# 第一章

## 填空题 3’ =3分

1.计算机的五个基本部件：输入设备、输出设备、运算器、存储器、控制器

CPU包括运算器、控制器

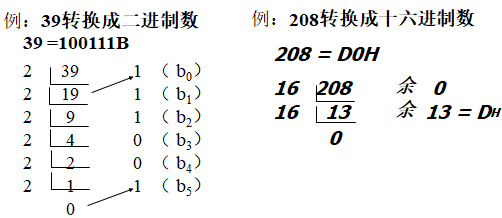
外部设备包括输入设备、输出设备

计算机主机包括运算器、控制器、存储器



2.数制转换

整数部分： B-D-H转换

P10-12

一个二进制位是一位：1bit，8个二进制位是一个字节：1Byte，2个字节，16位是一个字：1Word，双字32位：Dword

字长：CPU并行处理二进制的数据位数。8位机、16位机、32位机和64位机。

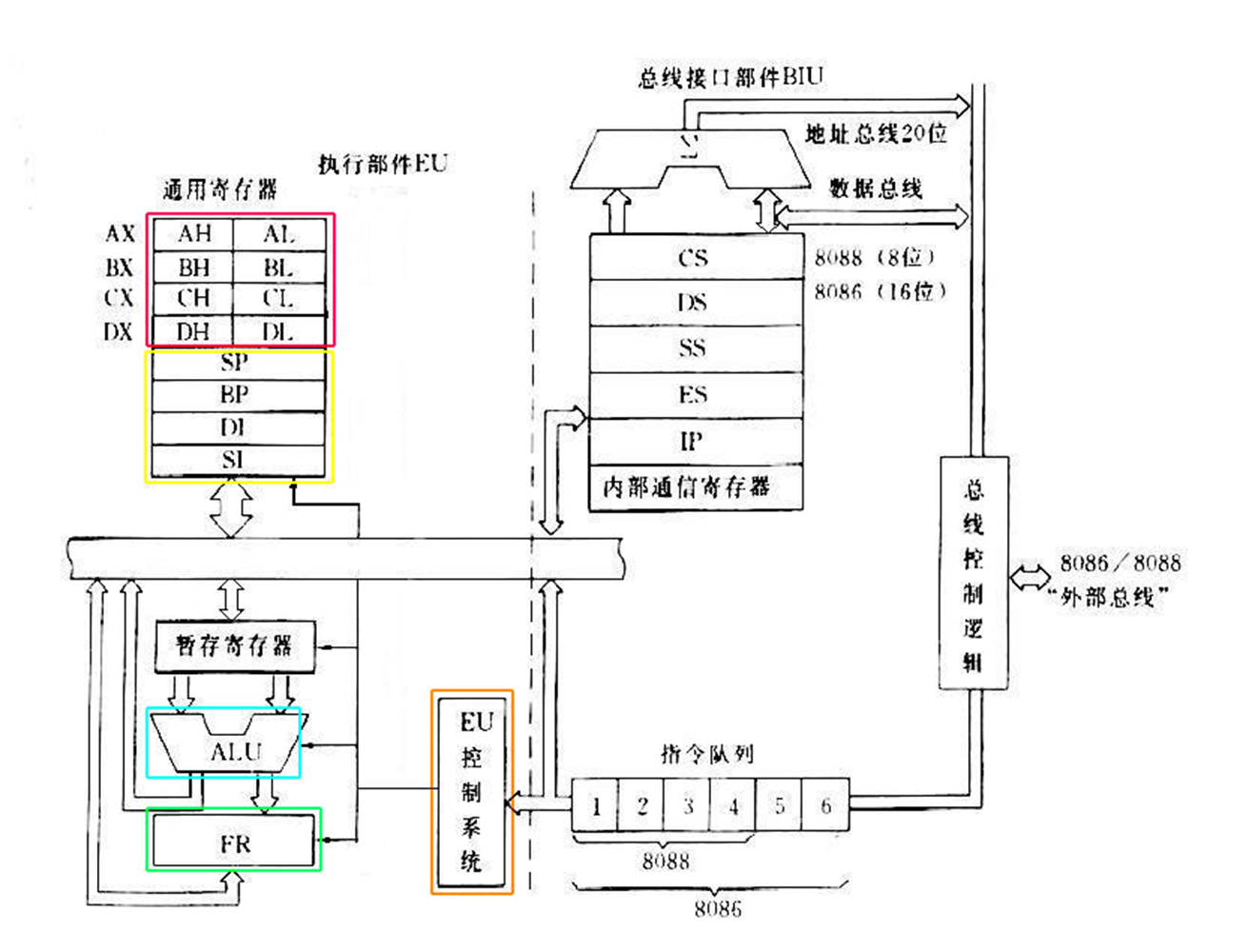
# 第二章 8086系统结构

## 填空4’+简答2’=14

1.8086CPU由指令执行部件（EU）和总线接口部件（BIU）组成

EU：负责执行指令，主要由算术逻辑运算单元（ALU）、标志寄存器（FR）、通用寄存器、EU控制电路组成。

BIU：主要由地址加法器、专用寄存器组、指令队列和总线控制电路等4个部件组成，其主要功能是形成访问存储器的物理地址、访问存储器并取指令暂存到指令队列中等待执行，访问存储器或I／O端口读取操作数参加EU运算或存放运算结果等。（取指令、独处操作数和写入结果）



