

第 5 章 总线及其形成

1. 微处理器的外部结构表现为 数量有限的输入输出引脚，它们构成了微处理器级总线。
2. 微处理器级总线经过形成电路之后形成了 系统级总线。
3. 简述总线的定义及在计算机系统中采用标准化总线的优点。

答：总线是计算机系统中模块（或子系统）之间传输数据、地址和控制信号的公共通道，它是一组公用导线，是计算机系统的重要组成部分。

采用标准化总线的优点是：

- 1) 简化软、硬件设计。
 - 2) 简化系统结构。
 - 3) 易于系统扩展。
 - 4) 便于系统更新。
 - 5) 便于调试和维修。
4. 在微型计算机应用系统中，按功能层次可以把总线分成哪几类。

答：在微型计算机应用系统中，按功能层次可以把总线分成：片内总线、元件级总线、系统总线和通信总线。

5. 简述 RESET 信号的有效形式和系统复位后的启动地址。

答：RESET为系统复位信号，高电平有效，其有效信号至少要保持四个时钟周期，且复位信号上升沿要与 CLK下降沿同步。

系统复位后的启动地址为 0FFFF0H 即：(CS)=0FFFFH, (IP)=0000H。

6. 8086 CPU 的 $\overline{M}/\overline{IO}$ 信号在访问存储器时为 高 电平，访问 I/O 端口时为 低 电平。
7. 在 8086 系统总线结构中，为什么要有地址锁存器？

答：8086CPU有 20 条地址线和 16 条数据线，为了减少引脚，采用了分时复用，共占了 20 条引脚。这 20 条引脚在总线周期的 T₁ 状态输出地址。为了使地址信息在总线周期的其他 T 状态仍保持有效，总线控制逻辑必须有一个地址锁存器，把 T₁ 状态输出的 20 位地址信息进行锁存。

8. 根据传送信息的种类不同，系统总线分为 数据总线、地址总线 和 控制总线。
9. 三态逻辑电路输出信号的三个状态是 高电平、低电平 和 高阻态。
10. 在 8086 的基本读总线周期中，在 T₁ 状态开始输出有效的 ALE 信号；在 T₂ 状态开始输出

低电平的 \overline{RD} 信号,相应的 DEN 为__低__电平,DT/ R 为__低__电平;引脚 $AD_{15} \sim AD_0$

上在 T_1 状态期间给出地址信息,在 T_4 状态完成数据的读入。

11. 利用常用芯片 74LS373 构成 8086 系统的地址总线, 74LS245 作为总线收发器构成数据总线,画出 8086 最小方式系统总线形成电路。

答: 8086 最小方式系统总线形成电路如图 5.1 所示。

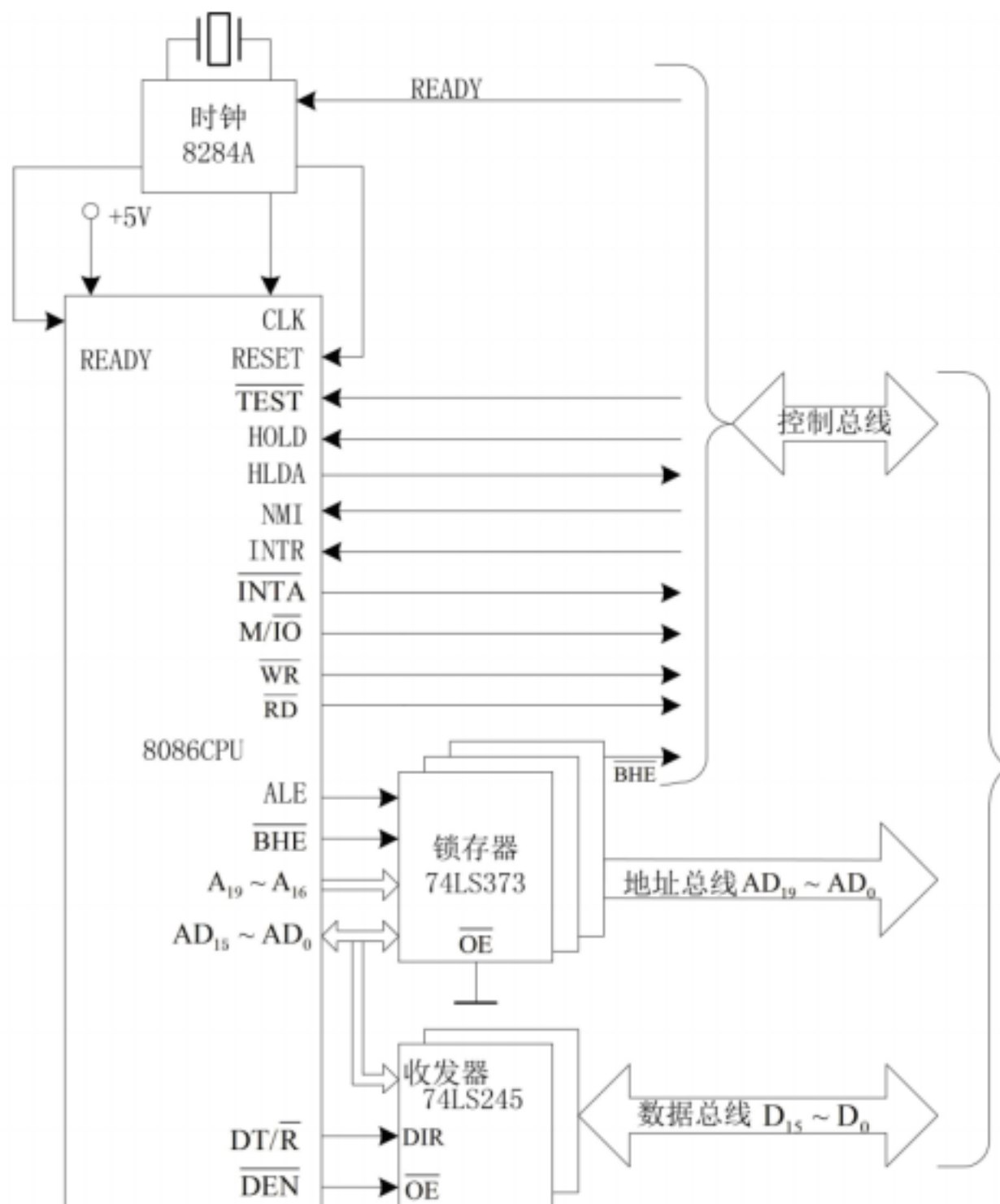


图 5.1 8086 最小方式系统总线形成电路

12. 微机中的控制总线提供 H。

- A. 数据信号流;
- B. 存储器和 I/O 设备的地址码;
- C. 所有存储器和 I/O 设备的时序信号;
- D. 所有存储器和 I/O 设备的控制信号;
- E. 来自存储器和 I/O 设备的响应信号;
- F. 上述各项;
- G. 上述 C, D 两项;

H. 上述 C, D 和 E 三项。

13. 微机中读写控制信号的作用是 E。
- A. 决定数据总线上数据流的方向；
B. 控制存储器操作读 / 写的类型；
C. 控制流入、流出存储器信息的方向；
D. 控制流入、流出 I/O 端口信息的方向；
E. 以上所有。
14. 8086 CPU 工作在最大方式，引脚 $\overline{MN}/\overline{MX}$ 应接 地。
15. RESET 信号在至少保持 4 个时钟周期的 高 电平时才有效，该信号结束后，CPU 内部的 CS 为 0FFFFH，IP 为 0000H，程序从 0FFFF0H 地址开始执行。
16. 在构成 8086 最小系统总线时，地址锁存器 74LS373 的选通信号 G 应接 CPU 的 ALE 信号，输出允许端 OE 应接 地；数据收发器 74LS245 的方向控制端 DIR 应接 $\overline{DI}/\overline{R}$ 信号，输出允许端 \overline{E} 应接 \overline{DEN} 信号。
17. 8086 CPU 在读写一个字节时，只需要使用 16 条数据线中的 8 条，在 一 个总线周期内完成；在读写一个字时，自然要用到 16 条数据线，当字的存储对准时，可在 一 个总线周期内完成；当字的存储未对准时，则要在 两 个总线周期内完成。
18. CPU 在 T_3 状态开始检查 READY 信号，高 电平时有效，说明存储器或 I/O 端口准备就绪，下一个时钟周期可进行数据的读写；否则，CPU 可自动插入一个或几个 等待周期 (T_w)，以延长总线周期，从而保证快速的 CPU 与慢速的存储器或 I/O 端口之间协调地进行数据传送。
19. 8086 最大系统的系统总线结构较最小系统的系统总线结构多一个芯片 8288 总线控制器。
20. 微机在执行指令 MOV [DI], AL 时，将送出的有效信号有 B, C。
- A. RESET B. 高电平的 $\overline{M}/\overline{IO}$ 信号 C. \overline{WR} D. \overline{RD}
21. 设指令 MOV AX, DATA 已被取到 CPU 的指令队列中准备执行，并假定 DATA 为偶地址，试画出下列情况该指令执行的总线时序图：
- (1) 没有等待的 8086 最小方式；
(2) 有一个等待周期的 8086 最小方式。

