

## 微机原理与接口

### 1. 微型计算机系统

1. 微型计算机：以微处理器为核心，配上大规模集成电路的存储器、输入输出接口电路和系统总线组成的计算机
2. 总线
  - a) 地址总线：单向，CPU 发出，决定可直接寻址的内存容量
  - b) 数据总线：三态双向
  - c) 控制总线：单向，命令信号线、状态信号线

### 2. 8086 微处理器

1. 8086/8088：对内数据通路均为 16 位
  - a) 8086：16 位数据线，20 位地址线
  - b) 8088：8 位数据线(对外)
2. CPU 内部结构：两级流水线
  - a) 总线接口部件 BIU：控制 CPU 与存储器和 I/O 接口(端口)读写
  - b) 执行部件 EU：从指令队列取出指令并执行
3. CPU 内部寄存器
  - a) 段寄存器
    - i. CS：代码段
    - ii. SS：堆栈段
    - iii. DS：数据段
    - iv. ES：扩展段
  - b) 数据寄存器
    - i. AX：累加器
    - ii. BX：基址寄存器，常用存放访存有效地址
    - iii. CX：计数器
    - iv. DX：数据寄存器，常用存放 I/O 端口地址
  - c) 指针寄存器
    - i. BP：基址指针（栈底）
    - ii. SP：堆栈指针(SS:SP)
  - d) 变址寄存器：SI、DI
  - e) 指令指针寄存器 IP(CS:IP)：下一指令对代码段基址的偏移量
  - f) 标志寄存器(9 位有效)
    - i. 状态标志(6 位)：反映刚完成的操作结果的特点
      - CF 进位标志，CF=1：运算最高位向更高位有进位或借位
      - AF 辅助进位标志，AF=1：D3 位向 D4 位有借位
      - PF 奇偶校验位，PF=1：低 8 位有偶数个 1
      - ZF 全零标志，ZF=1：运算结果全为 0
      - SF 符号标志：与结果最高位相同
      - OF=1：运算有溢出
    - ii. 控制标志(3 位)：在某些指令操作中起到控制 CPU 行为的作用

IF 中断允许标志, IF=1: 允许响应可屏蔽中断

DF 方向标志: 0+, 1-

TF 跟踪标志, TF=1: 单步执行指令

4. 20 位地址加法器: 逻辑地址→物理地址(段基址\*16+偏移地址)

5. 指令队列

a) 8086: 6 字节

b) 8088: 4 字节

c) 每当指令队列有两字节或以上空闲时, BIU 根据 CS 和 IP 的值, 形成物理地址  
取出指令存放在指令队列中

d) EU 从指令队列取指并执行

e) 执行转移、调用、返回指令时, BIU 自动清除指令队列

6. 逻辑地址(段基址:偏移地址)

a) 逻辑段: 最小  $2^4=16B$ , 最大  $2^{16}=64KB$

b) 段与段之间可以分开、连续、重合、重叠

c) 每个存储单元可以定义在一个或多个段中

7. 8086 引脚

a)  $AD_{15}-AD_0$

b)  $A_{19}/S_6-A_{16}/S_3$

i. CPU 访问 I/O 端口时,  $A_{19}-A_{16}$  保持为 0

ii.  $S_4-S_3$  确定当前使用的段寄存器,  $S_5$  中断标志,  $S_6$  恒为 0

c)  $\overline{RD}$ : 读信号(CPU 读)