EU每执行完一条指令，从指令队列队首取指

2.寄存器功能

指令执行部件（EU）设有8个16位**通用寄存器**：

包括通用数据寄存器：AX BX CX DX以及SI、DI、BP、SP主要用来存放存储器和I/O端口的地址

AX 累加器，一般用来存放参加运算的数据和结果，在乘、除法运算、I/O操作、BCD数运算中有不可替代的作用。

BX 基址寄存器，除可作数据寄存器外，还可放内存的逻辑偏移地址，而AX，CX，DX则不能。

CX 计数寄存器，是因为它既可作数据寄存器，又可在串指令和移位指令中作计数用。

DX 除可作通用数据寄存器外，还在乘、除法运算、带符号数的扩展指令中有特殊用途。

SI 源变址寄存器，多用于存放内存的逻辑偏移地址，隐含的逻辑段地址在DS寄存器中，也可放数据。

DI 目标变址寄存器

BP 基址指针，用于存放内存的逻辑偏移地址，隐含的逻辑段地址在SS寄存器中。

SP 堆栈指针

总线接口部件BIU设有4个16位**段寄存器**

CS（Code Segment），代码段寄存器，中存放程序代码段起始地址的高16位。

DS（Data Segment），数据段寄存器，中存放数据段起始地址的高16位。

SS（Stack Segment），堆栈段寄存器，中存放堆栈段起始地址的高16位。

ES（Extended Segment），扩展段寄存器，中存放扩展数据段起始地址的高16位。

**标志寄存器FR**

共有9个标志位，可分成两类：

状态标志 表示运算结果的特征，它们是

CF、PF、AF、ZF、SF和OF

控制标志 控制CPU的操作，它们是IF、

DF和TF。

CF(Carry Flag)：进位标志位

CF＝l，表示本次运算中最高位(D15或D7)有进位(加法运算时)或有借位(减法运算时)。

PF(Parity Flag)：奇偶校验标志位

PF＝1，表示本次运算结果中有偶数个“l”，PF＝0，表示本次运算结果中有奇数个“1”。

AF(Auxiliary Carry Flag)：辅助进位标志位。

AF＝l，表示运算结果的8位数据中，低4位向高4位有进位(加法运算时)或有借位(减法运算时)，这个标志位只在十进制运算中有用。

ZF(Zero Flag)：零标志位

ZF＝1，表示本次运算结果为零，否则即运算结果非零时，ZF＝0。

SF(Sign Flag)：符号标志

SF＝1，表示本次运算结果的最高位(第7位或第15位)为“l”，否则SF＝0。

OF(Overflow F1ag)：溢出标志

OF=1，表示本次算术运算结果产生溢出。

IF(Interrupt Flag)：中断标志位

DF(Direction Flag)：方向标志位

