

C 语言

第一章

选择

ACACC ACBBA BAADA CBDBC CCCBD C

填空

1、26 2、2 3、5.5 4、3.5 5、1 6、0 7、 $a/(b*c)$ 8、28

第二章

选择

DBADB BABAD BA

第三章

选择

BCDBB BACDA C

写结果

1、33 2、2 3、-1 4、 $a=2, b=1$ 5、26 53 6、*&

7、**1**

 3

第四章

选择

CACBB BCDBB CBAAB ABCDB ABC

写结果

1、16 2、#### 3、#### 4、54321 5、3 6、4, 10

第五章

选择

CBDAD DDACA DAA

写结果

1、 $sum=114$ 2、22111 3、107

4、The array a:

 1 2 3

 4 5 6

 The array b:

 1 4

 2 5

 3 6

5、1 2 3

 0 5 6

 0 0 9

6、1 0 0

 4 5 0

 7 8 9

字符数组

选择

ABAAC CDBAC CDD

写结果

1、abct 2、n=5 3、t*M 4、mo 5、LMNP

第八章

选择

BACBA DCCA

写结果

1、i=2, j=1

i=1, j=2

2、23

45

3、a=5, b=3

a=4, b=6

4、6

5、10

6、sum=14

7、1, 1

2, 2

6, 3

24, 4

8、9

9、3, 4, 8

3, 4, 4

10、8

11、1800

12、1, 2, 3, 4, 9, 8, 7, 6, 5, 0,

第九章

选择

BABDD BCBDC CAD

写结果

1、BCG

2、0 0 0 8

3、*2*4*6*8*

第十章

选择

CBDBC

第十一章

选择

DBCDB CBDAD BBD



第一章

参考答案

一、完成下列数制转换。

- (1) $101.011B=5.375D=5.3Q=5.6H$
- (2) $101110B=46D=56Q=2EH$
- (3) $1101.01B=13.25D=15.2Q=D.4H$
- (4) $10011010.1011B=154.6875D=232.54Q=9A.B H$
- (5) $253.74Q=171.9375D=10101011.1111B=AB.F H$
- (6) $712Q=458D=111001010B=1CAH$
- (7) $72D=1001000B=110Q=48H$
- (8) $49.875D=11001.111B=61.7Q=31.E H$
- (9) $0.6875D=0.1011=0.54Q=0.BH$
- (10) $58.75D=111010.11 B=72.6 Q= 3A.CH$
- (11) $0E12H=3602D=7022Q=111000010010B$
- (12) $1CB.D8H=459.84375D=713.66Q=111001011.11011B$
- (13) $FF.1H=255.0625D=377.04Q=11111111.0001B$
- (14) $70ADH=28845D=70255Q=111000010101101B$

二、给出下列数的原码和补码的二进制表示。(设机器数字长为8。)

- (1) $[-38D]$ 原=10100110B $[-38D]$ 反=11011001B $[-38D]$ 补=11011010B
- (2) $[32D]$ 原=00100000B $[32D]$ 反=00100000B $[32D]$ 补=00100000B
- (3) $[-63D]$ 原=10111111B $[-63D]$ 反=11000000B $[-63D]$ 补=11000001B
- (4) $[-64D]$ 原=11000000B $[-63D]$ 反=10111111B $[-63D]$ 补=11000000B
- (5) $[-0D]$ 原=10000000B $[-0D]$ 反=11111111B $[-0D]$ 补=00000000B
- (6) $[42D]$ 原=00101010B $[42D]$ 反=00101010B $[42D]$ 补=00101010B
- (7) $[-45D]$ 原=10101101B $[-45D]$ 反=11010010B $[-45D]$ 补=11010011B
- (8) $[45D]$ 原=00101101B $[45D]$ 反=00101101B $[45D]$ 补=00101101B
- (9) $[-72D]$ 原=11001000B $[-72D]$ 反=10110111B $[-72D]$ 补=10111000B
- (10) $[72D]$ 原=01001000B $[72D]$ 反=01001000B $[72D]$ 补=01001000B
- (11) $[-1111111B]$ 原=11111111B; $[-1111111B]$ 反=10000000B; $[-1111111B]$ 补=10000001B
- (12) $[+1001100B]$ 原=01001100B; $[+1001100B]$ 反=01001100B; $[+1001100B]$ 补=01001100B

三、已知 X, Y, 求 $[X+Y]$ 补=? , $X+Y=?$ $[X-Y]$ 补=? , $X-Y=?$ 并指出结果是否有溢出?

- (1) $[X+Y]$ 补= 50H, $X+Y=80D$, 结果无溢出。
 $[X-Y]$ 补= 38H, $X-Y=56D$, 结果无溢出。
- (2) $[X+Y]$ 补= EDH, $X+Y= -19D$, 结果无溢出。
 $[X-Y]$ 补= D3H, $X-Y= -45D$, 结果无溢出。
- (3) $[X+Y]$ 补= 22H, $X+Y=34D$, 结果无溢出。
 $[X-Y]$ 补= 9EH, $X-Y= -98D$, 结果无溢出。
- (4) $[X+Y]$ 补=DEH, $X+Y= -34D$, 结果无溢出。
 $[X-Y]$ 补=56H, $X-Y= -170D$, 结果有溢出。
- (5) $[X+Y]$ 补= A9H, $X+Y= -87D$, 结果无溢出。
 $[X-Y]$ 补= EBH, $X-Y= -21D$, 结果无溢出。

- (6) $[X+Y]_{\text{补}}=55\text{H}$, $X+Y=85\text{D}$, 结果无溢出。
 $[X-Y]_{\text{补}}=97\text{H}$, $X-Y=151\text{D}$, 结果有溢出。
 (7) $[X+Y]_{\text{补}}=\text{FEH}$, $X+Y=-2\text{D}$, 结果无溢出。
 $[X-Y]_{\text{补}}=54\text{H}$, $X-Y=-172\text{D}$, 结果有溢出。
 (8) $[X+Y]_{\text{补}}=\text{B2H}$, $X+Y=178\text{D}$, 结果有溢出。
 $[X-Y]_{\text{补}}=08\text{H}$, $X-Y=8\text{D}$, 结果无溢出。

四、将下列压缩的 8421BCD 码表示成十进制数和二进制数。

- (1) $10010100\text{BCD}=94\text{D}=01011110\text{B}$
 (2) $01101000\text{BCD}=68\text{D}=01000100\text{B}$
 (3) $00010101\text{BCD}=15\text{D}=00001111\text{B}$
 (4) $01001000\text{BCD}=48\text{D}=00110000\text{B}$

