

期中考试

第一章 计算机基础知识

ALU 算术逻辑单元

BIU 总线接口单元

EU 执行单元

冯诺依曼体系：控制器、运算器、存储器、输入输出设备

定点小数

两个补码的数进行相加，判断溢出的方法：（书上有讲过，补码运算连同符号位一起，并且进位自动舍去）

两个同号的整数相加，若结果的符号位与加数的符号位相反，则一定发生溢出

IEEE754标准

例如：257使用 IEEE754 标准来表示，并且转换为16进制

257的二进制为 100000001, 1.00000001×2^8 (IEEE754标准, 第一位一般为1)

阶码（偏移） $8 + 127$, 加上127 = 10000111

尾数 00000001 0000 0000 0000 0000 24位

机器码为 0 10000111 00000001 0000 0000 0000 0000 机器码表示的时候将小数点和前面的1省去

浮点数表示范围

- 最大正数: $0.111\dots1 \times 2^{(011\dots1)} = (1-2^{-23}) \times 2^{+127}$
- 最小正数: $0.100\dots0 \times 2^{(100\dots0)} = (1/2) \times 2^{(-128)} =$

浮点数格式化

原码尾数格式化：正数为 $0.1xxxxxxx$ 形式，最大值为 $0.1xxxx1$

负数为 $1.1xxxxxxx$ 形式

补码尾数规格化后：正数同原码正数 $0.1xxxxxxx$ 形式

负数为 $1.0xxxxxxx$ 形式

进位CF和溢出OF

简单粗暴的判别方式就是：CF对应无符号数，OF对应有符号数（有符号数时，CF的1和0对结果没有作用）

CF和OF都可以想象成溢出

无符号数：转换成十进制就是，两个数操作的结果范围在 $0 - 2^n - 1$ 就是正常，超出范围就是CF进位

有符号数：转换成十进制就是，两个数操作，正的加上正的，如果变为负的就是溢出

第二章 微处理器系统结构

8086/8088微处理器

内部功能结构

从CPU的功能上来说可以分为 总线接口单元（BIU）和执行单元（EU）

编程结构

14个16位寄存器

三大类：通用寄存器、控制寄存器、段寄存器

通用寄存器又分为两组：数据寄存器、指针寄存器和变址寄存器

数据寄存器：

AX（累加器，IO设备也有用途）

BX（基址寄存器，查表指令XLAT）

CX（作计数器+重复前缀REP+数据串操作，或者是LOOP循环的次数）

DX（数据寄存器，一般在除法和乘法中用到）

指针寄存器和变址寄存器：

SP 堆栈指针 BP (堆栈基址指针寄存器)

SI (源变址寄存器)

DI (目标变址寄存器)

段寄存器:

CS 代码段寄存器

DS 数据段寄存器

SS 堆栈段寄存器

ES 附加段寄存器

控制寄存器:

六个状态标志位: CF OF PF SF ZF AF

CF: 进位标志, 判断最高位是否产生进位

OF: 溢出标志, 判断有符号数加减运算

AF: 辅助进位标志, 判断第四位是否向高四位进位

PF: 奇偶标志, 判断低八位中含有的1的个数, 偶数为1, 奇数为0

SF: 符号标志, 和运算结果的最高位一样

ZF: 判读是否为 0

总线周期

T1 T2 T3 T4 四个总线周期

T3 周期一般考的概率最大, 其中有等待状态等等

80X86 微处理器

虚拟内存这个玩意的好处和作用为啥

- 程序不再受现有物理内存空间的限制
- 提高内存利用率
- 实现多道程序, 使更多的程序能够进入内存运行
- 实现多用户、多任务的操作

80386的分页和分段机制的作用是什么

分段的作用: 可以对容量可变的代码存储块或者数据存储块提供模块化和保护性。

分页的作用: 便于实现虚拟存储管理, 通常在内存和磁盘之间进行映射时。

80386的创新在哪方面：32位数据总线 32位地址总线、流水线、MMU

80486的创新点：采用RISC技术、高速缓存、粗发式总线、支持浮点运算、多处理器、五级流水线技术

高速缓存：程序A和程序B的速度相差很大，时间局部性、空间局部性

程序A，访问的顺序是按照存储的顺序进行的，因而空间局部性好，但是每个数据只访问一次，时间局部性好

变量sum，单个变量不考虑空间局部性，但是sum被多次访问，因而时间局部性好

for循环体，循环体按照顺序访问，空间局部性好。被连续执行多次，时间局部性好

程序B，跳着访问的，所以第一条访问顺序这一块，空间不好，时间也不好，其他的和程序A一样

Pentium处理器创新：双流水线技术 U、V

CPU的主要性能指标

CPU的速度

主频、外频、倍频

主频 = 外频 * 倍频

CPU的高速缓存，流水线，总线

数据带宽

缓存

制造工艺

第三章 微处理器的指令系统

指令组成

操作码（位数及位置）、操作数地址（操作数的个数）、操作结果的存储地址、下一条指令的地址

指令长度：定长（RISC一般为定长）和不定长

操作码：操作码的位数取决于计算机的指令系统的规模

指令的数据类型：地址、数值数据、位 位串 字符 字符串、逻辑数据

80806/8088的寻址方式

立即数寻址 `MOV AX, 56H`

寄存器寻址 `MOV AX, BX`

直接寻址 `MOV AX, [1680H]`

寄存器间接寻址 `MOV AX, [BX]`

基址加变址寻址 `MOV [BX + SI], BP`

寄存器相对寻址 `MOV AX, [BX + 54H]`

相对基址加变址寻址 `MOV AX, [BX + SI + 54H]`

通用数据传送

MOV 传送

不影响进位符

`MOV dest, source`

语法格式: `MOV reg/mem/seg, reg/mem/seg/imm`

但是有特殊情况不能操作成功:

