



第六章 数据表示与运算 习题与解答

第六章 6.1



□ 6.1 将十进制数127.125和25/32转换成二进制数，然后再转换成八进制和十六进制数。

□ 解：此题可采用简便方法转换如下：

$$127.125 = 128 - 1 + 0.125 = 2^7 - 1 + 2^{-3} = (1111111.001)_2$$

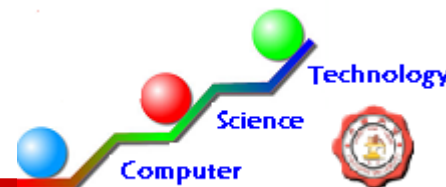
$$=(001, 111, 111.001)_2 = (177.1)_8$$

以小数点为起点，向左、向右三位一组分组，然后用八进制缩写。

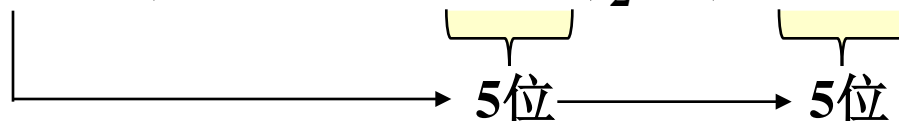
$$=(0111, 1111.0010)_2 = (7F.2)_{16}$$

以小数点为起点，向左、向右四位一组分组，小数末位补零凑足四位，然后用十六进制缩写。

第六章 6.1



$$\square \quad 25/32 = 25 \times 2^{-5} = (11001 \times 0.00001)_2 = (0.11001)_2$$



$$=(0.110, 010)_2 = (0.62)_8 \quad \text{—— 转换方法同上。}$$

$$=(0.1100, 1000)_2 = (0.C8)_{16} \quad \text{—— 转换方法同上。}$$

- 评注：数制转换的基本方法已在前导课程《数字逻辑》中进行过充分讨论，在《组成》中要求熟练应用，因此，方法上就应更灵活、简便、快速，技巧性更高，更具变通性。这类方法基本建立在数的按权展开多项式的基础上。

第六章 6.2



□ 6.2 把下列各数转换为十进制数。

(1) $(101.100\ 11)_2$; (2) $(1\ 101\ 000\ 110.101\ 01)_2$

(3) $(1702.32)_8$; (4) $(247.63)_8$

(5) $(F5B.48)_{16}$; (6) $(AD.4)_{16}$

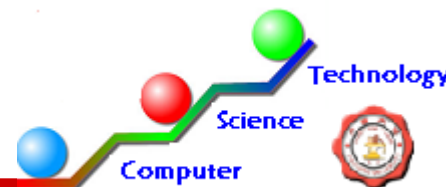
□ 解:

(1) **5.59375**; (2) **838.65625**

(3) **962.40625**; (4) **167.796875**

(5) **3931.28125**; (6) **173.25**

第六章 6.3



□ 6.3 设某十进制数 S ，在八进制中写成 $\sum_{i=0}^m L_i \times 8^i$ ，

在二进制中写成 $S = \sum_{j=0}^n K_j \times 2^j$ 。若令： $n = 3m + 2$ ，

试证： $(L_0)_8 = (K_2 K_1 K_0)_2$

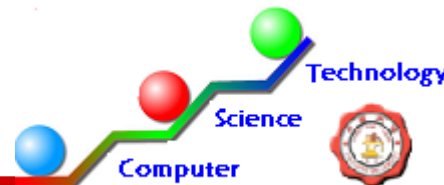
$(L_1)_8 = (K_5 K_4 K_3)_2$

\vdots

$(L_m)_8 = (K_n K_{n-1} K_{n-2})_2$

□ 评注：此题从理论上推导了二、八进制间的转换关系，用同样方法可推导二、十六进制间的转换关系。

第六章 6.3



□ 证：思路：先将 $(S)_2$ 按权展开，然后整理成位权为8的形式，既得 $(S)_8$ 。

$$(S)_2 = K_n \times 2^n + K_{n-1} \times 2^{n-1} + \dots + K_j \times 2^j + \dots + K_2 \times 2^2 + K_1 \times 2^1 + K_0 \times 2^0$$

将 $n = 3m + 2$ 代入2的幂得：

$$\begin{aligned}(S)_2 &= K_n \times 2^{3m+2} + K_{n-1} \times 2^{3m+1} + K_{n-2} \times 2^{3m} + \dots + K_j \times 2^{3i+2} + K_{j-1} \times 2^{3i+1} + \\ &\quad K_{j-2} \times 2^{3i} + \dots + K_2 \times 2^2 + K_1 \times 2^1 + K_0 \times 2^0 \\ &= (K_n \times 2^2 + K_{n-1} \times 2^1 + K_{n-2}) \times 2^{3m} + \dots + (K_j \times 2^2 + K_{j-1} \times 2^1 + K_{j-2}) \times 2^{3i} \\ &\quad + \dots + (K_5 \times 2^2 + K_4 \times 2^1 + K_3) \times 2^{3 \times 1} + (K_2 \times 2^2 + K_1 \times 2^1 + K_0) \times 2^{3 \times 0} \\ &= (K_n K_{n-1} K_{n-2}) \times 8^m + \dots + (K_j K_{j-1} K_{j-2}) \times 8^i + \dots + (K_5 K_4 K_3) \times 8^1 \\ &\quad + (K_2 K_1 K_0) \times 8^0 = (S)_8\end{aligned}$$

$$\text{故 } (L_0)_8 = (K_2 K_1 K_0)_2$$

$$(L_1)_8 = (K_5 K_4 K_3)_2$$

$$\vdots$$

$$(L_m)_8 = (K_n K_{n-1} K_{n-2})_2$$

证毕。

第六章 6.4



□ 6.4 用八位二进制数（含1位符号）表示下列各数的原码、反码、补码、移码。如果是小数，小数点在符号位之后；如果是整数，小数点在LSB之后。

(1) $-35/64$; (2) $23/128$; (3) -127 ;

(4) 用小数表示 -1 ; (5) 用整数表示 -1

□ 解：根据各种机器码与真值关系的远近程度，可按下述顺序及简易规则进行转换：

数制转换

符号代码化

正数不变
负数按位
变反

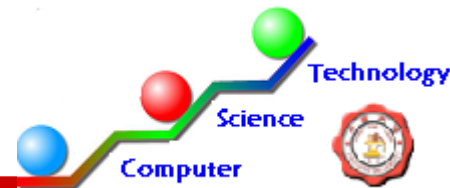
正数不变
负数反码
末位加1

补码
符号位
变反

十进制真值 → 二进制真值 → 原码 → 反码 → 补码 → 移码

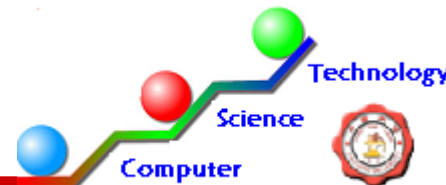
□ 各数的机器码表示列表如下：

第六章 6.4 各数机器码表



	十进制真值	二进制真值	原 码	反 码	补 码	移 码
(1)	-35/64	-0.100 0110	1.100 0110	1.011 1001	1.011 1010	0.011 1010
(2)	23/128	0.001 0111	0.001 0111	0.001 0111	0.001 0111	1.001 0111
(3)	-127	-0111 1111	1 111 1111	1 000 0000	1 000 0001	0 000 0001
(4)	小数表示-1	-1.000 0000	无	无	1.000 0000	0.000 0000
(5)	整数表示-1	-0000 0001	1 000 0001	1 111 1110	1 111 1111	0 111 1111

第六章 6.5



□ 6.5 对于模4补码，设 $[X]_{\text{补}} = X_s X_0 X_1 X_2 \dots X_n$

求证： $X = -2X_s + X_0 + \sum_{i=1}^n X_i \times 2^{-i}$ 。

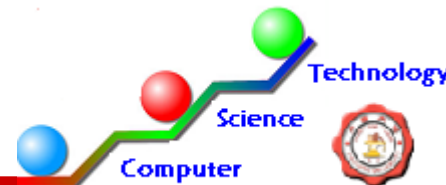
□ 证：据模4补码定义： $[X]_{\text{补}} = \begin{cases} X & 2 > X \geq 0 \\ 4+X & 0 > X \geq -2 \end{cases} \pmod{4}$

当 $2 > X \geq 0$ 时， $X_s = 0$ ，

$$\begin{aligned} X &= [X]_{\text{补}} = 0 X_0 X_1 X_2 \dots X_n \text{——位置表示法} \\ &= X_0 + \sum_{i=1}^n X_i \times 2^{-i} = -2X_s + X_0 + \sum_{i=1}^n X_i \times 2^{-i} \end{aligned}$$

↓ ↓
多项式表示法 → 配项

第六章 6.5



□ 当 $0 > X \geq -2$ 时, $X_s = 1$,

$$[X]_{\text{补}} = 1 X_0 X_1 X_2 \dots X_n = 2 + X_0 + \sum_{i=1}^n X_i \times 2^{-i} = 4 + X$$

$$\begin{aligned} \text{则 } X &= [X]_{\text{补}} - 4 = -4 + 2 + X_0 + \sum_{i=1}^n X_i \times 2^{-i} = -2 + X_0 + \sum_{i=1}^n X_i \times 2^{-i} \\ &= -2X_s + X_0 + \sum_{i=1}^n X_i \times 2^{-i} \end{aligned}$$



配项

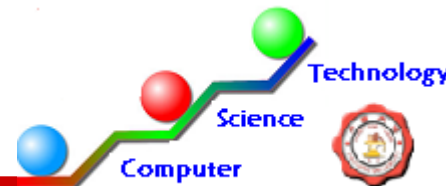
综上所述可知：无论 X 为正还是为负，均有：

$$X = -2X_s + X_0 + \sum_{i=1}^n X_i \times 2^{-i}$$

□ 证毕。

□ 评注：本题证明了真值与模4补码之间的转换公式，此类题基本上都是利用补码定义进行求证。

第六章 6.6

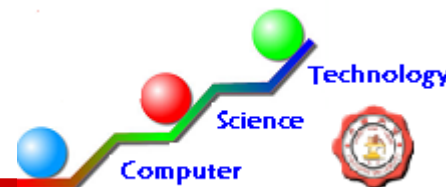


□ 6.6 设 X 为整数, $[X]_{\text{补}}=1$, $X_1X_2X_3X_4X_5$, 若要求 $X < -16$, 试问 $X_1 \sim X_5$ 应取何值?

□ 解: 若要 $X < -16$, 需 $X_1=0$, $X_2 \sim X_5$ 任意。

□ 注: 负数绝对值大的反而小。

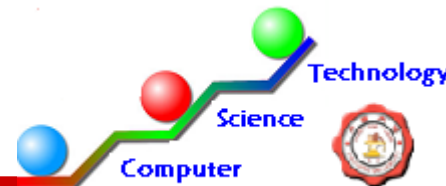
第六章 6.7



- 6.7 用除2取余法求： $(2019)_{10} = (\quad)_2$?
用乘2取整法求： $(0.543)_{10} = (\quad)_2$?
用减权定位法求： $(4091.629)_{10} = (\quad)_2$?
用按权相加法求： $(1010110101001.110100 \ 1)_2 = (\quad)_{10}$?
(注：十→二按0舍1入法取五位小数。)

- 解： $(2019)_{10} = (11 \ 111 \ 100 \ 011)_2$;
 $(0.543)_{10} = (0.100 \ 01)_2$;
 $(4091.629)_{10} = (111 \ 111 \ 111 \ 011.101)_2$;
 $(1 \ 010 \ 110 \ 101 \ 001.110 \ 100 \ 1)_2 = (5535.8203125)_{10}$

第六章 6.8



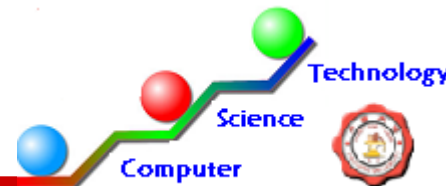
□ 6.8 已知数的补码表示，求数的原码与真值。

$$[X_1]_{\text{补}} = 00011010 \quad [X_2]_{\text{补}} = 10011010 \quad [X_3]_{\text{补}} = 11110001$$

□ 解：已知数的补码表示，数的原码与真值见下表：

补 码 $[X]_{\text{补}}$	原 码 $[X]_{\text{原}}$	真 值
0 001 1010	同补码	同补码
1 001 1010	1 110 0110	-110 0110
1 111 0001	1 000 1111	-000 1111

第六章 6.9



□ 6.9 讨论若 $[X]_{\text{补}} > [Y]_{\text{补}}$ ，是否有 $X > Y$ ？

□ 解：

□ 若 $[X]_{\text{补}} > [Y]_{\text{补}}$ ，**不一定**有 $X > Y$ 。

□ $[X]_{\text{补}} > [Y]_{\text{补}}$ 时 $X > Y$ 的**结论只在 $X > 0$ 、 $Y > 0$ ，及 $X < 0$ 、 $Y < 0$ 时成立。**

□ 当 $X > 0$ 、 $Y < 0$ 时，有 $X > Y$ ，但由于负数补码的符号位为1，则 $[X]_{\text{补}} < [Y]_{\text{补}}$ 。

□ 同样，当 $X < 0$ 、 $Y > 0$ 时，有 $X < Y$ ，但 $[X]_{\text{补}} > [Y]_{\text{补}}$ 。

第六章 6.10



□ 6.10 设 $[X]_{\text{补}} = a_0.a_1a_2a_3a_4a_5a_6$ ，其中 a_i 取0或1，若要 $X > -0.5$ ，求 $a_0, a_1, a_2, \dots, a_6$ 的取值。

□ 解：根据补码结构特点知：

- (1) a_0 为符号位，因此可任取0或1；
- (2) 当 $a_0=0$ 时，为正数，必有 $X > -0.5$ ，因此 $a_1 \sim a_6$ 可任取0或1；
- (3) 当 $a_0=1$ 时，为负数，必须有： $a_1=1, a_2+a_3+a_4+a_5+a_6=1$
(即 $a_2 \sim a_6$ 不全为0)，才满足 $X > -0.5$ 的条件。

□ 评注：当 X 为负数，且 $X > -0.5$ 时，其绝对值小于0.5，则据定义必有 $[X]_{\text{补}} > (1.5)_{10}$ 。作此题需注意：

- ① 负数的绝对值越小，补码值越大；
- ② 临界值0.5不在题意要求的范围内，条件 $a_2 \sim a_6$ 不全为0就是为此而设。

第六章 6.11



□ 6.11 已知 $X=0.a_1a_2a_3a_4a_5a_6$ (a_i 为0或1)，讨论下列几种情况时 a_i 各取何值。

(1) $X > 1/2$; (2) $X \geq 1/8$; (3) $1/4 \geq X > 1/16$

□ 解:

(1) 若要 $X > 1/2$ ，只要 $a_1=1$ ， $a_2 \sim a_6$ 不全为0即可

(a_2 or a_3 or a_4 or a_5 or $a_6 = 1$) ;

(2) 若要 $X \geq 1/8$ ，只要 $a_1 \sim a_3$ 不全为0即可

(a_1 or a_2 or $a_3 = 1$) , $a_4 \sim a_6$ 可任取0或1;

(3) 若要 $1/4 \geq X > 1/16$ ，只要 $a_1=0$ ， a_2 可任取0或1;

当 $a_2=0$ 时，若 $a_3=0$ ，则必须 $a_4=1$ ，且 a_5 、 a_6 不全为0

(a_5 or $a_6 = 1$) ;

若 $a_3=1$ ，则 $a_4 \sim a_6$ 可任取0或1;

当 $a_2=1$ 时， $a_3 \sim a_6$ 可任取0或1。

第六章 6.12



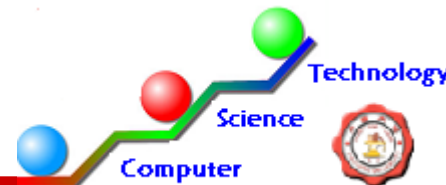
□ 6.12 当十六进制数9AH，80H和FFH分别表示原码、补码、反码、移码和无符号数时，对应的十进制真值各为多少（设机器数采用一位符号位）？

□ 解：真值和机器数的对应关系如下：

十六进制	真值	无符号数	原码	反码	补码	移码
9AH	二进制 十进制	1001 1010 154	-001 1010 -26	-110 0101 -101	-110 0110 -102	001 1010 26
80H	二进制 十进制	1000 0000 128	- 000 0000 - 0	-111 1111 -127	-1000 0000 -128	000 0000 0
FFH	二进制 十进制	1111 1111 255	-111 1111 -127	-000 0000 -0	-000 0001 -1	111 1111 127

□ 注意：9AH、80H、FFH为机器数，本身含符号位。

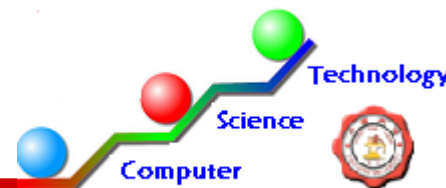
第六章 6.13



□ 6.13 设机器数字长为16位，写出下列各种情况下它能表示的数的范围。机器数采用一位符号位，答案均用十进制2的幂形式表示。

- (1) 无符号整数；
- (2) 原码表示的定点小数；
- (3) 补码表示的定点小数；
- (4) 原码表示的定点整数；
- (5) 补码表示的定点整数。

第六章 6.13



□ 解：各种表示方法的数据范围如下：

(1) 无符号整数： $0 \sim 2^{16} - 1$ ，即： $0 \sim 65535$ ；

(2) 原码定点小数： $-(1 - 2^{-15}) \sim 1 - 2^{-15}$

(3) 补码定点小数： $-1 \sim 1 - 2^{-15}$

(4) 原码定点整数： $-(2^{15} - 1) \sim 2^{15} - 1$

即： $32767 \sim -32767$ ；

(5) 补码定点整数： $-2^{15} \sim 2^{15} - 1$

即： $32767 \sim -32768$ ；

注意：

1) 应写出可表示范围的上下限精确值（用 \geq 或 \leq ，不要用 $>$ 或 $<$ ）；

2) 不要用十进制小数表示，不直观不精确且无意义；

3) 原码正、负域对称，补码正、负域不对称。

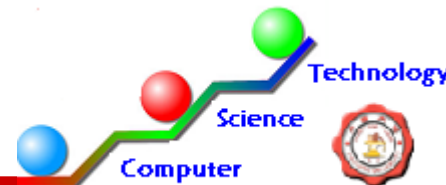
第六章 6.14



□ 6.14 设机器字长为8位（含一位符号位），分整数和小数两种情况讨论真值 X 为何值时， $[X]_{\text{补}} = [X]_{\text{原}}$ 成立。

□ 解：当 X 为小数时，若 $X \geq 0$ ，则据定义有
 $[X]_{\text{补}} = [X]_{\text{原}}$ 成立；
若 $X < 0$ ，则当 $X = -1/2$ 时，
 $[X]_{\text{补}} = [X]_{\text{原}}$ 成立。
当 X 为整数时，若 $X \geq 0$ ，则
 $[X]_{\text{补}} = [X]_{\text{原}}$ 成立；
若 $X < 0$ ，则当 $X = -64$ 时，
 $[X]_{\text{补}} = [X]_{\text{原}}$ 成立。

第六章 6.15



□ 6.15 用以下形式表示十进制数5862。

(1) 二进制数；(2) 8421码；(3) 余3码；(4) 2421码。

□ 解：

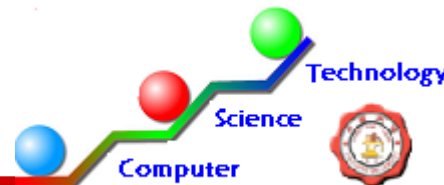
$$(1) (5862)_{10} = (1\ 0110\ 1110\ 0110)_2 = 16E6H$$

$$(2) (5862)_{10} = (0101\ 1000\ 0110\ 0010)_{8421} = 5862H$$

$$(3) (5862)_{10} = (1000\ 1011\ 1001\ 0101)_{E3} = 8B95H$$

$$(4) (5862)_{10} = (1011\ 1110\ 1100\ 0010)_{2421} = BEC2H$$

第六章 6.16



□ 6.16 用前分隔数字串表示法、后嵌入数字串表示法和压缩的十进制数串表示法表示下列十进制数，设存储器按字节编址。

+1980; -76543; +254; -1992

□ 解:

(1) 前分隔数字串表示: (ASCII码用十六进制表示, 下同)

+1980:

2Bh	31h	39h	38h	30h
-----	-----	-----	-----	-----

-76543:

2Dh	37h	36h	35h	34h	33h
-----	-----	-----	-----	-----	-----

+254:

2Bh	32h	35h	34h
-----	-----	-----	-----

-1992:

2Dh	31h	39h	39h	32h
-----	-----	-----	-----	-----

第六章 6.16



(2) 后嵌入数字串表示:

+1980:

31h	39h	38h	30h
-----	-----	-----	-----

-76543:

37h	36h	35h	34h	73h
-----	-----	-----	-----	-----

+254:

32h	35h	34h
-----	-----	-----

-1992:

31h	39h	39h	72h
-----	-----	-----	-----

(3) 压缩十进制数串表示:

+1980:

0	1	9	8	0	Ch
---	---	---	---	---	----

-76543:

7	6	5	4	3	Dh
---	---	---	---	---	----

+254:

2	5	4	Ch
---	---	---	----

-1992:

0	1	9	9	2	Dh
---	---	---	---	---	----

第六章 6.17



□6.17 用十六进制写出大写字母“F”、小写字母“a”和星号“*”的ASCII码。当最高位用作偶校验位时，写出它们的ASCII机内码字节。

解：通过查ASCII编码表（表2B-1），可得

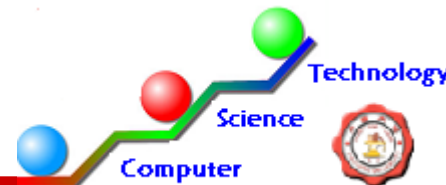
- 大写字母“F”的ASCII码为**46H**，当最高位用作偶校验位时，其ASCII机内码字节最高位为“1”，
46H+80H=C6H
- 同样，小写字母“a”的ASCII码为**61H**，机内码**E1H**
- 符号“*”的ASCII码为**2AH**，机内码为**AAH**

第六章 6.18



- 6.18 汉字“大”和“小”的国标区位码分别为2083和4801，要求
- (1) 分别写出这两个字对应的国标码；
- (2) 若采用汉字两个字节的最高位均设为“1”的机内表示方案，分别写出这两个字的机内码形式。
- 解：
- (1) 在已知区位码的情况下，只要将区码和位码分别转换成十六进制表示，然后再分别加上20H即可得到国标码。
- $2083 \rightarrow (1453H + 2020H) = 3473H$
- $4801 \rightarrow (3001H + 2020H) = 5021H$
- 则“大”字的国标码为3473H，“小”字的国标码为5021H。
- (2) 当采用汉字两个字节的最高位均设为“1”的机内表示方案时，只要将国标码的两个字节分别加上80H即可得其机内码。
- $3473H + 8080H = B4F3H$ ； $5021H + 8080H = D0A1H$
- 则“大”字的机内码为B4F3H，“小”字的机内码为D0A1H。

第六章 6.19



□ 6.19 用向量表示法，在32位字长的存储器中，用ASCII码分别按左→右（大端方式）和右→左（小端方式）的顺序表示下列字符串：

- (1) WHAT IS THIS?
- (2) THIS IS A DISK.

□ 解：(1) 左→右：

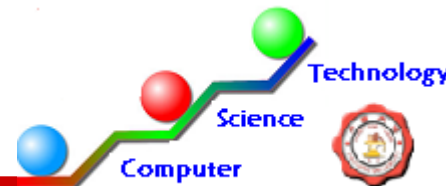
31					0
W H A T					
I S					
T H I S					
?					

