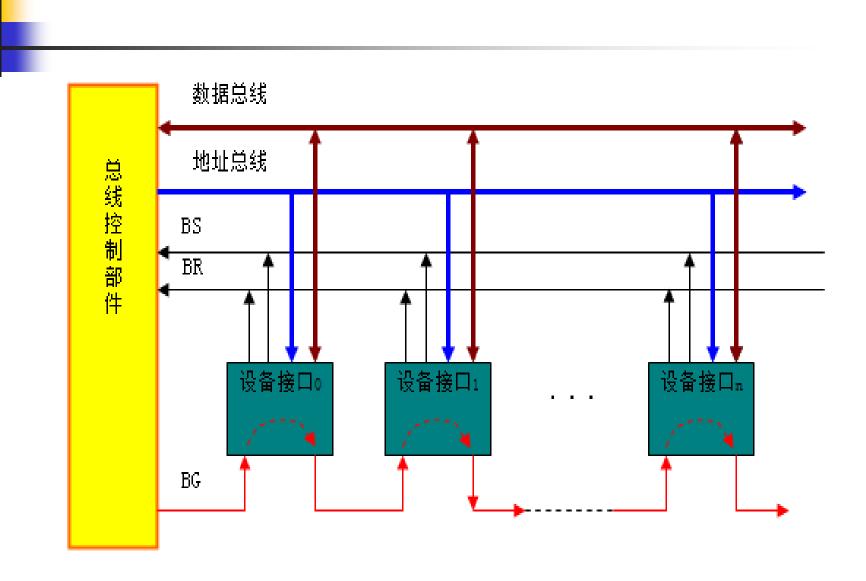
总线仲裁

- 用来决定某一时刻哪一个部件可以使用总线
 - 集中控制——统一由总线控制器进行控制
 - 分散控制——总线控制由各部件共同实现,所有部件均按统一的规则来访问总线

总线仲裁——集中控制

■ 链式查询

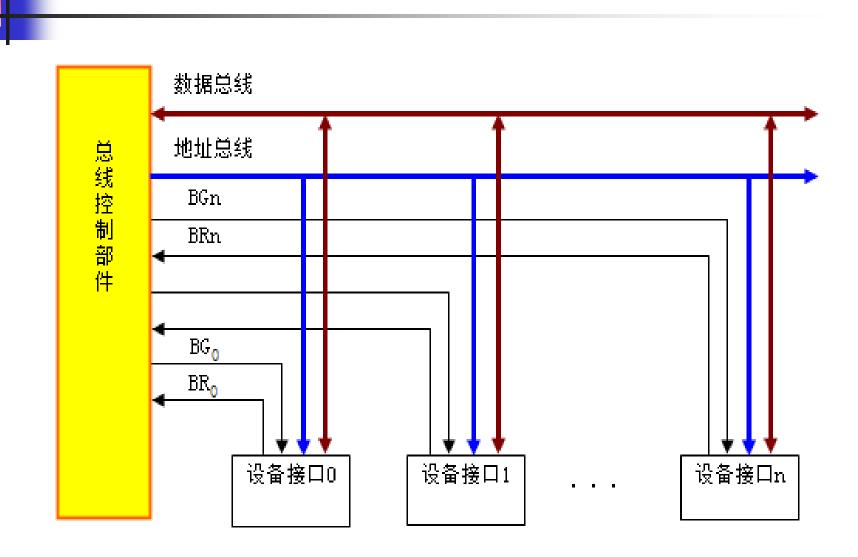
- 部件提出申请 (BR)
- 总线控制器发出批准信号 (BG)
- 提出申请的部件截获BG,并禁止BG信号进一步向后 传播
- 提出申请的部件发出总线忙信号(BS),开始使用总线。总线忙信号将阻止其他部件使用总线,直到使用总线的设备释放总线
- 电路最简单,但优先级固定,不能改变





