

计算机组成原理

习题串讲



2-1 设某机字长8位（含一位符号位），分别写出下列各二进制数的原码、补码和反码。

0, -0, 0.1000, -0.1000,

0.1111, -0.1111, 1101, -1101

解:

真值	原码	补码	反码
0	00000000	00000000	00000000
-0	10000000		11111111
0.1000	0.10000000	0.10000000	0.10000000
-0.1000	1.10000000	1.10000000	1.01111111

真值	原码	补码	反码
0.1111	0.1111000	0.1111000	0.1111000
-0.1111	1.1111000	1.0001000	1.0000111
1101	00001101	00001101	00001101
-1101	10001101	11110011	11110010

注意：不论是定点小数还是定点整数，
都应补足8位。



2-2 写出下列各数的原码、补码和反码。

$$\frac{7}{16} \quad \frac{4}{16} \quad \frac{1}{16} \quad 0 \quad -\frac{1}{16} \quad -\frac{4}{16} \quad -\frac{7}{16}$$

解: $\frac{7}{16} = 7 \times 2^{-4} = 0.0111$

$$\frac{4}{16} = 4 \times 2^{-4} = 0.0100$$

$$\frac{1}{16} = 1 \times 2^{-4} = 0.0001$$



真值	原码	补码	反码
7/16	0.0111	0.0111	0.0111
4/16	0.0100	0.0100	0.0100
1/16	0.0001	0.0001	0.0001
0	0.0000	0.0000	0.0000
	1.0000		1.1111
-1/16	1.0001	1.1111	1.1110
-4/16	1.0100	1.1100	1.1011
-7/16	1.0111	1.1001	1.1000



2-7 设 $[X]_{\text{补}} = 1.a_1a_2a_3a_4a_5a_6$,

a. 若要 $X > -1/2$, $a_1 \sim a_6$ 要满足什么条件?

b. 若要 $-1/8 \geq X \geq -1/4$, $a_1 \sim a_6$ 要满足什么条件?

解: a. 1.100000 $-1/2$

1.100001 $-31/64$ $a_1=1$

\vdots

1.111111 $-1/64$

$a_2+a_3+a_4+a_5+a_6=1$

习题讲评

b.

1.110000 -1/4

1.110001 -15/64

⋮

1.110111 -9/64

1.111000 -1/8

$$(1) a_1 \cdot a_2 = 1$$

$$a_3 = 0$$

$$(2) a_1 \cdot a_2 \cdot a_3 = 1$$

$$a_4 + a_5 + a_6 = 0$$

2-8 一个 n 位字长的二进制定点整数，其中一位为符号位，分别写出在补码和反码两种情况下：

a.模数

b.最大的正数

c.绝对值最大的负数

d.符号位的权

e.-1的表示形式

f.0的表示形式



n位字长的二进制定点整数，其中一位为符号位。

	补码	反码
模数	$\text{Mod } 2^n$	$\text{Mod } (2^n-1)$
最大的正数	$2^{n-1}-1$	$2^{n-1}-1$
绝对值最大的负数	-2^{n-1}	$-(2^{n-1}-1)$
符号位的权	2^{n-1}	2^{n-1}
-1的表示形式	11111111	11111110
0的表示形式	00000000	11111111

2-9 某机字长16位，问下列几种情况下所能表示数值的范围。

- a. 无符号整数
- b. 用原码表示定点小数
- c. 用补码表示定点小数
- d. 用原码表示定点整数
- e. 用补码表示定点整数



解： a. 无符号整数

字长16位均用来表示数值，即绝对值。

最大数： $(2^{16}-1)$

最小数： 0

$$\therefore 0 \leq X \leq (2^{16}-1)$$



b.用原码表示定点小数

字长16位，包括一位符号位。

最大正数： $(1-2^{-15})$

最小正数： 2^{-15} （零除外）

绝对值最大的负数： $-(1-2^{-15})$

$$\therefore -(1-2^{-15}) \leq X \leq (1-2^{-15})$$

c.用补码表示定点小数

最大正数和最小正数与原码相同。

绝对值最大的负数： -1

$$\therefore -1 \leq X \leq (1-2^{-15})$$



d.用原码表示定点整数

字长16位，包括一位符号位。

最大正数： $(2^{15}-1)$

最小正数：1（零除外）

绝对值最大的负数： $-(2^{15}-1)$

$$\therefore -(2^{15}-1) \leq X \leq (2^{15}-1)$$

e.用补码表示定点整数

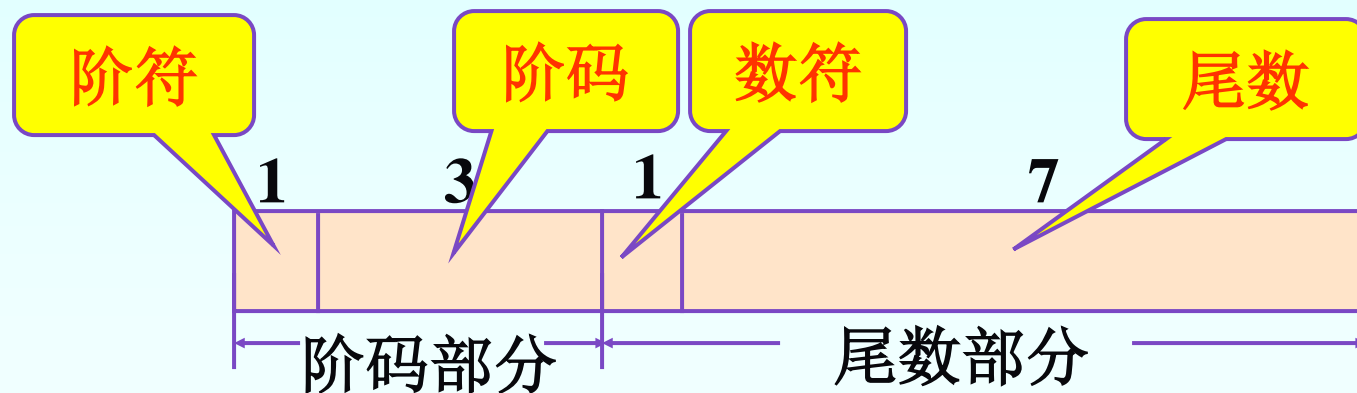
最大正数和最小正数与原码相同。

绝对值最大的负数： -2^{15}

$$\therefore -2^{15} \leq X \leq (2^{15}-1)$$



2-14 某浮点数字长12位，其中阶符1位，阶码3位，数符1位，尾数7位，阶码以2为底，阶码和尾数均用补码表示。它所能表示的最大正数是多少？最小规格化正数是多少？绝对值最大的负数是多少？



阶码部分为定点整数，尾数部分为定点小数。

最大正数：阶码为最大正数，尾数为最大正数。

$$X_{\text{最大正数}} = (1-2^{-7}) \times 2^{2^3-1} = (1-2^{-7}) \times 2^7 = 127$$

最小正数：阶码为绝对值最大负数（最接近零），尾数为最小正数。

$$X_{\text{最小正数}} = 2^{-7} \times 2^{-2^3}$$

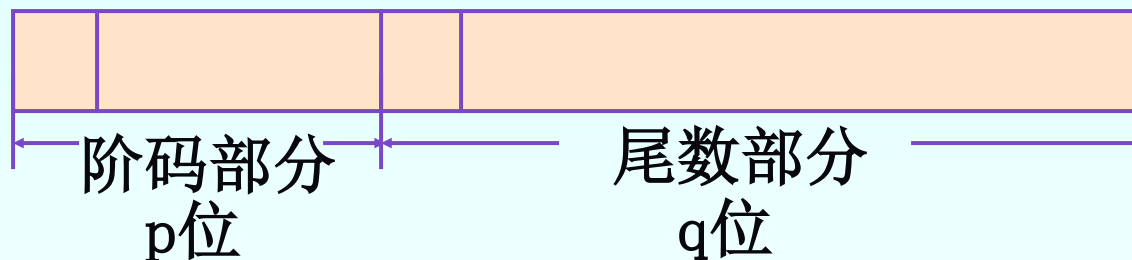
$$X_{\text{最小规格化正数}} = 2^{-1} \times 2^{-2^3} = 2^{-1} \times 2^{-8} = 2^{-9} = 1/512$$

绝对值最大负数：阶码为最大正数，
尾数为绝对值最大负数。

$$X_{\text{绝对值最大负数}} = -1 \times 2^{2^3-1} = -1 \times 2^7 = -128$$



2-16 一浮点数，其阶码部分为 p 位，尾数部分为 q 位，各包含一位符号位，均用补码表示；尾数基值 $r=2$ ，该浮点数格式所能表示数的上限、下限及非零的最小正数是多少，写出表达式。



上限（最大正数）：

$$X_{\text{最大正数}} = (1 - 2^{-(q-1)}) \times 2^{2^{(p-1)}-1}$$

下限（绝对值最大负数）：

$$X_{\text{绝对值最大负数}} = -1 \times 2^{2^{(p-1)}-1}$$

最小正数：

$$X_{\text{最小正数}} = 2^{-(q-1)} \times 2^{-2^{(p-1)}}$$

$$X_{\text{最小规格化正数}} = 2^{-1} \times 2^{-2^{(p-1)}}$$



2-17 若上题尾数基值 $r=16$ ，按上述要求写出表达式。

上限（最大正数）：

$$X_{\text{最大正数}} = (1 - 2^{-(q-1)}) \times 16^{2^{(p-1)}-1}$$

下限（绝对值最大负数）：

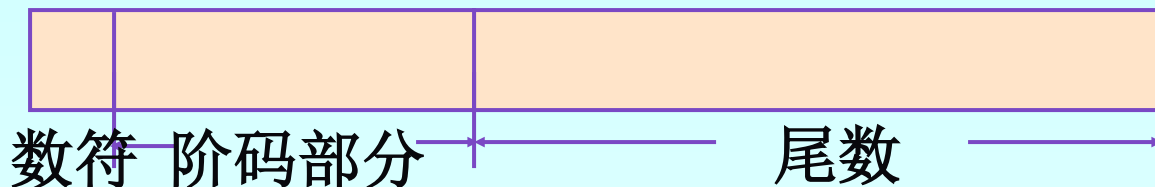
$$X_{\text{绝对值最大负数}} = -1 \times 16^{2^{(p-1)}-1}$$

最小正数：

$$X_{\text{最小正数}} = 2^{-(q-1)} \times 16^{-2^{(p-1)}}$$

$$X_{\text{最小规格化正数}} = 16^{-1} \times 16^{-2^{(p-1)}}$$

2-19 试将 $(-0.1101)_2$ 用**IEEE短实数浮点格式**表示出来。



IEEE32位的短浮点数最高位为数符位，其后8位为阶码，用移码表示，阶码的偏置值为127，其余23位是尾数，用原码表示。为了使尾数部分能表示更多一位有效值，采用隐含尾数最高数位的方法，因此尾数实际上是24位。应注意的是，隐含的1是一位整数（即位权为 2^0 ），在浮点格式中表示出来的23位尾数是纯小数。

$(-0.1101)_2$ 为负数，故数符位为1。

尾数： $(0.1101)_2 = (1.101 \times 2^{-1})_2$

阶码： $-1 + 127 = 126 = (01111110)_2$

(数符1)阶码 IEEE32位短浮点数 尾数:

1,011 1111 0,101 0000 0000 0000 0000 0000

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

B **F** **5** **0** **0** **0** **0** **0**

$\therefore (-0.1101)_2$ 写为IEEE短浮点数形式（十六进制） = **BF500000H**

2-25 已知下面数据块约定：横向校验、纵向校验均为**奇校验**，请指出**至少**有多少位出错。

	a7	a6	a5	a4	a3	a2	a1	a0	校验位
	1	0	0	1	1	0	1	1	→ 0 无错
	0	0	1	1	0	1	0	1	→ 1 无错
	1	1	0	1	0	0	0	0	→ 0 无错
	1	1	1	0	0	0	0	0	→ 0 无错
	0	1	0	0	1	1	1	1	→ 0 无错
校验位	↓ 1	↓ 0	↓ 1	↓ 0	↓ 1	↓ 1	↓ 1	↓ 1	
	出错	无错	无错	无错	无错	无错	无错	出错	

2-26 求有效信息位为01101110的海明校验码。

P_5 D_8 D_7 D_6 D_5 P_4 D_4 D_3 D_2 P_3 D_1 P_2 P_1

$$P_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7 = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$P_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7 = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0$$

$$P_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8 = 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$P_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$P_5 = D_1 \oplus D_2 \oplus D_3 \oplus D_5 \oplus D_6 \oplus D_8 = \\ 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$$

海明校验码 = 101100111001

2-27 设计算机准备传送的信息是：
1010110010001111，生成多项式是
 X^5+X^2+1 ，计算校验位，写出CRC码。

解： $X^5+X^2+1=100101$

余数=10011

CRC码=10101100100011110011



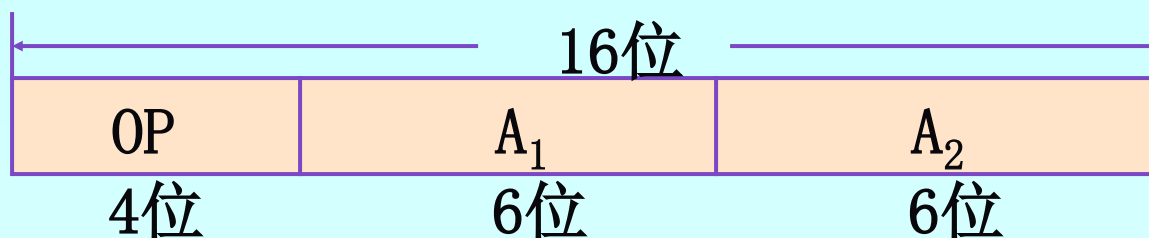
计算机组成原理

习题串讲

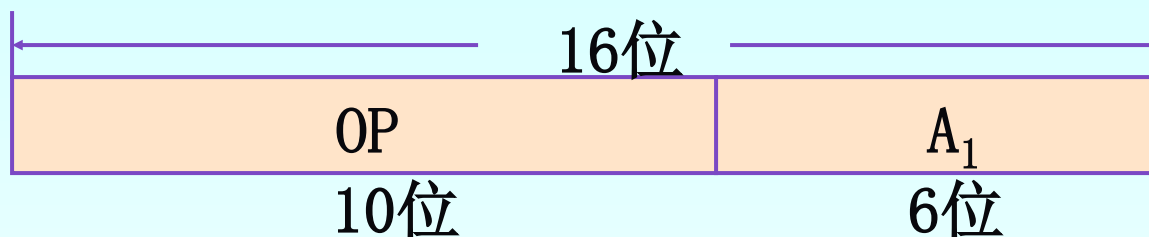


3-3 某机为定长指令字结构，指令长度16位；每个操作数的地址码长6位，指令分为无操作数、单操作数和双操作数三类。若双操作数指令已有 K 种，无操作数指令已有 L 种，问单操作数指令最多可能有多少种？上述三类指令各自允许的最大指令条数是多少？

