

《微机原理与应用 I》复习提要

一、概述

1. 整机概念

冯诺依曼计算机核心：

- (1) 指令和数据用二进制表示
- (2) 程序预存在存储器中，在执行时会将指令自动的逐条取出并分析执行
- (3) 硬件由运算器、存储器、控制器、输入输出设备五部分组成

计算机由 CPU、存储器 M、I/O 接口及三总线（数据总线 DB、地址总线 AB、控制总线 CB）组成。各总线的特点与作用。

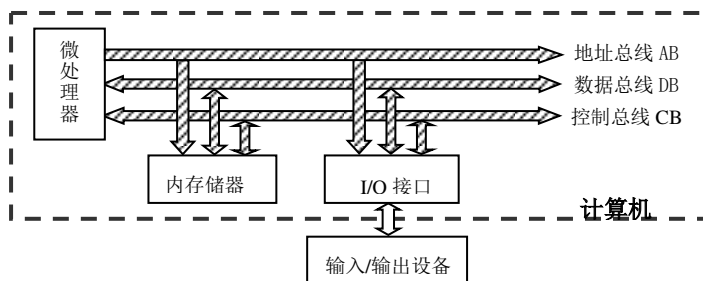


图 1. 计算机硬件结构

数据总线 DB 数量决定了处理数据的位数，也决定了处理器的位数
地址总线 AB 数量决定了寻址的范围

常用术语

第一页 共三页

位(Bit):

是计算机所能表示的最基本最小的数据单位。每一位只能有两种状态“0”或“1”。Bit是Binary Digit的缩写。

字节(Byte):

一个8位二进制数称为一个字节。字节是计算机中最基本的数据表示单位。常见的微处理器的位数均是8的倍数。如：8位微处理器，16位微处理器，32位微处理器，64位微处理器等。在计算机中的基本存储单元内容用字节表示。

字(Word):

字是微处理器内部数据传输、处理的基本单位。目前计算机中一般将2个字节定义为1个字，即一个字为16位二进制数。数据类型分为：字节、单字、双字(4字节组成)、四字(8字节组成)等。

字长:

字长指字的二进制数的位数。它是数据总线宽度。8位微处理器的字长为8位；32位微处理器的字长为32位。

2. 计算机中数和码的表示

a. 进制表示及相互之间的转换：二进制数、十进制数、十六进制数。

数制转换举例：

例1:将 87.65 转换成8位二进制整数、4位二进制小数。

解：

整数部分：

2)	余数	低位
2) 87		
2) 43	1	
2) 21	1	
2) 10	1	
2) 5	0	
2) 2	1	
2) 1	0	
	0	1 高位

小数部分：

x	整数	高位
x 0.65		
x 2	1	
x 1.30		
x 2	0	
x 0.60		
x 2	1	
x 1.20		
x 2	0	
	0	低位

所以整数部分： $(87)_{10} = (01010111)_2$
小数部分： $(0.65)_{10} = (0.1010)_2$
所以： $(87.65)_{10} = 01010111.1010B$

b. 有符号数的表示及相互之间的转换：真值、原码、反码、补码

原码：最高位为符号位，其余为数值位的二进制数，表示范围： $1-2^{(n-1)} \sim 2^{(n-1)}-1$ ，如 $n=8$ 时表示的范围为：-127~127

反码：正数反码与原码相同，负数反码等于原码的符号位不变，其他位取反。表示范围： $1-2^{(n-1)} \sim 2^{(n-1)}-1$ ，如 $n=8$ 时表示的范围为：-127~127

补码：正数补码与原码相同，负数补码等于原码的符号位不变，其他位取反后加 1。表示范围： $-2^{(n-1)} \sim 2^{(n-1)}-1$ 如 $n=8$ 时表示的范围为：-128~127

判断溢出的方法：

(1) 双进位法：

2个进位位分别为次高位向最高位的进位和最高位向进位位的进位。如果两个进位均有或均无则无溢出。如果两个进位中1个有进位而另1个无进位则一定有溢出。

(2) 符号判断法：

同号相减无溢出，
同号相加时结果符号与加数符号相反有溢出，相同则无溢出。
异号相加无溢出，
异号相减时结果符号与减数符号相同有溢出，相反则无溢出。

1. 原码、反码、补码相互转换

$[正数]_{原} = [正数]_{反} = [正数]_{补}$
 $[负数]_{反}$ 为 $[负数]_{原}$ 符号位不变其它位求反
 $[负数]_{补}$ 为 $[负数]_{原}$ 符号位不变其它位求反加1

2. 补码运算的运算规则：

$$\begin{aligned} [X]_{原} &= [[X]_{补}]_{补} \\ [X \pm Y]_{补} &= [X]_{补} \pm [Y]_{补} \end{aligned}$$

c. 编码的表示：非压缩型 BCD 码、压缩型 BCD 码、ASCII 码

BCD 码的表示：

BCD码就是用四位二进制数表示一位十进制数，它主要有两种表示形式：

非压缩型-- 用1个字节表示1位十进制数，高四位清0。

压缩型-- 用一个字节表示两位十进制数。

注意：

1. 计算机并不能识别BCD码,只是程序员自己明白;
2. 有些指令对BCD码的存放形式有要求。

下表给出了压缩和非压缩BCD码的对照,仅供参考:

十进制	压缩型(B)	非压缩型(B)
8	00001000	00001000
17	00010111	00000001 00000111
623	00000110 00100011	00000110 00000010 00000011

用美国标准信息交换代码，即ASCII码，也就是用7位二进制数对字符进行编码。

例如：

'1'----> 31H	'a'----> 61H
'2'----> 32H	'b'----> 62H
'9'----> 39H	'z'----> 7AH
'A'----> 41H	回车符CR-->0DH
'Z'----> 5AH	换行符LF-->0AH

