

《微型计算机原理与接口技术》习题与思考答案

第 1 章 微型计算机概论

- 1.1 ABC 1.2 B 1.3 B 1.4 C 1.5 ABC 1.6 BD
- 1.7 00000111B=7D=07H 11010100B=212D=D4H
 01101010B=106D=6AH 10110.101B=22.625D=16.AH
 11001.011B=25.375D=19.6H
- 1.8 127D=11111111B=FFH 12.625D=1100.101B=C.AH
 225.9375D=11100001.1111B=E1.FH 18.3125D=10010.0101B=12.5H
 206.125=11001110.001B=CE.2H
- 1.9 10H=10000B=16D 0.A8H=0.10101B=0.65625D
 28.9H=101000.1001B=40.5625D 4B.2AH=1001011.0010101B=75.6762D
 20E.4H=1000001110.01B=526.25D
- 1.10 $[+37]_{\text{原}}=00100101\text{B}$ $[+37]_{\text{反}}=00100101\text{B}$ $[+37]_{\text{补}}=00100101\text{B}$ $[+37]_{\text{过余}}=10100101\text{B}$
 $[+94]_{\text{原}}=01011110\text{B}$ $[+94]_{\text{反}}=01011110\text{B}$ $[+94]_{\text{补}}=01011110\text{B}$ $[+94]_{\text{过余}}=11011110\text{B}$
 $[-11]_{\text{原}}=10001011\text{B}$ $[-11]_{\text{反}}=11110100\text{B}$ $[-11]_{\text{补}}=11110101\text{B}$ $[-11]_{\text{过余}}=01110101\text{B}$
 $[-125]_{\text{原}}=11111101\text{B}$ $[-125]_{\text{反}}=10000010\text{B}$ $[-125]_{\text{补}}=10000011\text{B}$ $[-125]_{\text{过余}}=00000011\text{B}$
- 1.11 补码 00010101B 的真值为+21D 补码 41H 的真值为+65D
 补码 9BH 的真值为-101D 补码 FFH 的真值为-1D
 补码 11110101B 的真值为-11D
- 1.12 A 的 ASCII 码为 41H a 的 ASCII 码为 61H
 g 的 ASCII 码为 67H z 的 ASCII 码为 7AH
 0 的 ASCII 码为 30H 9 的 ASCII 码为 39H
 * 的 ASCII 码为 2AH + 的 ASCII 码为 2BH
 CR 的 ASCII 码为 0DH % 的 ASCII 码为 25H
- 1.12 一个 16×16 字形点阵占用存储空间 32B
 一个 24×24 字形点阵占用存储空间 72B
 一个 32×32 字形点阵占用存储空间 128B
- 1.14 (以 8 位补码为例)
 $[X]_{\text{补}}+[Y]_{\text{补}}=[+38]_{\text{补}}+[+100]_{\text{补}}=0001010\text{B}$, 溢出
 $[X]_{\text{补}}+[Z]_{\text{补}}=[+38]_{\text{补}}+[-20]_{\text{补}}=00010010\text{B}$, 未溢出
 $[Y]_{\text{补}}-[Z]_{\text{补}}=[+100]_{\text{补}}-[-20]_{\text{补}}=01111000\text{B}$, 未溢出
 $[Z]_{\text{补}}-[X]_{\text{补}}=[-20]_{\text{补}}-[+38]_{\text{补}}=11000110\text{B}$, 未溢出
- 1.15 X 与 Y=0100B X 或 Z=1111B Y 异或 Z=1101B 非 Y=1001B
- 1.16 微型计算机具有体积小、重量轻、功耗低；功能强；可靠性高；价格低廉；结构灵活、适应性强；使用方便、维护容易等特点。
- 1.17 (略)
- 1.18 字长、内存容量、主频、运算速度、指令系统、可靠性、兼容性、性能价格比等。
- 1.19 微型计算机系统的组成 (参考图1-3)

第 2 章 Intel 80x86 微处理器

2.1 B 2.2 B 2.3 C 2.4 B 2.5 B 2.6 A 2.7 A

2.8 8086 微处理器的指令队列按照“先进先出”的原则进行指令的存取操作，可预存 6 个字节的指令代码，并供执行部件 EU 取指并执指，从而为 EU 与 BIU 并行操作提供支持。

2.9 超标量流水线结构是指微处理器内含有多个指令执行部件、多条指令执行流水线的结构。

2.10 8086 最大模式和最小模式的主要区别是：最小模式为单处理机模式，所有的总线控制信号都由 CPU 直接产生，这种系统中的总线控制逻辑电路少，控制信号较少，一般不必接总线控制器。最大模式为多处理机模式，系统中包括两个或两个以上处理器，其中一个 8086 作主处理器，其他处理器为协处理器，最大模式下的控制信号较多，需要通过总线控制器 8288 与总线相连，控制总线驱动能力较强。

通过 $\overline{\text{MN}}/\overline{\text{MX}}$ 引脚进行最大模式或最小模式的选择，当 $\overline{\text{MN}}/\overline{\text{MX}}$ 接入 +5V 电源时 8086 系统工作于最小模式，当 $\overline{\text{MN}}/\overline{\text{MX}}$ 接地时系统处于最大模式状态。

2.11 8086 微处理器从功能上分为总线接口部件 BIU 和执行部件 EU 两部分。

BIU 的主要功能是逻辑地址到物理地址的转换、指令预取、根据 EU 所执行的当前指令要求进行存储器操作数的读/写。EU 的主要功能是：从 BIU 的指令队列中取来指令、对指令译码、执行，并产生相应的控制信号，另外控制 ALU 进行数据运算，向 BIU 提供访存的有效地址。

2.12 OF、SF、ZF、AF、PF、CF、DF、IF、TF

溢出时 OF 置位、结果为负时 SF 置位、结果为零时 ZF 置位、低四位向高位有进位时 AF 置位、结果的低 8 位中 1 的个数为偶时 PF 置位、结果的最高位向上有进位或有借位时 CF 置位。

需要按递减顺序对字符串操作时 DF 置位、允许 CPU 接受 INTR 中断请求时 IF 置位、令 CPU 进行单步工作时 TF 置位。

2.13 执行一条指令所需的时间称为指令周期，CPU 访问内存或 I/O 端口存/取一个数据或指令所用的时间为总线周期，时钟脉冲的重复周期称为时钟周期。一个最基本的总线周期由 4 个时钟周期组成，一个指令周期由一个或多个总线周期构成。