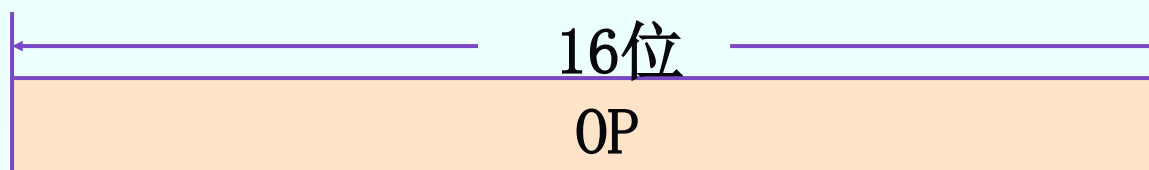




双操作数
指令K



设单操作
数指令X



无操作数
指令L



留给
的扩展窗口留给无地址
的扩展窗口

$$[(2^4 - K) \times 2^6 - X] \times 2^6 = L$$

$$X = (2^4 - K) \times 2^6 - \frac{L}{2^6}$$

上述三类指令各自允许的最大指令条数为：

双操作数： $2^4 - 1$

单操作数： $15 \times 2^6 - 1$ （假设双操作数指令仅1条，为无操作数指令留出1个扩展窗口）

。



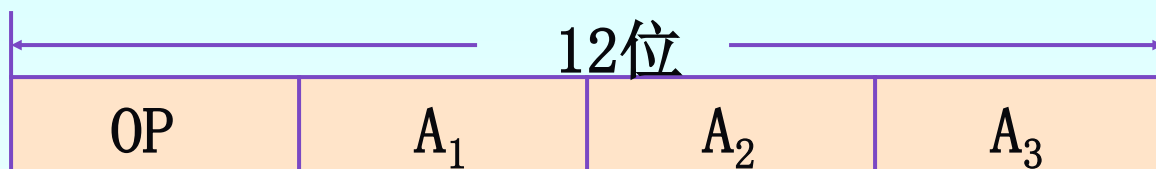
无操作数：假设双操作数指令仅1条，单操作数指令也仅取1条，则无操作数指令的数目具体的条数：

$$2^{16} - 2^{12} - 2^6$$

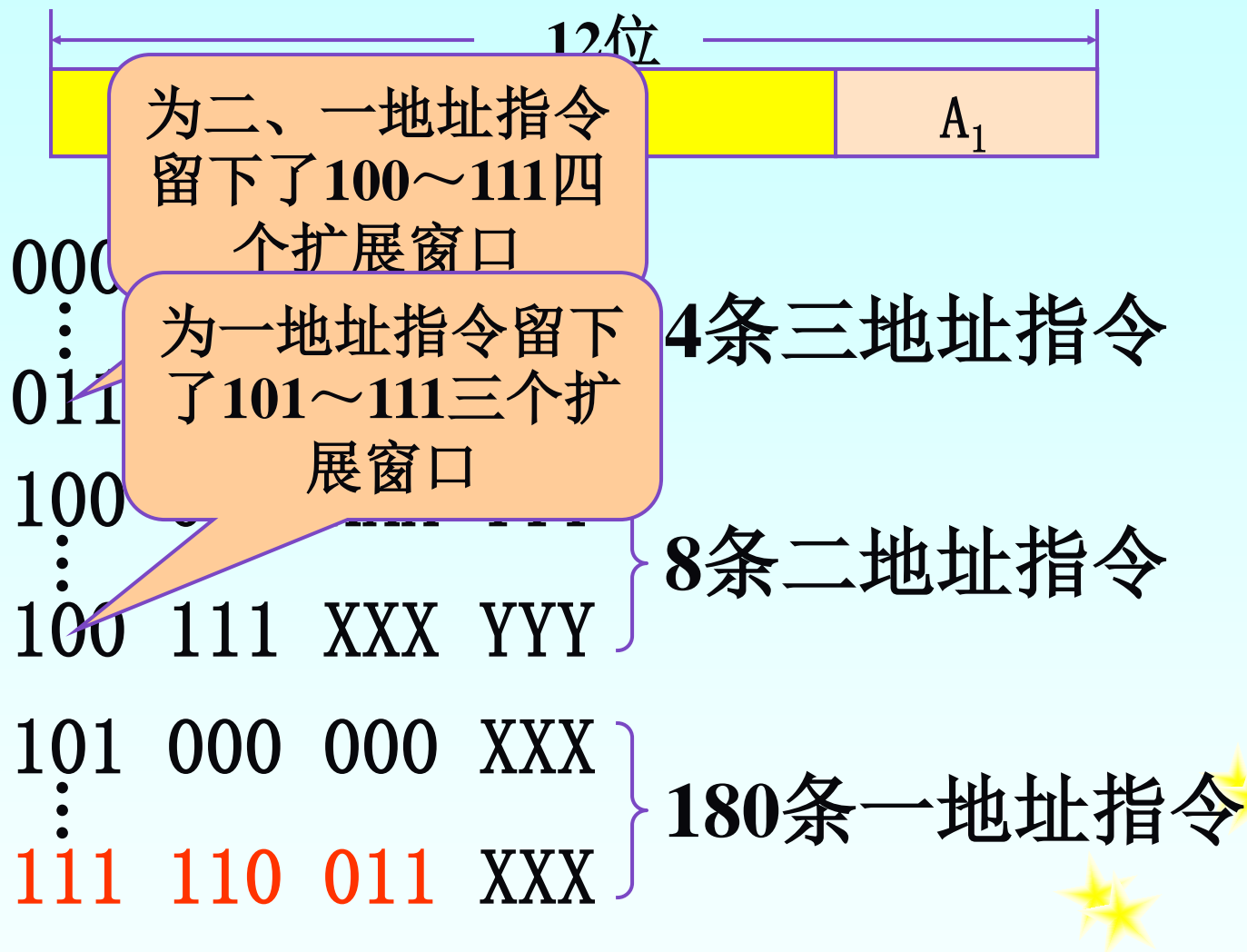
其中 2^{12} 为表示某条二地址指令占用的编码数， 2^6 为表示某条单地址指令占用的编码数。



3-4 设某机为定长指令字结构，**指令长度12位**，**每个地址码占3位**，试提出一种分配方案，使该指令系统包含：4条三地址指令，8条二地址指令，180条单地址指令。



若操作码采用定长编码方法，最多只能表示8条三地址指令，显然不能满足要求。故应采用变长编码方法（扩展操作码法）。



101~111三个扩展窗口可以扩展出192条一地址指令。

$$64 \times 3 = 192$$

现只需要180条一地址指令，故最后12个操作码编码为非法操作码。

111 111 111 最后一个操作码编码

— 1 100 12个非法操作码

111 110 011 最后一个指令的操作码编码



3-10 某机字长为16位，主存容量为64K字，指令格式为单字长单地址，共有64条指令。试说明：

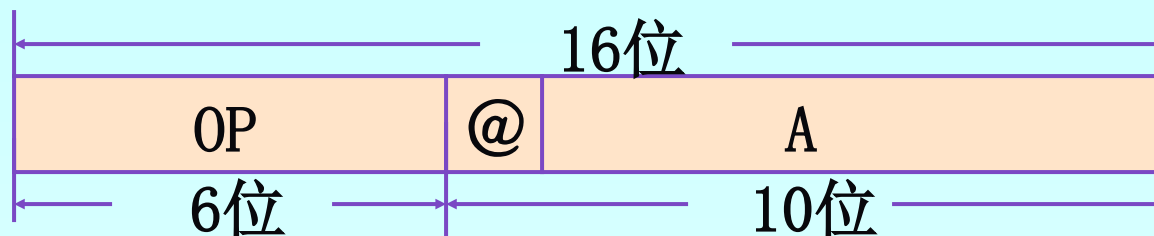
a.若只采用直接寻址方式，指令能访问多少主存单元？

b.为扩充指令的寻址范围，可采用直接/间接寻址方式，若只增加一位直接/间接标志，指令可寻址范围为多少？指令直接寻址的范围为多少？

c.采用页面寻址方式，若只增加一位Z/C（零页/现行页）标志，指令寻址范围为多少？指令直接寻址范围为多少？

d.采用b、c两种方式结合，指令的寻址范围为多少？指令直接寻址范围为多少？

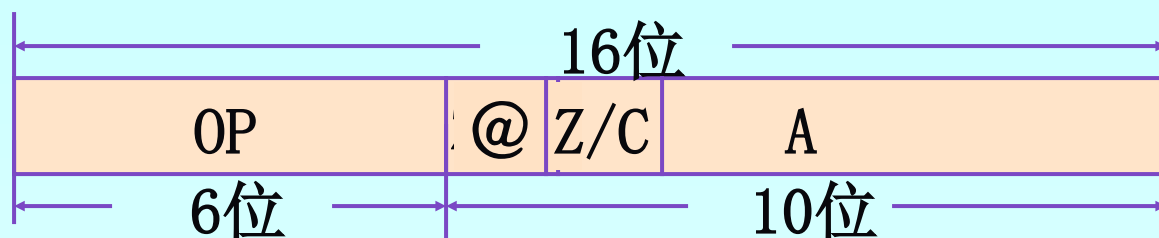




因机器共64条指令，操作码占6位，其余部分为地址码或标志位。

a.若只采用直接寻址方式，地址码部分为10位，指令能访问的主存单元数为 $2^{10}=1K$ 字。

b.若采用直接/间接寻址方式，将增加了一位直接/间接标志，地址码部分为9位，指令直接寻址的范围为 $2^9=0.5K$ 字，指令可寻址范围为整个主存空间 $2^{16}=64K$ 字。



c.若采用页面寻址方式，将增加一位Z/C（零页/现行页）标志，所以指令直接寻址范围仍为 $2^9=0.5K$ 字，指令寻址范围仍为 $2^{16}=64K$ 字。

d.采用b、c两种方式结合，将需要@和Z/C两个标志位，所以指令直接寻址范围为 $2^8=0.25K$ 字，指令的可寻址范围仍为 $2^{16}=64K$ 字。

3-13 计算下列四条指令的**有效地址**（**指令长度为16位**）。

a.000000Q b.100000Q

c.170710Q d.012305Q

假定：上述四条指令均用八进制书写；指令的最左边是一位间址指示位@。（@=0，直接寻址；@=1，间接寻址），且具有多重间访功能；指令的最右边两位为形式地址；主存容量 2^{15} 单元，下表为有关主存单元的内容（八进制）：

地址	内容
00000	100002
00001	046710
00002	054304
00003	100000
00004	102543
00005	100001
00006	063215
00007	077710
00010	100005

a.000000Q

因为指令的最高位为0，故为直接寻址，EA=A。

指令中的形式地址为00Q，因主存容量为 2^{15} 个单元，主存地址共15位，所以EA=00000Q。




地址	内容
00000	100002
00001	046710
00002	054304
00003	100000
00004	102543
00005	100001
00006	063215
00007	077710
00010	100005

b.100000Q

因为指令的最高位为1，故指令为间接寻址。

(00000)=100002，最高位仍为1，继续间接寻址。

(00002)=054304，其最高位为0，表示已找到有效地址，
EA=54304Q。



地址	内容
00000	100002
00001	046710
00002	054304
00003	100000
00004	102543
00005	100001
00006	063215
00007	077710
00010	100005

c.170710Q

因为指令的最高位为1，故指令为间接寻址。

$(00010)=100005$ ，最高位仍为1，继续间接寻址。

$(00005)=100001$ ，最高位仍为1，继续间接寻址。

$(00001)=046710$ ，其最高位为0，表示已找到有效地址， $EA=46710Q$ 。

地址	内容
00000	100002
00001	046710
00002	054304
00003	100000
00004	102543
00005	100001
00006	063215
00007	077710
00010	100005

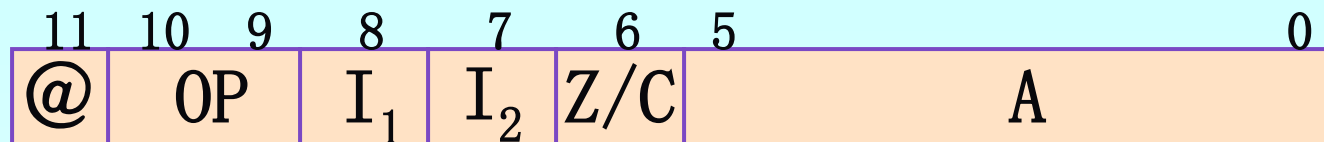
d.012305Q

因为指令的最高位为0，故为直接寻址，EA=A。

指令中的形式地址为05Q，因主存容量为 2^{15} 个单元，地址主存共15位，所以EA=00005Q。



3-14 假定某机的指令格式如下：



其中：

Bit8=1： 变址寄存器 I_1 寻址；

Bit7=1： 变址寄存器 I_2 寻址；

Bit6（零页/现行页寻址）：

Z/C=0， 表示0页面；

Z/C=1， 表示现行页面，即指令所在页面

若主存容量为 2^{12} 个存储单元，分为 2^6 个页面，每个页面有 2^6 个字。

设有关寄存器的内容为：

$(PC)=0340Q$ $(I_1)=1111Q$ $(I_2)=0256Q$

试计算下列指令的有效地址。

a.1046Q

b.2433Q

c.3215Q

d.1111Q



a.1046Q

将用八进制书写的指令变换成二进制。

1	0	4	6
↓	↓	↓	↓
001	000	100	110

因为四个标志位均为0，故为直接寻址。

$$EA=A=100110B=46Q$$

因主存容量为 2^{12} 个存储单元，主存地址共12位，所以EA=0046Q

b.2433Q

将用八进制书写的指令变换成二进制。

$\begin{array}{cccc} 2 & 4 & 3 & 3 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 010 & 100 & 011 & 011 \end{array}$