右→左：

31				0
T A H W				
S I				
S I H T				
?				

(2) 方法同上。

第六章 6.20



□ 6.20 如果采用偶校验，并将校验位安排在最高位，下列数据的校验码是什么？采用奇校验呢？

(1) 010 1010 (2) 001 1011

(3) 111 0001 (4) 100 1110

□ 解：对应的校验位设置见下表：

	奇 校 验 位	偶 校 验 位	数 据
(1)	0	1	0 101 010
(2)	1	0	0 011 011
(3)	1	0	1 110 001
(4)	1	0	1 001 110

第六章 6.21

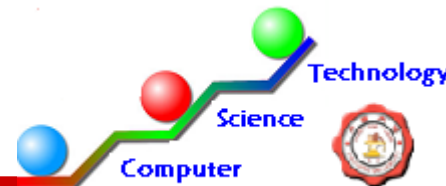


□ 6.21 设有16个信息位，如果采用海明校验，要求能分别指示无错、一位错和二位错，并纠正一位错，至少需要设置多少个校验位？应放在哪些位置上？

□ 解：设信息位数为 n ，校验位数为 k ，则据式： $2^{k-1} \geq n+k+1$ 可知，纠一检二码至少需要设置6个校验位，应放在位序为1, 2, 4, 8, 16, 22的位置上。设海明码用 H 表示，信息位用 B 表示，校验位用 P 表示，则校验位安排情况如下：

$$\begin{aligned} & H_{22} H_{21} H_{20} H_{19} H_{18} H_{17} H_{16} H_{15} H_{14} H_{13} H_{12} H_{11} H_{10} H_9 H_8 H_7 H_6 H_5 H_4 H_3 H_2 H_1 \\ & = P_6 B_{16} B_{15} B_{14} B_{13} B_{12} P_5 B_{11} B_{10} B_9 B_8 B_7 B_6 B_5 P_4 B_4 B_3 B_2 P_3 B_1 P_2 P_1 \end{aligned}$$

第六章 6.22



□ 6.22 设 $(7, 4)$ 码的生成多项式为 $G(X) = X^3 + X + 1$, 写出代码1011和0101的循环冗余校验码。

解：编码过程如下：

$$M(X)_1 = 1011, \quad M(X)_2 = 0101, \quad n = 4$$

$$G(X) = X^3 + X + 1 = 1011, \quad k+1 = 4, \quad k = 3$$

$$M(X)_1 \cdot X^3 = 1011 \ 000, \quad M(X)_2 \cdot X^3 = 0101 \ 000$$

$$M(X)_1 \cdot X^3 / G(X) = 1011 \ 000 / 1011 = 1000 + 000 / 1011$$

$$R(X)_1 = 000$$

$$M(X)_1 \cdot X^3 + R(X) = 1011 \ 000 + 000 = 1011 \ 000 = \text{CRC码}$$

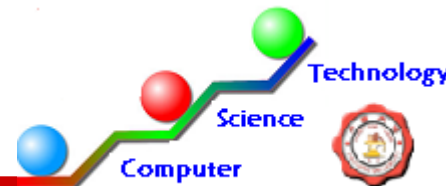
$$M(X)_2 \cdot X^3 / G(X) = 0101 \ 000 / 1011 = 0100 + 100 / 1011$$

$$R(X)_2 = 100$$

$$M(X)_2 \cdot X^3 + R(X) = 0101 \ 000 + 100 = 0101 \ 100 = \text{CRC码}$$

由于码制和生成多项式均与教材上的例题相同，故此 $(7, 4)$ 码的出错模式同教材 P_{257} 表6-8。

第六章 6.23

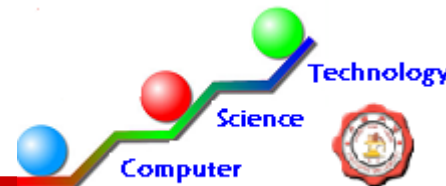


□ 6.23 已知接收到的8421海明码（按偶校验配置）为110 1100, 1 00 0100, 010 1101, 001 1010, 检查上述代码是否出错？第几位出错？若出错请写出正确代码（不考虑双错及多错情况）。

□ 解：设8421码为 $B_4B_3B_2B_1$ ，校验位为 $P_3P_2P_1$ ，海明码排列为 $B_4B_3B_2P_3B_1P_2P_1$ ，则检查结果及纠正情况如下表：

接收代码	出错情况	出错位序	正确代码
110 1100	有	B_3	100 1100
100 0100	有	P_3	100 1100
010 1101	无		
001 1010	有	B_1	001 1110

第六章 6.24



□ 6.24 有两位8421BCD码编码的十进制整数置于寄存器A中，可以通过一个加法器网络将其直接转换成二进制整数。试用半加器、全加器电路画出该加法器网络。

□ 解：算法分析：

设两位8421码 $A = A_1 A_2 = a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1$

二进制数 $B = b_7 b_6 b_5 b_4 b_3 b_2 b_1$

则
$$B = A_1 \times 1010 + A_2 = A_1 \times 1000 + A_1 \times 10 + A_2$$
$$= a_8 a_7 a_6 a_5 000 + a_8 a_7 a_6 a_5 0 + a_4 a_3 a_2 a_1$$

□ 为了更加清楚起见，进一步用竖式表示相加关系如下：

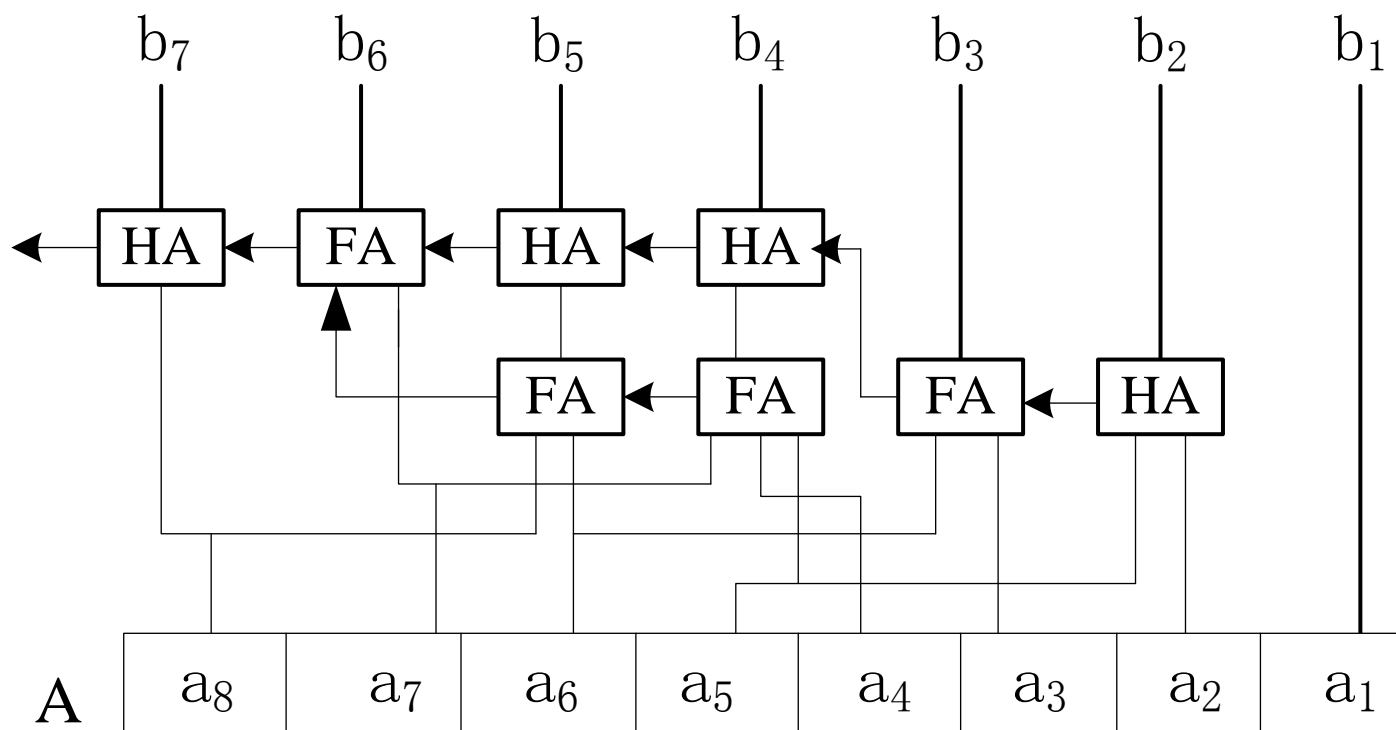
$$\begin{array}{r} a_8 \ a_7 \ a_6 \ a_5 \ 0 \ 0 \ 0 \\ 0 \ 0 \ a_8 \ a_7 \ a_6 \ a_5 \ 0 \\ + \ 0 \ 0 \ 0 \ a_4 \ a_3 \ a_2 \ a_1 \\ \hline b_7 \ b_6 \ b_5 \ b_4 \ b_3 \ b_2 \ b_1 \end{array}$$

□ 该竖式对应的加法网络电路图如下页。

第六章 6.24



□ 两位8421码—二进制整数转换加法网络电路图



第六章 6.25



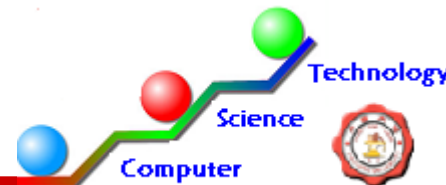
□ 6.25 设机器数字长为8位（含1位符号位），对下列各机器数算术左移一位、两位，算术右移一位、两位，并讨论结果是否正确。

$[x_1]_{\text{原}} = 0.001\ 1010$; $[x_2]_{\text{原}} = 1.110\ 1000$; $[x_3]_{\text{原}} = 1.001\ 1001$

$[y_1]_{\text{补}} = 0.101\ 0100$; $[y_2]_{\text{补}} = 1.110\ 1000$; $[y_3]_{\text{补}} = 1.001\ 1001$

$[z_1]_{\text{反}} = 1.010\ 1111$; $[z_2]_{\text{反}} = 1.110\ 1000$; $[z_3]_{\text{反}} = 1.001\ 1001$

第六章 6.25



□ 解：算术左移一位：

$[x_1]_{\text{原}} = 0.011\ 0100$ ；正确

$[y_1]_{\text{补}} = 0.010\ 1000$ ；溢出（丢1）出错

$[z_1]_{\text{反}} = 1.101\ 1111$ ；溢出（丢0）出错

算术左移两位：

$[x_1]_{\text{原}} = 0.110\ 1000$ ；正确

$[y_1]_{\text{补}} = 0.101\ 0000$ ；溢出（丢10）出错

$[z_1]_{\text{反}} = 1.011\ 1111$ ；溢出（丢01）出错

算术右移一位：

$[x_1]_{\text{原}} = 0.000\ 1101$ ；正确

$[y_1]_{\text{补}} = 0.010\ 1010$ ；正确

$[z_1]_{\text{反}} = 1.101\ 0111$ ；正确

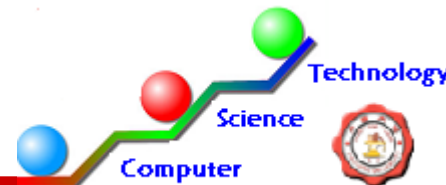
算术右移两位：

$[x_1]_{\text{原}} = 0.000\ 0110$ （10）；产生误差

$[y_1]_{\text{补}} = 0.001\ 0101$ ；正确

$[z_1]_{\text{反}} = 1.110\ 1011$ ；正确

第六章 6.25



□ 解：算术左移一位：

$[x_2]_{\text{原}} = 1.101\ 0000$ ；溢出（丢1）出错

$[y_2]_{\text{补}} = 1.101\ 0000$ ；正确

$[z_2]_{\text{反}} = 1.101\ 0001$ ；正确

算术左移两位：

$[x_2]_{\text{原}} = 1.010\ 0000$ ；溢出（丢11）出错

$[y_2]_{\text{补}} = 1.010\ 0000$ ；正确

$[z_2]_{\text{反}} = 1.010\ 0011$ ；正确

算术右移一位：

$[x_2]_{\text{原}} = 1.011\ 0100$ ；正确

$[y_2]_{\text{补}} = 1.111\ 0100$ ；正确

$[z_2]_{\text{反}} = 1.111\ 0100(0)$ ；产生误差

算术右移两位：

$[x_2]_{\text{原}} = 1.001\ 1010$ ；正确

$[y_2]_{\text{补}} = 1.111\ 1010$ ；正确

$[z_2]_{\text{反}} = 1.111\ 1010(00)$ ；产生误差

第六章 6.25



□ 解：算术左移一位：

$[x_3]_{\text{原}} = 1.011\ 0010$ ；正确

$[y_3]_{\text{补}} = 1.011\ 0010$ ；溢出（丢0）出错

$[z_3]_{\text{反}} = 1.011\ 0011$ ；溢出（丢0）出错

算术左移两位：

$[x_3]_{\text{原}} = 1.110\ 0100$ ；正确

$[y_3]_{\text{补}} = 1.110\ 0100$ ；溢出（丢00）出错

$[z_3]_{\text{反}} = 1.110\ 0111$ ；溢出（丢00）出错

算术右移一位：

$[x_3]_{\text{原}} = 1.000\ 1100\ (1)$ ；产生误差

$[y_3]_{\text{补}} = 1.100\ 1100\ (1)$ ；产生误差

$[z_3]_{\text{反}} = 1.100\ 1100$ ；正确

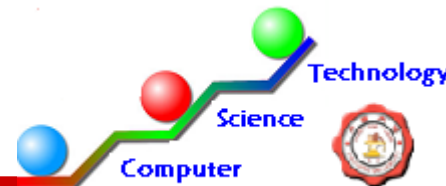
算术右移两位：

$[x_3]_{\text{原}} = 1.000\ 0110\ (01)$ ；产生误差

$[y_3]_{\text{补}} = 1.110\ 0110\ (01)$ ；产生误差

$[z_3]_{\text{反}} = 1.110\ 0110\ (01)$ ；产生误差

第六章 6.26

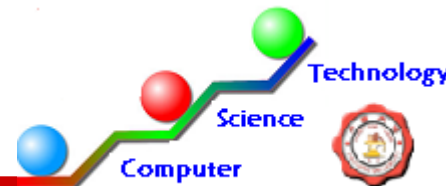


□ 6.26 设带符号数 $[Y]_{\text{原}}=[Y]_{\text{反}}=[Y]_{\text{补}}=1, 011\ 0010$ ，分别对这个8位字长的机器数进行算术左移一位、二位，算术右移一位、二位，逻辑左移一位、二位，逻辑右移一位、二位的操作，比较两种移位运算的区别，并分析结果的真值变化、误差及溢出情况。

□ 解：机器数移位结果如下：

注：表中“误差*”表示误差的绝对值。

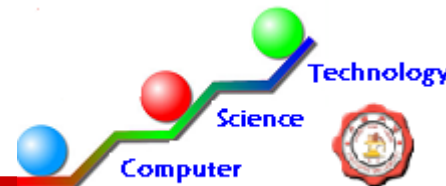
第六章 6.26



[Y]_原=1, 011 0010移位结果及分析

移位操作	溢出值	符号位	[Y] _原	丢掉值	十进制真值	结果分析
移位前		1	011 0010		-50	
逻辑右移1位		0	101 1001	0	+89	符号位移入MSB位引起错误, 符号位破坏
算术右移1位		1	001 1001	0	-25	无误差, 正确
逻辑右移2位		0	010 1100	10	+44	符号位移入数值位出错 符号位破坏
算术右移2位		1	000 1100	10	-12	误差*=1/2, 基本正确
逻辑左移1位	0	0	110 0100		+100	溢出出错, 符号位破坏
算术左移1位	0	1	110 0100		-100	无溢出, 正确
逻辑左移2位	1	1	100 1000		-72	溢出出错, 正确值=-200
算术左移2位	1	1	100 1000		-72	溢出出错, 正确值=-200

第六章 6.26



[Y]_反=1, 011 0010移位结果及分析

移位操作	溢出值	符号位	[Y] _反	丢掉值	十进制真值	结果分析
移位前		1	011 0010		-77	
逻辑右移1位		0	101 1001	0	+89	符号位移入MSB位引起错误, 符号位破坏
算术右移1位		1	101 1001	0	-38	误差*=1/2, 基本正确
逻辑右移2位		0	010 1100	10	+44	符号位移入数值位出错 符号位破坏
算术右移2位		1	110 1100	10	-19	无误差, 正确
逻辑左移1位	0	0	110 0100		+100	溢出出错, 符号位破坏
算术左移1位	0	1	110 0100		-27	溢出出错, 正确值=-155
逻辑左移2位	01	1	100 1000		-55	溢出出错, 正确值=-311
算术左移2位	01	1	100 1000		-55	溢出出错, 正确值=-311

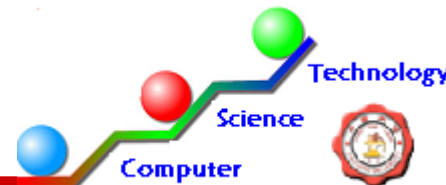
第六章 6.26



$[Y]_{\text{补}}=1, 011\ 0010$ 移位结果及分析

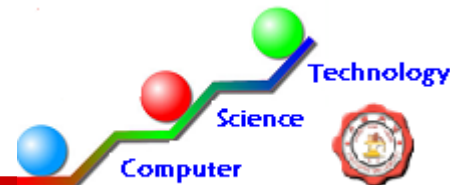
移位操作	溢出值	符号位	$[Y]_{\text{补}}$	丢掉值	十进制真值	结果分析
移位前		1	011 0010		-78	
逻辑右移1位		0	101 1001	0	+89	符号位移入MSB位引起错误, 符号位破坏
算术右移1位		1	101 1001	0	-39	无误差, 正确
逻辑右移2位		0	010 1100	10	+44	符号位移入数值位出错 符号位破坏
算术右移2位		1	110 1100	10	-20	误差*=1/2, 基本正确
逻辑左移1位	0	0	110 0100		+100	溢出出错, 符号位破坏
算术左移1位	0	1	110 0100		-28	溢出出错, 正确值=-156
逻辑左移2位	01	1	100 1000		-56	溢出出错, 正确值=-312
算术左移2位	01	1	100 1000		-56	溢出出错, 正确值=-312

第六章 6.27



- 6.27 设 $X_1=0.01\ 1100\ 0010$, $Y_1=-0.01\ 1100\ 0010$;
 $X_2=0.01\ 1100\ 1100$, $Y_2=-0.01\ 1100\ 1100$;
 $X_3=0.01\ 1100\ 0101$; $Y_3=-0.01\ 1100\ 0101$
- 分别用原码和补码表示, 如果只要求8位字长, 请采用截断法、恒置1法和0舍1入法对每一个操作数进行舍入, 并对舍入结果进行比较。
- 解: 先将真值 $X_1\sim X_3$ 、 $Y_1\sim Y_3$ 表示成机器码形式, 再进行舍入。为方便比较, 舍入结果用表格列出。**注意**相同下标的 X_i 、 Y_i 互为相反数, LSB^* 则表示误差方向是相对于最低有效位 LSB 的绝对值而言, 正误差使绝对值增大, 负误差使绝对值缩小。

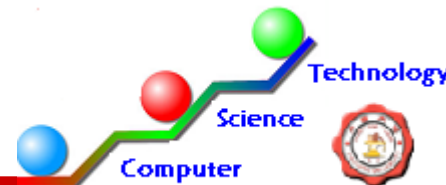
第六章 6.27



不同舍入方法的比较

舍入前 (11位)	舍入后 (8位)	丢掉位	结果真值	误差分析
$[X_1]_{\text{原}}=[X_1]_{\text{补}}=X_1$ =0.011 1000 010	截断=0舍1入 =0.011 1000 (舍) 恒置1=0.011 1001 (入)	010	0.011 1 0.011 1001	-1/4LSB* +3/4LSB*
$[Y_1]_{\text{原}}=1.011 1000 010$	截断=0舍1入 =1.011 1000 (舍) 恒置1=1.011 1001 (入)	010	-0.011 1 -0.011 1001	-1/4LSB* +3/4LSB*
$[Y_1]_{\text{补}}=1.100 0111 110$	截断=恒置1 =1.100 0111 (入) 0舍1入=1.1001000 (舍)	110	-0.011 1001 -0.011 1	+3/4LSB* -1/4LSB*
$[X_2]_{\text{原}}=[X_2]_{\text{补}}=X_2$ =0.011 1001 100	截断=恒置1 =0.011 1001 (舍) 0舍1入=0.011 1010 (入)	100	0.011 1001 0.011 101	-1/2LSB* +1/2LSB*
$[Y_2]_{\text{原}}=1.011 1001 100$	截断=恒置1 =1.011 1001 (舍) 0舍1入=1.011 1010 (入)	100	-0.011 1001 -0.011 101	-1/2LSB* +1/2LSB*
$[Y_2]_{\text{补}}=1.100 0110 100$	截断=0舍1入 =1.100 0110 (入) 恒置1=1. 100 0111 (舍)	100	-0.011 101 -0.011 1001	+1/2LSB* -1/2LSB*
$[X_3]_{\text{原}}=[X_3]_{\text{补}}=X_3$ =0.011 1000 101	截断=0.011 1000 (舍) 恒置1=0舍1入 =0.011 1001 (入)	101	0.011 1 0.011 1001	-5/8LSB* +3/8LSB*
$[Y_3]_{\text{原}}=1.011 1000 101$	截断=1.011 1000 (舍) 恒置1=0舍1入 =1.011 1001 (入)	101	-0.011 1 -0.011 1001	-5/8LSB* +3/8LSB*
$[Y_3]_{\text{补}}=1.100 0111 011$	截断=恒置1=0舍1入 =1. 100 0111 (入)	011	-0.011 1001	+3/8LSB*

第六章 6.28



□ 6.28 设机器数字长为8位（含1位符号位），用补码加减运算规则计算下列各题，并指出是否溢出。

(1) $X = -17/32$, $Y = 19/64$, 求 $X-Y$;

(2) $X = -21/32$, $Y = -67/128$, 求 $X+Y$;

(3) $X = 97$, $Y = -54$, 求 $X-Y$;

(4) $X = 118$, $Y = -36$, 求 $X+Y$ 。

第六章 6.28 (1)



□ 解:

$$(1) X = -17/32 = (-0.100\ 0100)_2$$

$$Y = 19/64 = (0.010\ 0110)_2$$

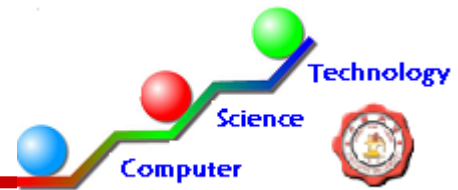
$$[X]_{\text{补}} = 1.011\ 1100$$

$$[Y]_{\text{补}} = 0.010\ 0110, [-Y]_{\text{补}} = 1.101\ 1010$$

$$\begin{array}{r} [X-Y]_{\text{补}} = 1.011\ 1100 \\ + 1.101\ 1010 \\ \hline 1.001\ 0110 \text{ ——无溢出} \end{array}$$

$$X-Y = (-0.110\ 1010)_2 = -53/64$$

第六章 6.28 (2)



□ 解:

$$(2) \quad X = -21/32 = (-0.101 \ 0100)_2$$

$$Y = -67/128 = (-0.100 \ 0011)_2$$

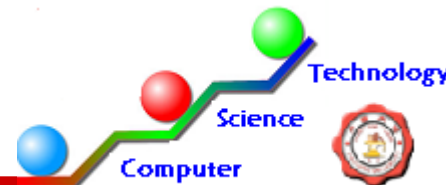
$$[X]_{\text{补}} = 1.010 \ 1100$$

$$[Y]_{\text{补}} = 1.011 \ 1101$$

$$\begin{array}{r} [X+Y]_{\text{补}} = 1.010 \ 1100 \\ + 1.011 \ 1101 \\ \hline 0.110 \ 1001 \text{ ——溢出} \end{array}$$

$$X+Y = (-1.001 \ 0111)_2 = -151/128$$

第六章 6.28 (3)