d)  $\overline{WR}$

e)  $M/\overline{IO}$ : 存储器或端口访问信号

f)  $\overline{BHE}/S_7$ : 总线高字节有效

$\overline{BHE}$	$\overline{AD}_0$	
0	0	字传送
0	1	高字节
1	0	低字节

(存储分体结构)读奇地址(非规格字)开始的字, 需要两个周期

g)  $\overline{READY}$ : 无效时插入等待周期, 直到有效

h)  $\overline{TEST}$ : 测试信号, 执行 WAIT 指令时, 每 5 个周期测试一次

i) INTR: 可屏蔽中断请求信号

j) NMI: 不可屏蔽中断请求信号

k)  $\overline{INTA}$ : 中断响应信号

l) RESET: 复位信号, 至少保持 4 个周期, 将除 CS 以外的寄存器清零, 指令队列  
清空, CS 置为 0FFFFH

m) HOLD: 总线请求信号

n) HLDA: 总线响应信号

o)  $\overline{ALE}$ : 地址锁存允许信号(最小模式中可做地址锁存器选通信号)

高电平时, CPU 地址线上地址有效

低电平时, 锁存地址

p)  $\overline{DEN}$ : 数据允许信号(数据收发器锁存信号)

低电平时, 数据线上有有效数据

- q)  $\overline{DT}/\overline{R}$ : (CPU)数据发送/接收信号, 控制数据传送方向
- r)  $\overline{MN}/\overline{MX}$ : 最小模式接+5V, 最大模式接地

#### 8. CPU 时序

- a) 时钟周期: 最小时间单位, T 状态
- b) 总线周期: BIU 完成一次总线操作需要的时间, 最少 4 个 T 状态 (基本总线周期)
- c) 机器周期: 若干 T 状态, 完成一个基本操作
- d) 指令周期: 若干机器周期, 取出并执行一条指令所需时间
- e) 读写总线周期

### 3. 8086 常用指令

#### 1. 寻址方式

- a) 立即寻址
- b) 寄存器寻址
- c) 存储器操作数寻址方式(EA 有效地址、偏移)

直接寻址	指令给出 EA(隐含段寄存器 DS)
寄存器间接寻址	EA 在 SI、DI、BX 中(DS)
	EA 在 BP 中(SS)
寄存器相对寻址	EA=SI/DI/BX/BP+位移量
基址变址寻址	EA=BX/BP(基址)+SI/DI
相对基址变址寻址	EA= BX/BP+SI/DI+位移量

- d) 固定寻址
- e) I/O 端口口操作数寻址
  - i. 直接寻址(0-255)
  - ii. 间接寻址(256+) DX 给出

#### 2. 传送类指令

- a) 通用数据传送指令  
不影响标志位
  - i. *MOV* DST, SRC  
目的操作数 DST 不能是立即数、CS  
立即数不能直送段寄存器  
DST, SRC 不能同为段寄存器, 不能同为存储器操作数  
SRC 为立即数若最高位为字母, 加前导零 *MOV AL, 0BH*  
操作数类型无法确定, 需显示指明 *MOV BYTE PTR [DI], 43H*
  - ii. *PUSH/POP* AX  
以字(2 字节)为单位传送  
低地址字节送低字节
  - iii. 换码指令 *XLAT*  
访问表  $AL \leftarrow DS: [BX + AL]$
- b) 地址传送指令
  - i. 有效地址送寄存器 *LEA*
- c) 标志寄存器传送指令