2.14 当系统中存储器或 I/O 外设的速度较慢时，需要在 T3 状态之后插入 Tw。

2.15 一个存储器写的总线操作过程及各引脚的变化情况是：从 T1 状态开始，20 位物理地址出现在总线上，同时 $\overline{\text{BHE}}$ 输出有效的低电平，ALE 输出一个正脉冲，在 ALE 的下降沿对分时复用的地址进行锁存，并一直保持到 T4 状态，控制数据收发器的数据传输方向引脚 $\text{DT}/\overline{\text{R}}$ 变为高电平，并一直保持到 T4 状态。进入 T2 状态，地址信息消失，CPU 向 AD15~AD0 发出数据，数据信息一直保持到 T4 状态，A19/S6~A16/S3 及 $\overline{\text{BHE}}/\text{S7}$ 线上输出状态信息，并一直持续到 T4 状态，数据允许信号 $\overline{\text{DEN}}$ 降为低电平，允许数据收发器工作，读控制信号 $\overline{\text{WR}}$ 降为有效的低电平，使被选中的存储单元数据送上数据总线。进入 T3 状态后，CPU 采样 READY 引脚，若 READY 为高电平，则无需插入 Tw，直接进入 T4 状态，否则就在 T3 之后自动插入 Tw，直到 READY 上升为高电平后进入 T4 状态。进入 T4 状态后，CPU 认为存储器已经完成了数据的写入，结束本次总线读操作， $\overline{\text{WR}}$ 、 $\text{DT}/\overline{\text{R}}$ 、 $\overline{\text{DEN}}$ 等信号失效，所有三态总线变为高阻状态，为下一个总线周期作准备。

2.16 因为 8086 微处理器的地址总线与数据总线是分时复用的，即不在同一时钟周期内使用，所以可以利用同一引脚既作地址线又作数据线，分时具有不同的功能。例如，AD0 引脚有时用作传输数据的数据总线 D0，有时又作为输出地址信号的地址总线 A0。

2.17 由于 8086 CPU 内部寄存器是 16 位的，只能寻址 64KB 空间，而内存储器容量为 1MB，

CPU 访存需要 20 位物理地址,所以 8086 存储系统把整个存储空间分成若干逻辑段,在 CPU 需要访存时,由指令给出逻辑地址(包括一个 16 位的段基址和一个 16 位的偏移地址),通过 CPU 内部的地址形成部件将逻辑地址转换为物理地址。

8086 系统内存储器的分段规则是:段起始地址必须能被 16 整除,即段起始单元物理地址的低四位为 0000;每个段的容量不超过 64KB。

2.18 8086 系统中物理地址的形成是由 CPU 内部的地址形成部件完成的:物理地址=段基址×16+偏移地址。

首单元物理地址:203A0H。尾单元物理地址:203BDH

2.19 输入操作

2.20 除 CS 为 FFFFH 外,其余各寄存器均为 0,指令队列空

2.21 80386 的内部主要包括:总线接口部件 BIU、指令预取部件 IPU、指令译码部件 IDU、执行部件 EU、存储器管理部件 MMU。

BIU 用于在 CPU 访问存储器或 I/O 端口时产生必须的地址、数据和控制信号。IPU 实现指令代码的预取。IDU 负责从指令队列中获得指令并进行译码。EU 完成指令所规定的操作。MMU 负责将逻辑地址转换为物理地址。

2.22 80486 较 80386 增加了高性能浮点运算部件(FPU)和高速缓冲存储器(Cache)

(1) 浮点运算部件 FPU。浮点运算部件专门用来完成一些超越函数和复杂的实数运算,它以极高的速度进行单精度或倍精度的浮点运算。浮点运算部件在 80486 芯片内部集成,且可以与高速缓存直接交换数据,有效地提高了微处理器的浮点运算能力和速度,它与 80387 所执行的是同一个指令系统,保持了同 80387 的兼容性,但其浮点处理性能却是 80387 的 2.8 倍。

(2) 高速缓冲存储器 Cache。80486 的片内配有一个 8KB 的高速缓冲存储器 Cache,它用于存放 CPU 最近要使用的数据和指令。它采用 4 路组相联的结构,每路有 128 个高速缓存行,每行可存放 16 个字节(即 128 位)的信息。这个片内 Cache 既可存放数据,又可存放指令,它比片外 Cache 进一步加快了 CPU 访问内存的速度,并减轻了系统总线的负载。

2.23 Pentium 微处理器的数据 Cache 和指令 Cache 分别实现数据预取和指令预取的功能。

2.24 Pentium 系列微处理器的主要特点是:

1) 采用超标量流水线结构,从而使 CPU 的运行速度成倍提高。

2) 采用双高速缓冲存储器结构,将指令 Cache 和数据 Cache 分离,使用它们同时分别预取指令和操作数。

3) 将常用指令进行固化,从而进一步提高指令的执行速度。

4) 采用全新设计的增强型浮点运算器(FPU),使得浮点运算速度大大提高。

5) CPU 内部采用指令预取和分支预测技术,从而大大提高了流水线的执行效率。

6) 系统可选择工作于实模式、保护模式、虚拟 8086 模式或系统管理模式。

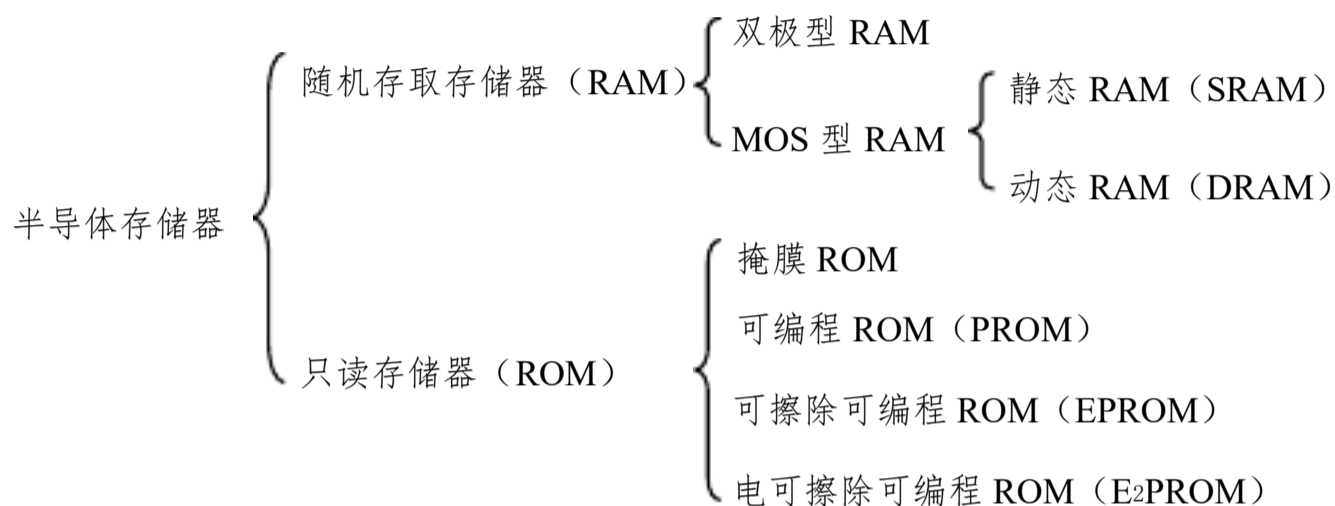
7) 系统使用 64 位的外部数据总线,提高了数据传输速度;采用 PCI 局部总线;系统内部还增强了错误检测与报告、支持多重处理等功能。

