

第四部分 习题解答

第2章 数据的机器层次表示（教材P46）

2-1. 设机器数的字长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：0，0.1000，-0.1111，1101.

[解]

真 值	原 码	补 码	反 码
0	0,0000000	0,0000000	0,0000000
0.1000	0.1000000	0.1000000	0.1000000
0.1111	0.1111000	0.1111000	0.1111000
-0.1111	1.1111000	1.0001000	1.0000111

2-3. 已知下列数的原码表示，分别写出它们的补码表示： $[X1]_{\text{原}}=0.10100$ ， $[X2]_{\text{原}}=1.10111$ 。

[解] $[X1]_{\text{补}}=0.10100$ ， $[X2]_{\text{补}}=1.01001$

2-4. 已知下列数的补码表示，分别写出它们的真值： $[X1]_{\text{补}}=0.10100$ ， $[X2]_{\text{补}}=1.10111$ 。

[解] $X1=0.10100$ ， $X2=-0.01001$

2-7. 设 $[X]_{\text{补}}=1.A_1A_2A_3A_4A_5A_6$,

(1) 若要 $X > -1/2$ ， $A_1 \sim A_6$ 要满足什么条件？

(2) 若要 $-1/8 \geq X \geq -1/4$ ， $A_1 \sim A_6$ 要满足什么条件？

[解] (1) $\because [-1/2]_{\text{补}}=1.100000$

$\therefore X > -1/2$ 的条件是：1. $A_1A_2A_3A_4A_5A_6 > 1.100000$

即 $A_1(A_2+A_3+A_4+A_5+A_6)=1$

(2) $\because [-1/8]_{\text{补}}=1.111000$ $[-1/4]_{\text{补}}=1.110000$

$\therefore -1/8 \geq X \geq -1/4$ 的条件是：

1. $A_1A_2A_3A_4A_5A_6 \leq 1.111000$ 并且 1. $A_1A_2A_3A_4A_5A_6 \geq 1.110000$

即 $A_1A_2(A_3\overline{A_4}\overline{A_5}\overline{A_6} + \overline{A_3})=1$

简化为： $A_1A_2(\overline{A_4}\overline{A_5}\overline{A_6} + \overline{A_3})=1$

2-9. 某机字长16位，问在下列几种情况下所能表示数值的范围：

(1) 无符号整数

(2) 用原码表示定点小数

(3) 用补码表示定点小数

(4) 用原码表示定点整数

(5) 用补码表示定点整数

[解] (1) $0 \leq X \leq (2^{16}-1)$

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

(3) $-1 \leq X \leq (1-2^{-15})$

(4) $-(2^{15}-1) \leq X \leq (2^{15}-1)$

(5) $-2^{15} \leq X \leq (2^{15}-1)$

2-15. 某浮点数字长 16 位，其中阶码部分 6 位（含一位阶符），移码表示，以 2 为底；尾数部分 10 位（含 1 位数符，位于尾数最高位），补码表示，规格化。分别写出下列各题的二进制代码与十进制真值。

- (1) 非零最小正数
- (2) 最大正数
- (3) 绝对值最小负数
- (4) 绝对值最大负数

[解] (1) 非零最小正数: 000000, 0.100000000; $2^{-1} \times 2^{-2^5} = 2^{-33}$ 。
 (2) 最大正数: 111111, 0.111111111; $(1-2^{-9}) \times 2^{2^5-1} = (1-2^{-9}) \times 2^{31}$
 (3) 绝对值最小负数: 000000, 1.011111111; $-(2^{-1}+2^{-9}) \times 2^{-2^5}$ 。
 (4) 绝对值最大负数: 111111, 1.000000000; $-1 \times 2^{2^5-1} = -2^{31}$ 。

2-18. 某浮点数字长 32 位，格式如下。其中阶码部分 8 位，以 2 为底，移码表示；尾数部分一共 24 位（含 1 位数符），补码表示。现有一浮点代码为 $(8C5A3E00)_{16}$ ，试写出它所表示的十进制真值。

0	7	8	9	31
阶码	数符	尾数		

[解] $(8C5A3E00)_{16} = 1000\ 1100\ \underline{0101\ 1010\ 0011\ 1110\ 0000\ 0000}\ B$
 真值: $\underline{+0.10110100011111} \times 2^{12} = (101101000111.11)_2 = (2887.75)_{10}$

2-20. 将下列十进制数转换为 IEEE 短浮点数:

- (1) 28.75;
- (3) -0.625;
- (5) -1000.5。

[解] (1) $28.75 = 11100.11 = 1.110011 \times 2^4$
 符号位=0
 阶码= $127+4=131$
 IEEE 短浮点数格式: 0, 10000011, 110 0110 0000 0000 0000 0000
 即 41E60000H
 (3) $-0.625 = 0.101 = -1.01 \times 2^{-1}$
 符号位=1
 阶码= $127-1=126$
 IEEE 短浮点数格式: 1, 01111110, 010 0000 0000 0000 0000 0000
 即 BF200000H
 (5) $-1000.5 = 1111101000.1 = 1.1111010001 \times 2^9$
 符号位=1
 阶码= $127+9=136$
 IEEE 短浮点数格式: 1, 10001000, 111 1010 0010 0000 0000 0000
 即 C47A2000H

2-21. 将下列 IEEE 短浮点数转换为十进制数:

- (1) 11000000 11110000 00000000 00000000
- (3) 01000011 10011001 00000000 00000000

[解] (1) 1, 10000001, 1110000000000000000000

符号位=1

阶码=129-127=2

1.111* 2^2 =111.1B=7.5

∴结果=-7.5

(3) 0, 10000111, 0011001000000000000000

符号位=0

阶码=135-127=8

1.0011001* 2^8 =100110010B=306。

∴结果=306。

2-23. 以下列形式表示 $(5382)_{10}$ 。

(1) 8421 码; (2) 余 3 码; (3) 2421 码; (4) 二进制码

[解] (1) 0101 0011 1000 0010;

(2) 1000 0110 1011 0101;

(3) 1011 0011 1110 0010;

(4) 1010100000110。

2-24. 填写下列代码的奇偶校验位，现设为奇校验：

1 0 1 0 0 0 0 1

0 0 0 1 1 0 0 1

0 1 0 0 1 1 1 0

[解] 3 个代码的校验位分别是：0，0，1。

补充题 2-1. 已知 $[X]_{\text{补}}=3\text{EH}$, $[Y]_{\text{补}}=\text{DCH}$, 求： $[2X]_{\text{补}}$, $[2Y]_{\text{补}}$, $[X/2]_{\text{补}}$, $[Y/4]_{\text{补}}$,
 $[X]_{\text{原}}$, $[Y]_{\text{原}}$, $[X]_{\text{反}}$, $[Y]_{\text{反}}$, $[X]_{\text{移}}$, $[Y]_{\text{移}}$

[解] $[2X]_{\text{补}}=7\text{CH}$; $[X/2]_{\text{补}}=1\text{FH}$; $[X]_{\text{原}}=3\text{EH}$; $[X]_{\text{反}}=3\text{EH}$; $[X]_{\text{移}}=\text{BEH}$

$[2Y]_{\text{补}}=\text{B8H}$; $[Y/4]_{\text{补}}=\text{F7H}$; $[Y]_{\text{原}}=\text{A4H}$; $[Y]_{\text{反}}=\text{DBH}$; $[Y]_{\text{移}}=5\text{CH}$

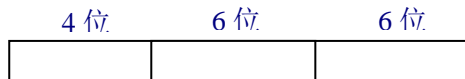
第3章 指令系统 (教材P75)

3-3. 某机为定长指令字结构, 指令长度为 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, 为单操作数最多可能有多少种? 上述三类指令各自允许的最大指令条数是多少?