答：(1) 没有等待的 8086 最小方式时序如图 5.2 所示。

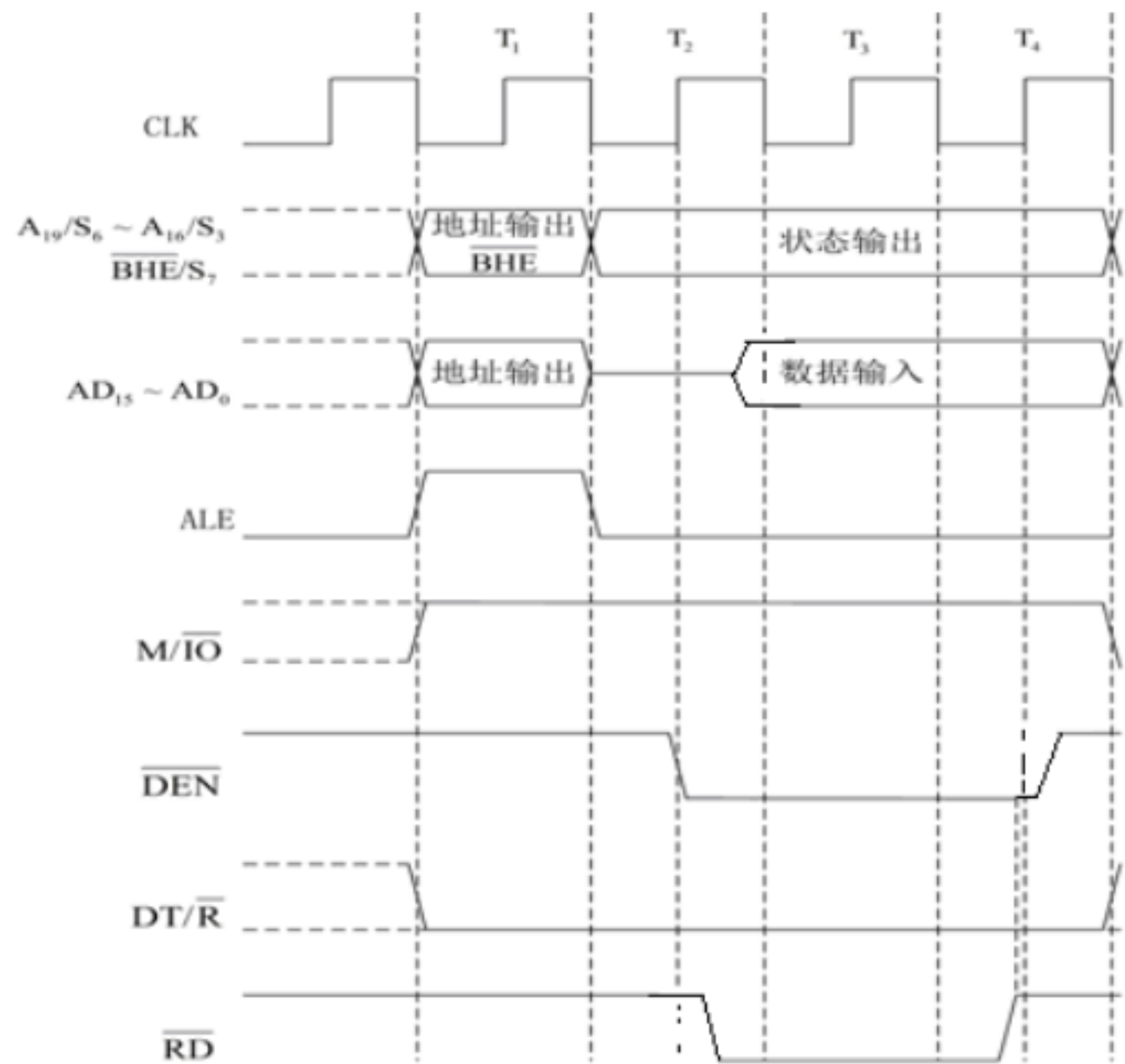


图 5.2 没有等待的 8086 最小方式时序

(2) 有一个等待周期的 8086 最小方式时序图如图 5.3 所示。

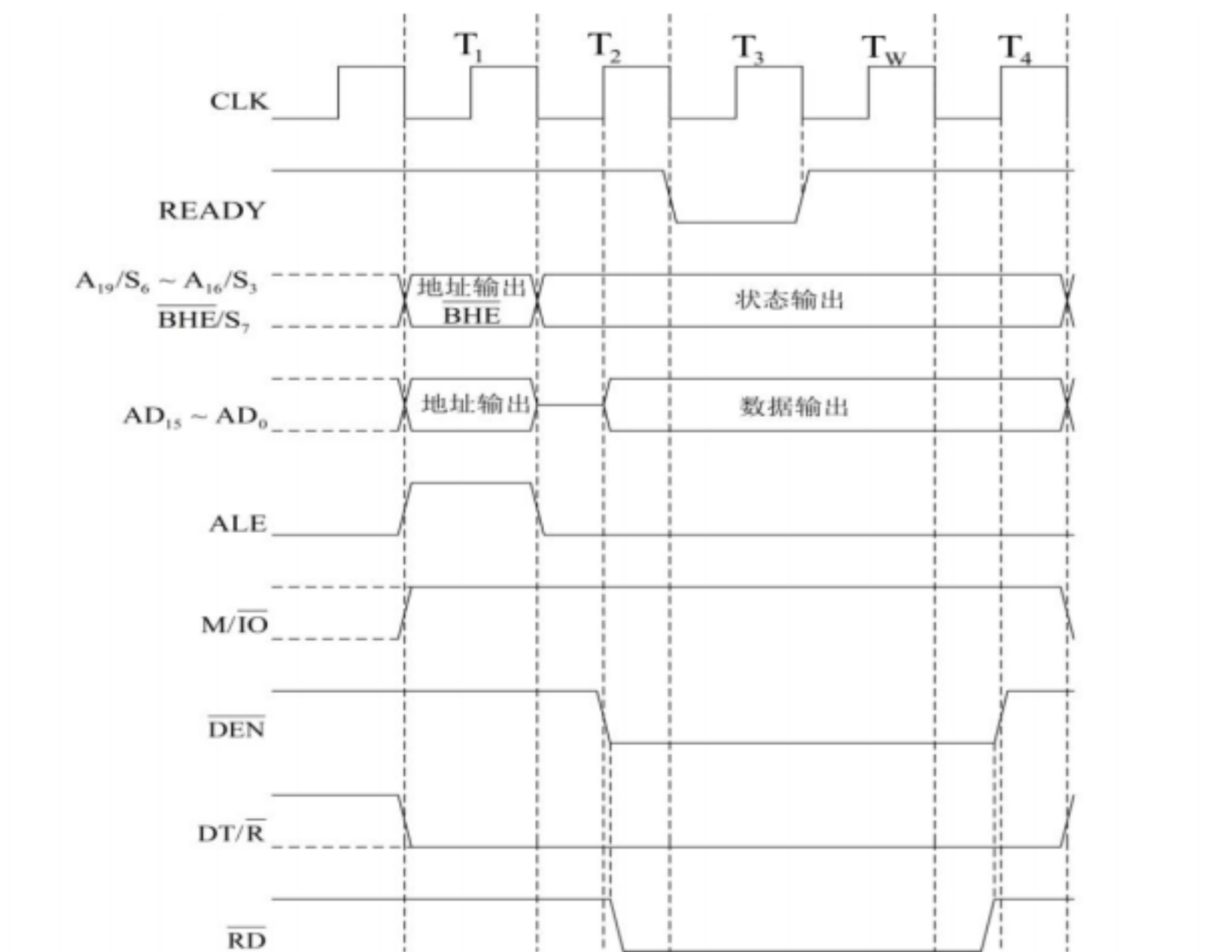


图 5.3 有一个等待周期的 8086 最小方式时序图

22. 上题中如果指令分别为：

- (1) MOV DATA+1 , AX
- (2) MOV DATA+1 , AL
- (3) OUT DX , AX (DX 的内容为偶数)
- (4) IN AL , 0F5H

重做上题 (1)。

答：(1) 因为 DATA 为偶地址，则 DATA+1 为奇地址。故要完成本条指令，需要两个总线周期。时序图如图 5.4 所示。

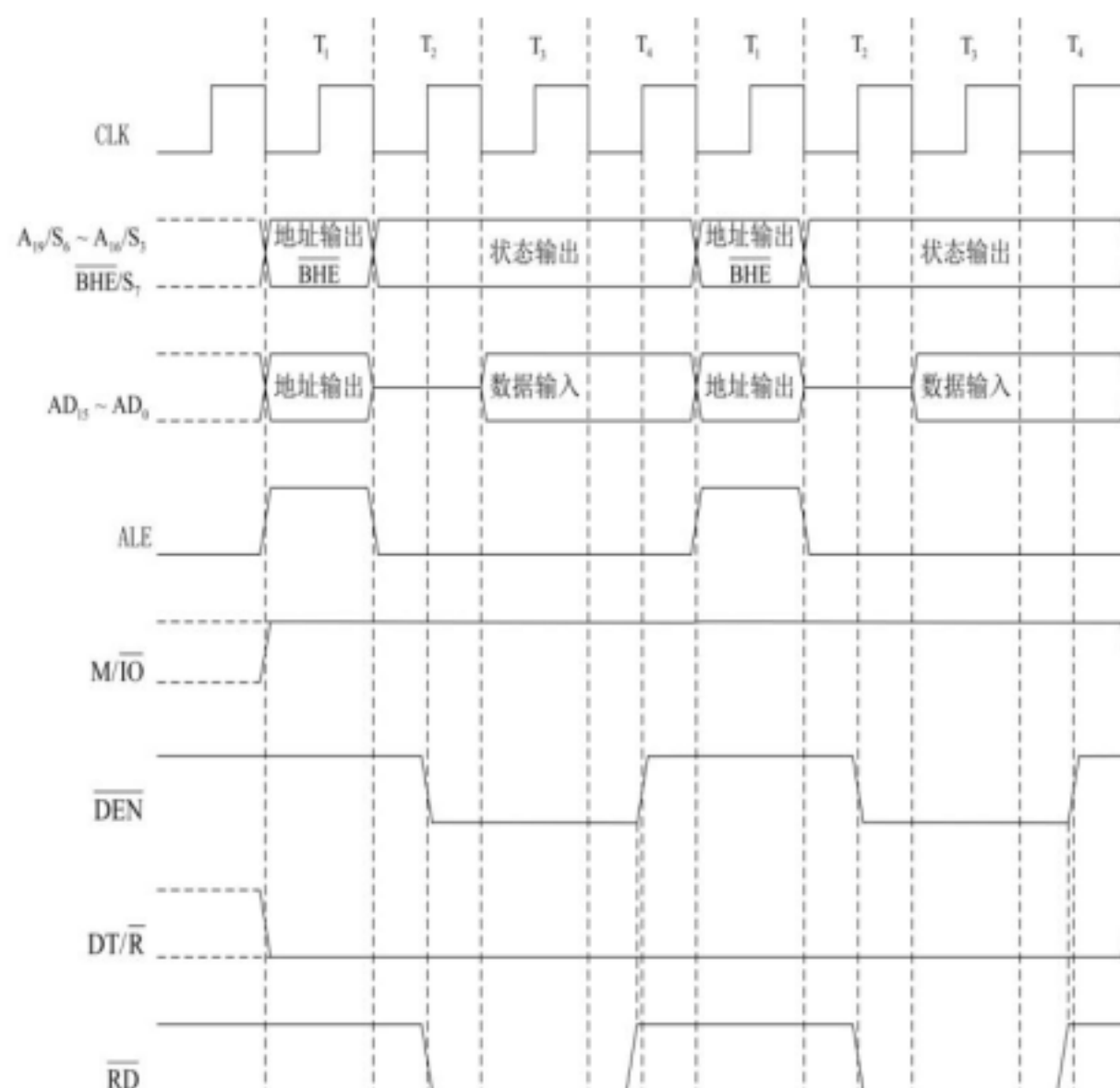


图 5.4 执行 MOV DATA+1, AX 指令的时序参考图

(2) DATA+1 虽然为奇地址，但是 AL 为八位存储器，故本条指令需用一个总线周期，时序图如图 5.5 所示。

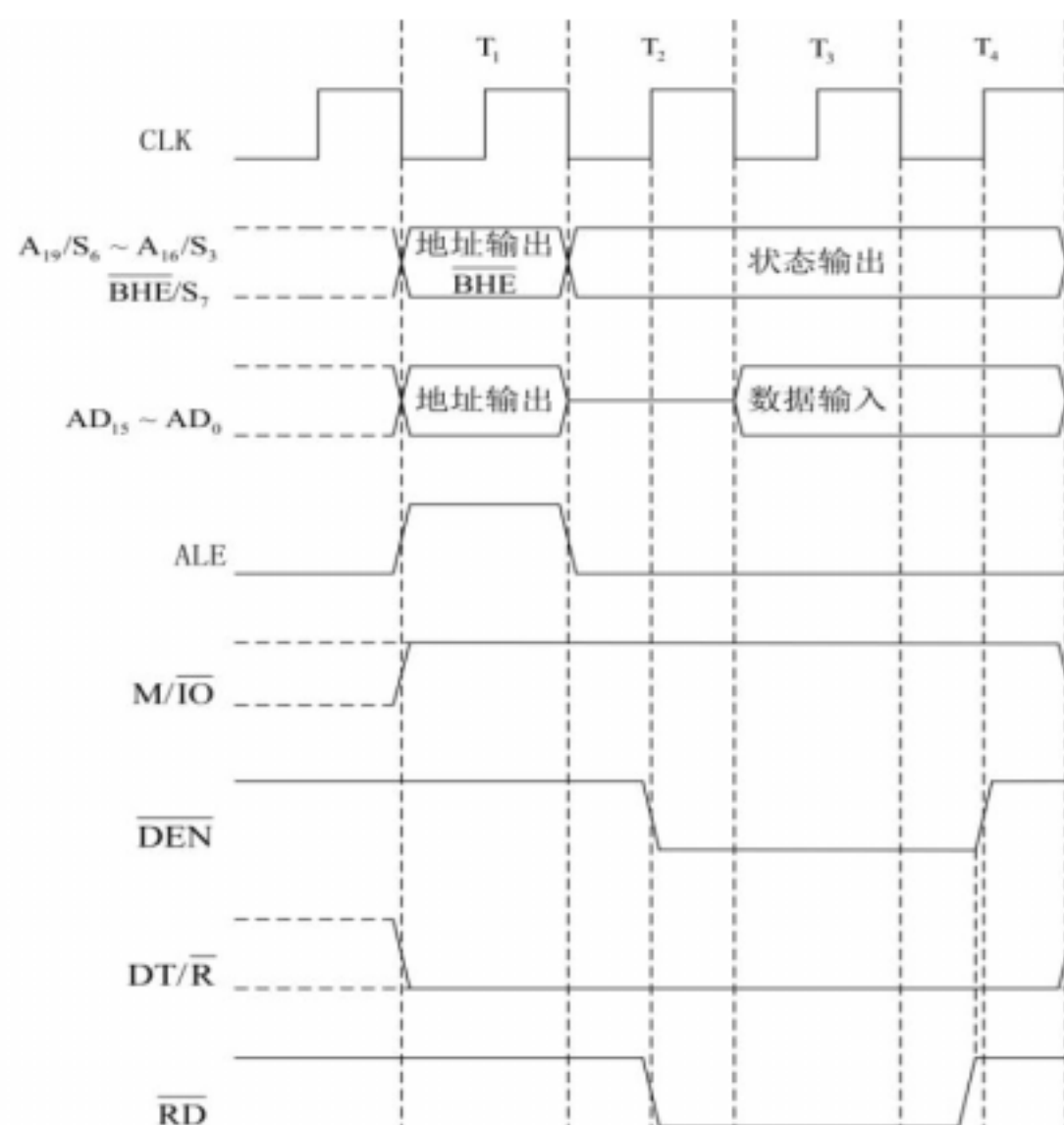


图 5.5 执行 MOV DATA+1, AL 指令的时序参考图

(3) 执行 OUT DX , AX (DX 的内容为偶数) 指令的时序图如图 5.6 所示。

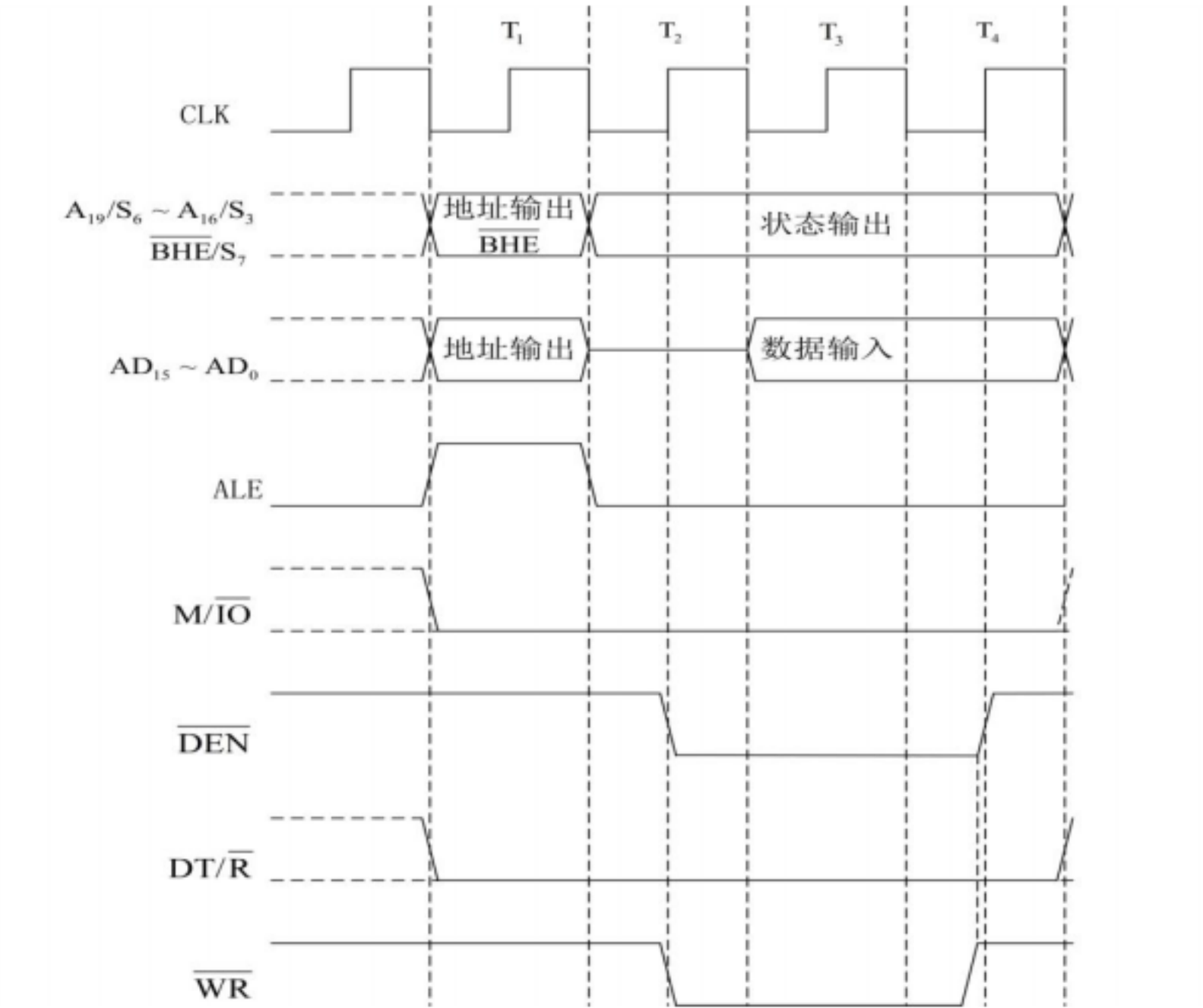


图 5.6 执行 OUT DX , AX 指令的时序参考图

(4) 执行 IN AL , 0F5H 指令的时序图如图 5.7 所示。

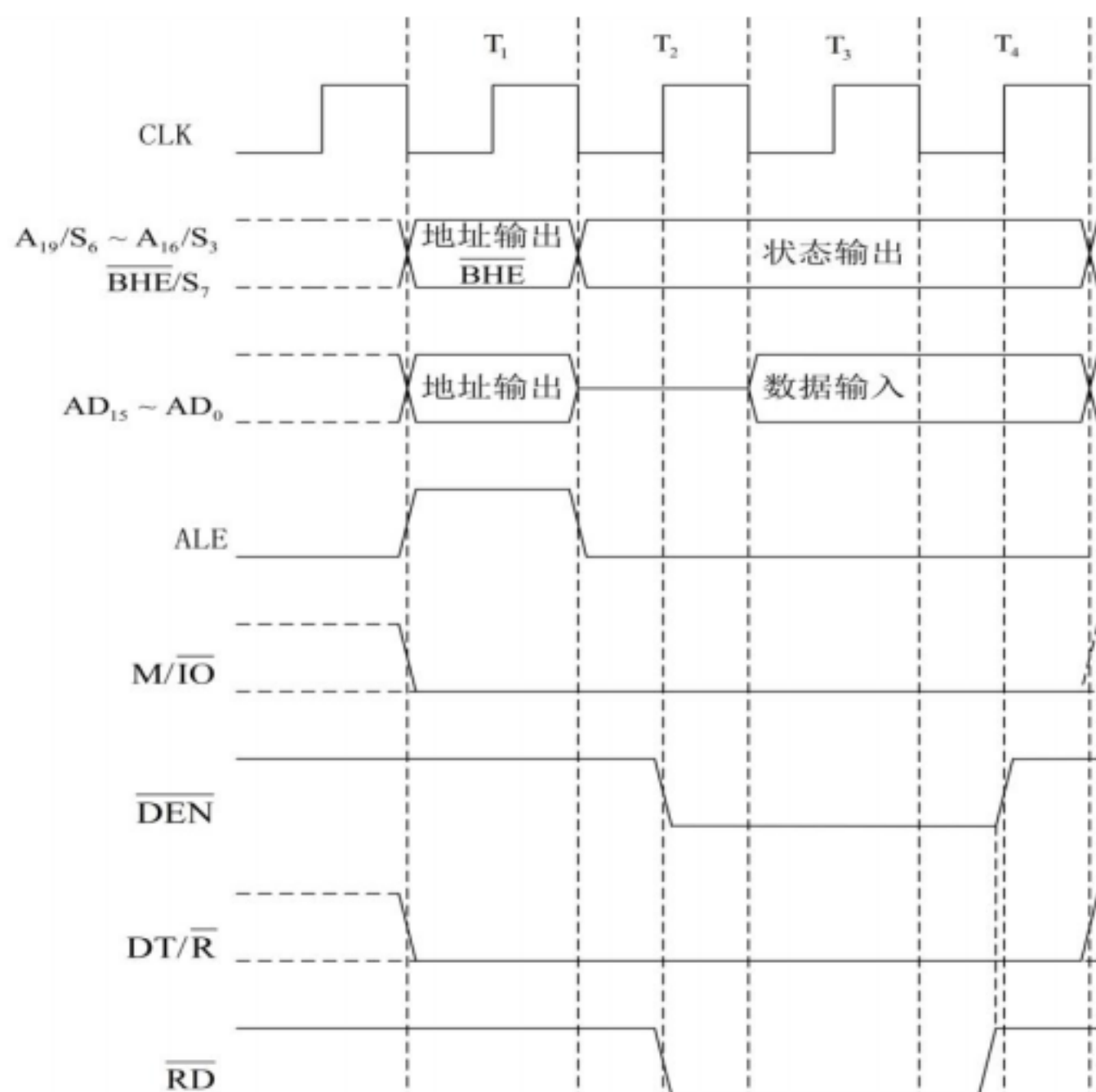


图 5.7 执行 IN AL , 0F5H 指令的时序参考图

23 . 8086 最小方式下，读总线周期和写总线周期相同之处是：在 T₁ 状态开始使 ALE 信号变为有效 高 电平，并输出 M/IO 信号来确定是访问存储器还是访问 I/O 端口，同时送出 20 位有效地址，在 T₁ 状态的后部，ALE 信号变为 低 电平，利用其下降沿将 20 位地址和 BHE 的状态锁存在地址锁存器中；相异之处从 T₂ 状态开始的数据传送阶段。

第 6 章 存储器设计

1. 简述内存储器的分类及每种存储器的用途？

解：内存存储器按其工作方式的不同，可以分为随机存取存储器（简称随机存储器或 RAM）和只读存储器（简称 ROM）。

随机存储器。随机存储器允许随机的按任意指定地址向内存单元存入或从该单元取出信息，对任一地址的存取时间都是相同的。由于信息是通过电信号写入存储器的，所以断电时 RAM 中的信息就会消失。计算机工作时使用的程序和数据等都存储在 RAM 中，如果对程序或数据进行了修改之后，应该将它存储到外存储器中，否则关机后信息将丢失。通常所说的内存大小就是指 RAM 的大小，一般以 KB 或 MB 为单位。

只读存储器。只读存储器是只能读出而不能随意写入信息的存储器。ROM 中的内容是由厂家制造时用特殊方法写入的，或者要利用特殊的写入器才能写入。当计算机断电后，ROM 中的信息不会丢失。当计算机重新被加电后，其中的信息保持原来的不变，仍可被读出。ROM 适宜存放计算机启动的引导程序、启动后的检测程序、系统最基本的输入输出程序、时钟控制程序以及计算机的系统配置和磁盘参数等重要信息。

2. 简述存储器的主要技术指标有哪些？

解：存储器的主要技术指标有：存储容量、读写速度、非易失性、可靠性等。

3. 在实际工程应用中，存储器芯片的速度怎样估算？

解：在选择存储器芯片时应注意是否与微处理器的总线周期时序匹配。作为一种保守的估计，在存储器芯片的手册中可以查得最小读出周期 $t_{cyc}(R)$ (Read Cycle Time) 和最小写周期 $t_{cyc}(W)$ (Write Cycle Time)。如果根据计算，微处理器对存储器的读写周期都比存储器芯片手册中的最小读写周期大，那么我们认为该存储器芯片是符合要求的，否则要另选速度更高的存储器芯片。

8086CPU 对存储器的读写周期需要 4 个时钟周期（一个基本的总线周期）。因此，作为一种保守的工程估计，存储器芯片的最小读出时间应满足如下表达式：

$$t_{cyc}(R) < 4T - t_{da} - t_D - T$$

其中：T 为 8086 微处理器的时钟周期； t_{da} 为 8086 微处理器的地址总线延时时间； t_D 为各种因素引起的总线附加延时。这里的 t_D 应该认为是总线长度、附加逻辑电路、总线驱动器引起的延时时间总和。

同理，存储器芯片的最小写入时间应满足如下表达式：

$$t_{cyc}(W) < 4T - t_{da} - t_D - T$$

4. 用下列 RAM 芯片构成 32kB 存储器模块，各需多少芯片？16 位地址总线中有多少位参与片内寻址？多少位可用作片选控制信号？

(1) 1k × 1 (2) 1k × 4 (3) 4k × 8 (4) 16k × 4

解：(1) 1k × 1

$$\frac{32\text{ K} \times 8}{1\text{ K} \times 1} = 256 \text{ 片},$$

片内寻址： $A_0 : A_9$ ，共 10 位；片选控制信号： $A_{10} : A_{15}$ ，共 6 位。

(2) 1k × 4

$$\frac{32\text{ K} \times 8}{1\text{ K} \times 4} = 64 \text{ 片},$$

片内寻址： $A_0 : A_9$ ，共 10 位；片选控制信号： $A_{10} : A_{15}$ ，共 6 位。

(3) 4k × 8

$$\frac{32\text{ K} \times 8}{4\text{ K} \times 8} = 8 \text{ 片},$$

片内寻址： $A_0 : A_{11}$ ，共 12 位；片选控制信号： $A_{12} : A_{15}$ ，共 4 位。

(4) 16k × 4

$$\frac{32\text{ K} \times 8}{16\text{ K} \times 4} = 4 \text{ 片},$$

片内寻址： $A_0 : A_{13}$ ，共 14 位；片选控制信号： $A_{14} : A_{15}$ ，共 2 位。

5. 若存储器模块的存储容量为 256kB，则利用上题中给出的 RAM 芯片，求出构成 256kB 存储器模块各需多少块芯片？20 位地址总线中有多少位参与片内寻址？多少位可用作片选控制信号？

解：(1) 1k × 1

$$\frac{256\text{ K} \times 8}{1\text{ K} \times 1} = 2048 \text{ 片},$$

片内寻址： $A_0 : A_9$ ，共 10 位；片选控制信号： $A_{10} : A_{19}$ ，共 10 位。

(2) 1k × 4

$$\frac{256\text{ K} \times 8}{1\text{ K} \times 4} = 512 \text{ 片},$$

片内寻址： $A_0 : A_9$ ，共 10 位；片选控制信号： $A_{10} : A_{19}$ ，共 10 位。

(3) $4\text{k} \times 8$

$$\frac{256\text{ K} \times 8}{4\text{ K} \times 8} = 64 \text{ 片},$$

片内寻址： $A_0 : A_{11}$ ，共 12 位；片选控制信号： $A_{12} : A_{19}$ ，共 8 位。

(4) $16\text{k} \times 4$

$$\frac{256\text{ K} \times 8}{16\text{ K} \times 4} = 32 \text{ 片},$$

片内寻址： $A_0 : A_{13}$ ，共 14 位；片选控制信号： $A_{14} : A_{19}$ ，共 6 位。

6. 一台 8 位微机系统的地址总线为 16 位，其存储器中 RAM 的容量为 32kB，首地址为 4000H，且地址是连接的。问可用的最高地址是多少？

解： $32\text{K} = 2^{15} = 8000\text{H}$ ，所以，最高地址为：

$$4000\text{H} + 8000\text{H} - 1 = \text{BFFFH}$$

则，可用的最高地址为 0BFFFH。

7. 某微机系统中内存的首地址为 4000H，末地址为 7FFFH，求其内存容量。

解： $7\text{FFFH} - 4000\text{H} + 1 = 4000\text{H} = 2^{14} = 16\text{KB}$

内存容量为 16KB

8. 利用全地址译码将 6264 芯片接在 8088 的系统总线上，其所占地址范围为 00000H~03FFFH，试画连接图。写入某数据并读出与之比较，若有错，则在 DL 中写入 01H；若每个单元均对，则在 DL 写入 EEH，试编写此检测程序。

解：因为 6264 的片容量为 8KB

RAM 存储区域的总容量为 $03\text{FFFH} - 00000\text{H} + 1 = 4000\text{H} = 16\text{KB}$ ，故需要 2 片 6264 芯片。

连接图如图 6.1 所示。

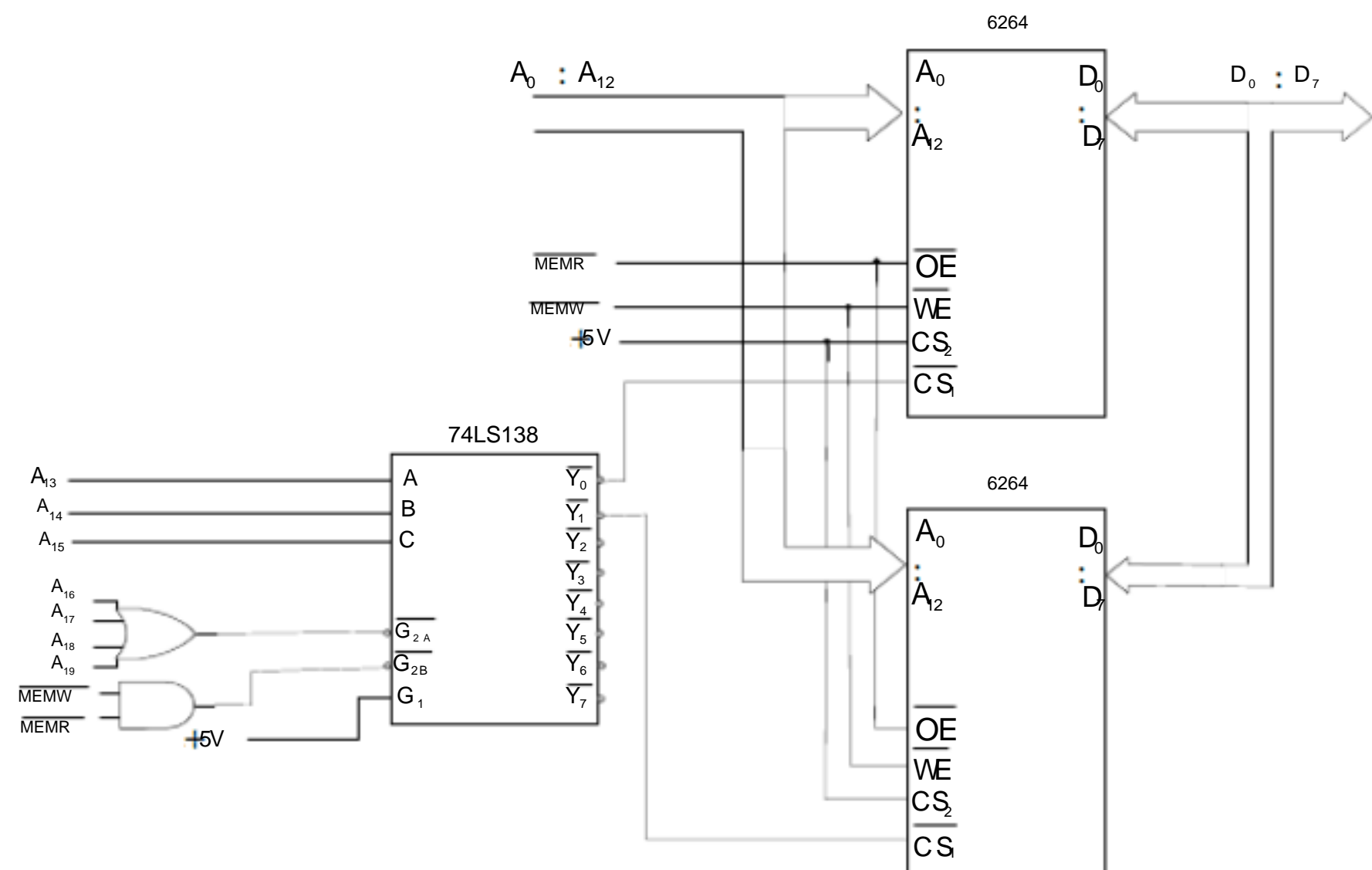


图 6.1 与 8088 系统总线的连接图

检测程序段：

```
MOV AX,0000H
MOV DS,AX
MOV SI,0
MOV CX,16*1024
MOV AL,55H
```

CMPL: MOV [SI],AL

```
MOV BL,[SI]
```

```
CMP BL,AL
```

```
JNE ERROR
```

```
INC SI
```

```
LOOP CMPL
```

```
MOV DL,0EEH
```

```
JMP NEXT
```

ERROR: MOV DL,01H

NEXT:

,

9. 简述 EPROM的编程过程，并说明 EEPROM的编程过程。

解：EPROM芯片的编程有两种方式：标准编程和快速编程。

在标准编程方式下，每给出一个编程负脉冲就写入一个字节的数。Vpp 上加编程电压，地址线、数据线上给出要编程单元的地址及其数据，并使 $\overline{CE} = 0, \overline{OE} = 1$ 。上述信号稳定后，在 PGM 端加上宽度为 $50\text{ms} \pm 5\text{ms}$ 的负脉冲，就可将数据逐一写入。写入一个单元后将 \overline{OE} 变低，可以对刚写入的数据读出进行检验。

快速编程使用 $100\text{ }\mu\text{s}$ 的编程脉冲依次写完所有要编程的单元，然后从头开始检验每个写入的字节。若写的不正确，则重写此单元。写完再检验，不正确可重写。

EEPROM编程时不需要加高电压，也不需要专门的擦除过程。并口线 EEPROM操作与 SRAM 相似，写入时间约 5ms 。串行 EEPROM的操作按时序进行，分为字节写方式和页写方式。

10. 若要将 4 块 6264 芯片连接到 8088 最大方式系统 A0000H~A7FFFH的地址空间中，现限定要采用 74LS138 作为地址译码器，试画出包括板内数据总线驱动的连接电路图。

解：8088 最大方式系统与存储器读写操作有关的信号线有：地址总线 $A_0 : A_{19}$ ，数据总线： $D_0 : D_7$ ，控制信号： $\overline{MEMR}, \overline{MEMW}$ 。

根据题目已知条件和 74LS138 译码器的功能，设计的板内数据总线驱动电路如图 6.2(a) 所示，板内存储器电路的连接电路图如图 6.2 (b) 所示。

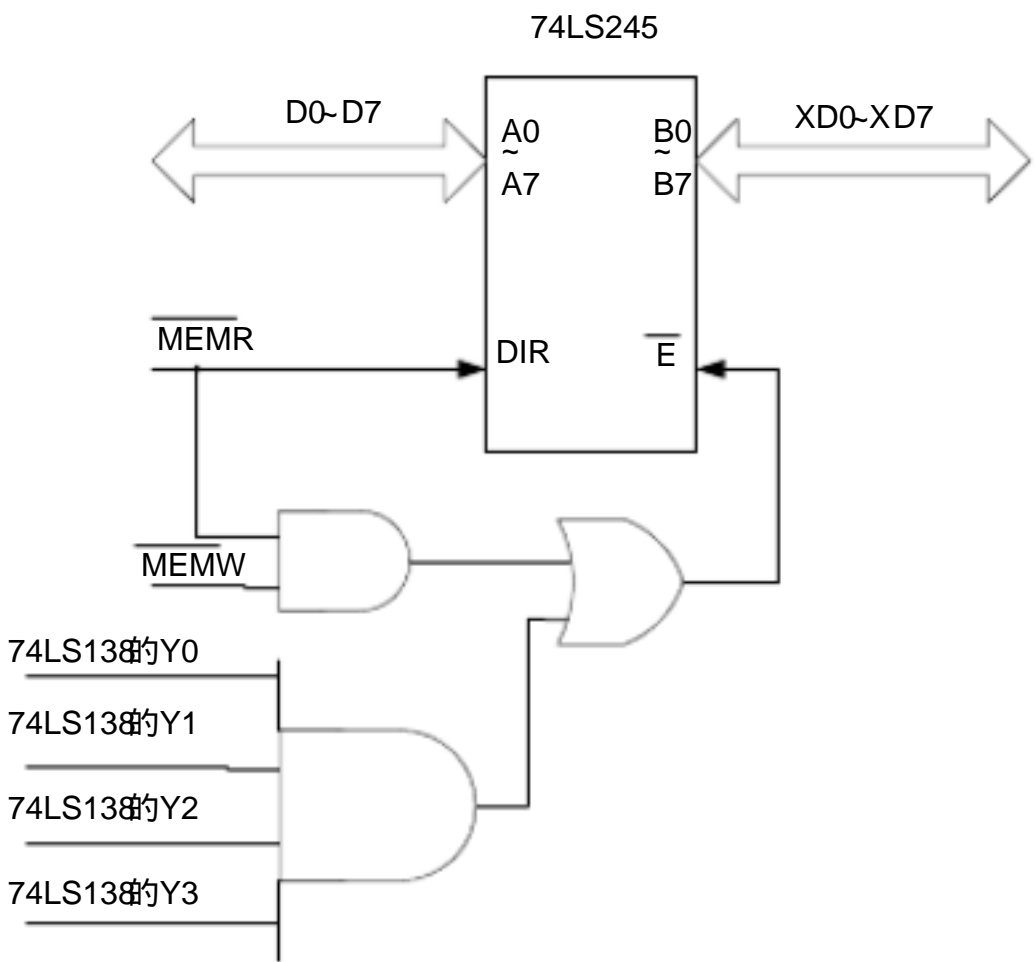


图 6.2 (a) 板内数据总线驱动电路

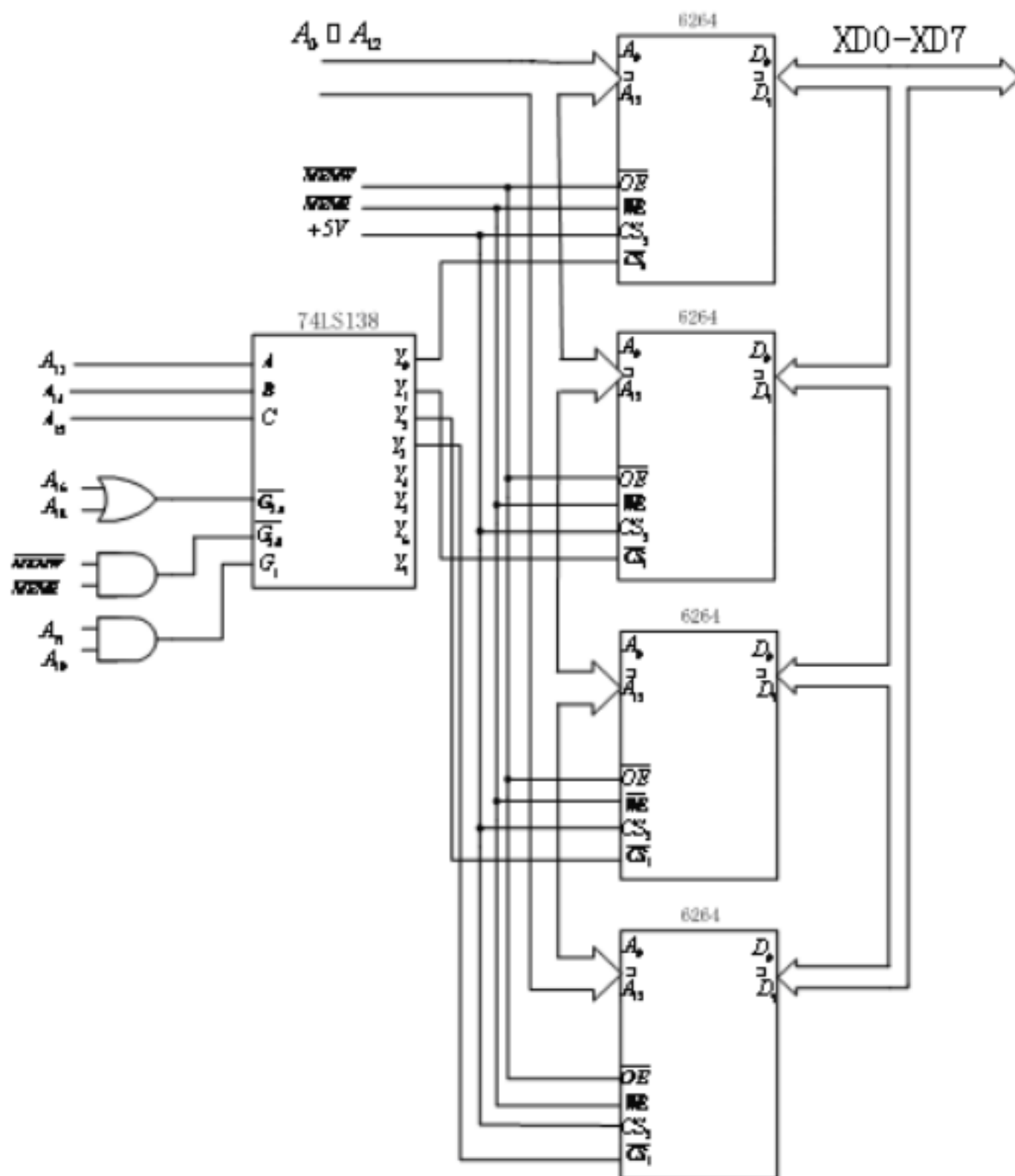


图 6.2 (b) 板内存储器电路的连接图

11. 若在某 8088 微型计算机系统中，要将一块 2764 芯片连接到 E0000H~E7FFFH 的空间中去，利用局部译码方式使它占有整个 32kB 的空间，试画出地址译码电路及 2764 芯片与总线的连接图。

解：Intel 2764 的片容量为 8KB，而题目给出的地址共 32KB，说明有 4 个地址区重叠，即采用部分地址译码时，有 2 条高位地址线不参加译码（即 A_{13} ， A_{14} 不参加译码）。

地址译码电路及 2764 与总线的连接如图 6.3 所示。

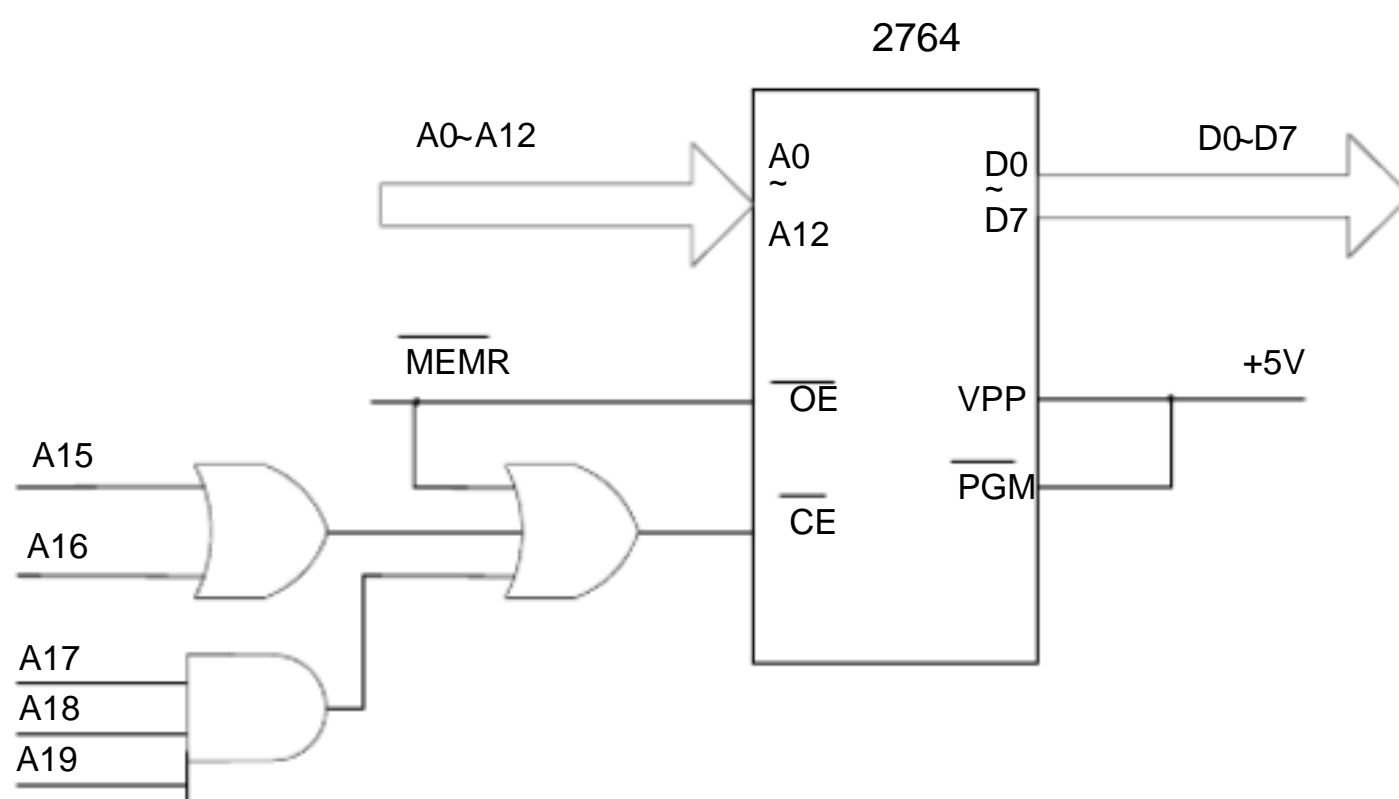


图 6.3 地址译码电路及 2764 与总线的连接

12. 在 8086 CPU 工作在最小方式组成的微机系统中，扩充设计 16kB 的 SRAM 电路，SRAM 芯片选用 Intel 6264，内存地址范围为 70000H~73FFFH，试画出此 SRAM 电路与 8086 系统总线的连接图。

解：73FFFH-70000H+1=4000H=16K

Intel 6264 的片容量为 8KB，RAM 存储区总容量为 16KB，故需要 2 片 6264。

8086 最小方式系统与存储器读写操作有关的信号线有：地址总线 $A_0 : A_{19}$ ，数据总线：

$D_0 : D_{15}$ ，控制信号： $\overline{M}/\overline{IO}$ ， \overline{RD} ， \overline{WR} ， \overline{BH} E。

此 SRAM 电路与 8086 系统总线的连接图如图 6.4 所示。

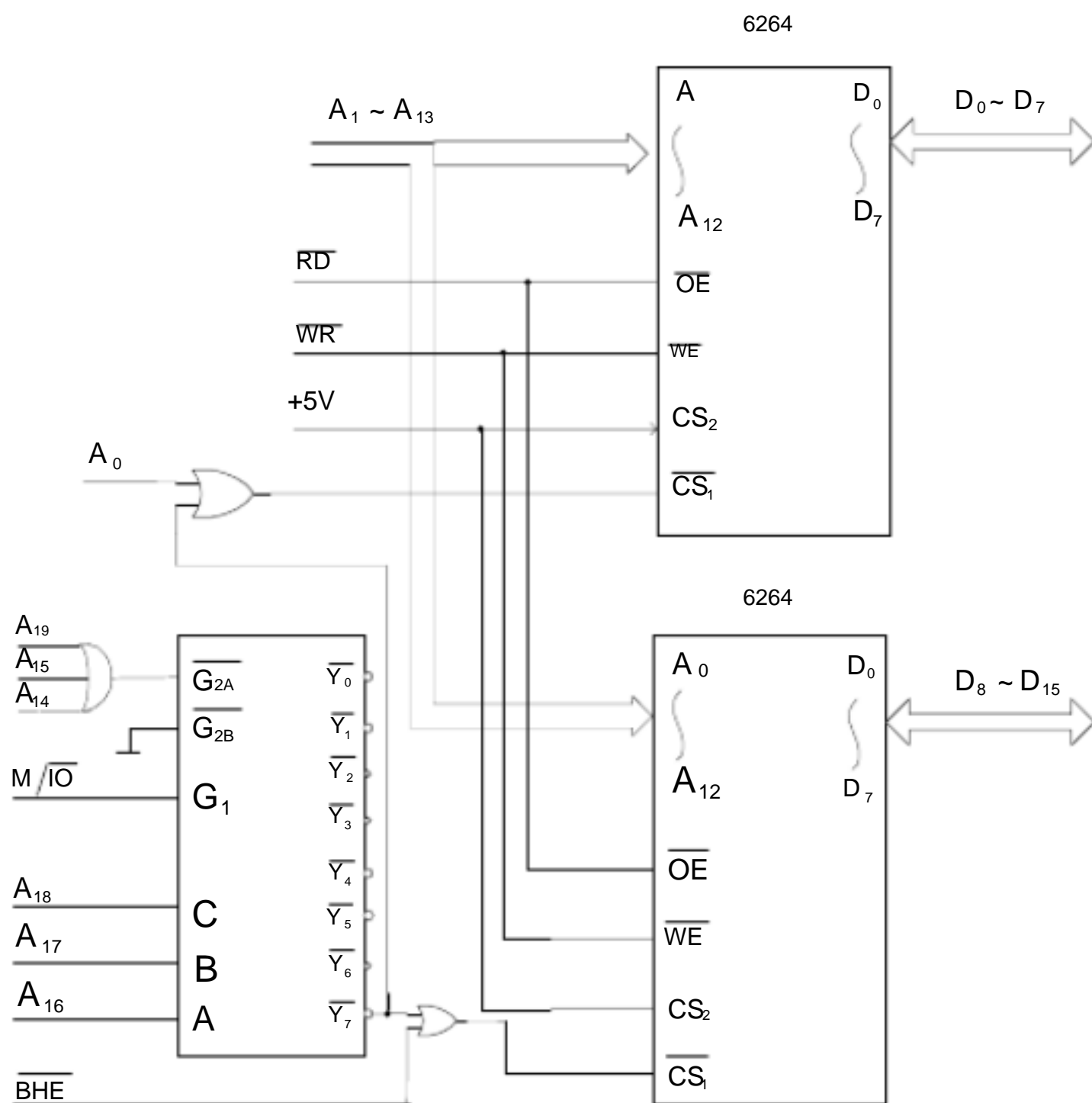


图 6.4 SRAM 电路与 8086 系统总线的连接图

13. E^2 PROM 28C16 芯片各引脚的功能是什么？如果要将一片 28C16 与 8088 系统总线相连接，并能随时改写 28C16 中各单元的内容，试画出 28C16 和 8088 系统总线的连接图（地址空间为 40000H~407FFH）。

解：28C16 的引脚功能：

VCC, GND 电源和地

$A_{10} \sim A_0$ ：11 位地址线，可寻址 2KB 地址空间

$D_7 \sim D_0$ ：8 位数据线

\overline{WE} ：写允许，低电平有效。

\overline{OE} ：输出允许，低电平有效。

\overline{CE} ：片选信号，低电平有效。

根据所学知识，28C16 与 8088 系统的连接图如图 6.5 所示。

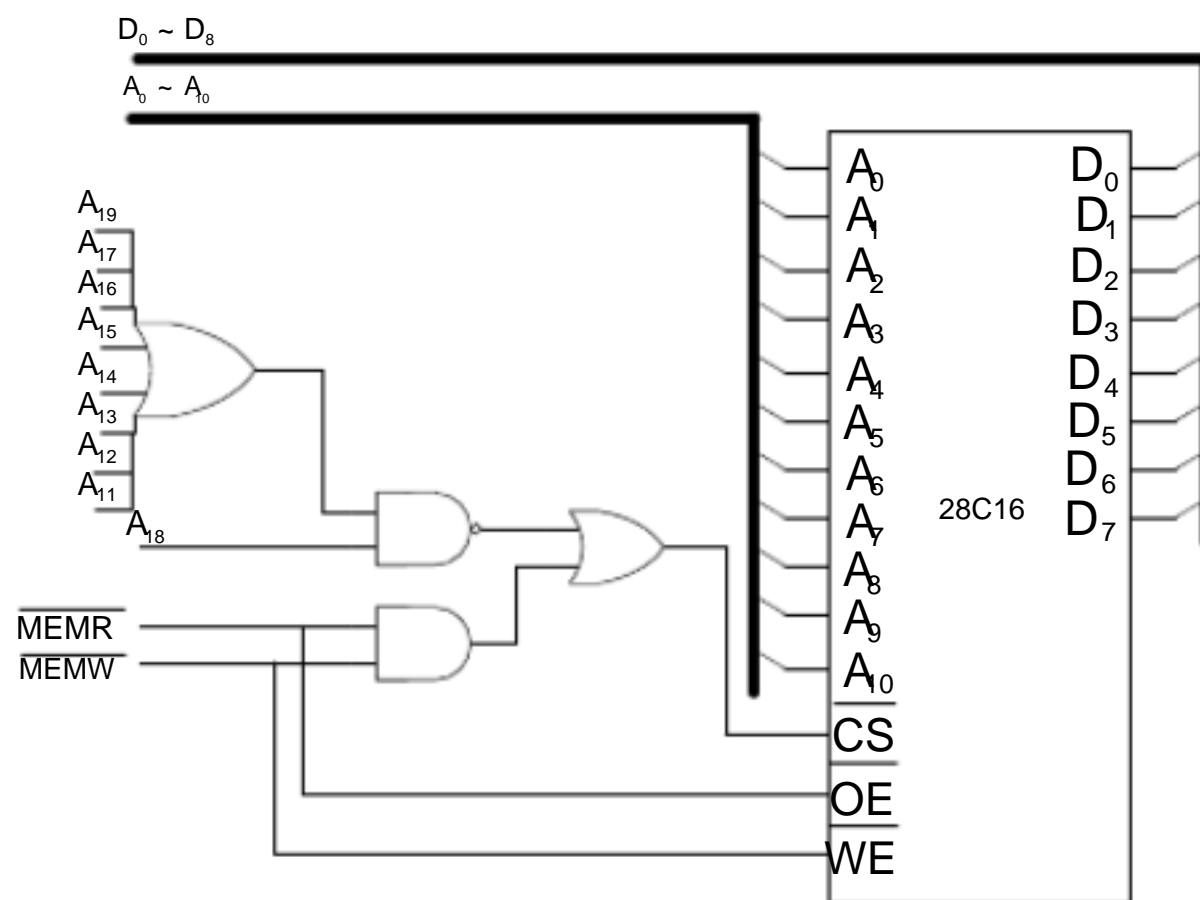


图 6.5 28C16 与 8088 系统的连接图

第 7 章 常用芯片的接口技术

1. 简述 I/O 接口的基本功能。

答：(1) 地址选择 (2) 控制功能 (3) 状态指示 (4) 速度匹配
(5) 转换信息格式 (6) 电平转换 (7) 可编程性

2. 简述 I/O 接口与 I/O 端口的区别。

答：I/O 接口是指 I/O 设备与系统总线之间的连接部件。

I/O 端口是指 I/O 接口内部可由 CPU 进行读写操作的各种寄存器，根据存放信息的不同，这些寄存器分别称为数据端口、控制端口和状态端口。

3. 简述 I/O 端口的编址方式及优缺点。

答：I/O 端口编址的方式可以分为独立编址和统一编址两种方式。

独立编址方式是指 I/O 端口与存储器有相互独立的地址空间。

统一编址方式是指 I/O 端口与存储器共享一个地址空间，所有的存储单元只占用其中的一部分地址，而 I/O 端口则占用另外一部分地址。

优缺点：独立编址方式的优点之一是存储器的容量可以达到与地址总线所决定的

地址空间相同；优点之二是访问 I/O 端口时的地址位数可以较少，提高总线的利用率。但是缺点是必须设置专门的 I/O 指令，增加了指令系统和有关硬件的复杂性。

与独立编址方式相比，统一编址方式的优点是无需专门的 I/O 指令，从而使编程较灵活，但是 I/O 端口占用了存储器的一部分地址空间，因而影响到系统中的存储器的容量，并且访问存储器和访问 I/O 端口必须使用相同位数的地址，使指令地址码加长，总线中传送信息量增加。