- i. LAHF、LASF
  - ii. PUSHF、POPF
- 3. 算术指令
  - a) 加法指令
    - i. 不带进位 ADD DST, SRC ;DST = DST+SRC
    - ii. 带进位 *ADC* DST, SRC ;DST = DST+SRC+CF
    - iii. INC DST ;DST++ *INC*、*DEC* 不影响 CF
  - b) 减法指令
    - i. SUB
    - ii. SBB (DST = DST-SRC-CF)
    - iii. DEC
    - iv. 求补 NEG DST
    - v. *CMP* DST, SRC  
计算 DST-SRC, 影响标志位, 不影响 DST
  - c) 十进制调整指令 影响标志位(OF 除外)
    - i. 压缩 BCD 码加法 DAA(无操作数, 隐含操作数寄存器 AL)  
压缩 BCD 码减法 DAS
- 4. 逻辑运算指令
  - a) 逻辑运算指令
    - i. AND,OR,XOR 影响 SF,PF,ZF,使 CF,OF 置零
    - ii. NOT 不影响标志位
    - iii. *TEST* OPR1, OPR2  
计算 OPR1^OPR2, 改变标志位, 不影响 OPR1
  - b) 循环移位指令
    - i. 循环移位指令 不影响 SF,ZF,PF  
循环左移 *ROL* OPR, CNT ;被移出的位填充 CF 和空位(符号位)  
循环右移 *ROR*  
带进位循环 RCL,RCR 被移出的位填充 CF,CF 填充空位
    - ii. 移位指令 影响 SF,ZF,PF  
逻辑左移 SHL, 右移 SHR 被移出的位填充 CF,0 填充空位  
算数左移 SAL(同 SHL),SAR(空位由本身填充)
- 5. 控制转移指令
  - a) JMP
  - b) 条件转移(根据 OF,SF,ZF,PF,CF)JO,JS,JZ,JP,JC
  - c) LOOP,LOOPZ/LOOPE,LOOPNZ/LOOPNE
  - d) CALL, RET
  - e) INT n
- 6. 处理机控制指令
  - a) 标志处理指令: CLC(CF=0)、STC(CF=1)、CLD、STD
  - b) NOP

## 4. I/O 端口地址译码技术

1. 端口是接口电路中能被 CPU 直接访问的寄存器

CPU 与外设之间的交换信息通过 I/O 端口进行

一个接口可以有多个端口

I/O 操作：对 I/O 端口的操作

## 2. 编址方式

a) 统一编址：存储器映射方式

i. 特点

主存单元与 I/O 端口一起编址

根据地址区分访问的是 I/O 端口还是主存

系统中可以不设 I/O 指令

ii. 优缺点

指令兼容，有较大编址空间，控制逻辑简单

指令比专门 I/O 指令长，执行速度慢，端口占用存储器的地址空间

b) 独立编址：I/O 映射方式（8086）

i. 特点

主存单元与 I/O 端口分开寻址

设置专门的 I/O 指令

指明是访问 I/O 端口，还是主存单元

ii. 优缺点

指令地址较短，所需译码硬件较少

指令格式较短，执行时间短

独立的控制结构，与存储器分开设计

3. 端口访问：对 I/O 端口的读写。I/O 端口与累加器(AL,AX)之间的数据传送

a) 单字节地址（00H-FFH）：直接寻址或 DX 间接寻址

IN AL,34H

OUT 34H,AL

b) 双字节地址：DX 间接给出端口地址

c) 允许用户使用地址：300H-31FH（地址开关避免冲突）

d) 在程序中执行 IN/OUT 指令，才能激活  $\overline{IOR}/\overline{IOW}$ 、

4. 端口地址译码电路：实现地址与端口的对应，产生片选信号

a) 高位地址线与 CPU 控制信号组合产生 I/O 接口芯片的片选信号

b) 低位地址线直接连接到 I/O 接口芯片

## 5. 端口地址译码电路结构形式

a) 固定式端口地址译码

b) 可选式端口地址译码（比较器/异或门+ 地址开关）

## 6. 74LS138 译码器

a)  $G_1 \overline{G_{2A}} \overline{G_{2B}}$

b) CBA

c)  $\overline{Y_0}-\overline{Y_7}$

## 5. 定时/计数技术（可编程定时/计数器 8254A）

1. 组成

a) 与 CPU 的接口

i. D7-D0

- ii. 寻址控制线: A0、A1、 $\overline{CS}$
- iii. 读写控制线  $\overline{WR}$ 、 $\overline{RD}$
- b) 与外设的接口 (三个独立计数器/定时器)
  - CLK 输入时钟、OUT 波形输出
  - GATE 门控信号

c) 计数器内部结构

- i. 16 位计数初值寄存器
  - ii. 16 位减一计数器
    - 先减一再判断, 计数最大值为 10000H 对应写入初值为 0
  - iii. 16 位当前计数值锁存器
2. 编程命令