因为 $\text{Bit}8(I_1)=1$ ，故为变址寄存器1寻址。

$$EA=(I_1)+A=1111Q+33Q=1144Q$$

$$\begin{array}{r}
 001\ 001\ 001\ 001 \\
 + \qquad \qquad \qquad 011\ 011 \\
 \hline
 001\ 001\ 100\ 100
 \end{array}$$



c.3215Q

将用八进制书写的指令变换成二进制。

$\begin{array}{cccc} 3 & 2 & 1 & 5 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 011 & 010 & 001 & 101 \end{array}$

因为Bit7(I_2)=1, 故为变址寄存器2寻址。

$$EA = (I_2) + A = 0256Q + 15Q = 0273Q$$

$$\begin{array}{r}
 000 \ 010 \ 101 \ 110 \\
 + \qquad \qquad \qquad 001 \ 101 \\
 \hline
 000 \ 010 \ 111 \ 011
 \end{array}$$



d.1111Q

将用八进制书写的指令变换成二进制。

1 1 1 1
↓ ↓ ↓ ↓
001 001 001 001

因为Bit6(Z/C)=1，故为当前页寻址。

$EA=(PC)_H // A=03Q // 11Q=0311Q$

000 011 001 001




3-16 设某计算机有变址寻址、间接寻址和相对寻址等寻址方式，当前指令的地址码部分为001AH，正在执行的指令所在地址为1F05H，变址寄存器中的内容为23A0H。

- (1) 当执行取数指令时，如为变址寻址方式，则取出的数是多少？
- (2) 如为间接寻址，取出的数是多少？
- (3) 当执行转移指令时，转移地址是多少？

已知主存部分地址及相应内容如下：

地址	内容	地址	内容
001AH	23A0H	23A0H	2600H
1F05H	241AH	23BAH	1748H
1F1FH	2500H		

(1) 变址: $EA = 23A0H + 001AH = 23BAH$  

$S = (EA) = 1748H$ 

已知主存部分地址及相应内容如下：

地址	内容	地址	内容
001AH	23A0H	23A0H	2600H
1F05H	241AH	23BAH	1748H
1F1FH	2500H		

(2) 间址: $EA = (001AH) = 23A0H$

$S = (EA) = 2600H$



已知主存部分地址及相应内容如下：

地址	内容	地址	内容
001AH	23A0H	23A0H	2600H
1F05H	241AH	23BAH	1748H
1F1FH	2500H		

(3) 转移指令采用相对寻址：

$EA = 1F05H + 001AH = 1F1FH$ ---转移地址

3-18 设相对寻址的转移指令占两个字节，第一个字节是操作码，第二个字节是相对位移量，用补码表示。假设当前转移指令第一字节所在的地址为2000H，且CPU每取一个字节便自动完成 $(PC)+1 \rightarrow PC$ 的操作。试问当执行 JMP^*+8 和 JMP^*-9 指令（*为相对寻址特征）时，转移指令第二字节的内容各为多少？转移的目的地址各是什么？



转移指令第一字节所在的地址为2000H，指令执行过程中自动完成 $(PC)+2=2000H+2=2002H$ 。当执行 JMP^*+8 时，转移指令第二字节的内容为08H，转移地址为200AH。当执行 JMP^*-9 指令时，转移指令第二字节的内容为F7H，转移地址为1FF9H。



3-22 在某些计算机中，调用子程序的方法是这样实现的：转子指令将返回地址存入子程序的第一个字单元，然后从第二个字单元开始执行子程序，请你回答下列问题：

a.为这种方法设计一条从子程序转到主程序的返回指令。

b.在这种情况下，怎么在主、子程序间进行参数的传递？

c.上述方法是否可用于子程序的嵌套？

d.上述方法是否可用于子程序的递归（即某个子程序自己调用自己）？

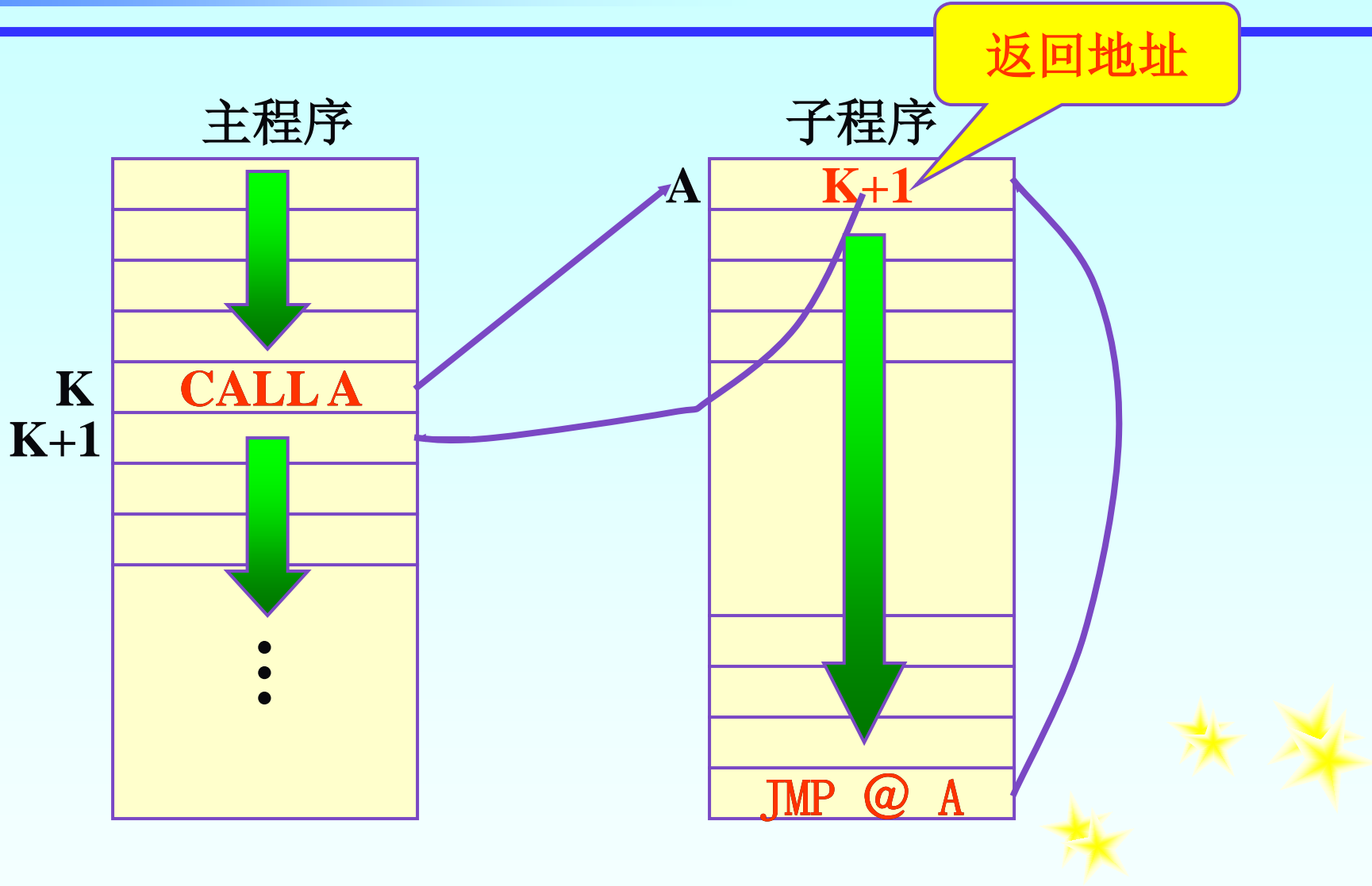
e.如果改用堆栈方法，是否可实现d所提出的问题？



a. 返回指令通常为零地址指令。返回地址保存在堆栈中，执行返回指令时自动从堆栈中弹出。而目前返回地址是保存在子程序的第一个单元中，故此时返回指令不能再是零地址指令了，而应当是一地址指令。如

OP	@	子程序首地址
----	---	--------

间接寻址可找到返回地址，然后无条件转移到返回的位置。



- b.在这种情况下，可利用**寄存器或主存单元**进行主、子程序间的参数传递。
- c.**可以**用于子程序的嵌套（多重转子）。因为每个返回地址都放在调用的子程序的第一个单元中。
- d.**不可以**用于子程序的递归，因为当某个子程序自己调用自己时，第一个单元的内容将被破坏。
- e.如果改用堆栈方法，可以实现子程序的递归，因堆栈具有后进先出的功能。

计算机组成原理

习题串讲



4-1 设计一个9位先行进位加法器，每3位为一组，采用两级先行进位线路。

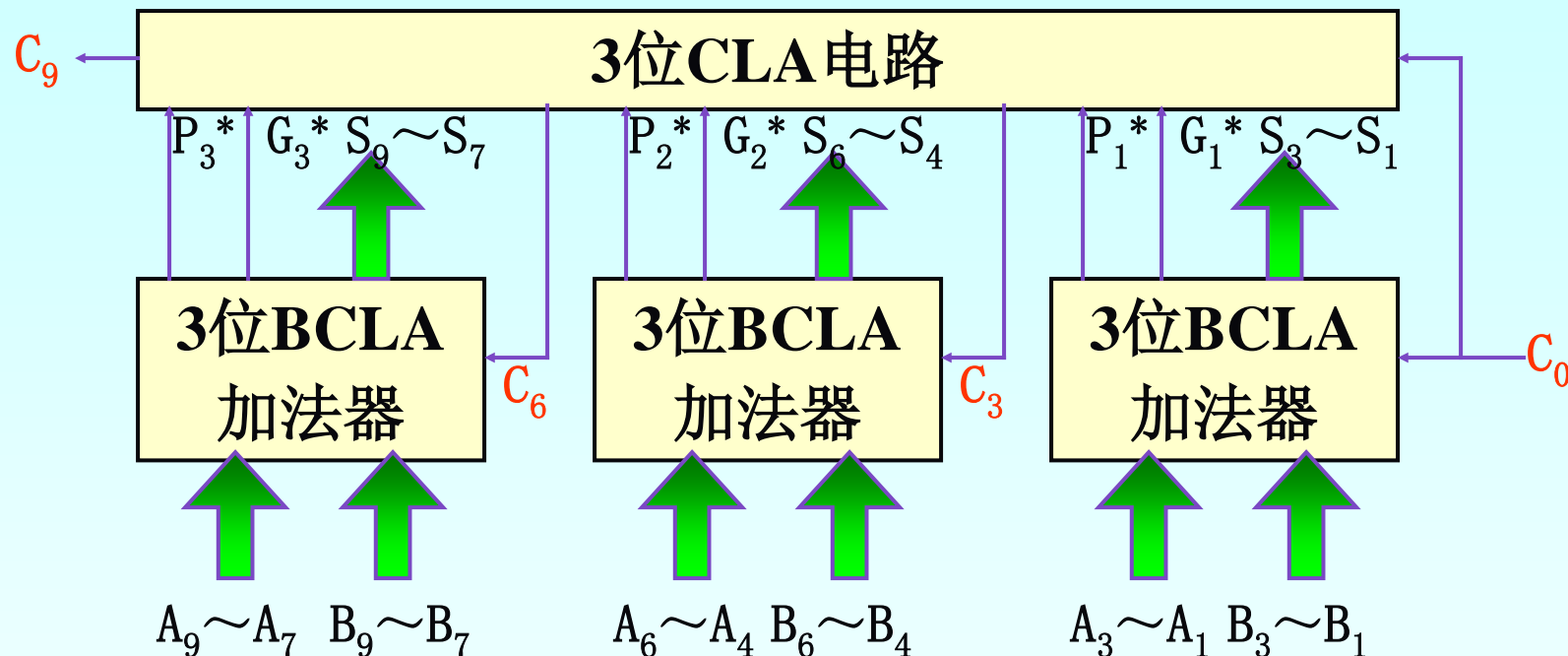
$$C_1 = G_1 + P_1 C_0$$

$$C_2 = G_2 + P_2 G_1 + P_2 P_1 C_0$$

$$C_3 = \underbrace{G_3 + P_3 G_2 + P_3 P_2 G_1}_{G_1^*} + \underbrace{P_3 P_2 P_1}_{P_1^*} C_0 = G_1^* + P_1^* C_0$$

$$C_6 = G_2^* + P_2^* G_1^* + P_2^* P_1^* C_0$$

$$C_9 = G_3^* + P_3^* G_2^* + P_3^* P_2^* G_1^* + P_3^* P_2^* P_1^* C_0$$



CLA: 先行进位

BCLA: 成组先行进位



4-4 已知X和Y，试用它们的变形补码计算出
 $X+Y$ ，并指出结果是否溢出。

a. $X=0.11011$, $Y=0.11111$

b. $X=0.11011$, $Y=-0.10101$

c. $X=-0.10110$, $Y=-0.00001$

d. $X=-0.11011$, $Y=0.11110$



a. $X=0.11011$, $Y=0.11111$

$[X]_{\text{补}}=0.11011$ $[Y]_{\text{补}}=0.11111$

$$\begin{array}{r} 00.11011 \\ + 00.11111 \\ \hline 01.11010 \end{array}$$

正溢

结果溢出错误

b. $X=0.11011$, $Y=-0.10101$

$[X]_{\text{补}}=0.11011$, $[Y]_{\text{补}}=1.01011$

$$\begin{array}{r} 00.11011 \\ + 11.01011 \\ \hline 00.00110 \end{array}$$

$[X+Y]_{\text{补}}=00.00110$

$X+Y=0.00110$

$$\text{c. } X = -0.10110, \quad Y = -0.00001$$

$$[X]_{\text{补}} = 1.01010, \quad [Y]_{\text{补}} = 1.11111$$

$$\begin{array}{r} 11.01010 \\ + 11.11111 \\ \hline \underline{11.01001} \end{array}$$

$$[X+Y]_{\text{补}} = 11.01001$$

$$X+Y = -0.10111$$



习题讲评

$$\text{d. } X = -0.11011, \quad Y = 0.11110$$

$$[X]_{\text{补}} = 1.00101, \quad [Y]_{\text{补}} = 0.11110$$

$$\begin{array}{r} 11.00101 \\ + \quad 00.11110 \\ \hline \underline{00.00011} \end{array}$$