二、CPU

1. 8086/8088 微处理器

a. 结构与工作原理

8086/8088 微处理器的内部编程结构分为哪两部分，两部分各自执行的功能以及这种机构的优点。、
14 个 16 位寄存器，（8 个 8 位寄存器）

逻辑地址与物理地址的关系：

第三页 共十页

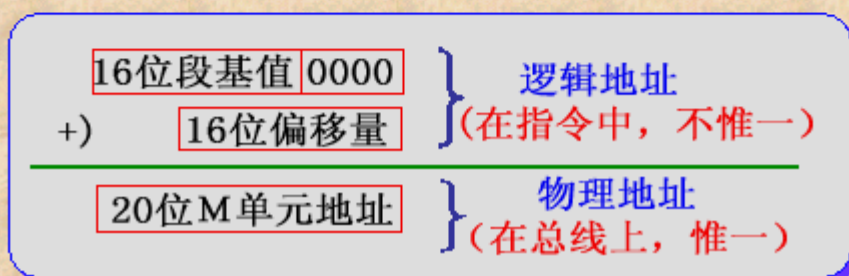
80x86工作在实模式下,存储器是分段管理的,每个存储单元都有两种地址:

物理地址----

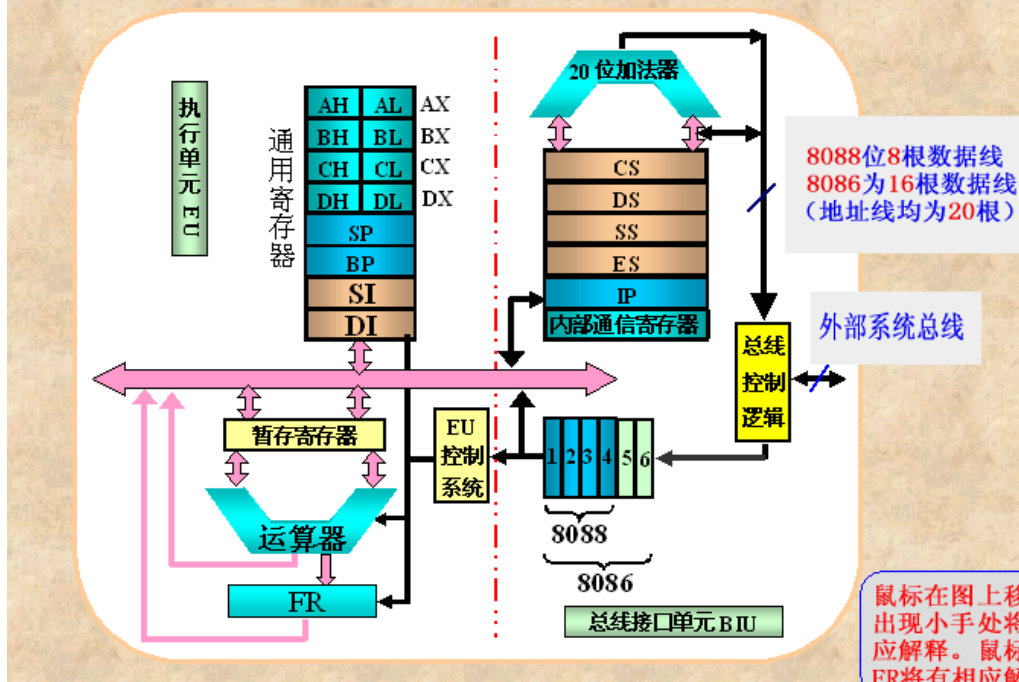
由20位二进制数组成,它的范围是00000H~FFFFFH。CPU与存储器M之间的任何信息交换都是使用物理地址,它是唯一的。

逻辑地址----

是由**段基值**和**偏移量**两部分组成,它们都是无符号16位二进制数。段基值表示存储单元所在段在内存中的位置,偏移量则指出存储单元在段中的位置。逻辑地址是不唯一的。



微处理器8086、8088结构类似,按其功能可以分为两大部分:
总线接口单元(BIU)和**执行单元(EU)**,如图:



b. 内部寄存器

8086/8088 有 14 个十六位寄存器各为哪些及用法, 比如:

- 哪些寄存器可以拆成 8 位寄存器使用
- 哪些寄存器可以存放偏移地址信息
- 哪些寄存器可以存放段基值信息
- 哪些寄存器可以存放端口地址
- 哪些寄存器可以描述栈区
- 哪些寄存器可以描述指令所在位置
- 16 位的 FR 寄存器中有 9 位有效位，其中 6 位是状态标志，3 位是控制标志，状态标志的各位在什么情况下置 1。
- ○○○○○○

c. 工作模式与引脚

- 8086/8088 有哪两种工作模式，两种模式的主要区别。

最小工作模式---

1. 构成单微处理器的简单系统；
2. 全部信号线均由 8086/8088 CPU 提供；
3. $\overline{MN}/\overline{MX}$ 接 +5V。

最大工作模式---

1. 构成多 CPU 工作方式的复杂系统；
2. 部分控制信号线由总线控制器 8288 提供，而不是全部由 CPU 直接提供；
3. $\overline{MN}/\overline{MX}$ 接地。

-
- 8086 与 8088 的区别。

8086 与 8088 的主要区别在于：

1. 8086 内部数据总线与外部数据总线均为 16 位，8088 内部数据总线为 16 位而外部数据总线为 8 位；
2. 8086 内部指令队列为 6 字节，当有 2 个字节空时 BIU 开始取指令；8088 内部指令队列为 4 字节，当有 1 个字节空时 BIU 开始取指令；
3. 8086 与 8088 大部分信号引脚定义相同，主要区别在于：
 - 第 28 脚，8086 为 $\overline{M}/\overline{IO}$ ，是对存储器或 I/O 操作的控制线，8088 为 $\overline{IO}/\overline{M}$ ，其作用正好与 8086 的 $\overline{M}/\overline{IO}$ 相反。
 - 第 34 脚，8086 为 \overline{BHE}/S_7 ，是高 8 位数据线有效信号线/状态线，8088 为 SS_0 ，是用于确定当前总线周期的读/写操作线。