第 3 章 半导体存储器及其接口

- 3.1 C 3.2 A 3.3 D 3.4 C 3.5 C
3.6

	A	B	C	D	E
地址线:	19	10	11	14	16
数据线:	4	8	1	8	1

- 3.7 A. 8片 B. 8片 C. 128片
3.8 (1) 64K (2) 8片 (3) 2
3.9 存储容量、存取速度
3.10



- 3.11

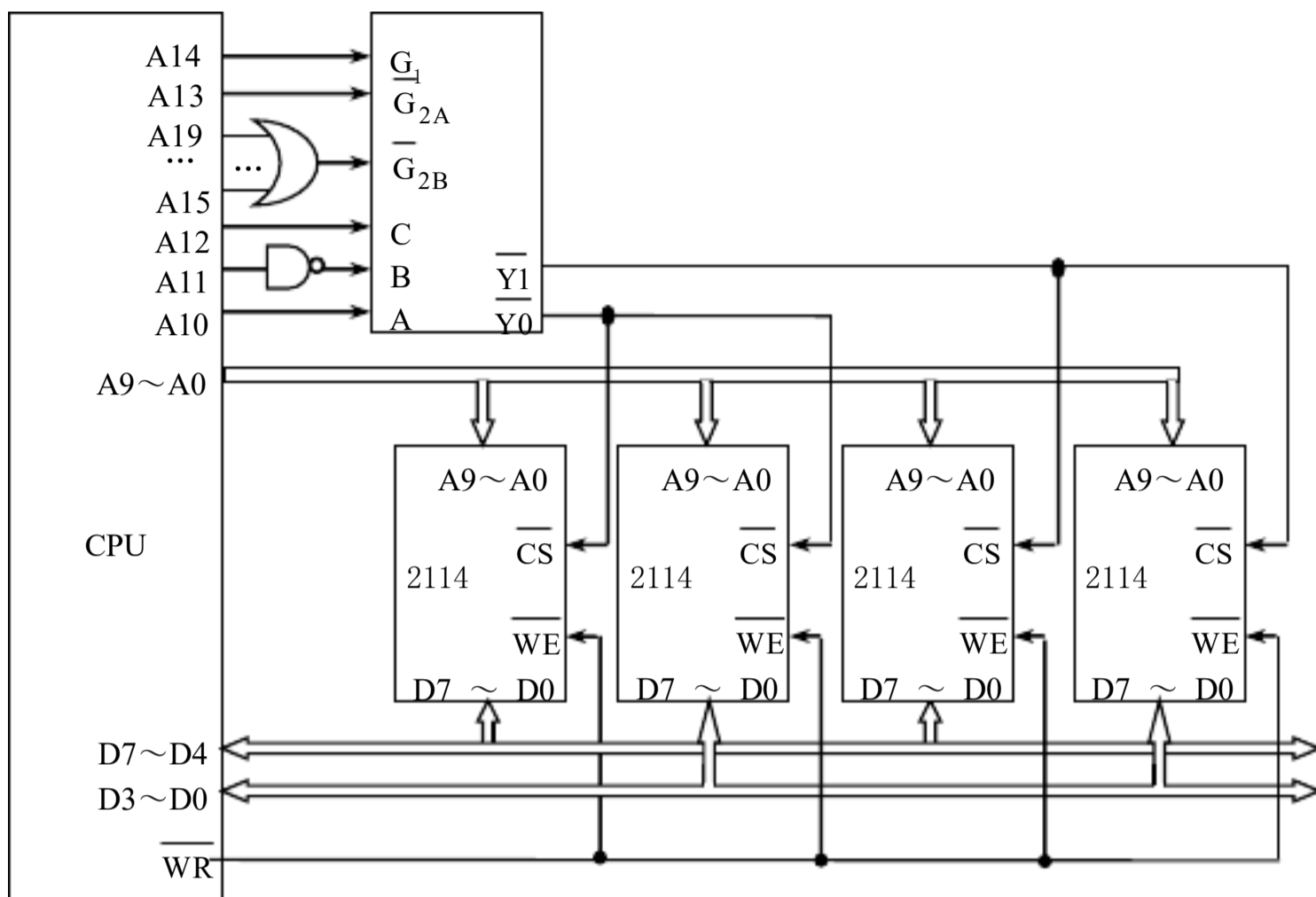
存储器	读写特点	一般应用场合
SRAM	可读可写，读写速度快，只要不掉电信息就不丢失	Cache
DRAM	可读可写，读写速度不及 SRAM，需要定时刷新，只要不掉电信息就不丢失	主存储器
ROM	信息在出厂前光刻写入，信息长期保存且不受掉电影响。但只能读出且不能再写入	固化程序、微程序控制器
PROM	出厂后只可一次性再写入，写入后的信息不因掉电而丢失，只能读出但不能再写入	自编程序，用于工业控制或电器中
EPROM	可多次紫外线擦除、高压重写入新信息，写入后的信息不因掉电而丢失，只能读出但不能随机再写入	用于产品试制阶段试编程序
E2PROM	可读可写，写入后的信息不因掉电而丢失	IC 卡上存储信息
Flash Memory	可快速读写，写入后的信息不因掉电而丢失	固态盘、IC 卡

3.12 以单管 DRAM 为例，基本存储电路中信息信息的存放依靠电容，电容中有电荷时表示存储的信息为 1，无电荷时表示存储的是 0。由于任何电容都存在漏电问题，所以即使电容中有电荷，过一段时间后随着电荷的流失，信息也就丢失了。所以必须进行定期刷新，即每隔一定时间刷新一次，使电容中原来处于逻辑电平 1 的电荷又得到补充，而原来处于电平 0 的电容仍保持 0。

