

微机系统复习资料

吻安，GPT-5.5

根据 2024 级微机系统第一章至第十章课件整理；页码引用格式为“章: 页”

目录

1 微型计算机系统总览	3
1.1 基本概念与组成	3
1.2 性能量与常考计算	3
2 8086/8088 体系结构	3
2.1 BIU 与 EU	3
2.2 寄存器速查	4
2.3 标志寄存器	4
2.4 总线周期与存储体	4
2.5 分段寻址与堆栈	7
3 8086 寻址方式	8
3.1 指令格式要点	8
3.2 操作数寻址速查	8
4 指令系统：按题型掌握	8
4.1 数据传送	8
4.2 算术运算	9
4.3 逻辑、移位和循环移位	10
4.4 串操作	11
4.5 转移、调用、循环与中断	11
5 汇编语言程序设计	14
5.1 伪指令分类与基本写法	14
5.2 程序结构与示例	15
6 存储器与 Cache	16
6.1 存储器指标、分类与芯片结构	16
6.2 CPU 连接、片选和地址范围	17
6.3 容量扩展与 8086 总线接口	18
6.4 Cache 基本流程与地址映射	18
6.5 Cache 替换、一致性与虚拟存储	19
7 中断系统与 8259A	20

7.1	中断概念、请求保存与 8086 响应	20
7.2	中断源识别：查询式到向量式	21
7.3	8259A 结构、连接和响应流程	21
7.4	8259A 工作方式	22
7.5	8259A 编程：ICW、OCW 与读状态	23
8	并行接口与 8255A	23
8.1	I/O 接口、端口与简单并行接口	23
8.2	8255A 结构、端口与引脚	24
8.3	初始化、方式控制字与 BSR	24
8.4	三种工作方式	25
8.5	状态读取与典型应用	25
9	定时/计数器 8253	25
9.1	定时/计数器概述与 8253 结构	26
9.2	六种工作方式：按课件顺序掌握	27
9.3	8253 编程与读写计数值	27
10	串行通信与 8251A	28
10.1	串行通信技术概述	28
10.2	8251A 功能、结构和引脚	28
10.3	8251A 编程与状态判断	29
10.4	常见串行接口标准	31
11	D/A 与 A/D 转换	31
11.1	转换概念与 DAC 原理	31
11.2	DAC0832：结构、时序与输出公式	31
11.3	A/D 转换原理、量化与逐次逼近	32
11.4	ADC0809：通道选择、转换时序与读数	32
12	综合做题模板	32
12.1	标志位与条件转移	32
12.2	存储器扩展与地址范围	33
12.3	接口芯片控制字与转换时序	33
13	考前一页速记	34

1 微型计算机系统总览

1.1 基本概念与组成

- **微处理器**：由一片或几片大规模集成电路组成的，具有运算和控制功能的中央处理器 (CPU, Central Processing Unit)。 [第一章:p10-14]
- **微型计算机**：以微处理器为核心，配上存储器、输入输出接口电路及系统总线所组成的计算机，又称为主机或微电脑。 [第一章:p10-14]
- **单片机**：把微处理器、存储器、输入输出接口电路组装在单个芯片上，称为单片微型计算机。 [第一章:p10-14]
- **微型计算机系统**：指以微型计算机为中心，配上相应的外围设备、电源和辅助电路（通称硬件），以及指挥微机工作的系统软件所构成的系统。 [第一章:p10-14]
- **硬件的核心连接关系是**：CPU 通过**地址总线**（*address bus*）选择单元，通过**数据总线**（*data bus*）传送数据，通过**控制总线**（*control bus*）发出读、写、中断响应等控制信号。 [第一章:p15-19]
- CPU 内部通常包括算术逻辑单元 ALU、控制器和寄存器组。控制器按“取指 → 译码 → 执行”循环工作，寄存器用于暂存操作数、地址和状态。 [第一章:p20-27]
- 存储器按地址组织；读操作把指定单元内容送上数据总线，写操作把数据总线内容写入指定单元。 [第一章:p28-32]

1.2 性能量与常考计算

指标	含义	做题判断
字长	CPU 一次能够处理的二进制位数	通常与内部寄存器和 ALU 宽度相关，不一定等于外部数据总线宽度
寻址能力	地址总线能够表示的地址数	n 根地址线可寻址 2^n 个地址单元
时钟频率	CPU 基本节拍频率	时钟周期 $T = 1/f$ ；指令执行时间还取决于所需时钟数
总线带宽	单位时间可传输的数据量	理想带宽约为“每次传输字节数 × 每秒传输次数”

[第一章:p49-62]

做题提醒：看到“16 位微处理器”时，先确认题目说的是内部字长、数据总线还是地址总线。8088 内部结构按 16 位处理，但外部数据总线为 8 位。

2 8086/8088 体系结构

2.1 BIU 与 EU

- 8086 是 16 位微处理器，外部数据总线 16 位、地址总线 20 位，可直接寻址 $2^{20} = 1\text{MB}$ ；8088 内部仍按 16 位组织，但外部数据总线是 8 位。 [第二章:p4]
- **总线接口部件**（*Bus Interface Unit, BIU*）负责形成物理地址、取指令、存取存储器和 I/O；它含段寄存器、IP、地址加法器和预取队列。8086 队列为 6 字节，8088 为 4 字节。 [第二章:p5-13]

- **执行部件**（*Execution Unit, EU*）负责译码和执行指令。BIU 可在 EU 执行当前指令时预取后续指令，形成简单流水并提高总线利用率。 [第二章:p5-13]
- **指令队列缓冲器**（*instruction queue buffer*）是 BIU 中按先进先出（FIFO）工作的预取队列：BIU 在总线空闲、队列有空位时从内存顺序取下面 1 条或几条指令送入队列，EU 按顺序从队列头取指执行。若发生 JMP/CALL/RET/INT 等改变 CS:IP 的转移，原队列内容不再对应新地址，需要清空并从新地址重新取指；若队列空而 EU 还要取指，EU 只能等待 BIU 补充。8086 队列 6 字节，8088 队列 4 字节。 [第二章:p7-10]

2.2 寄存器速查

类别	寄存器	典型用途
通用寄存器	AX, BX, CX, DX	AX 常作累加器；BX 常作基址；CX 常作计数；DX 用于乘除高半部及间接 I/O 端口地址
指针/变址	SP, BP, SI, DI	SP 指向栈顶；BP 常访问栈中参数；SI/DI 常作源/目的变址，串操作分别对应 DS:SI、ES:DI
段寄存器	CS, DS, SS, ES	代码段、数据段、堆栈段、附加段
控制寄存器	IP, FLAGS	IP 给出下一条指令的段内偏移；FLAGS 保存状态和控制标志

[第二章:p14-20；第三章:p22-30]

2.3 标志寄存器

标志	名称	置 1 条件或作用
CF	进位标志	最高位产生进位或借位；主要用于无符号数比较和多字长运算
PF	奇偶标志	结果低 8 位中 1 的个数为偶数
AF	辅助进位	位 3 向位 4 有进位/借位；BCD 调整使用
ZF	零标志	运算结果为 0
SF	符号标志	等于结果最高位
OF	溢出标志	有符号结果超出表示范围；等价于符号位的入进位异或出进位
DF	方向标志	串操作中 0 表示 SI/DI 递增，1 表示递减
IF	中断允许	1 允许可屏蔽中断 INTR，0 屏蔽
TF	单步标志	1 时每执行一条指令产生单步中断

[第二章:p22-27；第三章:p55-75]

做题提醒：无符号数看 CF/ZF；有符号数看 SF、OF、ZF。CF 和 OF 完全不是一回事。

2.4 总线周期与存储体

- 一个基本总线周期通常由 T_1 至 T_4 组成，用来完成一次存储器或 I/O 访问；若外设/存储器未准备好，READY 无效会在 T_3 与 T_4 之间插入一个或多个等待状态 T_W 。 [第二章:p28-31]
- 8086 的地址/数据引脚复用，因此系统常用 8282 或 74LS373 锁存地址，用 8286 或 74LS245 作总线收发器；最小方式适合单处理器，最大方式常配 8288 总线控制器。 [第二章:p34-56]
- 8086 的 16 位存储器分为低字节偶地址体和高字节奇地址体，以 A_0 和低有效信号 $\overline{BHE}$ 选择； $\overline{BHE} = 0$

时选通高字节体。

[第二章:p34-46; p57-61]

8086/8088 最小方式引脚功能图 ($MN/\overline{MX} = 1$, 只保留最小方式信号)			
8086 最小方式		8088 最小方式	
GND	VCC	GND	VCC
AD14	AD15	A_{14}	A_{15}
AD13	A_{16}/S_3	A_{13}	A_{16}/S_3
AD12	A_{17}/S_4	A_{12}	A_{17}/S_4
AD11	A_{18}/S_5	A_{11}	A_{18}/S_5
AD10	A_{19}/S_6	A_{10}	A_{19}/S_6
AD9	$\overline{BHE}/S_7$	A_9	SS_0 (HIGH)
AD8	$MN/\overline{MX} = 1$	A_8	$MN/\overline{MX} = 1$
AD7	$\overline{RD}$	AD7	$\overline{RD}$
AD6	HOLD	AD6	HOLD
AD5	HLDA	AD5	HLDA
AD4	$\overline{WR}$	AD4	$\overline{WR}$
AD3	$M/\overline{IO}$	AD3	$M/\overline{IO}$
AD2	$DT/\overline{R}$	AD2	$DT/\overline{R}$
AD1	$\overline{DEN}$	AD1	$\overline{DEN}$
AD0	ALE	AD0	ALE
NMI	$\overline{INTA}$	NMI	$\overline{INTA}$
INTR	TEST	INTR	TEST
CLK	READY	CLK	READY
GND	RESET	GND	RESET

[第二章:p32; p48-54]

引脚	方向/有效性	标注
NMI	输入, 上升沿触发	非屏蔽中断请求 (Non-Maskable Interrupt)。不受 IF 标志控制, 通常用于掉电、故障等不能被屏蔽的紧急中断。
INTR	输入, 高有效	可屏蔽中断请求 (Interrupt Request)。只有 IF=1 且当前指令结束后, CPU 才响应该请求。
CLK	输入	系统时钟输入, 给 CPU 内部操作和总线时序提供基本节拍。
$\overline{RD}$	输出, 低有效	读控制信号 (Read)。为低电平时表示 CPU 正在从存储器或 I/O 端口读取数据。
HOLD	输入, 高有效	总线保持请求。外部总线主控部件请求占用系统总线时置 1。
HLDA	输出, 高有效	总线保持响应 (Hold Acknowledge)。CPU 承认 HOLD 后释放总线并输出 HLDA=1。
$\overline{WR}$	输出, 低有效	写控制信号 (Write)。为低电平时表示 CPU 正在向存储器或 I/O 端口写入数据。
DT/ $\overline{R}$	输出	数据收发方向控制 (Data Transmit/Receive)。常接总线收发器方向端; 高电平通常表示 CPU 发送数据, 低电平表示 CPU 接收数据。
$\overline{DEN}$	输出, 低有效	数据允许信号 (Data Enable)。常接总线收发器使能端, 允许数据总线驱动器工作。
ALE	输出, 高有效脉冲	地址锁存允许 (Address Latch Enable)。在 T_1 期间锁存 AD 线上的地址, 用于分离地址/数据复用线。
$\overline{INTA}$	输出, 低有效	中断响应 (Interrupt Acknowledge)。CPU 响应 INTR 后发出, 用于通知外设/中断控制器送出中断类型码。
TEST	输入, 低有效	等待测试信号。执行 WAIT 时若 TEST=1, CPU 空转等待; TEST=0 时退出等待。
READY	输入, 高有效	准备好信号。READY=1 表示存储器/I/O 可完成传送; READY=0 时在总线周期中插入等待状态 T_W 。
RESET	输入, 高有效	复位信号。有效时使 CPU 初始化, 重新进入复位后的起始状态。