[解] (1) 设单操作数指令有 X 条,

$$\text{则 } ((2^4 - K) \times 2^6 - X) \times 2^6 \geq L$$

$$\text{即, } X \leq (2^4 - K) \times 2^6 - (L / 2^6)$$



\therefore 最多还可以设计出单操作数指令 $\lfloor (2^4 - K) \times 2^6 - (L / 2^6) \rfloor$ 条。

(2) 双操作数指令的最大指令数: $2^4 - 1$;

单操作数指令的最大指令数: $15 \times 2^6 - 1$

(假定双操作数指令仅 1 条, 为无操作数指令留出 1 个扩展窗口);

无操作数指令的最大指令数: $2^{16} - 2^{12} - 2^6$

(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, 此时双操作数和单操作数指令各仅有 1 条)

3-4. 设某机为定长指令字结构, 指令长度为 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。

[解] 4 条三地址指令: 000 XXX YYY ZZZ

.....

011 XXX YYY ZZZ

8 条二地址指令: 100 000 XXX YYY

.....

100 111 XXX YYY

180 条单地址指令: 101 000 000 XXX

.....

111 110 011 XXX

3-5. 指令格式同题 4, 能否构成: 三地址指令 4 条, 单地址指令 255 条, 零地址指令 64 条? 为什么?

[解] 不能构成这样的指令系统。

这是因为, 如果三地址指令有 4 条, 单地址指令有 255 条, 则零地址指令的条数最多只能有: $[(2^3 - 4) \times 2^6 - 255] \times 2^3 = 8$

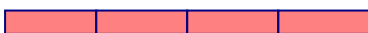
3 位 3 位 3 位 3 位



三地址指令: 可从 2^3 种编码中选 4 种, 剩 4 个编码



单地址指令: 可从扩展出 4×2^6 个编码种选 255 种, 剩 1 个编码



零地址指令: 剩下 1 个编码与 3 位地址只能扩展出 1×2^3 个编码

3-7. 试比较间接寻址和寄存器间址。

[解] 间接寻址方式其有效地址在内存中, 操作数也在内存中; 寄存器寻址方式其有效地址在 CPU 内的寄存器中, 操作数在内存中。所以前者寻找操作数较慢。

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：

(1) 若只采用直接寻址方式，指令能访问多少主存单元？

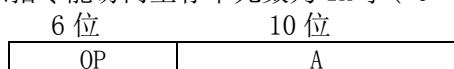
(2) 为扩充指令的寻址范围，可采用直接/间接寻址方式，若只增加一位直接/间接标志，指令可寻址范围为多少？指令直接寻址的范围为多少？

(3) 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？

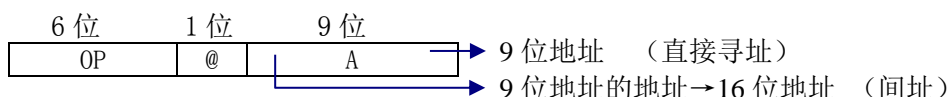
(4) 采用 (2)、(3) 两种方式结合，指令的寻址范围为多少？指令直接寻址范围为多少？

[解] 该计算机共有 64 条指令，所以指令的操作码需要占 6 位，其余 10 位地址码字段。

(1) 只采用直接寻址方式时，不需要标识寻址方式，地址码字段的 10 位全都作为直接地址，所以指令能访问主存单元数为 1K 字 ($0 \sim 2^{10}-1$ 单元)。

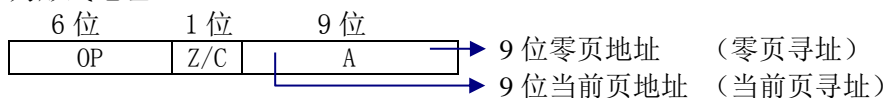


(2) 采用直接/间接寻址方式时，需从 10 位的地址字段中留出 1 位来作“直接/间接”寻址的标志，余下的 9 位为形式地址。



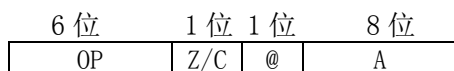
\therefore 指令直接寻址的范围为 $0 \sim 2^9-1$ 号存储单元，间接寻址的范围为 $0 \sim 2^{16}-1$ 号存储单元 (存储字长即操作数地址的位数)。

(3) 采用页面寻址方式时，也需从 10 位的地址字段中留出 1 位来作“Z/C”寻址的标志，余下的 9 位为形式地址。



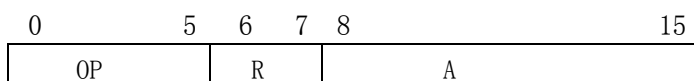
\therefore 指令的零页寻址的范围为 $0 \sim 2^9-1$ 号存储单元，当前页寻址的范围也指令所在页有关，可访问当前页内的 $0 \sim 2^9-1$ 号存储单元。由于“当前页”可以是 64K 字主存的任意一页，所以指令可能访问到的地址是 $0 \sim 2^{16}-1$ 号存储单元。

(4) 需从 10 位的地址字段中留出 2 位来标志“直接/间接”和“Z/C”，余下 8 位为形式地址。



\therefore 指令直接寻址的范围为 $0 \sim 2^8-1$ 号存储单元，间接寻址的范围为 $0 \sim 2^{16}-1$ 号存储单元 (存储字长即操作数地址的位数)。

3-12. 已知某小型机字长为 16 位，其双操作数指令的格式如下：



其中：OP 为操作码，R 为通用寄存器地址。试说明下列情况下能访问的最大主存区域有多少机器字？

- (1) A 为立即数
- (2) A 为直接主存单元地址
- (3) A 为间接地址（非多重间址）
- (4) A 为变址寻址的形式地址，假定变址寄存器为 R1（字长为 16 位）

[解] (1) 1 个机器字（即指令字）；

(2) 256 个机器字（可直接寻址的地址有 8 位）；

(3) 65536 个机器字

（指令的地址字段给出的操作数地址的地址是 8 位，从内存取出的操作数地址是 16 位，与字长相同）

(4) 65536 个机器字

（变址寄存器 16 位，与 8 位位移量相加并取 16 位模，所得到的地址还是 16 位）

补充题 3-1. 根据操作数所在的位置，指出其寻址方式的名称。

(1) 操作数在寄存器中

(2) 操作数的地址在通用寄存器中

(3) 操作数在指令中

(4) 操作数的地址在指令中

(5) 操作数地址的地址在指令中

(6) 操作数的地址为寄存器内容与位移量之和（寄存器分别为基址寄存器，变址寄存器和程序计数器）

(7) 操作数为某二进制位

(8) 操作数为栈顶元素

[解] 各小题对应寻址方式的名称是：

(1) 寄存器寻址 (2) 寄存器间址 (3) 立即寻址 (4) 直接寻址

(5) 间接寻址 (6) 分别为基址、变址和自相对寻址

(7) 位寻址 (8) 堆栈寻址

第4章 数值的机器运算 (教材P120)