$$[X+Y]_{\text{补}} = 00.00011$$

$$X+Y = \mathbf{0.00011}$$



4-5 已知X和Y，试用它们的变形补码计算出X-Y，并指出结果是否溢出。

a. $X=0.11011$, $Y=-0.11111$

b. $X=0.10111$, $Y=0.11011$

c. $X=0.11011$, $Y=-0.10011$

d. $X=-0.10110$, $Y=-0.00001$



$$\text{a. } X=0.11011, Y=-0.11111$$

$$[X]_{\text{补}}=0.11011, [Y]_{\text{补}}=1.00001, [-Y]_{\text{补}}=0.11111$$

正溢

$$\begin{array}{r} 00.11011 \\ + 00.11111 \\ \hline 01.11010 \end{array}$$

结果溢出错误

$$\text{b. } X=0.10111, Y=0.11011$$

$$[X]_{\text{补}}=0.10111, [Y]_{\text{补}}=0.11011, [-Y]_{\text{补}}=1.00101$$

$$\begin{array}{r} 00.11011 \\ + 11.00101 \\ \hline 11.11100 \end{array}$$

$$[X-Y]_{\text{补}}=11.11100$$

$$X-Y=-0.00100$$

$$\text{c. } X=0.11011, Y=-0.10011$$

$$[X]_{\text{补}}=0.11011, [Y]_{\text{补}}=1.01101, [-Y]_{\text{补}}=0.10011$$

正溢

$$\begin{array}{r} 00.11011 \\ + 00.10011 \\ \hline 01.01110 \end{array}$$

结果溢出错误

$$\text{d. } X=-0.10110, Y=-0.00001$$

$$[X]_{\text{补}}=1.01010, [Y]_{\text{补}}=1.11111, [-Y]_{\text{补}}=0.00001$$

$$\begin{array}{r} 11.01010 \\ + 00.00001 \\ \hline 11.01011 \end{array}$$

$$[X-Y]_{\text{补}}=11.01011$$

$$X-Y=-0.10101$$



4-6 已知: $X=0.1011$, $Y=-0.0101$

求: $[1/2X]_{\text{补}}$, $[1/4X]_{\text{补}}$, $[-X]_{\text{补}}$, $[1/2Y]_{\text{补}}$,
 $[1/4Y]_{\text{补}}$, $[-Y]_{\text{补}}$

$X=0.1011$

$[X]_{\text{补}}=0.1011$, $[1/2X]_{\text{补}}=0.0101$,

$[1/4X]_{\text{补}}=0.0010$, $[-X]_{\text{补}}=1.0101$

$Y=-0.0101$

$[Y]_{\text{补}}=1.1011$, $[1/2Y]_{\text{补}}=1.1101$

$[1/4Y]_{\text{补}}=1.1110$, $[-Y]_{\text{补}}=0.0101$



4-8 分别用原码一位乘法和补码一位乘法计算 $X \times Y$ 。

(1) $X=0.11011$, $Y=-0.11111$

(2) $X=-0.11010$, $Y=-0.01110$

(1) $X \times Y = -0.1101000101$

(2) $X \times Y = 0.0101101100$



4-10 分别用原码加减交替法和补码加减交替法计算 $X \div Y$ 。

(2) $X = -0.10101$, $Y = 0.11011$

(4) $X = -0.10110$, $Y = -0.11011$

(2) 原码除法: $X \div Y = -(0.11000 + \frac{0.11000 \times 2^{-5}}{0.11011})$

补码除法: $X \div Y = -0.11001 + \frac{0.00011 \times 2^{-5}}{0.11011}$

(4) $X = -0.10110$, $Y = -0.11011$

原码除法: $X \div Y = 0.11010 + \frac{0.00010 \times 2^{-5}}{0.11011}$

补码除法: $X \div Y = 0.11011 - \frac{0.11001 \times 2^{-5}}{0.11011}$



4-12 设浮点数的阶码和尾数部分均用补码表示，按照浮点数的运算规则，

(1) $X=2^{101} \times (-0.100010)$, $Y=2^{100} \times (-0.111110)$, 求 $X+Y$, $X-Y$ 。

$$[X]_{\text{浮}} = 0101; 1.011110$$

$$[Y]_{\text{浮}} = 0100; 1.000010$$

1. 求 $X+Y$

(1) 对阶：小阶向大阶看齐。所以

$$[Y]_{\text{浮}} = 0101; 1.100001$$



(2) 尾数相加

$$\begin{array}{r}
 11.011110 \\
 + 11.100001 \\
 \hline
 10.111111
 \end{array}$$

(3) 尾数结果规格化

右规一次。尾数右移一位，阶码加1。

$$[X+Y]_{\text{浮}} = 0110; 1.011111$$

$$X+Y = 2^{110} \times (-0.100001)$$

2. 求 $X-Y$

(1) 对阶：小阶向大阶看齐。所以

$$[Y]_{\text{浮}} = 0101; 1.100001$$



(2) 尾数相减

$$\begin{array}{r}
 [-Y]_{\text{尾补}} = 0.011111 \\
 11.011110 \\
 + 00.011111 \\
 \hline
 11.111101
 \end{array}$$

(3) 尾数结果规格化

左规四次。尾数每左移一位，阶码减1。

$$[X-Y]_{\text{浮}} = 0001; 1.010000$$

$$X-Y = 2^{001} \times (-0.110000)$$



4-13 设浮点数的阶码和尾数部分均用补码表示，按照浮点数的运算规则，计算下列各题：

(1) $X=2^3 \times \frac{13}{16}$, $Y=2^4 \times (-\frac{9}{16})$, 求 $X \times Y$ 。

(2) $X=2^3 \times (-\frac{13}{16})$, $Y=2^5 \times \frac{15}{16}$, 求 $X \div Y$ 。

(1)1) 阶码相加

$$\begin{array}{r} 00011 \\ + 00100 \\ \hline 00111 \end{array}$$

2) 尾数相乘

由补码乘法规则求得： -0.01110101

3) 结果规格化
左规一次，阶码减一。

$$X \times Y = -0.11101010 \times 2^6$$

(2)1) 尾数调整

$$\because |X_{\text{尾数}}| \leq |Y_{\text{尾数}}|$$

\therefore 无需尾数调整

2) 阶码相减

$$\begin{array}{r} 00011 \\ + 11011 \\ \hline 11110 \end{array}$$

$$\text{阶码} = -010$$



3) 尾数相除

由补码除法规则求得：

$$-0.1101 + \frac{-0.1101 \times 2^{-4}}{0.1111}$$

$$\therefore X \div Y = \left(-0.1101 + \frac{-0.1101 \times 2^{-4}}{0.1111} \right) \times 2^{-2}$$



4-15 设计一个一位5421码加法器。

十进制数	5421码 C4S4S3S2S1	校正前的二进制数 C4'S4'S3'S2'S1'	校正与否
0 4	0 0 0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 1 0 0	不校正
5 6 7 8 9	0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0	0 0 1 0 1 1000 0 0 1 1 0 1001 0 0 1 1 1 1010 0 1 0 0 0 1011 0 1 0 0 1 1100	A<5, B<5, +3校正

十进制数	8421码 C4S4S3S2S1	校正前的二进制数 C4'S4'S3'S2'S1'	校正与 否
10	1 0 0 0 0	0 1 1 0 11 0000	A<5, B>5, +3校正
11	1 0 0 0 1	0 1 1 1 01 0001	
12	1 0 0 1 0	0 1 1 1 11 0010	
13	1 0 0 1 1	1 0 0 0 01 0011	
14	1 0 1 0 0	1 0 0 0 11 0100	
15	1 1 0 0 0	1 0 1 0 1	A≥5 B≥5 +3校正
16	1 1 0 0 1	1 0 1 1 0	
17	1 1 0 1 0	1 0 1 1 1	
18	1 1 0 1 1	1 1 0 0 0	
19	1 1 1 0 0	1 1 0 0 1	

①不用校正，结果正确

②当 $A < 5$ ， $B < 5$ ，需+3校正，如：

$$1+4=5, 4+4+1=9$$

当 $A < 5$ ， $B \geq 5$ 或 $A \geq 5$ ， $B < 5$ ，无需校正，
如：

$$\underline{0+5=5}, \quad 4+5=9$$

$$A4B4(S4'+S3'S2'+S3'S1')$$

③当 $A < 5$ ， $B \geq 5$ ，或 $A \geq 5$ ， $B < 5$ ，需+3校正，
如：

$$4+9+1=14$$

当 $A \geq 5$, $B \geq 5$, 无需校正, 如:

$$5+9=14$$

$$(A4 \oplus B4)(C4' + S3'S2' + S3'S1')$$

④ $A \geq 5$, $B \geq 5$, 一定+3校正。

$$A4B4(S4' + S3'S2' + S3'S1')$$

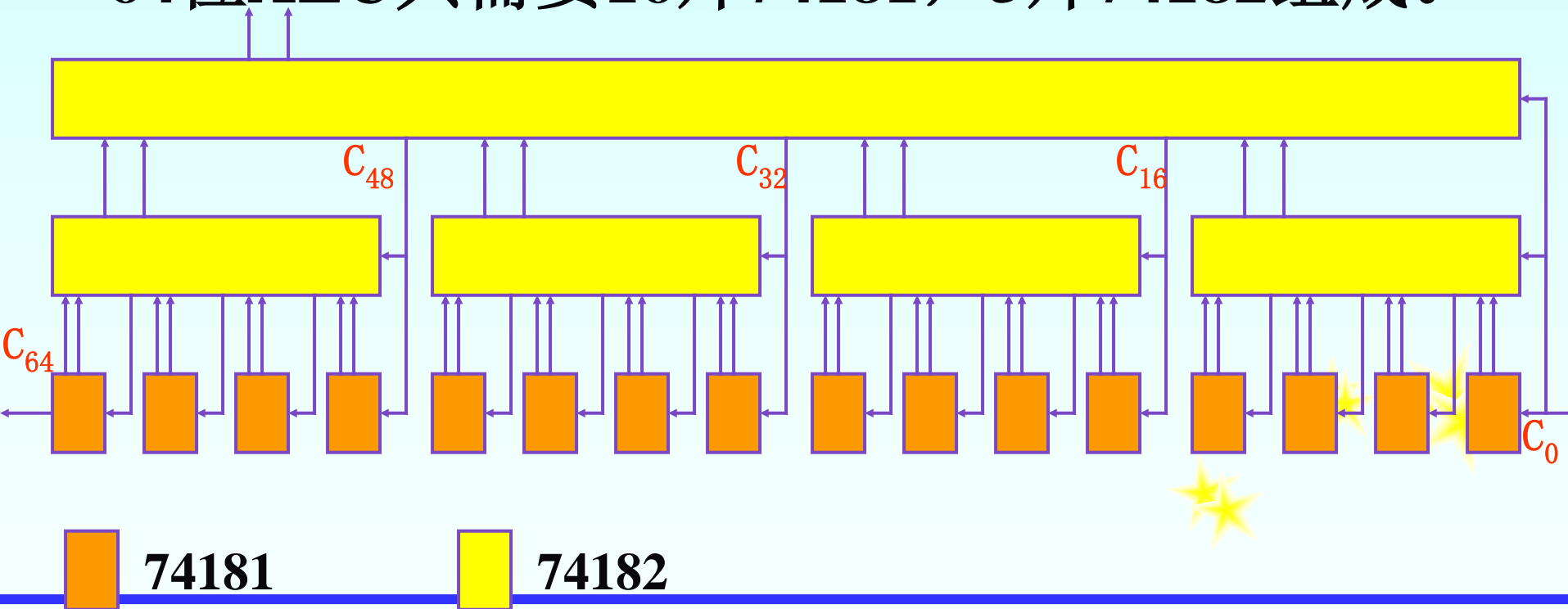
∴校正函数=

$$S3'S2' + S3'S1' + \overline{(A4 \oplus B4)}S4' + (A4 \oplus B4)C4'$$

$$= S3'S2' + S3'S1' + C3'$$

4-16 用74181和74182芯片构成一个64位的ALU，采用多级分组并行进位链（要求速度尽可能快）。

64位ALU共需要16片74181，5片74182组成。



计算机组成原理

习题串讲



5-4 某机存储字长**64**位，主存储器按字节编址，现有四种不同长度的数据：字节、半字（**16**位）、单字（**32**位）、双字（**64**位），请采用一种既节省存储空间，又能保证任一个数据都在单个存取周期中完成读写的方法将不同长度的数据存入主存（采用大端方案）。

（1）写出不同长度数据存放在主存中地址的限定要求（即第一个字节的地址）。

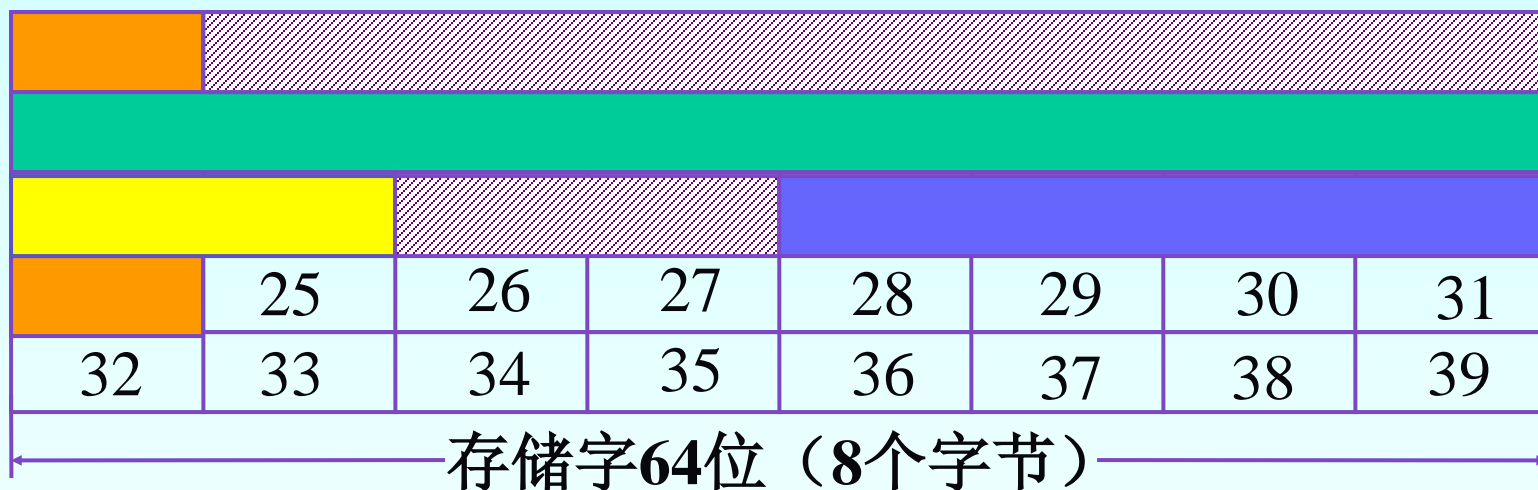
（2）画出将字节、双字、半字、单字、字节这5个数据依次存放在主存中的示意图（不能改变顺序）。