□ 解:

$$(3) \quad X = 97 = (110\ 0001)_2$$

$$Y = -54 = (-11\ 0110)_2$$

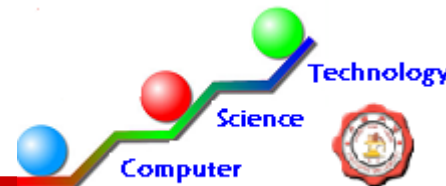
$$[X]_{\text{补}} = 0,110\ 0001$$

$$[Y]_{\text{补}} = 1,100\ 1010, \quad [-Y]_{\text{补}} = 0,011\ 0110$$

$$\begin{array}{r} [X-Y]_{\text{补}} = 0, \ 1\ 1\ 0 \quad 0\ 0\ 0\ 1 \\ + \quad 0, \ 0\ 1\ 1 \quad 0\ 1\ 1\ 0 \\ \hline 1, \ 0\ 0\ 1 \quad 0\ 1\ 1\ 1 \text{ —— 溢出} \end{array}$$

$$X-Y = (+1001\ 0111)_2 = 151$$

第六章 6.28 (4)



□ 解:

$$(4) X = 118 = (111\ 0110)_2$$

$$Y = -36 = (-10\ 0100)_2$$

$$[X]_{\text{补}} = 0, 111\ 0110$$

$$[Y]_{\text{补}} = 1, 101\ 1100$$

$$\begin{array}{r} [X+Y]_{\text{补}} = 0, \ 1\ 1\ 1\ 0\ 1\ 1\ 0 \\ + \quad 1, \ 1\ 0\ 1\ 1\ 1\ 0\ 0 \\ \hline 0, \ 1\ 0\ 1\ 0\ 0\ 1\ 0 \text{——无溢出} \end{array}$$

$$X+Y = (+101\ 0010)_2 = 82$$

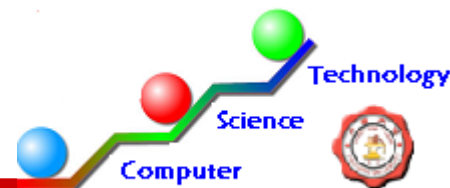
- 注意:
- ① 单符号位运算要用单符号位的判溢出方法;
 - ② 结果的真值形式上要和原始数据一致。

第六章 6.29



- 6.29 分别用8421码加法和余3码加法求 $57+48=?$ $316+258=?$ 要求列出竖式计算过程。
- 解： $(57)_{\text{BCD}} = 0101, 0111$; $(48)_{\text{BCD}} = 0100, 1000$
 $(57)_{\text{E}_3} = 1000, 1010$; $(48)_{\text{E}_3} = 0111, 1011$
 $(316)_{\text{BCD}} = 0011, 0001, 0110$; $(258)_{\text{BCD}} = 0010, 0101, 1000$
 $(316)_{\text{E}_3} = 0110, 0100, 1001$; $(258)_{\text{E}_3} = 0101, 1000, 1011$
- 为清楚起见，这里将各位BCD码之间用逗号隔开。
- 加法过程如下：

第六章 6.29



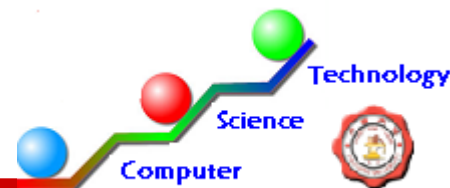
□ 57+48的8421码加法过程：

$$\begin{array}{rcl} & 0101, 0111 & \text{——} 57 \\ + & 0100, 1000 & \text{——} 48 \\ \hline & 1001, 1111 & \text{—— 低位无进位} \\ + & 0110 & \text{——} +6\text{修正} \\ \hline & 1010, 0101 & \text{—— 低位进到高位, 高位无进位} \\ + & 0110 & \text{—— 高位+6修正} \\ \hline 0001, 0000, 0101 & & \text{——} 105 \end{array}$$

□ 57+48的余3码加法过程：

$$\begin{array}{rcl} & 1000, 1010 & \text{——} 57 \\ + & 0111, 1011 & \text{——} 48 \\ \hline & 1, 0000, 0101 & \\ + & 0011, 0011, 0011 & \text{—— 有进位+3 (+0011), 无进位-3 (+1101)} \\ \hline 0100, 0011, 1000 & & \text{——} 105 \end{array}$$

第六章 6.29



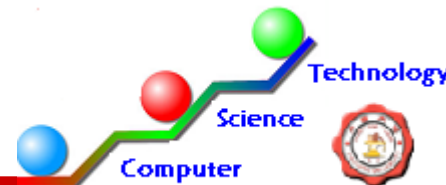
□ 316+258的8421码加法过程：

$$\begin{array}{r} 0011, 0001, 0110 \text{ —— } 316 \\ + 0010, 0101, 1000 \text{ —— } 258 \\ \hline 0101, 0110, 1110 \text{ —— 低位无进位} \\ + 0110 \text{ —— } +6 \text{修正} \\ \hline 0101, 0111, 0100 \text{ —— } 574 \end{array}$$

□ 316+258的余3码加法过程：

$$\begin{array}{r} 0110, 0100, 1001 \text{ —— } 316 \\ + 0101, 1000, 1011 \text{ —— } +258 \\ \hline 1011, 1101, 0100 \\ + 1101, 1101, 0011 \text{ —— 有进位+3 (+0011), 无进位-3 (+1101)} \\ \hline 1000, 1010, 0111 \text{ —— } 574 \end{array}$$

第六章 6.30



- 6.30 假定在一个 8 位字长的计算机中，定点整数用单字长表示，其中带符号整数用补码表示（符号占1位）；浮点数用双字长表示，阶码为8位移码（包括1位符号位），尾数用8位原码（包括1位符号位）。运行如下类 C 程序段：

```
int x1 = -124;
int x2 = 116;
unsigned int y1 = x1;
float f1 = x1;
int z1 = x1 + x2;
int z2 = x1 - x2;
```

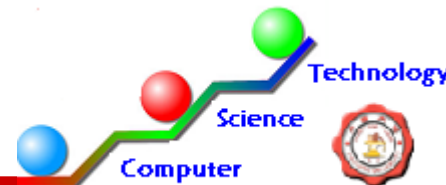
- 请问：
- （1）执行上述程序段后，所有变量的值在该计算机内的数据表示形式各是多少？所有变量的值对应的十进制形式各是多少？
- （2）在该计算机中，无符号整数、带符号整数和规格化浮点数的表示范围各是什么？（要求用十进制2的幂形式表示）
- （3）执行上述程序段后，哪些运算语句的执行结果发生了溢出？

第六章 6.30



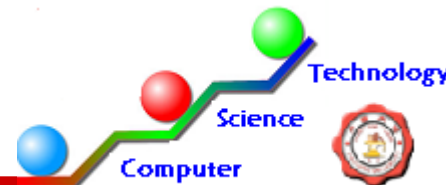
- ☐ 解：
- ☐ (1) 执行上述程序段后，变量
- ☐ x1值的十进制表示形式： -124
- ☐ x1值的机内表示形式： 1,000 0100
- ☐ x2值的十进制表示形式： 116
- ☐ x2值的机内表示形式： 0,111 0100
- ☐ y1 值的十进制表示形式： 132
- ☐ y1 值的机内表示形式： 1000 0100
- ☐ f1 值的十进制表示形式： -124.0
- ☐ f1 值的机内表示形式： 1,000 0111;1.111 1100
- ☐ z1值的十进制表示形式： -8
- ☐ z1值的机内表示形式： 1,111 1000
- ☐ z2值的十进制表示形式： 16
- ☐ z2值的机内表示形式： 0,001 0000

第六章 6.30



- (2) 无符号整数表示范围： $0 \sim 2^8 - 1$
- 带符号整数表示范围： $-2^7 \sim 2^7 - 1$
- 规格化浮点数表示范围：
 $-(1-2^{-7}) \times 2^{127} \sim -2^{-1} \times 2^{-128}, 0, 2^{-1} \times 2^{-128} \sim (1-2^{-7}) \times 2^{127}$
- (3) 执行上述程序段后，语句 `int z2 = x1-x2` 的执行结果发生了溢出。

第六章 6.31



□ 6.31 已知X和Y，用变形补码计算 $X \pm Y$ ，同时指出结果是否溢出。

- (1) $X=0.11011$; $Y=-0.10101$
- (2) $X=0.10111$; $Y=0.11011$
- (3) $X=-0.10110$; $Y=-0.00011$
- (4) $X=0.11011$; $Y=-0.11111$

□ 解：

□ 首先将X、Y转换为变形补码（模4补码），然后计算，最后判溢出。

$$(1) [X]_{\text{补}} = 00.11011; [Y]_{\text{补}} = 11.01011; [-Y]_{\text{补}} = 00.10101$$

$$[X+Y]_{\text{补}} = 00.11011 + 11.01011 = 00.00110$$

$$0 \oplus 0 = 0 \text{ —— 无溢出, } X+Y = 0.00110$$

$$[X-Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = 00.11011 + 00.10101 = 01.10000$$

$$0 \oplus 1 = 1 \text{ —— 溢出, } X-Y = +1.10000$$

第六章 6.31

- (2) $[X]_{\text{补}} = 00.10111$; $[Y]_{\text{补}} = 00.11011$; $[-Y]_{\text{补}} = 11.00101$
 $[X+Y]_{\text{补}} = 00.10111 + 00.11011 = 01.10010$
 $0 \oplus 1 = 1$ —— 溢出, $X+Y = +1.10010$
 $[X-Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = 00.10111 + 11.00101 = 11.11100$
 $1 \oplus 1 = 0$ —— 无溢出, $X-Y = -0.00100$
- (3) $[X]_{\text{补}} = 11.01010$; $[Y]_{\text{补}} = 11.11101$; $[-Y]_{\text{补}} = 00.00011$
 $[X+Y]_{\text{补}} = 11.01010 + 11.11101 = 11.00111$
 $1 \oplus 1 = 0$ —— 无溢出, $X+Y = -0.11001$
 $[X-Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = 11.01010 + 00.00011 = 11.01101$
 $1 \oplus 1 = 0$ —— 无溢出, $X-Y = -0.10011$
- (4) $[X]_{\text{补}} = 00.11011$; $[Y]_{\text{补}} = 11.00001$; $[-Y]_{\text{补}} = 00.11111$
 $[X+Y]_{\text{补}} = 00.11011 + 11.00001 = 11.11100$
 $1 \oplus 1 = 0$ —— 无溢出, $X+Y = -0.00100$
 $[X-Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = 00.11011 + 00.11111 = 01.11010$
 $0 \oplus 1 = 1$ —— 溢出, $X-Y = +1.11010$

注意: ① 补码运算有可能产生模溢出, 这是补码运算的正常情况;
 ② 结果要求真值表示时, 应将补码结果转换为真值。

第六章 6.32



□ 6.32 已知真值X, Y, 用移码运算求 $X \pm Y = ?$ 并判断溢出。

(1) $X=101101$; $Y=-110110$

(2) $X=-101101$; $Y=-110110$

(3) $X=-10001$; $Y=101011$

□ 解: 为了判溢出, 运算采用双符号位移码, 最高符号位初始为零。

(1) $[X]_{\text{移}}=01,101\ 101$; $[Y]_{\text{移}}=00,001\ 010$;

$[Y]_{\text{补}}=11,001\ 010$; $[-Y]_{\text{补}}=00,110\ 110$;

$[X+Y]_{\text{移}}=[X]_{\text{移}}+[Y]_{\text{补}}=01,101\ 101+11,001\ 010=00,110\ 111$

↓
无溢出

$[X-Y]_{\text{移}}=[X]_{\text{移}}+[-Y]_{\text{补}}=01,101\ 101+00,110\ 110=10,100\ 011$

↓
溢出

$X+Y=-001\ 001$; $X-Y=+1\ 100\ 011$

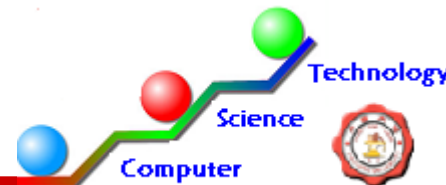
第六章 6.32



□ (2) $[X]_{\text{移}} = 00,010\ 011$; $[Y]_{\text{移}} = 00,001\ 010$
 $[Y]_{\text{补}} = 11,001\ 010$; $[-Y]_{\text{补}} = 00,110\ 110$
 $[X+Y]_{\text{移}} = [X]_{\text{移}} + [Y]_{\text{补}} = 00,010\ 011 + 11,001\ 010 = 11,011\ 101$ ——溢出
 $[X-Y]_{\text{移}} = [X]_{\text{移}} + [-Y]_{\text{补}} = 00,010\ 011 + 00,110\ 110 = 01,001\ 001$ ——无溢出
 $X+Y = -1\ 100\ 011$; $X-Y = +001\ 001$

□ (3) $[X]_{\text{移}} = 00,101\ 111$; $[Y]_{\text{移}} = 01,101\ 011$
 $[Y]_{\text{补}} = 00,101\ 011$; $[-Y]_{\text{补}} = 11,010\ 101$
 $[X+Y]_{\text{移}} = [X]_{\text{移}} + [Y]_{\text{补}} = 00,101\ 111 + 00,101\ 011 = 01,011\ 010$ ——无溢出
 $[X-Y]_{\text{移}} = [X]_{\text{移}} + [-Y]_{\text{补}} = 00,101\ 111 + 11,010\ 101 = 00,000\ 100$ ——无溢出
 $X+Y = +011\ 010$; $X-Y = -111\ 100$

第六章 6.33



□ 6.33 用原码一位乘法和补码一位乘比较法、两位乘比较法计算 $X \times Y$ 。

(1) $X = 0.110\ 111$, $Y = -0.101\ 110$;

(2) $X = -0.010\ 111$, $Y = -0.010\ 101$;

□ 解:

□ 先将数据转换成所需的机器数, 然后计算, 最后结果转换成真值。

(1) $[X]_{\text{原}} = X = 0.110111$, $[Y]_{\text{原}} = 1.101110$

$X^* = 0.110111$, $Y^* = 0.101110$

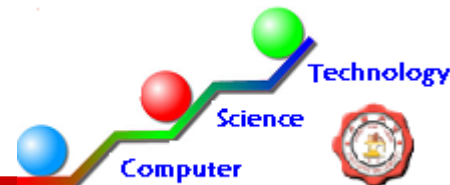
$X_0 = 0$, $Y_0 = 1$, $Z_0 = X_0 \oplus Y_0 = 0 \oplus 1 = 1$

$X^* \times Y^* = 0.100\ 111\ 100\ 010$

$[X \times Y]_{\text{原}} = 1.100\ 111\ 100\ 010$

$X \times Y = -0.100\ 111\ 100\ 010$

第六章 6.33 (1) 原码一位乘



部分积	乘数Y*
0.000 000	.1 0 1 1 1 0 ——— +0
→1 0.000 000	0 .1 0 1 1 1 ——— +X*
+ 0.110 111	
0.110 111	
→1 0.011 011	1 0 .1 0 1 1 ——— +X*
+ 0.110 111	
1.010 010	
→1 0.101 001	0 1 0 .1 0 1 ——— +X*
+ 0.110 111	
1.100 000	
→1 0.110 000	0 0 1 0 .1 0 ——— +0
→1 0.011 000	0 0 0 1 0 .1 ——— X*
+ 0.110 111	
1.001 111	
→1 0.100 111	1 0 0 0 1 0

第六章 6.33 (1) 补码乘



$$[X]_{\text{补}} = X = 0.110111$$

$$[Y]_{\text{补}} = 1.010010$$

$$[-X]_{\text{补}} = 1.001001$$

$$[2X]_{\text{补}} = 01.101110$$

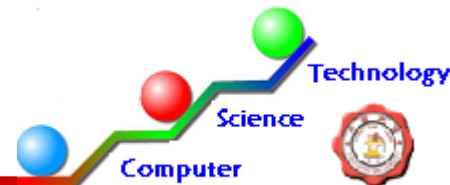
$$[-2X]_{\text{补}} = 10.010010$$

$$[X \times Y]_{\text{补}} = 1.011 \ 000 \ 011 \ 110 \ 0$$

$$X \times Y = -0.100 \ 111 \ 100 \ 010 \ 0$$

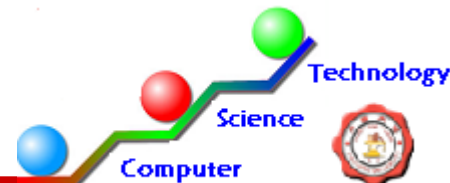
补码一位乘、两位乘运算过程如下：

第六章 6.33 (1) 补码一位乘



部分积	乘数[Y] _补	Y _n	Y _{n+1}	
00.000 000	1.0 1 0 0 1	0	0	—— +0
→1 00.000 000	0 1.0 1 0 0	1	0	
+ 11.001 001				+[-X] _补
11.001 001				
→1 11.100 100	1 0 1.0 1 0	0	1	
+ 00.110 111				+ [X] _补
00.011 011				
→1 00.001 101	1 1 0 1 .0 1	0	0	—— +0
→1 00.000 110	1 1 1 0 1.0	1	0	
+ 11.001 001				+ [-X] _补
11.001 111				
→1 11.100 111	1 1 1 1 0 1.0	1	1	
+ 00.110 111				+ [X] _补
00.011 110				
→1 00.001 111	0 1 1 1 1 0	1	0	1 .0
+ 11.001 001				+ [-X] _补
11.011 000	0 1 1 1 1 0	0		—— 清0

第六章 6.33 (1) 补码两位乘



部分积	乘数[Y] _补	Y _{n-1} Y _n Y _{n+1}
000.000 000	11.010 0	0 1 0 0
+ 110.010 010		+[-2X] _补
110.010 010		
→2 111.100 100	10 11.0	1 0 0 1
+ 000.110 111		+ [X] _补
000.011 011		
→2 000.000 110	11 10 1	1.0 1 0
+ 000.110 111		+ [X] _补
000.111 101		
→2 000.001 111	01 11 1 0	1 1 . 0
+ 111.001 001		+ [-X] _补
111.011 000	01 11 1 0	0 0 清0

结果同补码一位乘, $X \cdot Y = -0.100\ 111\ 100\ 010\ 00$

第六章 6.33 (2)



$$(2) \quad X = -0.010111, \quad Y = -0.010101$$

$$[X]_{\text{原}} = 1.010111, \quad [Y]_{\text{原}} = 1.010101$$

$$X^* = 0.010111, \quad Y^* = 0.010101$$

$$[-X^*]_{\text{补}} = 1.101001$$

$$X_0 = 1, \quad Y_0 = 1, \quad Z_0 = X_0 \oplus Y_0 = 1 \oplus 1 = 0$$

$$[X]_{\text{补}} = 1.101001, \quad [Y]_{\text{补}} = 1.101011$$

$$[-X]_{\text{补}} = 0.010111, \quad [2X]_{\text{补}} = 1.010010$$

$$[-2X]_{\text{补}} = 0.101110$$

$$X^* \times Y^* = 0.000 \ 111 \ 100 \ 011$$

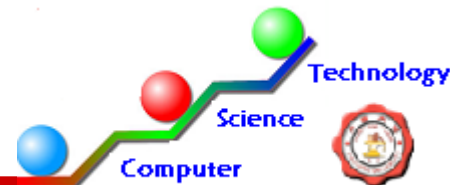
$$[X \times Y]_{\text{原}} = 0.000 \ 111 \ 100 \ 011$$

$$[X \times Y]_{\text{补}} = 0.000 \ 111 \ 100 \ 011 \ 0$$

$$X \times Y = 0.000 \ 111 \ 100 \ 011$$

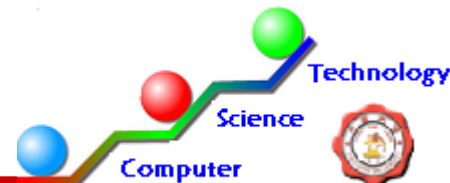
运算过程如下：

第六章 6.33 (2) 原码一位乘



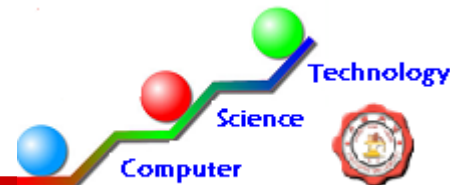
部分积	乘数Y*
0.000 000	.0 1 0 1 0 <u>1</u> ——— +X*
+ 0.010 111	
0.010 111	
→1 0.001 011	1 .0 1 0 1 <u>0</u> ——— +0
→1 0.000 101	1 1 .0 1 0 <u>1</u> ——— +X*
+ 0.010 111	
0.011 100	
→1 0.001 110	0 1 1 .0 1 <u>0</u> ——— +0
→1 0.000 111	0 0 1 1 .0 <u>1</u> ——— +X*
+ 0.010 111	
0.011 110	
→1 0.001 111	0 0 0 1 1 .0 ——— +0
→1 0.000 111	1 0 0 0 1 1

第六章 6.33 (2) 补码一位乘



部分积	乘数[Y] _补	Y _n	Y _{n+1}
00.000 000	1.1 0 1	0	1
+ 00.010 111		1	0
00.010 111		+[-X] _补	
→1 00.001 011	1 1.1 0	1	0
→1 00.000 101	1 1 1.1	0	1
+ 11.101 001		+[X] _补	
11.101 110			
→1 11.110 111	0 1 1	1	0
+ 00.010 111		+[-X] _补	
00.001 110			
→1 00.000 111	0 0 1	1	1
+ 11.101 001		+[X] _补	
11.110 000			
→1 11.111 000	0 0 0	1	1
+ 00.010 111		+[-X] _补	
00.001 111			
→1 00.000 111	1 0 0	0	1
		清0	

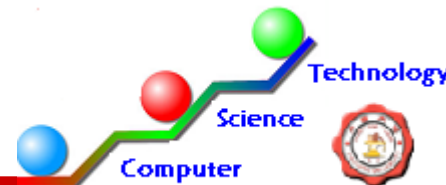
第六章 6.33 (2) 补码两位乘



部分积		乘数		Y_{n-1}	Y_n	Y_{n+1}
	0 0 0 . 0 0 0 0 0 0	1 1 . 1 0 1	0	1	1	0
+	0 0 0 . 0 1 0 1 1 1					$+[-X]_{补}$
	0 0 0 . 0 1 0 1 1 1					
→2	0 0 0 . 0 0 0 1 0 1	1 1	1 1 . 1	0	1 0	1
+	0 0 0 . 0 1 0 1 1 1					$+[-X]_{补}$
	0 0 0 . 0 1 1 1 0 0					
→2	0 0 0 . 0 0 0 1 1 1	0 0	1 1 1	1 . 1	0	1
+	0 0 0 . 0 1 0 1 1 1					$+[-X]_{补}$
	0 0 0 . 0 1 1 1 1 0					
→2	0 0 0 . 0 0 0 1 1 1	1 0	0 0 1	1	1 1 . 1	
						$+0, 清0$

结果同补码一位乘, $X \times Y = 0.000\ 111\ 100\ 011\ 00$

第六章 6.34



□ 6.34 用原码加减交替除法和补码加减交替除法计算 $X \div Y$ 。

(1) $X = -0.10101$, $Y = 0.11011$;

(2) $X = 13/32$, $Y = -27/32$ 。

□ 解:

(1) $[X]_{\text{原}} = 1.101\ 01$, $X^* = 0.101\ 01$

$Y^* = [Y]_{\text{原}} = Y = 0.110\ 11$

$[-Y^*]_{\text{补}} = 1.001\ 01$

$Q_0 = X_0 \oplus Y_0 = 1 \oplus 0 = 1$

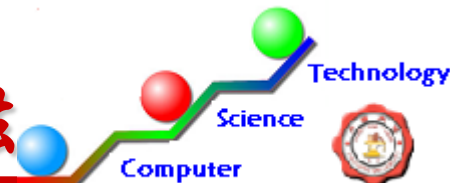
$X^* \div Y^* = 0.110\ 00$, $[X \div Y]_{\text{原}} = 1.110\ 00$

$X \div Y = -0.110\ 00$,

$R^* = 0.110\ 00 \times 2^{-5} = 0.000\ 001\ 100\ 0$

计算过程如下:

第六章 6.34 (1) 原码加减交替除法



被除数/余数	商
0.101 01	0.000 00
+ 1.001 01	试减, $+[-Y^*]_{\text{补}}$
1.110 10	
1← 1.101 00	0.
+ 0.110 11	$R < 0, +Y^*$
0.011 11	
1← 0.111 10	0.1
+ 1.001 01	$R > 0, +[-Y^*]_{\text{补}}$
0.000 11	
1← 0.001 10	0.11
+ 1.001 01	$R > 0, +[-Y^*]_{\text{补}}$
1.010 11	
1← 0.101 10	0.1 10
+ 0.110 11	$R < 0, +Y^*$
1.100 01	
1← 1.000 10	0.11 00
+ 0.110 11	$R < 0, +Y^*$
1.111 01	1← 0.110 00
+ 0.110 11	$R < 0, +Y^*$ (恢复余数)
0.110 00	

第六章 6.34 (1) 补码加减交替除法



$$X = -0.101\ 01, Y = 0.110\ 11$$

$$[X]_{\text{补}} = 1.010\ 11$$

$$[Y]_{\text{补}} = y = 0.110\ 11$$

$$[-Y]_{\text{补}} = 1.001\ 01$$

$$[X \div Y]_{\text{补}} = 1.001\ 11$$

$$X \div Y = -0.110\ 01$$

$$[R]_{\text{补}} = 1.010\ 00$$

$$R = -0.000\ 001\ 100\ 0$$

运算过程如下：

第六章 6.34 (1) 补码加减交替除法



被除数/余数 11.010 11	商 0.000 00
+ 00.110 11	试减, X、Y异号, +[Y] _补
00.001 10	
1← 00.011 00	1.
+ 11.001 01	R、Y同号, +[-Y] _补
11.100 01	
1← 11.000 10	1.0
+ 00.110 11	R、Y异号, +[Y] _补
11.111 01	
1← 11.110 10	1.00
+ 00.110 11	R、Y异号, +[Y] _补
00.101 01	
1← 01.010 10	1.001
+ 11.001 01	R、Y同号, +[-y] _补
00.011 11	
1← 00.111 10	1.0 011
+ 11.001 01	R、Y同号, +[-Y] _补
00.000 11	1← 1.0 0111 —— 恒置1
+ 11.001 01	R、X异号, 恢复余数
11.010 00	且R、Y同号, +[-Y] _补

注: 恒置1引入误差。

第六章 6.34(2)



□ $X = 13/32 = (0.011\ 01)_2$
 $Y = -27/32 = (-0.110\ 11)_2$
 $X^* = [X]_{\text{原}} = X = 0.011\ 01$
 $[Y]_{\text{原}} = 1.110\ 11$
 $Y^* = 0.110\ 11$
 $[-Y^*]_{\text{补}} = 1.001\ 01$
 $Q_0 = X_0 \oplus Y_0 = 0 \oplus 1 = 1$
 $X^* \div Y^* = 0.011\ 11$
 $[X \div Y]_{\text{原}} = 1.011\ 11$
 $X \div Y = (-0.011\ 11)_2 = -15/32$
 $R^* = 0.010\ 11 \times 2^{-5} = 0.000\ 000\ 101\ 1$

运算过程如下：

第六章 6.34 (2) 原码加减交替除法



被除数/余数 0.011 01	商 0.000 00
+ 1.001 01	试减, $+[-Y^*]_{\text{补}}$
1.100 10	
1← 1.001 00	0.
+ 0.110 11	$R < 0, +Y^*$
1.111 11	
1← 1.111 10	0.0
+ 0.110 11	$R < 0, +Y^*$
0.110 01	
1← 1.100 10	0.0 1
+ 1.001 01	$R > 0, +[-Y^*]_{\text{补}}$
0.101 11	
1← 1.011 10	0.0 1 1
+ 1.001 01	$R > 0, +[-Y^*]_{\text{补}}$
0.100 11	
1← 1.001 10	0.0 1 1 1
+ 1.001 01	$R > 0, +[-Y^*]_{\text{补}}$
0.010 11	1← 0.0 1 1 1 1
	$R > 0, \text{结束}$

第六章 6.34 (2) 补码加减交替除法



$$X=13/32=(0.011\ 01)_2$$

$$Y= -27/32=(-0.110\ 11)_2$$

$$[X]_{\text{补}} = x = 0.011\ 01$$

$$[Y]_{\text{补}} = 1.001\ 01$$

$$[-Y]_{\text{补}} = 0.110\ 11$$

$$[X \div Y]_{\text{补}} = 1.100\ 01$$

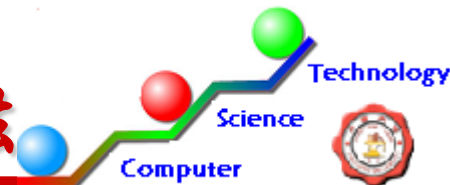
$$X \div Y = (-0.011\ 11)_2 = -15/32$$

$$[R]_{\text{补}} = 0.010\ 11$$

$$R = R^* = 0.000\ 000\ 101\ 1$$

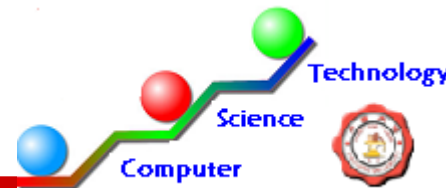
运算过程如下：

第六章 6.34 (2) 补码加减交替除法



被除数 (余数)	商
00.011 01	0.000 00
+ 11.001 01	试减, X、Y异号, +[Y] _补
11.100 10	
1← 11.001 00	1.
+ 00.110 11	R、Y同号, +[-Y] _补
11.111 11	
1← 11.111 10	1.1
+ 00.110 11	R、Y同号, +[-Y] _补
00.110 01	
1← 01.100 10	1.10
+ 11.001 01	R、Y异号, +[Y] _补
00.101 11	
1← 01.011 10	1.100
+ 11.001 01	R、Y异号, +[Y] _补
00.100 11	
1← 01.001 10	1.1 000
+ 11.001 01	R、Y异号, +[Y] _补
00.010 11	1← 1.1 000 1 —— 恒置1
	R、X同号, 结束

第六章 6.35



□ 6.35 设机器字长为16位（含1位符号位），若一次移位需 $1\mu\text{s}$ ，一次加法需 $1\mu\text{s}$ ，试问原码一位乘法、补码一位乘法、原码加减交替除法和补码加减交替除法最多各需多少时间？

□ 解：

原码一位乘最多需时= $1\mu\text{s} \times 15(\text{加}) + 1\mu\text{s} \times 15(\text{移位}) = 30\mu\text{s}$

补码一位乘最多需时= $1\mu\text{s} \times 16 + 1\mu\text{s} \times 15 = 31\mu\text{s}$

原码加减交替除最多需时= $1\mu\text{s} \times (16+1) + 1\mu\text{s} \times 15 = 32\mu\text{s}$

补码加减交替除最多需时= $1\mu\text{s} \times (16+1) + 1\mu\text{s} \times 15 = 32\mu\text{s}$

第六章 6.36



□ 有下列16位字长的逻辑数（八进制表示）：

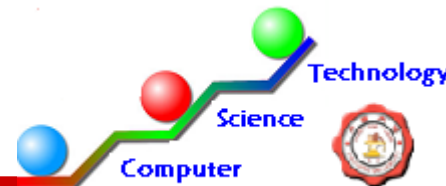
$A = 000\ 377$; $B = 123\ 456$; $C = 054\ 321$ 。

试计算： $X_1 = (B \oplus C) \cdot A$;
 $X_2 = \neg(\neg B \cdot \neg C) + A$;
 $X_3 = (A \oplus B) + \neg(A \cdot C)$;
 $X_4 = (A \oplus B) \oplus C$;

□ 解：

$X_1 = 000\ 377$;
 $X_2 = 177\ 777$;
 $X_3 = 177\ 777$;
 $X_4 = 177\ 400$

第六章 6.37



□ 6.37 设4位二进制加法器进位信号为 $C_4C_3C_2C_1$ ，最低位进位输入为 C_0 ；输入数据为 $A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$ ；进位生成函数为 $g_3g_2g_1g_0$ ，进位传递函数为 $p_3p_2p_1p_0$ ；请分别按下述两种方式写出 $C_4C_3C_2C_1$ 的逻辑表达式：

(1) 串行进位方式；(2) 并行进位方式。

□ 解：

令 $g_i = A_i B_i$ ； $p_i = A_i \oplus B_i$ ； ($i=0, 1, 2, 3$) 则

(1) 串行进位方式： (2) 并行进位方式：

$$C_1 = g_0 + p_0 C_0;$$

$$C_1 = g_0 + p_0 C_0$$

$$C_2 = g_1 + p_1 C_1;$$

$$C_2 = g_1 + p_1 g_0 + p_1 p_0 C_0$$

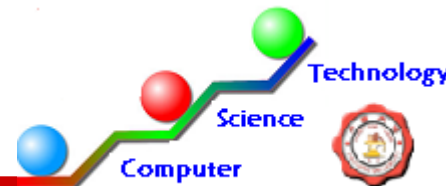
$$C_3 = g_2 + p_2 C_2;$$

$$C_3 = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 C_0$$

$$C_4 = g_3 + p_3 C_3;$$

$$C_4 = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 C_0$$

第六章 6.38



□ 6.38 设机器字长为32位，用与非门和与或非门设计一个并行加法器（假设与非门的延迟时间为10ns，与或非门的延迟时间为15ns），要求完成32位加法时间不得超过0.2μs。画出进位线路逻辑框图及加法器逻辑框图。

□ 解：首先根据题意要求选择进位方案：

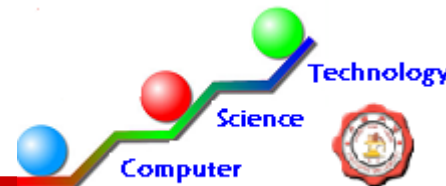
1) 若采用串行进位链（行波进位），则在 g_i 、 p_i 函数的基础上，实现32位进位需要的时间为：

$$T = 2t_y \times 32 = 64t_y = 64 \times 10 = 640\text{ns} = 0.64\mu\text{s}$$

不满足0.2μs的加法时间限制，不能用。

（设 $1t_y = 10\text{ns}$ ）

第六章 6.38



2) 若采用成组先行-级联进位方式，则在 g_i 、 p_i 的基础上，4位一组分组，小组内部先行进位，小组间串行进位，则32位进位需：

$$T = 2.5t_y \times 8 \text{组} = 20t_y = 20 \times 10 = 200\text{ns} = \mathbf{0.2\mu s}$$

刚好满足 $\mathbf{0.2\mu s}$ 加法时间的限制。

考虑到一次加法除进位时间外，还需 g_i 、 p_i 函数的产生时间、和的产生时间（最高位和）等因素，故此进位方案仍不适用。

结论：若采用成组先行-级联进位，小组规模需在 $\mathbf{6位}$ 以上较为合适。即：

$$T = 2.5t_y \times 6 \text{组} = 15t_y = 15 \times 10 = 150\text{ns} = \mathbf{0.15\mu s}$$

除进位外还有 $\mathbf{50ns}$ （约 $5t_y$ ）左右的时间供加法开销，较充裕。

3) 若采用二级先行-级联进位方式，4位一小组，4小组为一大组分组。小组内部先行进位，小组间即大组内也先行进位，大组间串行进位，则32位进位需：

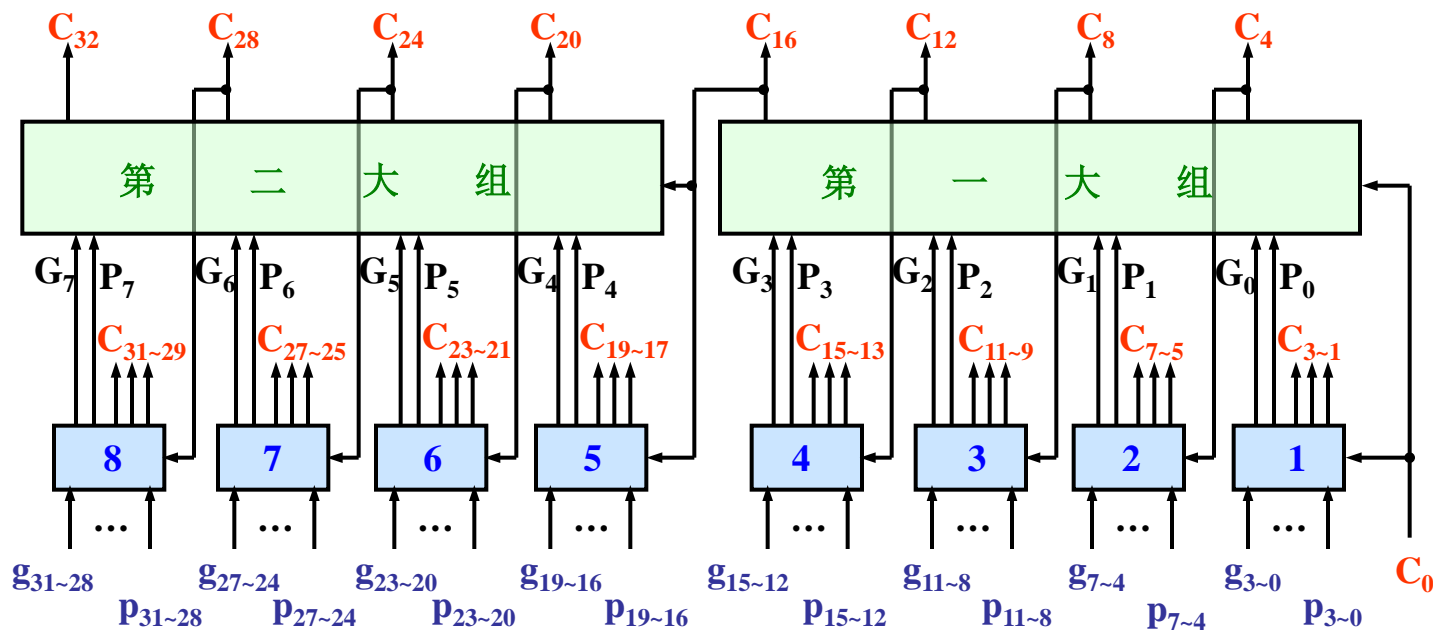
$$T=2.5t_y \times 4 \text{级} = 10t_y = 10 \times 10 = \mathbf{100ns}$$