`MOV mem, mem`

`MOV seg, seg`

`dest` 不能是 `CS`

`dest` `src` 必须类型匹配

堆栈操作

不影响进位符

`POP AX` IP指针加2

`PUSH AX` IP指针减2

XCHG操作

不影响进位符

`XCHG OPRD1, OPRD2`

`xchg reg/mem , reg/mem`

但是 `xchg mem, mem` 不允许

字节翻译指令 XLAT

MOV BX, 0030H

MOV AL, 5

XLAT

该指令完成的操作为，将待转换的序号转换为代码

$AL \leftarrow [BX + AL]$

例如：想要查找数字 5 对应的七段码

上述三个步骤为先将首地址0030H送入BX，然后加上05H偏移地址（此时数字5就是偏移量）

找到地址以后，执行XLAT，将对应的数值送入AL

目标传送地址 LEA LDS LES

标志位传送指令

I/O数据传送指令

IN 累加器 端口号

这条指令的作用是将端口号中的内容传送到累加器 AL/AX中

端口号可以是8位立即数直接给出，也可以是事先安排在DX中

IN AL, PORT 把端口号PORT中的字节内容送到AL中

IN AX, PORT 把端口号PORT相邻的字内容读入AX中

IN AL, DX

IN AX, DX

OUT 端口号 累加器

作用和 IN 累加器 端口号 相反

算术运算指令

加法指令

ADD d, s ; 会影响 六个标志 不允许两者同时为存储器操作数，目标数不能为立即数

ADC d, s ; $d \leftarrow d + s + CF$ 加上一个进位位

INC d; $d += 1$; 加一操作

INC 指令只会影响 OF SF ZF AF PF 五个标志位

减法指令

SUB d, s; $d = d - s$ 不允许段寄存器的减法，不能同时为存储器操作数 DEST(存储器操作数、通用寄存器)

SBB d, s; $d = d - s - CF$ 连进位位一起减

DEC d; $d = d - 1$; 自减

NEG d; $d = \sim d + 1$; 求补码的指令

CMP d, s; $d - s$ 只置标志位

详见结果不送回dest，但是会根据结果影响六个标志位

乘法指令

MUL s; 无符号乘法指令，完成两个无符号的 8/16 位二进制数相乘，被乘数隐藏在 AL/AX 中
相乘所得的结果位数乘二

对 8 位二进制乘法，高 8 位放在 AH 之中，第 8 位放在 AL 中

对 16 位二进制乘法，高 16 位放在 DX 中，第 16 位放在 AX 中

若运算结果的高字节为有效，即 DX 或者 AX 不为 0，则将 CF 和 OF 同时置为 1

IMUL s; 有符号的乘法指令

除法指令

DIV s; 完成两个无符号的二进制数进行相除，被除数隐藏在 AX（字节除）或者 DX:AX（字除）

对于字节除法 商存在 AL 中，余数存在 AH 之中

对于字的除法，商存在 AX 中，余数存在 DX 中

IDIV s; 有符号的二进制除法

CBW CWD 专门用来为 IDIV 设置的符号扩展指令，用来将被除数字节/字拓展为字/双字节

逻辑运算指令

AND d, s; 与 运算

OR d, s; 或 运算

XOR d, s; 异或 运算

NOT d; 非 运算

TEST d, s; 与 运算 但是不返回结果

移位指令

算术移位

对带符号数进行移位，移位过程中符号位保持不变

SAL 算术左移

SAR 算术右移

逻辑移位

对无符号数进行移位，总是用0填补空出的位

SHL

SHR

循环移位指令

串操作指令

串操作指令是唯一能够在存储器内的源与目标之间进行操作的指令

对字符串进行各种基本的操作：传送(MOVS)，比较(CMPS)，搜索(SCAS)，读(LODS)，写(STOS)

在指令前加上重复前缀：

无条件重复(REP)，相等时重复(REPE)，为0时重复(REPZ)，不等时重复(REPNE)，不为0时重复(RepNZ)

传送(MOVS) 目标串，源串 REP

DS + SI 源串

ES + DI 目标串

CX 作为循环

比较(CMPS) 目标串，源串

REPE REPZ 具有相等的含义，即 $ZF = 1$ 且 $CX \neq 0$ 才会执行

DS + SI 源串

ES + DI 目标串

CX 作为循环

搜索(SCAS) 目标串

REPNE REPZ , 当 $ZF = 0$ 和 $CX \neq 0$ 才会执行

当 $ZF = 1$ 或者 $CX = 0$ 才会停止

读(LODS) 源串

将 SI 指向的字符串读入 AX 或者 AL 之中

写(STOS) 目标串

将AX 或者 AL 中的内容写入由 DI 指向的目标串

程序控制类指令

无条件转移指令 JMP

JMP 目标标号

段内直接转移：目标标号偏移地址 = $(IP) + \text{指令中位移指令}$

例如：JMP ADDR1 ADDR1 与 IP 之间的距离为 1235H, CS = 1500H, IP = 2400H, 执行该指令后, CPU 转移到

物理地址 18638H, 因为当前这个指令也占据了三个字节长度

JMP SHORT ABC

JMP NEAR PTR ABC

段内间接转移

JMP BX IP 直接 转移到 BX 指向的地址

段间直接转移

JMP FAR PTR ADDR2

段间间接转移

JMP DWORD PTR[BX + ADDR3]