五、将下列数值或字符串表示为相应的 ASCII 码。

- (1) 00H (2) 51H
 (3) 3531H (4) 48656C6C6F21H

第二章

参考答案

一、问答题

- 8088 属于 (准 16) 位微处理器。它有 (8) 根数据线, (20) 根地址线, 寻址空间为 220B, 即 1MB。
- 有指令队列后, 在执行部分执行指令的同时, 总线接口部件就能从存储器向指令队列中取下一条指令, EU 和 BIU 并行工作, 从而提高了 CPU 的工作效率。
- 指示偏移地址的寄存器有 BX、BP、SI、DI、SP、IP。
 - BX: 在寄存器间接寻址、寄存器相对寻址、基址变址寻址和相对基址变址寻址方式中, 隐含的数据段是 DS。
 - BP: 在寄存器间接寻址、寄存器相对寻址、基址变址寻址和相对基址变址寻址方式中, 隐含的数据段是 SS。
 - SI: 寄存器间接寻址、寄存器相对寻址、基址变址寻址和相对基址变址寻址方式中, 隐含的数据段是 DS。在字符串操作时, SI 作为变址, 隐含的数据段是 DS。
 - DI: 寄存器间接寻址、寄存器相对寻址、基址变址寻址和相对基址变址寻址方式中, 隐含的数据段是 DS。在字符串操作时, DI 作为目标变址, 隐含的数据段是 ES。
 - SP: 在堆栈操作中 (PUSH、POP、CALL、RET 等) 使用, 隐含的数据段是 SS。
 - IP: 在取指令时使用, 隐含的数据段是 CS。一般用户在程序中不使用。
- 其中 DF、IF、TF 为控制标志, 不受计算结果影响。 5. (1) OF=1 (2) ZF=1 (3) SF=1 (4) TF=1 (5) AF=1 (6) IF=1 (7) CF=1 (8) PF=0
- 逻辑地址是由段基址和偏移地址两部分组成, 无论是段基址还是偏移地址, 都是 16 位地址。通过把段基址自动左移 4 位, 加上对应的偏移地址, 形成 20 位物理地址。
- 连接 8086 有 20 位地址线, BHE 也需要锁存, 共有 21 个信号需锁存, 选用 8282 作为锁存器, 每片 8 位, 共需 3 片。8082 的输入端 DI 与 CPU 的地址总线相连, 输出端 DO 与系统总线的地址总线相连。8082 的 STB 与 CPU 的 ALE 相连。如 CPU 不需出让总线, 则 8282 的 OE 可直接接地。在总线周期的 T1 状态, AD15~AD0 和 A19/S6~A16/S3 均输出地址信息。BHE/S7 作为 BHE 使用。ALE 输出一个正脉冲, 它的下降沿把地址信息打入锁存器; 在其他 T 状态,

以上引脚都改变了意义。但是，锁存器可提供系统有效的地址信息。

8. AAA40H

9. 复位期间，8086/8088 内部寄存器被设置为如下初值：

标志寄存器	清 零
指令指针(IP)	0000H
CS 寄存器	FFFFH
DS 寄存器	0000H
SS 寄存器	0000H
ES 寄存器	0000H
指令队列	空
其他寄存器	0000H

二、填空题

1. BIU, EU
2. 0006H, 2300H
3. 4, 16, 16, 6, 20, 总线
4. 4, 4, 1, ALU
5. 状态, 控制
6. 物理地址, 逻辑地址
7. 20, 00000H~FFFFFH
8. 偏移地址
9. 1, 2
10. 20, 1M
11. READY=0, T3, TW
12. 0, 0
13. 时钟周期, T, 4

三、选择题

1. C; 2. C; 3. A; 4. D; 5. C; 6. B; 7. B; 8. D; 9. A; 10. D;
11. C; 12. C; 13. A; 14. D; 15. B; 16. B; 17. D; 18. B; 19. B; 20. A;

四、计算题

1. (1) ACC1H, OF=1, SF=1, ZF=0, AF=0, PF=0, CF=0。
(2) 0700H, OF=0, SF=0, ZF=0, AF=0, PF=1, CF=1。
2. (1) 5D14H, OF=0, SF=0, ZF=0, AF=0, PF=1, CF=0。
(2) C754H, OF=0, SF=1, ZF=0, AF=0, PF=0, CF=1。
(3) 1F88H, OF=0, SF=0, ZF=0, AF=0, PF=1, CF=1。
(4) 45B0H, OF=1, SF=0, ZF=0, AF=0, PF=0, CF=0。
- 3.

段 基 址	偏 移 地 址	物 理 地 址
1234H	5678H	179B8H
ABCDH	2345H	AE015H
CD02H	ACB8H	D7CD8H
2000H	1218H	21218H

第三章

参考答案

1. A; 2. B; 3. B; 4. A; 5. B; 6. C; 7. ①D②D; 8. D; 9. A; 10. B;
11. D; 12. B; 13. C; 14. A; 15. C; 16. C; 17. D; 18. D; 19. C; 20. A;
21. A; 22. A; 23. B; 24. A;
25. D 26. D 27. C 28. D、C 29. A 30. C 31. A 32. C
33. B 34. C 35. A 36. A 37. D 38. C 39. A 40. B
41. D 42. C 43. A 44. C 45. D 46. C 47. B 48. B
49. B, D 50. D 51. A 52. D 53. C 54. B 55. B 56. D
57. B 58. A 59. C、B 60. A 61. C

二、分析程序题

1. AX=0123H, BX=0A567H
2. AX= 1200H, BX= 5600H, CX= 3400H
3. ① MOV CX, 2② RCR AX, 1
4. ① N ② ARRAY[SI]
5. ①10② JL ③ SI ④ JNZ
6. ① JLE ② CMP ③ XCHG
7. ① MOV BX, ② JLE NEXT③ JNZ NEXT④ JNZ AGAIN
8. ① 0100H② JZ NEXT
9. ① CMP② JNL NEXT
10. ① AND AX, AX ② JNC L1
11. (1)该程序段完成的功能是什么?(统计以 DATA 开始的 200 带符号数中大于等于 20 的字节个数, 其个数放 DX 中。)
(2)如果将 JL 改为 JG, 该程序段完成的功能是什么?(统计以 DATA 开始的 200 带符号数中小于等于 20 的字节个数, 其个数放 DX 中。)
12. 依次将~9 的 ASCII 码存入从 BUF 开始的单元中。
13. 在 BUFFER 开始的内存单元中有 10 个 8 位无符号字节数据, 找出最大值并送 ARG 单元。
14. 在 BUF 单元开始, 存储了 20 个无符号字节数, 试编程统计大于等于 60 及小于 60 的个数, 并将统计结果分别存放在寄存器 DH 和 DL 中。
15. (DX): (AX)的内容左移 4 位。
16. ①STR[SI]②SI③LOOP