完全满足 $0.2\mu s$ 的加法时间限制，可以使用。

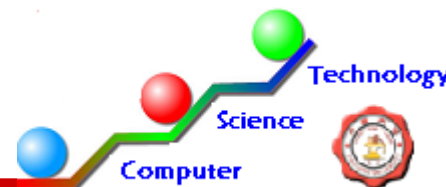
进位线路及加法器逻辑框图如下页。

第六章 6.38

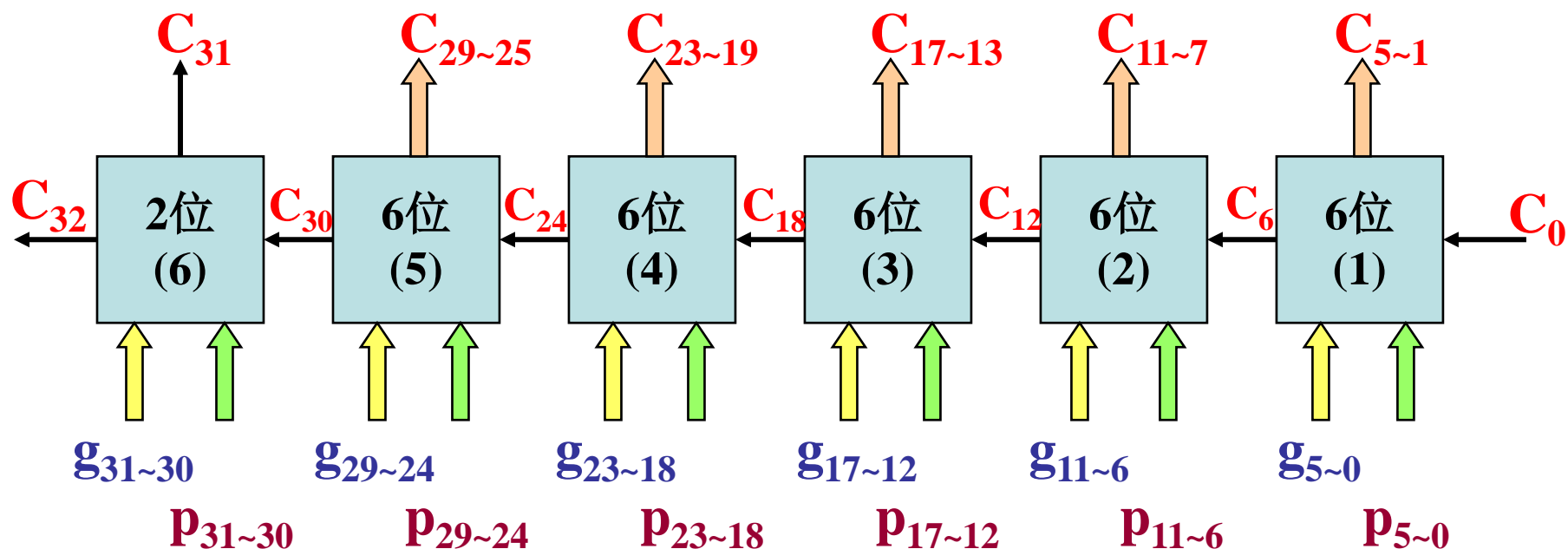
32位二级先行-级联进位线路



第六章 6.38

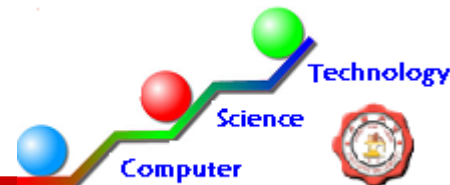


6位一组成组先行-级联进位链

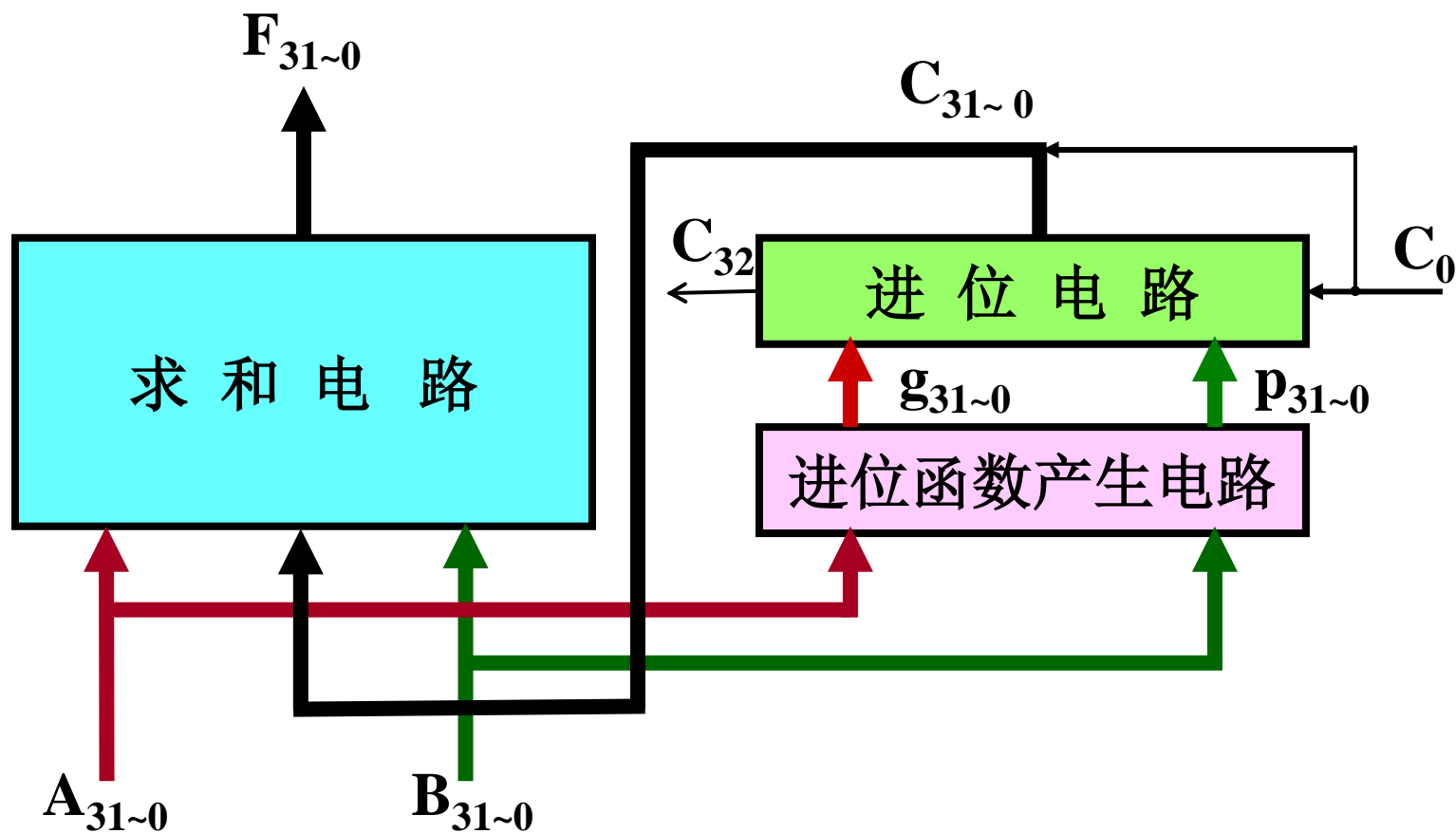


注：为讨论方便，上述电路忽略门电路扇入系数的影响。另外，一个完整的加法器还应考虑 g_i 、 p_i 产生电路、求和电路等。

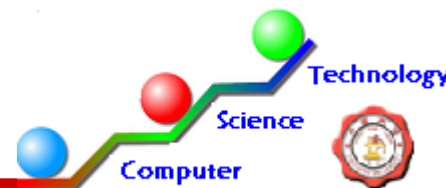
第六章 6.38



32位加法器逻辑框图：图中进位电路可选上述两种方案之一



第六章 6.39



□ 6.39 设机器字长为16位，分别按4、4、4、4和3、5、3、5分组后，

(1) 画出两种分组方案的成组先行-级联进位线路框图，并比较哪种方案运算速度快。

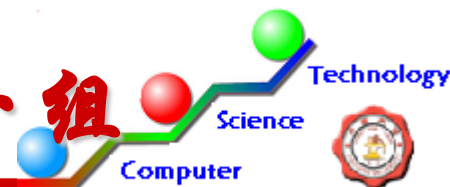
(2) 画出两种分组方案的二级先行进位线路框图，并对这两种方案的速度进行比较。

(3) 用74181和74182画出成组先行-级联进位和二级先行进位的并行进位线路框图。

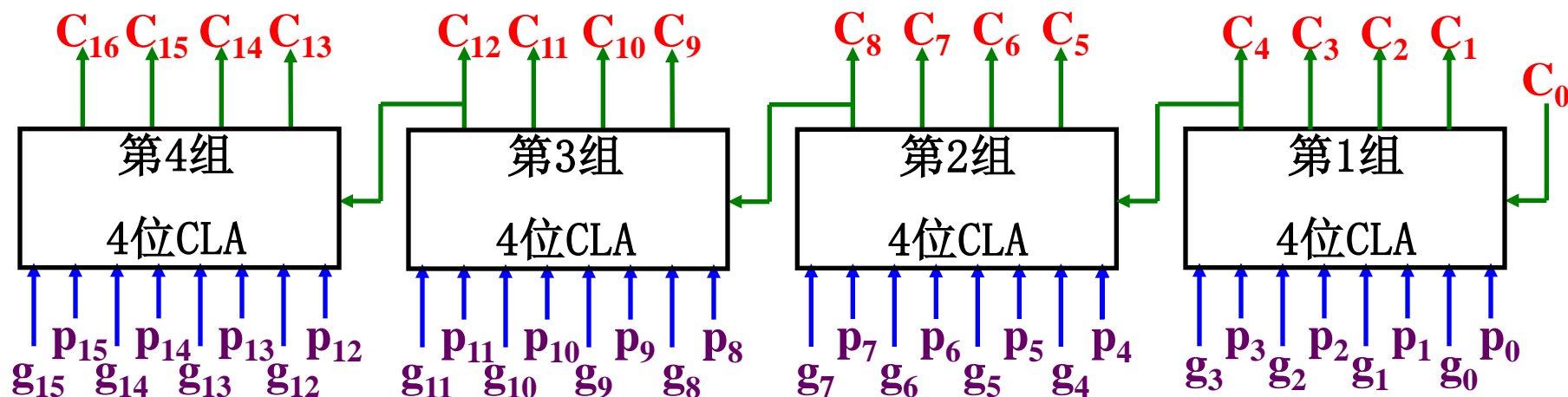
解：

(1) 4-4-4-4分组的16位成组先行-级联进位线路框图见下页。

第六章 6.39 (1) 4-4-4-4分组

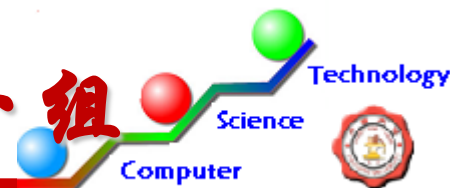


4-4-4-4分组16位成组先行-级联进位线路框图

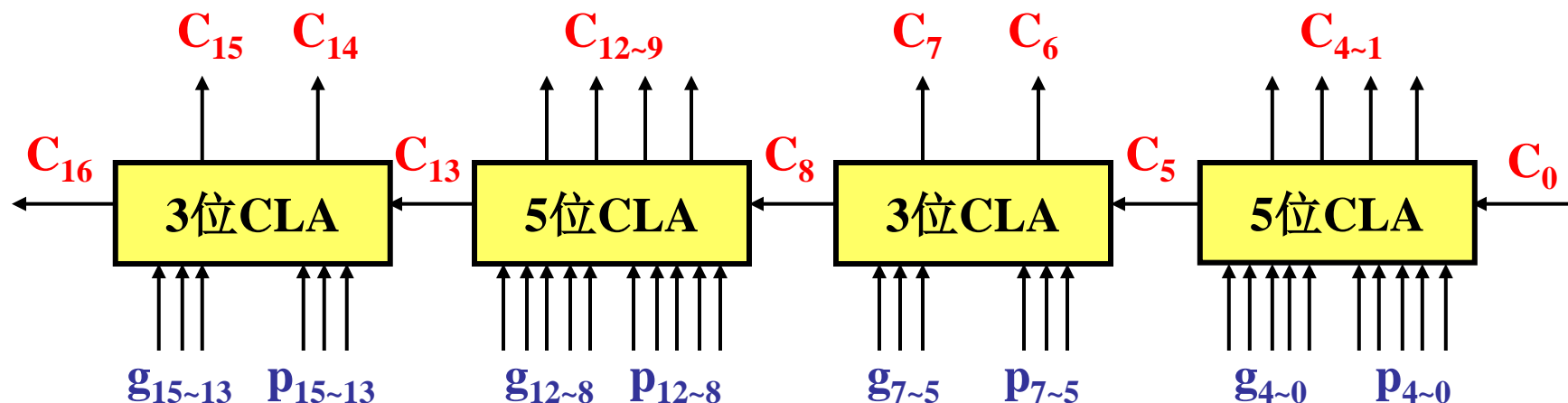


□ 4-4-4-4分组16位成组先行-级联进位线路最长进位延迟时间为： $2.5t_y \times 4 = 10t_y$

第六章 6.39 (1) 3-5-3-5分组



3-5-3-5分组16位成组先行-级联进位线路框图



□ 运算速度比较：4-4-4-4分组的进位时间= $2.5t_y \times 4 = 10t_y$;

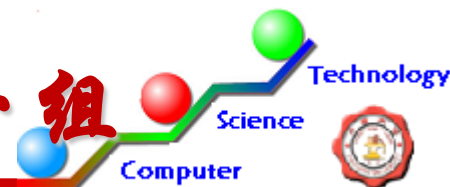
3-5-3-5分组的进位时间= $2.5t_y \times 4 = 10t_y$;

两种分组方案最长加法时间相同。

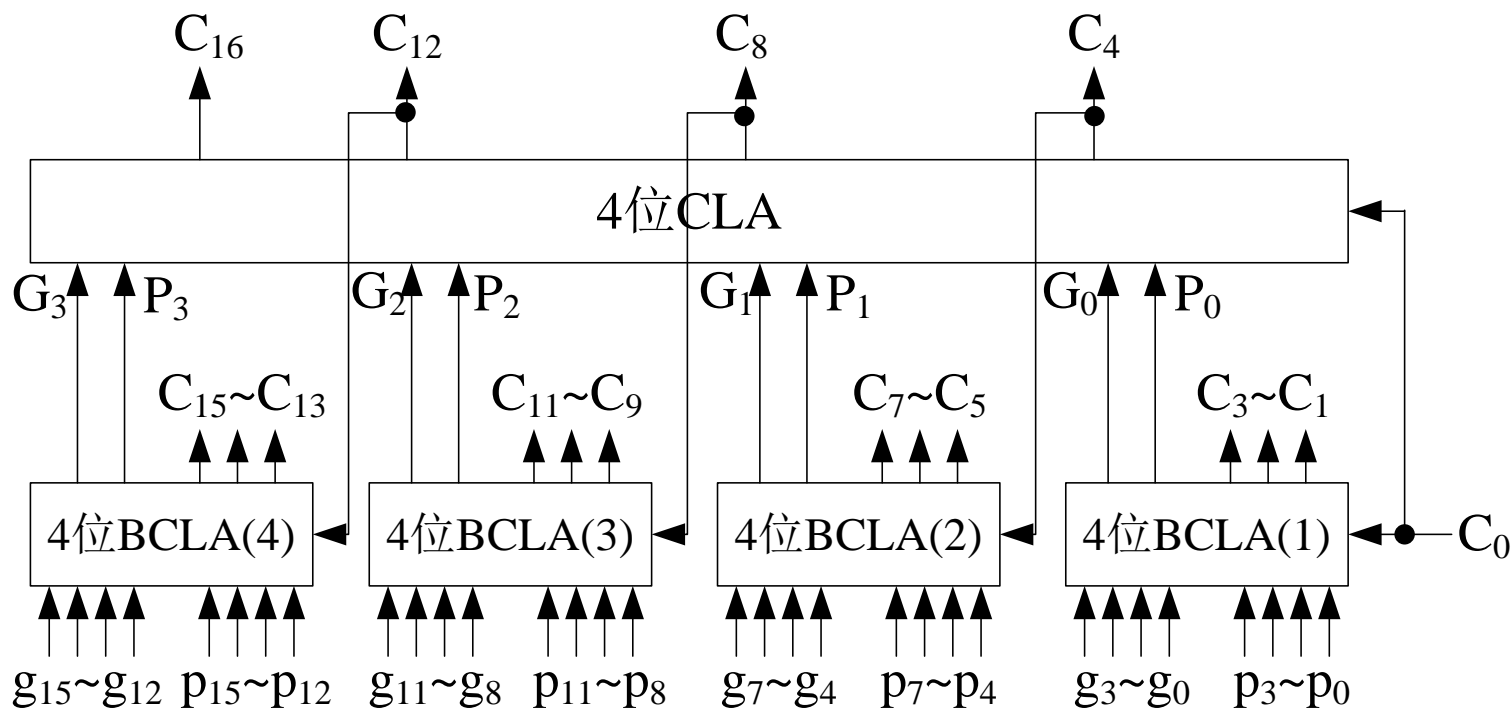
□ 结论：成组先行-级联进位的最长进位时间只与组数有关，与组内位数无关。

□ 注：为便于比较，3-5-3-5分组忽略扇入影响。

第六章 6.39 (2) 4-4-4-4分组

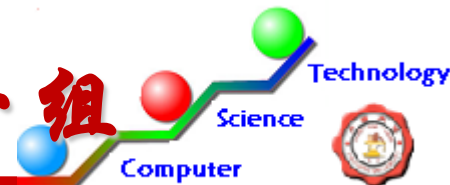


4-4-4-4分组16位二级先行进位线路框图

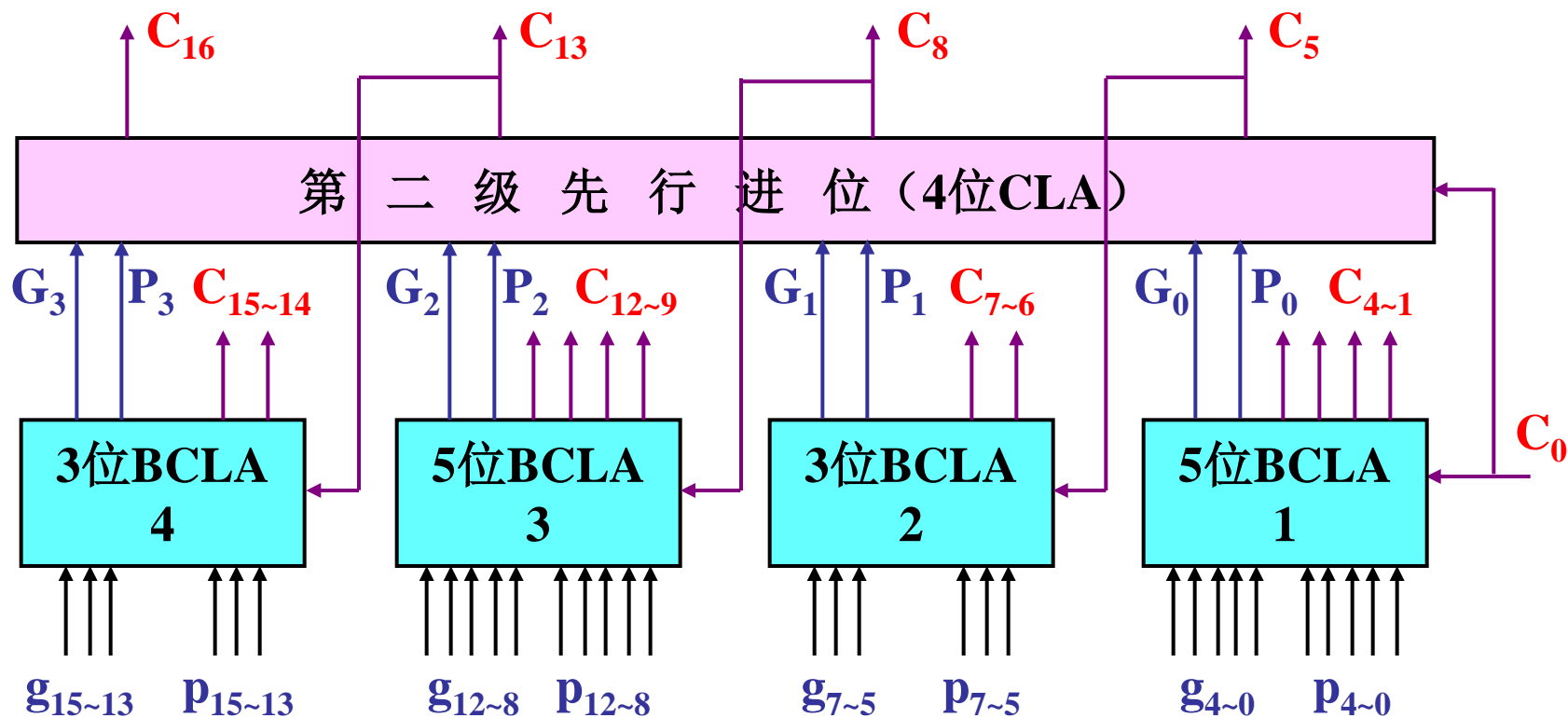


□ 4-4-4-4分组16位二级先行进位线路最长进位延迟时间为：
 $2.5t_y \times 3 = 7.5t_y$

第六章 6.39 (2) 3-5-3-5分组

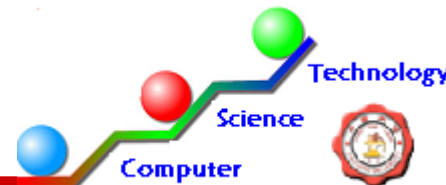


3-5-3-5分组16位二级先行进位线路框图



□ 3-5-3-5分组16位二级先行进位线路最长进位延迟时间为：
 $2.5t_y \times 3 = 7.5t_y$

第六章 6.39 (2)



□ 运算速度比较：

4-4-4-4分组的进位时间= $2.5t_y \times 3 = 7.5t_y$ ；

3-5-3-5分组的进位时间= $2.5t_y \times 3 = 7.5t_y$ ；

两种分组方案最长加法时间相同。

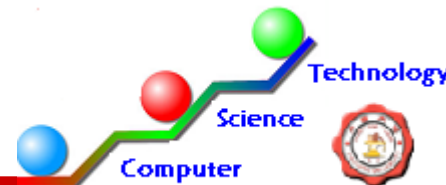
□ **结论：**二级先行进位的最长进位时间只与组数和级数有关，与组内位数无关。

Diagram illustrating a 4-stage ripple-carry adder structure using 74181 4-bit ALUs. The carry signal propagates from right to left, with delays labeled C_{16} , C_{12} , C_8 , C_4 , and C_0 .

Figure 1 illustrates the block diagram of a 16-bit parallel adder. The circuit consists of four 74181 4-bit ALUs and a 74182 4-bit BCLA (Binary Carry Look-Ahead) block. The ALUs are connected to the BCLA block via carry signals C_0, C_4, C_8, C_{12} . The BCLA block outputs carry signals $C_n, C_{n+x}, C_{n+y}, C_{n+z}$. The ALUs also output propagate signals P_0, P_1, P_2, P_3 and generate signals G_0, G_1, G_2, G_3 . The final carry-out is C_{16} .

Copyright ©2012 Computer Organization Group. All rights reserved.

第六章 6.39 注意



- ❑ 181芯片正、负逻辑引脚的表示方法;
- ❑ 为强调可比性, 3-5-3-5分组时不考虑扇入影响;
- ❑ 181芯片只有最高、最低两个进位输入/输出端, 组内进位无引脚;
- ❑ 181为4位片, 无法3-5-3-5分组, 只能4-4-4-4分组;
- ❑ 成组先行-级联进位只用到181, 使用182的一般是二级以上先行进位;
- ❑ 成组先行-级联进位是并行进位和串行进位技术的结合; 注意在位数较少时, 二级以上进位可以采用全先行进位技术实现; 位数较多时, 可采用二级并行进位和串行进位技术结合实现(二级先行-级联进位)。

第六章 6.40

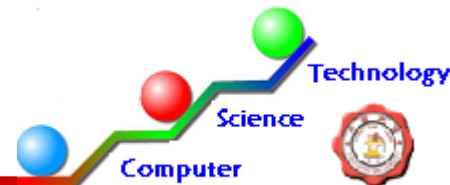


- 6.40 用预加6方案设计一位8421码加法单元，并以设计好的加法单元为模块，进一步设计一个4位的8421码加法器。

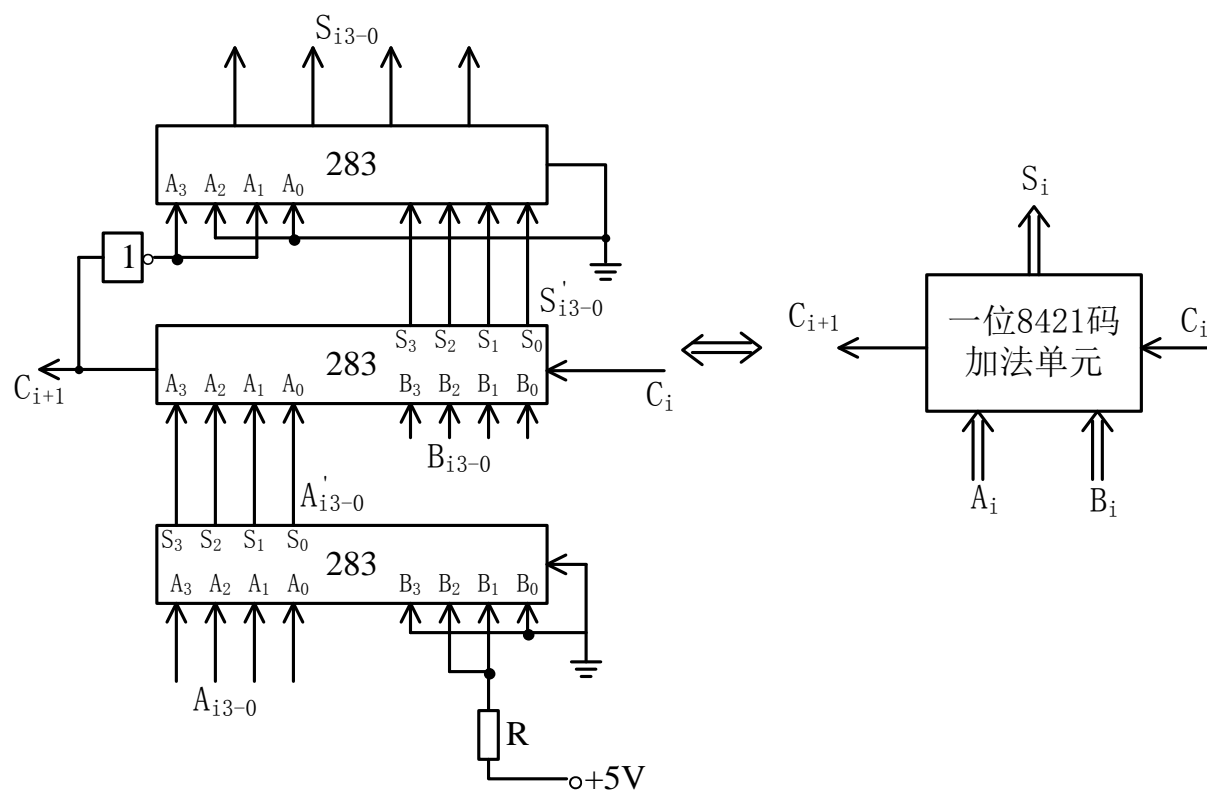
- 解：预加6方案一位8421码加法器算法分析：
 - ① 相加前，对其中一个加数预加6；
 - ② 做二进制加法；
 - ③ 十进制进位自动产生；
 - ④ 有进位时，不修正；
 - ⑤ 无进位时，减6修正（+1010）。

- 一位8421码加法单元线路见下页：（采用MSI芯片74LS283设计）

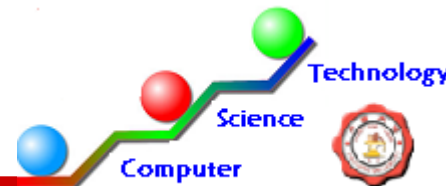
第六章 6.40



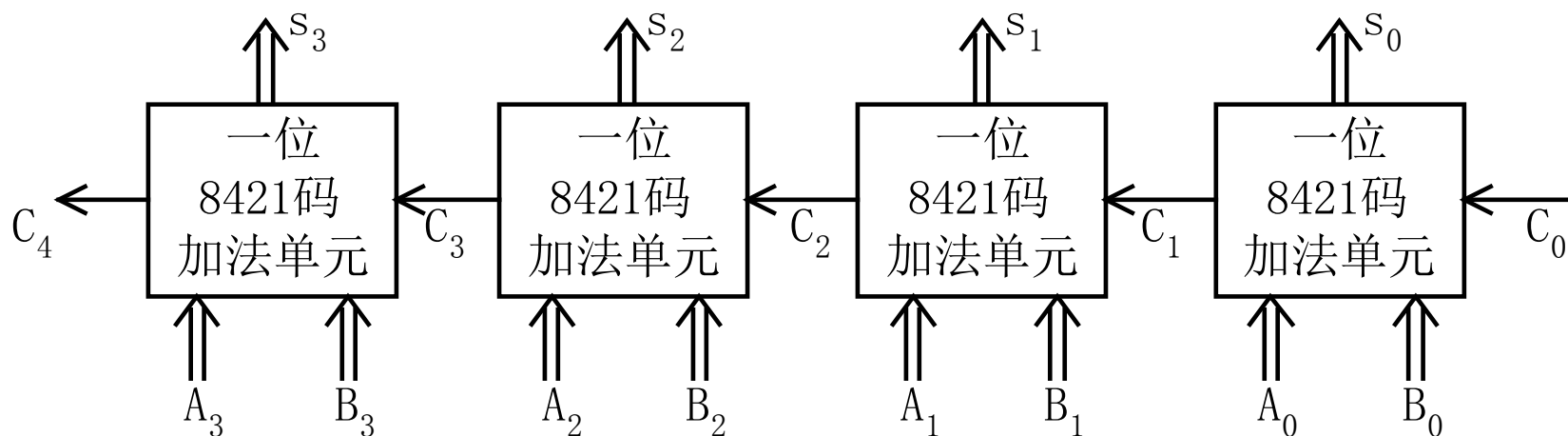
一位预加6方案8421码加法单元线路图



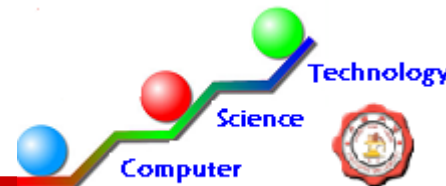
第六章 6.40



□ 用一位8421码加法器作基本加法单元，4位8421码加法器线路如下：



第六章 6.41



- 6.41 (1) 设计一个一位的余3码加法器，并分析其修正规律；
(2) 用8位并行二进制加法器实现2位余3码加法，试提出你的方案。

□ 解：

- (1) 由一位余3码加法和的修正关系表可得：

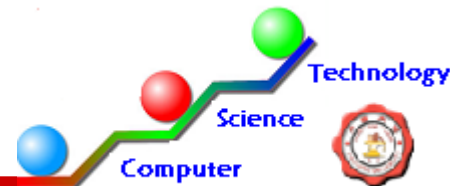
一位余3码加法修正规律：无进位，-3 (+1101) 修正；
有进位，+3 (+0011) 修正。