设置当前 CS = 1000H, IP = 026AH, DS = 2000H, BX = 1400H, ADDR3 = 020AH

(2160AH) = 0EH, (2160BH) = 32H, (2160CH) = 00H, (2160DH) = 40H, 则执行指令时

目标地址的偏移地址为 320EH 送入 IP, 而段地址 4000H 送入 CS

CALL 过程名

与无条件指令在一起使用

条件转移指令

18条指令

循环控制指令

LOOP 循环指令

中断指令

INT 中断类型

练习题

第一题：若用MB作为PC机的主存容量单位， $1\text{MB} = 220$ 个字节

第二题：运算器在执行两个用补码表示的整数加法是，判断其溢出的规则为：

两个同号的整数相加，若结果的符号位与加数的符号位相反，则一定发生溢出

第三题： $\text{SI} = 0053\text{H}$ ， $\text{BP} = 0054\text{H}$ ，则执行 SUB SI, BP 后，则 $\text{CF} = 1$ ， $\text{OF} = 0$

第四题：实时模式下80486对指令的寻址由 CS ， IP 决定

第五题：下列指令不能清“进位”位的是哪个操作 MOV AL, 0

第六题：下列能使CF标志置1 的指令是 STC 指令

第七题：编程人员不能直接读写的寄存器： IP 指令指针寄存器

第八题：直接、间接、立即三种寻址方式指令的执行速度，由快到慢为 立即寻址、直接寻址、间接寻址

第九题：控制器的功能：从主存中取出指令、分析指令并产生有关的操作控制信号

第十题：8086要输出地址信号是在总线周期的 T1 状态，传输数据是在 T3 状态

最小工作模式的特点为：CPU提供全部信号

最大工作模式的特点为：需要8288总线控制器

第十一题：当存储器读写过慢时，产生一个READY信号与CPU同步，CPU将在总线周期的 T3 时刻采样该信号

如果检测到READY信号为低电平，则插入一个 TW 信号

第十二题：当断电时，能保存数据的存储器件为： ROM （只读存储器）、闪存（一类存储东西，可读可写）

RAM （随机存储器，断电不能保存）

第十三题：浮点数由 尾数、阶码、基数三部分组成

定点数有符号位和数值位两部分组成

浮点数进行相加，需要 对阶、规格化、尾数相加、舍入处理、判断溢出

第十四题：一个较完善的指令系统应包含 数据传送 类指令， 算术运算 类指令， 逻辑运算 类指令， 程序控制类 指令， I/O 类指令， 字符串 类指令， 系统控制 类指令

期末考试

第四章 汇编语言

8086汇编语言伪指令

符号定义伪指令

欲将WordVar的第0个字节值为1，如何处理？去地址WordVar + 1的数据到 AL，又如何处理

方法一：用PTR操作符

```
MOV BYTE PTR WordVar (十六位) , 1
```

```
MOV AL, BYTE PTR WordVar + 1
```

段定义

段定义由SEGMENT

第五章 存储系统

一、存储器的分类

1. 按存储介质分类

半导体存储器：双极型、静态 MOS 型、动态 MOS 型

磁表面存储器：磁盘（Disk）、磁带（Tape）

光存储器：CD，CD-ROM，DVD

2. 按断电后信息的可保存性分类

非易失性存储器：信息可以一直保留，不需要电源维持，ROM，磁表面存储器，光存储器

易失性：RAM，Cache

读写存储器：可读可写

只读存储器：只能读不能写

3. 按存取方式

随机存取存储器 Random Access Memory RAM

顺序存取存储器 Sequential Access Memory SAM

直接存取存储器 Direct Access Memory DAM

相联存储器：Associate Memory CAM

4. 按容量（速度）

寄存器

高速缓存

主存

外存储器

主存储器的性能指标

存储容量

存取速度

二、存储器的组成

1. 存储体

单元和地址线的关系 存储单元数 $N = 2^n$

存储容量 $B = 2^n$

2. 读写过程

读：

CPU先把读单元地址送到AR，经过地址总线送往主存，同时CPU通过控制总线发一个读请求，然后CPU等待从主存储器发来的信号，通知CPU读操作已经完成

存储器通过ready线回答，如果ready信号为1，说明存储字的内容已经读出，并放在数据总线上，送往DR

写：

CPU先将写单元地址经AR送往地址总线，并把写内容字送往DR，同时通过控制总线发出写命令，然后CPU等待写操作完成信号

主存把收到的信息字写入CPU指定的地址后通过ready线发出完成信号

三、存储器连接

1. 扩充技术

位扩充技术：相当于并联，用四块16 K X 2位的芯片拓展成 16 K X 8 位的芯片

字扩充技术：相当于串联，用四块16 K X 8位的芯片拓展成 64 K X 8位的芯片
字位同时扩充

2. 注意问题

CPU的负载能力

各种信号线的配合

CPU时序和存储器存取速度之间的配合

存储器的地址分配和片选信号的产生

四、内存条

SDRAM：同步动态随机读写存储器

DDR

五、外部存储器

1. 磁盘存储器

磁盘容量的计算

公式：

- 有效储存区域=(外直径-内直径)÷2
- 磁道数= 圆柱面数量=有效储存区域×磁道密度
- 总磁道数= 记录面数×磁道密度×有效储存区域
- 每磁道的信息量=内层位密度×内层周长
- 面信息量=磁道数×每磁道信息量
- 非格式化盘组总容量=面信息量×面数
- 格式化容量=总磁道数×每道扇区数×扇区容量
- 平均数据传输速率=最内圈直径×3.14×位密度×盘片转速