-
- 8086/8088 为 40 引脚 DIP 芯片，采用复用技术。8086 主要引脚包括：AD0 ~ AD15, A16/S3 ~ A19/S6, $\overline{MN}/\overline{MX}$ 、ALE、 \overline{BHE} 、 \overline{RESET} (CPU 的初始状态)，读 RD、写 WR、存储器 I/O 控制信号 $\overline{IO}/\overline{M}$ ($\overline{M}/\overline{IO}$) 等。采用分时复用技术的信号如何分离？如何获得存储器读、存储器写、IO 读、IO 写等信号？

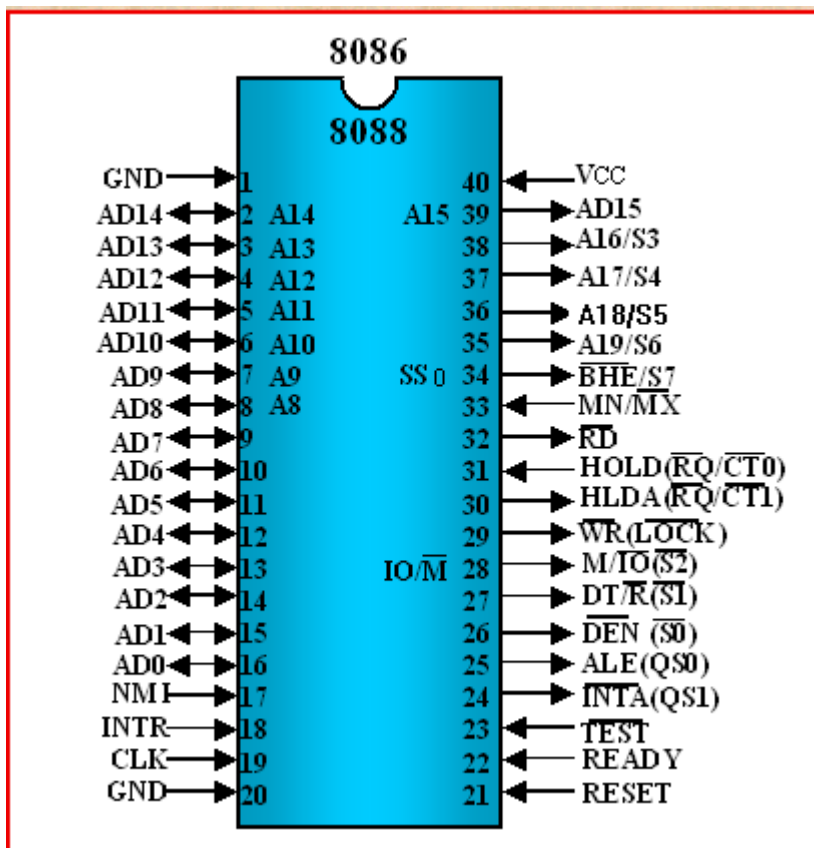
。在实模式下工作，微处理器仅可以寻址第一个1M字节存储空间，即便是Pentium微处理器也是一样。实模式又可分为以下两种工作模式：

最小工作模式---

1. 构成单微处理器的简单系统；
2. 全部信号线均由 8086/8088 CPU 提供；
3. MN/MX接+5V。

最大工作模式---

1. 构成多CPU工作方式的复杂系统；
2. 部分控制信号线由总线控制器8288提供，而不是全部由CPU直接提供；
3. MN/MX接地。



BHE/S₇ 高8位数据选通线/状态，三态输出线。
BHE与A₀组合形成数据线有效形式：

BHE	A ₀	所用数据线	操作
0	0	AD15—AD0	传输 16 位字
0	1	AD15—AD8	传输 8 位字节
1	0	AD7—AD0	传输 8 位字节
1	1	无	无

S₇ 没有定义，作为备用。

d. 存储器组织

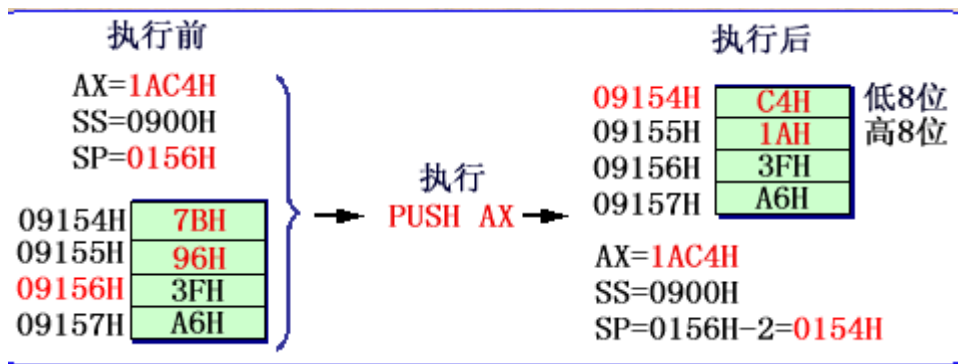
分段管理的概念，逻辑地址和物理地址，物理地址的形成，字节和字的地址，8086 的存储器分奇偶两个存储体。

对于8086有16位外部数据总线，可以一次传输两个字节数据。但是，为了能使8086即可传输8位数据，又可传输16位数据。8086对存储器1M空间用奇存储体（即奇地址存储体）和偶存储体（即偶地址存储体）两块存储体组成。两块存储体分别用BHE和A₀控制，两存储体可同时传输数据，达到传输16位数据的目的，也可以只一块存储体传输8位数据。