一位余3码加法器结构：

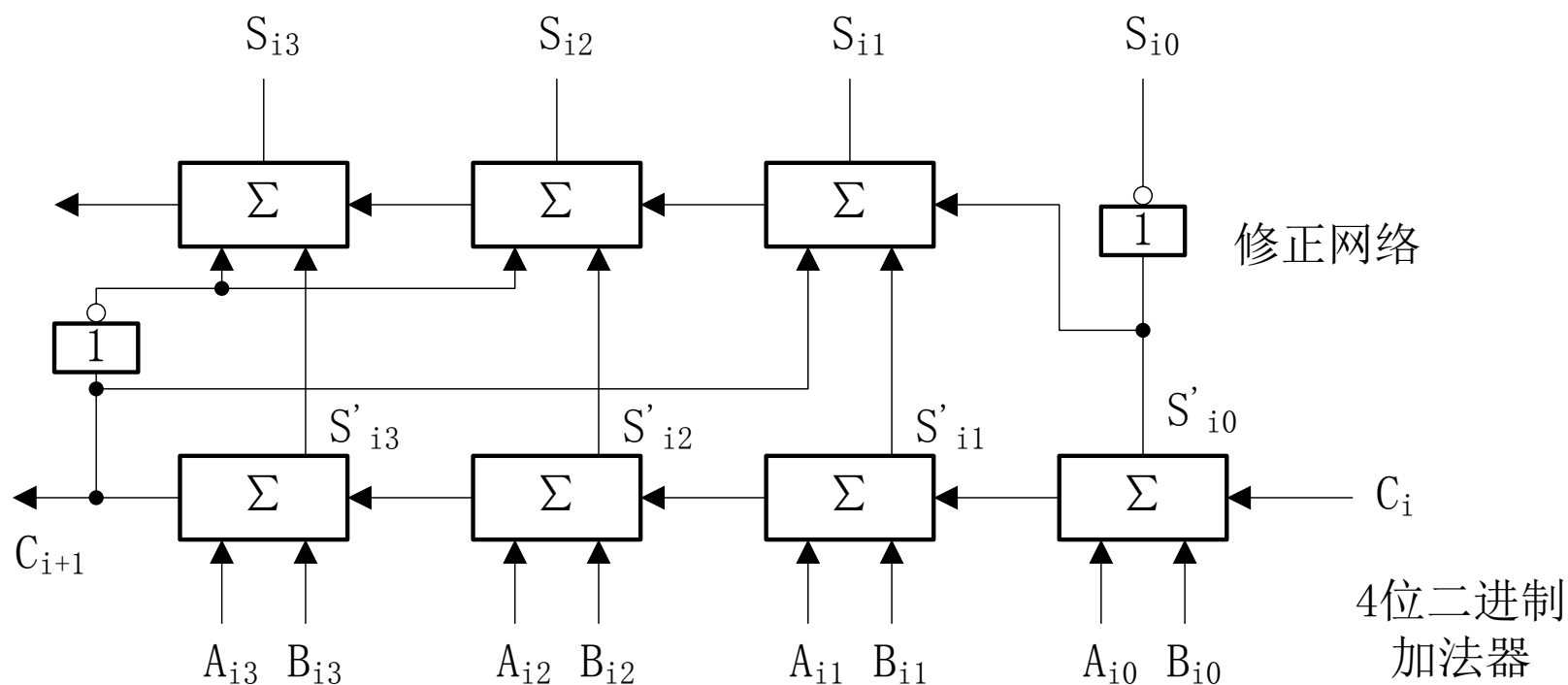
由两级4位二进制加法器组成，第一级常规的二进制加法器实现二进制加法，第二级简化的二进制加法器实现+3、-3修正，加减3的控制由第一级加法器的进位信号完成。线路实现既可采用SSI加法器件，也可采用MSI芯片。设计方案如下：

方案一：采用SSI全加器构成，线路见下页。

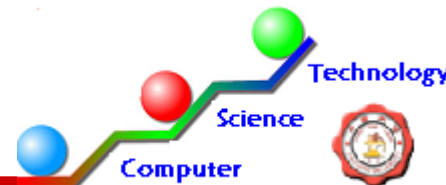
第六章 6.41 (1) SSI设计



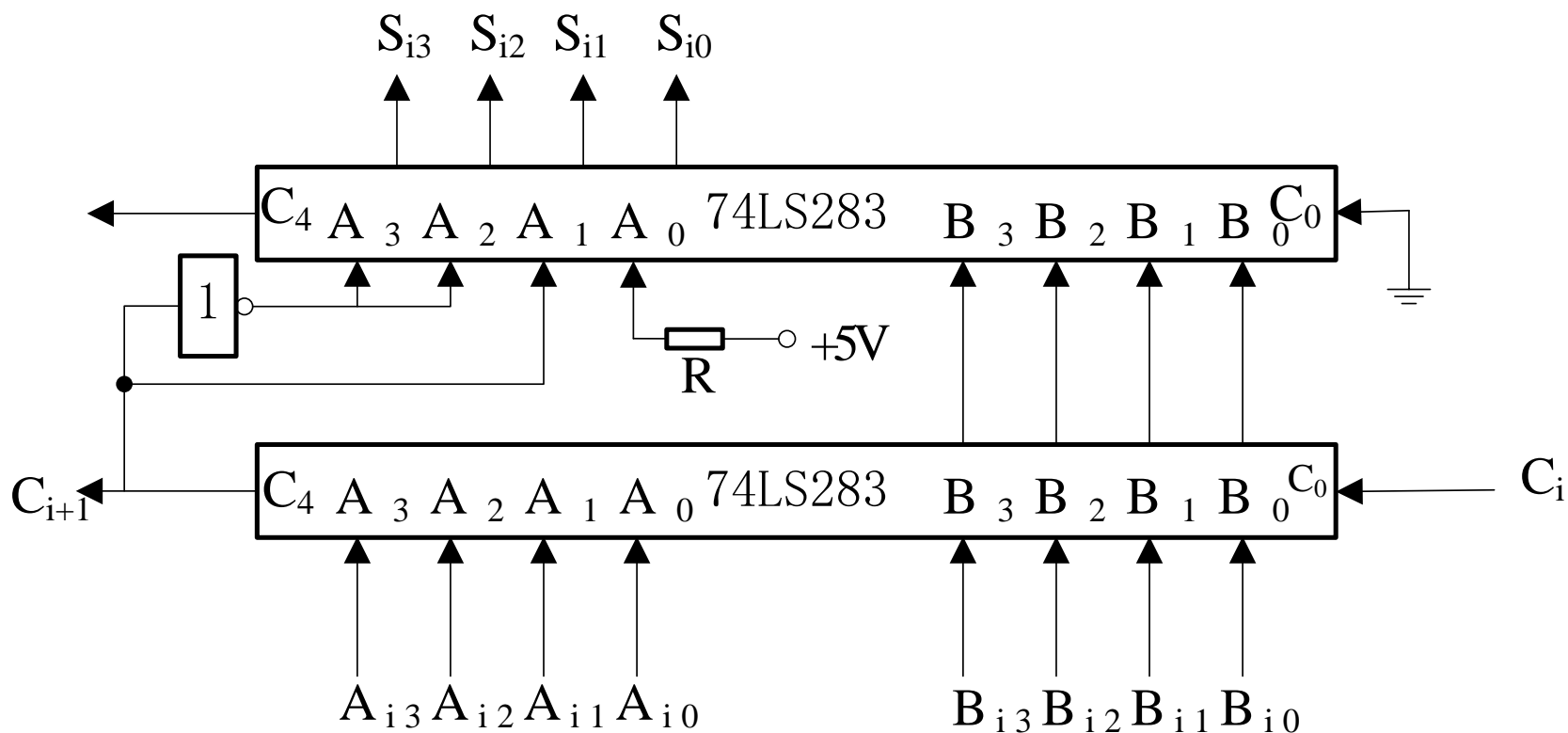
采用SSI全加器设计方案的线路图



第六章 6.41 (1) MSI设计



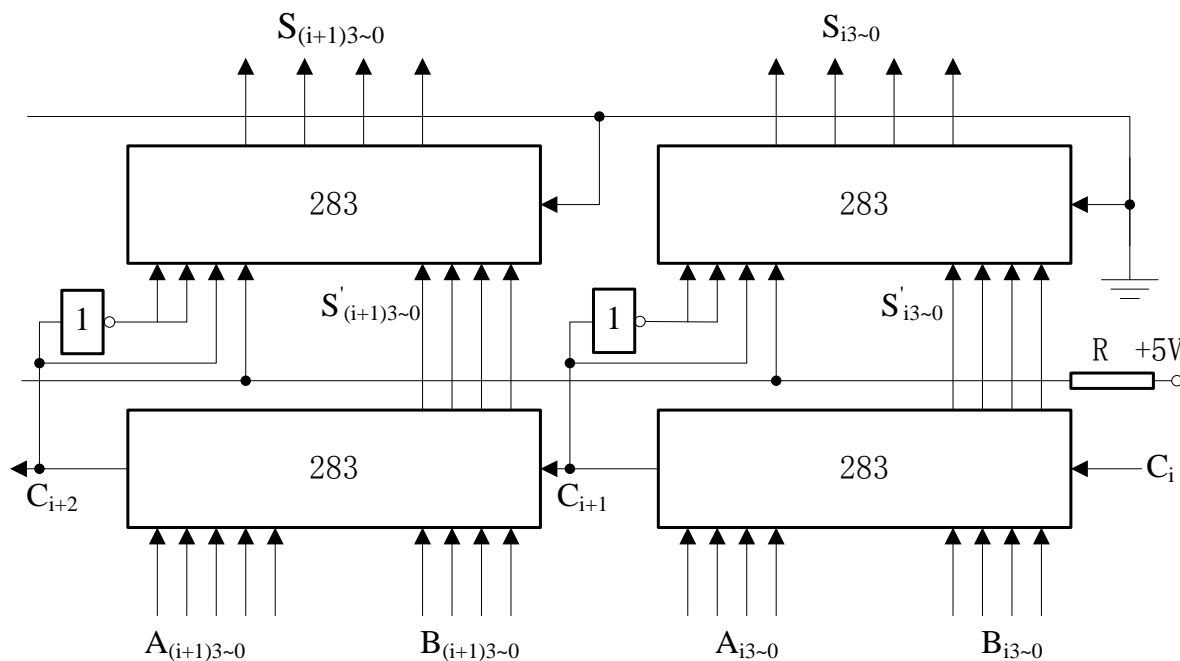
□ 方案二：采用4位先行进位二进制加法器MSI芯片构成（74LS283，也可选其他MSI加法器），线路如下：



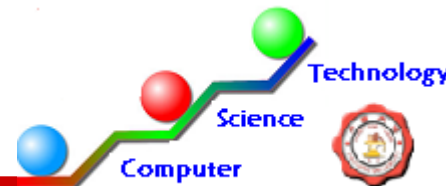
第六章 6.41 (2)



- (2) 2位余3码加法实现方案：用两个一位余3码加法器作为加法单元，采用较简单的串行进位方式，构成2位余3码加法线路（也可采用其他进位方式，进位原理与n位二进制加法器基本一样）。此方法也适用于n位余3码加法器的构成。线路结构如下：

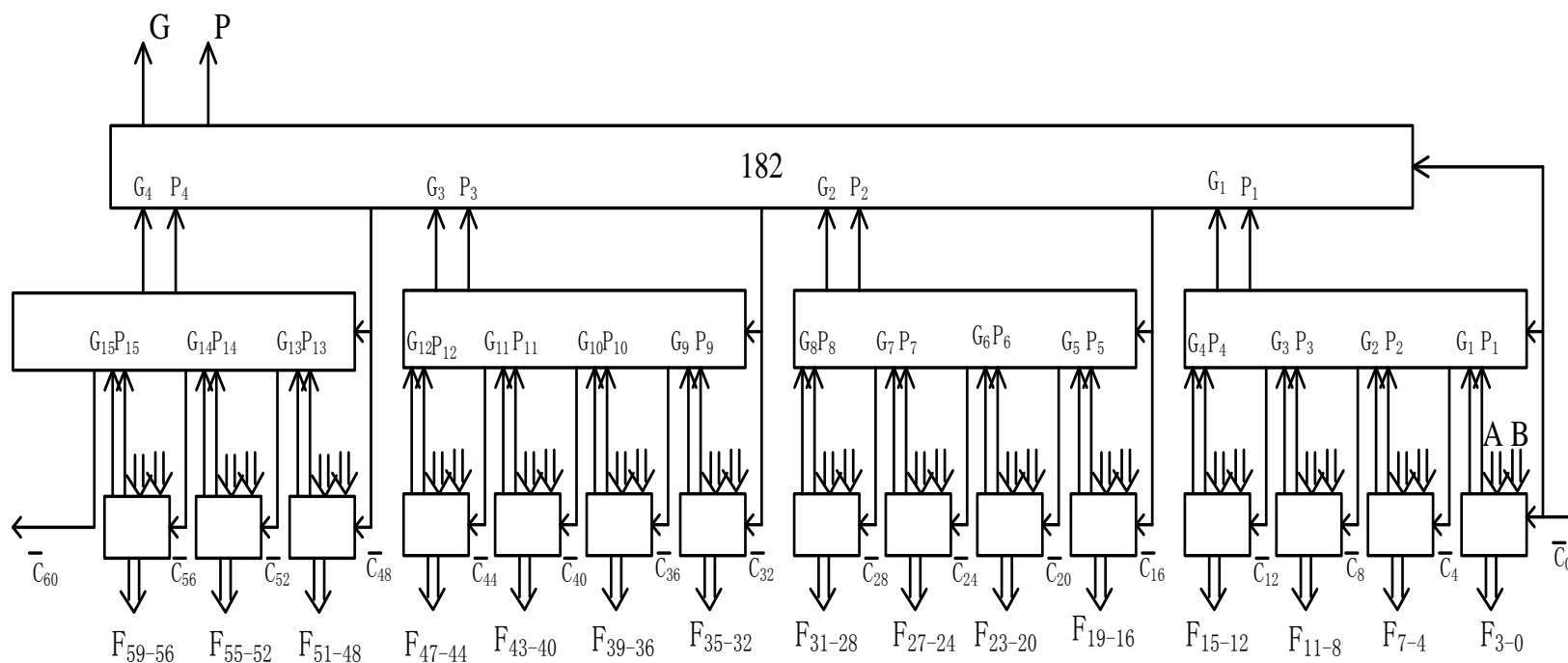


第六章 6.42

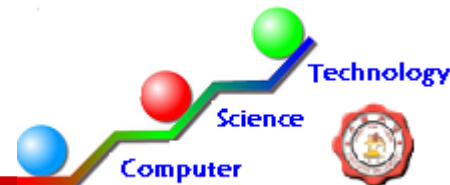


□ 6.42 试用74LS181、74LS182等中规模集成电路芯片组成一个分级先行进位的60位ALU。

□ 解：分级先行进位的60位ALU线路如下（正逻辑）：

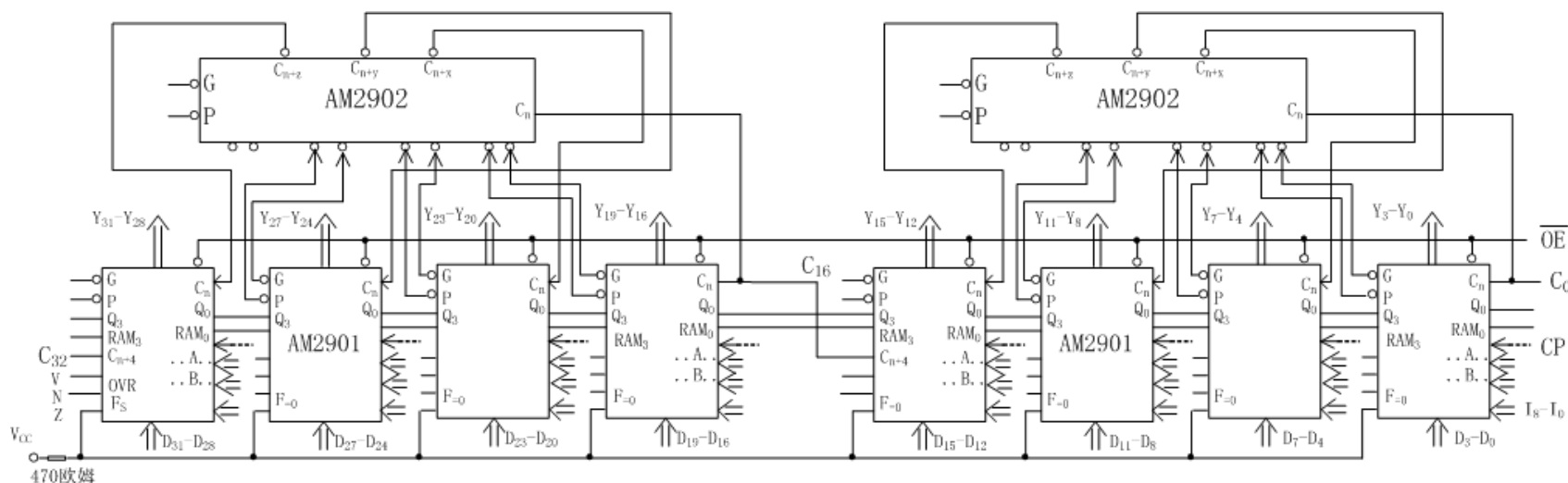


第六章 6.43

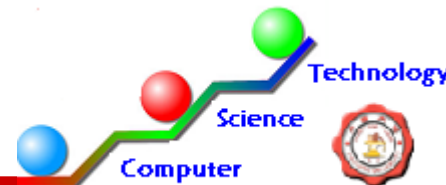


□ 6.43 用4位位片式运算器芯片AM2901和AM2902组成一个32位定点加、减、乘、除四则运算器，采用二级先行级联进位结构，请画出其逻辑电路图。

□ 解：32位定点四则运算器逻辑电路图如下：



第六章 6.44

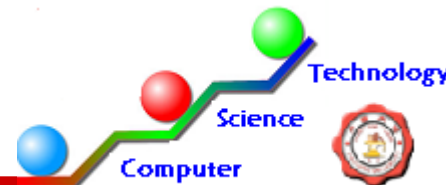


- 6.44 浮点数的格式为：阶码6位（含1位阶符），尾数10位（含1位数符）。按下列要求分别写出正数和负数的表示范围，答案均用2的幂形式的十进制真值表示。
- (1) 阶原尾原非规格化数；
 - (2) 阶移尾补规格化数；
 - (3) 按照(2)的格式，写出 $-27/1024$ 和 7.375 的浮点机器数。

□ 解：(1) 据题意画出该浮点数格式：

1	5	1	9
阶符	阶 码	数符	尾 数

第六章 6.44 (1)



□ 当采用阶原尾原非规格化数时，

○ 最大正数=0, 11 111; 0.111 111 111

○ 最小正数=1, 11 111; 0.000 000 001

○ 则正数表示范围为：

$$2^{-31} \times 2^{-9} \sim 2^{31} \times (1-2^{-9})$$

○ 最大负数=1, 11 111; 1.000 000 001

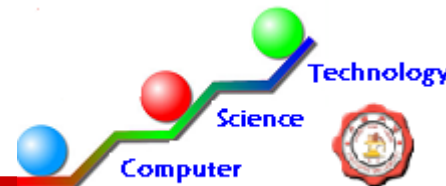
○ 最小负数=0, 11 111; 1.111 111 111

○ 则负数表示范围为：

$$-2^{31} \times (1-2^{-9}) \sim -2^{-31} \times 2^{-9}$$

□ 注：零视为中性数，不在此范围内。

第六章 6.44 (2)



(2) 当采用**阶移尾补规格化数**时，

○最大正数=**1**, 11 111; 0.111 111 111

○最小正数=0, 00 000; 0.100 000 000

○其对应的正数真值范围为：

$$2^{-32} \times 2^{-1} \sim 2^{31} \times (1 - 2^{-9})$$

○最大负数=0, 00 000; 1.011 111 111

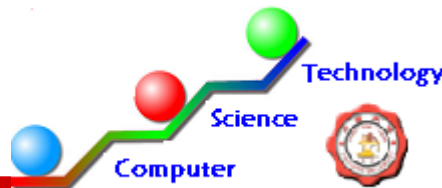
○最小负数=**1**, 11 111; 1.000 000 000

○其对应的负数真值范围为

$$2^{31} \times (-1) \sim -2^{-32} \times (2^{-1} + 2^{-9})$$

□ 注意：原码正、负域**对称**，补码正、负域**不对称**。浮点补码规格化尾数范围满足条件：**数符 \oplus MSB位=1**

第六章 6.44 (3)



(3) 首先将十进制数-27/1024和7.375转换为二进制:

○ $-27/1024 = (-0.000\ 001\ 101\ 1)_2 = 2^{-5} \times (-0.110\ 11)_2$

○ $7.375 = (111.011)_2 = 2^3 \times (0.111\ 011)_2$

○ 再写成浮点机器数形式:

○ -27/1024的阶移尾补规格化数=0, 11011; 1.001 010 000

○ 7.375的阶移尾补规格化数=1, 00011; 0.111011000

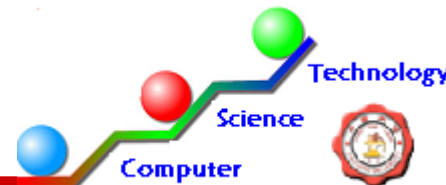
注: 以上浮点数也可采用如下格式:

1	1	5	9
数符	阶符	阶 码	尾 数

□ 此时只要将上述答案中的数符位移到最前面即可。

□ **注意:** 机器数末位的0不能省。

第六章 6.45



□ 6.45 (1) 将十进制数138.75 转换成32位的IEEE754短浮点数格式，并用十六进制缩写表示。

(2) 将IEEE754短浮点数C1B7 0000H转换成对应的十进制真值。

□ 解：

□ (1) 首先把十进制真值转换成符合IEEE754标准要求的二进制规格化真值形式： $(138.75)_{10} = (10001010.11)_2 = 1.0001\ 0101\ 1 \times 2^{11}$

然后计算阶码的移码 (=偏置常数+阶码真值)

$$E = +127 + 7 = 111\ 1111 + 111 = 1000\ 0110$$

写成短浮点数格式

$$S = 0, E = 1000\ 0110, \text{隐藏位} = 1.$$

$$M = .0001\ 0101\ 1000\ 0000\ 0000\ 000\ (23\text{位})$$

则 $(138.75)_{10}$ 的短浮点数机器码为：

$$0, 100\ 0011\ 0; .000\ 1010\ 1100\ 0000\ 0000\ 0000$$

□ 对应的十六进制缩写为：430A C000H

第六章 6.45



- (2) 首先把十六进制缩写展开成二进制机器码形式，并分离出符号位、阶码和尾数部分

$$\begin{aligned} \text{C1B7 0000H} &= 1100\ 0001\ 1011\ 0111\ 0000\ 0000\ 0000\ 0000\text{B} \\ &= 1, 100\ 0001\ 1; .011\ 0111\ 0000\ 0000\ 0000\ 0000 \end{aligned}$$

- 则 $S=1$, $E=1000\ 0011$, 隐藏位=1.

$$M = .011\ 0111\ 0000\ 0000\ 0000\ 0000\ (23\text{位})$$

- 计算出阶码的真值（即移码—偏置常数）

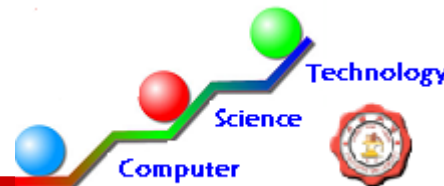
$$1000\ 0011 - 111\ 1111 = 100$$

- 写出此数的规格化二进制真值形式： $-1.011\ 0111 \times 2^{100}$

- 进一步去掉指数项： $-1.011\ 0111 \times 2^{100} = -1011\ 0.111$

- 转换成十进制真值： $(-1011\ 0.111)_2 = (-22.875)_{10}$

第六章 6.46



□ 6.46 设浮点数字长为32位，欲表示±60000间的十进制数，在保证数的最大精度条件下，除阶符、数符各取一位外，阶码和尾数各取几位？按这样分配，该浮点数溢出的条件是什么？