- 平均寻址时间=平均寻道时间+平均等待时间

平均等待时间与磁盘转速有关，用磁盘旋转一周所需时间的一半来表示。

- 平均数据传输率= 每道扇区数×扇区容量×盘片转速 或者 最内圈直径×3.14×位密度×盘片转速

例题：

假设一个硬盘有3个盘片，共4个记录面，转速为 7200r/min,盘面有效记录区域 的外直径为30 cm，内直径为 10cm，记录位密度为 250b/mm，磁道密度为8道/mm，每磁道分16个扇区，每扇区512字节，试计算该磁盘的非格式化容量，格式化容量 和数据传输率。

答：

非格式化容量=最大位密度×最内圈周长×总磁道数

最内圈周长=100*3.1416= 314.16mm

每记录面的磁道数=(150-50)×8=800道；

因此，每记录面的非格式化容量=314.16×250×800/8=7.5 M

格式化容量=每道扇区数×扇区容量×总磁道数=16×512×800×4/1024/1024=25 M

硬盘平均数据传输率公式：

平均数据传输率=每道扇区数×扇区容量×盘片转速=16×512×7200/60=960 kb/s

2. 光盘存储器

光盘种类

只读型光盘

具有ROM特性，只读不能写。

只写一次型光盘

只能写一次，写入后不能修改。

可擦写型光盘

可以重复读写。

3. 磁带存储器

特点

- 存储容量大
- 价格便宜
- 替换方便

作用

- 资料保存
- 数据备份
- 数据迁移

4. 固态硬盘

半导体器件（断电保存信息）

DRAM或NAND（闪存芯片）

耗电量低

速度快

DRAM（ns级）、SSD（μs级）、硬盘（ms级）

防震抗摔性

寿命、容量、售价

六、提高DRAM存储器速度

1. 采用芯片内部行缓冲，以提高芯片本身的速度

DRAM容量扩大了很多倍，但速度没有明显提高

2. 双口存储器

双口存储器（能同时进行两个数据的读/写）

两套独立的读/写控制电路、地址缓存、地址译码及地址线和数据线

- 指令预取部件
- 视频缓存

3. 多模块技术

包含多个小体

每个体有其自己的MAR、MDR和读写电路

可同时对多个模块进行访问

4. 连续编址多模块存储器

5. 交叉编址多模块存储器

- 按低地址划分模块
- 各个模块读写周期还是T
- 以T/M间隔轮流启动各模块
- 经过T时间后，在总线上每隔T/M时间就可得到数据
- 吞吐率提高了M倍
- 80386

七、高速缓存

1. 设某计算机的cache采用4路组相联映像，已知cache容量为32 KB，主存容量为16 MB，每个字块有16个字，

每个字64位。请回答：

1. 主存地址多少位？各字段如何划分？
2. 设cache起始为空，CPU从主存单元0. 1. 100依次读出101个字（主存一次读出1个字），并重复按此次序读11次，问命中率为多少？
3. 若cache速度是主存速度的5倍，问采用cache与无cache比较速度提高多少倍？

- (1) **块大小**：每个块 16 字，每字 64 位→8 字节→128 个字节/块→块内地址 7 位；
Cache 32KB： $2^{15}\text{Byte} \rightarrow 2^{15-7} \rightarrow 2^8$ 块，4 路组相联→组内地址 2 位→组地址 6 位（设 $c=6$ ）；
主存：16MB→24 位地址；**主存字块标记**→ $24-7(\text{块大小})-6(\text{分组数})=\text{主存标记位 } 11 \text{ 位}$ （设 t ）；
- (2) 主存的第 0、 2^c 、 2^{c+1} 、...、 2^{10} 块映像到 Cache 的第 0 组，
主存的第 1、 2^c+1 、 $2^{c+1}+1$ 、...、 $2^{10}+1$ 块映像到 Cache 的第 1 组，
主存的第 m 、 2^c+m 、 $2^{c+1}+m$ 、...、 $2^{10}+m$ 块映像到 Cache 的第 m 组($m < 2^c$)

块 0 包含了 0-15 单元，块 1 包含了 16-31 单元，...，块 6 包含了 96-111 单元，因此 0-100 单元在 0-6 块中；

第一次读 0 单元不命中，将 0 块（16 个字）装入 cache 的 0 组→后面 10 次的读 1-7 单元全命中；
以此类推，0-100 单元共 7 个块，共有 7 次不命中，cache 有 64 个组，没有装满，以后的重复访问全部命中。