三、简答题

1. 答案:

- (1)源操作数采用直接寻址方式, 物理地址 PA=23567H。
(2)源操作数采用直接寻址方式, 物理地址 PA=ES×16+4678H=5AD78 H。
(3)源操作数采用 SI 的寄存器间接寻址方式, 物理地址 PA=DS×16+EA=2ABCDH。
(4)源操作数采用相对基址加变址的寻址方式, 物理地址 PA=DS×16+EA=2FF1DH。
(5)源操作数采用寄存器相对寻址方式, 物理地址 PA=2ABEBH。
(6)目标操作数采用基址加变址的寻址方式, 该指令使用段超越前缀 ES, 物理地址=6331DH。
(7)源操作数采用 BX 的寄存器间接寻址方式, 物理地址 PA=47650H。

(8)源操作数采用 BP 的寄存器间接寻址方式，物理地址 PA=4CE96H。

2. 答案：

(1)MOV DS, 5000H

错误。不允许直接向段寄存器送立即数，可改为：
MOV AX, 5000H
MOV DS, AX

(2)MOV CX, AL

错误。源操作数和目标操作数类型不一致，可改为：
MOV CX, AX

(3)MOV BL, [SI]

对。

(4)MOV [1200H], 56H

错误。存储器操作数与立即数在同一个指令时，无法确定操作数的类型，必须用 PTR 运算符指出存储器操作数的类型，用 BYTE PTR 指示字节类型，WORD PTR 指示字类型，则第一操作数前应加上 BYTE PTR 或 WORD PTR 说明。可改为：

MOV BYTE PTR [1200H], 56H

(5)PUSH BL

错误。堆栈指令操作数只能以字为单位进行，而 BL 是一个字节。可改为：
PUSH BX

(6)MOV AX, [SI][DI]

错误。使用存储器寻址时，2 个基址寄存器或 2 个变址寄存器不能同时使用，源操作数寻址方式错，2 个寄存器都是变址寄存器。可改为：

MOV AX, [SI+BX]

(7)XCHG BX, [1000H]

对。

(8)IN AH, DX

输入指令从端口读入数据送给累加器 AL 或 AX。可改为：

IN AL, DX

(9)POP CS

CS 不能做为目标操作数。可改为：

POP AX

(10)MOV [DI], ES: [BX]

两个存储单元不能直接传送数据。可改为：

MOV AX, ES: [BX] MOV [DI], AX

(11)IN AL, 1FFH

错误。端口地址 1FFH 为 16 位，端口地址应用 DX 间址。可改为：

MOV DX, 1FFH

IN AL, DX

(12)RCL CX, 3

错误。当逻辑移位的次数大于 1 时，用 CL 指出移位位数。可改为：

MOV CL, 3

RCL CX, CL

(13)LEA ES, [SI]

错误。目标操作数必须是 16 位通用寄存器。可改为：

LEA BX, [SI]

(14) MOV DS, ES

错误。段寄存器之间不能直接传送。可改为：

MOV AX, ES

MOV DS, AX

(15) MOV BL, 300

错误。500 超出一个字节所表示的范围。可改为：

MOV BX, 300

(16) MOV BYTE PTR [DI], 400

错误。用 BYTE PTR 指示字节类型，400 超出一个字节所表示的范围，数据不能正确保存，应该用 WORD PTR 说明。可改为：

MOV WORD PTR [DI], 400

(17) INC [DI]

错误。若操作数为存储单元，则必须指明其操作数的类型，字节或字类型。该指令没指明操作数的类型。可改为：

INC WORD PTR [DI]；对 DI 所指的单元的内容加 1

INC BYTE PTR [DI]；对 DI 所指的字节单元的内容加 1

(18) ADD [DI], 20H

错误。立即数和存储器操作数本身没有明确的类型，必须用 PTR 运算符指出存储器操作数的类型。可改为：

ADD BYTE PTR [DI], 20H

(19) POP [AX]

错误。AX 不能用于间接寻址，间接寻址只能用 BX、BP、SI、DI 四个寄存器之一。可改为：

POP [BX]

(20) MOV AX, [DX]

错误。间接寻址只能用 BX、BP、SI、DI 四个寄存器之一，DX 不能作间接寻址寄存器。可改为：

MOV AX, [SI]

(21) ADD AX, [BX+BP+30]

错误。使用存储器寻址时，2 个基址寄存器或 2 个变址寄存器不能同时使用，源操作数寻址方式错，2 个寄存器都是基址寄存器。可改为：

ADD AX, [BX+DI+30]

3. 答案：

(1) AND AX, 00FFH

(2) OR AX, 0F000H

(3) XOR AX, 00FFH

(4) MOV AX, ES: [1000H]

(5) MOV SI, 6

MOV AX, BUFFER[SI]

(6) MOV AX, 1000H

MOV DS, AX

MOV AL, DS: [5000H]

另一种解法：

MOV AX, 1000H

MOV DS, AX

MOV SI, 5000H

MOV AL, [SI]

4. 答案:

- (1) AX=1448H, SF=0, ZF=0, AF=1, PF=1, CF=1, OF=0。
- (2) AX=687BH, SF=1, ZF=0, AF=1, PF=0, CF=1, OF=1。
- (3) AX=2849H, SF=0, ZF=0, AF=X, PF=0, CF=OF=0。
- (4) AX=0C3B6H, SF=1, ZF=0, AF=X, PF=0, CF=OF=0。
- (5) AX=687BH, SF=1, ZF=0, AF=1, PF=0, CF=1, OF=1。
- (6) AX=687CH, SF=0, ZF=0, AF=0, PF=1, CF 未影响, OF=0。

5. 答案:

(1) 直接寻址方式

操作数在存储器中, 16 位偏移地址直接包含在指令的方括号中, 即: EA=指令给出的数值。

直接寻址方式指令为

MOV AX, [2C69H]