- a) 初始化编程: 先写控制字, 再写计数初值

b) 控制字

- i. 设置工作方式

D7	D6	D5	D4	D3	D2	D1	D0
计数器选择		写高位	写低位	工作方式			计数方式
00/01/10 : 0/1/2 11:读回(8254)		00:锁存 01/10:低/高 11:先低后高					0:二进制 1:十进制

- ii. 读回命令

读计数值: 先锁存, 再读回

设置读回命令, 再读回 (8254)

c) 状态字

3. 工作方式

a) 启动

- i. 软件启动 0、2、3、4

GATE=1 时, 一旦计数初值写入减法计数器就开始计数, 由 IOW 信号驱动

- ii. 硬件启动 1、5

计数初值写入减法计数器并不立即开始计数, GATE 上升沿出现才计数

b) 停止

- i. 强制停止 (自动重装初值)
- ii. 自动停止 (单次计数或定时)

c) 共同点

- i. 控制字写入后所有控制逻辑复位, OUT 进入初始态
- ii. 写入初值后, 过一个周期计数执行部件 CE 才开始工作
- iii. CLK 上升沿采样 GATE

	方式 0	方式 1
	计数结束(减到 0) 产生正跳变信号后	可重触发的单稳触发器
GATE	GATE=0 中止计数 GATE=1 继续计数	GATE 上升沿重装初值
计数期间, 写入新初值	按新初值计数	本次计数完, GATE 出现上升沿, 才按新初值计数

	方式 2	方式 3
	分频器	方波发生器
	减到 1, 输出一个 CLK 脉宽的低电平	计数值完成一半(奇数, $(n+1)/2$ ) 时, 使 OUT 变低
GATE	上升沿重新启动	
工作期间, 写入新初值	完成本次计数后, 按新初值计数	不影响现行计数

	方式 4	方式 5
	软触发选通发生器	硬触发选通发生器
	减到 0, 输出一个脉宽低电平	
GATE	上升沿从初值开始计数	
计数期间, 写入新初值	新初值立即起作用	

## 6. 中断技术

1. 中断: 程序的转移
  - a) 中断类型码: 8086 分为 256 级中断
  - b) 中断向量: 中断服务程序入口地址(CS:IP)
  - c) 中断向量表: 所有中断向量集中放到内存地址 0000H-03FFH  
高地址: 段基址  
低地址: 偏移地址
  - d) 中断向量指针: 中断类型号\*4
  - e) 一个中断号所对应的中断服务不是唯一的: 中断向量可以改变
  - f) 中断识别: 形成入口地址
    - i. 软件查询识别
    - ii. 中断向量识别 (8086)
  - g) 中断优先级确定
    - i. 软件查询
    - ii. 硬件优先级排队
2. 中断系统
  - a) 触发器: 中断请求触发器、中断屏蔽触发器, 中断允许触发器
  - b) 功能: 中断响应、断电保护和中断处理、终端优先权排队、中断嵌套
  - c) CPU 响应可屏蔽中断的条件
    - i. 有中断源发出的中断请求信号
    - ii. 允许中的 (IF=1)
  - d) 8086 中断系统
    - i. 中断向量机制
    - ii. 专用中断寄存器 8259A 实现优先权管理
  - e) 中断分类
    - i. 软件中断: 确定或随机, 中断类型码自动形成, 不执行中断响应周期  
被动: 溢出中断、除零中断  
主动: INT