另外,在地址线方向,还需要BHE/S₇线,它与A₀一起做M的选通。A₀=0，则选通低8位存储体或说是选通偶存储体；BHE=0，则选通高8位存储体或说是选通奇存储体。

e. 堆栈组织

堆栈的构造、堆栈段寄存器 SS、栈指针 SP、栈操作（指令 PUSH 和 POP 的操作对指针的影响）、堆栈原则。



f. 时序*

什么是时钟周期，总线周期，等待周期。最基本的总线周期由 4 个时钟周期组成。

2. 8086/8088 寻址方式

a. 指令包含信息：操作码+操作数

b. 寻址方式，即寻找操作数位置的方式，操作数在计算机中所处的位置不同，寻址方式不同。操作数在计算机中的位置可以为：

- 在指令中-----立即数寻址
- 在 R 中-----寄存器寻址

- 在 M 中-----段 R: EA，其中：
$$EA = [\begin{matrix} BX \\ BP \end{matrix} + \begin{matrix} SI \\ DI \end{matrix} + \begin{matrix} 0 \\ n16 \end{matrix}]$$

● 在 I/O 端口中-----分：直接寻址（如：IN AL,n8）和 R 间接寻址（如：IN AL,DX）。
注意：寻址方式中的隐含规定

EA中只能使用寄存器**BX**、**BP**、**SI**、**DI**，且**BX**和**BP**不能同时使用，**SI**和**DI**不能同时使用。

如果**EA**中使用寄存器**BP**，则段基值隐含使用堆栈段寄存器**SS**。如果没有使用寄存器**BP**，则隐含使用**DS**。

隐含使用的段寄存器可以缺省不写，否则必须明确写出。

80x86 采用I/O接口与存储器独立编址的方式，对接口操作指令只有数据输入**IN**指令和数据输出**OUT**指令。寻址范围为**64K**字节(**8**位端口)，也可为**32K**字(**16**位端口)。

I/O端口不分段,直接把输入/输出指令中给出的端口地址送上地址总线，要传送的数据**只能**通过累加器**AX**(传送**16**位数据)或**AL**(传送**8**位数据)。

端口地址可以是**8**位，也可以是**16**位，**8位口地址**是可以采用**直接寻址**（即：指令中直接给出**8**位口地址）；**16位口地址**需要采用**DX**间接寻址方式。

直接转移通常叫做**远转移**(far jump)，因此它能够转移到任何地方。

术语**相对**意味着**相对于当前指令指针**(即为取出该转移指令之后的**IP**值)，且是**段内**转移。

此指令书写为：**JMP 目标地址**

如果用16位寄存器或16位存储器存放JMP指令的目标地址，则是近转移(即:段内转移)。如果用32位存储器保存目标地址，则这种转移是远转移(即:段间转移)。

3. 8086/8088 指令系统

了解指令的格式、特点、用法、对 FR 的影响。常用指令要熟练掌握。主要指令包括：

a. 数据传送指令（MOV、LEA、PUSH、POP、IN、OUT、XCHG、XLAT、LDS、LES 等）

功能: 即把某指定的内存单元的有效地址EA(即偏移量)送到指定的寄存器中。

格式: `LEA dest, src ;`

操作: 将src这个源操作数的偏移量传送到dest。

说明: 1. 在此传送的不是存储器中的操作数,而是存储器单元的偏移地址的有效值。
2. 其中: dest -- 必须是通用寄存器之一。
src -- 必须是内存操作数。
3. 对FR无影响。

比较指令(CMP)

功能:与SUB指令一样执行减法操作，但它并不保存结果，只是根据结果设置条件标志位，后面往往跟着一条条件转移指令，根据比较结果选择不同的程序分支。

格式: **CMP** dest, src

操作: dest - src

说明: 1. 源、目标操作数来源同加法指令。
2. 影响FR中**AF、OF、SF、PF、CF、ZF**。
3. 比较指令执行后不影响源、目标操作数值，只影响**FR**。
当 src = dest, 则**ZF=1**,
当 src > dest, 则**CF=1**。
常通过判断状态决定是否转移。

b. 算术运算指令 (ADD、ADC、INC、AAA、DAA、SUB、SBB、DEC、AAS、DAS、CMP、NEG、MUL、IMUL、DIV、IDIV 等)

ADD和ADC指令是对二进制操作的,如果两个BCD码相加结果不是BCD码.要想得到BCD码,需要进行十进制调整。80x86 CPU提供了两条指令，分别对非压缩型BCD码和压缩型BCD码加法操作后进行BCD码调整。

功能:

AAA ; 对**非压缩型BCD**码加法操作进行
; BCD码调整。

DAA ; 对**压缩型BCD**码加法操作进行
; BCD 码调整。

c. 逻辑运算指令 (AND、OR、NOT、XOR、TEST)

- d. 位移指令（SHL、SHR、SAL、SAR、ROL、ROR、RCL、RCR）
- e. 串操作指令（MOVS、CMPS、SCAS、LODS、STOS 及重复前缀 REP、REPE、REPNE 等）
- f. 控制转移指令（LOOP、JMP、Jxx、CALL/RET、INT/IRET 等）

条件转移类指令说明

格式:

Jxx dest	;	操作码(1字节)	位移量disp (1字节)
----------	---	----------	---------------

操作:

其中J后面的xx是指"条件",目标地址dest的IP'=IP当前值+位移量disp。

说明:

1. 这类指令当条件成立或称满足条件时转移到dest处, 否则顺序执行程序。这个条件均是FR中的一个或几个状态位作为判断依据。
2. 条件转移指令的操作数常为一个短标号, 转移的位移量disp是8位二进制补码, 转移距离是相对于当前IP的-128~+127个字节。

过程调用的结构：



调用指令执行时将自动完成：

1. 将断点处CS:IP自动压栈保护；
2. 将子程序或过程的入口地址装入CS和IP中。

INT n ; 其中n为中断类型码,可为0至255中任意一值,n确定了,服务程序入口地址即可找到。

功能: 1. FR进栈, IF、TF清零;
2. 断点进栈(断点的CS进栈, IP进栈);
3. 将服务程序入口地址放入CS和IP

操作: $SP \leftarrow SP - 2$, $SS:[SP] \leftarrow FR$, $IF \leftarrow 0$, $TF \leftarrow 0$,
 $SP \leftarrow SP - 2$, $SS:[SP] \leftarrow CS$, $SP \leftarrow SP - 2$, $SS:[SP] \leftarrow IP$,
 $IP \leftarrow 0000H:[4n]$, $CS \leftarrow 0000H:[4n+2]$