TF(Trap Flag)：单步标志位

**指令指针寄存器IP**

BIU要取指令的地址

3.8086CPU引脚（掌握最小模式的）

AD15～AD0 分时复用的地址数据总线

A19/S6～A16/S3 (Address/Status)：分时复用的地址/状态线

A19-A16和AD15-AD0 构成访问存储器的20位物理地址（1MB）

P21

4.奇偶地址的确定（简答可能性很大）

8086系统中，存储器是分体结构，1M字节的存储空间分成两个512K字节的存储体。

一个是偶数地址存储体，一个是奇数地址存储体，两个存储体采用字节交叉编址方式

地址A0

5.逻辑地址

格式为：段基地址:偏移地址 （均为无符号16位二进制数）

20位物理地址：16为段基址左移4位（后加4个0）加上偏移地址

6.CPU执行指令所涉及的周期：时钟周期、总线周期、指令周期

指令周期 :CPU从内存取出一条指令并执行这条指令的时间总和

时钟周期：一个时钟脉冲所需要的时间。为时钟脉冲的倒数。在计算机组成原理中又叫T周期或节拍脉冲。

总线周期：CPU通过总线对微处理器外部（存贮器或I/O接口）进行一次访问所需的时间。

一个总线周期一般包含4个时钟周期

# 第三章 8086指令系统

## 填空4’+简答1’=9

1.指令由操作码和操作数组成，

操作码：描述指令的操作功能

操作数：指出指令的操作对象

EA 有效地址

2.寻址方式

①立即寻址

操作数就在指令中，紧跟在操作码后面

MOV AX , 34EAH

②寄存器寻址

操作数在寄存器中，指令中源操作数和目的操作数都可用这种寻址方式。

MOV BP, SP

③直接寻址

当指令中的源操作数或目的操作数，采用直接给出被访问内存单元的逻辑地址（在[ ]内，方括号内给出的是被访问的内存单元的逻辑偏移地址，逻辑段地址隐含在寄存器DS数据段寄存器中）

MOV AX , [3E4CH] 物理地址（DS）X10H+3E4CH

④寄存器间接寻址

被访问的操作数在内存单元中，现将被访问的内存单元的逻辑偏移地址传送给寄存器，在指令中再由寄存器给出被访问的内存单元的逻辑偏移地址。

MOV SI , 61A8H

MOV DX , [SI] 指令中指定寄存器名用[ ]

⑤基址/变址寻址

在寄存器间接寻址给出的偏移地址上，加一个相对位移量，所以也称为相对寻址。

位移量是一带符号的16位16进制数。

当使用BX或BP寄存器时，称基址寻址；使用SI或DI寄存器时，称变址寻址。

⑥基址+变址寻址

EA是由三部分组成的，基址寄存器BX或BP的内容+变址寄存器的内容+位移量。

MOV AX , 8AH[BX][SI]

