

数字系统设计基础

第五次作业答案

1. 利用原码定点一位乘法计算 $X \times Y$ ，其中： $X = 0.1011, Y = -0.0110$ 。

答：

	部分积	乘数	移位
	0000	0110+	0000
(右移)	0000	1001	10
+	1011		
(右移)	0101	1001	10
+	1011		
	10000		
(右移)	1000	0100	010
结果: 1000010; 右移八位后: 0.01000010; 符号为负即: -0.01000010			

2. 利用补码定点一位乘法计算 $X \times Y$ ，其中： $X = 1.0111, Y = 0.1011$ 。

答：

	部分积	移位	乘数	附加位	部分积的加数
+	00.0000		01011	0	00.1001
	00.1001				[-X]补
右移	00.0100	1	0101	1	00.0000
+	00.0000				
右移	01	01	010	1	11.0111
+	11.0111				
	11.1001				
右移	101	101	01	0	00.1001
+	00.1001				
	100.0101				
右移	00.0010	1101	0	1	11.0111
+	11.0111				[+X]补
	11.1001				
结果:	11.1011101				

3. 已知 $X = 0.10100$, $Y = -0.10001$, 用原码不恢复余数除法 (加减交替法) 计算 X/Y 。

答:

$$x^* = [x]_{\text{原}} = [x]_{\text{补}} = x = 0.101\ 00$$

$$[y]_{\text{原}} = 1.100\ 01$$

$$y^* = 0.100\ 01$$

$$[-y^*]_{\text{补}} = 1.011\ 11$$

$$[y]_{\text{补}} = 1.011\ 11$$

$$[-y]_{\text{补}} = 0.100\ 01$$

$$q_0 = x_0 \oplus y_0 = 0 \oplus 1 = 1$$

$$x^* \div y^* = 1.001\ 01 \text{——溢出}$$

$$[x \div y]_{\text{原}} : \text{无定义}$$

$$x \div y = -1.001\ 01$$

$$r^* = 0.010\ 11 \times 2^{-5} = 0.000\ 000\ 101\ 1$$

计算过程如下:

原码加减交替法:

被除数(余数)	商
0.101 00	0.00 000
+ 1.011 11	试减, $+ [-y^*]_{\text{补}}$
0.000 11	
1 \leftarrow 0.001 10	1.
+ 1.011 11	$r > 0, + [-y^*]_{\text{补}}$
1.101 01	
1 \leftarrow 1.010 10	1.0
+ 0.100 01	$r < 0, + y^*$
1.110 11	
1 \leftarrow 1.101 10	1.00
+ 0.100 01	$r < 0, + y^*$
0.001 11	
1 \leftarrow 0.011 10	1.001
+ 1.011 11	$r > 0, + [-y^*]_{\text{补}}$
1.111 01	
1 \leftarrow 1.110 10	1.0 010
+ 0.100 01	$r < 0, + y^*$
0.010 11	1 \leftarrow 1.00 101
	$r > 0$, 结束

注:溢出,可停止运算,转溢出处理。

注:当 $x^* > y^*$ 时产生溢出,这种情况在第一步运算后判断 r 的正负时就可发现。此时数值位占领小数点左边的1位,原码无定义,但算法本身仍可正常运行。

4. 已知 $X = -0.1001$, $Y = 0.1101$, 用原码不恢复余数除法(加减交替法)求 X/Y 的商及余数。

答: $A=|X|=00.1001$, $B=|Y|=00.1101$, $-B=11.0011$, $C=|Q|=0.0000$ 。

步数	条件	操作	A	C	C_n
			00.1001	0.0000	
第 1 步		\leftarrow	01.0010		
		- B +	11.0011		
	$S_A = 0$		00.0101	0.0001	Q_1
第 2 步	$C_n = 1$	\leftarrow	00.1010		
		- B +	11.0011		
	$S_A = 1$		11.1101	0.0010	Q_2
第 3 步	$C_n = 0$	\leftarrow	11.1010		
		+ B +	00.1101		
	$S_A = 0$		00.0111	0.0101	Q_3
第 4 步	$C_n = 1$	\leftarrow	00.1110		
		- B +	11.0011		
	$S_A = 0$		00.0001	0.1011	Q_4

异号相除, 商为负数, 则 $X \div Y$ 的商为: -0.1011 。

余数符号与被除数的符号相同, 则 $X \div Y$ 的余数为: -0.0001×2^{-4} 。

5. 已知 $X = -0.1010$, $Y = 0.1101$, 用补码不恢复余数除法(加减交替法)计算 X/Y 。

【解答】 $[X]_{\text{补}} = 11.0110$, $[Y]_{\text{补}} = 00.1101$, $[-Y]_{\text{补}} = 11.0011$ (双符号位), 这里计算步骤略。最后经过计算得: $[X + Y]_{\text{补}} = [Q]_{\text{补}} = 1.0010$ (单符号位), 余数 $[R]_{\text{补}} = 1.1001 \times 2^{-4}$ (单符号位)。 $X/Y = -0.1110$, 余数 $= -0.0110 \times 2^{-4}$ 。