■ 独立请求

- 每个设备都拥有独立的总线请求线和总线应答线
- 总线控制器对所有的总线请求进行优先级排队,并 响应级别最高的请求
- 得到响应的设备将占用总线进行传输





在链式查询中仅用两根线确定总线使用权属于 那个设备。

■ 独立请求方式需采用2n根线。

2.3.3 常见的系统总线

- ISA (8/16位)
- PCI (32/64位)
- PCI-E总线

ISA总线

■ ISA(Industry Standard Architecture)是工业标准体系结构总线的简称。



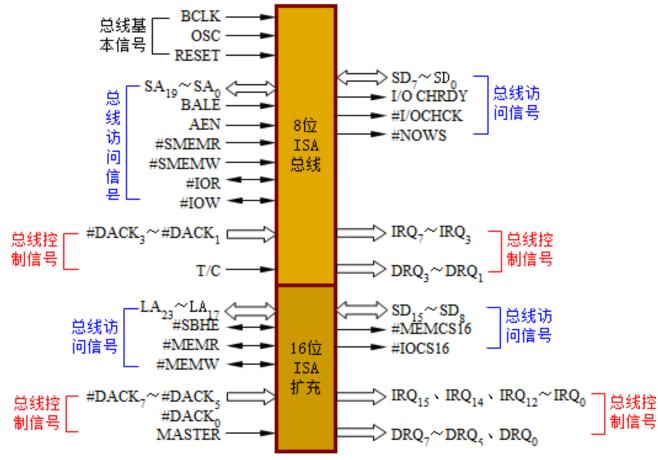
图中的黑色插槽即为16位 ISA总线插槽



■ ISA总线的主要性能指标:

- I/0地址空间范围为0100H~03FFH
- 24位地址线,可直接寻址的内存容量为16MB
- 总线宽度8位或16位,最高时钟频率8MHz,最大稳态 传输率分别为8MB/s和16MB/s
- 支持15级中断,并允许中断共享功能
- 8个DMA通道
- 开放式总线结构,允许多个CPU共享系统资源

- TSA 总线的信号引脚图



PCI总线

PCI (Peripheral Component Interconnect) 是

从几下法当此的答纸 PCI是Intel公司开发的一套局部总线系 统,它支持32位或64位的总线宽度,频率通 常是33MHz。目前最快的PCI2.0总线速度是 66MHz.

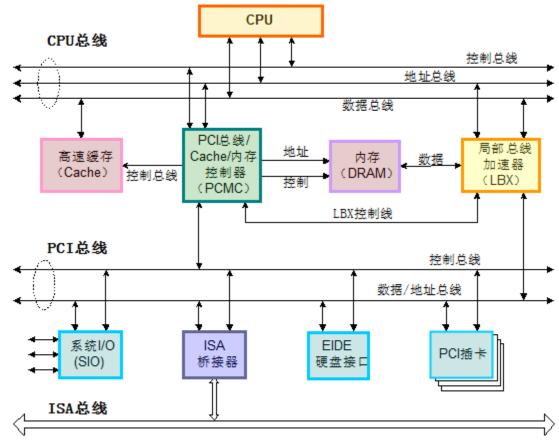
图中白色的插槽即为32位PCI总线插槽



■ PCI总线的主要特点:

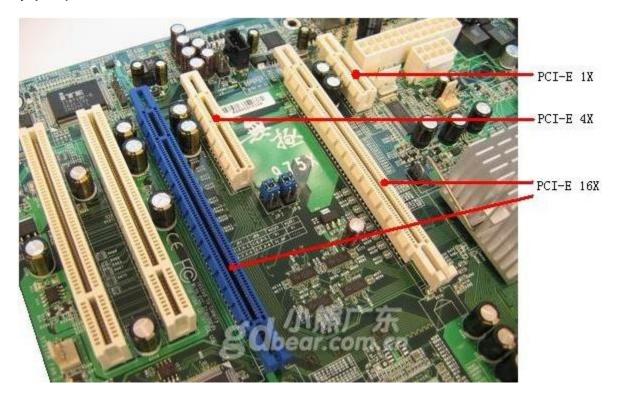
- 高速、低延迟。PCI总线宽度为32/64位,总线时钟频率33MHz/66MHz,最大数据传输速率528MB/s;
- 同步传输方式;
- 独立于处理器,与处理器频率无关,与处理器更新换代无关;
- 自动识别外设,全自动配置与资源申请/分配(即插即用);
- 具有与处理器和存储器子系统完全并行操作的能力;
- 具有隐含的集中式中央仲裁系统;
- 采用地址线和数据线复用技术,减少了引线数量;
- 支持一次读/写多个数据的Burst传输方式;
- 完全的多总线主控能力;
- 提供地址和数据的奇偶校验,使系统更可靠。

■ PCT 单线的休系结构



PCI-Express总线

■ PCI Express, 简称PCIe或PCI-E, 是微机系统中一种用来代替PCI、AGP接口规范的新型系统总线标准。





■ PCI-E的特点:

- PCI-E总线的连线很少,可以根据所连接的硬件设备的不同而使用不同的传输速率。
- 多种连接方式,PCI-E可以像USB或者1394一样。
- 将具有更大的传输带宽来满足<u>图形技术</u>日益增加的数据量
- 点对点连接。每个PCI-E设备都独立地使用自己专用的连接。
- PCI-E具有高级电源管理和监视功能,所有的PCI-E设备都支持热插拔。
- 内存纠错成为标准功能。。
- 最大提供功率达到了70W,比AGP接口有了很大的提高, 基本满足了主流显卡的需求。

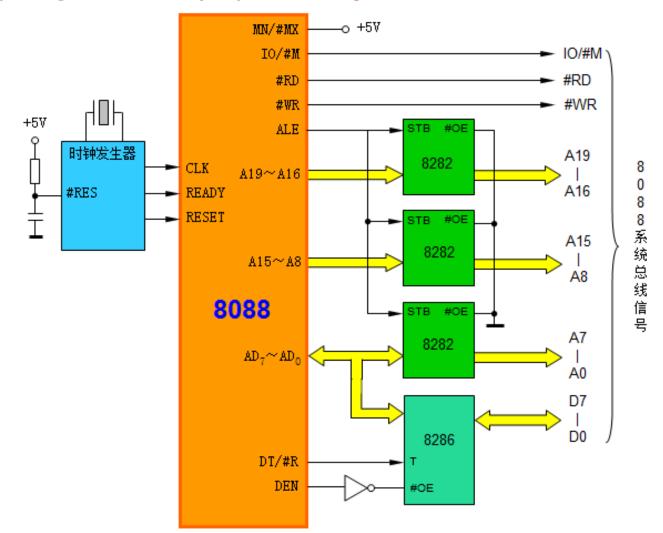
2.3.4 8088系统总线

- 最小模式——仅支持单处理器
 - 最小模式下主要解决:
 - 地址与数据的分离
 - ▶ 地址锁存
 - 电路实现方案
 - 用3片8位的锁存器8282实现地址锁存。ALE为锁存控制信号, OE#≡0使锁存的地址直接输出;
 - 用1片双向三态门8286用作数据总线驱动和隔离, DT/R#作为方向控制, DEN#作为开门信号;
 - 其他控制信号由8088直接产生。

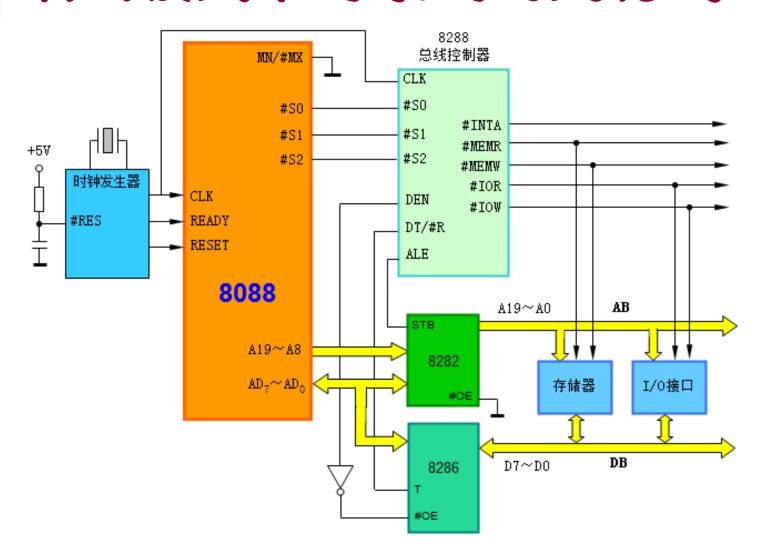
最大模式

- 最大模式——可支持多处理器
 - -大多数控制信号是由总线控制器8288对 S_0 #、 S_1 #、 S_2 #三个信号译码得到,如DT/R#、ALE、DEN#、IOR#、IOW#、MEMR#、MEMW#信号。DB和AB的构成基本同最小模式。
 - ■PC/XT机的总线采用了最大模式,但有三点区别:
 - 地址总线驱动用2个74LS373和1个74LS244 代替3个8282;
 - 数据总线驱动用74LS245代替8286;
 - ■支持DMA传送。