4. 简述程序查询、中断和 DMA 三种方式的优缺点。

答：程序查询方式的优点在于可以防止数据的丢失，实现起来较为简单；缺点是它占用了微处理器的大量时间，实时性较差。

中断方式具有较好的实时性；但在一定程度上增加成本和复杂性。

DMA 方式的突出优点是传送过程无须处理器的控制，数据也无须经过微处理器，而是直接在 I/O 设备与主存储器间进行，因此既节约了微处理器的时间，也使传送速率大大提高；缺点是输入/输出操作占用微处理器时间，而且很难达到较高的数据传输率。

5. 8086 CPU 有 20 条地址总线，可形成 1MB 的存储器地址空间，可寻址范围为 00000H--FFFFFFH; 地址总线中的 16 条线可用于 I/O 寻址，形成 64KB 的输入输出地址空间，地址范围为 0000H--FFFFH; PC 机中用了 10 条地址线进行 I/O 操作，其地址空间为 1KB，可寻址范围为 000H—3FFH。
6. 对于微机而言，任何新增的外部设备，最终总是要通过 I/O 接口 与主机相接。
7. 在主机板外开发一些新的外设接口逻辑，这些接口逻辑的一侧应与 I/O 设备 相接，另一侧与 系统总线 相接。
8. 需要靠在程序中排入 I/O 指令完成的数据输入输出方式有 BC。
(A) DMA (B) 程序查询方式 (C) 中断方式
9. 8086CPU用 IN 指令从端口读入数据，用 OUT 指令向端口写入数据。
10. 在 8088 CPU组成的计算机系统中有一接口模块，片内占用 16 个端口地址 300~30FH，设计产生片选信号的译码电路。

解：由于片内有 16 个端口，非别占用 300~30FH 地址。因此，该接口模块的片选信号的译码电路设计时，A3~A0 不参加译码。其译码电路如图 7.1 所示。

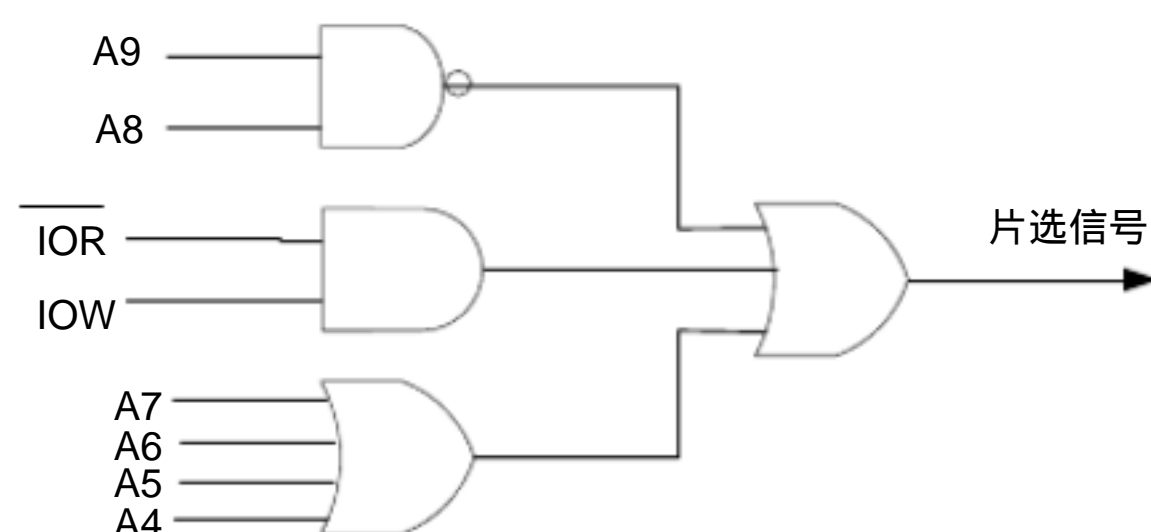


图 7.1

11. 在 IBM PC 系统中，如果 AEN 信号未参加 I/O 端口地址译码，会出现什么问题？
在没有 DMA 机构的其它微机系统中，是否存在同样的问题？
答：在 IBM PC 系统中，如果 AEN 信号未参加 I/O 端口地址译码，则会出现 DMA 机构与 I/O 端口竞争总线的问题。在没有 DMA 机构的其他微机系统中，不会存在同样的问题。
12. 在 8088 CPU 工作在最大方式组成的微机系统中，利用 74LS244 设计一个输入端口，分配给该端口的地址为 04E5H，试画出连接图。

解：连接图如图 7.2 所示。

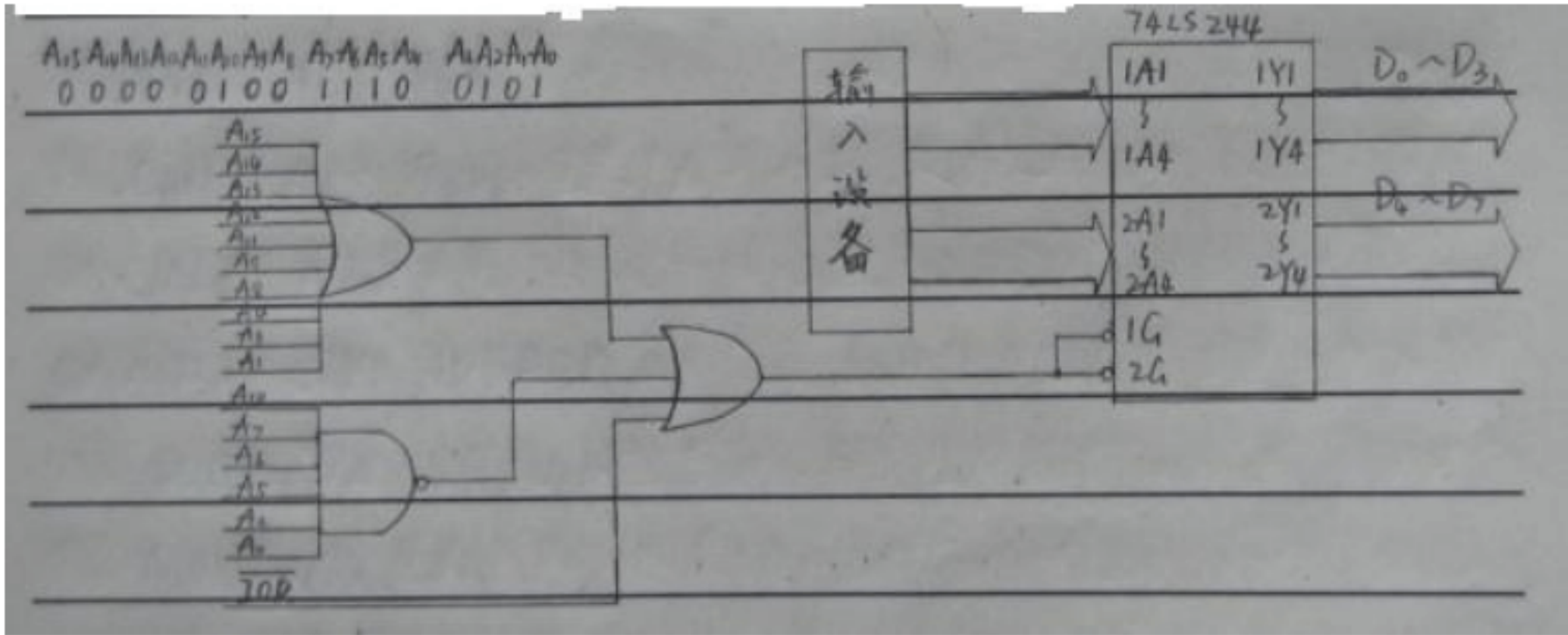


图 7.2

13. 在上题的基础上，利用 74LS374 设计一个输出端口，分配给该端口的地址为 E504H，试画出连接图。若上题中输入端口的 bit3、bit4 和 bit7 同时为 1，将内存 BUFFER 开始的连续 10 个字节单元的数据由 E504H 端口输出；若不满足条件，则等待。试编写程序。

解：连接图如图 7.3 所示。

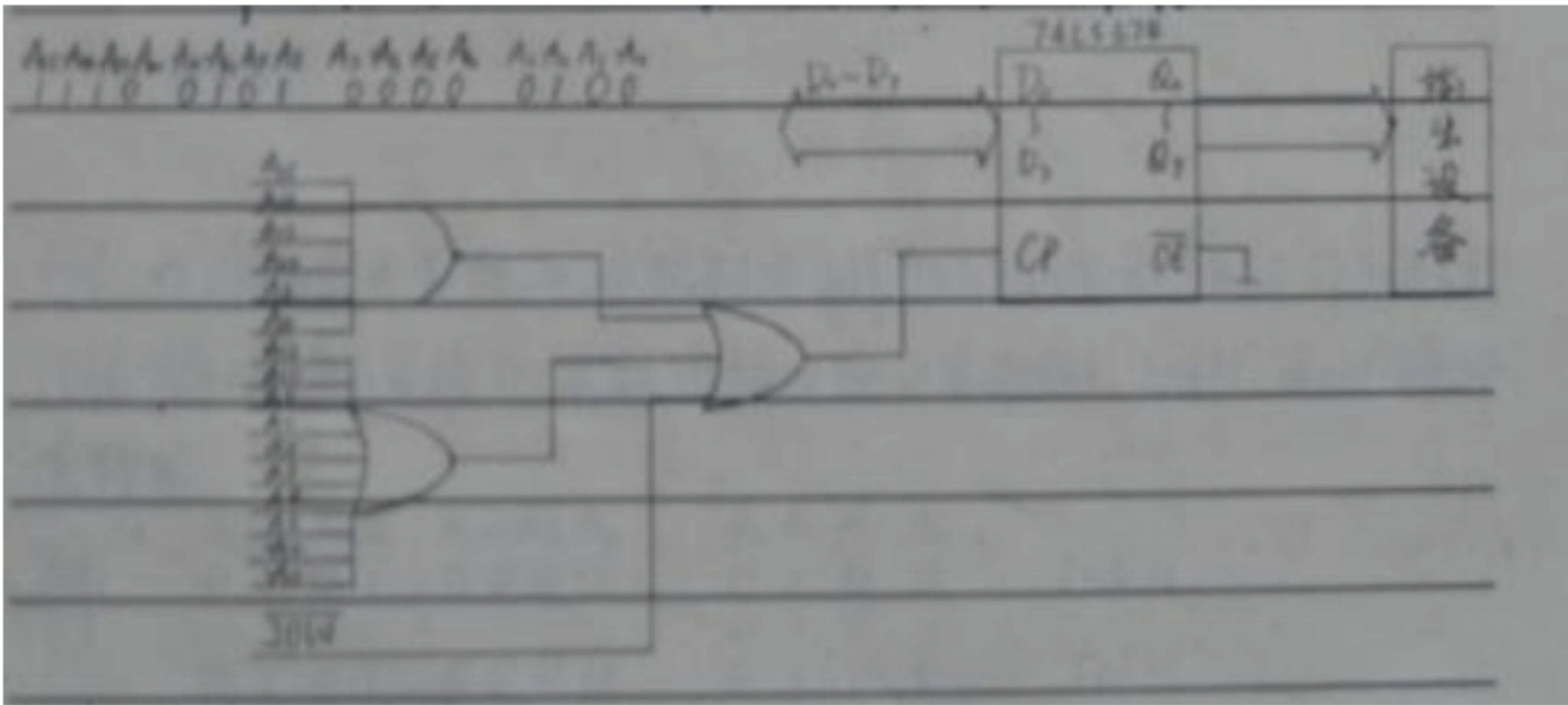


图 7.3

```
程序如下：

MOV    CX,10

LEA    SI,BUFFER

MOV    DX,04E5H
```

```

WAIT1 : IN    AL,DX

        AND    AL,98H

        CMP    AL,98H

        JNZ    WAIT1

        MOV DX,0E504H

L1 : MOV    AL,[SI]

        OUT    DX,AL

        INC    SI

        LOOP   L1

        HLT

```

14 . 在 8086 最大系统中，分别利用 2 片 74LS244 和 74LS273 设计 16 位输入和输出接口，其起始端口地址为 504H、506H，画出硬件连接图

解：硬件连接图如图 7.4 所示。

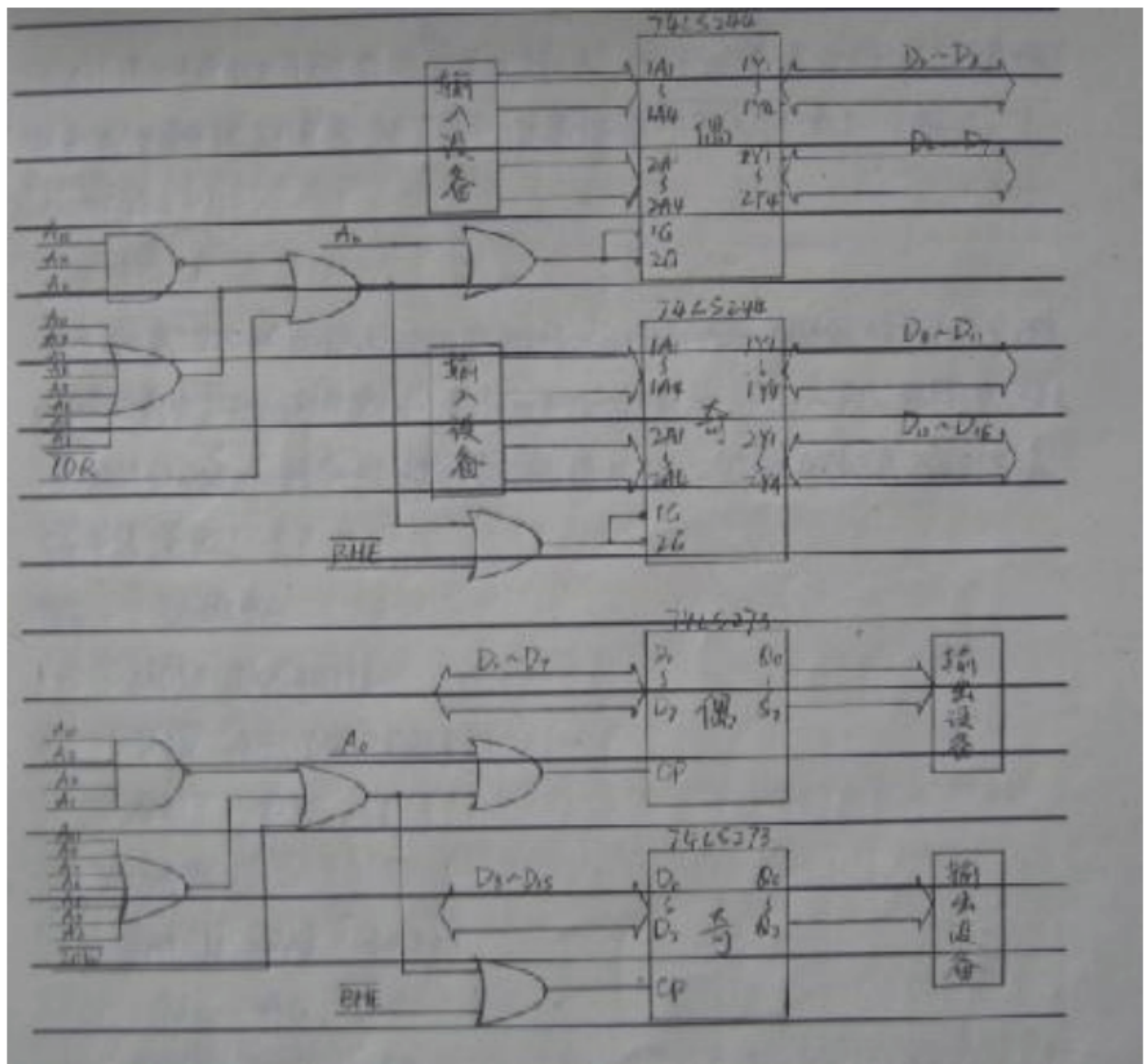


图 7.4 硬件连接图

第 8 章 中断系统与可编程中断控制器 8259A

1. 什么叫中断？8086 微机系统中有哪几种不同类型的中断？

答：在 CPU 执行程序的过程中，由于某个事件的发生，CPU 暂停当前正在执行的程序，转去执行处理该事件的一个中断服务程序，待中断服务程序执行完成后，CPU 再返回到原被中断的程序继续执行。这个过程称为中断。

8086 微机系统中有 3 种中断：

- 1) 外部可屏蔽中断。
- 2) 外部不可屏蔽中断。
- 3) 内部中断

2. 什么是中断类型？它有什么用处？

答：通常用若干位二进制编码来给中断源编号，该编号称为中断类型号。8086 微处理器用 8 位二进制码表示一个中断类型，有 256 个不同的中断。这些中断可以划分为内部中断、外部不可屏蔽中断、外部可屏蔽中断三类。

用处：使 CPU 识别中断源，从而能正确地转向该中断源对应的中断服务程序入口。

3. 什么是中断嵌套？使用中断嵌套有什么好处？对于可屏蔽中断，实现中断嵌套的条件是什么？

答：微处理器在处理低级别中断的过程中，如果出现了级别高的中断请求，微处理器停止执行低级中断的处理程序而去优先处理高级中断，等高级中断处理完毕后，再接着执行低级的未处理完的程序，这种中断处理方式成为中断嵌套。

使用中断嵌套的好处是能够提高中断响应的实时性。对于某些对实时性要求较高的操作，必须赋予较高的优先级和采取中断嵌套的方式，才能保证系统能够及时响应该中断请求。

对于可屏蔽中断，实现中断嵌套的条件有：（1）微处理器处于中断允许状态（IF=1）（2）中断请求的优先级高于正在执行的中断处理程序的优先级。（3）中断请求未被 8259 屏蔽。（4）没有不可屏蔽中断请求和总线请求。

4. 什么是中断向量？中断类型号为 1FH 的中断向量为 2345H:1234H，画图说明它在中断向量表中的存放位置。

答：中断向量为每个中断服务子程序的入口地址，为 32 位（16 位的偏移地址和 16 位的段地址），在中断向量表中占用 4 个地址单元。在 8086CPU 组成的计算机系统中，采用最低的 1024 个地址单元（称为 0 页）来存储中断向量。这 1024 个地址单元成为中断向量表。

因此，中断类型号位 1FH 的中断向量，在中断向量表中的存储位置为 $1FH \times 4 = 07CH$ 。
 中断类型号为 1FH 的中断向量在中断向量表中的存放位置如图 8.1 所示。

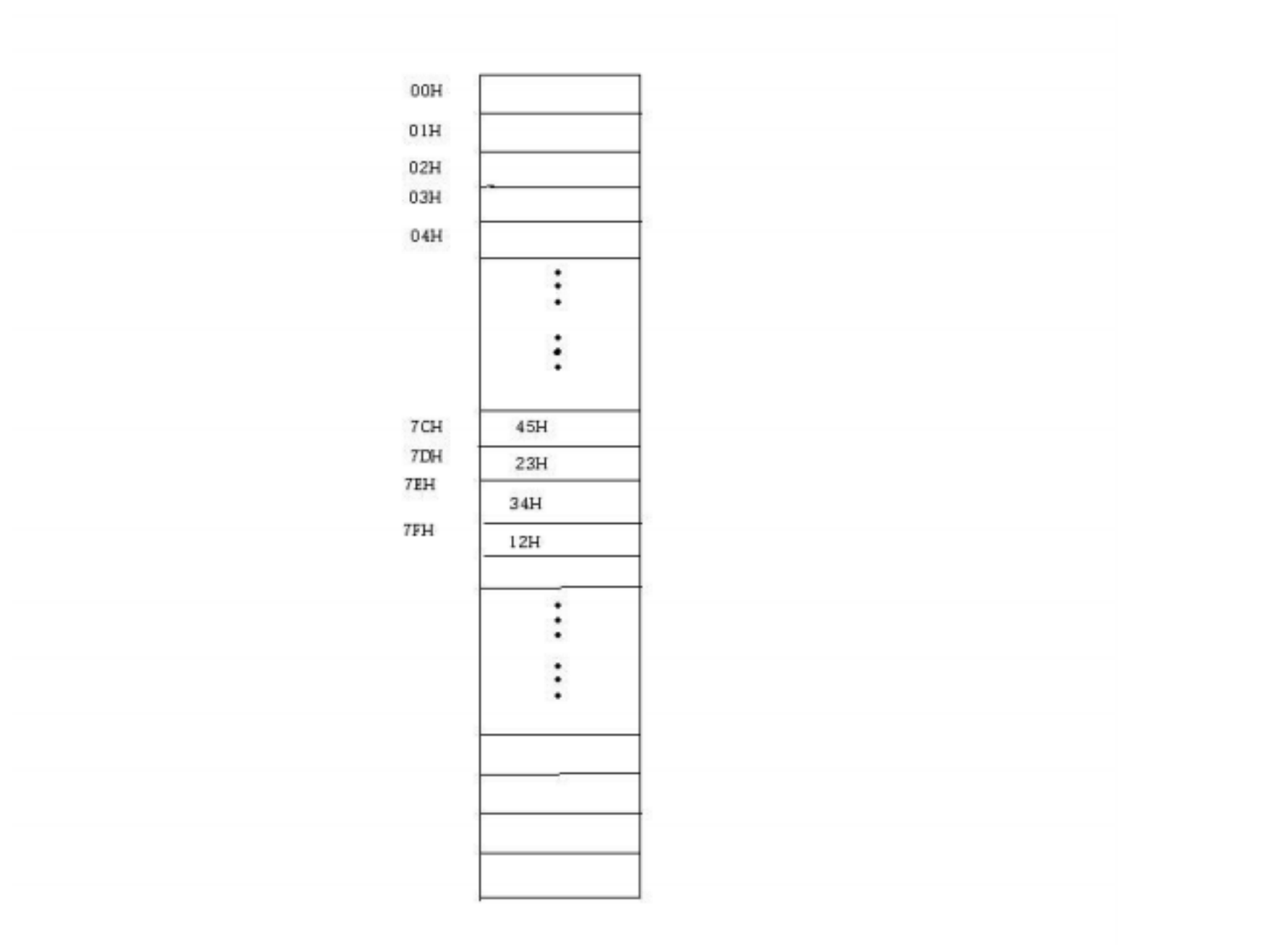


图 8.1 中断类型号为 1FH 的中断向量在中断向量表中的存放位置

5. 中断向量表的功能是什么？叙述 CPU 利用中断向量表转入中断服务程序的过程。
 答：中断向量表是用来存放中断向量的。 是中断类型号与它对应的中断服务程序入口地址之间的换算表。