INTO ; 溢出中断,无操作数,相当于INT 4, 即n=4

功能: 检测OF,如果OF=1,启动类似于INT n的中断过程,否则无操作。

操作: $SP \leftarrow SP - 2$, $SS:[SP] \leftarrow FR$, $IF \leftarrow 0$, $TF \leftarrow 0$,
 $SP \leftarrow SP - 2$, $SS:[SP] \leftarrow CS$, $SP \leftarrow SP - 2$, $SS:[SP] \leftarrow IP$,
 $IP \leftarrow 0000H:[0010H]$, $CS \leftarrow 0000H:[0012H]$

IRET ; 中断返回指令,放在中断服务程序末尾。

功能: 1.断点出栈(IP出栈,CS出栈) 2.FR出栈

操作: $IP \leftarrow SS:[SP]$, $SP \leftarrow SP + 2$,
 $CS \leftarrow SS:[SP]$, $SP \leftarrow SP + 2$,
 $FR \leftarrow SS:[SP]$, $SP \leftarrow SP + 2$ 。

g. 处理器控制指令（CLC、STC、CLD、STD、CLI、STI、CMC、NOP、HLT 等）

4. 汇编语言程序设计

a. 汇编语言源程序的基本结构

b. 伪指令，要求熟练掌握如下伪指令：

- 常数，变量，标号
- 分析运算符：OFFSET、SEG
- 综合运算符：PTR
- 数据定义：DB、DW、EQU
- 段或过程定义：SEGMENT、ENDS、ASSUME、ORG、PROC、ENDP、END、\$

① **SEG** : 加在变量名或标号前面，将得到变量名或标号的段基值。

② **OFFSET** : 加在变量名或标号前面，将得到变量名或标号的偏移量。

PTR : 指定由地址表达式确定的存储单元类型（**BYTE** 字节、**WORD** 字、**DWORD** 双字、**NEAR** 近、**FAR** 远）。

格式：类型 **PTR** 地址表达式

c. DOS 系统功能调用（INT 21H），要求掌握功能号为：01H，02H，09H，0AH，25H，4CH 等的用法。

d. 汇编语言源程序结构包括：顺序程序，分支程序，循环程序，子程序，能按要求编写简单程序。

三、存储器 M

1. 存储器的分类及主要技术指标

2. 存储器扩充接口设计

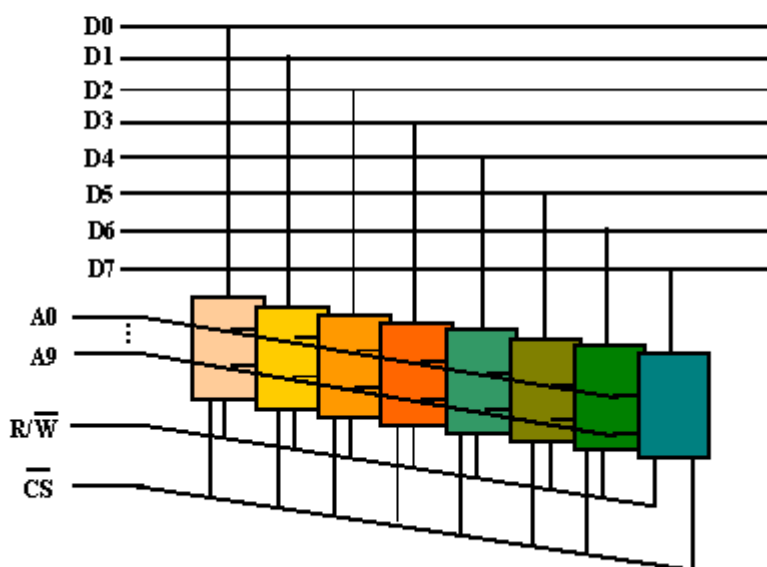
要求：在给出 CPU、工作方式、存储器芯片、起始地址等的条件下，设计并画出 M 扩充接口电路原理图，可分四步实现：

- a. 选片（确定 M 芯片的类型和数量）
- b. 地址分配（应可说出每片存储器芯片所占的地址范围）
- c. 地址译码（译码电路的设计）
- d. M 与 CPU 信号连接（包括：DB、AB、CB）

注意：8086 和 8088CPU 的区别，最大和最小工作方式的差别。

a. 芯片的扩充—位数的扩充

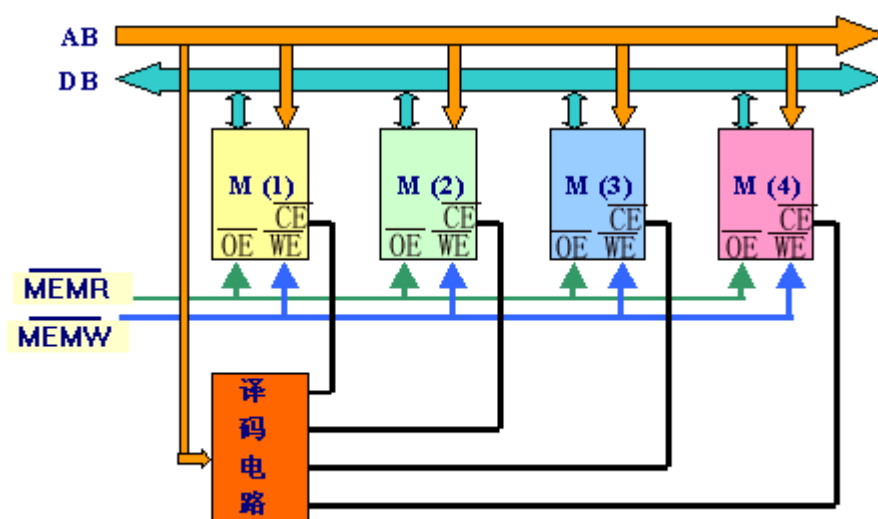
如图所示,每个芯片地址A9--A0和控制线R/W, \overline{CS} 复接在一起,而数据线分别接至数据总线的各位。如果用1K*4位的存储器芯片来组成1K字节的存储器,只需2片就可以了。