4-2. 某加法器采用组内并行，组间并行的进位链，4 位一组，写出进位信号 C_6 逻辑表达式。

[解] 组间并行的进位链，产生的最低一组的进位输出为：

$$C_4 = G_1^* + P_1^* C_0 \quad (\text{其中: } G_1^* = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1; P_1^* = P_4 P_3 P_2 P_1)$$

然后在组内产生进位信号： $C_6 = G_6 + P_6 C_5 = G_6 + P_6 G_5 + P_6 P_5 C_4$

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

- (1) $X=0.11011, Y=0.11111$
- (2) $X=0.11011, Y=-0.10101$
- (3) $X=-0.10110, Y=-0.00001$
- (4) $X=-0.11011, Y=0.11110$

[解]

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

00.11011	$[X]_{\text{补}}$	
+00.11111	$[Y]_{\text{补}}$	
01.11010	$[X+Y]_{\text{补}}$	结果正溢出！

- (2) $[X]_{\text{补}}=0.11011, [Y]_{\text{补}}=1.01011$

00.11011	$[X]_{\text{补}}$	
+11.01011	$[Y]_{\text{补}}$	
00.00110	$[X+Y]_{\text{补}}$	结果不溢出！

$$X+Y=0.00110$$

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

11.01010	$[X]_{\text{补}}$	
+11.11111	$[Y]_{\text{补}}$	
11.01001	$[X+Y]_{\text{补}}$	结果不溢出！

$$X+Y=-0.10111$$

- (4) $[X]_{\text{补}}=1.00101, [Y]_{\text{补}}=0.11110$

11.00101	$[X]_{\text{补}}$	
+00.11110	$[Y]_{\text{补}}$	
00.00011	$[X+Y]_{\text{补}}$	结果不溢出！

$$X+Y=0.00011$$

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。

- (1) $X=0.11011, Y=-0.11111$
- (2) $X=0.10111, Y=0.11011$
- (3) $X=+0.11011, Y=-0.10011$
- (4) $X=-0.10110, Y=-0.00001$

[解]

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

00.11011	$[X]_{\text{补}}$	
+00.11111	$[-Y]_{\text{补}}$	
01.11010	$[X-Y]_{\text{补}}$	结果正溢出！

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

$$\begin{array}{r} 00.10111 \\ +11.00101 \\ \hline 11.11100 \end{array} \quad \begin{array}{l} [X]_{\text{补}} \\ [-Y]_{\text{补}} \\ [X-Y]_{\text{补}} \end{array} \quad \begin{array}{l} \\ \\ \text{结果不溢出!} \end{array}$$

$X-Y = -0.00100$

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

$$\begin{array}{r} 00.11011 \\ +00.10011 \\ \hline 01.01110 \end{array} \quad \begin{array}{l} [X]_{\text{补}} \\ [-Y]_{\text{补}} \\ [X-Y]_{\text{补}} \end{array} \quad \begin{array}{l} \\ \\ \text{结果正溢出!} \end{array}$$

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

$$\begin{array}{r} 11.01010 \\ +00.00001 \\ \hline 11.01011 \end{array} \quad \begin{array}{l} [X]_{\text{补}} \\ [-Y]_{\text{补}} \\ [X-Y]_{\text{补}} \end{array} \quad \begin{array}{l} \\ \\ \text{结果不溢出!} \end{array}$$

$X-Y = -0.10101$

4-6. 已知: $X=0.1011$, $Y=-0.0101$

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

[解] 由 $[X]_{\text{补}}=0.1011$, $[Y]_{\text{补}}=1.1011$ 得:

$$[X/2]_{\text{补}}=0.0101, [X/4]_{\text{补}}=0.0010, [-X]_{\text{补}}=1.0101$$

$$[Y/2]_{\text{补}}=1.1101, [Y/4]_{\text{补}}=1.1110, [-Y]_{\text{补}}=0.0101, [2Y]_{\text{补}}=1.0110$$

思考 4-1. 在全加器里进位输出表达式 $C_i = A_i B_i + (A_i \oplus B_i) C_{i-1}$ 为什么可以用下式替代:

$$C_i = A_i B_i + (A_i + B_i) C_{i-1} ?$$

答: 由真值表可得到: $(A_i + B_i) = (A_i \oplus B_i) + A_i B_i$

$$\begin{aligned} \therefore C_i &= A_i B_i + (A_i + B_i) C_{i-1} = A_i B_i + [(A_i \oplus B_i) + A_i B_i] C_{i-1} \\ &= A_i B_i + (A_i \oplus B_i) C_{i-1} + A_i B_i C_{i-1} \\ &= A_i B_i + (A_i \oplus B_i) C_{i-1} \end{aligned}$$

思考 4-2. 先行进位主要是解决什么问题, 采用什么设计思路?

答: 先行进位解决的问题是加法进位的传递速度问题。基本设计原理是: 让各位的进位与低位的进位无关, 仅与两个参加操作的数有关。由于每位的操作数都是同时给出的, 各进位信号几乎同时产生, 和数也随之产生, 所以先行进位可以提高进位的传递速度, 从而提高加法器的运算速度。

4-8. 分别用原码乘法和补码乘法计算 $X \times Y$.

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

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

[解]

原码乘法:

(1) $|X|=0.11011 \rightarrow B$, $|Y|=0.11111 \rightarrow C$, $0 \rightarrow A$

$$\begin{array}{r} \boxed{A} \\ 00.00000 \\ + |X| \quad 00.11011 \\ \hline \end{array} \quad \begin{array}{r} \boxed{C} \\ 0.11111 \\ \hline \end{array} \quad \begin{array}{l} \text{说明} \\ C_5=1, + |X| \end{array}$$

	00.11011		
	00.01101	1 0 1 1 1 <u>1</u>	部分积右移 1 位
+ X	00.11011		$C_5=1, + X $
	01.01000		
	00.10100	0 1 0 1 1 <u>1</u>	部分积右移 1 位
+ X	00.11011		$C_5=1, + X $
	01.01111		
	00.10111	1 0 1 0 1 <u>1</u>	部分积右移 1 位
+ X	00.11011		$C_5=1, + X $
	01.10010		
	00.11001	0 1 0 1 0 <u>1</u>	部分积右移 1 位
+ X	00.11011		$C_5=1, + X $
	01.10100		
	00.11010	0 0 1 0 1 0	部分积右移 1 位

所以, $|X \times Y| = 0.1101000101$
 $X \times Y = -0.1101000101$

(2) $|X| = 0.11010 \rightarrow B, |Y| = 0.01110 \rightarrow C, 0 \rightarrow A$

A 寄存器		C 寄存器	说明
00.00000		0.0 1 1 1 0	
+ 0	00.00000		$C_5=0, + 0$
→	00.00000		
	00.00000	0 0 0 1 1 <u>1</u>	部分积右移 1 位
+ X	00.11010		$C_5=1, + X $
→	00.11010		
	00.01101	0 0 0 0 1 <u>1</u>	部分积右移 1 位
+ X	00.11010		$C_5=1, + X $
→	01.00111		
	00.10011	1 0 0 0 0 <u>1</u>	部分积右移 1 位
+ X	00.11010		$C_5=1, + X $
→	01.01101		
	00.10110	1 1 0 0 0 <u>0</u>	部分积右移 1 位
+ 0	00.00000		$C_5=0, + 0$
→	00.10110		
	00.01011	0 1 1 0 0 <u>0</u>	部分积右移 1 位