该例中EA＝8AH＋BX＋SI、 物理地址＝DS×10H＋8AH＋BX＋SI

⑦串寻址

规定源操作数的逻辑地址为DS:SI；目的操作数的逻辑地址为ES:DI。

MOV ES:DI, DS:SI

⑧I/O端口寻址

1. 当外部设备地址用8位寻址→直接端口寻址方式 端口地址28=256个，0-255
2. 当外部设备地址为超过8位寻址→间接寻址方式

DX寄存器作为间接寻址寄存器，端口地址2n个

3.指令格式：操作码 操作数

一．数据传送指令

①传送指令MOV

**MOV 目的操作数，源操作数**

目的操作数和源操作数可以采用不同的寻址方式，但操作类型要相同

目的操作数不能是立即操作数

源操作数和目的操作数不能同时为段寄存器（CS/DS/SS/ES）

②进栈指令PUSH

**PUSH 源操作数**

进栈操作是把数据传输到以SS为段基址、SP为偏移地址的栈中

只能对16位数据进行操作（即字不能是字节8位）

SP－2，指示堆栈中可以存放数据的位置

（SP-1，指示堆栈中可以存放数据的位置，存源操作数的高8位，SP-2存源操作数的低8位，完成进栈操作）

③出栈指令POP

**POP 目的操作数**

出栈操作是把以SS为段基址、SP为偏移地址的栈顶内容传输到目的操作数中

将SS:SP所指示的栈顶处的2字节的数据传输到目的操作数中，SP+2，指示当前栈顶位置，完成出栈操作

④XCHG 交换指令

**XCHG 目的操作数 ，源操作数**

段寄存器不能作为操作数

累加器专用传输指令

①IN 输入指令

IN AL , n

IN AX , n

IN AL , DX

IN AX , DX

②OUT 输出指令

OUT n , AL

OUT n , AX

OUT DX , AL

OUT DX , AX

③XLAT 换码指令

目标地址传送指令

1）LEA 有效地址传送到寄存器

2）LDS 装入一个新的物理地址

3）LES 装入一个新的物理地址

二．算数运算指令

①ADD 算术加法

**ADD 目的操作数，源操作数**

指令功能：目的操作数←目的操作数＋源操作数

算术指令影响CF标志位

②ADC 带进位算术加法

**ADC 目的操作数，源操作数**

指令功能：目的操作数←目的操作数＋源操作数＋CF

③INC 加1指令

**INC 目的操作数**

指令功能：目的操作数←目的操作数＋1

1)  SUB 算术减法

指令功能：目的地目的操作数－源操作数

2)  SBB 带进位算术减法

指令功能：目的地目的操作数－源操作数－CF

3) DEC 减1指令

指令功能：目的地目的操作数－1

4)MUL 无符号数乘法

5)DIV 无符号数除法

4.对某些位进行置1、清零、取反

AND 可以使指定位为0

OR 可以使指定位为1

XOR 可以使指定位为非（与“1”做异或取反，与“0”则保留不变）

P53

①NOT 逻辑求反

**NOT 目的操作数**

指令功能：将8位、16位寄存器或存储器内容求反

②AND 逻辑与操作

**AND 目的操作数, 源操作数**

将8位、16位寄存器或存储器内容和源操作数“与”

④OR 逻辑或操作

**OR 目的操作数, 源操作数**

指令功能：将8位、16位寄存器或存储器内容和源操作数“或”

4）XOR 逻辑异或操作

**XOR 目的操作数, 源操作数**

指令功能：将8位、16位寄存器或存储器内容和源操作数“异或”

逻辑左移SHL

算术左移SAL

逻辑右移SHR

算术右移SAR

不带进位循环左移 ROL

不带进位循环右移ROR

带进位循环左移RCL

带进位循环右移RCR

# 第四章 8086汇编语言程序设计

## 填空4’+简答1’+分析1’+编程1’=21

汇编语言由指令性语句和指示性语句组成

1. 汇编基本格式：

指令性语句（CPU执行的语句） [标号：] 操作码 [操作数1，] [操作数2]  [;注释]

标号相当于一个不确定的符号地址

指示性语句（汇编程序执行的语句） [标识符（名字）] 指示符（伪指令）表达式 [;注释]

1. 伪指令（伪操作）的基本功能

在指示性语句中做指示符，伪指令可以对数据进行定义，为变量分配存储区。但不能产生目标代码。

①等值语句

格式：符号名 EQU 表达式

一个符号名只能用EQU语句定义一次

②等号语句

格式： NUM＝34

允许对同一符号名多次赋不同的值

③变量定义语句

格式： 符号名 DB/DW/DD 表达式

用伪指令DB/DW/DD定义后就称为一个变量。

DB 字节型数据（8位） 12

DW 字型数据（16位） 1234H

DD 双字型数据（32位） 12345678H

复制操作符DUP

ALL\_ZERO DB 0，0，0，0，0用复制操作可改为：ALL\_ZERO **DB 5 DUP（0）**

①段定义语句格式：

段名 SEGMENT [定位类型] [组合类型] [‘类别’]