[第二章:p32; p41-54]

做题提醒: 最小方式下 CPU 直接给出存储器/I/O 读写、地址锁存、数据允许/方向、中断响应和总线保持信号; 记图时先看 $MN/\overline{MX} = 1$, 再把 8086 的 AD0-AD15 与 8088 的 A8-A15、AD0-AD7 区分开。

状态	总线上的主要动作	做题抓手
T_1	CPU 在复用地址/数据线上输出地址信息，并给出能让外部锁存器保存地址的时序；存储器或 I/O 端口在这一拍被选中。	看到“地址有效、ALE 锁存地址”，基本就是 T_1 。地址/数据复用的分离从这里开始。
T_2	复用线从“送地址”转为“传数据”的方向准备；CPU 给出读/写等控制信号，读周期准备接收数据，写周期开始送出待写数据。	重点是方向和控制信号建立，不要把 T_2 当成最终采样时刻。
T_3	数据传送的主要时段；读周期中外设/存储器把数据送上数据总线，写周期中 CPU 保持数据稳定。CPU 在此附近检查 READY。	若 READY 表示“未准备好”，正常 T_3 后不能直接进入 T_4 ，要插入 T_W 。
T_W	等待状态；地址、控制和数据传送关系保持，CPU 空等慢速外设/存储器准备好。	T_W 不是固定每次都有，而是 READY 无效时才插入，可连续多个。
T_4	总线周期收尾；读周期 CPU 锁存/接收数据，写周期撤销写控制，复用总线回到空闲或进入下一个周期。	一个总线周期结束，之后可接下一次读/写，也可能进入空闲状态。

[第二章:p28-31]

$\overline{\text{BHE}}$	A_0	被选存储体	传送
0	0	高、低两体	偶地址开始的一个字，1 个总线周期
0	1	高字节体	奇地址的一个字节
1	0	低字节体	偶地址的一个字节
1	1	均不选	无传送

[第二章:p57-61]

做题提醒：从奇地址读取一个 16 位字必须分两次总线周期；从偶地址读取规则字只需一次。

2.5 分段寻址与堆栈

物理地址 $PA = \text{段地址} \times 10_{\text{H}} + \text{偏移地址}$

- 20 位物理地址由 16 位段值左移 4 位再加 16 位偏移形成。每段最大 64KB，段起始物理地址一定能被 16 整除，不同逻辑段可以重叠。

[第二章:p63-68]

- 默认组合包括 CS:IP 取指、SS:SP 访问栈顶、DS 配合多数数据地址、ES:DI 作为串操作目的地址。[第二章:p63–71]
- 堆栈按后进先出工作，8086 栈向低地址增长。PUSH 一个字时先令 SP 减 2 再写入；POP 读出后令 SP 加 2。[第二章:p69–71]

3 8086 寻址方式

3.1 指令格式要点

- 8086 基本机器指令通常为 1–6 字节，由操作码、寻址方式字段和立即数/位移量等组成；段超越、REP、LOCK 等前缀还会各占一个前缀字节。D 常表示传送方向，W 表示字节或字，MOD/REG/R/M 共同描述寄存器和存储器操作数。[第三章:p3–10； p86–95； p125–128]
- MOD=00 通常无位移量，MOD=01 带符号扩展的 8 位位移，MOD=10 带 16 位位移，MOD=11 表示寄存器操作数。**例外：MOD=00 且 R/M=110 表示直接寻址，后跟 16 位地址位移，即 EA=disp16。**因此无位移的 [BP] 不能用 MOD=00 编码，通常编码为 MOD=01、R/M=110、disp8=0。¹ [第三章:p3–10； p22–25]

3.2 操作数寻址速查

方式	示例	核心判断
立即寻址	MOV AX,1234H	常数在指令中，只能作源操作数
寄存器寻址	MOV AX,BX	操作数在寄存器中，速度快
直接寻址	MOV AX,[1234H]	EA 就是指令给出的位移量，默认段为 DS
寄存器间接	MOV AX,[BX]	EA 在 BX/BP/SI/DI 中；BP 默认用 SS，其余默认 DS
基址/变址	[BX+SI+disp]	EA 可由基址、变址和位移相加；不能任意组合寄存器
串寻址	MOVSB	源固定 DS:SI，目的固定 ES:DI，方向由 DF 决定
I/O 直接	IN AL,80H	8 位端口立即数，范围 00H–FFH
I/O 间接	OUT DX,AL	端口号在 DX，可覆盖 0000H–FFFFH

[第三章:p11–32]

存储器操作数 PA = 默认或超越段寄存器 × 10_H + EA

做题提醒：有效地址组合中，基址只能取 BX 或 BP，变址只能取 SI 或 DI。出现 BP 时默认段为 SS；段超越前缀可改变默认段，但串操作目的仍固定为 ES:DI。

4 指令系统：按题型掌握

4.1 数据传送

- MOV 不影响标志位。常见限制：两个操作数长度一致；不能存储器到存储器；CS 不能作目的；段寄存器之间不能直接传送；立即数不能直接送段寄存器。[第三章:p34–40]
- XCHG 交换两个同长度操作数，但不允许两个存储器操作数；PUSH/POP 以字为单位操作栈。[第三章:p41–45]

¹Intel iAPX 86/88 User’s Manual, Table 1-20.

- LEA 取有效地址；LDS/LES 从连续 4 字节同时装入一个通用寄存器和 DS/ES。[第三章:p46-49]
- IN/OUT 在累加器 AL/AX 与端口之间传送；端口可由 8 位立即数或 DX 指定。[第三章:p50-52]
- XLAT 用 AL 作表内偏移，执行后 $AL \leftarrow DS:[BX+AL]$ ，适合查表转换。[第三章:p53-54]

4.2 算术运算

类别	指令	易错点
加法	ADD, ADC, INC	ADC 加上 CF，适合多字长；INC 不影响 CF
减法	SUB, SBB, DEC, NEG, CMP	SBB 减去 CF；DEC 不影响 CF；CMP 只改标志、不保存结果
乘法	MUL, IMUL	单操作数形式隐含 AL/AX；8 位积在 AX，16 位积在 DX:AX
除法	DIV, IDIV	8 位除法被除数 AX；16 位除法被除数 DX:AX；商和余数写入隐含寄存器
扩展	CBW, CWD	CBW 将 AL 符号扩展到 AX；CWD 将 AX 符号扩展到 DX:AX
BCD 调整	AAA, DAA, AAS, DAS, AAM, AAD	必须与相应加减乘除步骤配合，不能把调整指令当普通算术指令

[第三章:p55-75]

指令	名称/含义	复习时要会说出的作用
ADD	加法	目的操作数加源操作数，结果送回目的操作数；影响 CF、OF、SF、ZF、PF、AF。
ADC	带进位加法	在 ADD 基础上再加 CF，常用于多字节/多字加法的高位部分。
INC	加 1	操作数加 1，但不影响 CF；循环计数或地址递增常见。
SUB	减法	目的操作数减源操作数，结果送回目的操作数；用于普通减法。
SBB	带借位减法	在 SUB 基础上再减 CF，常用于多字节/多字减法的高位部分。
DEC	减 1	操作数减 1，但不影响 CF；循环变量递减常见。
NEG	求补/变号	用 0 减操作数，得到二进制补码意义下的相反数。
CMP	比较	本质做减法并只改标志位，不保存结果；后面通常接条件转移。
MUL	无符号乘法	隐含被乘数在 AL 或 AX；8 位乘法积在 AX，16 位乘法积在 DX:AX。
IMUL	有符号乘法	与 MUL 类似，但把操作数按补码有符号数解释。
DIV	无符号除法	8 位除法用 AX 作被除数，商 AL、余 AH；16 位除法用 DX:AX 作被除数，商 AX、余 DX。
IDIV	有符号除法	与 DIV 类似，但按补码有符号数解释；除法前常要用 CBW/CWD 扩展被除数。
CBW	字节扩展为字	把 AL 的符号扩展到 AH，形成 AX，常用于 8 位有符号除法前。

CWD	字扩展为双字	把 AX 的符号扩展到 DX，形成 DX:AX，常用于 16 位有符号除法前。
DAA	压缩 BCD 加法调整	Decimal Adjust after Addition; ADD/ADC 两个压缩 BCD 数后调整 AL，使结果仍是压缩 BCD。
DAS	压缩 BCD 减法调整	Decimal Adjust after Subtraction; SUB/SBB 两个压缩 BCD 数后调整 AL。
AAA	非压缩 BCD 加法调整	ASCII Adjust after Addition; 两个非压缩 BCD 数相加后调整 AX，低位数字留在 AL，进位反映到 AH/CF/AF。
AAS	非压缩 BCD 减法调整	ASCII Adjust after Subtraction; 两个非压缩 BCD 数相减后调整 AX，配合借位修正 AH/AL。
AAM	非压缩 BCD 乘法调整	ASCII Adjust after Multiplication; MUL 得到一位十进制乘积后，把 AL 中的二进制结果调整为 AH= 十位、AL= 个位。
AAD	非压缩 BCD 除法调整	ASCII Adjust before Division; DIV 前把 AH:AL 中的十位、个位合成为 AL 中的二进制数，AH 清 0。

[第三章:p55-75]

做题提醒：做乘除题先写出“隐含被乘数/被除数”和“结果寄存器”，再代数值。DIV 商超范围或除数为 0 会触发除法错误。

4.3 逻辑、移位和循环移位

- AND/OR/XOR 保存逻辑结果；TEST 只改标志不保存结果；NOT 按位取反且不影响标志。 [第三章:p76-80]
- SHL/SAL 左移并以 0 补低位；SHR 逻辑右移并以 0 补高位；SAR 算术右移并复制符号位。 [第三章:p81-85]
- ROL/ROR 只在操作数内部循环；RCL/RCR 把 CF 也纳入循环。移位次数只能是 1 或放在 CL 中。 [第三章:p81-85]