- 1) CPU获得中断类型号 n 后，将当前 PSW CS和 IP 的内容依次压入堆栈，保存断点的状态和断点地址，以便返回时恢复。
- 2) 将 PSW中的 IF 位和 TF 位清 0，关闭中断。
- 3) 把地址为 $4 \times n$ 和 $4 \times n+1$ 两个单元的 16 位数作为中断服务程序入口的偏移地址置入 IP，把地址为 $4 \times n+2$ 和 $4 \times n+3$ 两个单元的 16 位数作为中断服务程序入口的段地址置入 CS
- 4) 转入中断服务程序。

6. 叙述可屏蔽中断的响应过程。

答：在 IF 位为 1 情况下，从 INTR 端加入中断请求信号开始， 到进入中断服务程序为止

所经过的一系列操作，称为可屏蔽中断的响应过程。首先等待当前指令结束后，进入中断响应周期。然后，从微处理器外部的中断控制逻辑获得中断类型号。随后把当前的 PSW、CS 和 IP 的内容依次压入堆栈，接着清除 PSW 中的 IF 位和 TF 位为 0。最后把中断服务程序的入口地址置入 IP 和 CS。至此，完成了可屏蔽中断的响应过程，开始进入中断服务程序。

7. 简要叙述 8259A 内部 IRR、IMR、ISR 三个寄存器的作用。

答：IRR：中断请求寄存器，用来锁存外部设备送来的 $IR_7 \sim IR_0$ 中断请求信号。当外部中断请求线 IR_i 有中断请求时，IRR 中与之对应的第 i 位被置 1。该寄存器内容可以被微处理器读出。

IMR：用于设置中断请求的屏蔽信号。此寄存器第 i 位被置 1 时，与之对应的外部中断请求线 IR_i 被屏蔽，不能向微处理器发出 INT 信号。

ISR：用于记录当前正在被服务的所有中断级，包括尚未服务完而中途被更高优先级打断的中断级。若微处理器响应了 IR_i 中断请求，则 ISR 中与之对应的第 i 位置 1。中断处理结束前，要用指令清除这一位。

8. 中断控制器 8259A 的初始化编程是如何开始的？

答：初始化编程就是指对 ICW（初始化命令字）的初始化，对 ICW 的初始化有一定的顺序，必须从 ICW1 开始。对中断控制器 8259A 的初始化编程次序如图 8.2 所示。

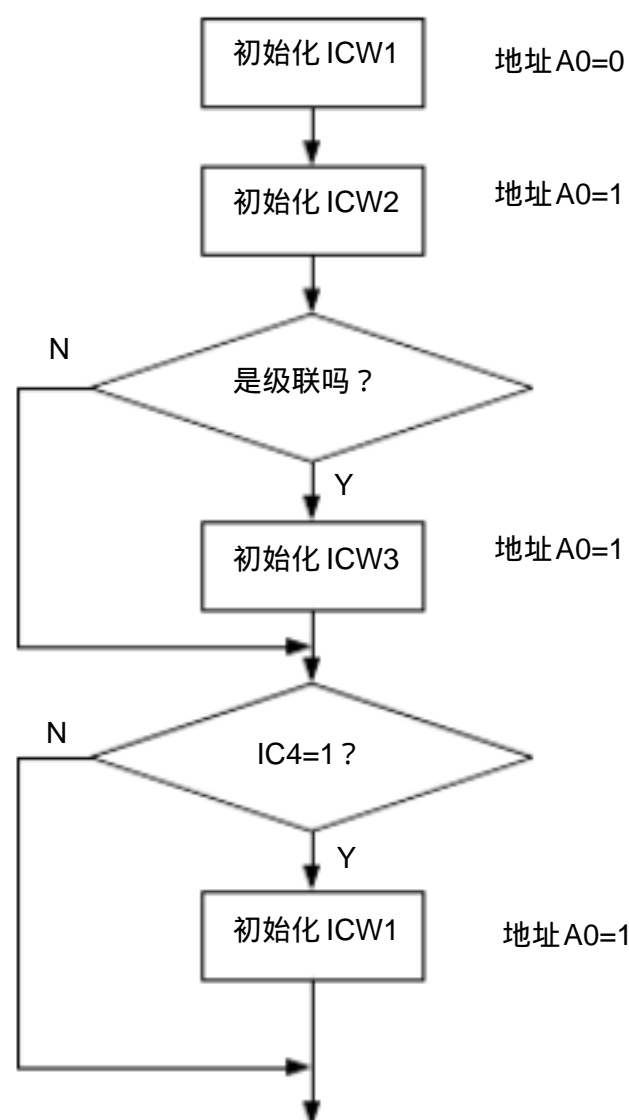


图 8.2 对中断控制器 8259A 的初始化编程次序

9. 设某微机系统需要管理 64 级中断，问组成该中断机构时需 _____ 片 8259A。

解：一片 8259 可控制 8 级中断，第 1 片通过级联 8 片 8259 可将中断扩充至 64 级。所以组成该中断机构共需 9 片 8259。

10. 完全嵌套的优先级排序方式的规则是什么？如何设置这种方式？

答：在全嵌套方式下，中断优先级的级别是固定的，即 IR_0 优先级最高， $IR_1 \sim IR_6$ 逐级次之， IR_7 最低。如果对 8259A 进行初始化后没有设置其他优先级别，那么 8259A 就按全嵌套方式工作。

11. 如果设备 D1、D2、D3、D4、D5 按完全嵌套优先级排列规则。设备 D1 的优先级最高，D5 最低。在下列中断请求下，给出各设备的中断处理程序的次序（假设所有的中断处理程序开始后就有 STI 指令，并在中断返回之前发出结束命令）

（1）设备 3 和 4 同时发出中断请求；

（2）设备 3 和 4 同时发出中断请求，并在设备 3 的中断处理程序完成之前，设备 2 发出中断请求；

（3）设备 1、3、5 同时发出中断请求，在设备 3 的中断处理程序完成之前，设备 2 发出中断请求。

解：

- (1) 设备 3 和 4 同时发出中断请求；
- 中断处理次序为： D3 D4 ，示意图如图 8.3 (a) 所示。

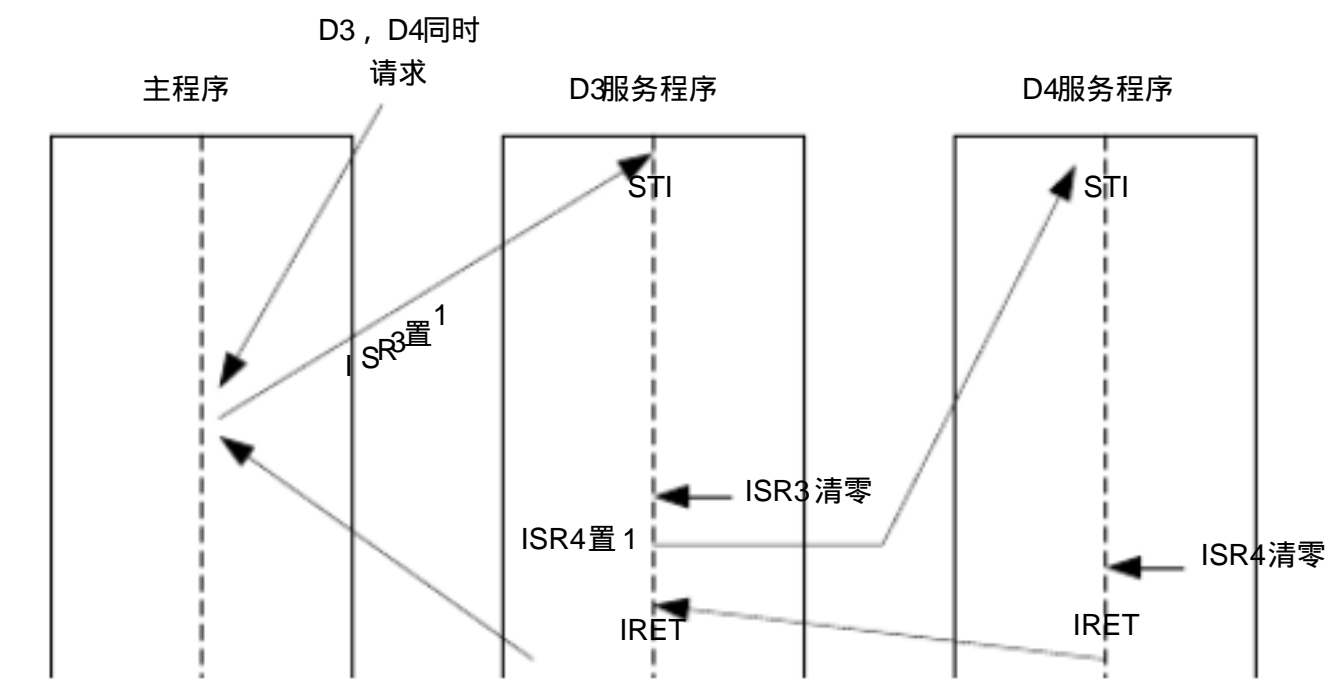


图 8.3 (a) 设备 3 和 4 同时发出中断请求的中断处理程序的次序示意图

- (2) 设备 3 和 4 同时发出中断请求，并在设备 3 的中断处理程序完成之前，设备 2 发出中断请求；
- 中断处理次序为： D3 D2 D3 D4 ，示意图如图 8.3 (b) 所示。

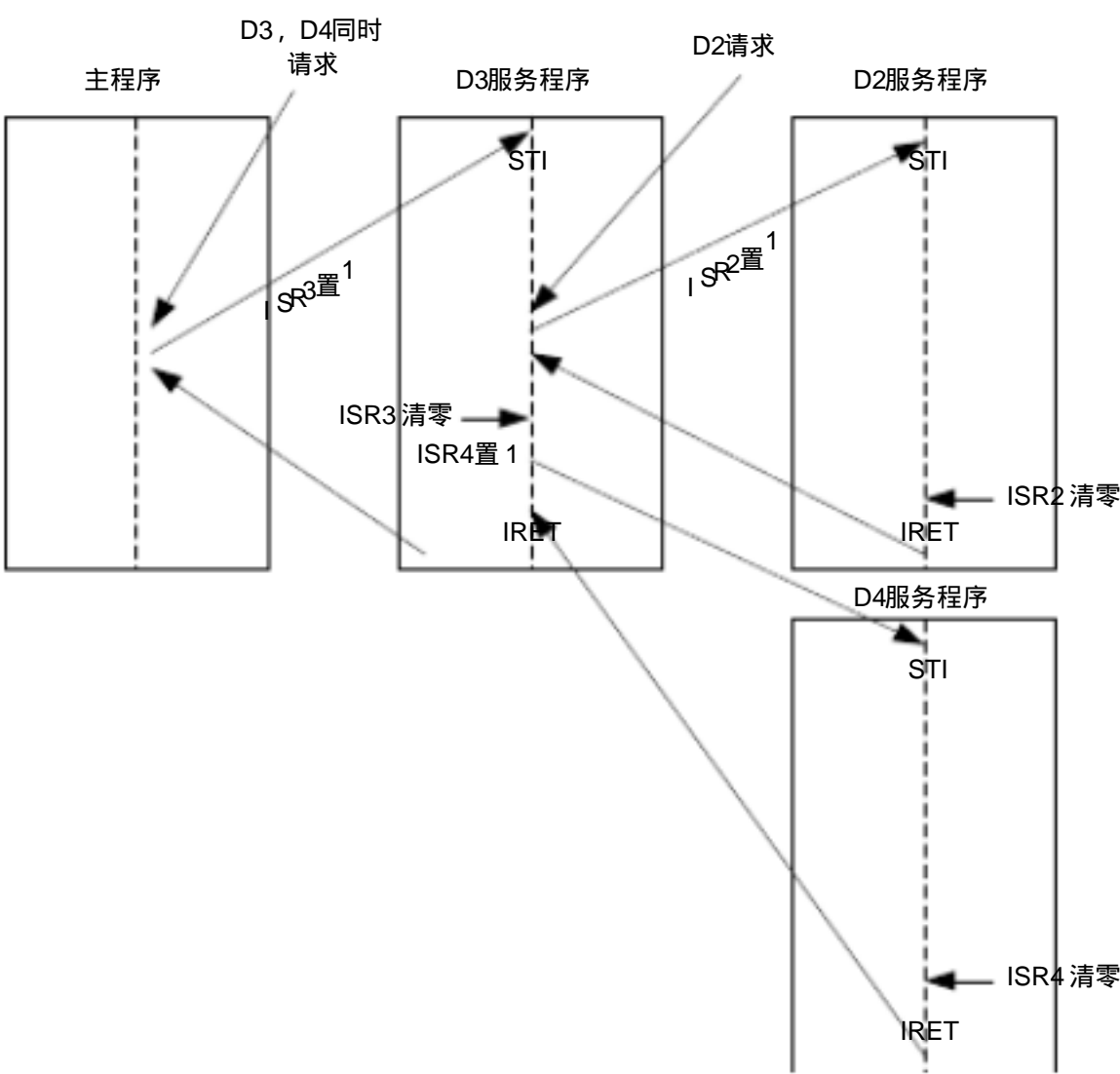


图 8.3 (b) 中断处理程序的次序示意图

- (3) 设备 1、3、5 同时发出中断请求，在设备 3 的中断处理程序完成之前，设备 2 发出

中断请求。

中断处理次序为： D1 D3 D2 D3 D5，示意图如图 8.3（c）所示。

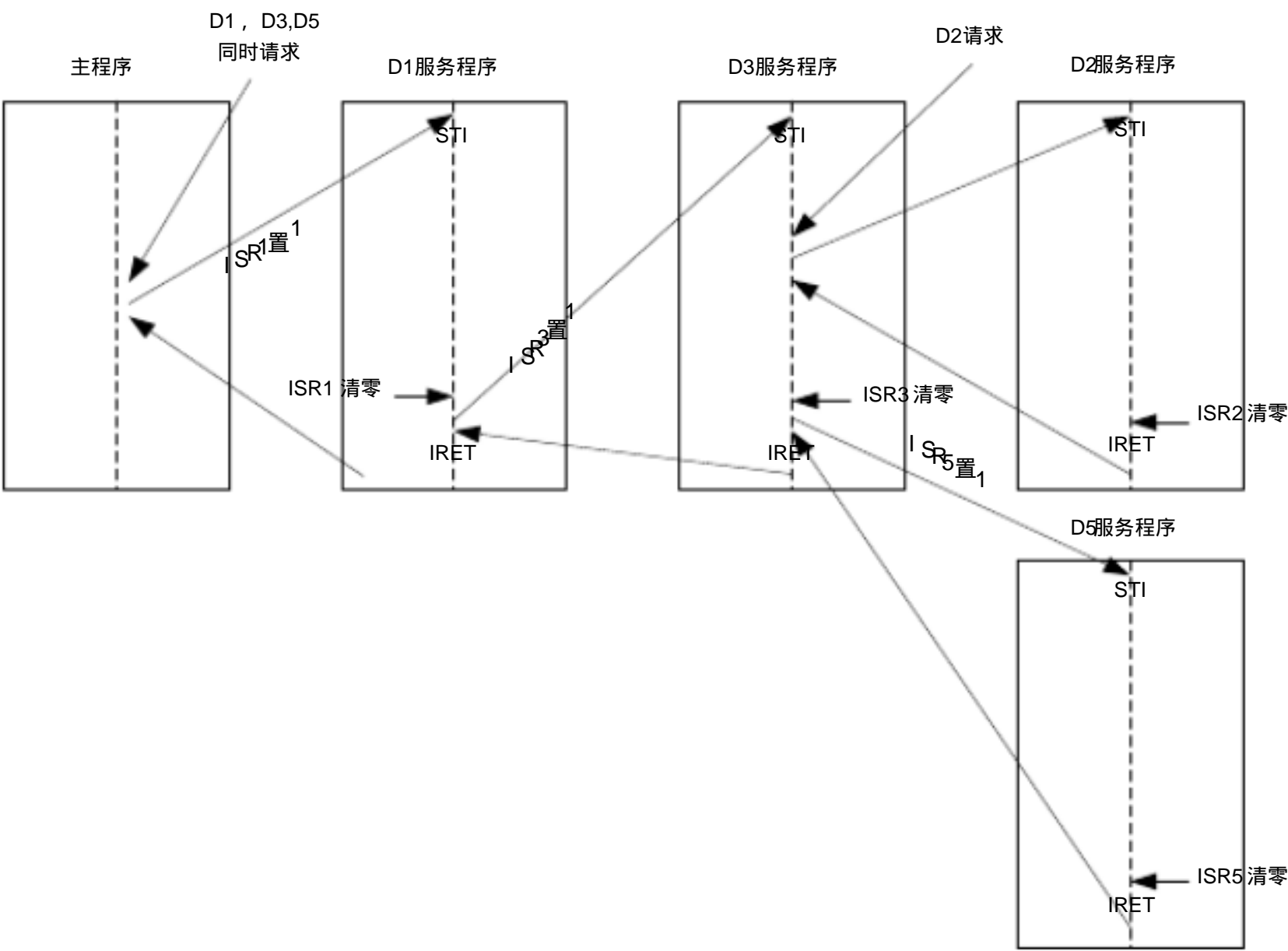


图 8.3（c）中断处理程序的次序示意图

12. 8259A 是怎样进行中断优先权管理的？

答：8259A 利用优先权处理器来识别和管理中断请求信号的优先级别。当几个中断请求信号同时出现时，优先权处理器根据控制逻辑规定的优先权级别和 IMR 的内容来判断这些请求信号的最高优先权。微处理器响应中断请求时，把优先权最高的 IRR 中的“1”送入 ISR。当 8259A 正在为某一级中断服务时，若又出现新的中断请求，则由优先权处理器判断新出现的中断请求的级别是否高于正在处理的那一级。若是，则进入多重中断处理。

13. 特殊全嵌套方式有什么特点？它的使用场合是什么？

答：多片 8259 级联时，主片必须工作在特殊全嵌套方式下。

假设从片工作在全嵌套方式下，先后收到了两次中断请求，而且第二次中断请求有较高的优先权，那么该从片就会两次通过 INT 引脚向上一级申请中断。如果主片采用全嵌套方式，则它不会响应来自同一引脚的第二次中断请求。而采用特殊全嵌套方式后，就会响应该请求。

14. 向 8259A 发送“中断结束”命令有什么作用？8259A 有哪几种中断结束方式？分析各自

的利弊。

答：向 8259A 发送“中断结束”命令，使相应的中断级在 ISR 中的相应位清“0”，表示该中断处理已经结束。

8259A 有自动中断结束方式（ AEOI ）和非自动中断结束方式（ EOI ）。

（1）自动中断结束方式（ AEOI ）。

在这种方式下，系统一旦进入中断响应， 8259A 就在第二个中断响应周期 $\overline{\text{INTA}}$ 信号的后沿，自动将 ISR 中被响应中断级的对应位清“0”。这是一种最简单的中断结束处理方式，可以通过初始化命令来设定，但这种方式只能用在系统中只有一个 8259A，且多个中断不会嵌套的情况。

（2）非自动中断结束方式（ EOI ）。

在这种工作方式下，从中断服务程序返回前，必须在程序里向 8259A 输出一个中断结束命令（ EOI ），把 ISR 对应位清“0”。具体做法有一般的中断结束方式和特殊的中断结束 EOI 命令。这种方式可用于系统中有多个 8259A 或多个中断中有嵌套的情况，但这种方式要比自动中断结束方式复杂。而且在这种方式下，如果在程序里忘记了将 ISR 对应位清零，那么， 8259A 在一般情况下将不再响应这个中断以及比它级别低的中断请求。

15. 初始化 8259A 时设置为非自动结束方式，则在中断服务程序即将结束时必须设置什么操作命令？不设置这种命令会发生什么现象？如果初始化时设置为自动结束方式时，还需要设置这种操作吗？

答：必须发送中断结束命令，即向操作控制字 OCW_2 的 EOI 位写 1。如果不设置这种命令，则该中断服务程序结束后， CPU 将无法响应优先级比它低的中断请求以及它自身的下一次中断请求。若初始化为自动结束方式，则不需设置这种操作。