1K*4位的芯片组成 1K*8 的存储器

b. 芯片的扩充—容量的扩充

用容量较小的存储器芯片组成容量较大的存储器,可用地址串联法.即用若干块小容量的芯片接在一起构成一个需要的存储器容量.每个芯片的地址要经过译码器分配,以免重叠.如图所示。



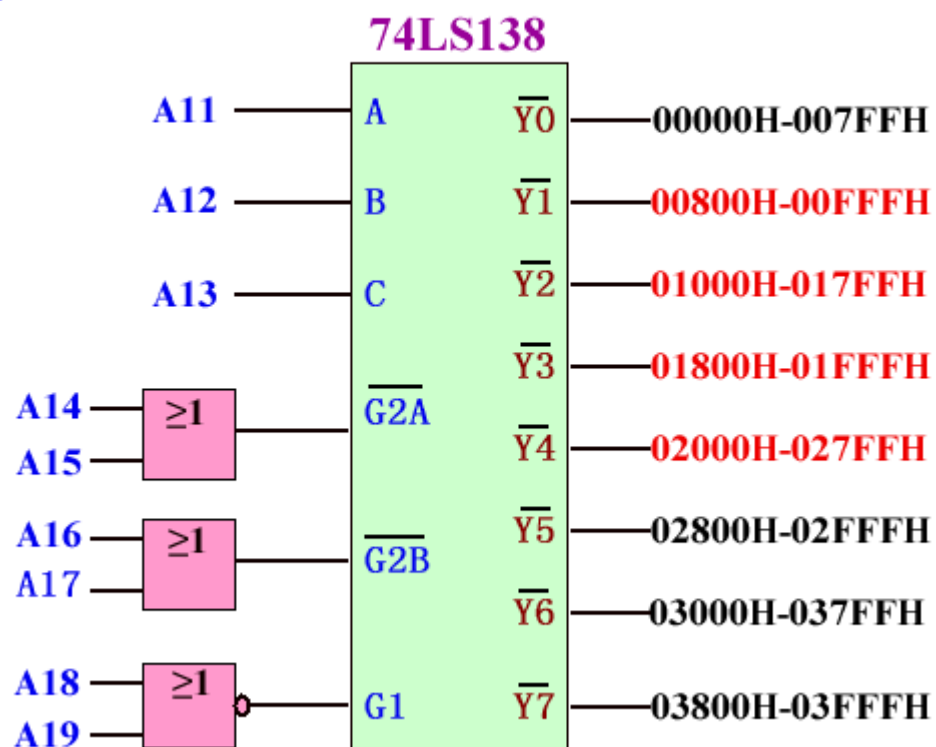
地址分配: 即将地址展开, 找规律

如上例：用2K*8bit的存储芯片构成8K*8bit的存储区，其地址从00800H--027FFH。每个芯片所占用的地址为：

2K	00800H	0000	0000	1000	0000	0000	B
	⋮	0000	0000	1000	0000	0001	B
2K	00FFFH	0000	0000	1111	1111	1111	B
	⋮	0000	0001	0000	0000	0000	B
2K	017FFH	0000	0001	0111	1111	1111	B
	⋮	0000	0001	1000	0000	0000	B
2K	01800H	0000	0001	1000	0000	0000	B
	⋮	0000	0001	1111	1111	1111	B
2K	01FFFH	0000	0001	1111	1111	1111	B
	⋮	0000	0010	0000	0000	0000	B
2K	02000H	0000	0010	0000	0000	0000	B
	⋮	0000	0010	0111	1111	1111	B
	027FFH	0000	0010	0111	1111	1111	B

片选地址
片内地址

确根据上例的地址分配设计译码电路

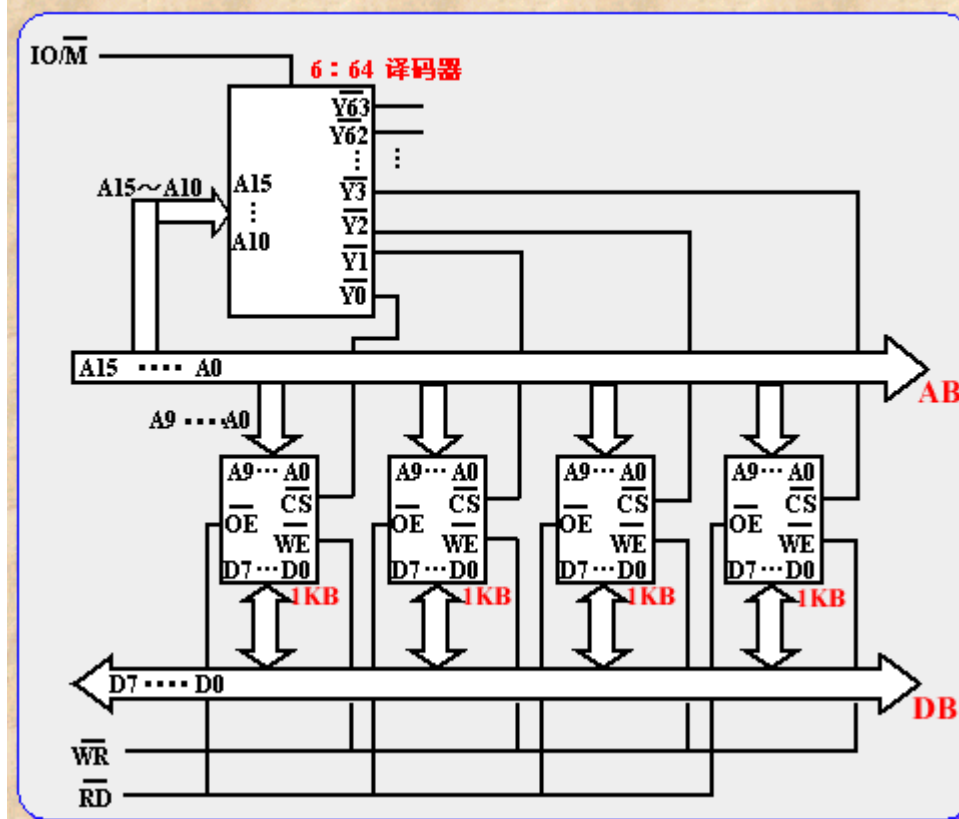


全译码方式就是对全部地址总线进行译码,使整个存储器地址空间能连续变化,每个存储单元只有唯一的地址。当有16根地址线时可直接寻址 64K 字节。有20根地址线时可直接寻址1M字节。