指令	全称/方向	关键含义
SHL	Shift Logical Left	逻辑左移，最低位补 0，最高移出位进 CF；无符号数可看作乘 2，但要检查溢出/进位。
SAL	Shift Arithmetic Left	算术左移；在 8086 中与 SHL 操作相同，只是常按有符号数语境理解。
SHR	Shift Logical Right	逻辑右移，最高位补 0，最低移出位进 CF；无符号数可看作除 2。
SAR	Shift Arithmetic Right	算术右移，最高位复制原符号位，最低移出位进 CF；用于有符号数右移。
ROL	Rotate Left	不带进位循环左移：最高位移到最低位，同时进入 CF。
ROR	Rotate Right	不带进位循环右移：最低位移到最高位，同时进入 CF。
RCL	Rotate through Carry Left	带进位循环左移：把 CF 当成操作数旁边的额外一位一起循环，常用于多字长左移。
RCR	Rotate through Carry Right	带进位循环右移：把 CF 当成额外一位一起循环，常用于多字长右移或串行位处理。

[第三章:p80-85]

做题提醒：SHL 和 SAL 在机器动作上相同；ROL/ROR 不经过 CF 循环，RCL/RCR 会把 CF 当成第 9/17 位参与循环。移位次数若不是 1，必须先放入 CL。

4.4 串操作

指令	操作	关键寄存器
MOVS	串传送	DS:SI → ES:DI
CMPS	比较两个串元素	比较 DS:SI 与 ES:DI，修改标志
SCAS	扫描串	比较 AL/AX 与 ES:DI
LODS	装入累加器	DS:SI → AL/AX
STOS	存储累加器	AL/AX → ES:DI

[第三章:p86-95]

- CLD 令 DF=0，使 SI/DI 递增；STD 令 DF=1，使其递减。字节操作每次变化 1，字操作每次变化 2。[第三章:p86-95]
- REP 按 CX 重复；REPE/REPZ 在 CX 非零且 ZF=1 时继续；REPNE/REPNZ 在 CX 非零且 ZF=0 时继续。[第三章:p86-95]
- LODSB/LODSW 将 DS:SI 指向的串元素装入 AL/AX，并按 DF 修改 SI。它**一般不使用重复前缀**，因为每重复一次都会改写累加器，最后只保留最后一次装入的数据；更常用于显式循环中逐个取数并统计正数、负数、0 或字符类别。[第三章:p94]

4.5 转移、调用、循环与中断

- 程序控制类指令主要包括：子程序调用和返回、无条件/条件转移、循环控制、中断控制。除中断类会按中断机制保存 FLAGS 等现场外，普通 CALL/JMP/RET/Jcc/LOOP 本身通常不把运算结果写回通用寄存器，也不按算术方式改变标志位。[第三章:p96-124]
- **调用和转移的区别：**CALL 用于进入子程序，调用前必须把断点地址压入堆栈，子程序最后用 RET 回到原调用处；JMP 只是直接改 IP 或 CS:IP，不保存返回地址，目标代码执行后不会自动回到原处。[第三章:p97-107； p119]

CALL 四种调用方法 [第三章:p98-103]

类型	例子	动作和记忆点
段内直接调用	CALL 1000H	子程序与调用处在同一代码段；先压入返回 IP，再把指令给出的 16 位偏移量送入 IP。
段内间接调用	CALL BX	先压入返回 IP，再由寄存器或存储器操作数给出新的 IP；如 CALL WORD PTR [SI]。
段间直接调用	CALL 2500:3600	子程序不在同一代码段；先保护断点的 CS 和 IP，再把指令中的段地址和偏移地址装入 CS:IP。
段间间接调用	CALL DWORD PTR [DI]	操作数给出 4 字节入口地址的存放位置，低字为 IP，高字为 CS；适合远过程入口地址存放在内存中的情况。

- RET 作为子程序最后一条指令使用。段内返回弹出 IP；段间返回还要恢复 CS。RET EXP 先弹出返回地址，

再令 SP 额外增加 EXP，用来丢弃调用前压栈传入、返回后已经无用的参数或参数地址；EXP 通常按字节数写成偶数。[第三章:p104–105]

转移指令先分两类[第三章:p106]

- 无条件转移：JMP 无条件到目标地址执行新指令。
- 条件转移：Jcc 以一个或多个标志位，或 CX 的值，决定是否转移；条件不满足就顺序执行下一条。

JMP 五种写法[第三章:p107–110]

类型	例子	动作和范围
段内直接短转移	JMP SHORT NEXT	8 位有符号位移，相对下一条指令地址转移，范围约 −128 至 +127 字节。
段内直接近转移	JMP 1000H	16 位位移或偏移量，仍在当前 CS 内转移，只改变 IP。
段内间接转移	JMP BX	目标偏移由寄存器或存储器操作数给出，如 JMP WORD PTR [SI]；只改变 IP。
段间直接远转移	JMP 2000:0100H	指令中直接给出新的段地址和偏移地址，同时改变 CS 和 IP。
段间间接远转移	JMP DWORD PTR [SI]	内存双字给出远地址，低字送 IP，高字送 CS。

条件转移四大类[第三章:p111–117]

类别	指令	条件和说明
单标志 ZF	JZ/JE, JNZ/JNE	结果为 0 或相等：ZF=1；结果不为 0 或不相等：ZF=0。
单标志 SF	JS, JNS	结果为负：SF=1；结果为正/非负：SF=0。
单标志 OF	JO, JNO	溢出：OF=1；不溢出：OF=0。
无符号	JB/JNAE/JC	低于、不高于也不等于、有借位或进位：CF=1。无符号数比较用 below/above 这套词。
无符号	JNB/JAE/JNC	不低于、高于或等于、无借位或无进位：CF=0。
无符号	JBE/JNA	低于或等于、不高于：CF=1 或 ZF=1。
无符号	JNBE/JA	高于：CF=0 且 ZF=0。
有符号	JL/JNGE	小于、不大于也不等于： $SF \oplus OF = 1$ 。有符号数比较用 less/greater 这套词。
有符号	JNL/JGE	不小于、大于或等于： $SF \oplus OF = 0$ 。
有符号	JLE/JNG	小于或等于、不大于： $SF \oplus OF = 1$ 或 ZF=1。
有符号	JNLE/JG	大于： $SF \oplus OF = 0$ 且 ZF=0。
CX 值判断	JCXZ	若 CX=0 则转移，常用于进入循环前先判断计数是否为 0。

条件转移做题口令：所有条件转移都是相对短转移，目标范围约为 -128 至 +127 字节；比较大小前先问“无符号还是有符号”。无符号看 CF/ZF，用 below/above；有符号看 $SF \oplus OF$ 与 ZF，用 less/greater。很多条件有两到三个助记符，它们机器条件相同，只是语义名称不同。

[第三章:p111-117]

循环指令

[第三章:p120-123]

- 循环指令的目标地址也是短相对范围，循环次数由 CX 指定。
- LOOP LABEL 先令 CX 减 1，再在 CX≠0 时转移，等价于 DEC CX 后 JNZ LABEL。
- LOOPZ/LOOPE 在 CX≠0 且 ZF=1 时继续循环；LOOPNZ/LOOPNE 在 CX≠0 且 ZF=0 时继续循环。它们是在 CX 计数之外再增加 ZF 条件。

中断控制指令

[第三章:p124]

指令	类型/长度	说明
INT n	软件中断，2 字节	n 为 0-255 的中断向量号，通过中断向量表转入相应中断服务程序。
INT0	溢出中断，1 字节	仅当 OF=1 时产生中断，固定使用中断向量号 4；OF=0 时继续顺序执行。
INT3	调试断点中断，1 字节	软件断点中断，常用于调试程序，固定使用 3 型中断。
IRET	中断返回	从中断服务程序返回，按 IP、CS、FLAGS 恢复现场；这与普通 RET 只恢复返回地址不同。

- 处理器控制指令不直接完成普通算术或数据传送，而是改变标志、控制处理器状态，或配合外部处理器/总线操作。

[第三章:p125-128]

指令	作用对象	含义和常见用途
CLC	CF	Clear Carry，清进位标志，令 CF=0。
CMC	CF	Complement Carry，进位标志取反，令 $CF \leftarrow \overline{CF}$ 。
STC	CF	Set Carry，置进位标志，令 CF=1。
CLD	DF	Clear Direction，令 DF=0，串操作后 SI/DI 自动增加。
STD	DF	Set Direction，令 DF=1，串操作后 SI/DI 自动减少。
CLI	IF	Clear Interrupt，令 IF=0，屏蔽可屏蔽中断 INTR。
STI	IF	Set Interrupt，令 IF=1，允许响应可屏蔽中断 INTR。
NOP	无操作	No Operation，不改变寄存器和标志；常用于占位、延时或对齐。
HLT	处理器暂停	Halt，使 CPU 进入暂停状态；通常由外部中断、NMI 或复位等事件唤醒/退出。
WAIT	TEST 引脚	等待测试信号；执行后若 TEST 为高电平则空转等待，直到 TEST 变低或有中断/复位等事件。注意它不是总线等待状态 T_W 。
ESC	协处理器/外部处理器	Escape，交权/逃逸指令，用于把后续操作交给 8087 等外部协处理器配合执行。
LOCK	总线锁定前缀	指令前缀；使下一条可锁定的存储器操作在执行期间锁住总线，避免其他总线主设备插入，常用于原子读改写。

[第三章:p125-128]

5 汇编语言程序设计

5.1 伪指令分类与基本写法

- 伪指令 (*pseudo-instruction/directive*) 不是 CPU 运行时直接执行的机器指令，而是给汇编程序看的“说明书”：定义段、分配存储单元、定义符号、标记过程、展开宏等。

[第四章:p7]

类别	常用伪指令	复习时要会读什么
段定义伪指令	SEGMENT/ENDS, ASSUME, END	SEGMENT/ENDS 成对定义代码段、数据段、堆栈段等；ASSUME 只告诉汇编器段寄存器和段名的对应关系，除 CS 外，DS/ES/SS 仍要在程序中用指令装入；END START 指出主程序入口。
等值语句伪指令	EQU, =	给常数、表达式、变量地址或助记符定义符号名，但不分配存储单元。EQU 左边符号通常不能重新定义；等号伪指令可再次赋值。
数据定义伪指令	DB, DW, DD, DQ, DT, DUP	真正分配存储单元：DB/DW/DD/DQ/DT 分别按 1/2/4/8/10 字节单位定义；DUP 表示重复；? 表示只保留空间不赋初值。多字节数按小端方式存放，低字节在低地址。
子程序伪指令	PROC/ENDP, PUBLIC, EXTRN	PROC/ENDP 给过程定界，过程名也是可被 CALL 的符号地址；PUBLIC 声明本模块符号可外用，EXTRN 声明本模块使用外部定义的符号。
定位/基数控制伪指令	EVEN, ORG, .RADIX	EVEN 使下一个字节地址成为偶地址；ORG 常数表达式把下一个存储单元定位到指定偏移；.RADIX 改变默认数制，取值范围通常为 2-16。
宏定义伪指令	MACRO/ENDM, LOCAL, PURGE, INCLUDE	宏必须先定义后调用；宏指令使用可分为宏定义、宏调用、宏扩展三步。MACRO 到 ENDM 是宏体，调用时由宏汇编程序在汇编期展开。LOCAL 处理宏体内标号重名，PURGE 取消宏定义，INCLUDE 引入宏库。

[第四章:p7-24； p37-48]