□ 解：若要保证数的最大精度，应取阶的基=2。

若要表示±6万间的十进制数，由于 $32768 (2^{15}) < 6万 < 65536 (2^{16})$ ，则：阶码除阶符外还应取5位（向上取2的幂）。

故：尾数位数为 $32-1-1-5=25$ 位

按此格式，该浮点数上溢的条件为：阶码 ≥ 32

该浮点数格式如下：

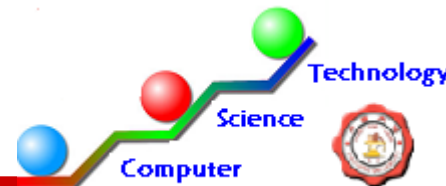
1	5	1	25
阶符	阶 值	数符	尾 数

□ 6.47 对于尾数为40位的浮点数（不包括符号位在内），若采用不同的机器数表示，试问当尾数左规或右规时，最多移位次数各为多少？

□ 解：

□ 对于尾数为40位的浮点数，若采用原码表示，当尾数左规时，最多移位**39次**；反码表示时情况同原码；若采用补码表示，当尾数左规时，正数最多移位**39次**，同原码；负数最多移位**40次**。当尾数右规时，不论采用何种码制，均只需右移**1次**。

第六章 6.48



□ 6.48 按机器补码浮点运算步骤计算 $[X \pm Y]_{\text{补}}$

(1) $X = 2^{-011} \times 0.101\ 100$, $Y = 2^{-010} \times (-0.011\ 100)$

(2) $X = 2^{101} \times (-0.100\ 101)$, $Y = 2^{100} \times (-0.001\ 111)$

解：设检测0步骤省略。

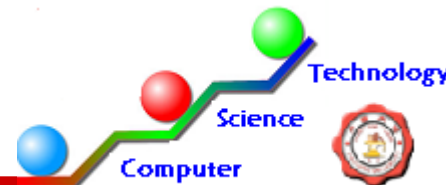
(1) 先将X、Y转换成浮点机器数形式：

$$[X]_{\text{补}} = 1, 101; 0.101\ 100$$

$$[Y]_{\text{补}} = 1, 110; 1.100\ 100$$

注：为简单起见，源操作数可直接写成浮点格式，不必规格化。

第六章 6.48 (1)



1) 对阶:

$$[\Delta E]_{\text{补}} = [E_x]_{\text{补}} + [-E_y]_{\text{补}} = 11, 101 + 00, 010 = 11, 111$$

$[\Delta E]_{\text{补}} < 0$, 应 E_x 向 E_y 对齐, 则:

$$[E_x]_{\text{补}} + 1 = 11, 101 + 00, 001 = 11, 110$$

$$[\Delta E]_{\text{补}} + 1 = 11, 111 + 00, 001 = 00, 000 = 0$$

至此, $E_x = E_y$, 对阶毕。

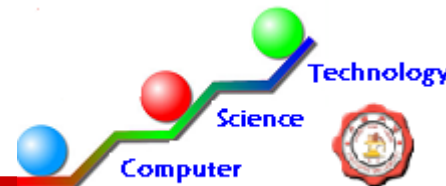
$$[X]_{\text{补}} = 1, 110; 0.010 \quad 110$$

2) 尾数相加减:

$$\begin{array}{r} [M_x]_{\text{补}} + [M_y]_{\text{补}} = 00.010 \quad 110 \\ + 11.100 \quad 100 \\ \hline 11.111 \quad 010 \end{array}$$

$$\begin{array}{r} [M_x]_{\text{补}} + [-M_y]_{\text{补}} = 00.010 \quad 110 \\ + 00.011 \quad 100 \\ \hline 00.110 \quad 010 \end{array}$$

第六章 6.48 (1)



3) 结果规格化:

$[X+Y]_{\text{补}} = 11, 110; 11.111\ 010 = 11, 011; 11.010\ 000$
(左规3次, 阶码减3, 尾数左移3位)

$[X-Y]_{\text{补}} = 11, 110; 00.110\ 010$
已是规格化数。

4) 舍入: 无

5) 溢出: 无

则: $X+Y = 2^{-101} \times (-0.110\ 000)$
 $X-Y = 2^{-010} \times 0.110\ 010$

第六章 6.48 (2)



$$(2) X=2^{101} \times (-0.100\ 101), Y=2^{100} \times (-0.001\ 111)$$

$$[X]_{\text{补}}=0, 101; 1.011\ 011, [Y]_{\text{补}}=0, 100; 1.110\ 001$$

1) 对阶:

$$[\Delta E]_{\text{补}}=[E_x]_{\text{补}}+[-E_y]_{\text{补}}=00, 101+11, 100=00, 001$$

$[\Delta E]_{\text{补}} > 0$, 应 E_y 向 E_x 对齐, 则:

$$[E_y]_{\text{补}}+1=00, 100+00, 001=00, 101$$

$$[\Delta E]_{\text{补}}+[-1]_{\text{补}}=00, 001+11, 111=00, 000=0$$

至此, $E_y=E_x$, 对阶毕。

$$[Y]_{\text{补}}=0, 101; 1.111\ 000\ (1)$$

2) 尾数运算:

$$\begin{array}{r} [M_x]_{\text{补}}+[M_y]_{\text{补}}=1\ 1.0\ 1\ 1\ 0\ 1\ 1 \\ +\ 1\ 1.1\ 1\ 1\ 0\ 0\ 0\ (1) \\ \hline 1\ 1.0\ 1\ 0\ 0\ 1\ 1\ (1) \end{array}$$

$$\begin{array}{r} [M_x]_{\text{补}}+[-M_y]_{\text{补}}=1\ 1.0\ 1\ 1\ 0\ 1\ 1 \\ +\ 0\ 0.0\ 0\ 0\ 1\ 1\ 1\ (1) \\ \hline 1\ 1.1\ 0\ 0\ 0\ 1\ 0\ (1) \end{array}$$

第六章 6.48 (2)



3) 结果规格化:

$$[X+Y]_{\text{补}} = 00,101; 11.010 \ 011 \ (1)$$

已是规格化数。

$$[X-Y]_{\text{补}} = 00,101; 11.100 \ 010 \ (1) = 00,100; 11.000 \ 101$$

(左规1次, 阶码减1, 尾数左移1位)

4) 舍入:

$$[X+Y]_{\text{补}} = 00,101; 11.010 \ 011 \ (\text{舍})$$

$$[X-Y]_{\text{补}} \text{ 不变。}$$

$$[X-Y]_{\text{补}} = 00,100; 11.000 \ 101$$

5) 溢出: 无

$$\text{则: } X+Y = 2^{101} \times (-0.101 \ 101)$$

$$X-Y = 2^{100} \times (-0.111 \ 011)$$

第六章 6.49



□6.49 设浮点数阶码取3位，尾数取6位（均不包括符号位），要求阶码用移码运算，尾数用原码运算，计算 $X \times Y$ 和 $X \div Y$ ，且结果保留1倍字长。

$$(1) X=2^{100} \times 0.100\ 111, \quad Y=2^{011} \times (-0.101\ 011)$$

$$(2) X=2^{101} \times (-0.101\ 101), \quad Y=2^{001} \times (-0.111\ 100)$$

解：设检测0步骤省略。

(1) 先将 X 、 Y 转换成机器数形式：

$$[X]_{\text{阶移尾原}} = 1,100; 0.100\ 111$$

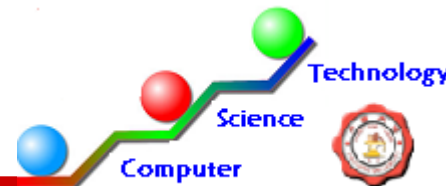
$$[Y]_{\text{阶移尾原}} = 1,011; 1.101\ 011$$

1) 阶码相加减：

$$[Ex]_{\text{移}} + [Ey]_{\text{补}} = 01,100 + 00,011 = 01,111 \quad (\text{无溢出})$$

$$[Ex]_{\text{移}} + [-Ey]_{\text{补}} = 01,100 + 11,101 = 01,001 \quad (\text{无溢出})$$

第六章 6.49 (1)



2) 尾数相乘除:

○ 尾数相乘:

$$[M_x]_{\text{原}} = 0.100\ 111, [M_y]_{\text{原}} = 1.101\ 011$$

$$M_x^* = 0.100\ 111, M_y^* = 0.101\ 011$$

$$M_{x_0} = 0, M_{y_0} = 1, M_{p_0} = M_{x_0} \oplus M_{y_0} = 0 \oplus 1 = 1$$

$$M_x^* \times M_y^* = 0.011\ 010\ 001\ 101$$

$$[M_x \times M_y]_{\text{原}} = 1.011\ 010\ 001\ 101$$

$$[P]_{\text{浮}} = [X \times Y]_{\text{阶移尾原}} = 01,111; \ 1.011\ 010\ 001\ 101$$

○ 尾数相除:

$$[-M_y^*]_{\text{补}} = 1.010\ 101$$

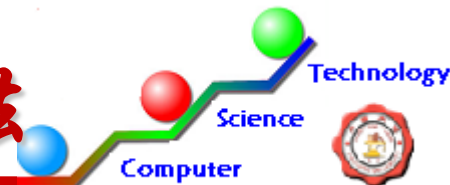
$$M_x^* \div M_y^* = 0.111\ 010, [M_x \div M_y]_{\text{原}} = 1.111\ 010$$

$$r^* = 0.000\ 010 \times 2^{-6} = 0.000\ 000\ 000\ 010$$

$$[Q]_{\text{浮}} = [X \div Y]_{\text{阶移尾原}} = 01,001; \ 1.111\ 010$$

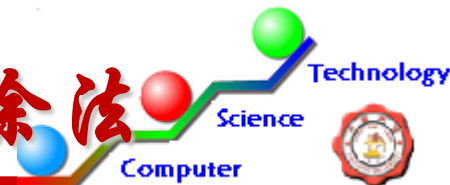
运算过程如下:

第六章6. 49 (1) 原码一位乘法



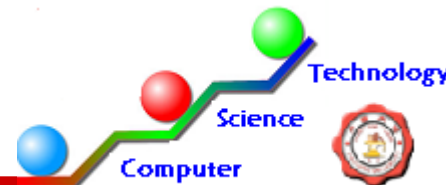
部分积	乘数Y*
$\begin{array}{r} 0.000\ 000 \\ +\ 0.100\ 111 \\ \hline 0.100\ 111 \end{array}$	$.1\ 0\ 1\ 0\ 1\ \underline{1} \text{---} +X^*$
$\begin{array}{r} 0.100\ 111 \\ \rightarrow 1\ 0.010\ 011 \\ +\ 0.100\ 111 \\ \hline 0.111\ 010 \end{array}$	$\begin{array}{r} 1\ .1\ 0\ 1\ 0\ \underline{1} \text{---} +X^* \\ \hline 0\ 1\ .1\ 0\ 1\ \underline{0} \text{---} +0 \\ 1\ 0\ 1\ .1\ 0\ \underline{1} \text{---} +X^* \end{array}$
$\begin{array}{r} 0.111\ 010 \\ \rightarrow 1\ 0.011\ 101 \\ \rightarrow 1\ 0.001\ 110 \\ +\ 0.100\ 111 \\ \hline 0.110\ 101 \end{array}$	$\begin{array}{r} 1\ 1\ 0\ 1\ .1\ \underline{0} \text{---} +0 \\ 0\ 1\ 1\ 0\ 1\ .\underline{1} \text{---} +X^* \end{array}$
$\begin{array}{r} 0.110\ 101 \\ \rightarrow 1\ 0.011\ 010 \\ \rightarrow 1\ 0.001\ 101 \\ +\ 0.100\ 111 \\ \hline 0.110\ 100 \end{array}$	$\begin{array}{r} 0\ 0\ 1\ 1\ 0\ \underline{1} \end{array}$
$\begin{array}{r} 0.110\ 100 \\ \rightarrow 1\ 0.011\ 010 \end{array}$	

第六章 6.49 (1) 原码加减交替除法



被除数 (余数) 00.100 111	商 0.000 000
+ 11.010 101	试减, $+[-My^*]_{补}$
11.111 100	$r < 0$, 商0
1← 11.111 000	0.
+ 00.101 011	$+My^*$
00.100 011	$r > 0$, 商1
1← 01.000 110	0.1
+ 11.010 101	$+[-My^*]_{补}$
00.011 011	$r > 0$, 商1
1← 00.110 110	0.1 1
+ 11.010 101	$+[-My^*]_{补}$
00.001 011	$r > 0$, 商1
1← 00.010 110	0.1 1 1
+ 11.010 101	$+[-My^*]_{补}$
11.101 011	$r < 0$, 商0
1← 11.010 110	0.1 1 1 0
+ 00.101 011	$+My^*$
00.000 001	$r > 0$, 商1
1← 00.000 010	0.1 1 1 0 1
+ 11.010 101	$+[-My^*]_{补}$
11.010 111	1← 0.1 1 1 0 1 0, $r < 0$, 商0
+ 00.101 011	(恢复余数), $+My^*$
00.000 010	

第六章 6.49 (1)



3) 结果规格化:

$$\begin{aligned}[X \times Y]_{\text{阶移尾原}} &= 01,111; \quad 1.011 \ 010 \ 001 \ 101 \\ &= 01,110; \quad 1.110 \ 100 \ 011 \ 010 \quad (\text{左规1位}) \\ [X \div Y]_{\text{阶移尾原}} &= 01,001; \quad 1.111 \ 010 \quad (\text{不变}) \\ &\quad \text{——已是规格化数}\end{aligned}$$

4) 舍入:

$$\begin{aligned}[P]_{\text{浮}} &= [X \times Y]_{\text{阶移尾原}} = 01,110; \quad 1.110 \ 100 \ 011 \ 010 \\ &= 01,110; \quad 1.110 \ 100 \quad (\text{舍}) \\ [Q]_{\text{浮}} &= [X \div Y]_{\text{阶移尾原}} = 01,001; \quad 1.111 \ 010 \quad (\text{不变})\end{aligned}$$

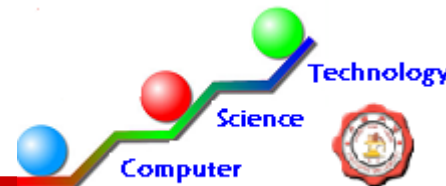
5) 溢出: 无

则

$$X \times Y = 2^{110} \times (-0.110 \ 100)$$

$$X \div Y = 2^{001} \times (-0.111 \ 010)$$

第六章 6.49 (2)



$$(2) \quad X=2^{101} \times (-0.101 \ 101), \quad Y=2^{001} \times (-0.111 \ 100)$$

$$[X]_{\text{阶移尾原}} = 1, \ 101; \ 1.101 \ 101$$

$$[Y]_{\text{阶移尾原}} = 1, \ 001; \ 1.111 \ 100$$

1) 阶码相加减:

$$[Ex]_{\text{移}} + [Ey]_{\text{补}} = 01, \ 101 + 00, \ 001 = 01, \ 110 \quad (\text{无溢出})$$

$$[Ex]_{\text{移}} + [-Ey]_{\text{补}} = 01, \ 101 + 11, \ 111 = 01, \ 100 \quad (\text{无溢出})$$

2) 尾数相乘除:

○ 尾数相乘:

$$[Mx]_{\text{原}} = 1.101 \ 101, \quad [My]_{\text{原}} = 1.111 \ 100$$

$$Mx^* = 0.101 \ 101, \quad My^* = 0.111 \ 100$$

$$Mx_0 = 1, \quad My_0 = 1, \quad Mp_0 = Mx_0 \oplus My_0 = 1 \oplus 1 = 0$$

第六章 6.49 (2) ——尾数相乘除



○ 尾数相乘:

$$M_x^* \times M_y^* = 0.101\ 010\ 001\ 100$$

$$[M_x \times M_y]_{\text{原}} = 0.101\ 010\ 001\ 100$$

$$[P]_{\text{浮}} = [X \times Y]_{\text{阶移尾原}} = 01,110; \quad 0.101\ 010\ 001\ 100$$

○ 尾数相除:

$$[-M_y^*]_{\text{补}} = 1.000\ 100$$

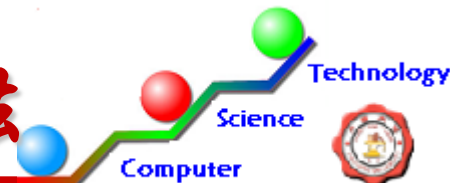
$$M_x^* \div M_y^* = [M_x \div M_y]_{\text{原}} = 0.110\ 000$$

$$r^* = 0.000\ 000 \times 2^{-6} = 0.000\ 000\ 000\ 000 = 0$$

$$[Q]_{\text{浮}} = [X \div Y]_{\text{阶移尾原}} = 01,100; \quad 0.110\ 000$$

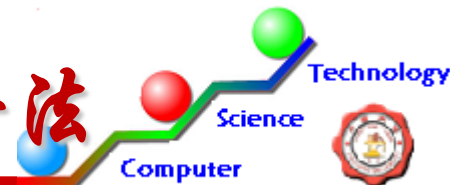
○ 运算过程如下:

第六章 6.49 (2) 原码一位乘法



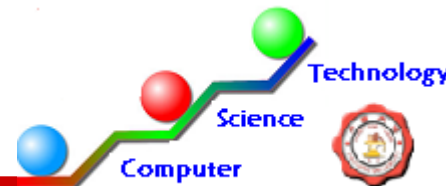
部分积	乘数Y*
0.000 000	.1 1 1 1 0 0 ——— +0
→1 0.000 000	0 .1 1 1 1 0 ——— +0
→1 0.000 000	0 0 .1 1 1 1 ——— +X*
+ 0.101 101	
0.101 101	
→1 0.010 110	1 0 0 .1 1 1 ——— +X*
+ 0.101 101	
1.000 011	
→1 0.100 001	1 1 0 0 .1 1 ——— +X*
+ 0.101 101	
1.001 110	
→1 0.100 111	0 1 1 0 0 .1 ——— +X*
+ 0.101 101	
1.010 100	
→1 0.101 010	0 0 1 1 0 0

第六章 6.49 (2) 原码加减交替除法



被除数 (余数)	商
00.101 101	0.000 000
+ 11.000 100	试减, $+[-My^*]_{补}$
11.110 001	$r < 0$, 商0
1← 11.100 010	0.
+ 00.111 100	$+My^*$
00.011 110	$r > 0$, 商1
1← 00.111 100	0.1
+ 11.000 100	$+[-My^*]_{补}$
00.000 000	$r > 0$, 商1
1← 00.000 000	0.1 1
+ 11.000 100	$+[-My^*]_{补}$
11.000 100	$r < 0$, 商0
1← 10.001 000	0.1 1 0
+ 00.111 100	$+My^*$
11.000 100	$r < 0$, 商0
1← 10.001 000	0.1 1 0 0
+ 00.111 100	$+My^*$
11.000 100	$r < 0$, 商0
1← 10.001 000	0.1 1 0 0 0
+ 00.111 100	$+My^*$
11.000 100	1← 0.1 1 0 0 0 0, $r < 0$, 商0
+ 00.111 100	恢复余数, $+My^*$
00.000 000	

第六章 6.49(2)



3) 结果规格化：已是规格化数。

4) 舍入：

$$[P]_{\text{浮}} = [X \times Y]_{\text{阶移尾原}} = 01,110; 0.101\ 010\ 001\ 100 \\ = 01,110; 0.101\ 010 \text{ (舍)}$$

$$[Q]_{\text{浮}} = [X \div Y]_{\text{阶移尾原}} = 01,100; 0.110\ 000 \text{ (不变)}$$

5) 溢出：无

则

$$X \times Y = 2^{110} \times 0.101\ 010$$

$$X \div Y = 2^{100} \times 0.110\ 000$$

注：由于加减交替除法算法中缺少对部分余数判“0”的步骤，因此算法运行中的某一步已除尽时，算法不会自动停止，而是继续按既定步数运行完。

第六章 6.50



- 6.50 设数的阶码3位，尾数6位，均不含符号位；阶码用移码表示，尾数用补码表示；阶的基为2。用浮点算法计算 $X+Y$ 、 $X-Y$ 、 $X \times Y$ 、 $X \div Y$ ，结果要求为规格化数。

已知： $X=2^{-2} \times 11/16$ ； $Y=2^3 \times (-15/16)$

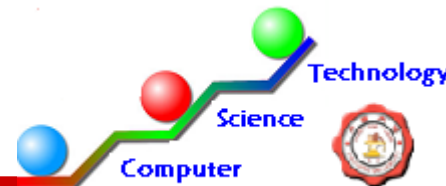
- 解：为判溢出方便，阶符和尾符均采用双符号位，先将 X 、 Y 转换成浮点规格化格式（如果仅作为练习，原始数据也可直接转换，不要求一定为规格化形式，运算完后再规格化）：

$$[X]_{\text{阶移尾补}} = 00, 110; 00.101\ 100$$

$$[Y]_{\text{阶移尾补}} = 01, 011; 11.000\ 100$$

- 为便于讨论，设阶码用 E 表示，尾数用 M 表示。

第六章 6.50 浮点加减法



1. 浮点加减法:

(1) 对阶:

求阶差: $[\Delta E]_{\text{移}} = [E_x]_{\text{移}} + [-E_y]_{\text{补}} = 00\ 110 + 11\ 101 = 00\ 011$

$[\Delta E]_{\text{移}} < 0$, $E_x < E_y$, $\Delta E = -5$, E_x 向 E_y 对齐。

M_x 右移 5 位。每右移一次, $E_x + 1$, 直到 $E_x = E_y$ 为止。

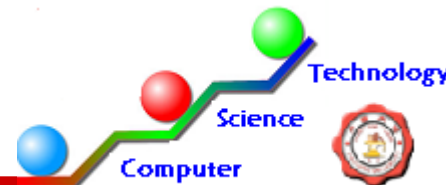
对阶后: $[X]_{\text{阶移尾补}} = 01, 011; 00.000\ 001\ (01100)$

(2) 尾数运算:

$$\begin{aligned} [M+]_{\text{补}} &= [M_x]_{\text{补}} + [M_y]_{\text{补}} \\ &= 00.000\ 001\ (011) + 11.000\ 100 \\ &= 11.000\ 101\ (011) \end{aligned}$$

$$\begin{aligned} [M-]_{\text{补}} &= [M_x]_{\text{补}} + [-M_y]_{\text{补}} \\ &= 00.000\ 001\ (011) + 00.111\ 100 \\ &= 00.111\ 101\ (011) \end{aligned}$$

第六章 6.50 浮点加减法



(3) 结果规格化:

设尾数的高三位为 $M_s M_0 \cdot M_{MSB} \dots$,

加法时: $M_0 \oplus M_{MSB} = 1 \oplus 0 = 1$;

减法时: $M_0 \oplus M_{MSB} = 0 \oplus 1 = 1$;

则: $[M+]_{\text{补}}$ 、 $[M-]_{\text{补}}$ 已是规格化数, 不需再规格化。

(4) 舍入:

采用0舍1入法

$[X+Y]_{\text{阶移尾补}} = 01, 011; 11.000\ 101$ (舍)

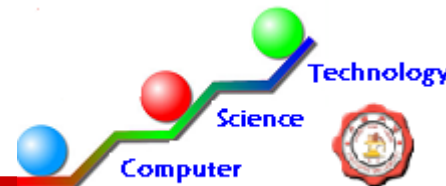
$[X-Y]_{\text{阶移尾补}} = 01, 011; 00.111\ 101$ (舍)

(5) 溢出判断:

由于 $[X+Y]_{\text{阶移尾补}}$ 、 $[X-Y]_{\text{阶移尾补}}$ 的阶码均未溢出, 故结果无溢出。

则: $X+Y = 2^3 \times (-59/64)$; $X-Y = 2^3 \times 61/64$

第六章 6.50 浮点乘法



2. 浮点乘法:

(1) 阶码相加:

$$[E_x]_{\text{移}} = [E_x]_{\text{移}} + [E_y]_{\text{补}} = 00\ 110 + 00\ 011 = 01\ 001 \text{—无溢出}$$

(2) 尾数相乘:

采用补码两位乘比较法，有：

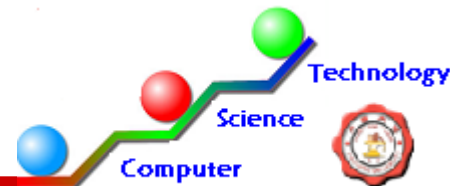
$$[M_x]_{\text{补}} = 00.101\ 100; [M_y]_{\text{补}} = 11.000\ 100$$

$$[-M_x]_{\text{补}} = 11.010\ 100$$

$$[M_x]_{\text{补}} = 111.010\ 110\ (110\ 000\ 00)$$

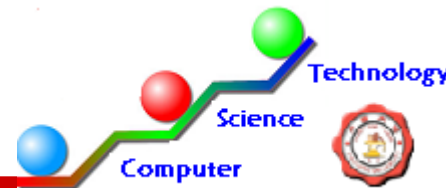
机器运算步骤如下：

第六章 6.50 浮点乘法



部分积	乘数	$Y_{n-1}Y_nY_{n+1}$
0 0 0 . 0 0 0 0 0 0 0	1 1 . 0 0 0 1	<u>0 0 0</u> —+0
→2 0 0 0 . 0 0 0 0 0 0	0 0	<u>1 1 . 0 0</u> <u>0 1 0</u>
+ 0 0 0 . 1 0 1 1 0 0		$+[M_x]_{补}$
0 0 0 . 1 0 1 1 0 0		
→2 0 0 0 . 0 0 1 0 1 1	0 0 0 0	<u>1 1 . 0 0 0</u> —+0
→2 0 0 0 . 0 0 0 0 1 0	1 1 0 0 0 0	<u>1 1 . 0</u>
+ 1 1 1 . 0 1 0 1 0 0		$+[-M_x]_{补}$
1 1 1 . 0 1 0 1 1 0	1 1 0 0 0 0	<u>0 0</u>
		清0

第六章 6.50 浮点除法



(3) 结果规格化:

$$M_0 \oplus M_{\text{MSB}} = 1 \oplus 0 = 1, [M_x]_{\text{补}} \text{ 已是规格化数。}$$

(4) 舍入: 采用0舍1入法

$$[X \times Y]_{\text{阶移尾补}} = 01, 001; 11.010\ 111 \text{ (入)}$$

(5) 判溢出:

由于 $[X \times Y]_{\text{阶移尾补}}$ 的阶码未溢出, 故结果无溢出。

$$\text{则: } X \times Y = 2^1 \times (-41/64)$$

3. 浮点除法:

(1) 阶码相减:

$$[E_{\div}]_{\text{移}} = [E_x]_{\text{移}} + [-E_y]_{\text{补}} = 00\ 110 + 11\ 101 = 00\ 011 \text{ ——无溢出}$$

(2) 尾数相除:

采用补码加减交替除法, 由于满足 $X^* < Y^*$ 条件, 除法过程无溢出。

$$[M_x]_{\text{补}} = 00.101\ 100; [M_y]_{\text{补}} = 11.000\ 100; [-M_y]_{\text{补}} = 00.111\ 100$$

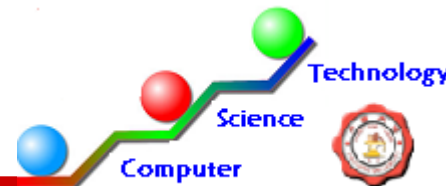
$$[M_{\div}]_{\text{补}} = 1.010\ 001, \text{ 余数忽略。}$$

若考虑余数还要进行一次恢复余数操作。机器运算步骤如下:

第六章 6.50 浮点除法

被除数 $[M_x]_{补}$ /余数 $[M_r]_{补}$	商q
$\begin{array}{r} 00.101100 \\ + 11.000100 \\ \hline \end{array}$	0.000000
$\begin{array}{r} 11.110000 \\ \leftarrow 1 \quad 11.100000 \\ + 00.111100 \\ \hline \end{array}$	$X、Y \text{ 异号, } +[M_y]_{补}$
$\begin{array}{r} 00.011100 \\ \leftarrow 1 \quad 00.111000 \\ + 11.000100 \\ \hline \end{array}$	$1. \text{ —— } R、Y \text{ 同号, 商1}$
$\begin{array}{r} 11.111100 \\ \leftarrow 1 \quad 11.111000 \\ + 00.111100 \\ \hline \end{array}$	$+[-M_y]_{补}$
$\begin{array}{r} 00.110100 \\ \leftarrow 1 \quad 01.101000 \\ + 11.000100 \\ \hline \end{array}$	$1.0 \text{ —— } R、Y \text{ 异号, 商0}$
$\begin{array}{r} 00.101100 \\ \leftarrow 1 \quad 01.011000 \\ + 11.000100 \\ \hline \end{array}$	$+ [M_y]_{补}$
$\begin{array}{r} 00.011100 \\ \leftarrow 1 \quad 00.111000 \\ + 11.000100 \\ \hline \end{array}$	$1.01 \text{ —— } R、Y \text{ 同号, 商1}$
$\begin{array}{r} 11.111100 \\ \leftarrow 1 \quad 11.111000 \\ + 00.111100 \\ \hline \end{array}$	$+ [-M_y]_{补}$
$\begin{array}{r} 00.101100 \\ \leftarrow 1 \quad 01.101000 \\ + 11.000100 \\ \hline \end{array}$	$1.010 \text{ —— } R、Y \text{ 异号, 商0}$
$\begin{array}{r} 00.011100 \\ \leftarrow 1 \quad 00.111000 \\ + 11.000100 \\ \hline \end{array}$	$+ [M_y]_{补}$
$\begin{array}{r} 00.101100 \\ \leftarrow 1 \quad 01.101000 \\ + 11.000100 \\ \hline \end{array}$	$1.0100 \text{ —— } R、Y \text{ 异号, 商0}$
$\begin{array}{r} 00.011100 \\ \leftarrow 1 \quad 00.111000 \\ + 11.000100 \\ \hline \end{array}$	$+ [M_y]_{补}$
$\begin{array}{r} 00.101100 \\ \leftarrow 1 \quad 01.101000 \\ + 11.000100 \\ \hline \end{array}$	$1.01000 \text{ —— } R、Y \text{ 异号, 商0}$
$\begin{array}{r} 00.011100 \\ \leftarrow 1 \quad 00.111000 \\ + 11.000100 \\ \hline \end{array}$	$+ [M_y]_{补}$
$\begin{array}{r} 11.111100 \\ \leftarrow 1 \quad 11.111000 \\ + 00.111100 \\ \hline \end{array}$	1.010001 —— 恒置1

第六章 6.50 浮点除法



(3) 结果规格化:

$$M_0 \oplus M_{MSB} = 1 \oplus 0 = 1, [M \div]_{\text{补}} \text{ 已是规格化数。}$$

(4) 舍入:

由于尾数除法采用了恒置1法舍入, 故不用再进行其他舍入操作。
(若采用0舍1入法舍入, 可多求几位商作为保护位。)

$$[X \div Y]_{\text{阶移尾补}} = 00, 011; 11.010\ 001$$

(5) 判溢出:

由于 $[X \div Y]_{\text{阶移尾补}}$ 的阶码无溢出, 故结果无溢出。

$$\text{则: } X \div Y = 2^{-5} \times (-47/64)$$

□ 评注: 浮点运算与定点运算的主要区别在运算步骤上, 每一步的具体操作方法基本以定点算法为基础; 阶码运算与尾数运算分别进行, 溢出判断以阶码溢出为标志, 尾数溢出可通过规格化操作进行调整。浮点运算时舍入问题比较突出, 为尽量减少精度损失, 一般设有若干保护位, 因此本题在运算过程中保留多余位, 直到舍入操作时才对保留位进行处理。注意最后结果按题意要求用浮点真值表示, 真值的形式要与原始数据一致。

第六章 6.51



□ 6.51 当采用最高位为奇校验位的ASCII码方案时，写出下列字符的机内码。

G, g, 7, !, &

□ 解：

□ G的ASCII码=47H； G的ASCII机内码=C7H

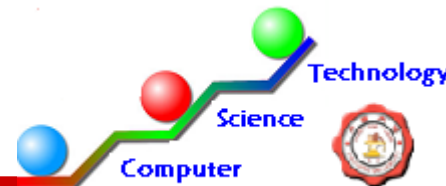
g的ASCII码=67H； g的ASCII机内码=67H

7的ASCII码=37H； 7的ASCII机内码=37H

! 的ASCII码=21H； ! 的ASCII机内码=A1H

&的ASCII码=26H； &的ASCII机内码=26H

第六章 6.52



□ 6.62 设有效信息为110，试用生成多项式 $G(X) = 11011$ 将其编成循环冗余校验码。

□ 解：编码过程如下：

$$M(X) = 110, \quad n = 3$$

$$G(X) = 11011, \quad k+1 = 5, \quad k = 4$$

$$M(X) \cdot X^4 = 110 \text{ } 0000$$

$$\begin{aligned} M(X) \cdot X^4 / G(X) &= 110 \text{ } 0000 / 11011 \\ &= 100 + 1100 / 11011, \quad R(X) = 1100 \end{aligned}$$

$$\begin{aligned} M(X) \cdot X^4 + R(X) &= 110 \text{ } 0000 + 1100 \\ &= 110 \text{ } 1100 = \text{CRC码} \quad (7, 3) \text{ 码} \end{aligned}$$

□ 注：此题的 $G(X)$ 选得不太好，当最高位和最低位出错时，余数相同，均为0001。此时只能检错，无法纠错。

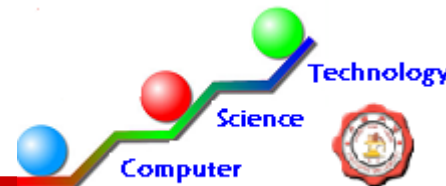
□ 6.53 试比较逻辑移位和算术移位。

□ 解：逻辑移位和算术移位的区别：

逻辑移位是对逻辑数或无符号数进行的移位，其特点是不论左移还是右移，**空出位均补0**，移位时不考虑符号位。

算术移位是对带符号数进行的移位操作，其关键规则是移位时**符号位保持不变**，空出位的补入值与数的正负、移位方向、采用的码制等有关。补码或反码右移时具有**符号延伸特性**。左移时可能**产生溢出**错误，右移时可能**丢失精度**。

第六章 6.54



□ 6.54 在整数定点机中，设机器数采用一位符号位，写出 ± 0 的原码、补码、反码和移码，得出什么结论？

□ 解：不同码制0的机器数形式如下：

真值	原码	补码	反码	移码
+0	0, 00...0	0, 00...0	0, 00...0	1, 00...0
-0	1, 00...0	0, 00...0	1, 11...1	1, 00...0

□ 结论：补、移码0的表示唯一，原、反码不唯一。

□ 注意：本题不用分析不同编码间的其他特性。

第六章 6.55



□ 6.55 求证：定点小数运算时， $[X]_{\text{补}} + [Y]_{\text{补}} = [X + Y]_{\text{补}} \pmod{4}$

□ 证：根据模4补码定义：

$$[X]_{\text{补}} = \begin{cases} X & 2 > X \geq 0 \\ 4 + X & 0 > X \geq -2 \end{cases} \pmod{4}$$

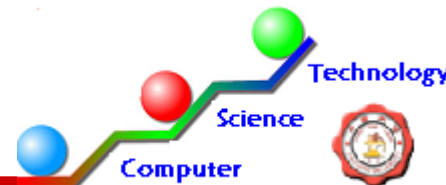
□ 设X、Y为定点小数，X+Y亦为定点小数，分四种情况证明：

(1) $1 > X \geq 0$ ， $1 > Y \geq 0$ ，则

$$[X]_{\text{补}} + [Y]_{\text{补}} = X + Y = [X + Y]_{\text{补}} \pmod{4}$$

当 $2 > X + Y \geq 1$ 时，X + Y溢出。

第六章 6.55



(2) $1 > X \geq 0$, $0 > Y \geq -1$, 则

$$\begin{aligned} [X]_{\text{补}} + [Y]_{\text{补}} &= X + 4 + Y \\ &= 4 + (X + Y) = \begin{cases} X + Y = [X + Y]_{\text{补}}, & 1 > X + Y \geq 0 \\ [X + Y]_{\text{补}}, & 0 > X + Y \geq -1 \end{cases} \quad (\text{mod } 4) \end{aligned}$$

(3) $1 > Y \geq 0$, $0 > X \geq -1$, 证明方法同(2), 略。

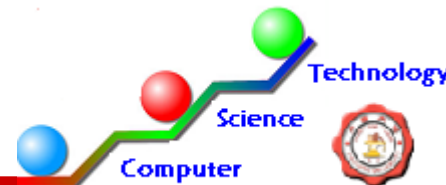
(4) $0 > X \geq -1$, $0 > Y \geq -1$, 则

$$\begin{aligned} [X]_{\text{补}} + [Y]_{\text{补}} &= 4 + X + 4 + Y = 4 + (4 + X + Y) = 4 + X + Y \quad (\text{mod } 4) \\ &= [X + Y]_{\text{补}} \quad (\text{mod } 4) \end{aligned}$$

当 $-1 > X + Y \geq -2$ 时, $X + Y$ 溢出。

□ 注: 此题需用模4补码定义进行论证。

第六章 6.56



□ 6.56 求证: $-[Y]_{\text{补}} = [-Y]_{\text{补}} \pmod{4}$

□ 证: 利用模4补码加法公式进行证明。

因为 $[X]_{\text{补}} + [Y]_{\text{补}} = [X + Y]_{\text{补}} \pmod{4}$

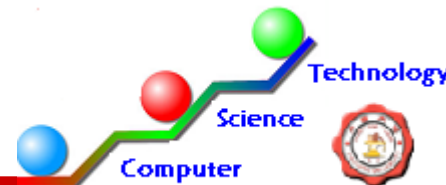
令 $X = -Y$ 代入上式得

$$[-Y]_{\text{补}} + [Y]_{\text{补}} = [-Y + Y]_{\text{补}} = [0]_{\text{补}} = 0 \pmod{4}$$

所以 $[-Y]_{\text{补}} = -[Y]_{\text{补}} \pmod{4}$ 证毕。

□ 评注: $[-Y]_{\text{补}}$ 称为 $[Y]_{\text{补}}$ 的机器负数, 利用此关系, 可方便地将补码减法转换成加法来做。

第六章 6.57



□ 6.57 已知: $[Y]_{\text{补}} = Y_0.Y_1Y_2\ldots Y_n$,

求证: $[-Y]_{\text{补}} = /Y_0./Y_1/Y_2\ldots /Y_n + 2^{-n}$

□ 证: 当 $1 > Y \geq 0$ 时, $Y_0 = 0$, $[Y]_{\text{补}} = [Y]_{\text{原}} = 0.Y_1Y_2\ldots Y_n$

$[-Y]_{\text{原}} = 1.Y_1Y_2\ldots Y_n$; 则

$[-Y]_{\text{补}} = 1./Y_1/Y_2\ldots /Y_n + 2^{-n} = /Y_0./Y_1/Y_2\ldots /Y_n + 2^{-n}$

当 $0 > Y \geq -1$ 时, $Y_0 = 1$, $[Y]_{\text{补}} = 1.Y_1Y_2\ldots Y_n$

$[Y]_{\text{原}} = 1./Y_1/Y_2\ldots /Y_n + 2^{-n}$

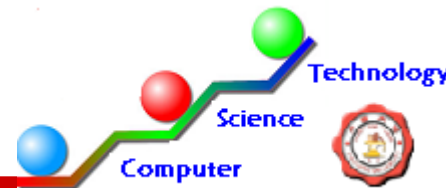
$[-Y]_{\text{原}} = 0./Y_1/Y_2\ldots /Y_n + 2^{-n}$

$[-Y]_{\text{补}} = [-Y]_{\text{原}} = 0./Y_1/Y_2\ldots /Y_n + 2^{-n}$
 $= /Y_0./Y_1/Y_2\ldots /Y_n + 2^{-n}$

□ 所以: 不论 Y 是正还是负, 在整个补码定义域均有

$[-Y]_{\text{补}} = /Y_0./Y_1/Y_2\ldots /Y_n + 2^{-n}$

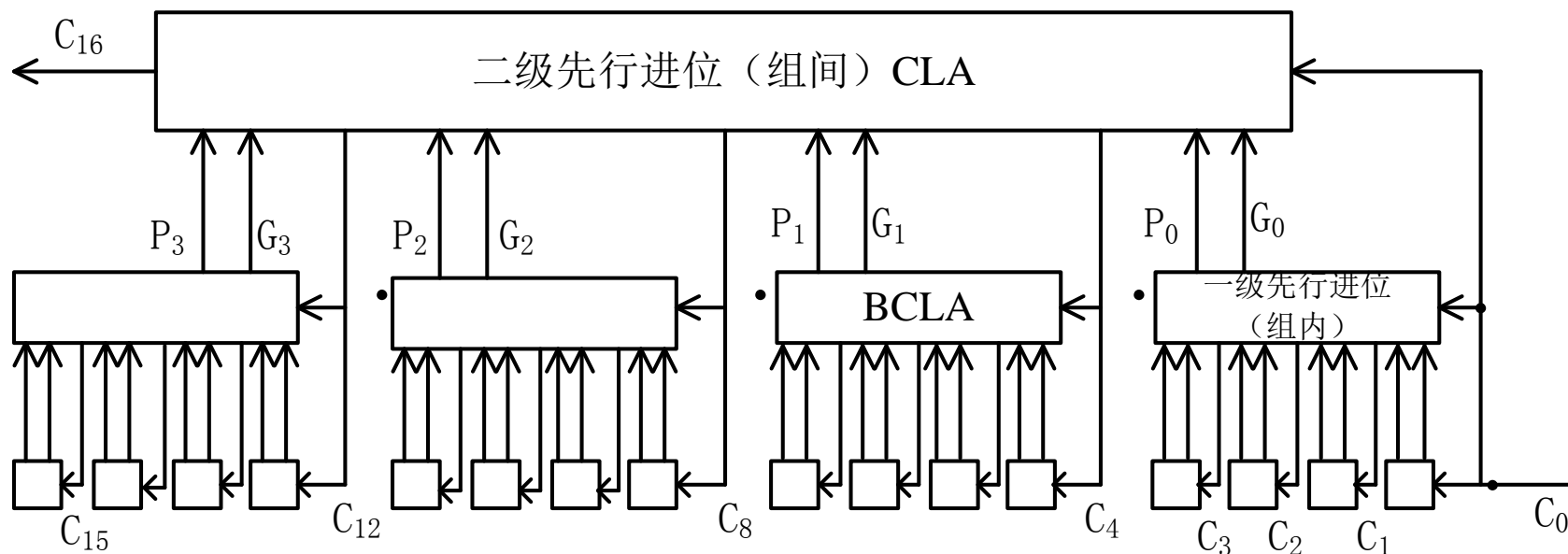
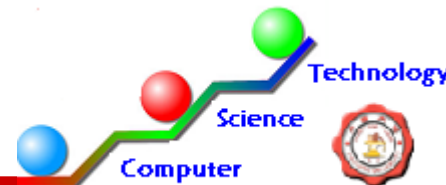
第六章 6.58



- 6.58 某16位机的加法器，由低至高位输出序号为 $F_0 \sim F_{15}$ ，4位一组分组，组内组间均采用先行进位结构，使用的器件为与-非、或-非、与或非、反相器等SSI门电路：
- (1) 按所用的逻辑结构写出第6位进位信号 C_6 的逻辑表达式；
 - (2) 画出与 C_6 有关部分的逻辑图，令原始输入为 A_i 、 B_i ($i=0 \sim 15$)、 C_0 ；
 - (3) 估算产生 C_6 所需的最长时间，设与-非、或-非、反相器等标准门电路的时延为 T ，与或非门的时延为 $1.5T$ 。

□ 解：该加法器逻辑框图如下所示：

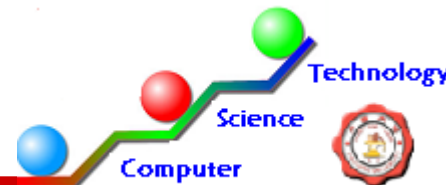
第六章 6.58 (1)



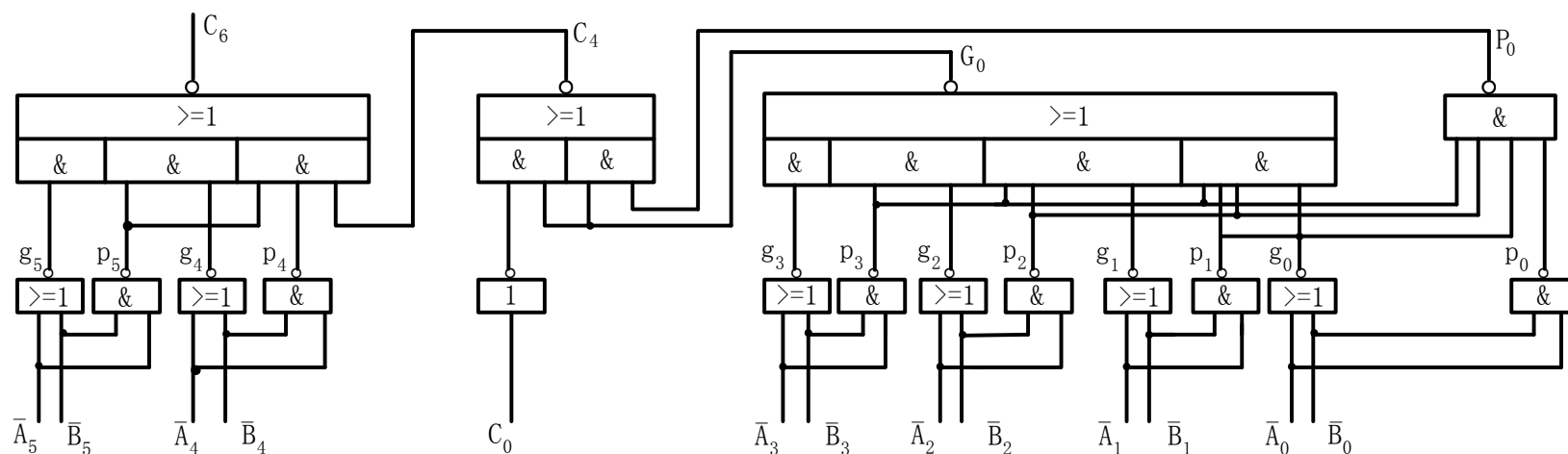
$$\begin{aligned}
 (1) \quad & /C_6 = /(g_5 + p_5 g_4 + p_5 p_4 C_4); \\
 & C_4 = /(/G_0 /P_0 + /G_0 /C_0); \\
 & /G_0 = /(g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0); \\
 & /P_0 = /(p_3 p_2 p_1 p_0); \\
 & g_i = /(/A_i + /B_i); \quad p_i = /(/A_i /B_i); \quad (i = 0 \sim 15)
 \end{aligned}$$

□ 注：该题逻辑设计使用了反演定律，以便减少门的级数。

第六章 6.58 (2)



(2) 与 C_6 有关部分的逻辑图如下:



(3) 产生 C_6 所需的最长路径为: $A_i, B_i \rightarrow g_i, p_i \rightarrow G_0, P_0 \rightarrow C_4 \rightarrow C_6$;
 与非(或非) 与或非 与或非 与或非

则产生 C_6 所需的最长时间为: $T_{\max} = T + 1.5T \times 3 = 5.5T$