……

段名 ENDS

②段假设语句

ASSUME 段寄存器名：段名[,…]

③ORG伪指令格式： ORG <表达式>

过程定义语句

过程名 PROC NEAR/FAR

……

RET

过程名 ENDP

1. 常用的汇编语言程序框架

**DATA SEGMENT**  ；定义数据段

VAL1 DB 12H , 8EH ；定义变量

……

**DATA ENDS** ；数据段结束

**CODE SEGMEN**T ；定义代码段

**ASSUME DS:DATA , CS: CODE** ；段属性说明

**START: MOV AX , DATA** ；初始化DS

**MOV DS , AX**

…… ; 填写代码

MOV AX , 4C00H ；返回DOS

INT 21H ;中断指令

**CODE ENDS** ；代码段结束

**END START**  ；源程序结束

特点：程序分段：分为数据段、代码段、堆栈段……

1. 汇编语言程序设计基本步骤

①分析问题，确定数学模型，确定算法

②绘制流程图或写出程序步骤

③分配存储空间及工作单元

④编写程序

⑤检查和调试

5.汇编语言程序的基本结构

顺序结构、分支结构、循环结构和子程序（主程序有调用指令call，子程序有返回指令ret）

# 第五章 存储器原理与接口

## 填空4’+简答1’+分析1’=15

1. 半导体存储器按存取方式分类

随机存储器RAM (Random Access Memory)

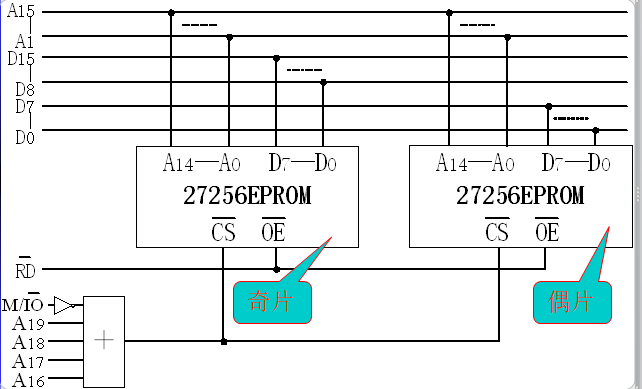
只读存储器ROM（Read-Only Memory）

串行访问存储器SAS(Serial Access Storage)

8086CPU对存储器的读方式：CPU总是16位的读；从偶地址读。

2.存储器的扩展及如何连接奇片、偶片

①ROM拓展电路



8086CPU EPROM

DB7…0 D7…0 (偶片接数据总线低8位)

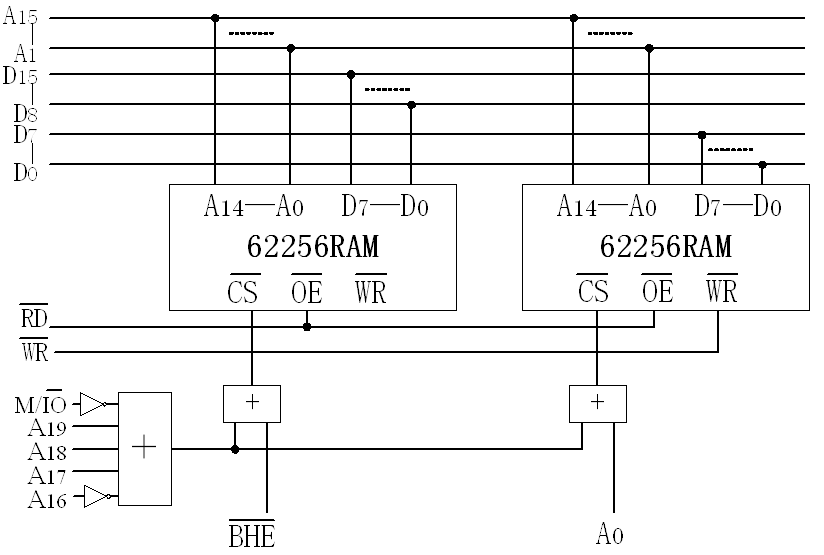
DB15…8 D7…0 (奇片接数据总线高8位)