- ii. 硬件（外部）中断：随机，中断类型码由中断控制器提供，执行中断响应周期
  - 非屏蔽中断 NMI
  - 可屏蔽中断 INTA：受 IF 控制
- iii. 优先级：软件中断（除法错、指令中断、溢出中断）>非屏蔽中断>可屏蔽中断>单步中断
- f) 中断处理过程
  - i. 中断申请：外设向 CPU 发送中断请求，如果中断开放，且没有其他设备申请 DMA 传送，则 CPU 在当前指令结束时响应中断
  - ii. 中断响应：进行断点及标志保存，PSW、CS、IP 依次入栈，根据中断类型号找到中断服务程序入口地址
    - 软中断、NMI：自动形成中断类型码
    - INTR：通过总线从外设获取，必须经过总线周期
    - 第一个中断响应周期，CPU 从出中断响应吸纳后数据线浮空
    - 第二个中断响应周期，被相应的外设送出中断类型码，CPU 从数据线得到中断类型码
  - iii. 中断服务程序：保护现场、开中断、恢复现场
  - iv. 中断返回：IP、CS、PSW 出栈，恢复程序断点
- g) CPU 中断处理：
  - 关中断
  - 断点保护
  - 形成中断入口地址
  - （开中断）
  - 中断服务程序
  - （关中断）
  - 恢复现场
  - 中断返回
  - 开中断
- h) 修改中断向量
  - i. 直接装入
  - ii. DOS 系统功能调用
- 3. 8259A：可编程中断控制器
  - a) 单片：8，级联：64
  - b) 功能
    - i. 中断优先权管理
    - ii. 接收中断请求
    - iii. 提供中断类型号
    - iv. 屏蔽与开放中断
  - c) 引脚
    - i. CAS<sub>0</sub>-CAS<sub>7</sub>：级联总线，主控输出，从控输入
    - ii. IR<sub>0</sub>-IR<sub>7</sub>
    - iii. INT：中断申请线
    - iv. SP/EN：



- v. INTA: 中断响应引脚
- d) 寄存器
  - i. 中断请求寄存器 IRR: 保留 8 根中断请求线的中断请求状态
  - ii. 中断服务寄存器 ISR: 保存正在被 8259A 服务着的中断状态
  - iii. 中断屏蔽寄存器 IMR: 对应位为 1 表示此中断已加屏蔽, 为 0 表示此中断未加屏蔽, 可进入 PR 作裁决。
  - iv. 优先权分析器 PR: 对新进入的中断请求和 ISR 中当前正在处理的中断比较
- 4. 工作方式
  - a) 中断屏蔽方式
    - i. 普通屏蔽方式
    - ii. 特殊屏蔽方式 允许优先级低的进入
  - b) 优先级管理方式
    - i. 一般完全嵌套方式 IR0-IR7, IR0 优先级最高
    - ii. 特殊完全嵌套方式 可响应同级中断
    - iii. 优先级自动循环方式 刚处理完的中断优先级最低
    - iv. 优先级特殊循环方式 指定最低或最高优先级
  - c) 结束中断方式 EOI
    - i. 自动结束 (单片, 无嵌套) 进入中断处理后清除 ISR 对应位
    - ii. 一般中断结束方式 CPU 向偶地址发命令, 消除 ISR 最高位
    - iii. 特殊中断结束方式 清楚指定位
- 5. 8259A 编程

a) 端口地址

	写	读
奇地址	OCW1、ICW2、ICW3、ICW4	IMR
偶地址	ICW1、OCW2、OCW3	IRR、ISR

- b) 初始化命令字
  - i. ICW1: 芯片控制初始化命令字
  - ii. ICW2: 中断类型码设置 已知接  $IR_x$  和 OCW2 求中断类型码
  - iii. ICW3: 级联设置
  - iv. ICW4: 方式控制初始化命令字
- c) 操作命令字:
  - i. OCW1: 屏蔽命令字
  - ii. OCW2: 循环优先方式、中断结束方式
  - iii. OCW3:

## 7. DMA 技术

### 1. DMA 传输过程

- a) 申请阶段
 

外设有 DMA 需求时, 向 DMAC 发送 DMA 请求 DREQ

DMAC 收到 DMA 请求的时候, 向 CPU 的 HOLD 引脚发送总线请求信号 HRQ
- b) 响应阶段
 

CPU 接到总线请求后, 若允许 DMA 传输, 则在当前总线周期结束后释放总线控制权, 并向 DMAC 发出 HLDA 信号, 通知 DMAC 已释放总线控制权

DMAC 获得总线控制权，向外设发送应答信号 DACK，通知外设可以进行 DMA 传输

c) 数据传送阶段

DMAC 送出数据和控制信号，进行数据传输

d) 传送结束阶段

传输完后，DMAC 产生信号给外设，外设撤销 DREQ 信号，使 HRQ 无效  
CPU 收到无效信号后，使 HLDA 无效，收回对总线的控制权

## 2. DMA 传送方式

a) 单字节传送方式

- i. 传送完一个字节后，DMAC 释放总线控制权
- ii. 效率低，总线不会长时间被 DMAC 占用

b) 数据块传输方式

- i. 连续传输
- ii. 效率高，CPU 长时间不能控制总线

c) 请求传输方式

- i. 每传输一个字节，检查外设 DREQ 信号

## 3. DMA 操作类型

a) DMA 读（读存储器）

b) DMA 写

c) 存储器到存储器

d) DMA 校验

## 4. 8237A（DMA 控制器、4 通道）引脚

a) 请求与应答

- i. DREQ0-3: DMA 通道请求信号，（DREQ0 优先级最高）
- ii. DACK: DMA 通道应答信号
- iii. HRQ: 使用总线请求信号
- iv. HLDA: 总线应答信号

b) 被动状态（CPU 控制 DMA）

- i. A0-A3
- ii. DB0-DB7
- iii. CLK、READY
- iv.  $\overline{\text{CS}}$ 、 $\overline{\text{IOR}}$ 、 $\overline{\text{IOW}}$

c) 主动状态

- i. A0-A7
- ii. DB0-DB7: 高 8 位地址/数据线，分时复用
- iii. ADSTB: 地址选通，有效时锁存高 8 位地址
- iv.  $\overline{\text{MEMR}}$ （读存储器）、 $\overline{\text{MEMW}}$
- v.  $\overline{\text{IOR}}$ 、 $\overline{\text{IOW}}$ 、READY
- vi.  $\overline{\text{EOP}}$ : 双向，过程结束信号。

## 5. 时钟周期

a) 空闲周期

b) 有效周期

- i. S0: 过渡周期

接到外设 DREQ 请求开始, 向 CPU 发出 HRQ 信号, 等待 CPU 释放总线  
接到 HLDA 结束

- ii. S1: 地址周期  
ADSTB: 锁存高位地址  
(连续传输数据, 高位地址不变可省略)  
AEN: 送高位地址
  - iii. S2: 地址输出周期  
发出 DACK 信号, 送出 RAM 地址
  - iv. S3: 数据读出周期  
送出数据读控制信号, 数据放到数据线上  
S3 结束时, 查看 READY 信号, 有效进入 S4, 无效插入等待周期  $S_w$
  - v. S4: 数据写入周期  
送数据写控制信号
  - c) 存储器之间数据传输
    - i. 源地址读出数据, 存入暂存寄存器
    - ii. 写入目的地址
  - d) 扩展写: 写信号提前有效, 加快写速度
  - e) 压缩时序: 省略 S3, S4 用于读写
  - f) 一个字节传输最少: S2, S4
6. 8237A 内部寄存器
- a) 通道专用寄存器 (16 位)
    - i. 基本地址寄存器: 内存起始地址
    - ii. 当前地址寄存器: 当前内存地址
    - iii. 基本字节寄存器: 总数 N 时, 设置为 N-1
    - iv. 当前字节寄存器: -1 时, EOP 有效
  - b) 通道公用寄存器
    - i. 工作方式寄存器 (W):  
设置操作类型、操作方式、地址改变方式、自动预置、通道选择
    - ii. 命令寄存器 (W)
    - iii. 状态寄存器 (R): 传输结束、DMA 请求
    - iv. 屏蔽寄存器 (R): 单通道屏蔽、四通道屏蔽
    - v. 请求寄存器
    - vi. 暂存寄存器
  - c) 公用地址 8 个地址
7. 软命令: 对特定地址写即可生效, 与写入数据无关, 无需数据线参与译码
- a) 总清命令 (0DH)
  - b) 清 4 通道屏蔽寄存器 (0EH)
  - c) 清先后触发器 (0FH)
8. PC 中 DMA
- a) 页面地址寄存器
  - b) DACK 作为片选信号寻址 I/O