小例子	读法
DATA SEGMENT ... DATA ENDS	定义一个数据段，段名 DATA 只是符号；如果代码要访问它，通常还要在入口处执行 MOV AX, DATA、MOV DS, AX。
COUNT EQU 100	定义符号常量 COUNT，使用处相当于写常数 100，不占内存单元。
BUF DB 20 DUP(?)	分配 20 个字节但不初始化，适合作缓冲区。
DATA2 DW 3010H	定义 1 个字，内存低地址存 10H，高地址存 30H。
SUM PROC NEAR ... RET ... SUM ENDP	定义段内近过程，调用时用 CALL SUM，返回时执行 RET。
EVEN; ORG 1000H	EVEN 用于让后续数据从偶地址开始；ORG 1000H 表示后续分配从偏移 1000H 处继续。
.RADIX 16	把默认基数改为 16；未带后缀的数按十六进制理解，之后可用 .RADIX 10 改回十进制。

[第四章:p8-23]

- 属性操作符：OFFSET 取偏移、SEG 取段值、TYPE 取单元素字节数、LENGTH 取重复元素数、SIZE 通常为 TYPE×LENGTH；PTR 临时指定操作数类型，例如 WORD PTR d1 或 BYTE PTR d2。 [第四章:p25-27]

最小程序骨架读法：ASSUME CS:CODE, DS:DATA 只是建立汇编器认知；真正让 DS 指向数据段，仍要写 MOV AX, DATA、MOV DS, AX。最后 END START 的 START 告诉装入程序从哪条标号处开始执行；若是子模块，END 通常不带入口标号。

宏三步读法：宏定义是用 MACRO/ENDM 写出宏体，如 SHIFT MACRO A,B,C 中的 A,B,C 是形参；宏调用是写 SHIFT HL,BX,3，其中 HL,BX,3 是实参；宏扩展是在汇编期用实参替换形参，并把宏体插入调用处，展开成类似 MOV CL,03H 和 SHL BX,CL 的实际指令。形参还可以是助记符的一部分，如 S&A 配合实参 HL/HR 展开为 SHL/SHR。

[第四章:p39-43]

5.2 程序结构与示例

结构	常见写法	做题/读程序重点
顺序程序	POP AX; POP BX; ADD AX,BX; PUSH AX	按指令顺序执行，重点看每条指令对寄存器、内存和标志位的连续影响。
分支程序	CMP/TEST 后接 Jcc; 两分支常用 JMP FINISH 汇合	先判断比较对象是有符号还是无符号，再选条件转移；一个功能段尽量保持一个入口和一个出口。
循环程序	MOV CX,N ... LOOP AGAIN, 或 CMP ... JNE AGAIN	计数控制常用 CX 和 LOOP；条件控制靠比较结果决定是否继续。每个循环都要能指出初始化、循环体、变量修改和结束条件。
子程序	PROC/ENDP 定义,CALL 调用, RET 返回	参数可通过寄存器、堆栈或存储器传递；子程序若改了调用者还要用的寄存器，应先保护，最后恢复。

[第四章:p28-36]

- 分支例子读法：CMP AX,0 后若用 JS NEGATIVE，说明按符号位判断负数分支；正数分支执行完要用 JMP

- FINISH 跳过负数分支，最后在 FINISH 汇合。[第四章:p29–30]
- 循环例子读法：LOOP LABEL 隐含“CX 先减 1，再在 $CX \neq 0$ 时跳转”；若内外层循环都想用 CX，进入内层前要保存外层 CX，或把某一层计数改放到其他寄存器/内存中。[第四章:p31–32]
- 子程序例子读法：CALL SUB1 会先把返回地址压栈，再转入 SUB1；RET 再把返回地址弹出，使程序回到调用点的下一条指令。子程序嵌套和递归都依赖堆栈空间，递归必须有出口条件。[第四章:p33–35]

对比点	子程序	宏指令
代码空间	程序中通常只有一段可执行代码	每调用一次就展开一段代码，代码可能变长
执行时间	运行时用 CALL/RET 跳转和返回，有调用开销	汇编期已展开，运行时没有跳转返回过程，通常更快
定义位置	只要连接生成前可见即可	必须先定义后调用
适合场景	平时编程、代码复用、节省空间	核心短功能、频繁调用且希望减少运行开销的片段

[第四章:p37–48]

做题提醒：读汇编程序时先分层：哪些是伪指令、哪些是 CPU 真正执行的指令。伪指令决定“怎么汇编、内存怎么分配、符号怎么替换”；真正改变 AX、DS、FLAGS、内存内容的是机器指令。

6 存储器与 Cache

6.1 存储器指标、分类与芯片结构

芯片容量 = 存储单元数 × 每单元位数		n 根片内地址线 $\Rightarrow 2^n$ 个单元
指标	含义	做题读法
存储容量	可保存的二进制位数，常写成“单元数 × 位数”。	6116 为 16K bit，即 $2K \times 8 = 2KB$ ；6264 为 64K bit，即 $8K \times 8 = 8KB$ 。同理 28256 为 256K bit，即 32KB；27512 为 512K bit，即 64KB。
存取速度	从 CPU 给出地址开始，到数据读出或写入完成所需时间。	时间越小越快。芯片标号中的 -7、-10 常表示约 70 ns、100 ns；若存储器慢于总线周期，要插入等待状态 T_W 。
可靠性	工作稳定性与单位容量成本。	考试多用来解释“为什么要多级存储”：高速器件贵，低速器件容量大而便宜。

[第五章:p5–8; p33]

- 存储层次自上而下大致为寄存器、Cache、主存、磁盘缓存、硬盘/移动硬盘；容量越来越大，速度越来越慢。Cache 可提高速度，但计算机可以没有 Cache，不能没有主存；CPU 能直接访问主存，不能直接访问辅存中的数据。[第五章:p7–8]
- RAM 可随机读写但通常断电失效；ROM 断电后信息不丢失，按写入方式可分为生产时写入、一次性写入、多次改写、光擦除、电擦除等类型。[第五章:p13–15]
- 存储器芯片内部通常包括存储体矩阵、地址译码器、读写控制逻辑和三态数据缓冲器。地址译码器从片内地址线中选中一个存储单元；三态缓冲器保证未选中的芯片不会占用数据总线。[第五章:p9–10]

引脚/信号	常见记号	作用
地址线	A0–An	连接片内地址译码器，选择芯片内部唯一存储单元； k 根地址线对应 2^k 个片内单元。
数据线	D0–Dm	传送数据；RAM 数据线通常双向，ROM/EPROM 读出方向为主。
片选	$\overline{\text{CS}}$	低有效时选中芯片；未选中时数据输出高阻，避免多片同时驱动总线。
读允许	$\overline{\text{OE}}$	Output Enable，低有效时允许芯片把数据送到数据线上。
写允许	$\overline{\text{WE}}$	Write Enable，低有效时把数据线上的数据写入被选中单元。
早期读写控制	$R/\overline{W}$	用一个方向信号区分读、写，具体有效电平按芯片手册。

[第五章:p10–11]

6.2 CPU 连接、片选和地址范围

- 8086/8088 对存储器有 20 根地址线，可形成 1MB 存储器地址空间；I/O 端口地址线为 16 根。8086 外部数据线 16 根，8088 外部数据线 8 根。[第五章:p16]
- 存储器芯片与 CPU 的基本连接：数据线按数据位宽并到 CPU 数据总线；低位地址线接芯片片内地址端；剩余高位地址线用于片选；读写控制线要由 CPU 的存储器/I/O 选择信号与 $\overline{\text{RD}}$ 、 $\overline{\text{WR}}$ 组合，生成只在存储器周期有效的读/写信号。[第五章:p17–19]
- 读写控制不能只看 $\overline{\text{RD}}/\overline{\text{WR}}$ ，还要问“这是存储器周期还是 I/O 周期”。因此常把 M/IO 与读写控制组合成 MEMR/MEMW 类信号，再接到存储器的 $\overline{\text{OE}}/\overline{\text{WE}}$ 。[第五章:p19]

片选方法	做法	优缺点与注意
线选法	直接用 CPU 的某些高位地址线作为芯片片选信号。	线路简单，适合小容量系统；但没有充分译码，容易浪费地址空间或形成镜像地址。总线上 不能同时选中两片输出芯片 。
译码法	把高位地址线送入译码器，译码输出作为片选信号。	能更有效利用地址空间，便于得到连续地址段。74LS138 是常用 3-8 译码器，控制端有效时，输入 CBA 从 000 到 111 选择 $\overline{Y_0}$ 到 $\overline{Y_7}$ 中某一路为低。

[第五章:p20–27]

做题提醒：地址范围题先分两层：低位地址线决定“片内从 0 到最大”的范围，高位地址线决定“这片芯片什么时候被选中”。线选法常有地址浪费或不连续；译码法看译码器输入和使能条件，逐个写出高位固定模式。

6.3 容量扩展与 8086 总线接口

扩展类型	含义	例子和做题步骤
位扩展	增加每个存储单元的位数，也就是扩展字长。	两片 $2K \times 4$ 并联片内地址线和片选线，数据线分别接低 4 位和高 4 位，可组成 $2K \times 8$ 。
字数扩展	增加存储单元数量，也就是扩展地址容量/字数。	四组 $2K \times 8$ 通过不同片选选中，可组成 $8K \times 8$ 。先算每组需要多少片，再算需要多少组。
混合扩展	同时扩展位宽和字数。	总片数 = 位扩展片数 $\times$ 字数扩展组数；低位地址线接片内地址，高位地址译码产生每组片选。

[第五章:p16–27]

- 8086/8088 的地址/数据复用线需要锁存：8086 的 AD0–AD15、8088 的 AD0–AD7 在 T_1 送地址，之后改作数据线；A16–A19 也与状态信号复用。常用 8282 或 74LS373 配合 ALE 锁存地址。 [第五章:p28–29]
- 8282 可看成 8 位地址锁存器：DI 为输入，DO 为锁存输出，STB 接 ALE， $\overline{OE}$ 有效时输出地址。粗略数片数时，8086 通常 AD0–AD15 需两片，再锁存高位地址/状态线需一片；8088 通常 AD0–AD7 需一片，再锁存高位地址/状态线需一片。 [第五章:p28–29]
- 数据总线常用 8286 或 74LS245 作双向总线驱动器。8086 的 $DT/\overline{R}$ 控制数据方向，DEN 控制输出允许；74LS245 中 $\overline{G}$ 为低有效使能，DIR 控制 A 到 B 或 B 到 A。 [第五章:p28–30]
- 8086 有 16 根数据线，但多数存储器芯片为 8 位，因此 16 位数据传送通常要同时选中偶地址低字节库和奇地址高字节库。低字节库接 D7–D0，存偶地址字节；高字节库接 D15–D8，存奇地址字节，由 A_0 与 $\overline{BHE}$ 共同选择。 [第五章:p31–32]

$\overline{BHE}$	A_0	8086 存储体选择
0	0	高、低两库同时选中，偶地址开始的 16 位字一次传送。
0	1	只选高字节库，访问奇地址字节；若从奇地址开始访问 16 位字，这是第一拍。
1	0	只选低字节库，访问偶地址字节；若从奇地址开始访问 16 位字，这是第二拍。
1	1	无效组合。