解：（1）边界对齐的数据存放方式对数据的存放位置有下列要求：

- ①字节数据的地址为 $\times \dots \times \times \times \times$ （任意）
- ②半字数据的起始地址为 $\times \dots \times \times \times 0$ （2的整倍数）
- ③单字数据的起始地址为 $\times \dots \times \times 0 0$ （4的整倍数）
- ④双字数据的起始地址为 $\times \dots \times 0 0 0$ （8的整倍数）

5.2 主存储器的组织

(2) 画出将字节、双字、半字、单字、字节这5个数据依次存放在主存中的示意图。



解：（1）边界对齐的数据存放方式对数据的存放位置有下列要求：

- ①字节数据的地址为 $\times \dots \times \times \times \times$ （任意）
- ②半字数据的起始地址为 $\times \dots \times \times \times 0$ （2的整倍数）
- ③单字数据的起始地址为 $\times \dots \times \times 0 0$ （4的整倍数）
- ④双字数据的起始地址为 $\times \dots \times 0 0 0$ （8的整倍数）

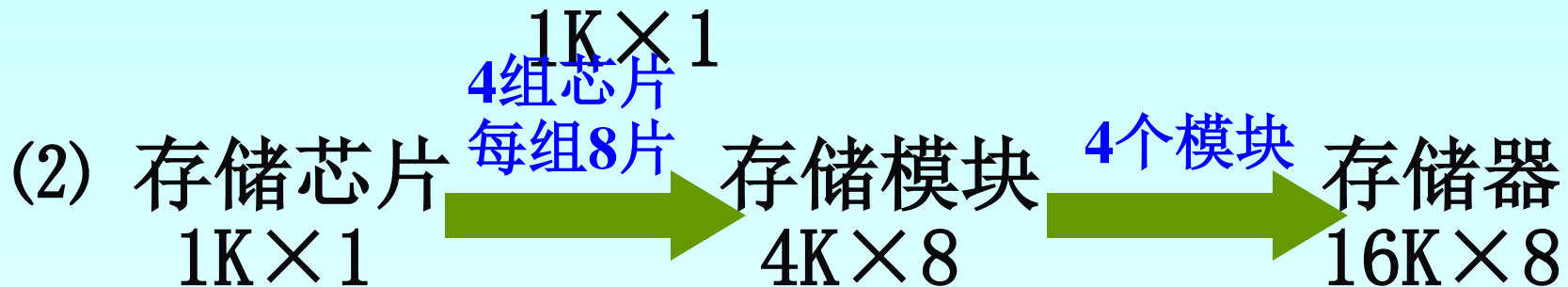
5-10 一个 $1\text{K} \times 8$ 的存储芯片需要多少根地址线、数据输入线和输出线？

$1\text{K} \times 8$ 的存储芯片共需地址线**10**根，数据输入/输出线**8**根。

5-13 现有 1024×1 的存储芯片，若用它组成容量为 $16\text{K} \times 8$ 的存储器。试求：

- (1) 实现该存储器所需的芯片数量。
- (2) 若将这些芯片分装在若干块板上，若每块板的容量为 $4\text{K} \times 8$ 。该存储器所需的地址码总位数是多少？其中几位用于选板？几位用于选片？几位用作片内地址？

(1) 芯片数量 = $\frac{16K \times 8}{1K \times 1} = 128$ (片)



存储器
 $16K \times 8$

需地址线14根

存储芯片
 $1K \times 1$

需地址线10根

∴ 存储器所需的地址码总位数14位，其中2位用于选板（4个模板），2位用于选片（4组芯片），10位用作片内地址。

5-14 已知某机字长8位，现采用半导体存储器作主存，其地址码为16位，若使用1K×4的SRAM芯片组成该机所允许的最大主存空间，并采用存储模板结构形式。

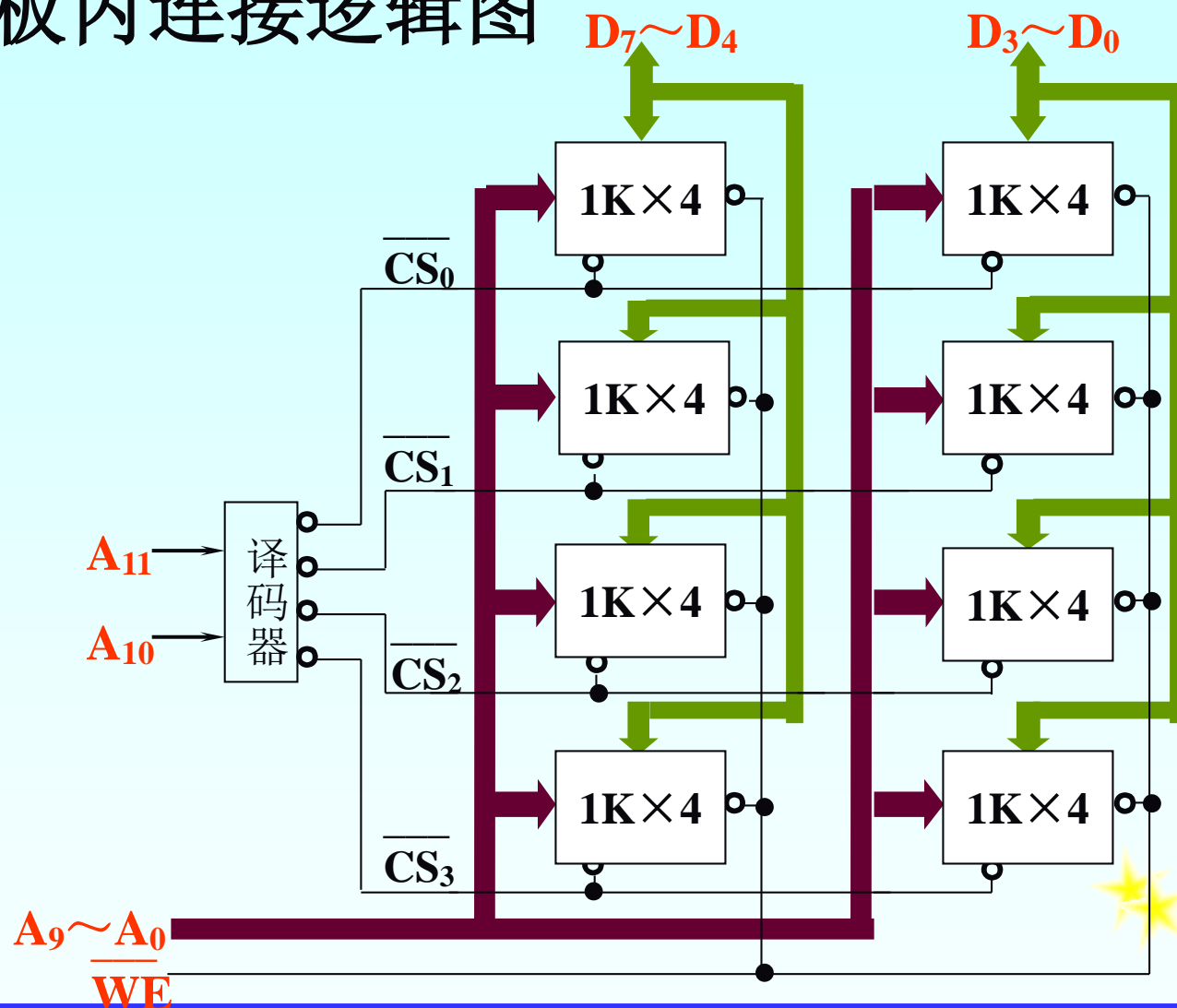
(1) 若每块模板容量为4K×8，共需多少块存储模板？

(2) 画出一个模板内各芯片的连接逻辑图。

$$(1) \text{ 模板数量} = \frac{64\text{K} \times 8}{4\text{K} \times 8} = 16 \text{ (个)}$$



(2) 模板内连接逻辑图



5-16 现有如下存储芯片：2K×1的ROM；4K×1的RAM；8K×1的ROM。若用它们组成容量为16KB的存储器，前4KB为ROM，后12KB为RAM，地址码采用16位。试问：

- (1) 各种存储芯片分别用多少片？**
- (2) 正确选用译码器及门电路，并画出相应的逻辑结构图。**
- (3) 指出有无地址覆盖现象。**



(1)RAM

$$\frac{12\text{K} \times 8}{4\text{K} \times 1} = 3 \times 8 = 24 \text{片}$$

将24片4K×1的RAM先构成3个4KB的芯片组。

ROM不能使用8K×1的芯片，因为它大于ROM应有的空间。

$$\frac{4\text{K} \times 8}{2\text{K} \times 1} = 2 \times 8 = 16 \text{片}$$

将16片2K×1的ROM先构成2个2KB的芯片组。

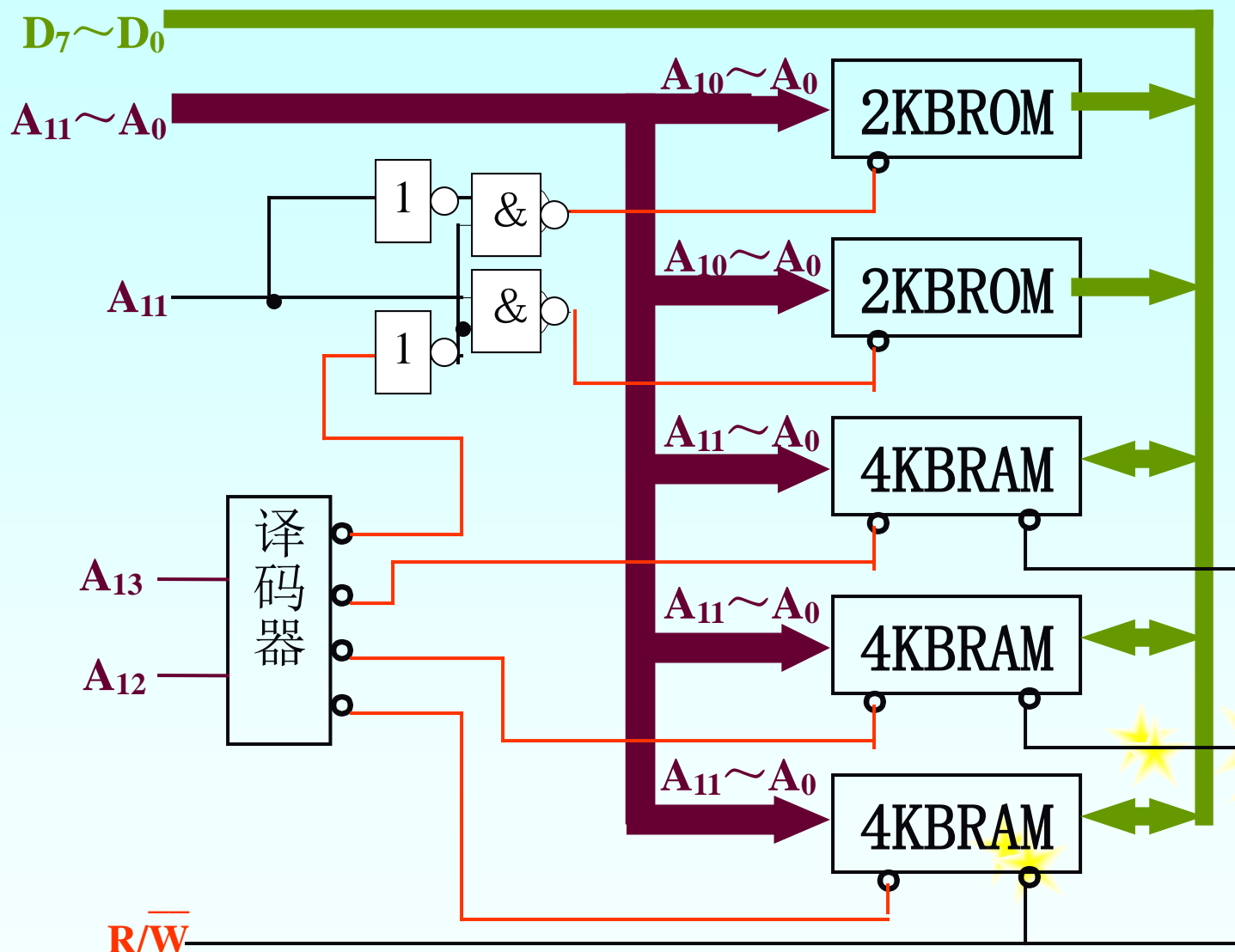
(2)地址分配

A15	A14	A13	A12	A11	A10~A0	
X	X	0	0	0	——	2KB ROM
X	X	0	0	1	——	2KB ROM
X	X	0	1		——	4KB RAM
X	X	1	0		——	4KB RAM
X	X	1	1		——	4KB RAM