有效地址 EA=2C69H

物理地址=DS×16+EA=32000H+2C69H=34C69H

(2) 用 SI 的寄存器间接寻址方式

操作数在存储器中, 操作数在段内的偏移地址由指令指定的寄存器 SI 提供。即: EA=寄存器的值。用 SI 的间接寻址方式指令为: MOV AX, [SI]

有效地址 EA=2598H

物理地址=DS×16+EA=32000H+2598H=34598H

(3) 用 BX 的寄存器相对寻址方式

操作数的有效地址是指令指定的寄存器 BX 内容与指令给定的位移量(8 位或 16 位)之和, 有效地址 EA=[BX/BP/SI/DI] + [8/16 位位移量]。用 BX 的寄存器相对寻址方式指令为:

MOV AX, [BX+2C69H]

有效地址 EA=105AH+2C69H=3CC3H

物理地址= DS×16+EA=32000H+3CC3H=35CC3H

(4) 用 BX 和 SI 的基址加变址寻址方式

有效地址由基址寄存器 BX 的内容加上变址寄存器 SI 的内容, 有效地址=BX+SI。基址加变址寻址方式指令为:

MOV AX, [BX + SI]

有效地址 EA=105AH+2598H=35F2H

物理地址=32000H+35F2H=355F2H

(5) 用 BX 和 SI 的相对基址变址寻址

操作数的有效地址是, 由指令中使用基址寄存器 BX 的内容, 加上变址寄存器 SI 的内容, 再加上指令给定的位移量的内容构成, 即: EA=BX 的值+SI 的值+位移量。相对的基址加变址的寻址指令为:

MOV AX, [BX + SI+1B57 H]

有效地址 EA=105AH+2598H+2C69H=625BH 物理地址=DS×16+EA=32000H+625BH=3825BH

四、编程题

1.

MOV CL, COUNT

MOV CH, 0

LEA SI, STRING

```

LEA DI, BUFFER
MOV DL, 0
AGAIN: MOV AL, [SI]
INC SI
CMP AL, 30H( '0' )
JB NEXT
CMP AL, 3AH
JAE NEZT
AND AL, 0FH
MOV [DI], AL
INC DI
INC DL
NEXT: DEC CX
JNZ AGAIN
HLT

```

2.

```

MOV DL, BH
AND DL, 0F0H
MOV CL, 4
SHR DL, CL
PUSH BX
AND BH, 0FH
MOV CL, BH
AND BL, 0F0H
MOV CL, 4
SHR BL, CL
POP BX
AND BL, 0FH
MOV AL, BL

```

3. 分析：一种可能的程序实现如下：（注意高字节应该使用带符号数比较指令 JG、JL 等，低字节应使用无符号数比较指令 JA、JB 等。）

```

...
CMP DX, CX           ; 比较高 16 位
JG EXIT              ; 若大于，则 DX: AX 存的是大数，转 EXIT
JL NEXT              ; 若小于，则 CX: DX 存的是大数，转 NEXT
CMP AX, BX           ; 若等于，进一步比较低 16 位
JAE EXIT              ; 若 AX ≥ BX，则 DX: AX 存的是大数，结束
NEXT: MOV AX, BX      ; 将大数送到 DX: AX 中
MOV DX, CX
EXIT: HLT

```

另解：

```

PUSH CX
PUSH BX
SUB BX, AX

```

```
SBB CX, DX
JLE EXIT
POP AX
POP DX
EXIT: HLT
```

4.

(1)用一般的传送指令完成

```
MOV AX, 1000H
MOV DS, AX
MOV SI, 2000H
MOV DI, 3000H
MOV CX, 100
AGAIN: MOV AX, [SI]
MOV [DI], AX
INC SI
INC SI
INC DI
INC DI
DEC CX
JNZ AGAIN
HLT
```

(2)用串传送指令完成

```
MOV AX, 1000H
MOV DS, AX
MOV ES, AX
MOV SI, 2000H
MOV DI, 3000H
MOV CX, 100
CLD
AGAIN: MOVSW
DEC CX
JNZ AGAIN HLT
```

(3)用带重复前缀的串传送指令完成

```
MOV AX, 1000H
MOV DS, AX
MOV ES, AX
MOV SI, 2000H
MOV DI, 3000H
MOV CX, 100
CLD
REP MOVSW
HLT
```



第四章

参考答案

- 5
- (1) AX=4241H, (2) BX=1234H, (3) CX=4142H
- “?”表示不写入数据,但保留对应的存储单元,用 n DUP(a, b, ...)形式表示存入的数据以括号中的规律重复地排列 n 次。

X	5
	—
	—
Y	0A
	—
	—
Z	0F
	0
	0B
	—
	0F
	0B
—	
0F	

图 4-25 习题 3 数据在内存的存放形式

- ①0BH, ②8AH, ③05H, ④0D7H, ⑤2, ⑥CLC, ⑦0,
- (1) 该程序完成什么功能?

$$C = \begin{cases} -10 & X < Y \\ 0 & X = Y \\ 10 & X > Y \end{cases}$$

- (2) 程序运行完后, C 中的内容是什么?

C=10

- (1) 该程序完成什么功能?

对 DAT1 单元中存放的带符号字节数取绝对值送 S。

- (2) 程序运行后, S 中的内容是什么?

11110011 → 00001101B = 0DH → DAT2

- ① TEST ② JZ ③ MEMB ④ MEMA

8.

- (1) 画出内存分配图。

变量	值	偏移地址
BUF1	41H	2000H
	42H	2001H
	43H	2002H
	44H	2003H
	45H	2004H

(2) 程序执行后，从 BUF2 开始的 N 个字节单元中的内容是 61H、62H、63H、64H。

变量	值	偏移地址
BUF1	41H	2000H
	42H	2001H
	43H	2002H
	44H	2003H
	45H	2004H
BUF2	61H	2005H
	62H	2006H
	63H	2007H
	64H	2008H
	65H	2009H

(3) 用指令“ADD AL, 20H”代替指令“OR AL, 20H”，程序功能不变。

9. 一个非压缩码占用一个字节，且高 4 位为，转为 ASCII 码时，应将其高 4 位加 3，可通过“ADD DL, 30H”或“OR DL, 30H”实现。

```

DATA    SEGMENT
ALF     DB  08H, 09H
BLF     DB  ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS: CODE, DS: DATA

```

```

START:  MOV  AX, DATA
        MOV  DS, AX
        MOV  BX, OFFSET ALF
        MOV  SI, OFFSET BLF
        MOV  DL, [BX]           ; 取第一个数
        ADD  DL, 30H           ; 转为 ASCII 码
        MOV  [SI], DL
        INC  BX
        INC  SI
        MOV  DL, [BX]
        ADD  DL, 30H
        MOV  [SI], DL
CODE    ENDS
        END  START

```

英文大写字母的 ASCII 码表

序号	字母	MEMORY	
0	A	41H(A')	TABLE
1	B	42H(B')	
2	C	43H(C')	
3	D	44H(D')	
4	E	45H(E')	
5	F	46H(F')	
⋮	⋮	⋮	
⋮	⋮	⋮	
25	Z	5AH(Z')	

10.