[第五章:p31–32]

做题提醒：8086 存储体题不要把“字节”和“字”混在一起：单字节访问一次完成，只是选高库还是低库不同；奇地址开始的 16 位字才需要两次总线周期。

6.4 Cache 基本流程与地址映射

- 主存速度通常慢于 CPU。8086/8088 在 5 MHz 时钟周期约 200 ns，8 MHz 时钟周期约 125 ns；若存储器存取时间跟不上，读写总线周期中要插入等待状态 T_W 。提高系统速度的两条路是提高器件速度和使用 Cache。 [第五章:p33–34]
- Cache 用小容量高速存储器保存主存中近期可能访问的一部分副本，利用程序顺序执行和数据顺序存放带来的局部性。Cache 对 CPU 透明：有没有 Cache，CPU 发出的仍是主存地址。 [第五章:p34–36]
- Cache 访问流程：CPU 发出主存地址 $\rightarrow$ 地址保存在 Cache 控制部件的主存地址寄存器 MA 中 $\rightarrow$ 用主

存页号/块号在地址变换部件或 CAM 中检索 → 命中则访问 Cache，不命中则访问主存，并把该信息所在的整块调入 Cache。 [第五章:p35-36]

- 设主存容量为 2^n ，Cache 容量为 2^m ，页/块大小为 2^p 。主存地址可分成“主存页号 $n-p$ 位 + 页内地址 p 位”；Cache 地址可分成“Cache 页号 $m-p$ 位 + 页内地址 p 位”。映射时只需决定主存页号放到哪个 Cache 页号。 [第五章:p37]

映射方式	规则	优缺点
全相联映射	主存任一页可装入 Cache 任一页；需保存主存高位页号到地址映射表，每次访问按主存页号检索。	冲突概率最小，任意两页只要 Cache 有空就可共存；但平均比较次数多，需高速且较大的映射表/CAM，硬件贵。
直接映射	主存页号 B 只能映射到固定 Cache 页号 b ： $b = B \bmod 2^{m-p}$ 其中 2^{m-p} 是 Cache 页数。	检索最快、实现最简单；但冲突概率最大，两个常用主存页若映射到同一 Cache 页，即使其他页空着也不能使用。
组相联映射	主存和 Cache 先分组；组间直接映射，组内全相联。主存地址包含区号、组号、组内页号、页内地址。	折中方案。比全相联检索快，比直接映射冲突少；若 Cache 分 a 组，每组 Y 页，则比较量约为 $1 + Y/2$ ，常用于实际系统。

[第五章:p38-42]

直接映射公式别写错：课件这一节用 2^m 表示 Cache 容量、 2^p 表示页大小，所以 Cache 页数是 2^{m-p} ，直接映射为 $b = B \bmod 2^{m-p}$ 。一般写法是“Cache 行号 = 主存块号 mod Cache 行数”。

6.5 Cache 替换、一致性与虚拟存储

替换算法	思想	注意点
随机替换	随机选择 Cache 中一页淘汰。	实现简单，但完全不利用程序访问历史，效果通常较差。
先进先出 FIFO	淘汰最早调入 Cache 的页。	实现简单，但最早调入的页可能正是频繁使用的页，可能刚换出又要换入。
最少使用 LFU	淘汰使用次数或使用频度最少的页。	符合“少用页未来可能也少用”的想法，但需要记录和比较频度，实现复杂。
最久未用 LRU	淘汰最长时间没有被访问的页。	用最近性估计未来访问，常用堆栈或近似堆栈思想实现，实际 Cache 中较常见。

[第五章:p43-47]

- Cache 一致性问题来自“Cache 保存的是主存一小部分副本”：CPU 可能只改了 Cache 而主存未改；外设或其他主控部件也可能改了主存而 Cache 副本未同步。 [第五章:p48]
- **写直达 (write-through)：** CPU 写 Cache 的同时写主存，优点是主存和 Cache 始终容易保持一致；缺点是每次写都要访问主存，速度下降、写流量大。 [第五章:p49]
- **写回 (write-back)：** CPU 写 Cache 时先不给主存更新，而是在 Cache 页上做修改标记；该页被替换时才整页写回主存，未修改页无需写回。优点是减少主存写次数，缺点是控制和一致性维护复杂。 [第五章:p49]
- 虚拟存储器用外存配合操作系统扩大程序可见的逻辑地址空间：暂不执行的指令段和暂不使用的数据先

放在外存，需要时再调入主存。

[第五章:p50]

做题提醒：Cache 题先问三件事：一是块/页大小决定页内地址位数 p ；二是 Cache 页数决定直接映射取模；三是未命中时是否需要替换，以及写操作采用写直达还是写回。

7 中断系统与 8259A

7.1 中断概念、请求保存与 8086 响应

- 中断（*interrupt*）是在 CPU 正常执行主程序时，由内部或外部事件迫使 CPU 暂停当前工作，转去执行相应服务程序，处理完再返回断点继续执行的过程。外部中断常来自键盘、鼠标等 I/O 设备，内部中断包括溢出、单步、除法错等。

[第六章:p2-3]
- 查询方式是 CPU 主动反复询问外设，硬件简单但效率低；中断方式是外设主动申请服务，CPU 被动响应，能提高 CPU 利用率，但外部控制电路更复杂。

[第六章:p2-3]
- 完整中断过程可分为**中断处理**和**中断服务**：前者包括仲裁、保护断点、查向量并转入服务程序，主要由硬件配合完成；后者是真正服务外设或事件的程序，由软件编写。

[第六章:p4-5]
- 中断与子程序调用都要保存返回位置，但子程序调用是事先安排的主动行为，通常服务主程序；中断是随机、被动的事件响应，通常服务外设或异常。

[第六章:p6]
- 每个中断源通常需要一个**中断请求触发器**（*interrupt request latch*）保存随机、短暂的请求，直到 CPU 响应后再清除；多个请求触发器组合起来就是请求寄存器 IRR 的思想。

[第六章:p7-8]
- 每个中断源还可配一个**中断屏蔽触发器**（*interrupt mask latch*），CPU 可逐位禁止或允许某一路请求；CPU 内部的 IF 是外部可屏蔽中断总开关，复位后 IF=0，要用 STI 开中断。

[第六章:p9-10]
- INTR 是受 IF 控制的可屏蔽中断，CPU 在当前指令最后一个 T 状态结束时检测；NMI 不受 IF 控制，固定 2 型、上升沿触发，通常用于更紧急事件。

[第六章:p11-14]
- NMI 请求不受 IF 标志影响，CPU 执行完当前指令后响应并转入 2 型中断入口。为防止干扰误触发，8086/8088 要求 NMI 高电平宽度至少保持两个 CPU 时钟周期；系统不用 NMI 时，应将该引脚接地。

[第六章:p13]
- 中断向量表位于低端内存，包含 256 个中断入口；每个入口 4 字节，低地址 2 字节保存 IP，高地址 2 字节保存 CS。类型号 n 的入口地址为 $4n$ 。

[第六章:p12]

8086 常见中断类型和指令		
类型/指令	名称	记忆点
INT n	软件中断	n 为 8 位中断向量号，范围 0-255；CPU 按向量号查中断向量表。
1 型	单步中断	TF=1 时，CPU 通常在每条指令后产生单步中断；课件按 INT 1 记。
2 型 / INT 2	不可屏蔽中断 NMI	NMI 硬件请求固定使用 2 型入口，不受 IF 屏蔽；软件 INT 2 也会查 2 型向量，但来源不同。
3 型 / INT3	断点中断	INT3 是单字节断点中断，常用于调试程序。
4 型 / INTO	溢出中断	INTO 只在 OF=1 时产生 4 型中断；OF=0 时不转移。直接写 INT 4 也会调用 4 型入口。

[第六章:p11]

做题提醒：课件 p11 把 INTO 和 4 型中断放在一起记：做题时核心是“OF=1 执行 INTO 才进 4 型”。除法错通常对应 0 型中断，不要把除法错当成 INTO 的触发条件。

8086/8088 INTR 响应流程：完成当前指令 → 发出第一个 $\overline{INTA}$ → 第二个 $\overline{INTA}$ 读中断类型号 → 自动压入 FLAGS、CS、IP 并清 IF/TF → 按类型号 n 到 $0000:4n$ 处取 IP、CS → 转入 ISR；ISR 中再由程序保护通用寄存器、执行服务、恢复现场，最后用 IRET 弹回 IP、CS、FLAGS。
[第六章:p12-15]

做题提醒：硬件自动保护的是 FLAGS、CS、IP；AX/BX/CX/DX、SI/DI、DS/ES 等是否保护，要看中断服务程序自己写没写 PUSH/POP。

7.2 中断源识别：查询式到向量式

- 8086/8088 只有一根外部可屏蔽中断申请引脚 INTR；当系统有多个中断源时，必须在 CPU 外部记录请求、比较优先级，并把当前最高优先级中断的类型号送给 CPU。
[第六章:p16]
- 多中断源系统还常要求两件事：CPU 正在服务某中断时，能屏蔽同级或较低级请求；若出现更高级请求，又能暂停当前服务而形成中断嵌套。若多个中断源等级差别不大，还希望能动态改变优先级。
[第六章:p16]

方法	工作过程	优缺点与考点
查询式	CPU 响应中断后，读入各请求触发器状态，再用 TEST、移位等程序判断是哪一路请求。	硬件简单,优先级可由软件测试顺序决定;中断源多时速度慢, CPU 仍要做大量判别。
向量中断式	中断控制硬件把中断类型号送上数据总线, CPU 直接按向量表找到 ISR 入口。	速度快, CPU 无需逐个查询来源;硬件和可编程控制逻辑更复杂。

[第六章:p16-21]

- 屏蔽法常见程序结构是读端口后依次 ‘TEST AL,80H/40H/...’, 先测谁谁优先；因此改变 TEST 顺序即可改变软件优先级。
[第六章:p19]
- 位移法用 RCL 等移位/带进位循环指令逐位检查请求状态，本质仍是查询式，判断工作由 CPU 完成。
[第六章:p20]
- 8259A 解决的是“多中断源如何连接、如何排优先级、如何给 CPU 中断向量号”的问题，因此它既是中断控制器，也是向量中断式的核心接口芯片。
[第六章:p21-22]

7.3 8259A 结构、连接和响应流程

- 一片 8259A 可接 8 路 IR0-IR7，支持 8 级优先级控制、逐路屏蔽和级联；一级级联最多可扩展到 64 路中断源。
[第六章:p22]
- 内部三个核心寄存器要分清：IRR 保存“谁提出请求”，IMR 保存“谁被屏蔽”，ISR 保存“谁正在服务”。普通题中某位为 1 的含义分别是请求中、屏蔽中、服务中。
[第六章:p25]
- 8259A 的空闲状态可理解为无有效请求或未被选中；工作状态则围绕 IR 输入、优先级仲裁、向 CPU 发 INT、响应 INTA、输出类型号和等待 EOI 展开。
[第六章:p23-26]