最小模式下的系统总线构成



最大模式下的系统总线构成





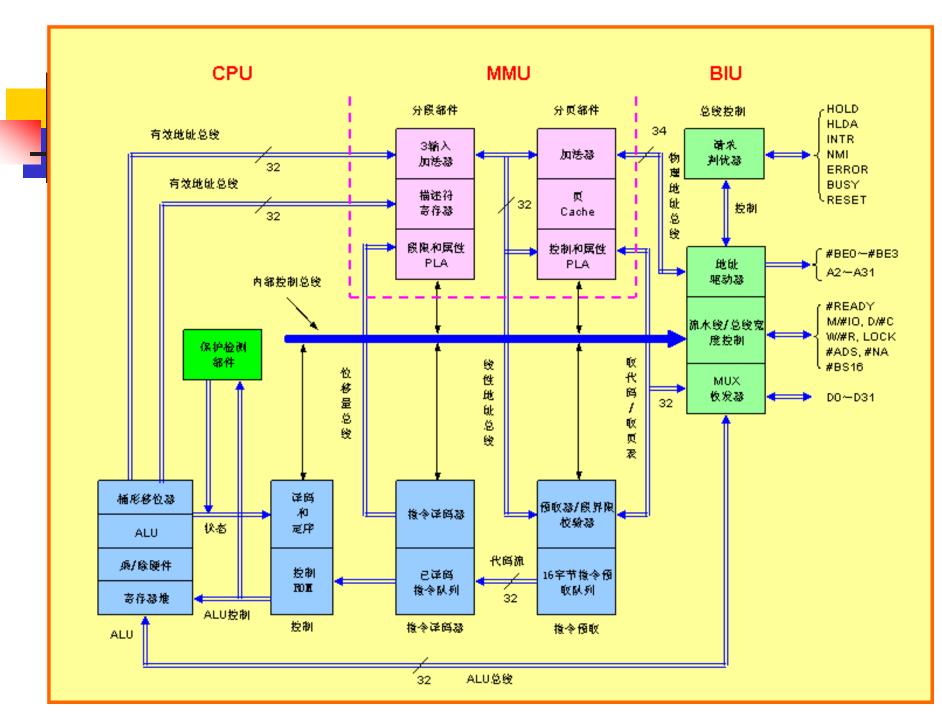
2.4 80386微处理器

- 80386微处理器的主要特性
- 80386内部结构
- 80386内部寄存器
- 80386处理器的引脚信号
- 80386工作模式



2.4.180386微处理器的主要特性

- 采用全32位结构,其内部寄存器、ALU和操作是32位,数据线和地址线均为32位。
- 提供32位外部总线接口,最大数据传输率为32MB/s, 具有自动切换数据总线宽度的功能。
- 片内集成存储器管理部件MMU,可支持虚拟存储和特权保护,虚拟存储器空间可达64TB(246字节)。
- 具有三种工作方式:实地址方式、保护方式和虚拟 8086方式。
- 采用了比8086更先进的流水线结构,能高效、并行地 完成取指、译码、执行和存储管理功能。具有增强的 指令预取队列,指令队列从8086的6字节增加到16字节。





- 总线接口部件(BIU: Bus Interface Unit)
 - 负责与存储器和I/0接口传送数据,并产生访问存储器和I/0端口所必须的地址和命令信号。
 - 80386的总线周期只包含2个时钟周期。没有其它 总线请求时,BIU将自动取出下条指令送到指令预 取队列。



- 中央处理部件(CPU)包括指令预取单元、指令 译码单元和执行单元三部分。
 - 指令预取单元(IPU, Instruction Prefetch Unit) 负责从存储器取出指令,放到一个16字节的指令队 列中。
 - 指令译码单元(IDU, Instruction Decode Unit)从 指令预取单元之中取出指令,进行译码。
 - 执行单元(EU, Execution Unit)包括8个32位的寄存器组,32位的算术逻辑单元ALU,一个64位桶形移位寄存器和一个乘法除法器。



■ 分段部件

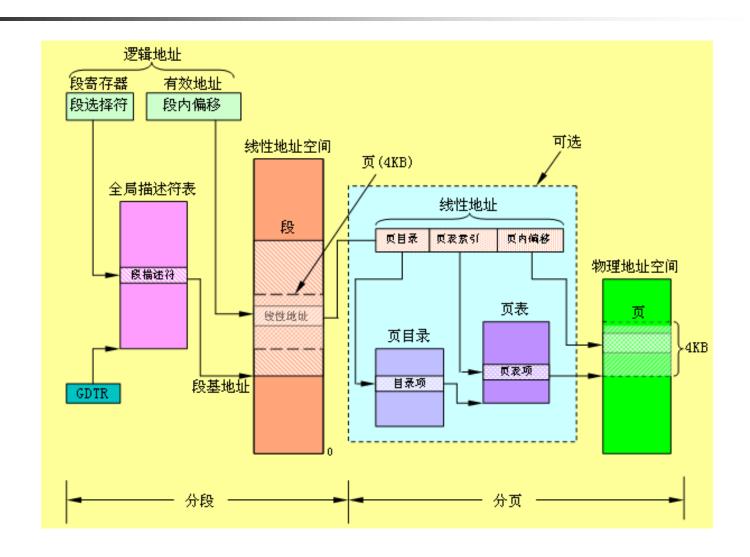
■ 应执行部件的请求,把逻辑地址转换成线性地址。在完成 地址转换的同时还要执行总线周期的分段合法性检验。该 部件可以实现任务之间的隔离,也可以实现指令和数据区 的再定位。