Intel 2164 芯片内部可寻址 64K 个单元(需要 16 条地址线)，为了减少地址线引脚数(只引出 8 条地址线)，该芯片采用分时复用技术，将片内地址线分为行地址线和列地址线，行地址选通信号 \overline{RAS} 和列地址选通信号 \overline{CAS} 分别用于行、列地址的选通。 \overline{RAS} 还用于刷新时的地址选通信号。

- 3.13 第一片 6116 的寻址范围：00000H~007FFH
第二片 6116 的寻址范围：00800H~00FFFH

- 3. 14 32KB
- 3. 15 4FFFH
- 3. 16



3. 17 由于计算机对存储器性能指标的基本要求是容量大、速度快、成本低，但是在一个存储器中却不能同时兼顾这些相互矛盾的指标。所以我们将各种不同存储容量、存取速度和价格的存储器按层次结构组织起来，并通过管理软件和辅助硬件有机地组成统一的整体，使所存放的程序和数据按层次分布在各级存储器中，形成存储器系统的多级层次结构。一般计算机存储器系统的多级层次结构主要由CPU内部寄存器、高速缓冲存储器（Cache）、主存储器和辅助存储器组成，由它们构成的存储器组织能够充分发挥存储速度快、容量大、价格低的特点。

3. 18 高速缓冲存储器（Cache）是一种存储容量较小但存取速度却很快的存储器，它位于CPU和主存之间，用来存放CPU频繁使用的指令和数据。由于使用Cache后可以减少对慢速主存的访问次数，解决了CPU与主存之间的速度差异，所以提高了CPU的工作效率。目前，在高档微型计算机中广泛使用高速缓冲存储器技术。

虚拟存储器是在“主存—辅存”层次结构上进一步发展和完善的存储管理技术。虚拟存储器把主存和辅存视为一个统一的虚拟主存，提供比实际主存容量大得多的、可使编程空间不受限制的虚存空间；在程序中使用虚地址，使程序不必作任何修改，即可用接近主存的速度在这个虚拟存储器上运行。使得在用户心目中，计算机系统好像只有一个大容量、高速度、使用方便的存储器，而没有主存、辅存之分。目前，几乎所有的计算机都采用虚拟存储器系统。

第4章 微型计算机输入/输出系统概述

4. 1 接口是指CPU与主存储器、外部设备之间，或者两个主机之间进行连接的逻辑电路，是CPU与外界进行信息交换的通道。

I/O设备种类繁多，结构和工作原理各不相同；各种I/O设备的工作速度差异大，与CPU速度不匹配；不同的I/O设备工作时序有差异，难以与CPU配合；各种I/O设备的信息表示格式不一致；各种I/O设备所处理的信息类型及信号电平不一致。由于这些原因，若让

CPU 直接控制和管理各种 I/O 设备，直接与 I/O 设备交换数据，由 CPU 直接控制外设的启动、数据转换，就会增加 CPU 的负担，严重降低 CPU 的效率，降低整个系统的性能。所以主机与 I/O 外设交换数据时，要通过 I/O 接口进行连接，利用接口来控制和管理 I/O 设备，解决如上所出现的各种问题。

4.2 端口即 I/O 接口电路中的寄存器。

端口

- 4.3 1) 数据缓冲与锁存功能。
- 2) 地址译码和设备选择功能。
- 3) 接收并执行 CPU 命令，控制和监测外设的功能。
- 4) 数据格式转换功能。
- 5) 信号转换功能。
- 6) 中断或 DMA 管理功能。
- 7) 可编程功能。

- 4.4 1). 按数据传送方式分为并行接口和串行接口
- 2). 按输入/输出的信号类型分为数字接口和模拟接口
- 3). 按使用灵活性分为不可编程接口和可编程接口
- 4). 按接口使用的功能特征分为通用接口和专用接口

4.5 数据信息、状态信息和控制信息

数据端口、状态端口、控制端口

数据端口可读可写，状态端口只可读，控制端口只可写

4.6 I/O 独立编址方式、存储器映像编址方式

I/O 独立编址方式下，外设端口与主存储器的地址空间分开，易于程序设计，I/O 指令执行速度快，但是对端口操作的专用指令少，程序设计的灵活性差。存储器映像编址方式的指令类型丰富，编程灵活、方便，端口地址空间和外设数目可以很多，读写控制逻辑也比较简单。但是对端口的寻址时间相对较长，延长了 I/O 操作时间，端口占用了存储器地址空间从而相对减少了主存的可用范围，由于访问存储器和端口的指令相同使得程序的可读性降低。

I/O 独立编址方式

0000H~FFFFH

0000H~03FFH

4.7 程序控制方式、中断控制方式、直接存储器存取方式（DMA 方式）和输入/输出处理机方式（IOP 方式）

程序控制方式下的硬件接口电路和软件设计较简单，但是 CPU 效率较低，只可用于传送速度要求不高的场合；中断控制方式下的 CPU 可与多个外设同时并行工作，可以同时响应多个外设的中断请求，系统效率较高，适用于处理中低速外设的 I/O 操作与随机请求的场合，尤其适合实时控制及紧急事件的处理，但因 CPU 在每次中断响应前后需要增加一些额外开销（用于断点和现场的保护与恢复），故这种方式不适合于需要频繁快速 I/O 的场合；DMA 方式下的 I/O 数据传送速度快，系统的吞吐能力强，节省 CPU 的开销，系统效率高，但因需要 DMAC，故硬件开销大，适合于高速外设与内存之间、内存的两个区域之间或两种高速外设之间的高速批量数据传输场合；IOP 方式下的 I/O 工作由 IOP 独立控制，CPU 的效率很高，IOP 和 CPU 并行工作，但是其并行程度受到系统总线的限制，适用于高档微型计算机及大、中型计算机系统中。

4.8 在微型计算机系统中，主机与所有部件都是通过数据总线进行数据传输的，在某一总线周期内，只有被选中的部件才能使用数据总线传输数据。对于输出设备，不可能在短短的

一个总线周期内接收并驱动设备产生动作，所以需要在输出接口电路中安排锁存器，以便锁存输出的数据，使较慢的外设有足够的时间进行处理，避免数据丢失；对于输入设备，向主机传送数据时，主机不一定及时响应，不能马上取走数据，所以需要输入接口设置缓冲器暂时保存数据。所以在 I/O 接口电路中需要使用锁存器和缓冲器。

4.9 在中断传送方式中，由于慢速外设在自身准备就绪后才会向 CPU 发出中断请求，并且 CPU 通常是执行自己的主程序和任务，只有在接到外设的中断请求并响应后，才去运行中断服务子程序，处理外部事件，处理后再恢复执行原来的主程序，这种中断方法使得 CPU 在一段时间内与外设并行工作，并可以同时管理多个外设的工作，所以说中断方式可以实现 CPU 与外设的并行工作。