分析: 英文大写字母的 ASCII 码值的地址为 TABLE 表的首地址与英文大写字母在 ASCII 码表中的序号之和。可以用 XLAT 指令, 也可以不用。

采用 XLAT 指令, 程序如下:

```

DSEG    SEGMENT
TABLE   DB  41H, 42H, ..., 5AH      ; 定义 ASCII 表
DATA    DB  ' D '
DATB    DB  ?
DSEG    ENDS
CSEG    SEGMENT
        ASSUME  CS: CSEG, DS: DSEG
START:  MOV  AX, DSEG
        MOV  DS, AX
        MOV  BX, OFFSET TABLE      ; 表的首地址偏移量送到 BX
        MOV  AL, DATA
        SUB  AL, 41H                  ; D 对应的序号 3 送到 AL
        XLAT                          ; AL=44H (D 的 ASCII 码)

```




```

        MOV  DATB, AL
        MOV  AH, 4CH
        INT  21H
CSEG   ENDS
        END  START

```

不采用 XLAT 指令，程序如下：

```

DSEG   SEGMENT
TABLE  DB  41H, 42H, ..., 5AH      ; 定义 ASCII 表
DATA   DB  ' D '
DATB   DB  ?
DSEG   ENDS
CSEG   SEGMENT
        ASSUME CS: CSEG, DS: DSEG
START:  MOV  AX, DSEG
        MOV  DS, AX
        MOV  BX, OFFSET TABLE    ; 表的首地址偏移量送到 BX
        MOV  AH, 0
        MOV  AL, DATA
        SUB  AL, 41H
        ADD  BX, AX
        MOV  AL, [BX]
        MOV  DATB, AL
        MOV  AH, 4CH
        INT  21H
CSEG   ENDS
        END  START

```

11.

```

DATA   SEGMENT
DAT1   DW  200
DAT2   DW  1000H
MAX    DW  0
MIN    DB  0
DATA   ENDS
CSEG   SEGMENT
        ASSUME CS: CSEG, DS: DATA
START:  MOV  AX, DATA
        MOV  DS, AX
        MOV  AX, DAT1
        CMP  AX, DAT2
        JG  GREAT
        JZ  ZERO
        MOV  MIN, AX
        JMP  DONE
ZERO:   MOV  AX, 0
GREAT:  MOV  MAX, AX
        JMP  DONE
DONE:   MOV  AH, 4CH
        INT  21H

```

```

CSEG      ENDS
          END START

```

11. 求三个无符号数中的最小值，只要把三个数两两比较，用 JL/JNL/JNG/JG。

```

DSEG      SEGMENT
          STRING DW 5678H, 9743H, 3567H
          MIN DW ?
DSEG      ENDS
CSEG      SEGMENT
          ASSUME CS: CSEG, DS: DSEG
START:    MOV AX, DSEG
          MOV DS, AX
          MOV SI, OFFSET STRING
          MOV AX, [SI]
          MOV BX, [SI+2]
          CMP AX, BX
          JLE MAX1
          MOV AX, BX ; 大数送到 AX
MAX1:     CMP AX, [SI+4]
          JLE FMAX2
          MOV AX, [SI+4]
MAX2:     MOV MIN, AX
          MOV AH, 4CH
          INT 21H
CSEG      ENDS
          END START

```

13.

```

DATA      SEGMENT
BLOCK     DB -23, 345, -1234, ....., 567
MAX       DB ?
MIN       DB ?
DATA      ENDS
CODE      SEGMENT
          ASSUME CS: CODE, DS: DATA
START:    MOV AX, DATA
          MOV DS, AX
          LEA SI, BLOCK
          MOV CX, 100
          MOV AL, [SI]
          INC SI
          MOV PMAX, AL
          MOV NMIN, AL
          DEC CX
AGAIN:    MOV AL, [SI]
          INC SI
          CMP AL, MAX
          JG GREAT
          CMP AL, MIN

```

```

        JL  LESS
        JMP GOON
GREAT:  MOV  MAX, AL
        JMP GOON
LESS:   MOV  MIN, AL
GOON:   DEC  CX
        JNZ AGAIN
        MOV  AH, 4CH
        INT  21H
CODE    ENDS
        END  START

```

14.

第 1 种解法:

```

CODE    SEGMENT
        ASSUME CS: CODE
START:  MOV  CX, 0
AGAIN:  CMP  BX, 0
        JZ  STOP
        SHL BX, 1
        JC  AGAIN
        INC CX
        JMP AGAIN
STOP:   MOV  AH, 4CH
        INT 21H
CODE    ENDS
        END  START

```

第 2 种解法:

```

CODE    SEGMENT
        ASSUME CS: CODE
START:  MOV  CX, 0
NEXT:   AND  AX, AX
        JZ  ST0
        SHL AX, 1
        JC  NEXT
        INC CX
        JMP NEXT
        MOV AH, 4CH
        INT 21H
CODE    ENDS
        END  START

```

第 3 种解法:

```

DATA    SEGMENT
DAT     DW  30
CONUT   DB  16
DAT1    DB  ?