(3)有地址覆盖。



习题讲评

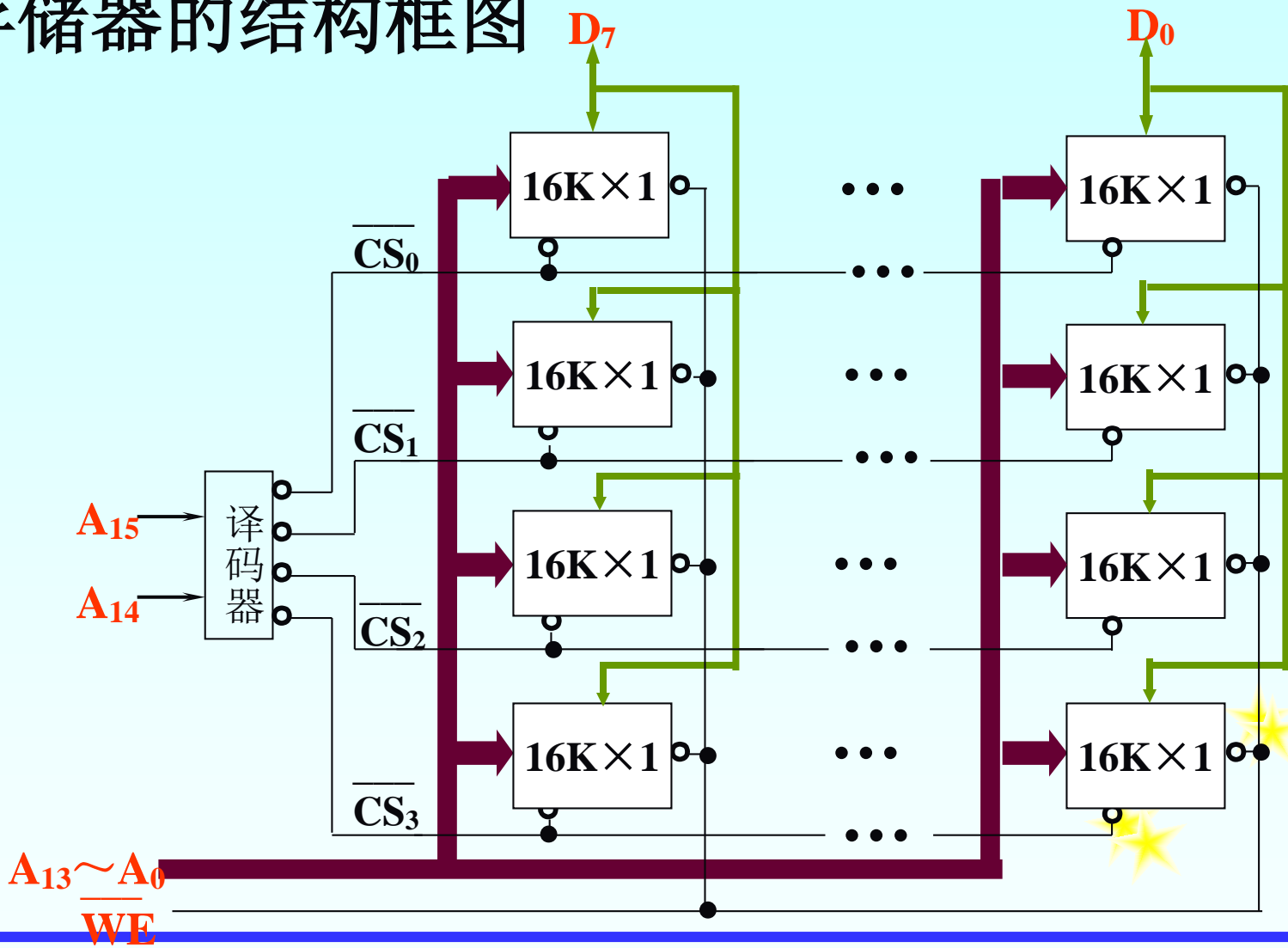
逻辑
框图

5-17 用容量为 $16\text{K} \times 1$ 的DRAM芯片构成64KB的存储器。要求：

- (1) 画出该存储器的结构框图。
- (2) 设存储器的读、写周期均为 $0.5\mu\text{s}$ ，CPU在 $1\mu\text{s}$ 内至少要访存一次。试问采用哪种刷新方式比较合理？相邻两行之间的刷新闻隔是多少？对全部存储单元刷新一遍所需的实际刷新时间是多少？



存储器的结构框图



(2)比较三种不同的刷新方式，可以发现不可采用集中刷新方式，因为集中刷新方式有死区，不能保证CPU在 $1\mu\text{s}$ 内至少要访存一次。采用分散和异步刷新方式都可以，但异步刷新方式更合理。

16K \times 1的DRAM芯片的内部结构为128 \times 128的矩阵，所以：


$$\begin{aligned} \text{相邻两行之间的刷新闻隔} &= \frac{2\text{ms}}{128} = 15.625\mu\text{s} \\ \text{全部存储单元刷新一遍所需的实际刷新时间} \\ &= 0.5\mu\text{s} \times 128 = 64\mu\text{s} \end{aligned}$$

5-18 有一个八位机，采用单总线结构，地址总线16位（ $A_{15} \sim A_0$ ），数据总线8位（ $D_7 \sim D_0$ ），控制总线中与主存有关的信号有 \overline{MREQ} （低电平有效允许访存）， R/\overline{W} （高电平为读命令，低电平为写命令）。

主存地址分配如下：从0~8191为系统程序区，由ROM芯片组成；从8192~32767为用户程序区；最后（最大地址）2K地址空间为系统程序工作区。（上述地址均用十进制表示，按字节编址）

现有如下存储芯片： $8K \times 8$ 的ROM； $16K \times 1$ 、 $2K \times 8$ 、 $4K \times 8$ 、 $8K \times 8$ 的SRAM。请从上述规格中选用芯片设计该机主存储器，画出主存的连接框图，并注意画出片选逻辑及与CPU的连接。

根据CPU的地址线、数据线，可确定整个主存空间为 $64K \times 8$ 。系统程序区由ROM芯片组成；用户程序区和系统程序工作区均由RAM芯片组成。



主存地址分配如下



共需:

8K×8的ROM一片

8K×8的SRAM三片

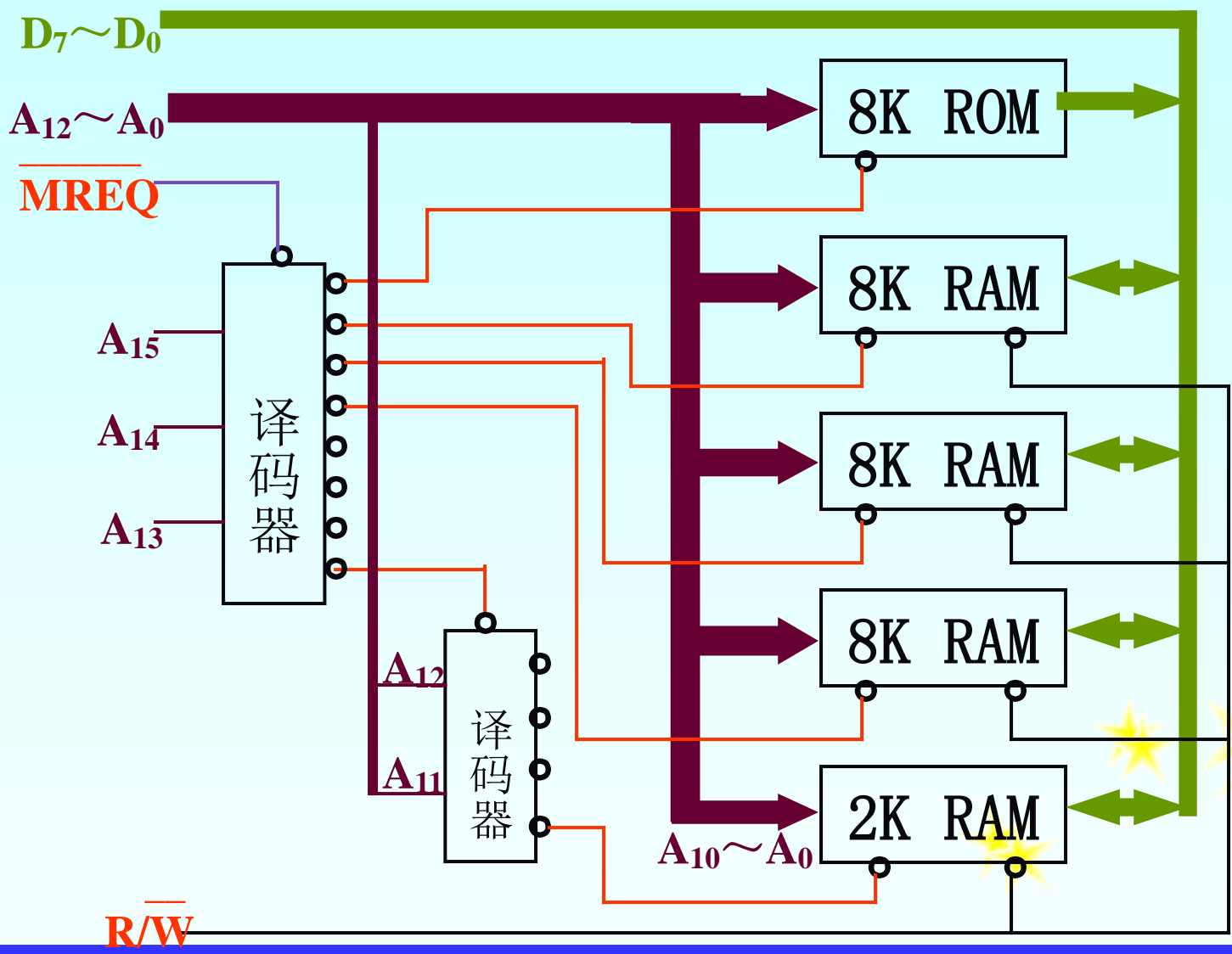
2K×8的SRAM一片



A15	A14	A13	A12	A11	A10~A0	
0	0	0			_____	8KBROM
0	0	1			_____	8KB RAM
0	1	0			_____	8KB RAM
0	1	1			_____	8KB RAM
1	1	1	1	1	_____	2KB RAM



逻辑
框图



计算机组成原理

习题串讲



6-7 以一条典型的单地址指令为例，简要说明下列部件在计算机的取指周期和执行周期中的作用。

- (1) 程序计数器PC;
- (2) 指令寄存器IR;
- (3) 算术逻辑运算部件ALU;
- (4) 存储器数据寄存器MDR;
- (5) 存储器地址寄存器MAR。



- (1)程序计数器PC，存放指令地址；
- (2)指令寄存器IR，存放当前指令；
- (3)算术逻辑运算部件ALU，进行算逻运算；
- (4)存储器数据寄存器MDR，存放写入或读出的数据/指令；
- (5)存储器地址寄存器MAR，存放写入或读出的数据/指令的地址。

以单地址指令加1为例（INC A），该指令分为三个周期：取指周期、分析周期、执行周期。

	取指周期	分析周期	执行周期
PC	$(PC) \rightarrow MAR$	--	--
IR	指令 \rightarrow MDR \rightarrow IR	--	--
ALU	$(PC) + 1$	--	$(A) + 1$
MAR	指令地址 \rightarrow MAR	$A \rightarrow MAR$	--
MDR	指令 \rightarrow MDR	$(A) \rightarrow MDR$	$(A) + 1 \rightarrow MDR$



6-9 指令和数据都存放在主存，如何识别从主存储器中取出的是指令还是数据？

指令和数据都存放在主存，它们都以二进制代码形式出现，区分的方式：

(1)从主存中取出的机器周期不同

取指周期取出的是指令，
分析或执行周期取出的是数据。

(2)取指令和数据时地址的来源不同

指令地址来源于程序计数器，
数据地址来源于地址形成部件。



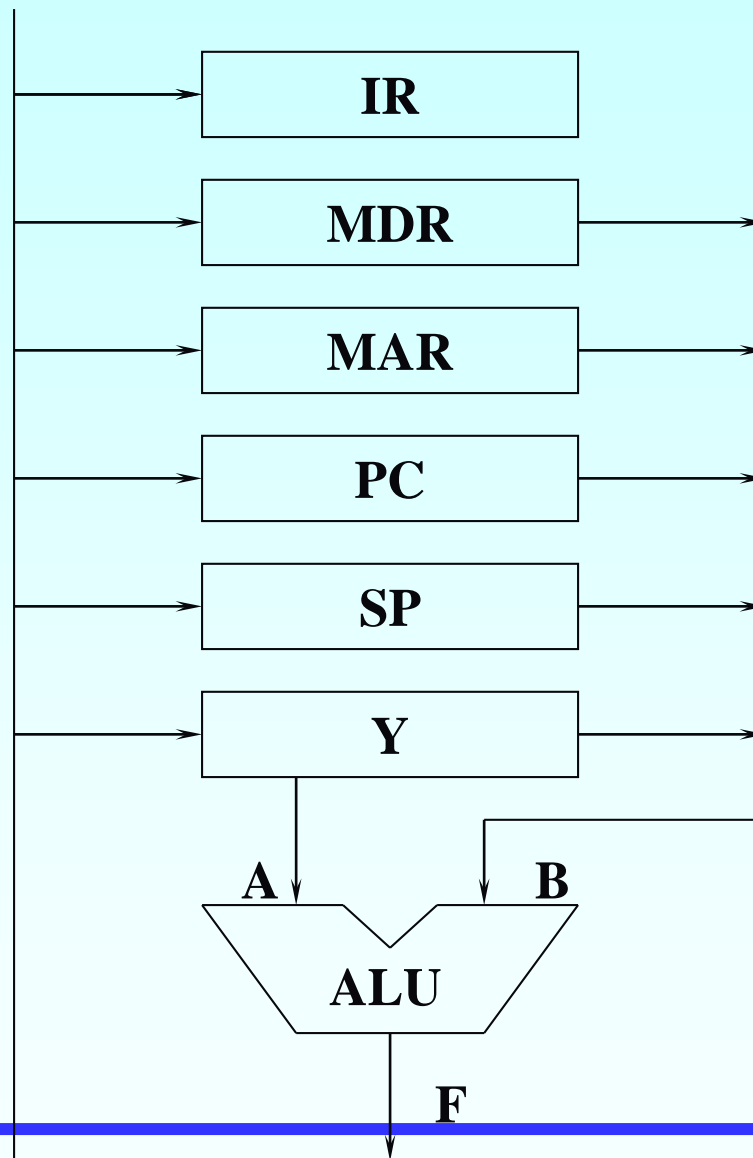
6-12 某计算机的CPU内部结构如图示。两组总线之间的所有数据传送通过ALU。ALU还具有完成以下功能的能力：

$$F=A \quad F=B$$

$$F=A+1 \quad F=B+1$$

$$F=A-1 \quad F=B-1$$

写出转子指令（JSR）的取指和执行周期的微操作序列。JSR指令占两个字，第一个字是操作码；第二个字是子程序的入口地址。返回地址保存在存储器堆栈中，堆栈指示器始终指向栈顶。