命中率 $H = (11 \times 101 - 7) / (11 \times 101) = 0.9937$

- (3) $T_a = T_c + (1-H) \cdot T_m = T_m / 5 + (1-H) \cdot T_m = T_m (1.2 - H) = 0.2063 T_m \rightarrow \text{提高倍数 } 1 / 0.2063 = 4.85$

第七章 中断技术

发生中断请求的条件是

一次I/O操作完成、一次DMA操作完成、机器内部发生故障

7.2 CPU与外设数据传送的方式

7.2.1 程序传送

CPU与数据之间的传送在（IN 和 OUT指令的控制之下进行）

无条件传送（同步传送）：

这种传送方式只对固定的外设（开关、继电器、七段显示码等），适合于少量的数据传送

有条件传送（程序查询传送、有条件的异步传送）：

这个条件是在执行输入（IN指令）或输出（OUT指令）前，要先查询接口中状态寄存器的状态

7.2.2中断传送

7.2.3 直接存储器 DMA 传送

直接由硬件电路执行 I / O 交换的传送方式，可以让外设接口直接与内存进行高速的数据传送，不必经过 CPU

操作过程：

- 1. 接口准备就绪，向DMA控制器发送DMA请求
- 2. CPU通过HOLD引脚接收DMA控制器发送的DMA请求
- 3. CPU在完成当前总线操作之后，就会在HLDA 引脚上向DMA控制器发出允许信号而响应总线请求
- 4. DMA控制器接收到此信号就接管了对总线的控制权

内存 ——> 数据总线——> DMA

分 类	程序查询方式	中断方式	DMA 方式
优 点	1.当计算机工作任务较轻或者CPU不太忙时，能够较好的协调外设与CPU之间定时的差别； 2.程序和接口电路比较简单	1.大大提高CPU的工作效率； 2. 可以实现CPU与设备之间的并行操作； 3.可以实现设备与设备之间的并行操作； 4.提高了系统中多台外设的工作效率，让外设能够处于主动申请中断的工作方式	1.提高了CPU的工作效率； 2.让外设接口直接与内存进行高速的数据传送，而不必经过CPU，省去了保护现场之类的额外操作。 3.减少了CPU中断处理的次数
缺 点	CPU必须作程序等待循环，不断测试外设的状态，直至外设为交换数据准备就绪时为止，这种循环方式花费大量时间，降低了CPU的运行效率。	1.在处理“保护现场”和“恢复现场”的过程中需要花费时间，对于高速外设以及成组交换数据的场合不适应。 2.容易造成数据丢失； 3.耗费CPU的时间； 4.中断次数多	有一定的局限性

7.3 中断技术

中断是指CPU正常执行程序时，由于某种随机出现的事件，使CPU暂停运行原来的程序而应更为急迫事件的需要转向去执行中断源服务的程序，待该程序处理完成后，再返回运行原程序，这一控制过程称为中断

中断与中断源：

外设中断源：由CPU的外部事件引发的中断

如：中慢速的键盘、鼠标、打印机等；数据通道，如磁盘、数据采集装置、网络等；实时时钟，如定时器；故障源，如电源停电、外设故障、存储器读出错误

内部中断源：CPU执行中断指令INT n引起的中断；由CPU的运算错误引发的中断，如除数为0、溢出等；为调试程序设置的中断，如单步中断、断点中断

中断源的中断过程

1. 中断源向CPU发出中断请求信号的条件：

中断源通过其接口电路向CPU发送中断请求信号，该信号能否发送给CPU需要满足以下条件

设置中断请求触发器

中断信号，由这个中断请求触发器QA产生

设置中断屏蔽触发器

中断信号能否以中断请求信号（如 INTR）发向CPU，应该能受CPU的控制，以增加中断处理的灵活性

QB为中断屏蔽触发器

2. CPU响应中断的条件

CPU开放中断

CPU采样到 INTR 信号之后是否响应它，由CPU内设置的中断允许触发器（如 IFF）的状态决定。当 $IFF = 1$ ，开中断时才能响应中断，当 $IFF = 0$ ，关中断时，不能响应

CPU在现行指令结束后响应中断

只有在现行指令结束之后才能响应中断（x86的CISC芯片一般是这样的，ARM架构的与之不一样）

3. CPU响应中断及处理过程

关中断

保留断点

保护现场

给出中断入口（地址），转入相应的中断服务程序

恢复现场

开中断与返回

7.4 8086、8088的中断系统和中断处理

外部中断

外部中断是由外部硬件或者外设接口产生的，包括可屏蔽与非屏蔽中断

可屏蔽中断：

用户可以用指令禁止和允许的外部硬件中断，由8086 CPU的 `INTR` 引脚进入

当 `INTR` 引脚上出现一个高电平有效信号时，它必须保持到当前指令的结束，这是因为CPU只在每条指令的最后一个时钟周期才对 `INTR` 引脚的状态进行采样。

接下来还要看中断允许标志位 `IF` 的状态，`IF = 1`，开中断，可以响应中断；`IF = 0`，关中断，不响应中断；

`IF` 的状态可以使用指令 `STI` 让其置位，即开中断；使用 `CLI` 使其复位，即关中断

一般在执行程序的过程中要用 `STI` 使其开中断，以便响应新的可屏蔽中断请求

非屏蔽中断：

非屏蔽中断可以无视标志寄存器 `IF` 的值，

内部中断

除法出错中断	类型 0
溢出中断	类型 4
单步中断	类型 1
断电中断	类型 3
用户定义的软件中断	类型 n

- 内部中断由一条 `INT n` 指令直接产生，其中断类型号 `n` 或者包括在指令中，或者已由系统预先定义好
- 除单步中断外，所有内部中断都不能被屏蔽
- 所有的内部中断都没有中断响应 `#INTA` 机器总线周期，这是因为内部中断不必通过查询外部来获得中断类型号