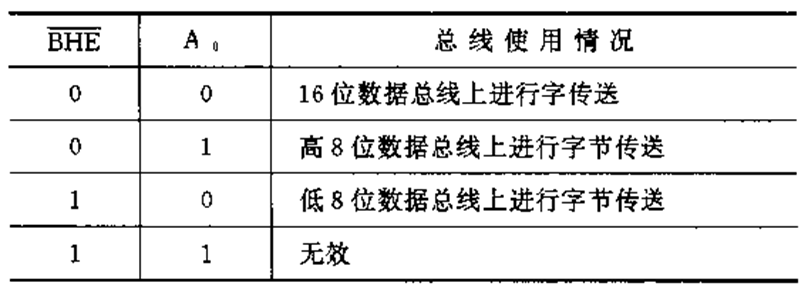
/RD /OE

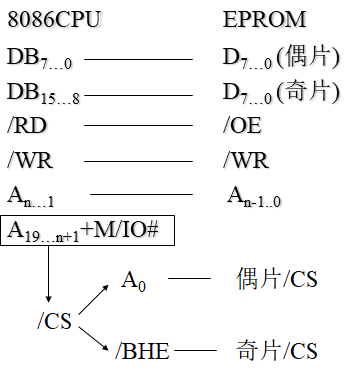
An…1 An-1..0

A19…n+1+M/IO# /CS

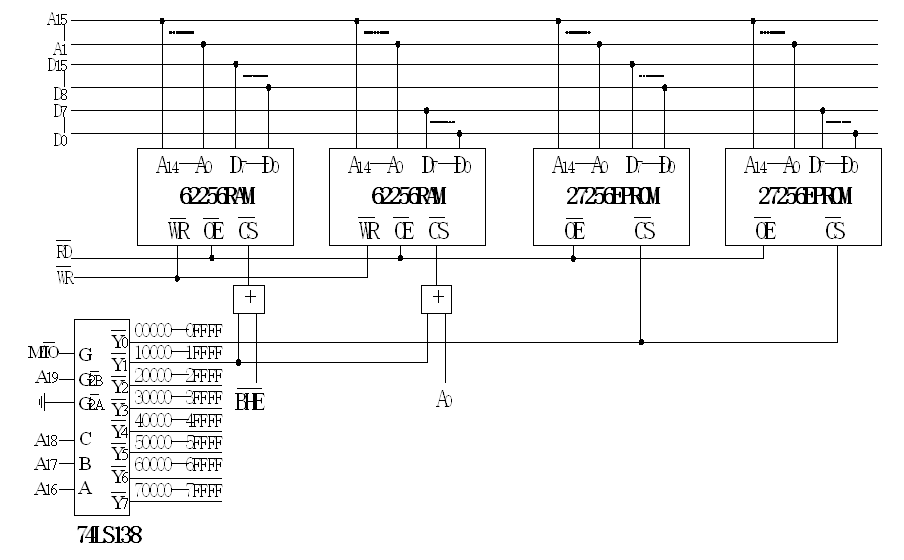
②RAM拓展电路







3. 3-8线译码器74LS138



# 第六章 微型计算机的输入/输出

## 填空4’+简答1’+分析1’=15

1.什么是I/O端口

接口电路通常包含一组能够与处理器交换信息的寄存器或缓冲器，称为I/O端口

（CPU与外设通信必须借助于必要的电路来实现，这样的电路成为I/O接口）

包括数据端口、状态端口、控制端口

2.为什么需要I/O端口

①速度不匹配：CPU的运行速率比外部设备的处理速度要快得多，通常简单第用一条输入/输出指令无法完成CPU与外设之间的信息交换。

②时序不匹配：外部设配有自己的时序电路，与CPU无法取得统一

③信息格式不匹配：CPU只能处理并行的信息，而有些外设的信息是串行的

④信息类型不匹配：CPU只能识别数字量，而有些外设的信息是模拟量

因此需要I/O端口作为CPU与外部通信的桥梁

微机的外部设备多种多样

工作原理、驱动方式、信息格式、以及工作速度方面彼此差别很大

它们不能与CPU直接相连

必须经过中间电路再与系统相连

这部分电路被称为I/O接口电路

接口的功能

①进行地址译码或设备选择，以便使CPU能与某一指定的外部设备通讯；

②状态信息的应答，以协调数据传送之前的准备工作；

③进行中断管理，提供中断信号；

④进行数据格式转换，如正负逻辑的转换，串行与并行数据转换等；

⑤进行电平转换，如TTL电平与MOS电平间的转换；

⑥协调速度；

⑦时序控制，提供实时时钟信号。

I/O端口寻址方式

①存储器映像的I/O寻址

存储单元和I/O端口的地址统一编址（对I/O端口的访问和对存储器的访问相同）

优点：可以用访向存储器的指令来访问I/O端口，