4.10 利用中断方式进行数据传送是在中断控制器的硬件支持下，通过 CPU 调用执行相应的中断服务子程序来实现的，是由软硬件相结合实现的。

4.11 由于在磁盘与内存之间的数据传送是大批量的，需要快速传送，故需要采用 DMA 方式实现。

第 5 章 并行接口技术

5.1 A B 5.2 B

5.3 并行接口的输入和输出都是多位并行的，传输的效率高。

并行接口多应用于主机与近距离外设进行大量高速数据传输的场合。

5.4 8255A 是可编程并行接口芯片。

8255A 主要由三个数据端口、两组控制电路、一个数据总线缓冲器和一个读/写控制逻辑电路组成，三个数据端口分别是 A 口、B 口、C 口，分别可以与外设连接，进行数据的并行输入或输出。

5.5 $\overline{CS}=0$ 、 $A1A0=01$ 、 $\overline{RD}=0$ 、 $\overline{WR}=1$

5.6 工作方式控制字、C 口置位/复位控制字

控制字格式及每位的含义参见 P125~P126。

程序设计时，把这两个控制字写入 8255A 的控制口。

8255A 通过控制字的 D7 位来区分这两个控制字，当 D7=1 时为工作方式控制字，当 D7=0 时为 C 口置位/复位控制字。

5.7 1) A 口工作于方式 2，B 口方式 1 输出时，8255A 端口 C 各位的作用是：当 B 口内的数据已满需要外设读走时，PC2 用于向外设输出 B 口的输入缓冲器满信号 \overline{OBF} ；当外设准备好了接收数据时，PC1 用于向 B 口输入外设的应答信号 \overline{ACK} ；当 B 口的数据被外设读走后，PC0 用于输出 B 口的中断请求信号 INTR，请求 CPU 向 B 口输出下一个数据。当 A 口接收到外设数据并等待 CPU 读走数据时，PC5 用于发出输入缓冲器满信号 IBF，通知当外设暂时不能向 A 口输入下一个数据；当外设向 A 口输入数据时，通过 PC4 向 A 口输入选通信号 \overline{STB} ；针对于 A 口 PC6 与 PC7 的作用与 PC1、PC2 的作用类同；当 A 口输入数据时的输入缓冲器满或 A 口输出数据时的输出缓冲器空时，需要 CPU 读走数据或送出下一个数据时，通过 PC3 向 CPU 发出中断请求信号。

2) (略)

3) (略)

4) (略)

```

5.8 1) MOV    AL,9CH
      OUT    93H,AL
      2) MOV    AL,0B4H
      OUT    93H,AL
      3) MOV    AL,0C0H
      OUT    93H,AL
      MOV    AL,09H
      OUT    93H,AL
      MOV    AL,0DH
      OUT    93H,AL
      4) MOV    AL,0A3H
      OUT    93H,AL
5.9  MOV    DX,控制口地址
      MOV    AL,06H
      OUT    DX,AL
      MOV    AL,09H
      OUT    DX,AL
5.10 MOV    AL,06H
      OUT    63H,AL
      MOV    AL,07H
      OUT    63H,AL
5.11 MOV    AL,09H
      OUT    63H,AL
      MOV    AL,0DH
      OUT    63H,AL
5.12 1) K1 闭合, K2 闭合时, P0、P1、P2、P3 灯亮
      2) K1 断开, K2 断开时, P2 灯亮
      3) K1 闭合, K2 断开时, P1 灯亮
      4) K1 断开, K2 闭合时, P0 灯亮

```

第 6 章 定时/计数技术及其接口

6.1 A 6.2 A 6.3 B 6.4 B 6.5 C 6.6 BF

6.7 CLK 引脚用于向计数通道输入工作时钟信号, 这个信号是计数通道工作的计时基准, 通道内的计数器就是按照这个时钟频率进行减 1 计数的; GATE 引脚为门控信号, 用于控制计数通道的启动或停止; OUT 引脚为计数器输出信号, 当通道内的计数单元计数结束时, 该引脚即产生输出信号。

6.8 在 IBM PC 系列机中 8253-5 的计数通道 0 工作于方式 3, 用于系统定时; 通道 1 工作于方式 2, 用动态存储器刷新定时; 通道 2 工作于方式 3, 用于扬声器发声控制。

6.9 在对 8253 初始化编程写入计数初始值时, 应在以下几方面注意与控制字保持一致:

- 1) 先写入控制字, 后写入计数初始值。
- 2) 计数初始值要写入控制字中所选定的计数通道。
- 3) 写入的计数初始值格式 (16 位或 8 位) 要与控制字中规定的一致。