8259A 引脚信号按连接对象记忆

信号	连接对象	作用
D_0-D_7	CPU 数据总线	传送 ICW/OCW、状态字和中断类型号。
A_0	CPU 地址线	8259A 只有一根片内地址线，通常占两个端口地址。
$\overline{CS}$ 、 $\overline{RD}$ 、 $\overline{WR}$	CPU 控制线	片选、读状态/寄存器、写命令字。
$\overline{INT}$ 、 $\overline{INTA}$	CPU 中断线	INT 向 CPU 申请中断；CPU 用两个 INTA 周期响应并取类型号。
IR_0-IR_7	外设或从片	外部中断请求输入，普通全嵌套下 IR_0 最高、 IR_7 最低。
CAS_0-CAS_2	级联片	主片输出级联编码，从片接收，用来确定哪个从片给出向量。
SP/EN	级联/缓冲控制	非缓冲方式下区分主从；课件按 $SP=1$ 主片、 $SP=0$ 从片记忆。

[第六章:p23-24]



[第六章:p24]

8259A 一次响应流程：外设或从片置 $IR \rightarrow IRR$ 记录请求 $\rightarrow$ 优先级电路判决 $\rightarrow$ INT 请求 CPU $\rightarrow$ 第一个 $\overline{INTA}$ 锁定本次响应并封锁相关 $IRR \rightarrow$ 第二个 $\overline{INTA}$ 输出类型号 $\rightarrow$ 置 ISR 、清 $IRR \rightarrow$ 服务结束时自动 $AEOI$ 或由 CPU 写 EOI 清 ISR 。

[第六章:p26]

7.4 8259A 工作方式

类别	课件顺序中的方式	记忆点
优先权设置	普通全嵌套、特殊全嵌套、自动循环、特殊循环	普通全嵌套固定 IR_0 最高；特殊全嵌套用于级联时允许从片内更高优先级继续嵌套；循环方式适合各中断源等级差别不大。
结束中断	自动 EOI 、普通 EOI 、特殊 EOI	$AEOI$ 在第二个 $\overline{INTA}$ 后就清 ISR ，通常不适合需要嵌套的场合；普通 EOI 清当前最高服务位；特殊 EOI 指定清哪一位。
屏蔽方式	普通屏蔽、特殊屏蔽	IMR 某位为 1 屏蔽对应 IR ；特殊屏蔽用于软件动态改变嵌套关系。
触发方式	边沿触发、电平触发	边沿触发记录上升沿；电平触发靠高电平申请，要及时撤除请求电平。
连接方式	单片/级联、缓冲/非缓冲	$SNGL$ 、 BUF 、 M/S 、 SP/EN 与系统连接方式相关。

[第六章:p27-33]

做题提醒：判断能否嵌套时先看 IF 是否重新开中断，再看 8259A 的 ISR/优先级/屏蔽方式。AEOI 虽省命令，但 ISR 过早清除，不能简单等同于“服务真的结束”。

7.5 8259A 编程：ICW、OCW 与读状态

命令字	写入条件/时机	主要内容
ICW1	A0=0,D4=1;初始化第一个写入	IC4 决定是否写 ICW4; SNGL 决定单片/级联; LTIM 决定电平/边沿触发; D4 恒为 1。
ICW2	A0=1;第二个写入	设置中断类型号高 5 位, 低 3 位由 IR0–IR7 自动填入, 因此一片 8259A 形成连续 8 个类型号。
ICW3	A0=1,SNGL=0 时写入	主片用 S7–S0 标出哪些 IR 接从片; 从片用 ID2–ID0 说明自己接到主片哪一路 IR。
ICW4	A0=1,IC4=1 时写入	8086/8088 模式位必须为 1; 还设置 AEOI、缓冲方式、主从标志和特殊全嵌套。

[第六章:p34–40]

命令字	写入条件/端口	主要内容
OCW1	A0=1	IMR 屏蔽字, D0–D7=1 时屏蔽相应 IR。
OCW2	A0=0, D4D3=00	由 R、SL、EOI 控制优先级是否循环、循环方式和 EOI 类型, L2–L0 可指定对象。
OCW3	A0=0, D3=1	设置/清除特殊屏蔽模式, 或选择下一次读 IRR、ISR、查询字。

[第六章:p41–45]

读状态顺序：IMR 可直接按 A0=1 读写；若要读 IRR、ISR 或查询字，先向 A0=0 写 OCW3 选择对象，再执行读操作。查询字中 D7=1 表示有中断请求，低位给出中断源号。

[第六章:p45]

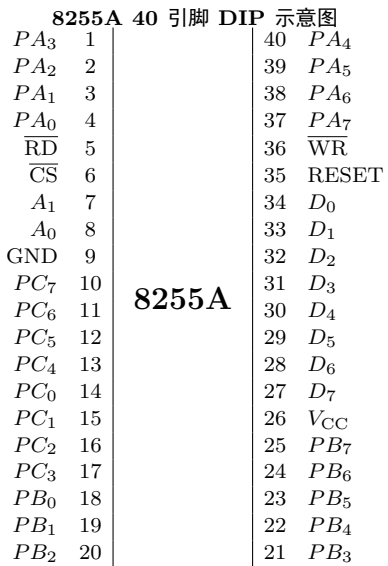
8 并行接口与 8255A

8.1 I/O 接口、端口与简单并行接口

- **I/O 接口** (*I/O interface*) 是 CPU 与外设之间协调数据形式、传递方式和速度差异的逻辑控制电路；它不是外设本身，而是 CPU 和外设之间的缓冲/控制层。[第七章:p3–4]
- 接口基本功能包括：I/O 地址译码和设备选择、数据输入输出、命令/状态管理、中断请求、信息锁存、格式转换、电平转换以及 A/D、D/A 等。[第七章:p5–6]
- **I/O 端口** (*I/O port*) 是接口中的寄存器。CPU 寻址的是端口地址，而不是笼统地寻址“某个外设”；常见端口分为数据端口、状态端口和控制端口。[第七章:p7–9]
- 并行接口一次传多位，速度快、距离近；串行接口逐位传输，线路少、距离远。并行数据传送仍需要状态线和控制线配合，如 M/IO、RD、WR。[第七章:p10–12]
- 简单并行输出常用锁存器 8282/74LS373；单向缓冲可用 74LS244；双向总线收发可用 8286/74LS245。锁存器解决“保持数据”，缓冲器主要解决“驱动能力/方向隔离”。[第七章:p13]

8.2 8255A 结构、端口与引脚

- 8255A 是通用可编程并行接口芯片，提供 PA、PB、PC 三个 8 位并行端口，40 引脚 DIP、单 5V 电源，通过一个方式控制字完成初始化。[第七章:p14–16]
- A 组由 PA 和 PC7–PC4 组成，B 组由 PB 和 PC3–PC0 组成；PC 口可作为普通 I/O，也可在方式 1/2 中被握手控制线占用。[第七章:p16–17]
- A_1A_0 选择四个内部端口：00 为 PA，01 为 PB，10 为 PC，11 为控制口。PC/XT 中 8255A 可映射成 60H–63H 这类连续端口地址。[第七章:p18]



[第七章:p15]

信号	说明
$D_0\text{--}D_7$	与 CPU 相连的双向数据总线，用于传送端口数据和控制字。
$\overline{CS}$ 、 $\overline{RD}$ 、 $\overline{WR}$	片选和读/写控制，低有效。未选中时 8255A 不响应总线。
A_1, A_0	内部端口选择线，决定访问 PA、PB、PC 还是控制口。
RESET	高电平复位；复位后所有内部寄存器清除，PA、PB、PC 均为输入方式。
PA/PB/PC	外设侧并行端口；PC 既可按高/低 4 位使用，也可作为握手和中断允许相关信号。

[第七章:p15–16]

8.3 初始化、方式控制字与 BSR

- 8255A 初始化就是向控制口写一个方式控制字，写入条件为 $A_1A_0 = 11$ 且 $D_7=1$ 。D6–D3 描述 A 组方式和方向，D2–D0 描述 B 组方式和方向。[第七章:p19–20]
- 方式控制字中，D6D5 选择 A 口方式：00 方式 0，01 方式 1，1x 方式 2；D2 选择 B 口方式：0 方式 0，1 方式 1。[第七章:p19–20]
- 当 $D_7=0$ 时，不再是方式设置，而是 PC 口位设置/复位 BSR 控制字：D3D2D1 选择 PC0–PC7，D0=1 置位，D0=0 复位。[第七章:p35–37]
- 端口 C 可直接按字节写 PC 端口，也可通过 BSR 单独改变某一位；在方式 1/2 中，部分 PC 位还用于置位/复位 INTE 等内部中断允许触发器。[第七章:p36–38]

位	方式控制字含义	取值
D7	类型选择	1: 方式设置; 0: PC 位操作 BSR
D6D5	A 组方式	00: 方式 0; 01: 方式 1; 1x: 方式 2
D4	PA 方向	1 输入, 0 输出
D3	PC 高 4 位方向	1 输入, 0 输出
D2	B 组方式	0: 方式 0; 1: 方式 1
D1	PB 方向	1 输入, 0 输出
D0	PC 低 4 位方向	1 输入, 0 输出

[第七章:p19–20; p35–37]

8.4 三种工作方式

方式	名称	关键特征
方式 0	基本输入/输出	无握手和中断逻辑; 输入不锁存, 保证实时读; 输出锁存, 保证写出的数据稳定; PC 可拆为两个 4 位口。
方式 1	选通单向 I/O	PA/PB 可作为带握手的单向输入或输出; 每个数据口用 PC 中 3 根线作 STB/IBF/OBF/ACK/INTR 等。
方式 2	双向选通 I/O	仅 PA 可工作于方式 2, 占用 PC 的 5 根握手线; B 组仍可选方式 0 或方式 1。

[第七章:p21–32]

- 方式 0 适合 LED、开关、继电器等简单设备; 若 CPU 与 I/O 不能及时读写, 方式 0 没有握手机制, 容易不适用。[第七章:p21–23]
- 方式 1 输入: 外设发低有效 STB, 把数据写入输入锁存器; 8255A 置 IBF=1 表示输入缓冲满; 若 INTE=1 且条件满足, INTR=1 向 CPU 请求中断; CPU 的 RD 后沿清 IBF。[第七章:p24–26]
- 方式 1 输出: CPU 写端口后, 低有效 OBF 通知外设 “输出缓冲满”; 外设用低有效 ACK 取数据并清 OBF; 必要时 INTR 通知 CPU 可继续输出。[第七章:p27–28]
- 方式 2 把方式 1 的选通输入和选通输出组合到 A 口。CPU 写入后数据先不直接出现在 PA 线上, 要等 ACK; 外设输入的数据也要等 CPU RD 才上系统数据总线, 用来避免总线冲突。[第七章:p30–32]

做题提醒: 8255A 题先问: 访问哪个端口? 写方式控制字还是 BSR? 端口是否被方式 1/2 征用? 被握手占用的 PC 位不能再当普通 I/O 位随便用。