■ 分页机构

■ 把由分段部件或代码预取单元产生的线性地址转换成物理 地址,并且要检验访问是否与页属性相符合。



- 分段把处理器的可寻址存储空间(称为线性地址空间)分成较小的、受保护的地址空间,称为段。 段可用来存放代码、数据和堆栈,或者存放系统数据结构(例如TSS或LDT)。
- 分页支持虚拟存储器环境,在这种环境中,用一个小容量的物理存储器(RAM和ROM)和一些磁盘空间来模拟一个非常大的线性地址空间。当使用分页时,每一个段都分为多个页(页面大小通常为4KB),页可被存储在内存中或磁盘中。



2.4.3 80386内部寄存器

■通用寄存器

- 80386有8个32位的通用寄存器,名字分别是: EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP。
- ■指令指针和标志寄存器
 - 80386的指令指针EIP是一个32位寄存器。
 - 80386的标志寄存器EFLAGS也是一个32位寄存器,其中只使用了15位。

■ 段寄存器

- 80386有6个段寄存器,分别是CS, DS, SS, ES, FS和GS。
- ■控制寄存器
 - 80386有4个32位控制寄存器(CR0、CR1、CR2和CR3)。
 - 保存全局性的机器状态。



■ 系统地址寄存器

■ 80386有4个系统地址寄存器,用来存储操作系统需要的保护信息和地址转换表信息、定义目前正在执行任务的环境、地址空间和中断向量空间。

■调试寄存器

- 80386设有8个32位调试寄存器DR0~DR7,它们为调试提供了硬件支持。
- DR0~DR3是4个保存线性断点地址的寄存器; DR4、DR5为备用寄存器; DR6为调试状态寄存器,通过该寄存器的内容可以检测异常,并允许或禁止进入异常处理程序; DR7为调试控制寄存器,用来规定断点字段的长度、断点访问类型、"允许"断点和"允许"所选择的调试条件。

■ 测试寄存器

- 80386设置了8个32位的测试寄存器TR0~TR7
- TRO~TR5由Intel公司保留,用户只能访问TR6、TR7
- TR6是测试控制寄存器,TR7是测试状态寄存器,保存测试结果的状态。

2.4.4 80386处理器的引脚信号

- CLK2 两倍时钟输入信号。
- D0~D31 数据总线信号,双向三态。
- A2~A31 地址总线信号输出,三态。
- #BE0~#BE3 字节选通输出信号,每条线控制选通一个字节
- W/#R 读/写控制,输出信号。
- D/#C 数据/控制输出信号,表示是数据传送周期还是控制周期。
- M/#I0 存储器与I/0选择信号,输出。
- LOCK 总线锁定输出信号。
- #ADS 地址状态,三态输出信号。表示总线周期中地址 信号有效。

- 1
- #NA 下一地址请求,输入信号。
- #BS16 总线宽度为16的输入信号。
- #READY 准备就绪,输入信号。
- HOLD 总线请求保持,输入。
- HLDA 总线响应保持,输出。
- PEREQ 处理器扩展请求,输入。
- #BUSY 协处理器忙,输入。
- #ERROR 协处理器出错,输入。
- NMI 不可屏蔽中断请求信号,输入。
- INTR 可屏蔽中断请求信号,输入。
- RESET 复位信号。

2.4.4 80386工作模式

- 实地址模式
 - 当80386加电或复位后,就进入实地址工作模式。
 - 保留了两个固定的存储区域
 - 中断向量表区: 00000H—003FFH, 在1K字节存储空间保留 256个中断服务程序的入口地址,每个入口地址占用4个字 节,与8088/8086一样。
 - 系统初始化区: FFFFFFF0H—FFFFFFFH, 存放ROM引导程序。



■ 保护虚地址模式

- 当80386工作在保护方式时,其能够访问的线性地址空间可达4GB,而且允许运行几乎不受存储空间限制的虚拟存储器程序。用户逻辑地址空间,即虚拟存储器地址空间可达64TB。
- 80386提供了复杂的存储管理和硬件辅助的保护机构, 且可运行现有8088/8086/80286的所有软件。
- 支持多任务操作系统的特别优化的指令。