```

```

DATA     ENDS
CODE     SEGMENT
        ASSUME CS: CODE, ,DS: DATA
START:   MOV  AX, DATA
        MOV  DS, AX
        MOV  CL, 0
        MOV  AX, DAT
        MOV  CH, COUNT
AGAIN:   ROL  AX, 1
        JNC  NEXT
        INC  CL
NEXT:    DEC  CH
        JNZ  AGAIN
        MOV  DAT1, CL
        MOV  AH, 4CH
        INT  21H
CODE     ENDS
        END  START

```

15.

```

DATA     SEGMENT
BLOCK    DB  28, 52, 77, 92, 83, ..., 56
COUNT   EQU  $ - BLOCK
ODD      DB  ?
DATA     ENDS
CODE     SEGMENT
        ASSUME CS: CODE, DS: DATA
START:   MOV  AX, DATA
        MOV  DS, AX
        MOV  CX, COUNT
        MOV  AH, 255
        MOV  SI, OFFSET BLOCK
AGAIN:   MOV  AL, [SI]
        TEST AL, 01H
        JZ  NEXT
        CMP  AL, AH
        JAE NEXT
        MOV  AH, AL
NEXT:    INC  SI
        DEC  CX
        JNZ  AGAIN
        MOV  ODD, AH
        MOV  AH, 4CH
        INT  21H
CODE     ENDS
        END  START

```

16. 分析：根据题意，被累加的自然数的个数事先是未知的，也就是说，循环的次数是未知的，因此不能用计数器方法控制循环。但题目中给定一个重要条件，即累加和大于 1000 则停止累加，因此，可以根据这一条件控制循环。我们用 CX 寄存器统计自然数的个数，用 AX 寄存器存放累加和，用 BX 寄存器存放每次取得的自然数。

```

DATA    SEGMENT
COUNT  DW    ?
SUM      DW    ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS: CODE, DS: DATA
START:  MOV  AX, DATA
        MOV  DS, AX
        MOV  AX, 0
        MOV  BX, 0
        MOV  CX, 0
AGAIN:  INC  BX
        ADD  AX, BX
        INC  CX
        CMP  AX, 1000
        JBE  AGAIN
        MOV  COUNT, CX
        MOV  SUM, AX
CODE    ENDS
        END  START

```

17.

```

        LEA  SI, BUFFER
        LEA  DI, ASC
        MOV  CX, 50
NEXT:   MOV  AL, [SI]
        MOV  BL, AL
        AND  AL, 0FH
        ADD  AL, 30H
        MOV  [DI], AL
        INC  DI
        MOV  AL, BL
        PUSH CX
        MOV  CL, 4
        SHR  AL, CL
        POP  CX
        ADD  AL, 30H
        MOV  [DI], AL
        INC  DI
        INC  SI
        LOOP NEXT
        HLT

```

17. 分析：根据公式

$$n^2=1+3+5+\dots+(2n-1)$$

从 1 开始连续 n 个奇数之和是 n 的平方值 N 。

$$1+3+5+7+9=25, n=5, N=n^2=25$$

已知 N ，求 N 的方根 n ，则可从 N 中依次减去从 1 开始的连续奇数，直到 N 为 0 时，减去奇数的个数 n 即为 N 的平方根。

此题循环次数是未知的，只能以 N 不断地减去连续奇数后是否为 0 来作为控制循环的条件。

两个问题需要解决：一是如何形成从 1 开始的连续奇数；二是记录减去奇数的次数。

奇数可以通过 $2n+1$ 来形成，用一个寄存器保存 n ，使之按 0、1、2、……的规律变化，将其各个状态乘 2 加 1，就可以形成从 1 开始的连续奇数，用计数器记录减去奇数的次数，初值置为 0，每当减去一个奇数就增 1，当减去某一个奇数后差为 0，减去的次数就是给定数的平方根，则该计数器的内容即为平方根。

```
DSEG      SEGMENT
DATA      DW  13924
ROOT      DW  0
DAEG      ENDS
CSEG      SEGMENT
          ASSUME  CS: CSEG, DS: DSEG
START:    MOV  AX, DSEG
          MOV  DS, AX
          XOR  CX, CX                ; 计数器清 0
          XOR  AX, AX                ; 设 n 的初值为 0
          MOV  DX, DATA            ; 被开方数 DX
AGAIN:    AND  DX, DX                ; 被开方数为“0”吗
          JZ   LROOT                ; 为“0”，转
          MOV  BX, AX                ; n 值送到 BX
          SHL  BX, 1                 ; 2n
          INC  BX                     ; 形成奇数 2n+1
          SUB  DX, BX
          INC  CX                     ; 计数器值增 1
          INC  AX                     ; n 值增 1
          JMP  AGAIN
LROOT:    MOV  ROOT, CX              ; 存结果
          MOV  AH, 4CH
          INT  21H
CSEG      ENDS
          END   START
```

另一解法：

```
DSEG      SEGMENT
DATA      DW  13924
ROOT      DW  0
DAEG      ENDS
CSEG      SEGMENT
          ASSUME  CS: CSEG, DS: DSEG
```

```

START:  MOV  AX, DSEG
        MOV  DS, AX
        MOV  BX, 1
        XOR  CX, CX                ; 计数器清 0
        MOV  DX, DATA            ; 被开方数 DX
AGAIN:  AND  DX, DX                ; 被开方数为“0”吗?
        JZ   LROOD                ; 为“0”，转
        SUB  DX, BX
        INC  CX                    ; 计数器值增 1
        ADD  BX, 2
        JMP  AGAIN
LROOT:  MOV  ROOT, CX             ; 存结果
        MOV  AH, 4CH
        INT  21H
CSEG    ENDS
        END  START

```

19.

```

DSEG    SEGMENT
DAT1    DB  07H
DAT2    DB  0BH
DATA    DB  0
DSEG    ENDS
CSEG    SEGMENT
        ASSUME CS: CSEG, DS: DSEG
START:  MOV  AX, DSEG
        MOV  DS, AX
        MOV  AL, DAT1
        MOV  CL, 04H
        SHL  AL, CL
        OR   AL, DAT2
        MOV  DATA, AL
        MOV  AH, 4CH
        INT  21H
CSEG    ENDS
        END  START

```

20.



```

DSEG    SEGMENT
DATA    DB 05H
BUF     DB ?
DSEG    ENDS
CSEG    SEGMENT
        ASSUME CS: CSEG, DS: DSEG
START:  MOV AX, DSEG
        MOV DS, AX
        MOV AL, DATA
        CMP AL, 0AH
        JC  AMA
        ADD AL, 7

AMA:    ADD AL, 30H
        MOV BUF, AL
        MOV AH, 4CH
        INT 21H
CSEG    ENDS
        END STSRT

```

21.

```

DATA    SEGMENT
BLOCK1  DB 'HELLO '
BLOCK2  DB 10 DUP(?)
BCD     DB 56
COUNT  EQU $-BLOCK1
CNT     EQU 30
D       DW BLOCK2
DATA    ENDS