所以, $|X \times Y| = 0.0101101100$
 $X \times Y = +0.0101101100$

补码乘法:

(1) $[X]_{\text{补}} = 0.11011 \rightarrow B, [Y]_{\text{补}} = 1.00001 \rightarrow C, 0 \rightarrow A$

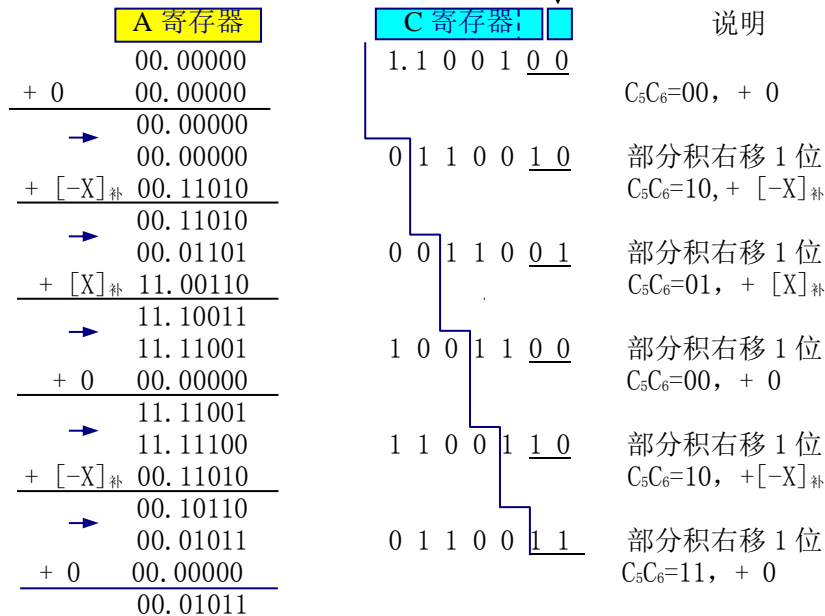
$[-X]_{\text{补}} = 1.00101$

A 寄存器		C 寄存器	说明
00.00000		1.0 0 0 0 1 0	
+ $[-X]_{\text{补}}$	11.00101		$C_5C_6=10, + [-X]_{\text{补}}$
→	11.00101		
	11.10010	1 1 0 0 0 0 <u>1</u>	部分积右移 1 位
+ $[X]_{\text{补}}$	00.11011		$C_5C_6=01, + [X]_{\text{补}}$
→	00.01101		
	00.00110	1 1 1 0 0 0 <u>0</u>	部分积右移 1 位
+ 0	00.00000		$C_5C_6=00, + 0$
→	00.00110		
	00.00011	0 1 1 1 0 0 <u>0</u>	部分积右移 1 位
+ 0	00.00000		$C_5C_6=00, + 0$
→	00.00011		

	00.00001	1 0 1 1 1 <u>0 0</u>	部分积右移 1 位
+ 0	00.00000		$C_5C_6=00, + 0$
	00.00001		
	00.00000	1 1 0 1 1 <u>1 0</u>	部分积右移 1 位
+ $[-X]_{补}$	11.00101		$C_5C_6=10, + [-X]_{补}$
	11.00101		

所以, $[X \times Y]_{补} = 1.0010111011$
 $X \times Y = -0.1101000101$

(2) $[X]_{补} = 1.00110 \rightarrow B, [Y]_{补} = 1.10010 \rightarrow C, 0 \rightarrow A$
 $[-X]_{补} = 0.11010$



所以, $[X \times Y]_{补} = 0.0101101100$
 $X \times Y = 0.0101101100$

4-10. 分别用原码和补码加减交替法计算 X/Y。

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

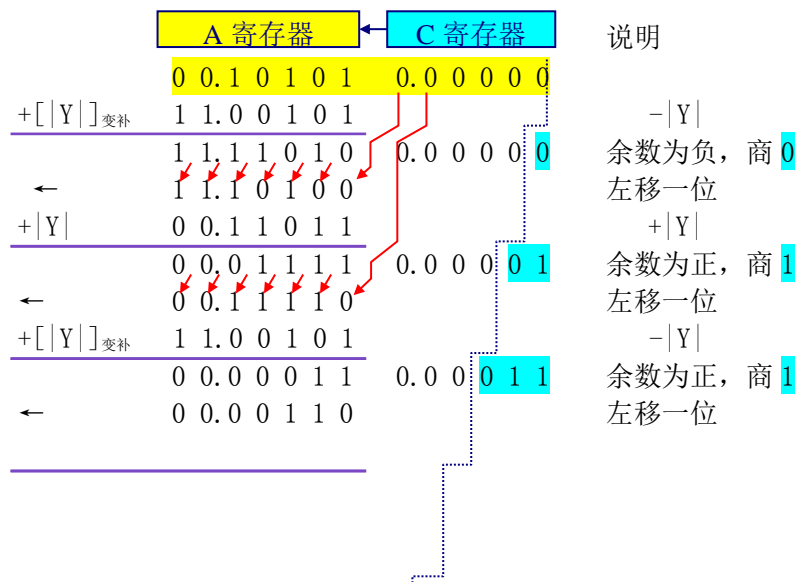
(4) $X = -0.10110, Y = -0.11011$

[解]

原码除法:

(2) $|X| = 00.10101 \rightarrow A, |Y| = 00.11011 \rightarrow B, 0 \rightarrow C$

$[|Y|]_{变补} = 11.00101$



$+[Y]_{\text{变补}}$	1 1.0 0 1 0 1		$- Y $
	1 1.0 1 0 1 1	0.0 0 1 1 0	余数为负, 商 0
←	1 0.1 0 1 1 0		左移一位
$+[Y]$	0 0.1 1 0 1 1		$+[Y]$
	1 1.1 0 0 0 1	0.0 1 1 0 0	余数为负, 商 0
←	1 1.0 0 0 1 0		左移一位
$+[Y]$	0 0.1 1 0 1 1		$+[Y]$
	1 1.1 1 1 0 1	0.1 1 0 0 0	余数为负, 商 0
$+[Y]$	0 0.1 1 0 1 1		$+[Y]$ 恢复余数
	0 0.1 1 0 0 0		

商符: $Q_s = X_s \oplus Y_s = 1 \oplus 0 = 1$

∴ 商 = -0.11000

$$X/Y = -[0.11000 + (0.11000 \cdot 2^{-5})/0.11011]$$

(中括号内余数项的分子和分母都取正)

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

$|X| = 00.10110 \rightarrow A$, $|Y| = 00.11011 \rightarrow B$, $0 \rightarrow C$