优先级	中断名	中断类型	说明
由	除法错	类型 0	商大于被除数（软件中断）
高	INT n	类型 n	内部检查用中断（软件中断）
到	INTO	类型 4	溢出用（软件中断）
低	NMI	类型 2	非屏蔽中断（硬件中断）
排	INTR	由外设送入	可屏蔽中断（硬件中断）
序	单步	类型 1	调试用（软件中断）

中断向量表

将 256 个中断向量制成了一张中断向量表

每个中断类型对应一个中断向量，每个向量包含 4 个字节，2 个低地址字节是 IP 偏移量，2 个高地址字节是 CS 段地址。存储在 000H - 3FFH，一共 1 MB 的大小

外设通过一个接口将一个 8 位的中断类型码放在数据总线，CPU 对编号 n 乘以 4 得到该中断向量的首字节

可屏蔽中断的处理全过程

- ① 中断请求信号 INTR 有外部设备接口电路产生并送至 8086 的 INTR 引脚上
- ② CPU 是否响应取决于 CPU 内部的 IF 标志，如果 IF 标志为 0，则在 IF 变成 1 之前 CPU 不会识别中断；当 IF = 1 并出现 INTR 请求信号时，CPU 在完成正在执行的指令后，便开始响应中断。
- ③ CPU 响应中断时，首先从外设接口电路读取中断类型号 n。CPU 将通过其 #INTR 引脚向中断接口电路发送响应信号，并启动中断过程；这个响应信号将使发出中断请求的接口 把其 1 字节的中断类型号通过数据总线发送给 CPU。
- ④ 按先后顺序把 PSW、CS 和 IP 的当前内容压入堆栈，以保护现场与断电。
- ⑤ 清除 IF 和 TF 标志，禁止在中断响应过程中有其他可屏蔽中断进入，也禁止单步中断。
- ⑥ 取中断向量新值，把 $4 \times n$ 的字存储单元内容读入 IP 中，把 $4 \times n + 2$ 的字存储单元中的内容读入 CS 中。
- ⑦ CPU 从新的中断向量 CS: IP 值得到中断入口地址，开始转入中断服务程序。
- ⑧ 若允许中断嵌套，则一般在中断服务程序保存各寄存器内容之后安排一条 STI 开放中断指令，这是因为 CPU 响应中断后便自动清除了 IF 与 TF 位，当执行了 STI 指令后，IF = 1，以便更高优先权的中断源获准中断响应。
- ⑨ 在中断服务程序结尾安排一条 IRET 中断返回指令，把保存在堆栈中的原 IP、CS 与 PSW 等值依次弹出堆栈。
- ⑩ 由弹出的原中断向量 CS: IP 控制 CPU 返回到发生中断的断点处去。

7.5 中断控制器 8259A

引脚及其功能：

IR0 - IR7：8级中断请求输入端。用于接收来自外设的中断请求。在主从级联的系统中，主片的 **IR0 - IR7** 端分别与各从片的 **INT** 端相联，用来接收来自从片的中断请求

内部主要组成部分：

- 数据总线缓冲器
- 读写逻辑
- 级联缓冲器 比较器
- 控制逻辑
- 中断请求寄存器 (**IRR**) interrupt requirement register
- 中断服务寄存器 (**ISR**) interrupt service register
- 中断屏蔽寄存器 (**IMR**) interrupt mask register
- 优先级判别器 (**PR**) priority register

例题：

在 **IBM PC/XT** 系统中，只用一片 8259 A中断控制器，用来提供8级中断请求，其中 **IR0** 优先级最高，**IR7** 优先级最低。它们分别用于日历时钟中断、键盘中断、保留、网络通信、硬盘中断、软盘中断和打印机中断。设8259 A的 **ICW2** 高5位 **T7~T3** = 00001，对应的中断类型码为 **08H ~ 0FH**；片选地址为 **20H、21H**。8259 A的初始化：

初始化

```
1  MOV AL, 13H      ;写ICW1, 单片, 边缘触发, 需要ICW4
2  OUT 20H, AL
3  MOV AL, 8        ;写ICW2, ICW2中断类型码, 从08H开始
4  OUTB 21H, AL
5  MOV AL, 0DH      ;写ICW4, 缓冲工作方式, 非特殊的嵌套方式
6  OUT 21H, AL
7
8  MOV AL, 0        ;写OCW1, 允许IR0~IR7全部8级中断请求
```

送中断向量

```
1  根据中断源的中断类型码送中断向量，例如异步通信中断IR4，其中断类型码为8+4 = 12 (0CH)
2  则中断向量的偏移量(IP值)与段地址(CS)在中断向量表中的存放地址为12x4 = 48(30H)、31H、32H、33H
3  其中30H、31H存放指令指针IP，32H、33H存放指令段码CS
```