16. 在哪些情况下需用 CLI 指令关中断？在哪些情况下需用 STI 指令开中断？

答：

关于有前缀的指令，前缀包括段寄存器前缀（例如指令 MOV AX, CS:VAR ）、重复前缀（例如指令 REP MOVSB ）和 LOCK 前缀。应明确的是，指令前缀并不是一个独立的指令，只是指令的一部分，所以不允许在前缀码和指令码之间响应中断，所以应在这条指令前置 CLI（关中断）指令，在这条指令后置 STI（开中断）指令。

对于目的地址是段寄存器的 MOV 和 POP 指令（除 CS 外），本条指令后不允许响应中

断，而是还要执行一条指令结束后才能响应中断，这时可以在指令前加 CLI ，指令后加 STI 。

需要进行中断嵌套时，由于微处理器在响应中断时已将 IF 清零，所以一定要在中断处理程序中加开中断指令，才有可能进行中断嵌套。

17. 某系统中有两片 8259A，从片的请求信号连主片的 IR_2 引脚，设备 A 中断请求信号连从片 IR_5 引脚。说明设备 A 在一次 I/O 操作完成后通过两片 8259A 向 8086 申请中断，8086 微处理器通过两片 8259A 响应中断，进入设备 A 的中断服务程序，发送中断结束命令，返回断点的全过程。

答：设备 A 在一次 I/O 操作完成后在 8259A 从片中断请求输入端 IR_5 上产生中断请求，中断请求被锁存在 8259A 从片 IRR 中，并经 IMR “屏蔽”，其结果送给优先权电路判优。控制逻辑接收中断请求，向 8259A 主片 IR_2 引脚输出 INT 信号，中断请求被锁存在 8259A 主片 IRR 中，并经 IMR “屏蔽”，其结果送给优先权电路判优。控制逻辑接收中断请求，向 CPU 输出 INT 信号。CPU 从 INTR 引脚接受 8259A 主片的 INT 信号，进入连续两个 $\overline{\text{INTA}}$ 周期。优先权电路设置 ISR 中的对应位，在收到第一个 $\overline{\text{INTA}}$ 信号后，8259A 主片把当前申请中断的 8259A 从片的 ID 代码 010，通过 $\text{CAS}_0 \sim \text{CAS}_2$ 送到相应的 8259A 从片。相应地 8259A 从片在收到第二个 $\overline{\text{INTA}}$ 信号时，将中断类型号 N 送到数据线上。

8086CPU 获得中断类型号 N 后，在 $N \times 4$ 和 $N \times 4 + 2$ 对应的中断向量表单元获取中断向量分别置入 IP 和 CS ，从此进入设备 A 的中断服务程序。

在中断服务程序结束前（即执行 IRET 指令前），应分别向从片 8259 和主片 8259 发送 EOI 中断结束命令，然后执行 IRET 指令，返回断点。

18. 某 8086 系统用 3 片 8259A 级联构成中断系统，主片中断类型号从 10H 开始。从片的中断申请连主片的 IR_4 和 IR_6 引脚，它们的中断类型号分别从 20H、30H 开始。主、从片均采用上升沿触发，非自动中断结束方式。且主片采用特殊全嵌套方式，从片采用完全嵌套方式。请编写他们的初始化程序。

答：设 8258A 主片的端口地址为 20H($A_0=0$) 和 21H($A_1=1$)，第一个 8259A 从片的端口地址为 50H($A_0=0$) 和 51H($A_1=1$)，第二个 8259A 从片的端口地址为 0A0H($A_0=0$) 和 0A1H($A_1=1$)，初始化程序如下：

主片：

```
MOV    AL , 00010001B ; ICW1 : 边沿触发 , 级联

OUT    20H , AL

MOV    AL , 00010000B ; ICW2 : 中断类型号  10H~17H

OUT    21H , AL

MOV    AL , 01010000B ; ICW3 : IR4 和 IR6 连有从片

OUT    21H , AL

MOV    AL , 00010001B ; ICW4 : 特殊全嵌套 , 非缓冲 , 非自动中断结束

OUT    21H , AL
```

从片 1：

```
MOV    AL , 00010001B ; ICW1 : 边沿触发 , 级联

OUT    50H , AL

MOV    AL , 00100000B ; ICW2 : 中断类型号  20H~27H

OUT    51H , AL

MOV    AL , 00000100B ; ICW3 : INT 引脚连主片 IR4

OUT    51H , AL

MOV    AL , 00000001B ; ICW4 : 完全嵌套 , 非缓冲 , 非自动中断结束

OUT    51H , AL
```

从片二：

```
MOV    AL , 00010001B ; ICW1 : 边沿触发 , 级联

OUT    0A0H , AL

MOV    AL , 00110000B ; ICW2 : 中断类型号  30H~37H

OUT    0A1H , AL

MOV    AL , 00000110B ; ICW3 : INT 引脚连主片 IR6

OUT    0A1H , AL

MOV    AL , 00000001B ; ICW4 : 完全嵌套 , 非缓冲 , 非自动中断结束
```

OUT 0A1H , AL

19. 设 8259A 的端口地址为 50H ($A_0=0$) 和 51H ($A_1=1$) , 请给下面的 8259A 初始化程序加上注释, 说明各命令字的含义。

MOV AL , 13H

OUT 50H , AL

MOV AL , 08H

OUT 51H , AL

MOV AL , 0BH

OUT 51H , AL

解 :

MOV AL , 13H

OUT 50H , AL ; 初始化 ICW₁ , 设置为边沿触发 , 单片工作

MOV AL , 08H

OUT 51H , AL; ICW₂ ; 中断类型号的高 5 位为 00001B , 即中断类型号为 08H~0FH

MOV AL , 0BH

OUT 51H , AL ; 初始化 ICW₄ , 一般全嵌套缓冲方式 , 中断非自动结束

20. 设 8259A 端口地址为 20H 和 21H , 怎样发送清除 ISR₃ 的命令 ?

答 : 8259A 可通过设置操作控制字 错误!未找到引用源。 发送中断结束命令 (EOI 命令) , 来清除 ISR 的指定位。 错误!未找到引用源。 的值为 23H , 错误!未找到引用源。 用低位端口地址 (即 $A_0=0$) 进行操作 , 因此可编程如下 :

MOV AL,23H

MOV 20H,AL

21. 根据中断过程的要求设计的一个中断系统 , 大致需要考虑哪些问题 ?

答 : 1) 设计必要的中断源 , 确定它们提出的中断请求的方式。

2) 编写初始化程序。根据急迫程度的不同 , 规定好中断源的优先级别 , 以确定当几个中断源同时请求时 , 处理机能有一个先后响应次序。

3) 当处理机响应中断后 , 需要把被中断程序的现场、断点保护起来 , 以便中断处理结束后能返回原程序。

4) 中断服务程序设计。

5) 恢复现场，返回原程序。

22. 给定 (SP)=0100H, (SS)=0300H, (PSW)=0240H, 以及存储单元的内容 (00020H)=0040H, (00022H)=0100H, 在段地址为 0900H 及偏移地址为 00A0H 的单元中有一条中断指令 INT 8, 试问执行 INT 8 指令后, SP、SS、IP、PSW 的内容是什么? 栈顶的三个字是什么?

答: 执行 INT8 之后, (SP) =00FAH, (SS) =0300H, (IP) =0040H, (PSW) =0040H。

由于 INT8 为双字节指令, 所以栈顶的三个字从上到下分别是 00A2H, 0900H, 0240H。

23. 中断服务程序结束时, 用 RETF 指令代替 IRET 指令能否返回主程序? 这样做存在什么问题?

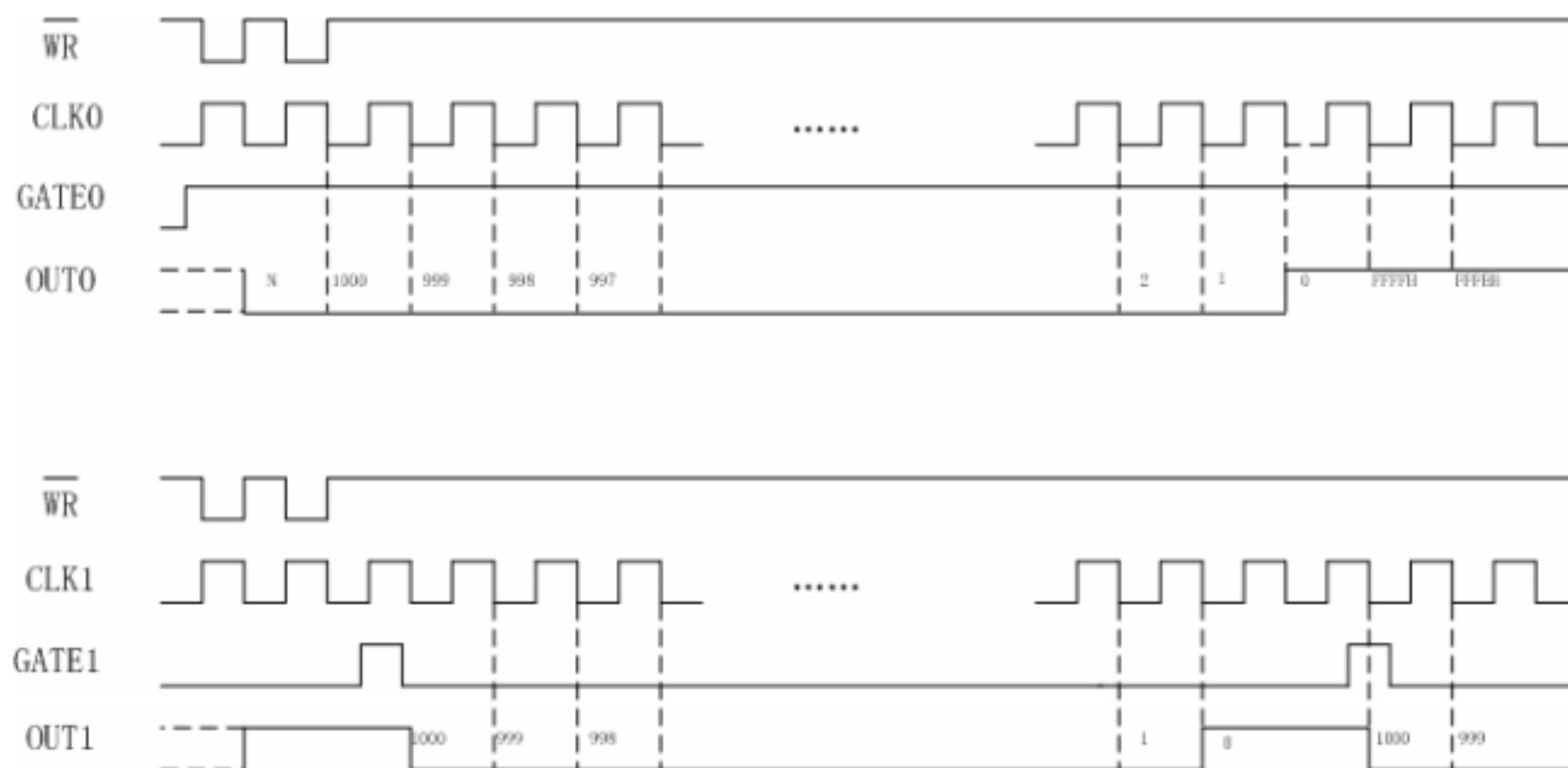
答: 可以。但这样做存在问题。IRET 指令执行时将出栈 IP, CS 和 PSW 的值, 而 RETF 指令只出栈 IP, CS 的值, 因而若中断服务程序中改变了标志位, 用 RETF 指令返回在主程序中将无法恢复标志位的值。尤其是在中断服务程序中关中断后 (IF=0), 用 RETF 指令将无法重新开中断, 从而导致中断服务结束后 CPU 不能再响应其他可屏蔽中断。

第 9 章 定时 / 计数器 8253 应用设计

1. 下列地址哪些能够分配给 8253/8254 的计数器 0? 为什么? (23H、54H、97H、51H、FCH、59H)

解：因为已经约定采用 A2,A1 作为 8253 的内部地址线，而且计数器 0 的地址为 00，所以在题中所给的地址中只有 51H,59H 的 A2 和 A1 同时为 0，即：A2A1=00.

2. 如果计数器 0 设定为方式 0，GATE0 = 1，CLK0 = 1MHz，时常数为 N = 1000，请画出 OUT0 的波形。如果计数器 1 设定为方式 1，其它参数与计数器 0 相同，画出 OUT1 的波形。



3. 编程实现：将 8253 计数器 0 设置成方式 4，并置时常数 10000，然后处于等待状态，直到 CE 的内容 1000 后再向下执行。

解：

```
MOV DX,COUNTD      ;写入计数器 0 的方式控制字
```

```
MOV AL,00111000B
```

```
OUT DX,AL
```

```
MOV DX,COUNTA      ;设置计数器 0 的常数
```

```
MOV AX,10000
```

```
OUT DX,AL
```

```
XCHG AL,AH
```

```
OUT DX,AL
```

```
L1: MOV DX,COUNTD      ;写入计数器 0 的方式控制字
```

```

MOV AL,0H

OUT DX,AL

MOV DX,COUNTA    ; 读入 CE

IN AL,DX

MOV AH,AL

IN AL,DX

XCHG AL,AH

CMP AX,1000      ; 判别 CE 当前大小

JA L1

```

4. 利用 8253 可以实现确定时间的延迟，编程实现延时 10 秒的程序段（设可以使用的基准时钟为 1MHz）。

解：本题使用计数器 0 和计数器 1，并且计数器 0 的输出 OUT0 作为计数器 1 的时钟输入 CLK1。

程序如下：

```

MOV    DX,COUNTD    ;写计数器 0 方式控制字

MOV    AL,00110100B

OUT    DX,AL

MOV    DX,COUNTA

MOV    AX,10000      ;写计数器 0 时常数，分频得到 100Hz 时钟频率

OUT    DX,AL

XCHG   AL,AH

OUT    DX,AL

MOV    DX,COUNTD    ; 写计数器 1 方式控制字

MOV    AL,01110000B

OUT    DX,AL

MOV    DX , COUNTB

MOV    AX,999        ; 分频得到 0.1Hz 时钟频率。（在方式 0 下，时常数为 N 时，

                        ; OUT 输出的低电平宽度为 N+1）。

OUT    DX,AL

```

```

XCHG    AL,AH

OUT      DX,AL

L1:                                     ; 延时

MOV      DX,COUNTD                    ; 当前 CE 的内容锁存到 OL

MOV      AL,01000000B

OUT      DX,AL

MOV      DX,COUNTB

IN        AL,DX

MOV      AH,AL

IN        AL,DX

XCHG     AL,AH

CMP      AX,999

JNA      L1                            ; 延时结束，则继续执行，否则，跳到 L1, 继续延时

....

```

5. 比较 8254 方式 0 与方式 4、方式 1 与方式 5 的区别？

方式 0 与方式 4

方式 0 OUT 端计数过程中为低，计数值减为 0 时，输出变高

方式 4 OUT 端计数过程中为高，计数值减为 0 时输出宽度为 1 个 CLK 的负脉冲

方式 1 与方式 5

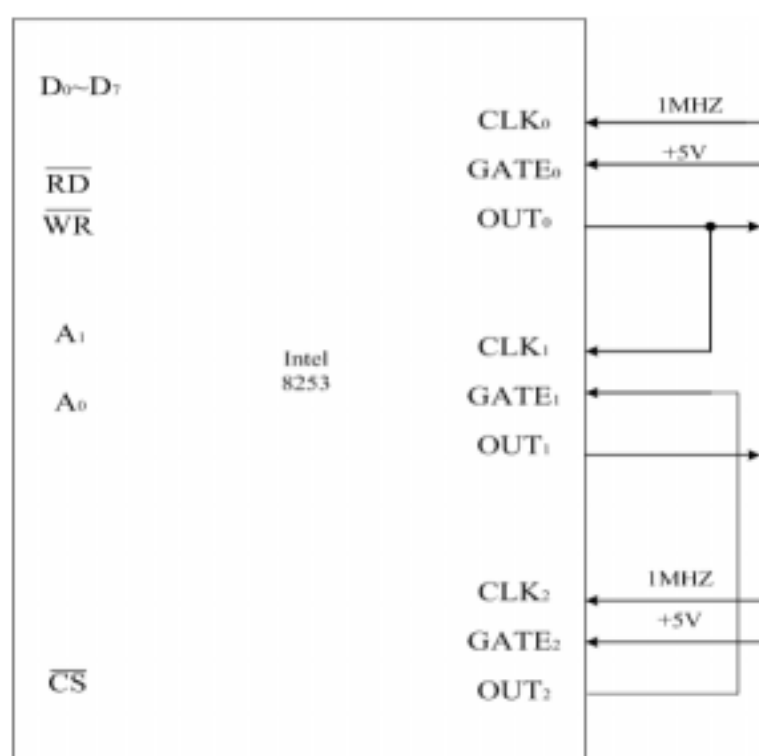
方式 1 OUT 端输出宽度为 n 个 CLK 的低电平，计数值减为 0 时，输出为高

方式 5 OUT 端计数过程中为高，计数值减为 0 时输出宽度为 1 个 CLK 的负脉冲

6. 在 8088 最小系统中，8253 的端口地址为 284H ~ 287H。系统提供的时钟为 1MHz，要求在 OUT0 输出周期为 20 微秒的方波，在 OUT1 输出周期为 200 微秒，其中每周期为负的时间是 180 微秒的信号。请编写 8253 的初始化程序。

解：

OUT0 输出为 20 微妙方波，可用方式三直接产生，OUT1 输出波形与书中例 9.2 类似，可用其中思想产生此信号。如果允许增加外部器件，也可在例 9.1 的基础上，将 OUT 端信号通过与非门取反，这样即可产生题目要求信号。本例利用例 9.1 思想解答



MOV DX,287H ;写计数器 0 控制方式字

MOV AL,00010110B

OUT DX,AL

MOV DX,284H ;写计数器 0 时常数

MOV AL,20

OUT DX,AL

MOV DX,287H ;写计数器 2 控制方式字

MOV AL,10010110B

OUT DX,AL

MOV DX,286H ;写计数器 2 时常数

MOV AL,200

OUT DX,AL

MOV DX,287H

MOV AL,01010010B ;写计数器 1 控制方式字

OUT DX,AL

MOV DX,285H

MOV AL,9 ;写计数器 1 时常数

OUT DX,AL

7. 通过 8253 计数器 0 的方式 0 产生中断请求信号， 现需要延迟产生中断的时刻， 可采用：

A) 在 OUT0变高之前重置初值；

B) 在 OUT0变高之前在 GATE0端加一负脉冲信号；

C) 降低加在 CLK0端的信号频率；

D) 以上全是。

解：D

A: 方式 0 下，在 OUT0变高之前重置初值，将在下一个 CLK的下降沿使时常数从 CR读入 CE并重新计数。

B: 在 OUT0变高之前在 GATE0端加一负脉冲信号可以延时一个时钟周期，达到延时的目的。

C: 降低加在 CLK0端的信号频率，可以增大时钟周期，达到延长 OUT0端低电平的时间。

(注：A 中，如果重置的初值为 1，则不会达到延时的效果)

8. 已知 8254 计数器 0 的端口地址为 40H, 控制字寄存器的端口地址为 43H, 计数时钟频率为 2MHz, 利用这一通道设计当计数到 0 时发出中断请求信号，其程序段如下，则中断请求信号的周期是 32.7675 ms。

```
MOV AL,00110010B
```

```
OUT 43H, AL
```

```
MOV AL, 0FFH
```

```
OUT 40H, AL
```

```
OUT 40H, AL
```

9. 若 8254 芯片可使用的 8086 端口地址为 D0D0H~ D0DFH, 试画出系统设计连接图。设加到 8254 上的时钟信号为 2MHz,

(1) 利用计数器 0~2 分别产生下列三种信号：

周期为 10us 的对称方波

每 1s 产生一个负脉冲

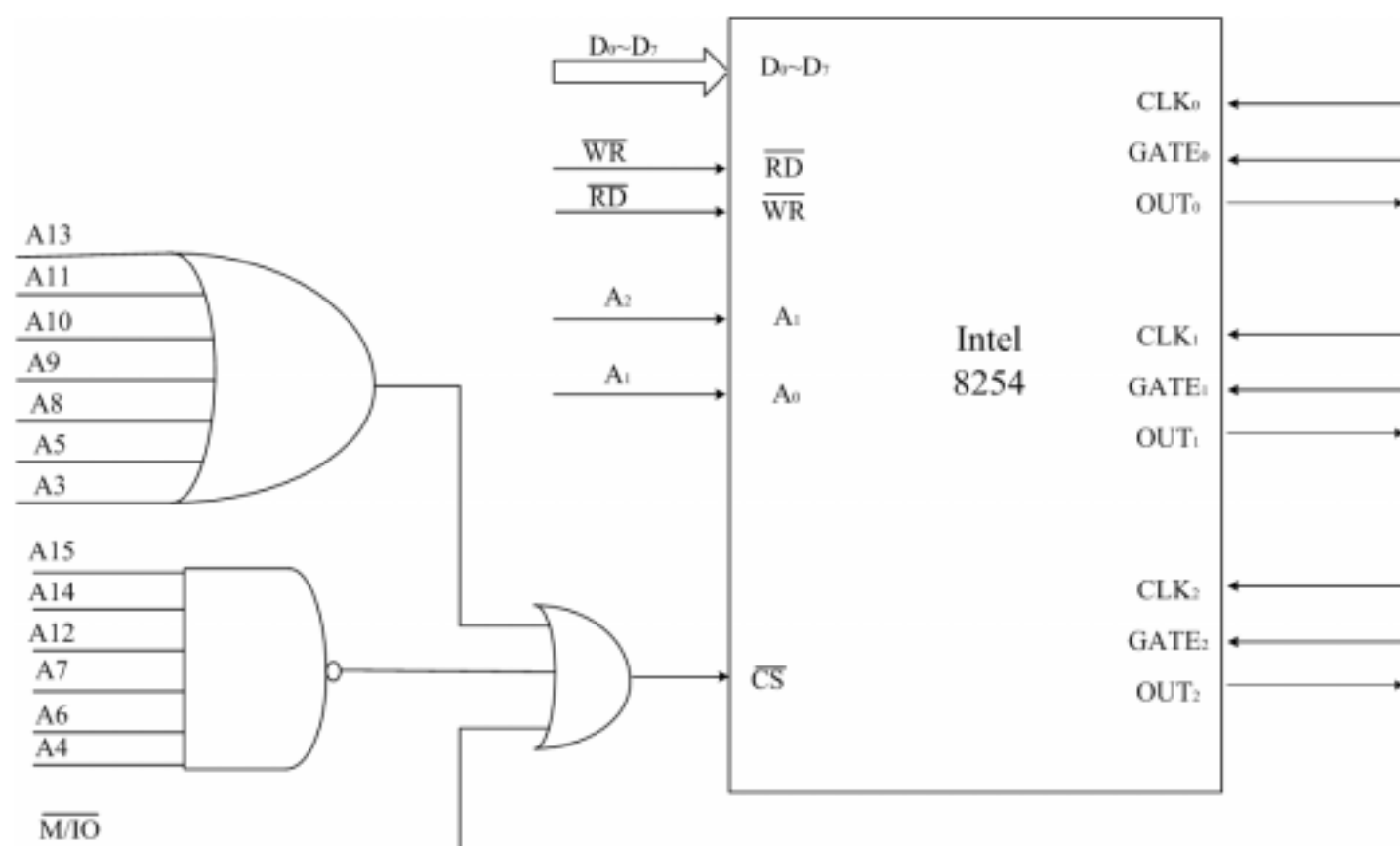
10s 后产生一个负脉冲

每种情况下，说明 8254 如何连接并编写包括初始化在内的程序段。

(2) 希望利用 8086 通过一专用接口控制 8253 的 GATE端，当 CPU使 GATE有效开始，20us 后在计数器 0 的 OUT端产生一个正脉冲，试设计完成此要求的硬件和软件。