6.10 CLK 输入的时钟频率 = OUT 输出的方波频率 \times 计数初始值

6. 11 2ms

6. 12 100

6. 13 MOV DX,30BH

MOV AL,29H

OUT DX,AL

MOV DX,308H

MOV AL,4H

OUT DX,AL

6. 14 MOV DX,343H

MOV AL,76H

OUT DX,AL

MOV DX,341H

MOV AX,0340H

OUT DX,AL

MOV AL,AH

OUT DX,AL

MOV DX,343H

MOV AL,0A5H

OUT DX,AL

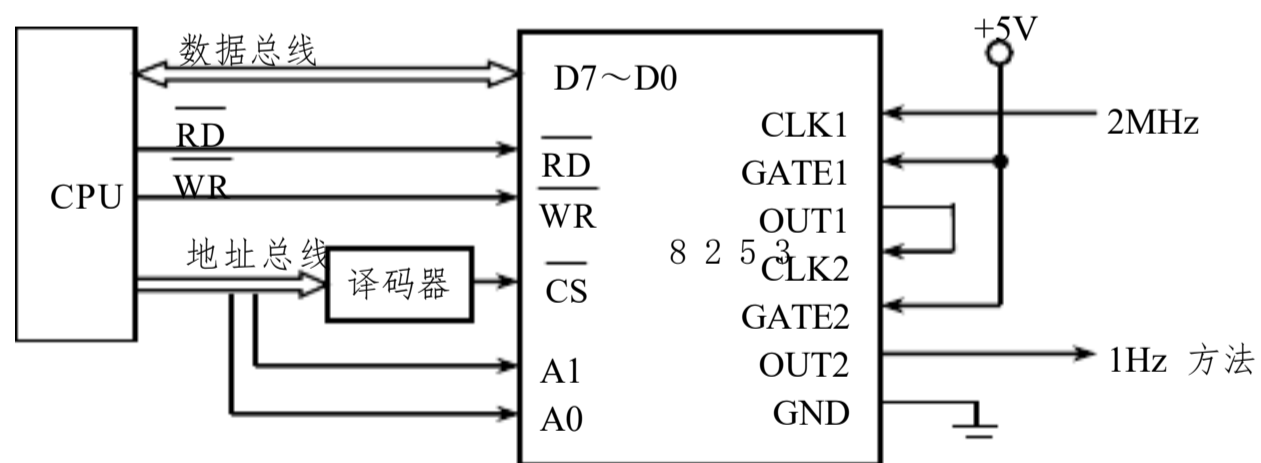
MOV DX,342H

MOV AL,01H

OUT DX,AL

6. 15 若使用 1 个计数通道，则计数初始值 n 应为 2×10^6 ($2\text{MHz} \div 1\text{Hz}$)，而 8253 的计数通道为 16 位，其最大计数值是 65536，远远小于 2×10^6 ，所以使用 1 个计数通道不能实现。

设计 8253 计数通道引脚连接图如下：



MOV DX,343H

MOV AL,67H

OUT DX,AL

MOV DX,341H

MOV AL,10H

; 计数通道 1 的计数初始值为 1000

OUT DX,AL

MOV DX,343H

MOV AL,0A7H

OUT DX,AL

```

                DX,342H
MOV     AL,10H      ;      2 的计数初始值为 1000
OUT     DX,AL

DATA    SEGMENT
PORT_0 EQU 40H
PORT_1 EQU 41H
PORT_2 EQU 42H
PORT_MODE EQU 43H
B_8255A EQU 61H
MUSIC  DW 3F0H,3F0H,380H,380H,3F0H,3F0H,5B0H,5B0H,310H,310H,310H,310H
                ; 乐谱 565177
                DW 0                ; 乐曲结束标志
DATA    ENDS
CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
START:  MOV     AX,DATA
        MOV     DS,AX
        MOV     DX,B_8255A
        IN      AL,DX                ; 将 PB1 和 PB0 置 1, 允许计数器通道计数, 扬声器发声
        OR      AL,03H
        OUT     61H,AL
        MOV     DX,PORT_MODE        ; 初始化
        MOV     AL,0B6H
        OUT     DX,AL
        MOV     BX,OFFSET MUSIC
        MOV     AX,[BX]              ; 取第一个音符
LLL:    MOV     DX,PORT_2            ; 向计数器通道 2 写入计数初始值, 即音符, 播放
        OUT     DX,AL
        MOV     AL,AH
        OUT     DX,AL
        INC     BX
        INC     BX
        MOV     AX,[BX]              ; 取下一音符数据
        TEST    AX,0FFFFH
        JZ      EXIT                ; 若到曲尾, 则结束播放, 否则播放下一音符
        CALL   DALLY
        JMP     LLL
DALLY  PROC
        MOV     CX,0A000H            ; 延时子程序
L1:    MOV     DX,0B00H
L2:    DEC     DX

```

```

                L2
    LOOP      L1
    RET
DALLY      ENDP
EXIT:      MOV      AX,4C00H
           INT      21H
CODE       ENDS
           END      START

```

串行通信及串行接口技术

7.1 C 7.2 B 7.3 AD 7.4 BC 7.5 B, C
 7.6 A 7.7 B 7.8 B 7.9 B

7.10 并行通信是利用多条传输线同时传输多位 (bit) 数据, 每条传输线传输一个 bit, 串行通信则是通过单条传输线依次逐位地传送多位数据。

并行通信的在传输速率上远远高于串行通信, 但并行通信的成本却高于串行通信。

因为并行通信的传输速度快, 但是成本高, 所以只适于短距离的高速传输。串行通信的成本低, 但是传输速度比较慢, 所以串通信适合于远距离的中低速通信场合。

7.11 调幅、调相、调频

7.12 异步方式把一个字符看作一个独立的信息传输单元, 而同步方式以数据块为基本传输单位, 一个数据块包括多个字符; 异步方式下的收发双方可以各自使用自己的发送时钟和接收时钟, 而同步方式中的收发两端需要用同一个时钟源作为时钟信号; 异步方式一般用奇偶校验方式, 而同步方式多采用 CRC 校验方式; 异步方式是靠起始位和停止位来实现字符的界定和同步的, 而同步方式下的数据块以同步字符开头, 以校验字符结束; 同步方式的通信效率高于异步方式的通信效率。

7.13 异步方式下的接收端不断地检测串行数据输入线路, 若采样到一个低电平信号 (起始位), 则视为收到一个数据帧的帧头, 然后接收有效数据位和奇偶校验位, 最后接收到一个高电平 (停止位), 视为该数据帧接收结束, 一个字符接收完毕。

同步方式下的接收端不断地检测串行数据输入线路, 当搜索到同步字符之后, 便开始接收数据位, 最终接收到校验字符, 则视为数据块接收结束。

7.14 1200

7.15 0.83ms, 8.3ms

7.16 120

```

7.17      MOV      AL,7BH
           OUT      3DH,AL
           MOV      AL,37H
           OUT      3DH,AL

```

7.18 方式选择控制字: 5EH (甲机); 5EH (乙机)
 操作命令控制字: 33H (甲机); 14H (乙机)
 程序 (参照例 7.5)

第 8 章 中断技术及中断控制器

8.1 C 8.2 B 8.3 A 8.4 A
 8.5 B 8.6 C 8.7 C 8.8 A

8.9 中断就是指 CPU 在正常运行程序时, 响应中断请求, 转而去执行中断服务子程序, 完

微型计算机系统的中断处理过程：

- 1) 识别中断源。
- 2) 关中断、保护断点、保护现场。
- 3) 开中断。
- 4) 中断服务。
- 5) 关中断。
- 6) 恢复现场、恢复断点、开中断，中断返回。