$[|Y|]_{\text{变补}} = 11.00101$

A 寄存器		C 寄存器		说明	
	0 0.1 0 1 1 0		0.0 0 0 0 0		
$+[Y]_{\text{变补}}$	1 1.0 0 1 0 1			$- Y $	
	1 1.1 1 0 1 1		0.0 0 0 0 0	余数为负, 商 0	
←	1 1.1 0 1 1 0			左移一位	
$+[Y]$	0 0.1 1 0 1 1			$+[Y]$	
	0 0.1 0 0 0 1		0.0 0 0 0 1	余数为正, 商 1	
←	0 1.0 0 0 1 0			左移一位	
$+[Y]_{\text{变补}}$	1 1.0 0 1 0 1			$- Y $	
	0 0.0 0 1 1 1		0.0 0 0 1 1	余数为正, 商 1	
←	0 0.0 1 1 1 0			左移一位	
$+[Y]_{\text{变补}}$	1 1.0 0 1 0 1			$- Y $	
	1 1.1 0 0 1 1		0.0 0 1 1 0	余数为负, 商 0	
←	1 1.0 0 1 1 0			左移一位	
$+[Y]$	0 0.1 1 0 1 1			$+[Y]$	
	0 0.0 0 0 0 1		0 0 1 1 0 1	余数为正, 商 1	
←	0 0.0 0 0 1 0			左移一位	
$+[Y]_{\text{变补}}$	1 1.0 0 1 0 1			$- Y $	
	1 1.0 0 1 1 1		0.1 1 0 1 0	余数为负, 商 0	
$+[Y]$	0 0.1 1 0 1 1			$+[Y]$ 恢复余数	

0 0.0 0 0 1 0

商符: $Q_s = X_s \oplus Y_s = 1 \oplus 1 = 0$

\therefore 商 = 0.11010

$X/Y = +[0.11010 + (0.00010 \times 2^{-5})/0.11011]$

(转下页)

思考 4-3. 计算机硬件的微操作有些什么特点?

答: 计算机硬件的微操作至少有这样一些特点: ①是寄存器到寄存器的操作 (各种记忆元件可抽象为寄存器); ②是硬件应该一步完成的操作 (该操作通常不能拆成几步来操作); ③需要有开通相应数据通路的电平型控制信号和记忆元件接收数据的脉冲信号相配合来完成操作; ④进行该操作时其数据流动是畅通的, 并且没有路径冲突。

<第 10 题续>

补码除法:

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

$[X]_{\text{补}} = 11.01011 \rightarrow A$, $[Y]_{\text{补}} = 00.11011 \rightarrow B$, $0 \rightarrow C$; $[-Y]_{\text{补}} = 11.00101$

	A	C	说明
	1 1 0 1 0 1 1	0.0 0 0 0 0 0	
$+ [Y]_{\text{补}}$	0 0 1 1 0 1 1		$[X]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 异号, $+ [Y]_{\text{补}}$
	0 0 0 0 1 1 0	0.0 0 0 0 0 <u>1</u>	$[r_i]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 同号, 商 1
\leftarrow	0 0 0 1 1 0 0		左移一位
$+ [-Y]_{\text{补}}$	1 1 0 0 1 0 1		$+ [-Y]_{\text{补}}$
	1 1 1 0 0 0 1	0.0 0 0 1 <u>0</u>	$[r_i]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 异号, 商 0
\leftarrow	1 1 0 0 0 1 0		左移一位
$+ [Y]_{\text{补}}$	0 0 1 1 0 1 1		$[X]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 异号, $+ [Y]_{\text{补}}$
	1 1 1 1 1 0 1	0.0 0 1 <u>0</u> <u>0</u>	$[r_i]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 异号, 商 0
\leftarrow	1 1 1 1 0 1 0		左移一位
$+ [Y]_{\text{补}}$	0 0 1 1 0 1 1		$[X]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 异号, $+ [Y]_{\text{补}}$
	0 0 1 0 1 0 1	0.0 1 <u>0</u> <u>0</u> <u>1</u>	$[r_i]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 同号, 商 1
\leftarrow	0 1 0 1 0 1 0		左移一位
$+ [-Y]_{\text{补}}$	1 1 0 0 1 0 1		$+ [-Y]_{\text{补}}$
	0 0 0 1 1 1 1	0 <u>1</u> <u>0</u> <u>0</u> <u>1</u> <u>1</u>	$[r_i]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 同号, 商 1
\leftarrow	0 0 1 1 1 1 0		左移一位
$+ [-Y]_{\text{补}}$	1 1 0 0 1 0 1		$+ [-Y]_{\text{补}}$
	0 0 0 0 0 1 1	<u>1</u> . <u>0</u> <u>0</u> <u>1</u> <u>1</u> <u>1</u>	末位恒置 1

$\therefore [商]_{\text{补}} = 1.00111$

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

$[X]_{\text{补}} = 11.01010 \rightarrow A$, $[Y]_{\text{补}} = 11.00101 \rightarrow B$, $0 \rightarrow C$; $[-Y]_{\text{补}} = 00.11011$

	A	C	说明
	1 1 0 1 0 1 0	0.0 0 0 0 0 0	
$+ [-Y]_{\text{补}}$	0 0 1 1 0 1 1		$[X]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 同号, $+ [-Y]_{\text{补}}$
	0 0 0 0 1 0 1	0.0 0 0 0 0 <u>0</u>	$[r_i]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 异号, 商 0
\leftarrow	0 0 0 1 0 1 0		左移一位

$+ [Y]_{\text{补}}$	1 1.0 0 1 0 1		$+ [Y]_{\text{补}}$	
	1 1.0 1 1 1 1	0.0 0 0 0 <u>1</u>	$[r_i]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 同号, 商 1	
\leftarrow	1 0.1 1 1 1 0		左移一位	
$+ [-Y]_{\text{补}}$	0 0.1 1 0 1 1		$[X]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 同号, $+ [-Y]_{\text{补}}$	
	1 1.1 1 0 0 1	0.0 0 0 1 <u>1</u>	$[r_i]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 同号, 商 1	
\leftarrow	1 1.1 0 0 1 0		左移一位	
$+ [-Y]_{\text{补}}$	0 0.1 1 0 1 1		$[X]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 同号, $+ [-Y]_{\text{补}}$	
	0 0.0 1 1 0 1	0.0 0 1 1 <u>0</u>	$[r_i]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 异号, 商 0	
\leftarrow	0 0.1 1 0 1 0		左移一位	
$+ [Y]_{\text{补}}$	1 1.0 0 1 0 1		$+ [Y]_{\text{补}}$	
	1 1.1 1 1 1 1	0.0 1 1 0 <u>1</u>	$[r_i]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 同号, 商 1	
\leftarrow	1 1.1 1 1 1 0		左移一位	
$+ [-Y]_{\text{补}}$	0 0.1 1 0 1 1		$[X]_{\text{补}}$ 与 $[Y]_{\text{补}}$ 同号, $+ [-Y]_{\text{补}}$	
	0 0.1 1 0 0 1	0.1 1 0 1 <u>1</u>	末位恒置 1	

$\therefore [商]_{\text{补}} = 0.11011$

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, 计算下列各题: [注: 此题中阶码用二进制表示]

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

(2) $X = 2^{-101} \times 0.101100$, $Y = 2^{-100} \times (-0.101000)$

(3) $X = 2^{-011} \times 0.101100$, $Y = 2^{-001} \times (-0.111100)$

求: $X+Y$, $X-Y$

[解]

(1) 浮点数: $[X]_{\text{浮}} = 0101; 1.011110$

$[Y]_{\text{浮}} = 0100; 1.000010$

对阶: 小阶向大阶看齐, $\Delta E = E_A - E_B = 1$,

$[Y]_{\text{浮}}' = 0101; 1.100001$

对阶之后, 尾数相加和相减。

相加:	11.011110	相减:	11.011110
	+ 11.100001		+ 00.011111
	<u>10.111111</u>		<u>11.111101</u>

规格化:

相加结果需要右规 1 次, 得: $[X+Y]_{\text{浮}} = 0110; 1.011111$

相减结果需左规 4 次, 得: $[X-Y]_{\text{浮}} = 0001; 1.010000$

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

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

(2) 浮点数: $[X]_{\text{浮}} = 1011; 0.101100$

$[Y]_{\text{浮}} = 1100; 1.011000$

对阶: 小阶向大阶看齐, $\Delta E = E_A - E_B = -1$,

$[X]_{\text{浮}}' = 1100; 0.010110$

对阶之后, 尾数相加和相减。

相加:	00.010110	相减:	00.010110
	+ 11.011000		+ 00.101000
	<u>11.101110</u>		<u>00.111110</u>

规格化:

相加结果需要左规一次, 得: $[X+Y]_{\text{浮}} = 1011; 1.011100$

$\therefore X+Y = 2^{-101} \times (-0.100100)$

$$X-Y = 2^{-100} \times 0.111110$$

(3) 浮点数: $[X]_{\text{浮}}=1101; 0.101100$ $[Y]_{\text{浮}}=1111; 1.000100$

对阶后: $[X]_{\text{浮}}'=1111; 0.001011$

尾数相加减后: $[X+Y]_{\text{浮}}=1111; 11.001111$ $[X-Y]_{\text{浮}}=1111; 01.000111$

尾数规格化后: $[X+Y]_{\text{浮}}=1111; 1.001111$ $[X-Y]_{\text{浮}}=0000; 0.100011$

$$\therefore X+Y = 2^{-001} \times (-0.110001)$$

$$X-Y = 2^{000} \times 0.100011$$

第5章 存储系统和结构 (教材P164)

5-17. 用容量为 $16K \times 1$ 的 DRAM 芯片构成 64KB 的存储器。

(2) 设存储器的读、写周期均为 $0.5\mu\text{S}$, CPU 的 $1\mu\text{S}$ 内至少要访存一次, 试问采用那种刷新方式比较合理? 相邻两行之间的刷新时间间隔是多少? 对全部存储单元刷新一遍所需的实际时间是多少?

[解] 因为要求 CPU 在 $1\mu\text{S}$ 内至少要访存一次, 所以不能使用集中刷新方式, 分散和异步刷新方式都可以使用, 但异步刷新方式比较合理。

$16K \times 1$ 的 DRAM 芯片需要 14 位地址, 所以行地址和列地址各 7 位, 芯片存储阵列的行数为 $2^7=128$, 列数也是 128。

相邻两行之间的刷新时间间隔 = 最大刷新时间 / 行数

$$= 2\text{ms}/128 = 15.625\mu\text{S}$$

取 $15.5\mu\text{S}$, 即进行读或写操作 31 次之后刷新一行。

对全部存储单元刷新一遍所需的实际刷新时间 = $0.5\mu\text{S} \times 128 = 64\mu\text{s}$ 。

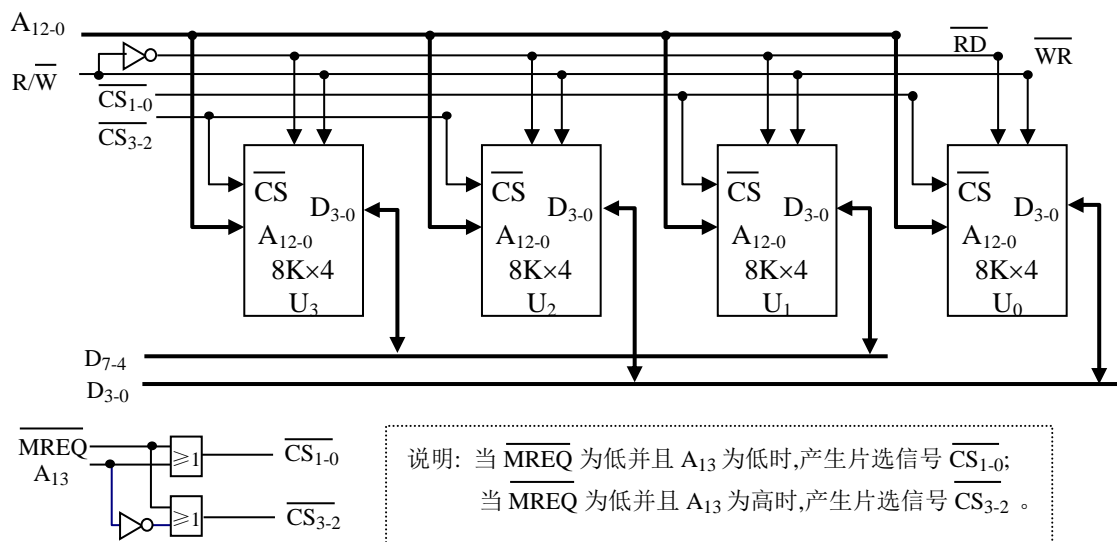
(每刷一行需占用 $0.5\mu\text{S}$ 的时间, 与读写周期相同)

补充 5-1. 有 $8K \times 4$ 的静态 RAM 芯片, 试组成一个 $16K \times 8$ 的随机存储器, 问需要多少芯片? 片选信号怎样产生? 若存储芯片的读写采用两条线 ($\overline{\text{RD}}$ 和 $\overline{\text{WR}}$) 进行控制, 而系统提供的控制信号有 $\overline{\text{R/W}}$ (高为读, 低为写) 和 $\overline{\text{MREQ}}$ (为低时表示访问内存), 试画出该存储器的组成逻辑图。

[解] 需要 RAM 芯片 4 片;

各芯片的片内地址线接外部提供的地址线的低 13 位 A_{12-0} ;

4 片 RAM 的双向数据线按每两片 (U_1 和 U_0 , U_3 和 U_2) 组成位扩展的方式分别接入数据线上, U_1 和 U_0 的片选线及 U_3 和 U_2 的片选线及其逻辑见存储器的组成逻辑图如下:



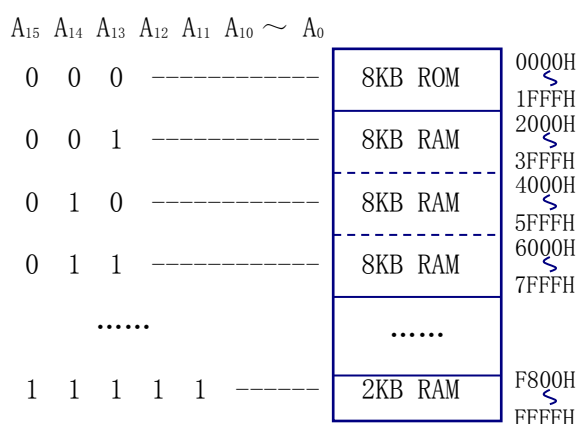
5-18. 有一个8位机,采用单总线结构,地址总线16位($A_{15} \sim A_0$),数据总线8位($D_7 \sim D_0$),控制总线中与主存有关的信号有 \overline{MREQ} (低电平有效允许访存) 和 R/\overline{W} (高电平为读命令,低电平为写命令)。

主存地址分配如下: 从 0~8191 为系统程序区, 有 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)。