缺点：I/O端口占用了一部分存储器地址空间，因而使用户的存储地址空间相对减小。另外不利于程序阅读。

②I/O映像的I/O寻址（8086采用这种寻址方式）

I/O端口地址与存储单元地址分开编址，使用不同的指令

优点：I/O端口不占用存储器地址，故不会减少用户的存储器地址空间；程序便于阅读。

缺点：指令少，编程灵活性相对减少（只能对端口数据进行输入/输出操作，不能直接进行移位、比较等其他操作）；硬件上需要I/O端口的译码芯片，增加了硬件开支。

3.输入/输出方式

包括程序控制方式、直接存储器存取方式

程序控制方式包括同步传输方式（无条件传输方式）、异步查询方式（条件传输方式）、中断方式

8086CPU的I／O指令

直接寻址输入／输出指令(八位端口地址)

IN AL，n (字节输入)

IN AX，n (字输入)

OUT n，AL (字节输出)

OUT n，AX (字输出)

DX寄存器间接寻址输入（十六位端口地址）

IN AL，DX (字节输入)

IN AX，DX (字输入)

OUT DX，AL (字节输出)

OUT DX，AX (字输出)

# 第七章 可编程接口芯片

## 填空4’+简答1’+编程1’=15

目前常用的可编程芯片有如下几种：

8255A 并行I/O接口

8253 计数器/定时器（减计数器，8051是加计数器）

8251 串行I/O

8259A 中断控制器

1.8255A

8255A具有**3**个带锁存或缓冲的数据端口，可与外设并行进行数据交换。可为CPU与外设之间提供并行输入／输出通道。

8255A的引脚

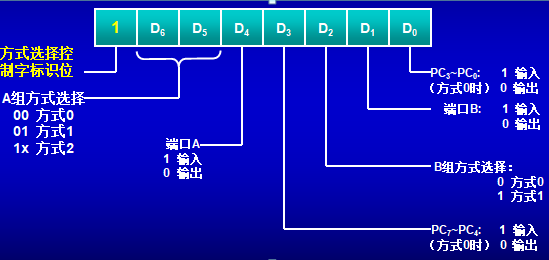
①与外设连接的引脚：有三个数据端口，每个端口是8位的。

PA0…PA7、PB0…PB7、PC0…PC7共有24位引脚

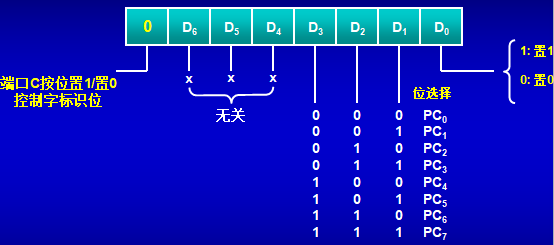
②与CPU连接的引脚：数据线D7—D0，读写控制线/WR、/RD，复位线RESET，以及和CPU地址线相连接的片选信号/CS、端口地址控制线A0和A1  
③电源线和地线：VCC和GND