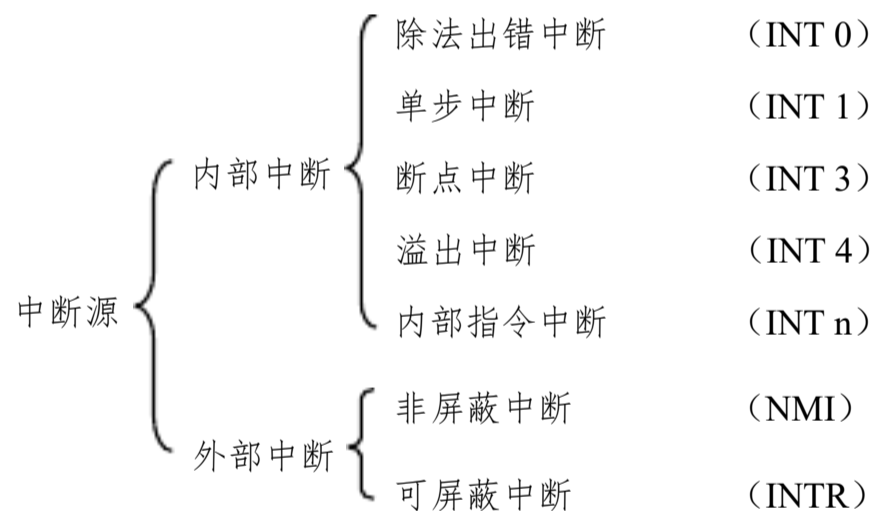
8.10 对 8259A 的编程有初始化编程和操作方式编程两类。

初始化编程是在 8259A 进入操作前，为了设置其初始状态，使用初始化命令字 (ICW) 实现的；操作方式编程是在 8259A 初始化之后，进入工作状态期间，使用操作控制字 (OCW) 以控制 8259A 按不同方式操作。

8.11 在软件中断方式下，CPU 根据 INT n 指令中的中断类型号 n，将其乘以 4 后，得到在中断向量表中的存放地址，按照这个地址从中断向量表中取出对应单元中的内容，即获得 n 号中断服务子程序入口地址。

在硬件中断方式下，系统中有专门的硬件中断控制器 8259A 进行中断管理，由其向 CPU 提供被响应中断源的中断类型号，CPU 在中断响应周期的第二个 $\overline{\text{INTA}}$ 周期，把 8259A 送上数据总线上的中断类型号读走，然后将其乘以 4 后，得到在中断向量表中的存放地址，按照这个地址从中断向量表中取出对应单元中的内容，即获得为该中断源服务的中断服务子程序入口地址。

8.12 8086 中断系统可处理的中断源：



由高到低的优先级顺序是：内部中断（除法出错中断、指令中断、溢出中断）→非屏蔽中断→可屏蔽中断→单步中断，其中各可屏蔽中断请求之间的中断优先级由 8259A 排队。

8.13 1) 固定优先级方式。这种优先级排队顺序固定不变。

2) 自动循环优先级方式。在这种方式下，从 IRQ0~IRQ7 各个中断轮流具有最高优先级。即当某一级中断被处理完毕后，它的优先级别就被改变为最低，而最高优先级分配给该中断相邻的下一级。

3) 特殊循环优先级方式。该方式下可以通过操作命令字 OCW2 设定某中断源编码为最低优先级。

8.14 0AH

8.15

```
MOV    AL,04H          ;OCW1
MOV    DX,奇地址
OUT    DX,AL
STI
```

8.16 硬件中断是由外部硬件引起的，软件中断是由主机内部产生或者由程序预先安排产生

而言大多是随机的，所以说断点是不可预知的，而软件中断则是 CPU 执行到程序中的中断指令或遇到执行错误时产生的，对于 CPU 而言软件中断是可预知的，断点是预知的；对于硬件中断，需要 CPU 在执行完每条指令后，检测中断请求输入线以确定是否有硬件中断请求信号，而软件中断则无需硬件检测；对于硬件中断，中断类型号是由专门的硬件控制电路向 CPU 提供的（NMI 中断除外），而软件中断却在程序中将中断类型号直接提供给 CPU。

8.17 18H 号中断的中断向量在中断向量表中存放的位置是 0000H: 0060H；存放的内容是 (00060H) = 14H, (00061H) = 63H, (00062H) = 20H, (00063H) = 00H

```

8.18      AL,13H          ;ICW1
          OUT 90H,AL
          MOV AL,80H      ;ICW2
          OUT 91H,AL
          MOV AL,03H      ;ICW4
          OUT 91H,AL
8.19      MOV AL,1BH      ;ICW1
          OUT 90H,AL
          MOV AL,80H      ;ICW2
          OUT 91H,AL
          MOV AL,01H      ;ICW4
          OUT 91H,AL
          IN  AL,91H      ;读 IMR
          CLI
          OR  AL,03H      ;屏蔽 IRQ0、IRQ1
          OUT 91H,AL      ;写 OCW1
          MOV AX,0        ;准备重设中断向量表，修改 80H、81H 号的中断向量
          MOV ES,AX
          MOV DI,200H     ;200H = 80H*4，IRQ0 中断类型号为 80H
          MOV AX,3500H
          STOSW           ;设置新 80H 号中断向量的偏移地址
          MOV AX,0000H
          STOSW           ;设置新 80H 号中断向量的基地址
          MOV DI,204H     ;204H = 81H*4，IRQ1 中断类型号为 81H
          MOV AX,4060H
          STOSW           ;设置新 81H 号中断向量的偏移地址
          MOV AX,0000H
          STOSW           ;设置新 81H 号中断向量的基地址
          IN  AL,91H      ;读 IMR
          AND AL,0FCH     ;0FCH = 11111100B，撤销 IRQ0、IRQ1 的屏蔽
          OUT 91H,AL      ;写 OCW1
          STI             ;开中断
          .....

```

第 9 章 DMA 技术及 DMA 控制器

9.1 C 9.2 D 9.3 D 9.4 D 9.5 D

9.6 1) 能接受 CPU 的编程，以便进行功能设定。

- 2) 能接收 I/O 接口的 DMA 请求，并向 CPU 发出总线请求信号，请求总线控制权。
- 3) CPU 响应总线请求之后，DMAC 能接管对总线的控制，进入 DMA 传送过程。
- 4) 能实现有效的寻址，即能输出地址信息并在数据传送过程中自动修改地址指针。
- 5) 能向存储器和 I/O 接口发出相应的读/写控制信号。
- 6) 能控制传送数据的字节数，判定 DMA 传送是否结束。

DMA 结束时，能发出 DMA 结束信号，释放总线，恢复 CPU 对总线的控制。

9.7 8237A 有主态和从态两种工作状态。

从态下的 8237A 与其他接口一样，可以接受 CPU 对它的读/写操作，这时的 8237A 为总线从部件，没有总线控制权；主态下的 8237A 作为总线主部件，获得了总线控制权，可以对 I/O 接口和存储器进行读/写操作，从而控制数据在 I/O 接口与存储器之间直接传送。

9.8 在 DMA 方式下，内存与高速外设之间的数据传输无需 CPU 的直接参与，而是由硬件 DMA 控制器直接控制系统总线进行的，所以说 DMA 方式能实现高速数据传送。