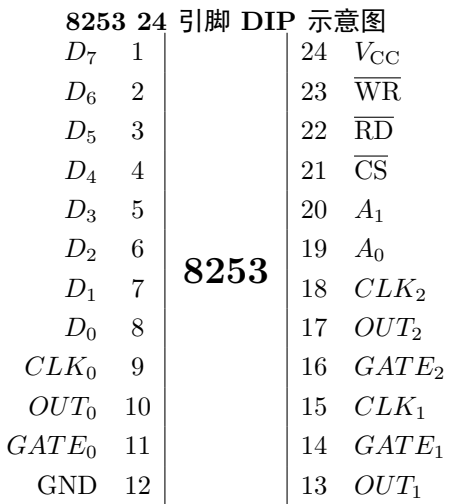
8.5 状态读取与典型应用

- 对未被方式 1/2 征用的 PC 引脚, 读到的是普通 I/O 信息; 对被征用为控制线的 PC 引脚, 读到的是反映 8255A 状态的状态字。[第七章:p38]
- LED 数码管接口是典型 8255A 应用: 一组端口可送段码, 一组端口可送位选或控制信号; 共阳极/共阴极决定段码有效电平。[第七章:p39–40]

9 定时/计数器 8253

9.1 定时/计数器概述与 8253 结构

- 定时器提供时间标准，每隔一定时间产生事件；计数器记录一段时间内发生的事件次数。两者本质上是类似电路，区别在于输入脉冲来源和工作目的。[第八章:p2]
- 定时方法包括软件延时、简单硬件定时、专用时钟芯片和可编程硬件定时。软件延时不需硬件但占 CPU 且不精确；可编程定时器灵活、精度高，是微机系统常用方法。[第八章:p3-6]
- 8253 含 3 个相互独立的 16 位减 1 计数器通道；每通道有预置寄存器、减 1 计数器和输出锁存器，可按二进制或 BCD 计数，有 6 种工作方式。[第八章:p7-9]



[第八章:p7]

信号	说明
D_0-D_7	CPU 侧 8 位双向数据总线，用于写控制字、写初值和读计数值。
A_1, A_0	00/01/10 选计数器 0/1/2，11 选控制字寄存器。
$\overline{CS}$ 、 $\overline{RD}$ 、 $\overline{WR}$	片选和读写控制，低有效。
CLK_0-CLK_2	三个计数器各自的计数时钟，课件按 CLK 下降沿减 1。
$GATE_0-GATE_2$	门控/触发输入，不同方式下可能是电平允许或上升沿触发。
OUT_0-OUT_2	输出定时结束信号、频率发生器脉冲、方波或选通信号。

[第八章:p7-11]

9.2 六种工作方式：按课件顺序掌握

方式	名称	输出与触发特征
0	计数结束中断	软件触发一次计数。写方式后 OUT 变低，写初值且 GATE=1 后开始计数，减到 0 时 OUT 变高，可作为中断请求。
1	硬件触发一次计数	GATE 上升沿装入初值并启动，计数期间 OUT 为低，结束后 OUT 变高；可重触发，常作 watchdog。
2	频率发生器	连续分频，OUT 大部分时间为高，计数到 1 时输出一个 CLK 宽度低脉冲并自动重装初值。
3	方波发生器	连续输出近似方波。偶数初值高低各一半；奇数初值高电平比低电平多一个时钟。
4	软件触发选通	类似方式 0，但 OUT 初始为高，计数结束输出一个 CLK 宽度低脉冲后回高。
5	硬件触发选通	类似方式 1，由 GATE 上升沿启动，计数结束输出一个 CLK 宽度低脉冲，可重触发。

[第八章:p12-34]

- 课件把方式 0/4 归为软件触发的一次性计数，方式 1/5 归为 GATE 上升沿硬件触发的一次性计数，方式 2/3 归为连续计数。[第八章:p13; p34]
- 方式 0 中，GATE=0 暂停计数，恢复为高后继续；方式 1 中，触发后 GATE 电平不影响当前计数，但新的上升沿会重新装入初值并重触发。[第八章:p18-24]
- 方式 2 中，计数过程中写入新初值不影响当前周期，从下一个周期开始使用新值；GATE 拉低使 OUT 保持高，恢复高电平后重新装入初值。[第八章:p25-28]
- 方式 3 进行减 2 为主的计数：奇数初值时高电平持续时间比低电平多一个 CLK，因此占空比略大于 0.5。[第八章:p29-31]

$f_{out} = \frac{f_{clk}}{N}$

$N = \frac{f_{clk}}{f_{out}}$

$t = \frac{N}{f_{clk}}$

[第八章:p28; p31; p35-39]

做题提醒：频率公式主要用于方式 2/3 这类周期输出；不要把方式 0/1/4/5 的一次性定时也机械写成“输出频率”。

9.3 8253 编程与读写计数值

- 编程顺序：写方式控制字 → 写计数初值 → 对方式 1/5 等待 GATE 上升沿，对其他方式按门控条件进入计数 → 下一个 CLK 下降沿把初值装入减 1 计数器。[第八章:p35-39]
- 8253 数据线为 8 位，计数器为 16 位，所以读写格式可选只低字节、只高字节或先低后高。写入 0 时，二进制代表 65536，BCD 代表 10000。[第八章:p37-38]
- 运行中直接分两次读 16 位计数值可能前后不一致；应先用锁存命令锁定当前计数值，再按设定顺序读取低/高字节。[第八章:p35-38]

位	含义	常用编码
D7D6	计数器选择	00/01/10 对应计数器 0/1/2；11 为锁存/特殊选择
D5D4	读写格式	01 低字节；10 高字节；11 先低后高；00 锁存当前值
D3D2D1	工作方式	000–101 对应方式 0–5，方式 2/3 有兼容编码
D0	数制	0 二进制，1 BCD

[第八章:p36–38]

10 串行通信与 8251A

10.1 串行通信技术概述

- 串行通信用少量线路逐位传输，速度通常低于并行通信，但距离远、成本低，适合系统之间的信息交换。
[第九章:p2–3]
- 串口必须约定：**波特率** (*baud rate*)、位同步、帧同步、帧格式、校验方式和错误处理。接收端采样时钟必须与发送端速率和相位保持在允许范围内。
[第九章:p4–6]
- 异步传输以字符为单位：空闲为 1，先发低电平起始位，再发 5–8 位数据、可选校验位和 1/1.5/2 位停止位。每字符重新同步，允许双方用各自局部时钟，但效率较低。
[第九章:p7–10]
- 同步传输以连续二进制流为单位，可一帧多个字符；内同步靠同步字符，外同步靠公共时钟线。同步方式效率高，但对时钟和硬件要求更高。
[第九章:p11–13]

$$\eta_{\text{async}} = \frac{n}{1 + n + p + s}$$

[第九章:p8–10]

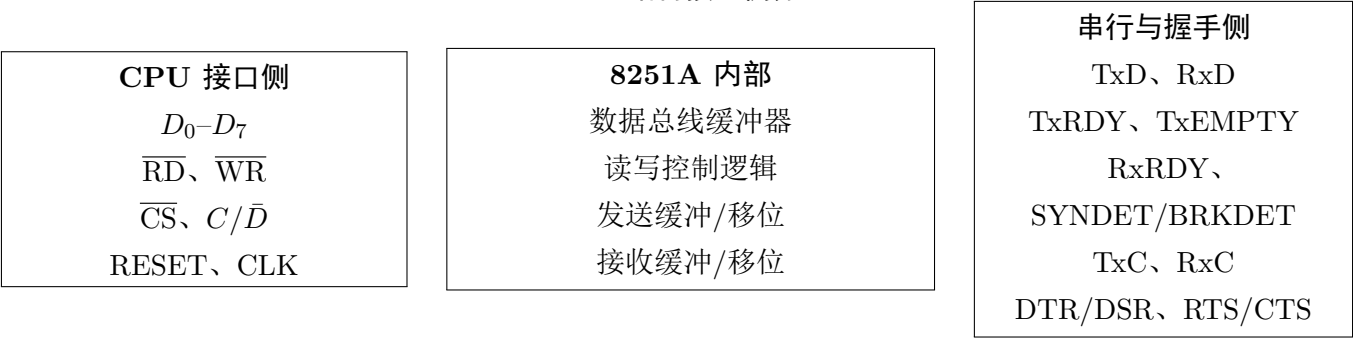
概念	复习要点
编码	NRZ、NRZI、曼彻斯特、差分曼彻斯特等编码把数据和同步信息结合；曼彻斯特类依靠位中间跳变帮助同步。
传输制式	单工只单向；半双工双向交替；全双工双向同时。
奇偶校验	能检出奇数个位错误，不能定位，也不能保证发现偶数个位错误。
纵横校验	在字符奇偶基础上增加列校验，有时可纠正一位错。
CRC	用生成多项式产生帧检验序列，接收端余数为 0 认为无差错，检错能力强。

[第九章:p14–21]

10.2 8251A 功能、结构和引脚

- 8251A 是可编程同步/异步串行接口，可设单工、半双工、全双工，可设 5–8 位字符、校验方式、同步方式、停止位和波特率系数。
[第九章:p22]
- 发送方向完成并-串转换并自动生成校验位；接收方向完成串-并转换并检查错误。8251A 只完成格式转换和收发控制，远距离通信还要外接 TTL/EIA 或差分驱动转换电路。
[第九章:p23]
- 发送和接收均为双缓冲。TxRDY=1 表示发送数据寄存器空、CPU 可写下下一个字符；TxEMPTY=1 才表示发送移位寄存器也空、串行线真正发完。
[第九章:p25–26]

8251A 结构按三侧分组



[第九章:p24-29]

信号/结构	分组	说明
D_0-D_7	CPU 数据线	CPU 通过它写发送数据、读接收数据、写模式/命令字和读状态字。
$C/\overline{D}$	端口选择	$C/\overline{D} = 0$ 访问数据口； $C/\overline{D} = 1$ 写模式字/命令字或读状态字。
CLK	内部时钟	不是收发波特率时钟；课件强调应高于设置波特率 30 倍以上。
TxD/RxD	串行数据线	TxD 输出串行数据，RxD 输入串行数据。
TxRDY/ RxRDY	就绪/中断请求	TxRDY 表示发送寄存器空；RxRDY 表示接收寄存器满，常接中断请求逻辑。
TxEMPTY	发送完成	发送移位寄存器空，判断最后一个字符是否真正发完要看它。
TxC/RxC	收发时钟	同步方式要求频率和相位一致；异步方式可选 $\times 1/ \times 16/ \times 64$ 过采样。
SYNDET/ BRKDET	同步/断缺	同步方式指示同步字符检测；异步方式检测 Rx 线长时间为低的 break 状态。
DTR/DSR	MODEM 握手	DTR 告知本端准备好，DSR 表示对方准备好。
RTS/CTS	MODEM 握手	RTS 请求发送，CTS 允许发送。

[第九章:p26-29]

10.3 8251A 编程与状态判断

- 8251A 只有一个控制口 $C/\overline{D} = 1$ ，复位后首先等待写工作模式字；写完模式字后进入写操作命令字状态，之后通常继续写命令字。命令字 IR=1 可内部复位，回到等待模式字状态。