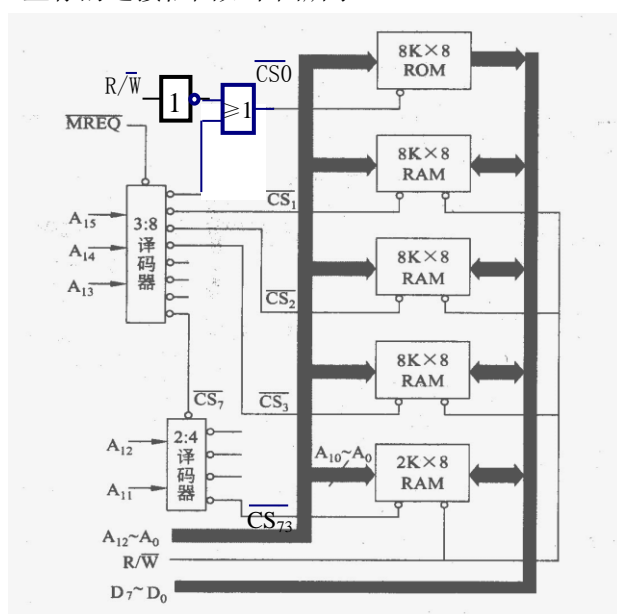
现有如下存储芯片: $8K \times 8$ 的 ROM, $16K \times 1$ 、 $2K \times 8$ 、 $4K \times 8$ 、 $8K \times 8$ 的 SRAM。请从上述规格中选用芯片设计该主存储器, 画出主存的连接框图, 并指出片选逻辑及与 CPU 的连接。

[解] 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。共需: $8K \times 8$ 的 ROM 芯片 1 片, $8K \times 8$ 的 SRAM 芯片 3 片, $2K \times 8$ 的 SRAM 芯片 1 片。

主存地址分配如下图所示。



主存的连接框图如下图所示。



片选信号的逻辑:

$$\overline{CS_0} = \overline{A_{15}} \overline{A_{14}} \overline{A_{13}} \overline{MREQ} R/\overline{W}$$

$$\overline{CS_1} = \overline{A_{15}} \overline{A_{14}} \overline{A_{13}} \overline{MREQ}$$

$$\overline{CS_2} = \overline{A_{15}} \overline{A_{14}} \overline{A_{13}} \overline{MREQ}$$

$$\overline{CS_3} = \overline{A_{15}} \overline{A_{14}} \overline{A_{13}} \overline{MREQ}$$

$$\overline{CS_{73}} = \overline{A_{15}} \overline{A_{14}} \overline{A_{13}} \overline{A_{12}} \overline{A_{11}} \overline{MREQ}$$

5-25. 设某机主存容量为 4MB, Cache 容量为 16KB, 每块包含 8 个字, 每字 32 位, 设

计一个四路组相联映像（即 Cache 每组内共有四个块）的 Cache 组织，要求：

- (1) 画出主存地址字段中各段的位数。
- (2) 设 Cache 的初态为空，CPU 需依次从主存第 0, 1, 2, ……，(99?) 号单元读出 100 个字（主存一次读出一个字。），并重复按此次序读 8 次，问命中率是多少？

(3) 若 Cache 的速度是主存的 6 倍，试问有 Cache 和无 Cache 相比，速度提高多少倍？、

[解]

- (1) 主存容量为 4MB，按字节编址，所以主存地址为 22 位，地址格式如下图所示：

区号 (8 位)	组号 (7 位)	组内块号 (2 位)	块内地址 (5 位)
----------	----------	------------	------------

- (2) Cache 的地址格式为

组号 (7 位)	组内块号 (2 位)	块内地址 (5 位)
----------	------------	------------

主存第 0, 1, 2, ……，(99?) 号单元在第 0 区前 0~12 块中，被映射到 Cache 的第 0~12 块（在 0~3 组）中，不存在映射时的组冲突。（参考教材 P178 图 5-29C）

由于 Cache 起始为空，所以第一遍读时每一块中的第一个单元没命中（此时要把该块的 8 个字都调入 Cache，这样 CPU 之后再读该块的其它单元就会命中（未考虑块调入过程）），而后面七遍读取时每个单元均可以命中。

$$\text{命中率} = N_c / (N_c + N_m) = (100 - 13 + 7 \times 100) / (8 \times 100) = 98.4\%$$

- (3) 设 Cache 的存取周期为 T，则主存的存取周期为 6T。

$$\text{有 Cache 的访存时间} = H \times T_c + (1 - H) \times (T_m + T_c) = T + (1 - 98.4\%) \times 6T = 1.096T$$

无 Cache 的访存时间为 6T

$$\text{所以提高后的速度倍数} = 6 / 1.096 = 5.47 \text{ 倍。}$$

第6章 中央处理器（教材P209）

6-6. 某机 CPU 芯片的主振频率为 8MHz，其时钟周期是多少 uS？若已知每个机器周期平均包含 4 个时钟周期，该机的平均指令执行速度为 0.8MIPS，试问：

- (1) 平均指令周期是多少？
- (2) 平均每个指令周期含有多少个机器周期？
- (3) 若改用时钟周期为 0.4uS 的 CPU 芯片，则计算机的平均指令执行速度又是多少 MIPS？
- (4) 若要得到 40 万次/S 的指令执行速度，则应采用主振频率为多少 MHz 的 CPU 芯片？

[解] 时钟周期 = $1 \div 8\text{MHz} = 0.125\mu\text{S}$

- (1) 平均指令周期 = $1 \div 0.8\text{MIPS} = 1.25\mu\text{S}$

- (2) 机器周期 = $0.125\mu\text{S} \times 4 = 0.5\mu\text{S}$

$$\text{平均每个指令周期的机器周期数} = 1.25\mu\text{S} \div 0.5\mu\text{S} = 2.5$$

- (3) 平均指令执行速度 = $1 \div (0.4 \times 4 \times 2.5) = 0.25 \text{ MIPS}$

- (4) 主振频率 = $8\text{MHz} \times (0.4\text{MIPS} / 0.8\text{MIPS}) = 4\text{MHz}$

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

[解]

- (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: 存放写入到内存或从内存读出的数据或指令
- (5) 存储器地址寄存器 MAR: 存放写入到内存或从内存读出的数据或指令的地址

以单地址指令“加 1 (INC A)”为例, 该指令分为 3 个周期: 取指周期、分析取数周期、执行周期。3 个周期完成的操作如表所示。

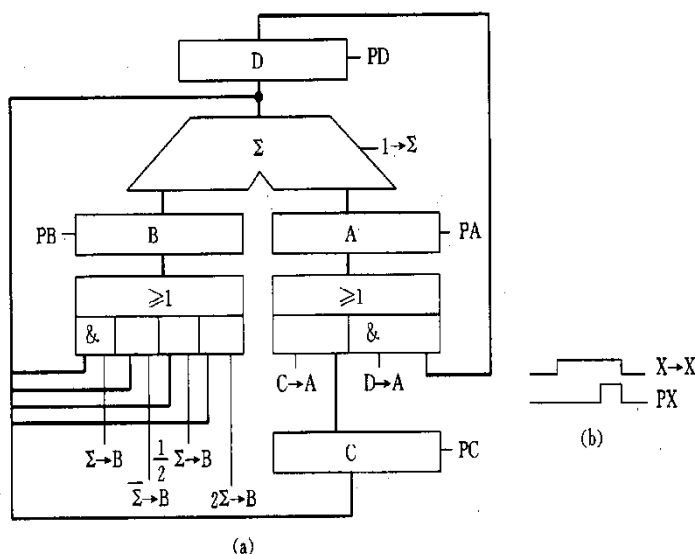
	取指周期	分析取数周期	执行周期
PC	(PC) → MAR	—	—
IR	指令 → MDR → IR	—	—
ALU	(PC) + 1	—	(A) + 1
MAR	指令地址 → MAR	A → MAR	—
MDR	指令 → MDR	(A) → MDR	(A) + 1 → MDR

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?