中断子程序结束

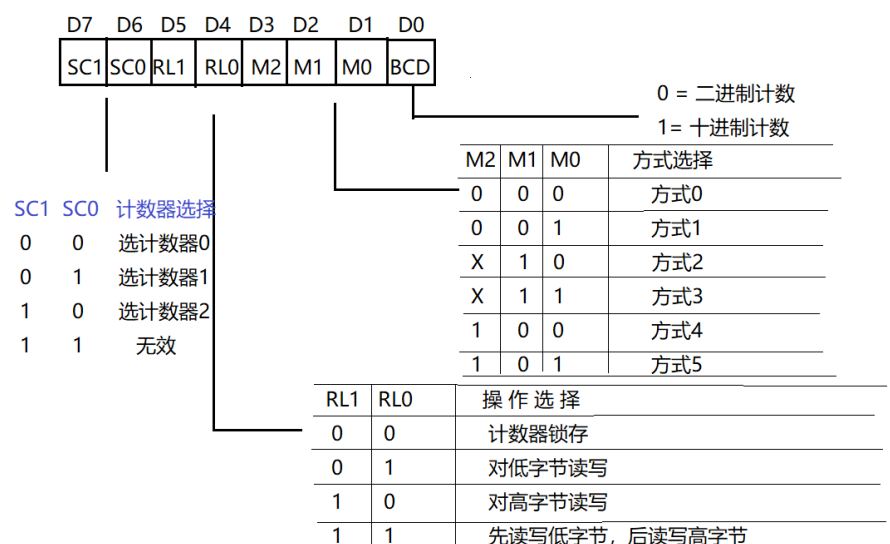
- 1 由于8255A采用中断工作方式，且ICW4中的D1位(即AEOI)为0，这意味着采用正常结束中断，因此，在中断子程序结束前必须发EOI命令和IRET命令
- 2 MOV AL, 20H ;写OCW2命令，使ISR相应位复位
- 3 OUT 20H, AL
- 4 IRET

第八章 可编程接口芯片

8.1 可编程计数器/定时器8253-5

作用是，计数与定时

8253-5的控制字格式如下：



举例应用：

三个计数器的用途如下：

计数器0

计数器0向系统日历时钟提供定时中断，它选用方式3工作，设置的控制字为36 H。计数器值预置为0（即65 536），GATE0，接+5 V，允许计数。因此OUT0输出时钟频率为 $1.19 \text{ MHz} / 65536 = 18.21 \text{ Hz}$ 。它直接接到中断控制器 8259 A 的中断请求 IR0 端，即0级中断，每秒出现18.2次。因此，每间隔55 ms 产生一次 0 级中断。并且，每一个输出脉冲均以其正跳变产生一次中断。

计数器1

计数器 1向 DMA 控制器定时发送动态存储器刷新请求，它选用方式2工作，设置的控制字为54 H。计数器初始值为18，GATE1 接 +5 V，允许计数。因此OUT1输出分频脉冲频率为 $1.19 \text{ MHz} / 18 = 66.1 \text{ kHz}$ ，相当于周期为15.1 us

计数器2

计数器2控制喇叭发声音调，采用方式3工作，设置的控制字为 B6H，计数器的初值置为533 H（即1331），OUT2输出方波频率为 $1.19 \text{ MHz} / 1331 = 894 \text{ Hz}$

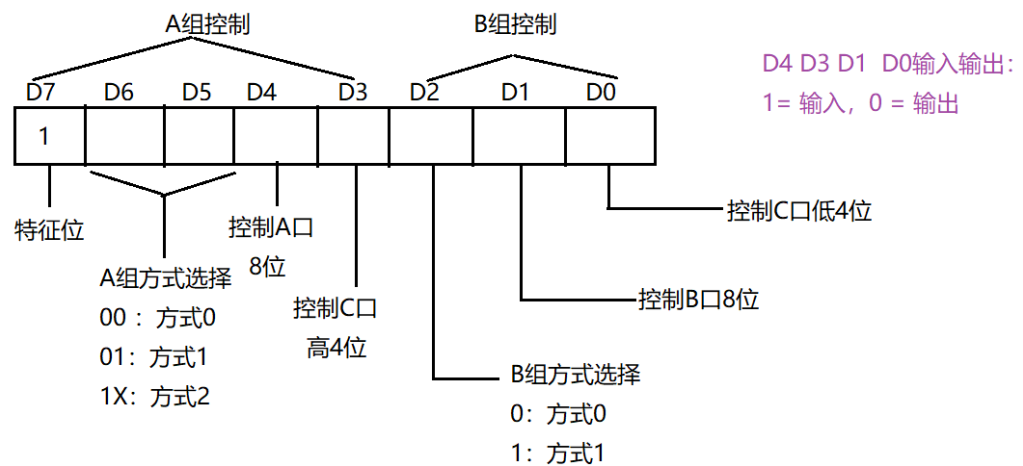
计数器预置程序

```
1  PR0:MOV AL, 36H           ;选择计数器0, 读写双字节(即高低都有 11), 方式3, 二进制计数
2      OUT 43H, AL
3      MOV AL, 0             ;预置计数值为65 536
4      OUT 40H, AL           ;先送低字节计数值
5      OUT 40H, AL           ;后送高字节计数值
6  PR1:MOV AL, 54H
7      OUT 43H, AL
8      MOV AL, 12H
9      OUT 41H, AL
10 PR2:MOV AL, 0B6H
11     OUT 43H, AL
12     MOV AX, 533H           ;送分频数1553
13     OUT 42H, AL           ;先送低字节
14     MOV AL, AH
15     OUT 42H, AL
```