DMA 方式传送的一般过程：（以数据从外设向内存传输为例说明）

1) CPU 对 DMAC 进行功能设定，送入存储器的起始地址，数据长度等参数。

2) 从 I/O 接口向 DMAC 发出 DMA 请求信号。

3) DMAC 向 CPU 发出总线请求。

4) CPU 执行完现行的总线周期后，向 DMAC 回送总线响应信号。

5) CPU 将控制总线、地址总线、数据总线让出，由 DMAC 控制。

6) DMAC 向外部设备发出 DMA 响应信号。

7) 进行 DMA 传送，即由 DMAC 发出 I/O 读信号，把数据读到数据总线上，向地址总线发出存储器地址，通过控制总线发出存储器写信号，把数据总线上的数据写入指定的存储器单元。

8) DMAC 修改内部地址寄存器增/减 1，字节计数器减 1，准备下一个数据的传送。

9) 重复第 7)、8) 步，直至设定的字节数传送完毕。

10) DMAC 撤消向 CPU 的请求信号，释放总线，CPU 重新控制总线。

9.9 8237A 的工作方式：单字节传送方式、块传送方式、请求传送方式、级联传送方式。

8237A 的传送类型：DMA 读、DMA 写、DMA 校验和存储器到存储器传送。

9.10 8237A 的先/后触发器的功能：因为 8237A 的数据总线为 8 位，所以对 16 位的地址寄存器和字节计数器进行读/写操作时需要连续两次进行，先/后触发器用来确定 CPU 对 8237A 内部 16 位寄存器的读/写是低字节还是高字节。先/后触发器可以跟头式触发两种状态（0 和 1），为 0 时 CPU 访问的是低字节；为 1 时 CPU 访问的是高字节。

9.11 自动预置方式就是指在 DMA 通道计数结束后，该通道自动将基地址寄存器内容装入当前地址寄存器，将基字节计数器内容装入当前字节计数器，做好进行下一次 DMA 传送的准备，而不必通过 CPU 对 DMAC 重新初始化。

9.12 因为 DMAC 具有主、从两种工作状态，为从态时，CPU 通过地址总线对 DMAC 进行初始化，此时的地址总线对于 DMAC 是输入状态；为主态时，DMAC 控制地址总线对内存和外设进行寻址，此时的地址总线对于 DMAC 是输出状态。

9.13 8237A 从态时， \overline{IOW} 和 \overline{IOR} 引脚为输入状态，用于 CPU 对 8237A 的读/写控制， \overline{MEMW} 和 \overline{MEMR} 引脚无效。8237A 主态时， \overline{IOW} 和 \overline{IOR} 引脚为输出状态，用于 8237A 对外设的读/写控制， \overline{MEMW} 和 \overline{MEMR} 引脚也为输出状态，用于对内存的读/写控制。

当 8237A 任意一个通道中的计数结束时， \overline{EOP} 引脚输出一个低电平，作为 DMA 传送结束信号。当外界要强行终止 8237A 的 DMA 传送时，可以通过该引脚送入一个低电平有效的信号。

9.14 8237A 包括哪几个寄存器？各有何作用？（参 P249~P254）

初始化时针对以下寄存器进行预置：向所选择通道的基地址寄存器、当前地址寄存器、基字节计数器和当前字节计数器写入初始值；写工作方式寄存器；写屏蔽寄存器；写命令寄存器；写请求寄存器。

9.15 8237A 主清除命令的作用是使 8237A 复位准备接收新命令，与硬件 RESET 信号的作用相同。

器清零，屏蔽寄存器置 1。

9.16 4 个

每个通道都相互独立。

PC/XT 中 8237A 的 4 个通道分别用于：通道 0 用于对 DRAM 刷新；通道 2 用于软盘和内存之间的高速数据传输；通道 3 用于硬盘和内存之间的高速数据传输；通道 1 保留。

9.17 在 PC/XT 系统中，地址总线为 20 位的，而 8237A 内部地址寄存器却只有 16 位，不能提供最高 4 位地址 (A19~A16)，所以当使用 8237A 时，系统中要增加页面寄存器，由它专门提供系统高 4 位地址。

9.18 (略)

9.19

```

OUT 0DH,AL           ;主清除命令
MOV AX,1200H
OUT 04H,AL           ;先向通道 2 写入地址初始值的低 8 位
MOV AL,AH
OUT 04H,AL           ;后向通道 2 写入地址初始值的高 8 位
MOV AX,1024          ;传送的字节数 1KB
DEC AX               ;初始化时写入的值少 1
OUT 05H,AL           ;先向通道 2 写入计数初始值的低 8 位
MOV AL,AH
OUT 05H,AL           ;后向通道 2 写入计数初始值的高 8 位
MOV AL,46H
OUT 0BH,AL           ;写工作方式寄存器
MOV AL,02H
OUT 0AH,AL           ;写屏蔽寄存器
MOV AL,00H
OUT 08H,AL           ;写命令寄存器

```

9.20 (略)

第 10 章 总线技术

10.1 B 10.2 B 10.3 A

10.4 总线是计算机系统的一组能为多个部件分时共享的公共信息传输通路。

微型计算机的总线通常分为片内总线、局部总线、系统总线和通信总线。

10.5 在微型计算机系统中，各个部件都是通过总线连接在一起的，系统中的各种信号都是利用总线进行传输的。标准总线规定了插件的尺寸、信号线的数目、各信号的定义以及时序和信号的电平标准等。正是由于在微型计算机系统的开发及应用使用了总线标准，所以简化了软件和硬件的设计，简化了系统的结构，便于系统的扩充和更新。

10.6 ISA 总线上前 62 引脚分布及功能与 PC 总线基本相同，可以利用 ISA 前 62 引脚的插槽插入与 PC 总线兼容的 8 位接口插件板，也可以利用整个插槽插入 16 位接口插件板，所以说 ISA 与 PC 总线保持向下兼容。

EISA 总线在物理结构上设计成深度不同的上下两层，上层的引脚分布是与 ISA 总线标准一致的，下层为 EISA 新增加引脚，下层引脚与上层引脚位置横向错开，下层并有用于阻止 ISA 插件板深插的访问键，从而保证 ISA 标准的插件板只能与 EISA 插槽上层的 ISA 信号相连接，而 EISA 标准的插件卡能够插到深层，与上下两层信号相连接，从而实现了 EISA 与 ISA 的向下兼容。

10.7 开发商为系统板设计了高速的局部总线插槽，将同样局部总线标准的高速外设控制卡直接插入这种插槽，使之挂到局部总线上，并以 CPU 速度运行。而对于慢速设备仍保持

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/925341222321012011>