JSR指令占两个字。第一个字是操作码；第二个字是子程序的入口地址。



指令需分两次取出。




- | | | |
|------------------------------|---|--------------|
| ① PCout, F=B, MARin, Read; | } | 取指令的
第一个字 |
| ② PCout, F=B+1, PCin, Wait; | | |
| ③ MDRout, F=B, IRin; | | |
| ④ PCout, F=B, MARin, Read; | } | 取指令的
第二个字 |
| ⑤ PCout, F=B+1, PCin, Wait; | | |
| ⑥ MDRout, F=B, Yin; | | |
| ⑦ SPout, F=B-1, SPin, MARin; | } | 返回地址
压入堆栈 |
| ⑧ PCout, F=B, MDRin, Write; | | |
| ⑨ Yout, F=A, PCin | } | 子程序的首地址→PC |
| ⑩ End | | |

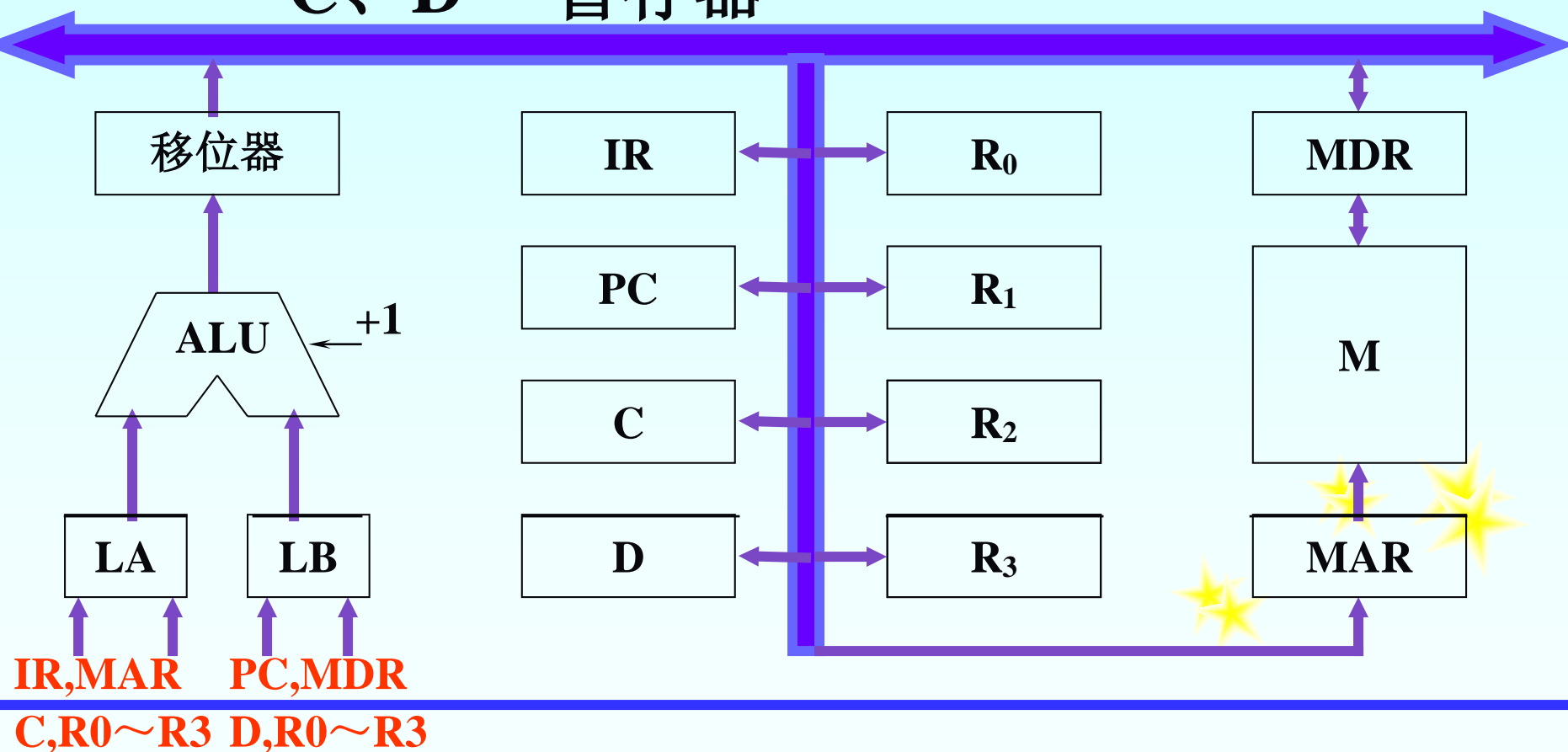
6-13 某机主要部件如图所示。

(1) 请补充各部件间的主要连接线，并注明数据流动方向。

(2) 拟出指令 **ADD (R1),(R2)+** 的执行流程（含取指过程与确定后继指令地址）。该指令的含义是进行加法操作，源操作数地址和目的操作数地址分别在寄存器 **R1** 和 **R2** 中，目的操作数寻址方式为自增型寄存器间址。



其中：LA — A输入选择器
LB — B输入选择器
C、D — 暂存器



指令 **ADD (R1),(R2)+**
((R1))+((R2)) → (R2)
(R2) + 1 →

源操作数地址

目的操作数地址

① $(PC) \rightarrow MAR$

② Read

③ $((MAR)) \rightarrow MDR \rightarrow IR$

④ $(PC) + 1 \rightarrow PC$

⑤ $(R1) \rightarrow MAR$

⑥ Read

⑦ $((MAR)) \rightarrow MDR \rightarrow C$

取指令

取被加数

⑧(R2)→MAR

⑨Read

⑩((MAR))→MDR→D

取加数

(11)(R2)+1→R2

修改目的地址

(12)C+D → MDR

(13)Write

求和并保存结果

(14)MDR→(MAR)



6-20 某机有8条微指令 $I_1 \sim I_8$ ，每条微指令所含的微命令控制信号如下表所示：

微指令	微命令信号									
	a	b	c	d	e	f	g	h	i	j
I_1	√	√	√	√	√					
I_2	√			√		√	√			
I_3		√						√		
I_4			√							
I_5			√		√		√		√	
I_6	√							√		√
I_7			√	√				√		
I_8	√	√						√		

a~j分别代表10种不同性质的微命令信号，假设一条微指令的操作控制字段为8位，请安排微指令的操作控制字段格式，并将全部微指令代码化。



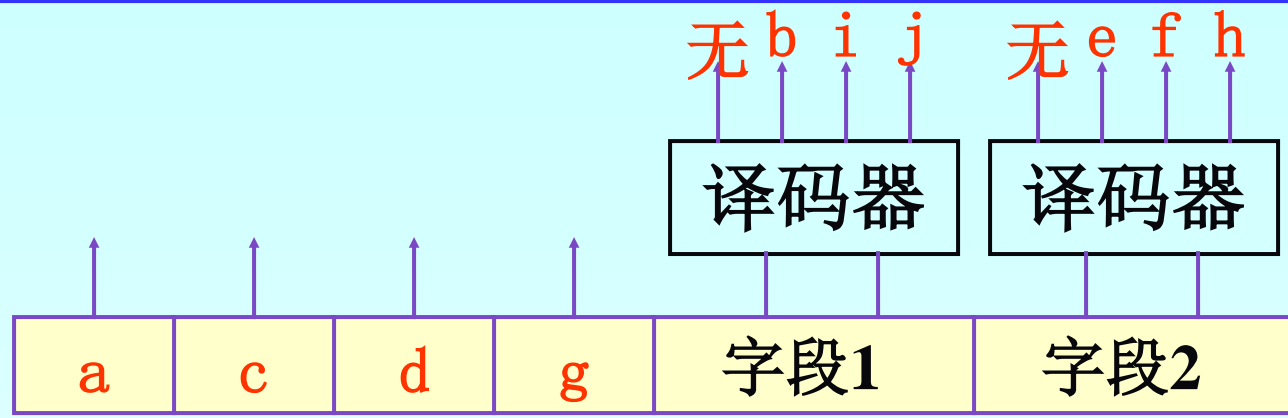
本系统中有10种不同性质的微命令信号，但一条微指令的操作控制字段只有8位，所以不能采用直接控制法。

微指令中有多个微命令是兼容的微命令，必须同时出现，如微指令I1中的a~e，故也不可以采用最短编码法。

将互斥的微命令安排在同一组内，兼容的微命令安排在不同的组内。

最终选用字段编码法和直接控制法相结合的方法。





- 字段1 {
00 无
01 b
10 i
11 j
- 字段2 {
00 无
01 e
10 f
11 h

- I1: 11100101
I2: 10110010
I3: 00000111
I4: 01000000
I5: 01011001
I6: 10001111
I7: 01100011
I8: 10000111



6-21 在微程序控制器中，微程序计数器 μPC 可以用具有加1功能的微地址寄存器 μMAR 来代替，试问程序计数器PC是否可以用具有加1功能的存储器地址寄存器MAR代替？

不可以。

主存中既有指令又有数据，它们都以二进制代码形式出现，但取指令和数据时地址的来源不同。

取指令：(PC)→MAR

取数据：地址形成部件→MAR

所以不能用MAR代替PC。



计算机组成原理

习题串讲



7-2 假设总线的工作频率为22MHz，总线宽度为16位，问总线带宽是多少？

解：时钟频率为 f ，数据位为 n ，总线带宽用 D_r 表示，则 $D_r = n \times f \div 8$ 。

$$D_r = n \times f \div 8 = 16 \times 22 \div 8 = 44 \text{MB/s}$$



7-3 PCI总线的时钟频率为33MHz/66MHz，
当该总线进行32/64位数据传送时，总线带宽
各是多少？



7-3 假设一个总线时钟周期 T 完成一个数据的传送，时钟频率为 f ，数据位为 n ，总线带宽用 D_r 表示，则 $D_r = n \times f \div 8$ 。

时钟频率为33MHz，数据32位时，
 $f = 33\text{MHz} = 33 \times 10^6/\text{s}$ ， $n = 32$ 位，根据定义可得
 $D_r = 4 \times 33 \times 10^6/\text{s} = 132\text{MB/s}$ ；依次类推，数据64位时，总线带宽为264MB/s。时钟频率为66MHz，数据32位时，总线带宽为264MB/s；数据64位时，总线带宽为528MB/s。

7-4 假定某同步总线在一个时钟周期内传送一个四字节的数据，总线时钟频率为33MHz，求总线带宽是多少？如果数据总线宽度改为64位，一个时钟周期能传送2次数据，总线时钟频率为66MHz，则总线带宽为多少？提高了多少倍？



7-4 假定某同步总线在一个时钟周期内传送一个四字节的数据，总线时钟频率为33MHz，求总线带宽是多少？如果数据总线宽度改为64位，一个时钟周期能传送2次数据，总线时钟频率为66MHz，则总线带宽为多少？提高了多少倍？



7-4 假定某同步总线在一个时钟周期内传送一个四字节的数据，总线时钟频率为33MHz，求总线带宽是多少？如果数据总线宽度改为64位，一个时钟周期能传送2次数据，总线时钟频率为66MHz，则总线带宽为多少？提高了多少倍？