全译码方式的特点：

例：设一具有16根地址总线的8位微机系统中由多个1K*8bit 的存储器芯片组成存储区。这时可将地址总线的低10位(A₉-A₀)作为片内的寻址线,直接和存储器芯片的地址线相连。而把A₁₅-A₁₀经过6: 64译码电路译码后,作为各M芯片的片选信号,这就是全译码。如下页图所示。

全译码方式示例



四、I/O 接口

1. I/O 端口编址方式分为：统一编址和单独编址（I/O 指令方式）。

8086/8088 采用哪种编址方式及端口地址译码电路的设计。

2. CPU 与外设数据传输控制方式（包括：直接传输、查询传输、中断传输、DMA 传输）
中断过程、中断源、中断类型码、中断向量（中断服务程序入口地址）、中断向量表及相互关系。

3. 智能接口芯片及应用

a. 可编程定时/计数器 8253:

8253 拥有几个 16 位的计数器，其六种工作方式的特点，8253 的初始化编程（要知使用那个计数器，计数值 $N = T/TCLK = T \times fCLK$ ，工作方式）。注意：方式字写入控制口，计数值写入相应的计时器，计数值要和方式字中计数形式相对应。所用到的计数器都要一一做初始化。8253 是 8 位接口芯片，而计数器为十六位的，如何设置。

b. 可编程并行接口芯片 8255A:

拥有几个 8 位并口、三种工作方式及特点、初始化编程（方式字，C 口的置位/复位字）、8255A 的应用（8255 和按键，LED 的连接）。注意：方式字，C 口置位/复位字都写入控制口，通过特征位区别。

c. 可编程串行接口芯片 8251A:

同步通信和异步通信的特点和格式，波特率，RS-232-C 标准总线。8251A 的编程（方式字，命令字，状态字）。计算机接收，发送程序。

一、8251A 模式字

对 8251A 进行初始化时，模式字(异步模式)是按照模式寄存器的格式来设置的，模式寄存器的格式如图 5.8 所示。图中说明了 8251A 工作在同步模式和异步模式两种情况下，模式寄存器各数位的含义。当模式寄存器最低两位为 0 时，8251A 便工作在同步模式，此时，最高位决定了同步字符的数目；如果模式寄存器的两个最低位不全为 0，则 8251A 就进入异步模式。

在同步模式中，接收和发送的波特率(即移位寄存器的移位率)分别和 TxC 引脚、RxC 引脚上的输入时钟的频率相等。但是在异步模式中，要用模式寄存器中的两个最低位来确定波特率因子，此时，模式寄存器的第 2、3 位用来指出每个字符所对应的数据位的数目，第 4 位用来指出是否用校验位，第 5 位则用来指出校验类型是奇校验还是偶校验。在异步模式中，用两个最高有效位指出停止位的数目；但在同步模式中，第 6 位用来决定引脚 SYNDDET 是作为输入还是输出，第 7 位则用来指出同步字符的数目。如图 5.8。

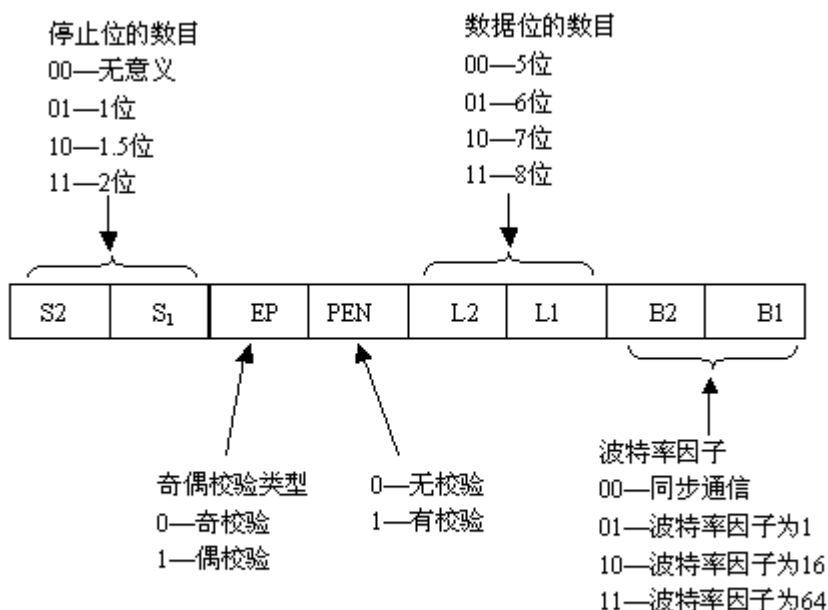


图 5.8 8251A 模式字格式（异步模式）

图 5.8 8251A 模式字格式（异步模式）

同步模式字不作要求。

二、8251A 控制字

对 8251A 进行初始化时，控制字是按照控制寄存器的格式写入的。控制寄存器的格式如图 5.9 所示。

控制寄存器的第 0 位为输出允许信号，只有将这一位设置为 1，才能使数据从 8251A 接口往外设传输。第 2 位为输入允许信号，在 CPU 从 8251A 接口接收数据前，先要使此位为 1。第 1 位是和引脚有直接联系的，引脚