## 8. 可编程并行接口芯片 8255A

### 1. 引脚

- a) PA0-PA7、PB0-PB7、PC0-PC7
- b) RESET
- c) D7-D0
- d)  $\overline{CS}$ 、 $\overline{WR}$ 、 $\overline{RD}$
- e) A0, A1

### 2. 工作方式

- a) 方式 0, 基本输入/输出方式
- b) 方式 1, 选通的输入输出方式

#### i. 输入

$\overline{STB}$  (PC2, PC4): 选通信号, 接受外设数据

IBF (PC1, PC5): 输入缓冲区满,  $\overline{STB}$  有效置位,  $\overline{RD}$  上升沿复位

INTR (PC0, PC3): 中断请求信号,  $\overline{STB}$  (选通信号结束), IBF, INTE 同时为 1

INTE: 中断允许信号, 由  $\overline{STB}$  控制

#### ii. 输出

$\overline{ACK}$  (PC6, PC2): 外设响应信号, 响应  $\overline{OBF}$

$\overline{OBF}$  (PC7, PC1): 输出缓冲区满,  $\overline{WR}$  上升沿置为位,  $\overline{ACK}$  有效复位

INTR (PC3, PC0):  $\overline{ACK}$ 、 $\overline{OBF}$ 、INTE 为 1 置位

INTE:  $\overline{ACK}$  控制

- c) 方式 2, 带选通信号的双向 I/O 方式

$\overline{STB}$	IBF	$\overline{ACK}$	$\overline{OBF}$	INTR
PC4	PC5	PC6	PC7	PC3

### 3. 编程

- a) 端口地址由低到高: A/B/C/控制

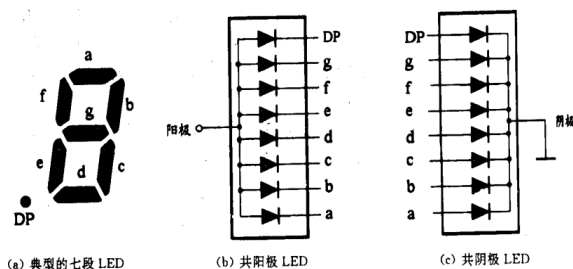
- b) 方式选择控制字

D7	D6	D5	D4	D3	D2	D1	D0
标志位	A-方式选择		Ai/o	PC7-PC4	B 方式	B i/o	PC3-PC0
1	00/01/1x:0/1/2		1:I 0:o				