解：

(1) 选用 D0D0H~D0DFH中的偶地址 D0D0,D0D2,D0D4,D0D6为基本地址作为 8254 的端口地址，设 8086 工作在最小方式下。8254 端口地址译码电路如下图：



计数器 0 输入端加 2MHz 的时钟信号，GATE₀ 加 +5V 电压，输出 OUT₀ 信号为周期为 10 μs 的对称方波。

初始化代码：

```
MOV DX,0D0D6H      ;写计数器 0 工作方式
MOV AL,00010110B
OUT DX,AL
MOV DX,0D0D0H      ;写计数器 0 时常数
MOV AL,20
OUT DX,AL
```

CLK₀ 加 2MHz 的始终信号，GATE₀,GATE₁ 加 +5V 电压，OUT₀ 输出加到 CLK₁ 做时钟信号，OUT₁ 输出为每 1s 产生一个负脉冲。

初始代码：

```
MOV DX,0D0D6H      ;写计数器 0 的工作方式
MOV AL,00010110B
OUT DX,AL
MOV DX,0D0D0H      ;写计数器 0 的时常数
MOV AL,100
OUT DX,AL
MOV DX,0D0D6H      ;写计数器 1 的工作方式
MOV AL,01110100B
OUT DX,AL
```

MOV DX,0D0D2H ; 写计数器 1 的时常数

MOV AX,20000

OUT DX,AL

XCHG AL,AH

OUT DX,AL

CLK₀加 2MHz 的始终信号， GATE₀,GATE₁加 +5V 电压， OUT₀输出加到 CLK₂ 做时钟信号， OUT₂ 输出为 10s 后产生一个负脉冲。

初始代码：

MOV DX,0D0D6H ; 写计数器 0 的工作方式

MOV AL,00110110B

OUT DX,AL

MOV DX,0D0D0H ; 写计数器 0 的时常数

MOV AX,1000

OUT DX,AL

XCHG AL,AH

OUT DX,AL

MOV DX,0D0D6H ; 写计数器 2 的工作方式

MOV AL,10111000B

OUT DX,AL

MOV DX,0D0D4H ; 写计数器 2 的时常数

MOV AX,20000

OUT DX,AL

XCHG AL,AH

OUT DX,AL

5) 选用地 址 D0D0,D0D2,D0D4,D0D6 为 8253 的端口地址， D0D8 为 GATE 端口地址，该端口采用 74LS373，8253 用方式 4，在 OUT 输出端加非门实现脉冲功能。 接口电路如图：

为： $10 \times 60 \times 0.5 \times 10^6 = 3 \times 10^9$.

由于一个计数器最多分频 65536，所以至少得使用 2 个计数器。我们采用计数器 0 和计数器 1。计数器 0 的时常数 CR0 为 60000，计数器 1 的时常数 CR1 为 50000。

连接方式为：把 0.5MHz 的时钟频率接到计数器 0 的 CLK0，然后把计数器 0 的 OUT0 接到计数器 1 的 CLK1。这样计数器 1 的 OUT1 端输出的就是 10 分钟产生一次的定时中断。

11. 织布机控制系统如图 9.26 所示，已知织布机每织 1 米发出一个正脉冲，每织 100 米要求接收到一脉冲，去触发剪裁设备把布剪开。（1）设 8253 的端口地址为 80H ~ 83H，编写对 8253 初始化程序。（2）假定系统提供的信号频率为 1MHz，希望利用 8253 的其余通道产生 0.1 秒的周期信号，编写初始化程序。

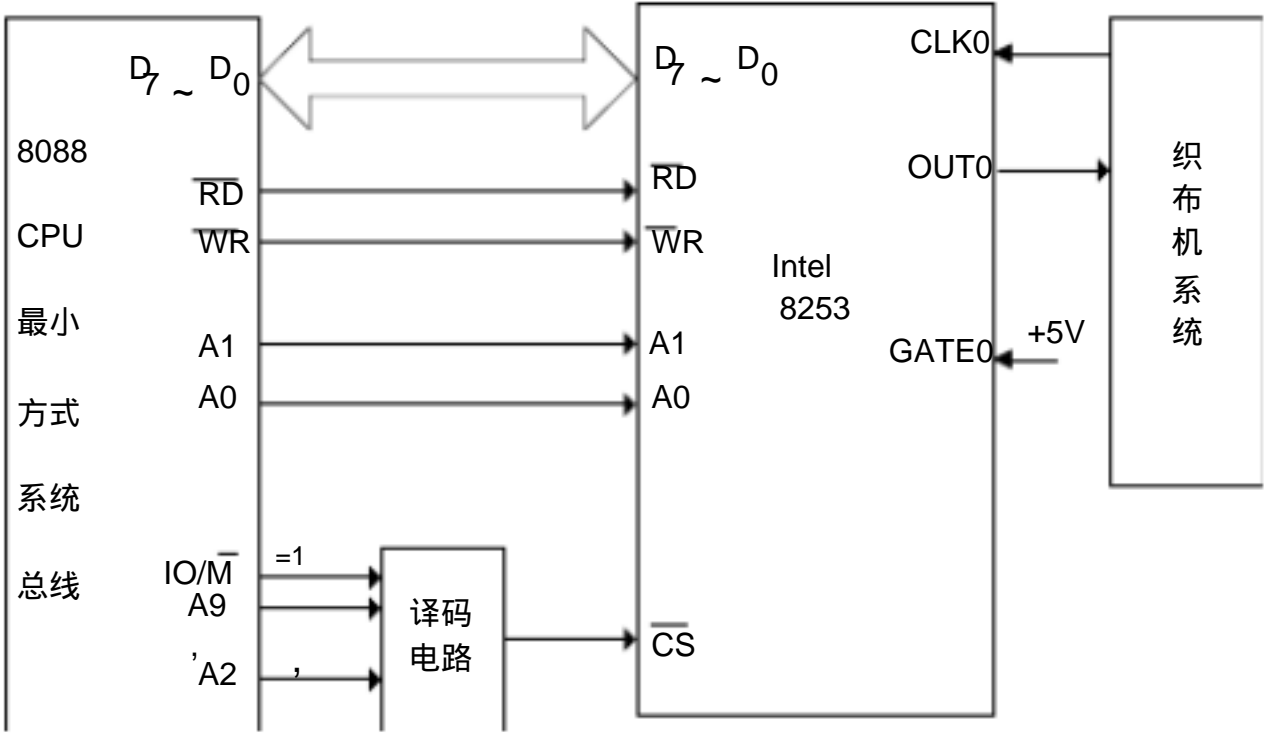


图 9.26 织布机控制系统

解：（1）

```
MOV    DX,83H
MOV    AL,00010100B
OUT    DX,AL
MOV    DX,80H
MOV    AL,100
OUT    DX,AL
```

（2）

将计数器 1 的输出 OUT1 信号作为计数器 2 的时钟输入 CLK2，计数器 1 的时钟输

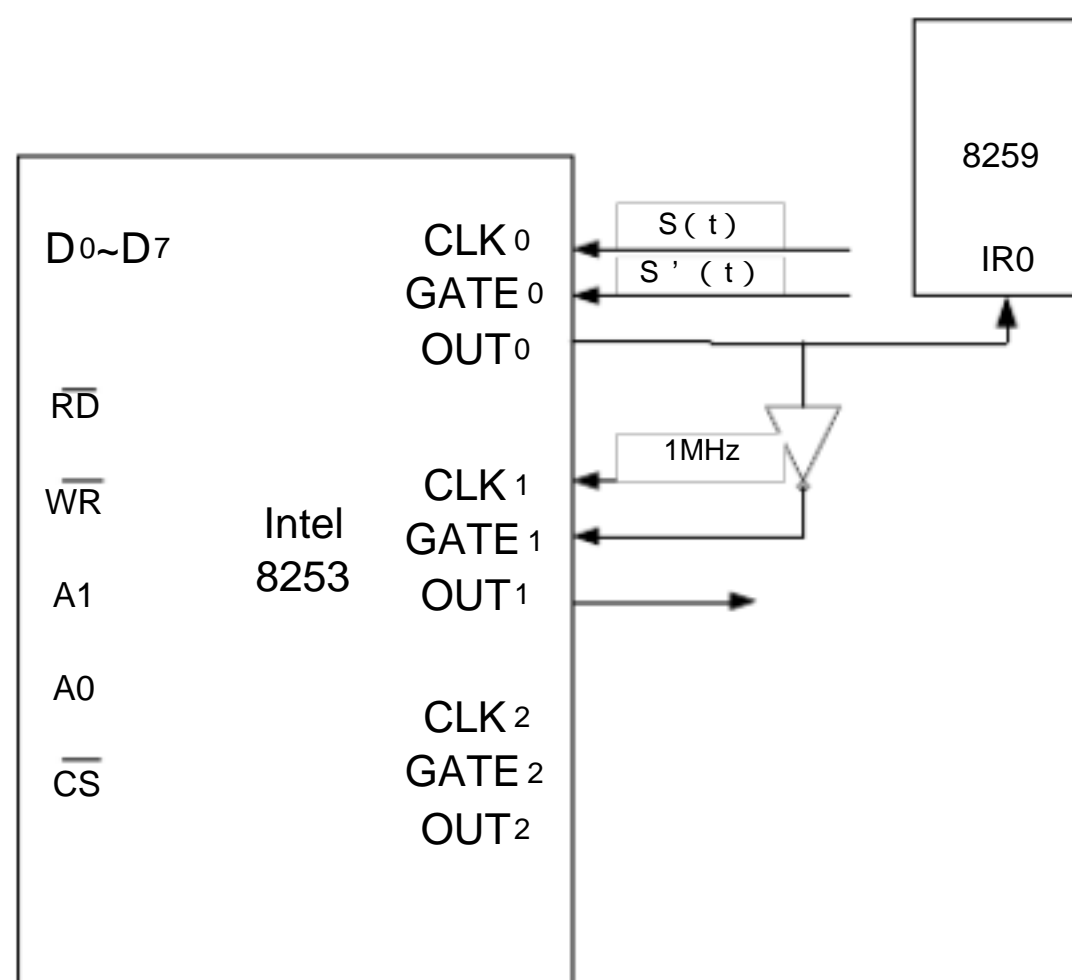
入为系统提供 1MHZ 的信号

```
MOV    DX,83H
MOV    AL,01110100B
OUT    DX,AL
MOV    DX,81H
MOV    AX,1000
OUT    DX,AL
XCHG   AL,AH
MOV    DX,AL
MOV    DX,83H
MOV    AL,10010110B
OUT    DX,AL
MOV    DX,82H
MOV    AL,100
OUT    DX,AL
```

12. 在 IBM PC 系统中根据下列不同条件设计接口逻辑， 利用 8253 完成对外部脉冲信号重复频率的测量。

- (1) 被测脉冲信号的重复频率在 10 ~ 1000Hz 范围内。
- (2) 被测脉冲信号的重复频率在 0.5 ~ 1.5Hz 范围内。
- (3) 被测脉冲信号重复频率在 10 ~ 100Hz 范围内。
- (4) 被测是间歇脉冲信号，每次有信号时有 100 个脉冲，重复频率为 0.8 ~ 1.2MHz，间歇频率大约每秒 15 次，要求测有信号时的脉冲重复频率。

解：用两个计数器，计数器 0 的 CLK接待测信号， GATE接半周期为 10s 的高电平信号， OUT接 8259，同时取反接计数器 1 的 GATE端。计数器 1 的 CLK接系统时钟，半周期为 T_0 。在这样的逻辑电路下，计数器 0 的功能是记录待测信号的脉冲数 N_0 ，计数器 1 的功能是记录在相同时间里系统时钟信号的脉冲数 N_1 。根据 $T=N_1 \cdot T_0 / N_0$ 可计算出待测信号的周期。 $S(t)$ 是待测信号， $S'(t)$ 为给定的周期大于 10s 的高电平信号。



端口声明：COUNTA为计数器 0 的地址，COUNTB为计数器 2 的地址，COUNTC为控制
器地址，COUNTD为 373 地址

程序如下：

```

MOV DX,COUNTD      ; 计数器 1 初始化
MOV AL,01110000B
OUT DX,AL
MOV DX,COUNTB
OUT DX,AL
MOV DX,COUNTB
MOV AL,0
OUT DX,AL

MOV DX,COUNTD      ; 计数器 0 初始化
MOV AL,00010000B
OUT DX,AL
MOV DX,COUNTA
MOV AL,0
OUT DX,AL
OUT DX,AL
STI

```

读两计数器的计数，并进行计算的中断服务子程序：

```
PUSH AX
PUSH BX
PUSH CX
PUSH DX
MOV DX,COUNTD
MOV AL,00000000B
OUT DX,AL
MOV DX,COUNTA
IN AL,DX
XCHG AL,AH
IN AL,DX
XCHG AL,AH
NEG AX
INC AX
MOV BX,AX
MOV DX,COUNTD
MOV AL,00010000B
OUT DX,AL
MOV DX,COUNTB
IN AL,DX
XCHG AL,AH
IN AL,DX
XCHG AL,AH
NEG AX
INC AX
MOV CX,T0
MUL CX
DIV BX
MOV SFR,AX
POP DX
```

POP CX

POP BX

POP AX

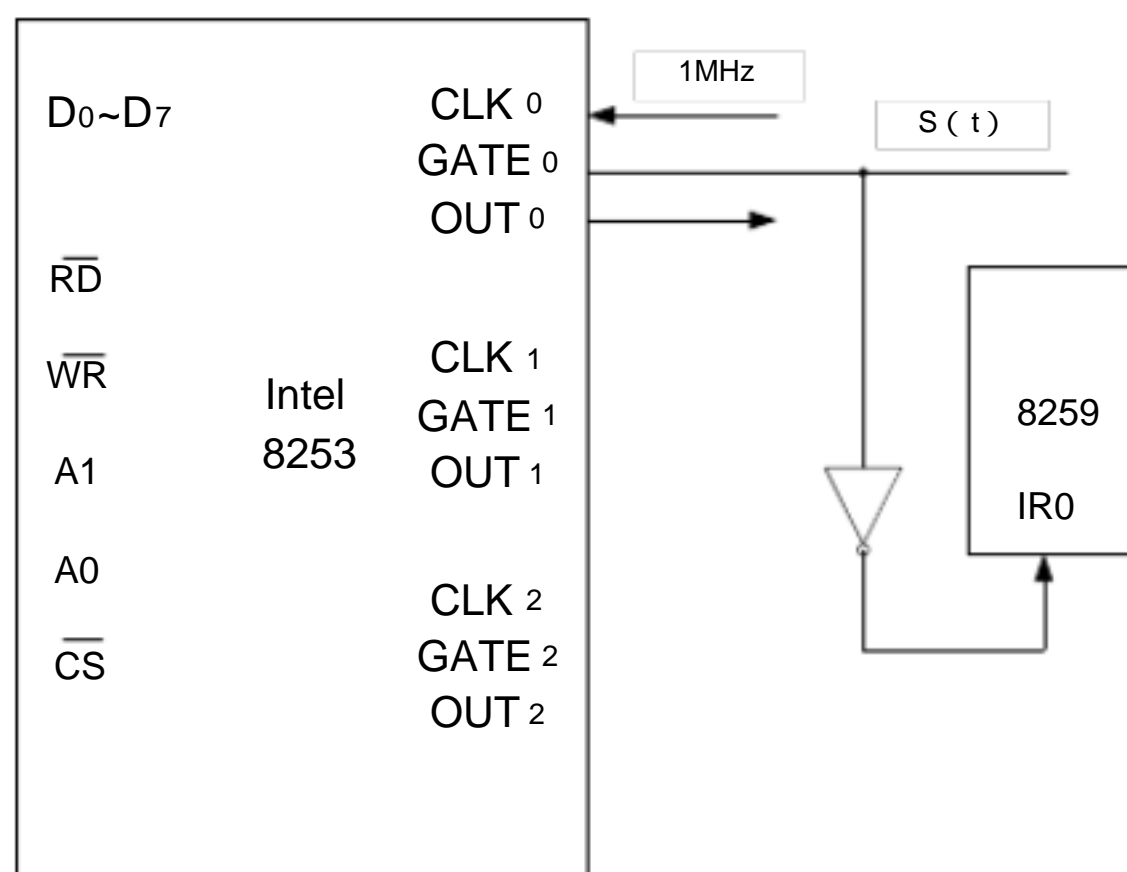
IRET

SFR中保存结果即为待测信号的周期。

对于（1）题， 10×10 不小于 100， 10×1000 不大于 65535，可以用计数法。

同理（3）也可用此方法。

对于（2）题，可用周期法。逻辑电路图如下：



程序如下：

```
MOV DX,COUNTD
```

```
MOV AL,0011 0100B
```

```
OUT DX,AL
```

```
MOV DX,COUNTA
```

```
MOV AL,0
```

```
OUT DX,AL
```

```
OUT DX,AL
```

```
STI
```

```
PUSH AX
```

```

PUSH BX

PUSH DX

MOV DX,COUNTD

MOV AL,0000 0000B

OUT DX,AL

MOV DX,COUNTA

IN AL,DX

XCHG AL,AH

IN AL,DX

XCHG AL,AH

NEG AX

INC AX

MOV BX,AX

MOV DX,000FH

MOV AX,4240H

DIV BX

MOV SFR,AX

POP DX

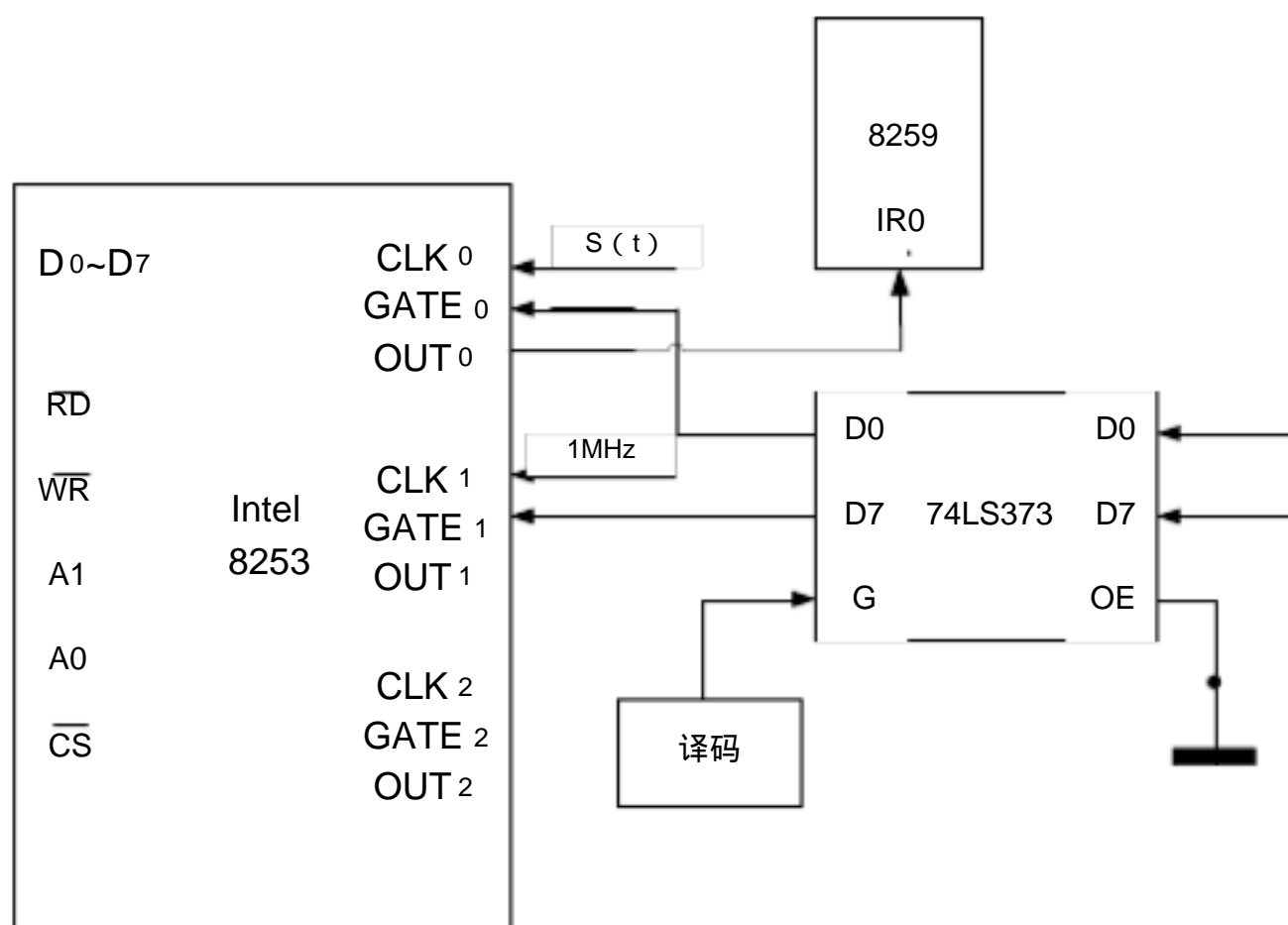
POP BX

POP AX

IRET