[第九章:p30]
- 同步方式初始化格式为：模式字 → 1 个或 2 个同步字符 → 操作命令字 → 数据块；异步方式没有同步字符这一步。

[第九章:p33-36]

模 式 字 位	字 段	编 码
D1D0	模式/波特率系数	00 同步；01 异步 ×1；10 异步 ×16；11 异步 ×64
D3D2	字符长度	00/01/10/11 对应 5/6/7/8 位
D4	校验允许 PEN	1 有校验，0 无校验
D5	偶校验 EP	PEN=1 时，0 奇校验，1 偶校验
D7D6	异步停止位	01/10/11 对应 1/1.5/2 位，00 在异步下无效
D7D6	同步设置	决定内/外同步和单/双同步字符

[第九章:p31]

位	操作命令字	置 1 作用
D7	EH	同步方式进入搜索模式
D6	IR	内部复位，返回等待模式字状态
D5	RTS	请求发送握手输出有效
D4	ER	清除 PE、OE、FE 等错误标志
D3	SBRK	强制 TxD=0，发送断缺状态
D2	RxE	允许接收
D1	DTR	数据终端准备好输出有效
D0	TxEN	允许发送

[第九章:p32]

状态位	名称	为 1 时的含义
D7	DSR	外部数据装置准备好
D6	SYNDET/BRKDET	同步已找到/异步检测到断缺
D5	FE	帧错误，异步字符未检测到停止位
D4	OE	溢出错误，新数据覆盖尚未读取的数据
D3	PE	奇偶校验错误
D2	TxEMPTY	发送移位寄存器空，全部发送完成
D1	RxRDY	接收缓冲器满，可读取
D0	TxRDY	发送缓冲器空，可写入下一字符

[第九章:p34]

做题提醒：8251A 最常考三分法：模式字决定帧格式和同步/异步；命令字决定允许发送接收、清错和握手；状态字用于查询 TxRDY、RxRDY、TxEMPTY 和错误位。

10.4 常见串行接口标准

标准	电气方式	常见制式	特点
RS-232C	单端 EIA 负逻辑	三线全双工	TTL 需经 1488/1489 或 MAX232 转换；距离短，抗干扰较弱。
RS-422A	差分发送/接收	四线全双工	点对点，全双工，距离远，抗干扰强；节点多时通常点对点分别连接。
RS-485	差分总线	两线半双工为主	多节点总线，同一时刻只能一个发送器有效；DE/RE 控制收发方向。

[第九章:p37-44]

- TTL 电平适合板内短距离；RS-232C 使用 EIA 负逻辑，正电压表示 0，负电压表示 1，需电平转换。[第九章:p23； p37-39]
- RS-485 总线中所有节点初始通常处于接收状态；任一节点发送时切到发送状态，若两个以上节点同时发送会冲突。[第九章:p41-43]
- RS-485 也可用四线构成主从式全双工：主机单独发给从机，从机回传只给主机，通信关系不再是任意节点互发。[第九章:p44]

11 D/A 与 A/D 转换

11.1 转换概念与 DAC 原理

- D/A 转换把 CPU 输出的有限数字量变成模拟电量，用来控制温度、压力、流量、速度等物理对象；A/D 转换把连续模拟量变成 CPU 能处理的有限数字码。[第十章:p1-3]
- A/D 是“连续变离散、无限变有限”，因此必然有量化；D/A 是“用有限级近似连续量”，输出不是数学上完全连续的任意值。[第十章:p2-3]
- R-2R 梯形网络只需两种阻值 R 和 2R，便于提高匹配精度；不同数字位控制不同权重电流，经运算放大器完成电流-电压变换。[第十章:p4]

11.2 DAC0832：结构、时序与输出公式

- DAC0832 是 8 位电流输出型 DAC，转换时间约 1μs，内部含输入寄存器和 DAC 寄存器，支持双缓冲、单缓冲和直通工作。[第十章:p5-7]
- DI_7-DI_0 接数字输入；ILE 为输入锁存允许，高有效； $\overline{CS}$ 、 $\overline{WR1}$ 控制写输入寄存器； $\overline{XFER}$ 、 $\overline{WR2}$ 控制写 DAC 寄存器。[第十章:p5-8]
- $I_{OUT1} + I_{OUT2}$ 为常数， I_{OUT1} 与 DAC 寄存器数字码成正比； V_{REF} 为梯形网络参考电压， R_{FB} 可直接接外部运放反馈。[第十章:p6]

$$D = \sum_{k=0}^7 b_k 2^k, \quad I_{OUT1} \approx I_{FS} \frac{D}{256}, \quad V_{out} \approx -V_{REF} \frac{D}{256}$$

[第十章:p4-6]

- 双缓冲方式先逐片写输入寄存器，再用公共 $\overline{XFER}/\overline{WR2}$ 同时更新 DAC 寄存器，适合多路 DAC 同步输出。[第十章:p8-9]

- 单缓冲方式可让一级寄存器长期有效，或把两级控制信号合并，使 CPU 一次写操作就更新输出；连接方式可以多样，但必须满足控制逻辑。 [第十章:p10]

做题提醒： DAC0832 没有 EOC 等“转换完成”查询步骤。检查程序时按“写输入寄存器 → 写 DAC 寄存器/更新输出”看时序。

11.3 A/D 转换原理、量化与逐次逼近

- 对 n 位 ADC，参考范围为 $V_{REF+} - V_{REF-}$ 时，1 LSB 或量化间隔为

$$\Delta V = \frac{V_{REF+} - V_{REF-}}{2^n}.$$

[第十章:p11]

- 按课件截断规则，输入电压 u_i 对应数字量可写为

$$D = \left\lfloor \frac{u_i - V_{REF-}}{V_{REF+} - V_{REF-}} 2^n \right\rfloor, \quad 0 \leq D \leq 2^n - 1.$$

例如 0~5V、8 位、 $u_i = 3.5V$ ， $D = \lfloor 3.5/5 \times 256 \rfloor = 179 = 10110011_B$ 。

[第十章:p11]

- 常见 A/D 原理包括计数器式、逐次逼近式、双积分式和并行式；本课件重点展开逐次逼近式和 ADC0809。 [第十章:p11-12]
- 逐次逼近 ADC 从最高位开始试置 1，用内部 DAC 输出与输入模拟量比较；DAC 输出低于输入则保留该位，否则清 0，逐位比较到最低位。 n 位 ADC 只需 n 次比较，速度较快且转换时间固定。 [第十章:p12]

11.4 ADC0809：通道选择、转换时序与读数

- ADC0809 是 8 位逐次逼近 ADC，含 8 路模拟输入通道和内部 DAC，典型转换时间约 $100 \mu s$ 。 [第十章:p13]
- IN0~IN7 接模拟输入；ADDA/ADDB/ADDC 选择通道；ALE 高有效锁存通道地址；START 启动转换；EOC 为状态输出，低电平表示转换中，高电平表示完成；OE 高有效时把结果送到 D_7-D_0 。 [第十章:p13-15]
- START 上升沿清逐次逼近寄存器，下降沿真正启动 A/D 转换；EOC 可接中断请求，也可由 CPU 查询。 [第十章:p15-16]
- ADC0809 可接单通道或多通道使用。单通道应用中，通道选择线可固定；多通道采集时，每次转换前都要先放通道号并用 ALE 锁存。 [第十章:p16]

ADC0809 读数流程： ADDA/B/C 放通道号 → ALE 锁存通道 → START 给启动脉冲，下降沿开始转换 → 等 EOC=1 → OE=1 打开三态输出 → CPU 读 D_7-D_0 。

[第十章:p13-16]

做题提醒： ADC0809 的四个控制/状态信号不要混：ALE 选通道，START 启动转换，EOC 报告完成，OE 只负责把已经完成的数字结果送上数据总线。

12 综合做题模板

12.1 标志位与条件转移

1. 先把操作数按指定位宽写成二进制或十六进制，确认是字节还是字。

2. 算结果并截断到指定位宽：由截断进位/借位判断 CF，由结果是否为 0 判断 ZF，由最高位判断 SF。
3. 有符号溢出用“同号相加得异号”或“异号相减得异于被减数符号”判断 OF。
4. 最后按无符号或有符号关系选择条件跳转，不能混用 CF 与 SF/OF。 [第二章:p22-27；第三章:p55-75；p111-117]

12.2 存储器扩展与地址范围

1. 用目标容量除以单片容量，分别判断位宽扩展倍数和字数扩展倍数。
2. 用低位地址线连接片内地址；8086 还需考虑偶/奇存储体及 A_0 、 $\overline{BHE}$ 。
3. 用剩余高位地址译码产生片选，写出每组高位固定模式。
4. 把最低地址设为片内地址全 0，最高地址设为片内地址全 1，检查连续性、重叠和镜像。 [第二章:p57-61；第五章:p16-32]

12.3 接口芯片控制字与转换时序

1. 先读题目中的硬件连接，确定端口地址和所用计数器/端口组。
2. 把每一位控制字写成字段表，不凭印象直接凑十六进制。
3. 8259A 区分初始化 ICW 与运行控制 OCW；8255A 区分 D7=1 方式控制与 D7=0 BSR；8253 区分计数器选择、读写格式、方式和数制；8251A 区分模式字、命令字和状态字。
4. 写完后反向解释每一位，确认方向、触发、屏蔽和初值写入顺序。涉及 ADC0809 时，按 ALE 锁存通道 → START 启动 → 等待 EOC 完成 → OE 读数检查；涉及 DAC0832 时，按写输入寄存器 → 写 DAC 寄存器/更新输出检查。 [第六章:p34-45；第七章:p19-20；p34-38；第八章:p35-39；第九章:p30-34；第十章:p7-16]

13 考前一页速记

必须记住	结论
物理地址	段值左移 4 位加偏移；每段最大 64KB
8086 存储体	偶地址开始的 16 位字一次传完；奇地址开始的 16 位字两次传完；单字节访问一次完成
栈	向低地址增长；PUSH 先 SP-2，POP 后 SP+2
默认段	BP 用 SS；BX/SI/DI 通常用 DS；串目的固定 ES:DI
无符号比较	CF/ZF；JB 看 CF=1，JA 看 CF=0 且 ZF=0
有符号比较	SF 与 OF；JL 看 SF≠OF，JG 看 SF=OF 且 ZF=0
串操作	常用 DS:SI 作源、ES:DI 作目的；MOVS/CMPS 同时用二者，LODS 只用源，STOS/SCAS 只用目的；CX 计数，DF 决定方向
中断向量	256 个，每个 4 字节；类型号 n 的表项在 $4n$
8259A	IRR 请求、IMR 屏蔽、ISR 服务；OCW1 中 1 表示屏蔽
8255A	D7=1 写方式，D7=0 控制 PC 单个位；复位后全输入
8253	三个 16 位计数器；方式 2/3 常用于周期波形，近似按 $f_{\text{out}} = f_{\text{clk}}/N$ 计算
8251A	模式字先写；同步时再写同步字符；TxEMPTY 才表示全部发送完
DAC0832	双缓冲分两级写入，可实现多路同步更新
ADC0809	ALE 锁存通道；START 下降沿开始转换；EOC=1 表示完成；OE 只打开三态输出，不负责启动