- c) 端口 C 按位置 1/置 0 控制字

D7	D6	D5	D4	D3	D2	D1	D0
标志位	无关			位选择			置 0/1
0	0	0	0				

### 4. 8 段码



## 9. 串行通信及其接口电路

### 1. 数据传送方式

- a) 单工通信方式
- b) 半双工通信方式
- c) 双工通信方式

### 2. 基本工作方式

- a) 异步通信方式

以字符为单位，字符之间没有固定的时间间隔要求，字符中的各位以固定时间传送

同步方式：起始位，停止位

- b) 同步通信方式

由统一的时钟控制发送方和接收方，若干字符组成一个信息组，字符连续传送，没有字符时发送空闲字符或同步字符

### 3. 波特率：每秒传输数据位数

### 4. 波特率因子 K: 每 bit 数据占时钟周期数

时钟频率 = 波特率因子 \* 波特率

### 5. 8251A: 可编程串行接口芯片

- a) 同步、异步

- b) 字符：5、6、7、8bit

- c) 出错检测

- d) 错误标志：奇偶错误、帧错误、溢出错误

- e) 组成

- i. 数据缓冲寄存器

- ii. 读写控制逻辑

- iii. 接收缓冲器、接收控制器

- iv. 发送缓冲器、发送控制器

- v. 调制解调器控制逻辑：保证数据传输正确，传输过程中不失真

$\overline{\text{DTR}}$ 、 $\overline{\text{DSR}}$ （数据终端准备好、应答）

$\overline{\text{RTS}}$ 、 $\overline{\text{CTS}}$ （请求发送、应答）

- f) 同步方式

- i. 内同步

- ii. 外同步：外部电路检测同步字符

- g) 编程

- i. 端口地址：数据端口、控制端口

- ii. 复位(命令字 40H)、方式字、(同步字符、)命令字、{传输数据，检测状态字}

- h) 方式字、命令字、状态字

- i) CPU 与 8251A 数据交换

- i. 查询方式：查询状态字

- ii. 中断方式

## 10. 数模转换器 DAC0832

1. 组成：数码缓冲寄存器、模拟电子开关、参考电压、解码网络、求和电路  
权电阻网络、倒 T 型电阻网络（0832）
2. DAC0832
  - a) 输入寄存器、DAC 寄存器、D/A 转换器
  - b) 两级锁存
  - c) 工作方式：双缓冲型、单缓冲型、直通型
3. 编程（直写）

```
MOV DX, P-AD
MOV AL, [BX]
OUT DX, AL
```

## 11. 模数转换器 ADC0809

1. 转换过程（基本工作原理）
  - a) 取样：时间上连续变化的信号转化为离散信号
  - b) 保持：保持样值脉冲，以便转换
  - c) 量化：将采样后的样值电平归化到与之接近的离散电平上
  - d) 编码：用二进制数码来表示各个量化电平
2. 转换方式
  - a) 并行比较型
  - b) 逐次逼近型：SAR 从高位开始，取样值与模拟电压- $\Delta/2$  比较，样值较小则为 0
3. ADC0809：8 位 8 通道逐次逼近型 A/D 转换器
4. 工作原理
  - a) 输入 3 位地址信号（ADDC/B/A），在 ALE 脉冲下降沿将地址锁存，经译码选通某一通道的模拟信号进入比较器；
  - b) 发出启动信号 START，在 START 的上升沿将 SAR 清 0，转换结束标志 EOC 变为低电平，在 START 的下降沿开始转换；
  - c) 转换过程在时钟脉冲 CLK 的控制下进行
  - d) 转换结束后，EOC 产生正跳变，在 OE 端输入高电平，从而得到转换结果输出。
5. ADC 与 CPU 接口
  - a) 数据输出：锁存器、三态缓冲器
  - b) 启动信号：脉冲信号、电平信号
  - c) 转换结束的处理方式：中断方式、查询方式、延时方式、DMA 方式
6. 编程
  - a) C、B、A 三个端子选择模拟量的输入(IN0-IN7)端
  - b) ALE 有效时，CBA 上信号存入 ADC 内的地址锁存器中
  - c) START 端有正脉冲，启动 AD 转换  
注：ALE 和 START 接到同一个引脚上。
  - d) 转换结束时，EOC 为“1”
  - e) ADC 的数据输出锁存器后接有缓冲器，缓冲器靠 OE=“1”选中。  
如果 OE=1，锁存器连通 CPU
  - f) 转换好的数据从 D7-D0 端传给 CPU