```

第五章

参考答案

一、选择题

1~5: DBBCB 6~10: DABDA
 11~15: CACAA 16~20: DBCCC
 21~25: CCBBC 26~30: BABCD
 31~35: ABCCD 36~37: BC

二、填空题

1. 主存储器
2. RAM 和 ROM
3. CPU、CPU
4. RAM 内的信息便完全丢失
5. 二进制

6. 只能读出而不能写入
7. 主存
8. 16
9. 双向、单向
10. 随机访问存储器、只读存储器

三、简答题

1. $(\text{BBFFFH} - 40000\text{H} + 1) / 1024 = 496\text{KB}$ 。
2. (1) 32 片, 16 组, 13 根地址线, 9 根;
(2) 128 片, 16 组, 16 根地址线, 12 根;
(3) 32 片, 32 组, 15 根地址线, 10 根。
3. (1) 8KB; (2) 16 片。
4. (1) 9 条地址线, 1 条数据 I/O 线;
(2) 14 条地址线, 4 条数据 I/O 线;
(3) 15 条地址线, 8 条数据 I/O 线。
5. 因为存储容量取决于地址线的位数。N 条地址线可以产生 2^N 个地址编码, 也就是存储容量为 2^N 个存储器单元, 所以存储容量不可能出现非 2 的指数关系的数。
6. 片内地址线: 用于寻址存储器芯片内的存储单元所需要的地址线。片选地址线: 用于确定某存储器芯片在存储系统空间中的位置所需要的地址线。片选地址用于确定所访问的存储单元在哪个存储器芯片中, 片内地址用于确定所访问的存储单元在存储器芯片中的具体位置。

四、应用题

1. 根据地址范围, 可知寻址能力为 4K, 因此需要 2 片 6116 芯片。具体连线见图 5-13。
2. SRAM 存储芯片的地址范围是: E8000H-EBFFFH。若改为 Y4 时, 地址范围变为 F0000H-F3FFFH; 若改为 Y7 时, 地址范围变为 FC000H-FFFFFH。

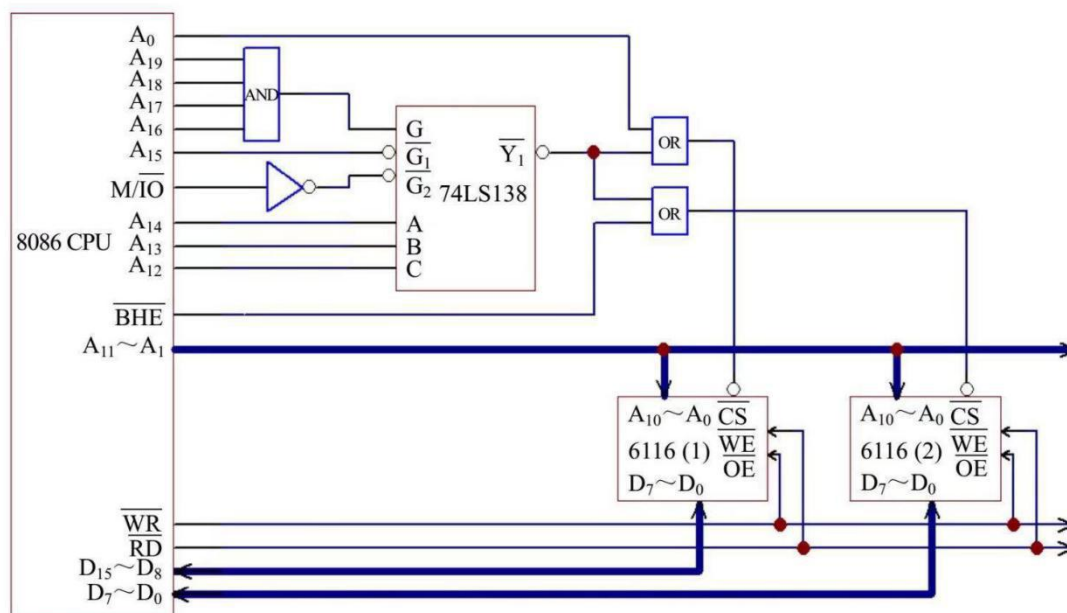


图 5-13 连接图

第六章

参考答案

一、选择题

1. B 2. B 3. C 4. C 5. B 6. A 7. B, C, F 8. C 9. D
10. A 11. A 12. B 13. B 14. B 15. C 16. A 17. D 18. D 19. B
20. A 21. C 22. C 23. B 24. B 25. C 26. C 27. B 28. D 29. C
30. D 31. A 32. B 33. B 34. A 35. C 36. C 37. C 38. A 39. A
40. A 41. C 42. B 43. C 44. D 45. C 46. B 47. C 48. B 49. C
50. D 51. B 52. A 53. ①B, ②D

二、填空题

1. 统一编址，独立编址
2. 大批量，DMA 控制器
3. 程序控制传送方式，中断传送方式，DMA 传送方式
4. 外设，无条件传送方式，查询传送方式，中断传送方式，DMA 传送方式 5. CPU, I/O 端口
6. 直接存储器存取方式
7. 数据总线，地址总线，控制总线
8. 数据信息，地址信息，控制信息，
9. 数字量，模拟量，开关量，脉冲量，
10. 65536, 0~65534
11. I/O 读，20H 号端口地址，IOR, 20H 号端口中的数据
12. T_w 等待状态
13. 速度
14. 并行接口，串行接口
15. 单字节方式，字组方式，连续方式
16. 寄存器组，控制逻辑
17. AX, DX 所寻址的 16 位端口，64K
18. 0FFFF
19. 22
20. 04H, 02H
21. 可屏蔽中断
22. 0000H:0070H, 0000H:0073H, 4
23. 0CH, 4020H:0200H
24. 中断服务程序入口地址
25. 8, 29, 64
26. 80, IR6
27. 外部中断，内部中断
28. 可屏蔽中断，不可屏蔽中断

三、应用题

1. 答案:

INT9 指令存放占 2 个字节，所以执行 INT 9 指令时，CS=0800H，IP=00A0H+ 2=00A2H。中断类型号为 9=1001B

中断向量地址=10 0100B=24H

即 9 号的中断向量存放在 24H、25H、26H、27H 等 4 个字节单元中，前两个单元内容是偏移地址送 IP，后两个单元内容是段地址送 CS。

所以, SS=0500H, SP=00FAH, FR0040H, CS=1000H, IP=0060H

执行 INT 9 指令, CPU 完成下列操作:

标志入栈: TF 和 IF 清零, 这时 FR=0040H

断电入栈: 栈顶的 3 个字按入栈顺序是 0240H, 0800H, 00A2H, SP=0100H-6=00FAH 计算中断

向量地址: 将中断向量送入 CS 和 IP

2. 答案:

```
MOV AL, 1BH
OUT 94H, AL ; 写 ICW1
MOV AL, 60H
OUT 93H, AL ; 写 ICW2
MOV AL, 13H
OUT 93H, AL ; 写 ICW4
```

3. 答案:

(1) 非自动结束

(2) 8AH

(3) 答: IR1 不能引起 CPU 中断, IR2 能引起 CPU 中断。(分析: OCW1 的 D1 和 D2, 是不屏蔽, 开放; 是 1 屏蔽, 禁止。)

4. 分析: 中断向量地址=中断类型号×4, 可以用二进制表示的中断类型号左移 2 位。

反过来, 中断类型号=用二进制表示的中断向量地址逻辑右移 2 位。

答案:

(1) 由 0040H=01000000B

右移后=00010000B=10H=16D

(2) 中断向量即中断服务程序的起始地址。

逻辑地址 CS: IP=5678H: 1234H

5. 答:

因为 IRR 的 8 个输入端分别接 8 个中断源, 所以只要某位有中断请求, 即 IRR 的某端 IRR_i 由低电平变为高电平时, 相应的 IRR 位置 1, 即 IRR_i=1, 直到中断请求已被响应, IRR 相应位复位。

IRR=0FFH, 说明 8 个中断源 IR₇~IR₀ 都有中断请求(由低电平变为高电平), 但没有一个被响应。

6. 答: 当有中断请求被 CPU 响应, 在 CPU 响应中断后发来第一个中断响应脉冲 INTA 时, 将对应的 ISR 相应位置 1, 直到结束中断或有中断结束命令才复位。

ISR=0FFH 是中断请求的最特殊情况, 即 8 个中断源依次进行中断请求, 且最先申请的是最低优先权的中断源。向最后申请的最高优先级的中断源, 并且中断处理都没有结束。

第七章

参考答案

一、选择题

1. C 2. D 3. B 4. C 5. C 6. B 7. B 8. D 9. C 10. B 11. A 12. C 13. B 14. C 15. D
16. B 17. A 18. D 19. A 20. D 21. C 22. D 23. D 24. A 25. D 26. D 27. D 28. B 29.
A 30. D 31. B 32. C 33. C 34. B 35. B 36. C 37. C 38. D 39. C 40. D 41. C 42. B 43.
C 44. D 45. C 46. B 47. C 48. D 49. D 50. B

二、填空题

1. 控制寄存器, 计数通道 2. 方式 3, 1000 3. 3, 16, 6
4. 3, 15 5. 3, 4, /CS、A1 和 A0 6. 控制口, 11B
7. 3, 2500 8. 1ms 9. 方波发生器, 400, 1000
10. 3, 6, 77H 11. 并行接口, 4, 3, 8 12. 3, 2, 1
13. 00001101B 14. 方式 1, 方式 2 15. 控制和状态
16. IBF 17. ± 0.02 18. 12
19. 80H, 1.875V 20. $1/256 \approx 3.9\%$, $5V/256 \approx 20mV$
21. $5V/1024 \approx 4.88 mV$ 22. 10 位 23. 33H

三、应用题

1. 解: 要产生周期性的中断信号, 可选择方式 2。计数初值为 $n=10ms \times 2MHz=20000=4E20H$
初始化程序为

```
MOV AL, 01110100B
OUT 83H, AL
MOV AL, 20H
OUT 81H, AL
MOV AL, 4EH
OUT 81H, AL
```

2. 解: 由图可知:

端口地址为 310H、312H、314H、316H, 3 通道所用时钟脉冲频率为 2MHz。

分析: 通道工作于方式 3, 时间常数 $N0=2MHz/2kHz=1000$;

通道 1 工作于方式 1, 时间常数 $N1=2000$;

通道 2 工作于方式 5, 时间常数 $N2=26=0026H(BCD)$ 。

初始化编程:

；通道 0 初始化程序

```
MOV    DX, 316H
MOV    AL, 00100111B
OUT    DX, AL
MOV    DX, 310H
MOV    AL, 10H
OUT    DX, AL
```

；通道 1 初始化程序

```
MOV    DX, 316H
MOV    AL, 01100011B
OUT    DX, AL
MOV    DX, 312H
MOV    AL, 20H
OUT    DX, AL
```

；通道 2 初始化程序

```
MOV    DX, 316H
MOV    AL, 10011011B
OUT    DX, AL
MOV    DX, 314H
MOV    AL, 26H
OUT    DX, AL
```

3. 解：

(1) 8255A 的端口地址为 F4H~F7H, 8253 的端口地址为 F0H~F3H。

(2) 8253 的初始化程序：

```
MOV AL, 2DH OUT 0F3H, AL MOV AL, 20H OUT 0F0H, AL
```

(3) 8255 的初始化程序：

```
MOV AL, 82H OUT 0F7H, AL
```

4. 解：根据接口图可知，8255 的地址为 200H~203H。(1) 8255 的初始化程序为

```
MOV    DX, 203H
MOV    AL, 10010001B
OUT    DX, AL
```

(2) 打印控制程序为

```

        PUSH  AX
        MOV  DX, 202H
WAIT:   IN   AL, DX
        TEST AL, 08H
        JNZ  WAIT
        MOV  DX, 201H
        POP  AX
        OUT  DX, AL
        MOV  DX, 203H
        MOV  AL, 0EH
        OUT  DX, AL
        MOV  AL, 0FH
        OUT  DX, AL

```

5. 解: $D=2N \times UA / U_{REF} = 256 \times 2.5 / 5 = 128 = 80H$

$UA = D \times U_{REF} / 2N = 96 \times 5 / 256 = 1.875V$

6. 解: D/A 转换的基本原理是应用电阻解码网络, 将 N 位数字量逐位转换为模拟量并求和, 从而实现将 N 位数字量转换为相应的模拟量。输出电压 UA 应与输入数字量 D 成正比:

$UA = (D_0 \times 2^0 + D_1 \times 2^1 + \dots + D_{N-1} \times 2^{N-1}) \times U_{REF} / 2^N = 101 \times 5 / 256 \approx 1.973V$

7. 解: 此程序是重复输出有规则的线性增长的电压。首先输出 0V, 然后每次递增 1LSB 对应的电压, 当输出达到 2FH 对应的电压输出时, 重新从开始输出线性增长的电压 (注意因参考电压为正, 此题电压增长是负增长)。所以, 此题答案是:

波形为负锯齿波, 幅值 $= 2FH \times 1LSB = 2FH \times (-10) / 256 = -1.84V$