8255A的工作方式  
①方式0——基本输入／输出方式；主要工作在无条件的输入/输出方式下（A/B/C）  
②方式1——选通输入／输出方式；主要工作在异步或条件传输方式（A/B）  
⑨方式2—一双向传送方式。是指在同一端口内分时进行输入/输出的操作（A）

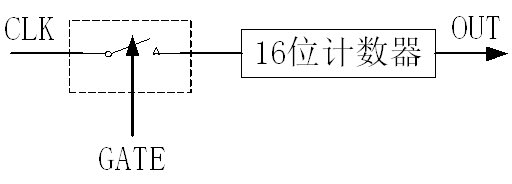
方式控制字



C口置“1”／清“0”控制字



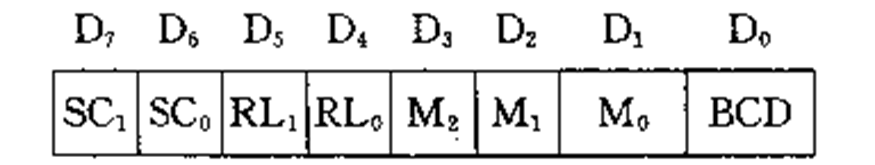
2. 8253内部有3个独立的16位定时／计数器通道。计数器可按照二进制或十进制计数，计数和定时范围可在1—65535之间改变，每个通道有6种工作方式，计数频率可高达2MHz以上



CLK：计数器脉冲输入

GATE：门控信号

OUT：计数器输出

P150

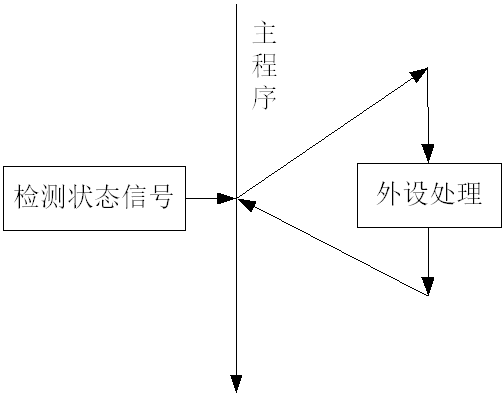
# 第九章 中断和中断管理

## 填空3’+简答1’=8

为什么需要中断？

当CPU用查询的方式与外设交换信息时，CPU就要浪费很多时间去等待外设

1.中断



当外部设备准备好向CPU传送数据，或者外设已准备就绪接收CPU的数据，或者有某些紧急情况要求处理，或者是定时时间到等，外设（通过改变状态线上的信号状态）向CPU发出中断请求，CPU接收到请求并在一定条件下，暂时停止执行原来的程序而转去中断处理，处理好中断服务再返回来执行原来程序

中断请求触发器

中断屏蔽触发器

中断允许触发器IF，IF=1，允许CPU响应中断，开中断（STI）；IF=0，不允许CPU响应中断，关中断（CLI）。

中断条件：

①中断屏蔽触发器处于非屏蔽状态。

②中断允许触发器是否处于开中断状态。只有CPU是开中断的条件下，CPU才能进入中断响应过程

2.中断过程

①外设发中断请求信号给CPU即中断请求

（当中断屏蔽触发器状态为1，则中断请求触发器输出的中断请求信号发给CPU）

②CPU对中断请求信号所作出的反应即中断响应

（CPU响应可屏蔽中断申请必须满足的3个条件:无总线请求；CPU被允许中断； CPU执行完现行指令．）

③CPU执行对外设操作的子程序即中断处理。

（关闭中断；CS、IP以及FR的内容推入堆栈；中断服务程序段地址送CS中,偏移地址送IP中。）

中断优先权：0号优先级最高，中断号越小，优先级越高