通常和调制解调器的 CD 引脚相连，当 CPU 将控制寄存器的位设置为 1 时，便使引脚变为低电平，从而通知调制解调器，CPU 已经准备就绪。第 3 位为 1 使引脚 TxD 变为低电平，于是，输出一个空白字符。第 4 位置 1 将清除状态寄存器中所有的出错指示位。第 5 位用来设置发送请求，如果将引脚通过外部电路和 MODEM 的 CA 引脚相连，那么，第 5 位置 1 会使 RTS 引脚输出低电平，而使 CA 引脚得到一个高电平，从而使 MODEM 获得一个发送请求。第 6 位使 8251A 复位从而重新进入初始化流程。第 7 位只用在内同步模式，当为 1 时，8251A 便会对同步字符进行检索。如图 5.9。

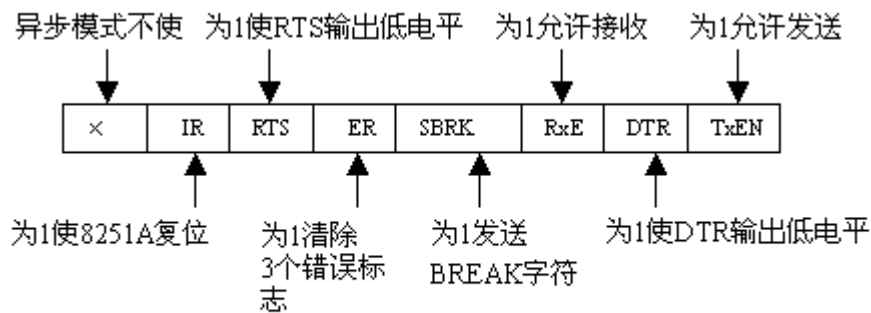


图 5.9 8251 控制字格式

图 5.9 8251 控制字格式

三、8251A 状态字

当需要检测 8251A 的工作状态时，经常要用到状态字。状态字是存放在状态寄存器中的。图 5.10 为 8251A 状态寄存器的格式。

状态寄存器的第 1、2、6 位分别与 8251A 引脚 RxRDY、TxEN、SYNDET 上的信号有关，第 0 位 TxRDY 为 1 用来指出当前数据输出缓冲器为空，这里要注意的一点是状态位 TxRDY 和引脚 TxRDY 上的信号不同，状态位 TxRDY 不受输入信号 CTS 和控制位 TxEN 的影响。而引脚 TxRDY 必须在数据输出缓冲器空、CTS 引脚为低电平且 TxEN 位为 1 时，才为 1，即 TxRDY 为 1 的条件为：数据缓冲区空且 CTS 为低电平且 TxEN 位为 1。

状态位 RxRDY 为 1 指出接口中已经接收到一个字符。通信程序中可以对 TxRDY 或 RxRDY 进行查询，来实现对 8251A 数据发送过程和接收过程的控制。

当 CPU 往 8251A 数据端口写入一个字符以后，状态位 TxRDY 会自动清 0，当 CPU 从 8251A 数据端口读取一个字符时，状态位 RxRDY 会自动清 0。

状态寄存器的第 2 位 TxEN 为 1 时，指出当前输出移位寄存器空，正等待输出缓冲寄存器送一个字符过来。

状态寄存器的第 3、4、5 位分别作为奇 / 偶校验错误指示、覆盖错（又叫做超越错）指示和帧格式错指示，当数据传输过程中产生其中某种类型的错误时，相应的出错指示位被置为 1。

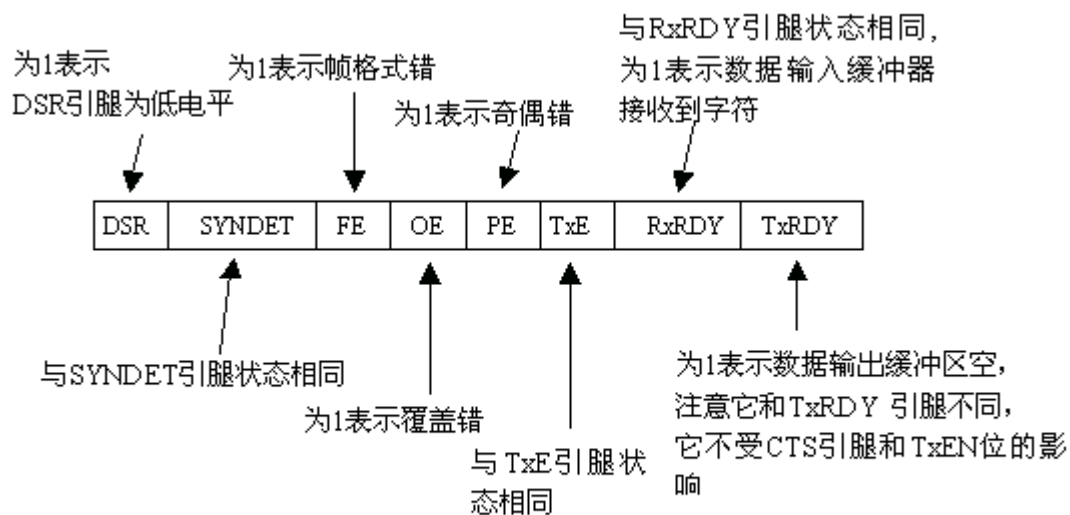


图 5.10 8251A 状态字

图 5.10 8251A 状态字

- d. **中断控制器 8259A**: 8259A 内部结构、工作方式、初始化编程 (ICW1~ ICW4, OCW1 ~ OCW3)。注意: 初始化写入顺序和地址。中断矢量装入矢量表的方法。

五、考试题型及评分标准

1. 考试题型: 选择、填空、问答、读程序、判断、M 设计、程序设计、接口设计
2. 评分标准: 卷面占 70%, 考勤 10%, 实验及作业占 20%