解：总线带宽为： $4\text{B} \times 33\text{MHz} \div 1 = 132\text{MB/s}$

。

总线性能改进后的带宽为：

$8\text{B} \times 66\text{MHz} \div 0.5 = 1056\text{MB/s}$ ，提高了8倍。



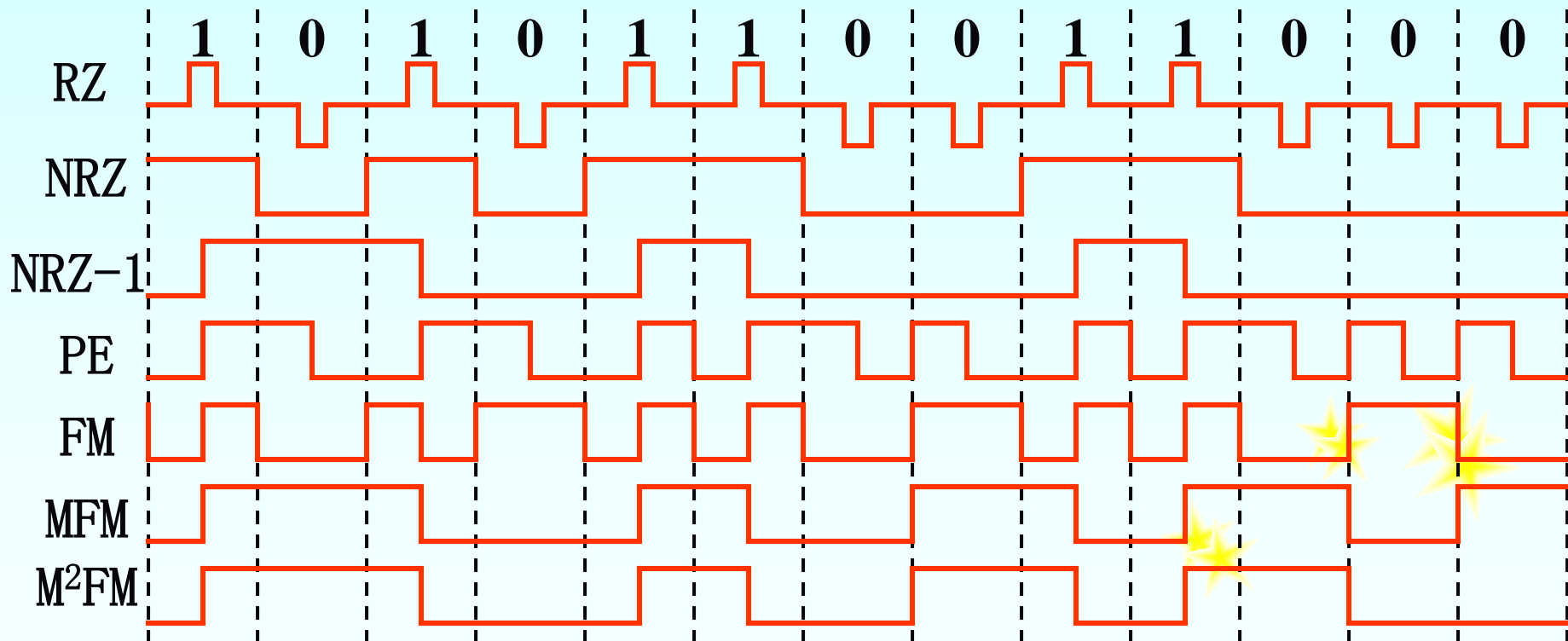
计算机组成原理

习题串讲

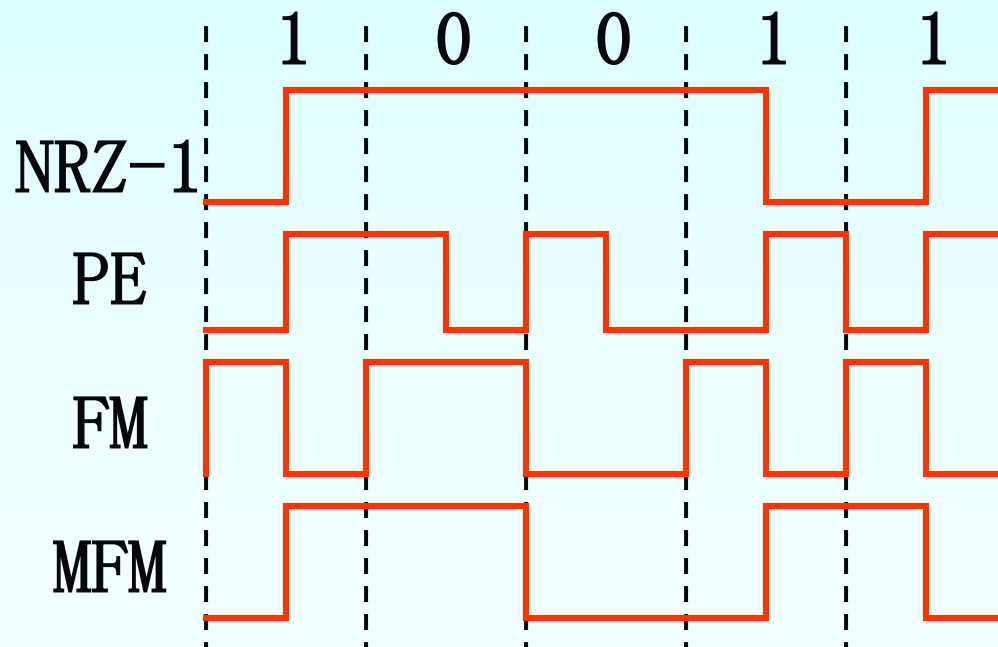


8-2 分别用RZ、NRZ、NRZ-1、PE、FM、MFM和M²FM制记录方式记录下述数据代码，画出写电流波形。

(2)1010110011000



8-3 若对磁表面存储器写入数据代码10011，请画出不归零-1制、调相制、调频制、改进的调频制等记录方式的写电流波形。



8-5 某磁盘组有六片磁盘，每片可有二个记录面，存储区域内径为22cm，外径为33cm，道密度40道/cm，位密度400位/cm，转速2400转/分。试问：

- (1) 共有多少存储面可用？
- (2) 共有多少个圆柱面？
- (3) 整个磁盘组的总存储容量有多少？
- (4) 数据传送率是多少？
- (5) 如果某文件长度超过一个磁道的容量，应将它记录在同一存储面上还是记录在同一圆柱面上？为什么？

(6) 如果采用定长信息块记录格式，直接寻址的最小单位是什么？寻址命令中如何表示磁盘地址？

解：(1) $6 \times 2 = 12$ （面），共有12个存储面可用。

(2) $40 \times \frac{33-22}{2} = 220$ （道），共有220个圆柱面。

(3) $12 \times 22\pi \times 400 \times 220 = 73 \times 10^6$ （位）
整个磁盘组的总存储容量 73×10^6 位。



$$(4) \frac{\frac{22\pi \times 400}{60}}{2400} = 1.1 \times 10^6 \text{ (位/秒)}$$
$$= 0.138 \times 10^6 \text{ (字节/秒)}$$

数据传送率 0.138×10^6 字节/秒

(5) 记录在同一圆柱面上。因为这样安排存取速度快。

(6) 如果采用定长信息块记录格式，直接寻址的最小单位是扇区。磁盘地址为：

驱动器号、圆柱面号、盘面号、扇区号

8-14 某字符显示器，采用 7×9 点阵方式，每行可显示60个字符，缓存容量至少为 1260×8 位，并采用7位标准编码，试问：

- (1) 如改用 5×7 字符点阵，其缓存容量为多少？
(设行距、字距不变——行距为5，字距为1)
- (2) 如果最多可显示128种字符，上述两种显示方式各需多大容量的字符发生器ROM？



(1)因为显示器原来的缓存为1260B，每行可显示60个字符，据此可计算出显示器的字符行数：

$$1260/60=21 \text{（行）}$$

$$\because \text{原字符窗口}=8 \times 14=(7+1) \times (9+5)$$

$$\text{现字符窗口}=6 \times 12=(5+1) \times (7+5)$$

\therefore 现显示器每行可显示80个字符，显示器可显示的字符行数为24行。

$$\text{故缓存的容量为 } 80 \times 24 = \mathbf{1920B}$$






(2) ROM中为行点阵码

7×9点阵方式: $128 \times 9 \times 7 = 1152 \times 7$ (位)
=1152 (字节)

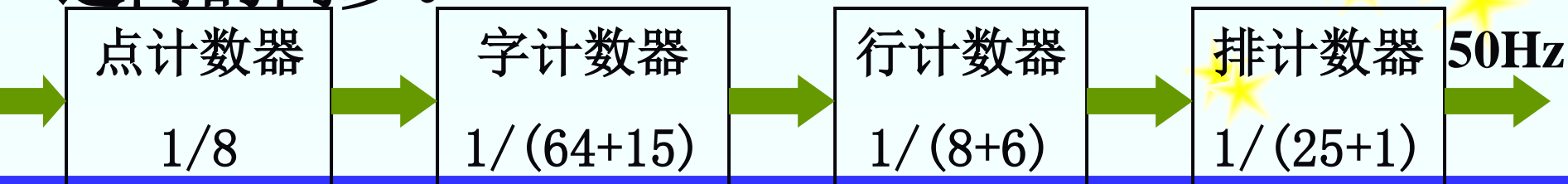
5×7点阵方式: $128 \times 7 \times 5 = 896 \times 5$ (位)
=896 (字节)



8-15 某CRT显示器可显示64种ASCII字符，每帧可显示64列 \times 25行，每个字符点阵为7 \times 8，即横向7点，字间间隔1点，纵向8点，排间间隔6点，帧频50Hz，采用逐行扫描方式。问：

- (1) 缓存容量有多大？
- (2) 字符发生器（ROM）容量有多大？
- (3) 缓存中存放的是字符的ASCII码还是字符的点阵信息？
- (4) 缓存地址与屏幕显示位置如何对应？ 
- (5) 设置哪些计数器以控制缓存访问与屏幕扫描之间的同步？它们的分频关系如何？

- (1) 缓存容量 $64 \times 25 = 1.6\text{KB}$ （不考虑显示属性），
 $64 \times 25 \times 2 = 3.2\text{KB}$ （考虑显示属性）
- (2) 字符发生器（ROM）容量 $64 \times 8 = 512\text{B}$
- (3) 缓存中存放的是字符的ASCII码。
- (4) 屏幕显示位置的排号和列号对于缓存地址，
据此找到对应的单元。
- (5) 设置四个计数器以控制缓存访问与屏幕扫描
之间的同步。



计算机组成原理

习题串讲



9-4 程序查询方式、程序中断方式、DMA方式各自适用的范围是什么？下面这些结论正确吗？为什么？

- (1) 程序中断方式能提高CPU利用率，所以在设置了中断方式后就没有再应用程序查询方式的必要了。
- (2) DMA方式能处理高速外部设备与主存间的数据传送，高速工作性能往往能覆盖低速工作要求，所以DMA方式可以完全取代程序中断方式。

程序查询方式、程序中断方式、DMA方式各自适用的范围。（略）

(1) 不正确。程序查询方式接口简单，可用于外设与主机速度相差不大，且外设数量很少的情况。

(2) 不正确。DMA方式用于高速外部设备与主存间的数据传送，但DMA结束时仍需程序中断方式做后处理。



9-15 假定某机的中断处理方式是：将断点存入00000Q单元，并从77777Q单元取出指令（即中断服务程序的第一条指令）执行。试排出完此功能的中断周期微操作序列，并判断出中断服务程序的第一条指令是何指令(假定主存容量为 2^{15} 个单元)?



中断周期微操作序列:

00000Q → MAR

(PC) → MDR

WRITE

0 → EINT

77777Q → PC

中断服务程序的第一条指令必须是一条无条件转移指令，否则PC+1将会变为00000Q，断点被当成指令。

9-19 设某机有5级中断： L_0 、 L_1 、 L_2 、 L_3 、 L_4 ，其中断响应优先次序为： L_0 最高， L_1 次之…… L_4 最低。现在要求将中断处理次序改为 $L_1 \rightarrow L_3 \rightarrow L_0 \rightarrow L_4 \rightarrow L_2$ ，试问：

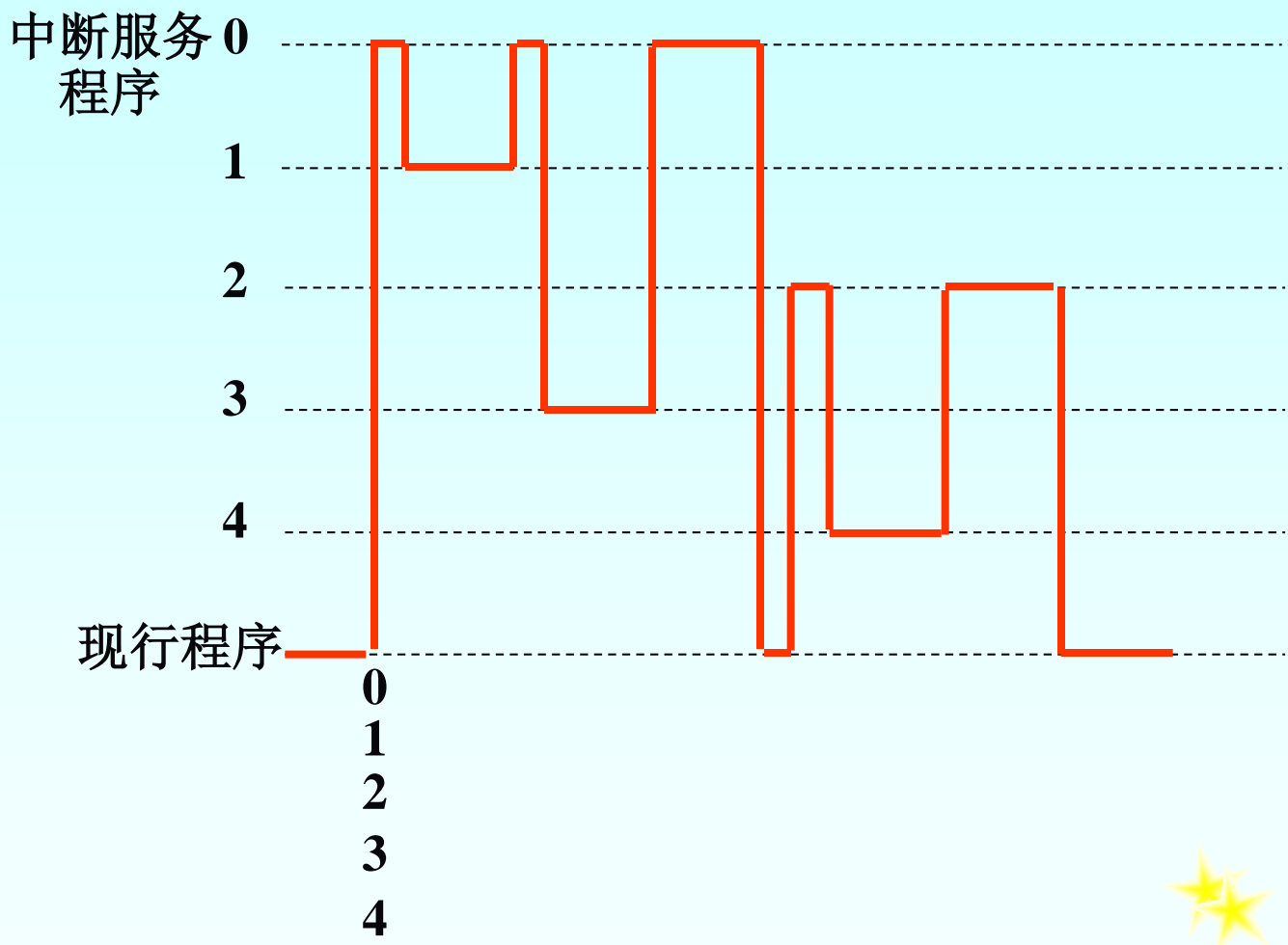
(1) 各级中断服务程序中的各中断屏蔽码应如何设置（设每级对应一位，当该位为“0”，表示中断允许；当该位为“1”，表示中断屏蔽）？

(2) 若这5级同时都发出中断请求，试画出进入各级中断处理过程示意图。



程序 级别	屏蔽码				
	0级	1级	2级	3级	4级
第0级	1	0	1	0	1
第1级	1	1	1	1	1
第2级	0	0	1	0	0
第3级	1	0	1	1	1
第4级	0	0	1	0	1





9-25 在主存接收从磁盘送来的一批信息时：

- (1) 假定主存的周期为 $1\mu\text{s}$ ，若采用程序查询方式传送，试估算在磁盘上相邻两数据字间必须具有的最短允许时间间隔是多少？
- (2) 若改为中断方式传送，这个时间又会怎样？是否还有更好的传送方式？
- (3) 在采用更好的传送方式下，假设磁盘上两数据字间的间隔为 $1\mu\text{s}$ ，主存又要被CPU占有一半周期时间，试计算这种情况下主存周期最少应是多少？

- (1) 根据程序查询方式的流程图可见，程序查询方式至少需要5条指令才能完成一个数据的传送，假定每条指令执行时间为 $1\mu\text{s}$ ，则两个数据字之间的最短时间间隔为 $5\mu\text{s}$ 。
- (2) 若改为中断方式传送，这个时间不会减少，因为程序切换时有许多辅助操作要执行。
- (3) DMA是更好的传送方式，假设磁盘上两数据字间的间隔为 $1\mu\text{s}$ ，主存又要被CPU占有一半周期时间，需要利用存储器分时法，此时主存周期应最少比原来提高一倍， $0.5\mu\text{s}$ 。