[解] 指令和数据都存放在主存, 它们都以二进制代码形式出现, 区分方法为:

- (1) 取指令或数据时所处的机器周期不同: 取指令周期取出的是指令; 分析取数或执行周期取出的是数据。
- (2) 取指令或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器件。

补充 6-1. 下图 (a) 为某模型机数据加工通路。其中 Σ 为并行加法器, $1 \rightarrow \Sigma$ 为最低位的进位。A, B, C, D 为四个寄存器, PA, PB, PC, PD 分别为四个寄存器的数据接收信号, 且均为脉冲信号。图中其余的 $X \rightarrow X$ 控制信号均为电平信号, 电平信号与脉冲信号的时间关系如下图 (b) 所示。试拟出在该图上实现下列运算所需的微操作。



图中其余的 $X \rightarrow X$ 控制信号均为电平信号, 电平信号与脉冲信号的时间关系如下图 (b) 所示。试拟出在该图上实现下列运算所需的微操作。

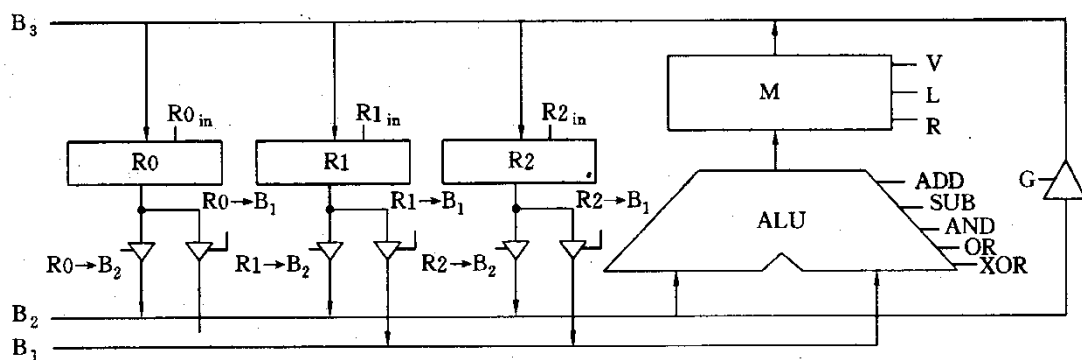
- (1) $D - C \rightarrow D$
- (2) $D/2 + C \rightarrow D$
- (3) $D - 1 \rightarrow D$
- (4) $2C + 1 \rightarrow C$

[解]

功能说明	微操作控制信号
(1) ① $C \rightarrow A, 0 \rightarrow B$	$C \rightarrow A, PA, PB$
② $\overline{A} \rightarrow B, D \rightarrow A$	$\overline{\Sigma} \rightarrow B, PB, D \rightarrow A, PA$ (此步也可分为两步)
③ $A+B+1 \rightarrow D$	$1 \rightarrow \Sigma, PD$
(2) ① $D \rightarrow A, 0 \rightarrow B$	$D \rightarrow A, PA, PB$
② $A/2 \rightarrow B, C \rightarrow A$	$1/2 \Sigma \rightarrow B, PB, C \rightarrow A, PA$ (此步也可分为两步)
③ $A+B \rightarrow D$	PD
(3) ① $D \rightarrow A$	$D \rightarrow A, PA$
② “FF” (全 1) $\rightarrow B$	$\Sigma \rightarrow B, \overline{\Sigma} \rightarrow B, PB$ (此步也可分为两步)
③ $A+B \rightarrow D$	PD
(4) ① $C \rightarrow A, “0” \rightarrow B$	$C \rightarrow A, PA, PB$
② $2A \rightarrow B, “0” \rightarrow A$	$2\Sigma \rightarrow B, PB, PA$ (此步也可分为两步)
③ $A+B+1 \rightarrow C$	$1 \rightarrow \Sigma, PC$

补充 6-2. 某机运算器为三总线结构，三总线分别称为 B_1, B_2, B_3 ，连接 B_2, B_3 的控制信号为 G 。算逻辑部件 ALU 可进行 ADD, SUB, AND, OR, XOR 五种运算，输出多路器可进行直送(V)，左移一位(L)，右移一位(R)三种操作。三个通用寄存器 R_0, R_1, R_2 都有输入和输出信号，如下图所示。

试写出实现下列功能所需的微操作： $1/2[R_1-R_2] \rightarrow R_1$



[解] 完成该微操作只需一步

功能说明	微操作控制信号
① $[R_1-R_2]$ 再右移后 $\rightarrow R_1$	$R_1 \rightarrow B_1, R_2 \rightarrow B_2, SUB, R, R1in$

补充 6-3. 何谓同步控制？何谓异步控制？试比较它们的特点及应用场合。

答：同步控制方式有统一的时序信号，每一个操作均在规定的操作时间内完成；异步控制方式无统一的时序信号，操作之间切换由应答方式进行，每个操作时间根据需要，需长则长，能短则短。

同步控制方式多用于部件内部的时序控制（比如 CPU 内部）；异步控制方式多用于不同部件之间的操作（例如 CPU 与外设之间）

第7章 输入输出系统（教材第9章 P313）

9-4. 程序查询方式、程序中断方式、DMA 方式各自适用什么范围？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。

[解] 程序查询方式是主机与外设间进行信息交换的最简单方式，输入和输出完全是通过 CPU 执行程序来完成的。这种方式控制简单，但外设和主机不能同时工作，系统效率很低，仅适用于外设的数目不多、对 I/O 处理的实时要求不高、CPU 的操作任务比较单一、并不很忙的情况。

程序中断方式无需查询等待，外设在做好输入输出准备时，向主机发中断请求，主机接到请求后就暂时中止原来执行的程序，转去执行中断服务程序对外部请求进行处理，在中断处理完毕后返回原来的程序继续执行。程序中断不仅适用于外部设备的输入输出操作，也适用于对外界发生的随机时间的处理。由于完成一次程序中断还需要许多辅助操作，因此主要适用于中、低速外设。

DMA 方式是在主存与外设之间（注意不是主机与外设之间）开辟的数据通路，可以进行基本上不需要 CPU 介入的主存和外设之间的信息传送，这样不仅保证 CPU 的高效率，而且能满足高速外设的需要。DMA 方式只能进行简单的数据传送操作，在数据块传送的起始和结束时还需 CPU 及中断系统进行预处理和后处理。

由此可见：

题中（1）的结论不正确。程序查询方式接口简单，可用于外设与主存速度相差不大，且外设数量较少的情况。

题中（2）的结论也不正确。DMA 方式在结束一批数据传输后仍需程序中断方式做后处理。此外，DMA 方式每次只能作简单的数据传输，不能象中断方式那样可以作较复杂的处理。

补充. 举例说明程序查询方式的工作原理。

[解] 以从键盘上读取一个数据为例，其过程是：CPU 首先启动键盘工作，然后测试键盘状态，若键盘数据未准备就绪，则输入缓冲寄存器的内容不可以使用，继续查询；若键盘数据已准备就绪，则执行输入指令取走该数据。这种方式的优点是控制简单，节省硬件，缺点是系统效率低。