```

(4) 如图设计接口，计数器 1 用来记录在 50 个脉冲所用时间，50 个信号脉冲最多用 $1/0.8 \times 50$ (约为 63us) 由于计数器 1 用 1MHz CLK, 故其计数个数 N 即为 N us，所以当 $N > 63$ 时，则有低电平间隔计入，须重新计数。当 $N > 63$ 时，则计算得待测频率。



程序如下：

```
MOV DX,COUNTD
```

```
MOV AL,00H
```

```
OUT DX,AL
```

```
MOV DX,COUNTD
```

```
MOV AL,0001 0000B
```

```
OUT DX,AL
```

```
MOV DX,COUNTA
```

```
MOV AL,50
```

```
OUT DX,AL
```

```
MOV DX,COUNTD
```

```
MOV AL,0111 0000B
```

```
OUT DX,AL
```

```
MOV COUNTB
```

```
MOV AL,0
```

```
OUT DX,AL
```

```
OUT DX,AL
```

```
L2: MOV DX,COUNT      ; 给 GATE0和 GATE1高电平，开始计数
```

```
MOV AL,81H
```

```
OUT DX,AL
```



```

L1: NOP

MOV DX,COUNTD

MOV AL,00000000B

OUT DX,AL

MOV DX,COUNTA

IN AL,DX

MOV DX,COUNTA

IN AL,DX

AND AL,AL          ; 判断是否计完 50 个脉冲，若未计完继续等待

JNZ L1

MOV DX,COUNT

MOV AL,00H          ; 若计完则暂停计数

OUT DX,AL

MOV DX,COUNTD       ; 读计数器 1 结果

MOV AL,01000000B

OUT DX,AL

MOV DX,COUNTB

IN AL,DX

XCHG AL,AH

IN AL,DX

XCHG AL,AH

NEG AX

INC AX

CMP AX,70H          ; 当 AL 大于 70，则有间歇计入，重新测试

JA L2

MOV BL,AL

MOV AL,50           ; 计算频率

DIV BL

MOV FREC,AL

```

第 10 章 定时 / 计数器 8253 应用设计

1. 试分析 8255A 方式 0、方式 1 和方式 2 的主要区别，并分别说明它们适合于什么应用场合。

答：方式 0 是基本的输入 / 输出，端口 A、B、C 都可以作为输入输出端口。适用于 CPU 与非智能 I/O 设备的数据传输；

方式 1 是有联络信号的输入 / 输出，端口 A、B 都可以设定成该方式，此时三个端口的信号线分成了 A、B 两组， $PC_7 \sim PC_4$ 用作 A 组的联络信号， $PC_3 \sim PC_0$ 用作 B 组的联络信号。适用于高速 CPU 与低速 I/O 设备的数据传输；

方式 2 是双向传输，只有 A 组端口可以设定成该方式， $PC_6 \sim PC_7$ 用作输出的联络信号， $PC_4 \sim PC_5$ 用作输入的联络信号， PC_3 用作中断请求信号。适用于双机之间数据的并行传送。

2. 8255A 的 A 组设置成方式 1 输入，与 CPU 之间采用中断方式联络，则产生中断请求信号 INTRA 的条件是 $STBA = \underline{\quad 1 \quad}$, $IBFA = \underline{\quad 1 \quad}$, $INTEA = \underline{\quad 1 \quad}$ 。
3. 如果 8255A 的端口地址为 300H~303H，A 组和 B 组均为方式 0，端口 A 为输出，端口 B 为输入， $PC_3 \sim PC_0$ 为输入， $PC_7 \sim PC_4$ 为输出，写出 8255A 的初始化程序段；编程实现将从端口 C 低 4 位读入的值从高 4 位送出。

解：MOV DX,303H

MOV AL,10000011B

OUT DX,AL

MOV DX,302H

IN AL,DX

MOV CL,4

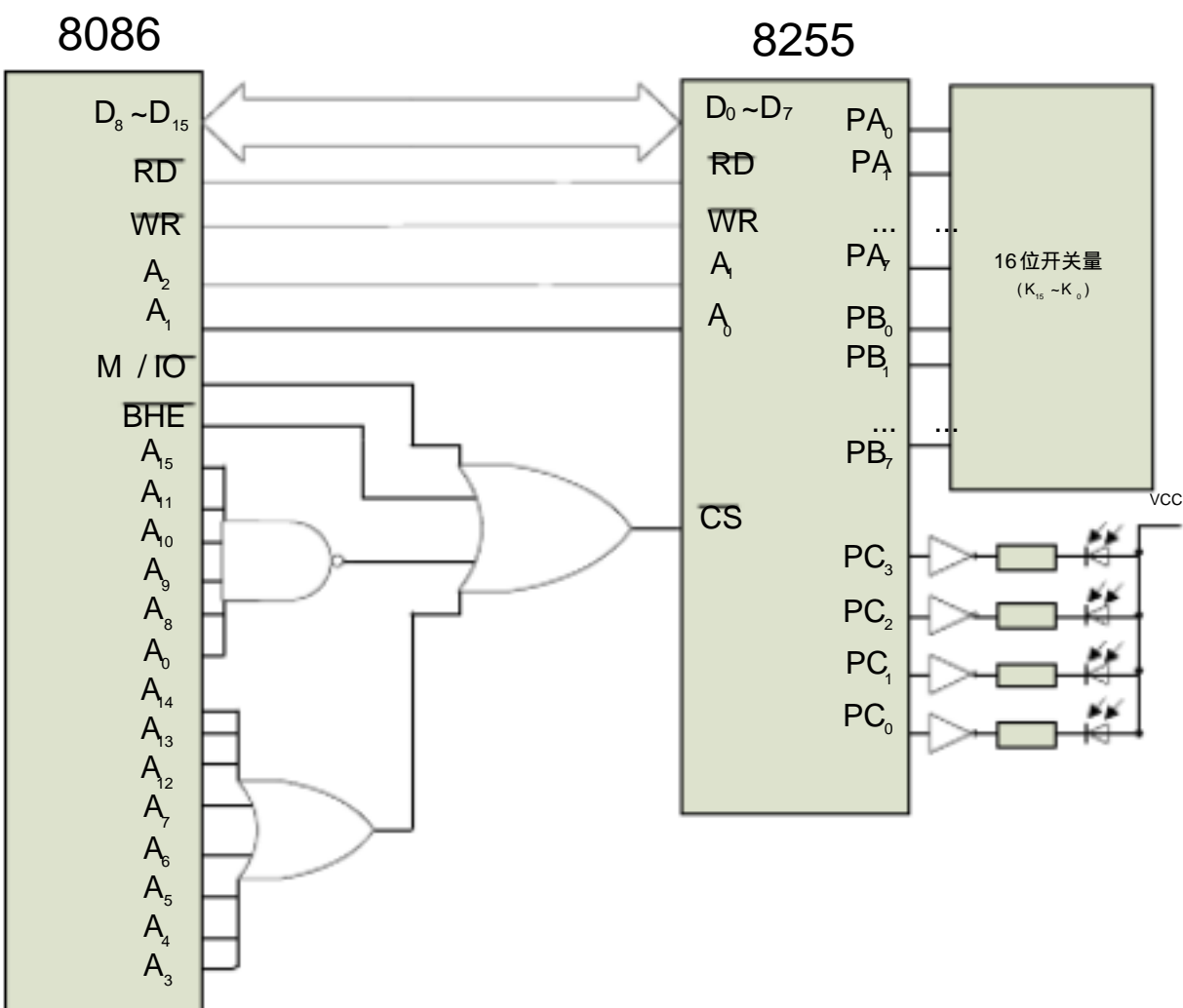
SHL AL,CL

OUT DX,AL

4. 在实际应用中经常需要检测设备的状态，并进行指示。在 8086 最小方式系统下，有一片 8255A，其分配的端口地址为 8F00H~8F07H 中的奇地址，外部设备产生的状态有 16 个 ($K_{15} \sim K_0$)，要求采用 4 个发光二极管来指示开关量中“1”的个数。(1) 画出 8255A

的连接图；（2）编写程序段实现连续检测并显示。

解：（1）



```
(2)  MOV DX,8F07H
      MOV AL,10010010B      ;      端口 A、B 方式 0 输入，端口 C 方式 0 输出
      OUT DX,AL
NEXT:
      MOV DX,8F03H          ;      从端口 B 读取高 8 位开关量
      IN AL,DX
      XCHG AL,AH
      MOV DX,8F01H          ;      从端口 A 读取低 8 位开关量
      IN AL,DX
      MOV BX,AX
      XOR AL,AL
      MOV CX,16
      CLC
L2:
      SHL BX,1
```

```

JNC L1

INC AL

L1:

LOOP L2

MOV DX,8F05H      ;      从端口 C 送出

OUT DX,AL

JMP NEXT          ;      进行下一次检测

```

5. 利用 IBM PC系统机的总线槽，开发由一片 8255A 构成的子系统，8255A端口地址为 260H~263H，编程实现产生如图 10.20 所示的 8 个信号（各个信号的节拍不必严格相等）。

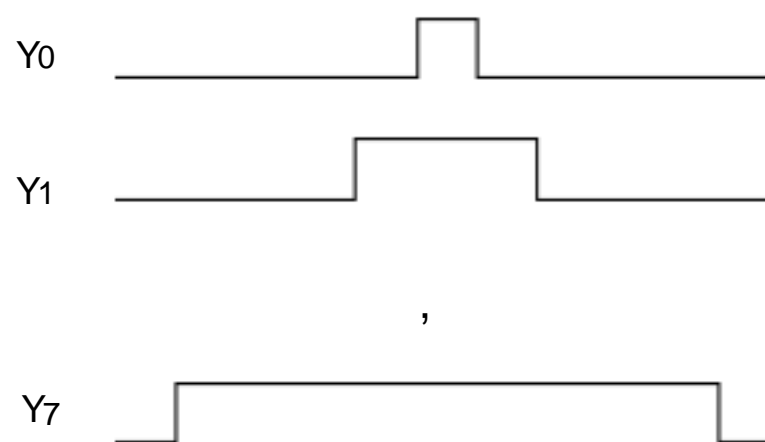
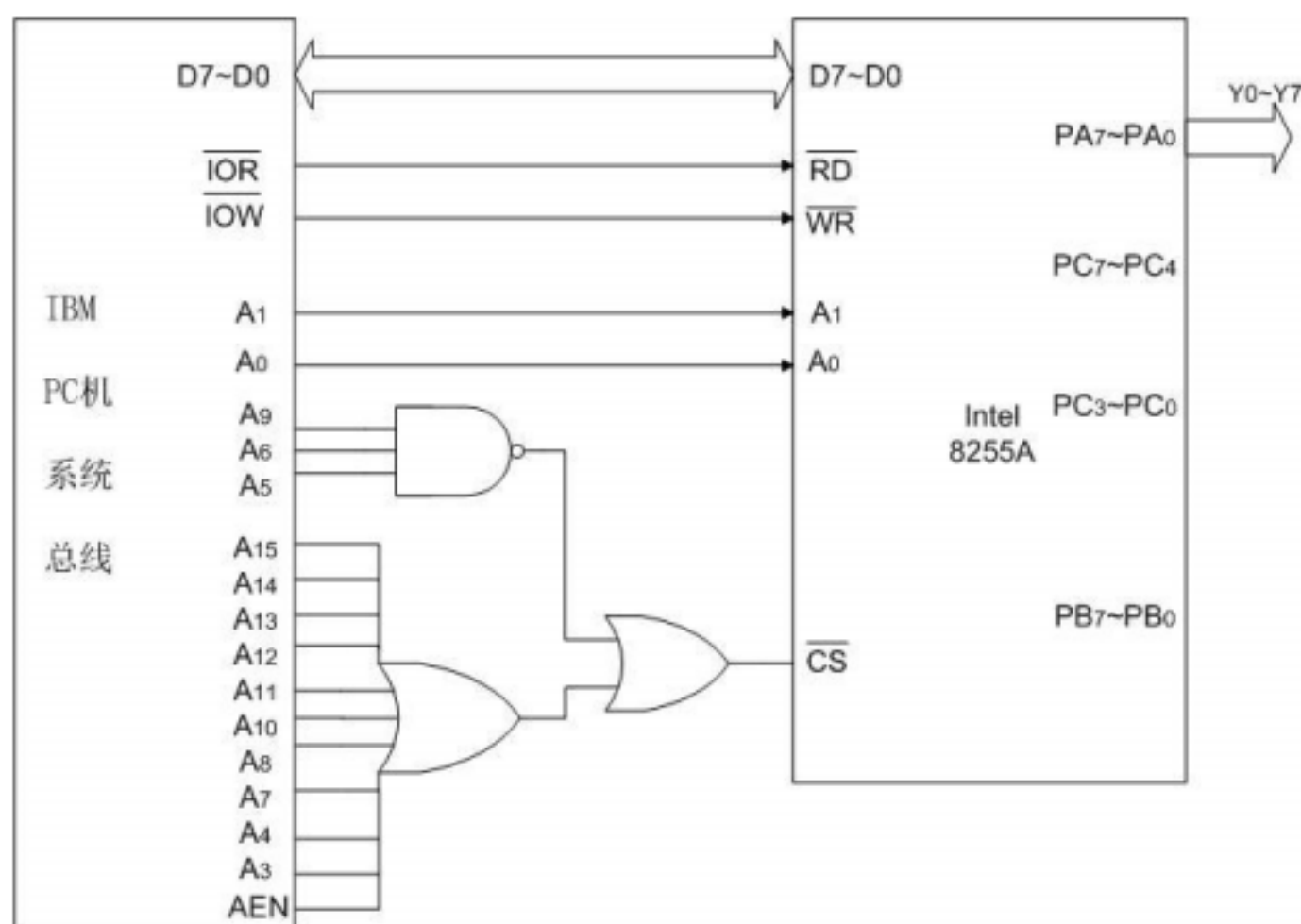


图 10.20 要求产生的信号波形

解：8255A 与 IBM PC 机总线的连接框图如下：



可将 8255A 的端口 A 作为要产生的信号的输出端口，设定为方式 0 输出，端口 B 和端口 C 不做使用，均设定为方式 0 输出。程序段如下：

```

MOV    DX,263H      ;设定 8255A 的工作方式

```

```
MOV    AL,10000000B

OUT    DX,AL

MOV    DX,260H           ;产生指定信号

XOR    AL,AL

OUT    DX,AL

REP:

MOV    AL,80H

MOV    CX,7

REP1:

OUT    DX,AL

SAR    AL,1

LOOP   REP1

MOV    CX,8

REP2:

SHL    AL,1

OUT    DX,AL

LOOP   REP2

JMP    REP
```

6. 在实际应用中，经常会遇到要求输入多个数据量，这时需要用到多路开关，如图 10.21

表示八选一的逻辑框图及其真值表。

现有 8 组 16 位开关量数据（无符号数），要求通过一片 8255A(端口地址为 260H~ 263H) 分时输入到 CPU(8088 最小方式系统) 中，并找出它们中的最大值，并通过 4 个发光二极管指示其序号（灯亮表示“ 1 ”）。画出 8255A 的连接图，并编程实现。

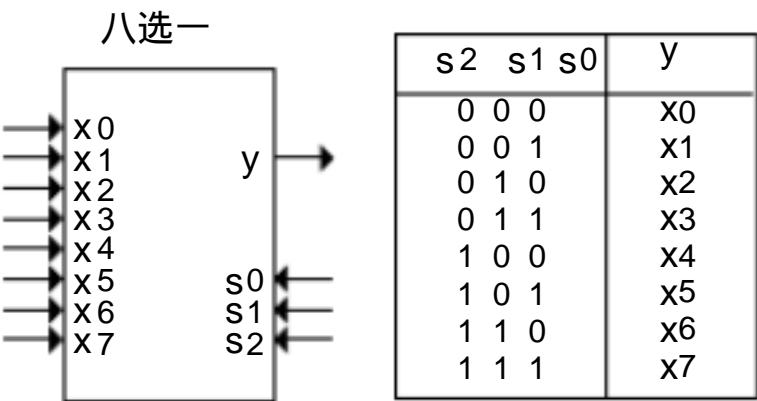
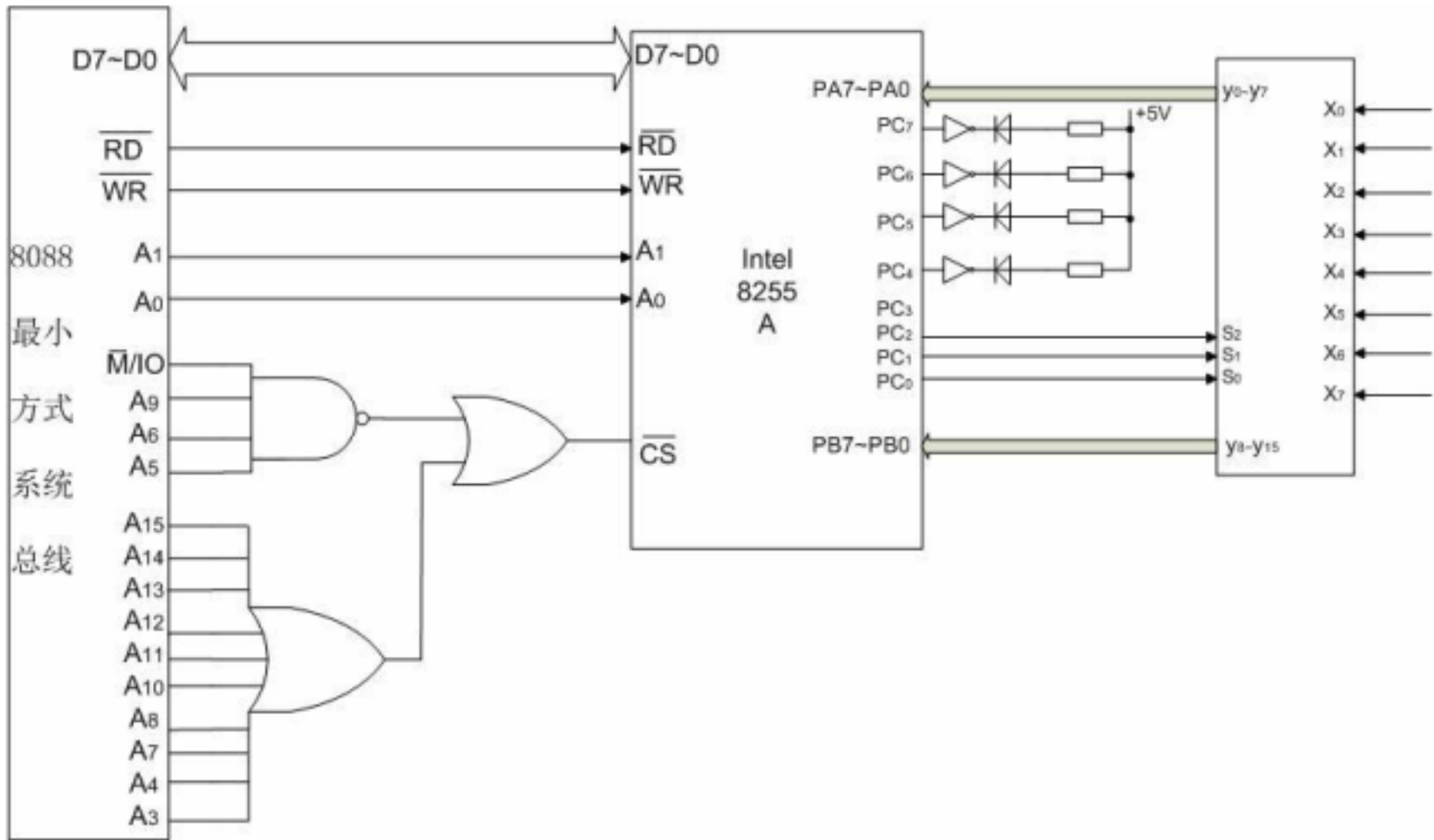


图10.21 八选一逻辑电路

解：由于开关量是 16 为数据，故可以将 8255A 的 PA端与 PB端设定为方式 0，分别读取开关量的低八位和高八位，以 PC低三位端口的控制八选一电路的输出依次从 x0 到 x7，

使用 PC端口的高八位输出最大开关量的序号（该序号为 1~8），控制发光二极管的亮灭以码指示序号。

连接图如下：



程序段如下：

```
MOV    DX,263H

MOV    AL,10011010B    ;设定工作方式， PA,PB 均工作于方式 0，PA、PB 为
                          输入，PC 为输出

OUT     DX,AL

MOV     CX,8

XOR     BX,BX

MOV     SI,0            ;SI 表示输入开关量的序号
```

ST1:

```
MOV     DX,262H

MOV     AX, SI

OUT     DX,AL

MOV     DX,261H        ;将开关量数据的高八位写入    AH

IN      AL,DX

MOV     AH,  AL
```

MOV	DX,260H	;将开关量数据的低八位写入	AL
IN	AL,DX		
CMP	BX, AX		
JA	NEXT		
MOV	BX,AX	;将当前最大值保存在	BX 中
INC	SI		
PUSH	SI	;将当前最大值的序号压栈	
DEC	SI		
NEXT:			
INC	SI		
LOOP	ST1		
POP	SI	;最大值的序号出栈	
XOR	AX,AX		
MOV	AX,SI		
MOV	CL,4		
ROL	AL,CL	;将最大值的序号 (4 位) 移至	AL 的高四位
MOV	DX,262H		
OUT	DX,AL	;PC 的高四位输出最大值序号	