8.2 可编程并行通信接口芯片8255A

8255A的控制字

方式选择控制字



例题: 若要将 8255A 设定为A口为方式0输入, B口为方式1输出, PC7-PC4 位输出, PC3-PC0 为输入。设 8255A 的4个端口地址范围为 0060H ~ 0063H, 则初始化编程的程序如下

```
1  MOV DX, 0063H
2  MOV AL, 10010101B
3  OUT DX, AL
```

端口C置位/复位控制字



若要使8255 A的PC 5 初始状态置位1， 设825 A端口地址范围为300 H ~ 303 H， 则设置端口 C 置位/ 复位控制字如下：

```

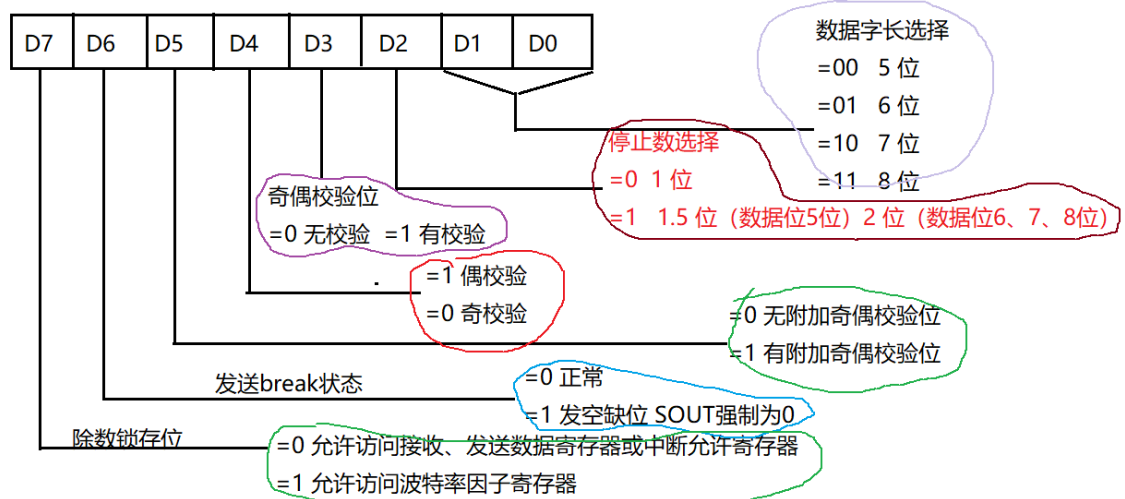
1  MOV DX, 303H
2  MOV AL, 00001011B
3  OUT DX, AL

```

8.3 可编程串行异步芯片8250

8250芯片中有很多个寄存器，其中比较重要的是 LCR（线路控制寄存器 3FBH）和 DLR（波特率因子寄存器 3F8H, 3F9H）

LCR设置的通信数据格式



计算波特率因子的公式：

波特率因子 = 1.8432 MHz / (波特率 X 16)

波特率和除数对照表

十进制	十六进制	波特率
1047	417	110
768	300	150
384	180	300
192	c0	600
96	60	1200
48	30	2400
24	18	4800
12	0c	9600

设 8250 串行接口芯片的外部的时钟频率为 1.8432 MHz，如果要求数据格式为 7 位，2 位停止位，偶校验，8250 的工作波特率是 19200，设线路控制寄存器、波特率因子寄存器分别是 2FBH、2F9H、2F8H，试编写对应的初始化程序段。

```
1  波特率为19200，计算出波特率因子为6
2  数据格式这一块为00011110 = 1EH，为什么D7为0，可以访问接收、发送寄存器？
3
4  MOV DX, 2FBH    ;设置LCR数据格式
5  MOV AL, 1EH
6  OUT DX, AL
7
8  MOV AL, 80H     ;开始设置波特率这一块的内容
9  OUT DX, AL
10 MOV DX, 2F8H
11 MOV AL, 06H
12 OUT DX, AL
13 INC DX          ;再送2F9H端口
14 MOV AL, 0
15 OUT DX, AL
```

第九章 总线系统

总线由来

总线的组成

微机系统总线

PCI、SATA、SAS、USB、1394各自特点

PCI Express：每个设备都有专用连接，独享带宽

SATA：1. 点对点串行连接，独享带宽、2. 两对低电平差分信号线，3. 3根地线、具备热插拔能力

SAS：

USB：同样性，连接方便，数据传输速度快，自备电源

总线有标准，谁来定呢

冯诺依曼的三种总线结构：

总线组成：数据、地址、控制

总线优先级管理：

总线通讯：

同步通信和异步通信

并行通信和串行通信（当下的趋势？）

总线驱动：

连接的部件数 功率 距离

总线性能：

数据宽度：8位、16位、64位、128位、256位

速度：数据传输率

总线结构：

即插即用：

解决资源冲突问题

自动资源分配

中断号

DMA号

I / O 地址空间

存储器缓冲区

热插拔：(Hot Swap)

不关系统，保证系统不间断运行

能够实时侦测及使用新的设备

BIOS 及OS 支持

外设接口 - ATA：

外设接口 - SATA

外设接口 - SCSI

PCI 总线是一个与处理器无关的高速外围总线

PCI 总线的基本传输机制是猝发式传送

第十章 控制器原理