6. 设 $[x]_{\text{补}} = x_0.x_1x_2 \dots x_n$, 求证:

$$[x]_{\text{补}} = x_0.x_1x_2 \dots x_n = -x_0 + \sum_{i=1}^n x_i 2^{-i}.$$

【证】当 $x \geq 0$ 时, $x_0 = 0$,

$$[x]_{\text{补}} = 0.x_1x_2 \dots x_n = \sum_{i=1}^n x_i 2^{-i} = x$$

当 $x < 0$ 时, $x_0 = 1$,

$$[x]_{\text{补}} = 1.x_1x_2 \dots x_n = 2 + x$$

$$x = 1.x_1x_2 \dots x_n - 2 = -1 + 0.x_1x_2 \dots x_n = -1 + \sum_{i=1}^n x_i 2^{-i}$$

综合上述两种情况, 可得出: $x = -x_0 + \sum_{i=1}^n x_i 2^{-i}$

7. 设 $[x]_{\text{补}} = x_0.x_1x_2 \dots x_n$, 求证:

$$\left[\frac{x}{2}\right]_{\text{补}} = x_0.x_0x_1x_2 \dots x_n.$$

【证】因为 $x = -x_0 + \sum_{i=1}^n x_i 2^{-i}$, 所以

$$\frac{1}{2}x = -\frac{1}{2}x_0 + \frac{1}{2}\sum_{i=1}^n x_i 2^{-i} = -x_0 + \frac{1}{2}x_0 + \frac{1}{2}\sum_{i=1}^n x_i 2^{-i} = -x_0 + \sum_{i=1}^n x_i 2^{-(i+1)}$$

根据补码与真值的关系则有:

$$\left[\frac{1}{2}x\right]_{\text{补}} = x_0 \cdot x_0 x_1 x_2 \cdots x_n$$

由此可见, 如果要得到 $[2^{-i}x]_{\text{补}}$, 只要将 $[x]_{\text{补}}$ 连同符号位右移 i 位即可。

8. 全加器可由异或门及进位逻辑电路组成, 根据 $A \oplus B = \bar{A} \oplus \bar{B}$, 可以设计利用原变量或反变量进行运算的加法器。进而可以推测, 对已设计好的加法器, 用原变量运算和反变量运算都是一样的。这种说法对不对? 为什么?

答: 对已设计好的加法器, 用原变量运算和反变量运算都能得到正确的结果。换句话说, 用原变量设计好的加法器, 如果将所有的输入变量和输出变量均变反, 那么该加法器就能适用于反变量的运算。因为该加法器把逻辑输入信号都反相所产生的功能仍然在这个集合之中, 可以由真值表来说明:

$$\begin{aligned} S_i &= A_i \oplus B_i \oplus C_i & C_{i+1} &= A_i B_i + B_i C_i + C_i A_i \\ S_i &= \bar{A}_i \oplus \bar{B}_i \oplus \bar{C}_i & \bar{C}_{i+1} &= \bar{A}_i \bar{B}_i + \bar{B}_i \bar{C}_i + \bar{C}_i \bar{A}_i \end{aligned}$$

输 入						输 出			
A_1	B_1	C_1	\bar{A}_1	\bar{B}_1	\bar{C}_1	S_1	C_{i+1}	\bar{S}_1	\bar{C}_{i+1}
0	0	0	1	1	1	0	0	1	1
0	0	1	1	1	0	1	0	0	1
0	1	0	1	0	1	1	0	0	1
0	1	1	1	0	0	0	1	1	0
1	0	0	0	1	1	1	0	0	1
1	0	1	0	1	0	0	1	1	0
1	1	0	0	0	1	0	1	1	0
1	1	1	0	0	0	1	1	0	0

表:全加器真值表

9. 某加法器进位链小组信号为 $C_4 C_3 C_2 C_1$, 低位来的进位信号为 C_0 , 请分别按下述两种方式写出 $C_4 C_3 C_2 C_1$ 的逻辑表达式:

- (1) 串行进位方式;
- (2) 并行进位方式。

答: (1). 串行进位方式:

$$\begin{aligned} C_1 &= G_1 + P_1 C_0 & \text{其中: } G_1 &= A_1 B_1 & P_1 &= A_1 \oplus B_1 \\ C_2 &= G_2 + P_2 C_1 & G_2 &= A_2 B_2 & P_2 &= A_2 \oplus B_2 \\ C_3 &= G_3 + P_3 C_2 & G_3 &= A_3 B_3 & P_3 &= A_3 \oplus B_3 \\ C_4 &= G_4 + P_4 C_3 & G_4 &= A_4 B_4 & P_4 &= A_4 \oplus B_4 \end{aligned}$$

(2). 并行进位方式:

$$C_1 = G_1 + P_1 C_0$$

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

$$C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_0$$

$$C_4 = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 + P_4 P_3 P_2 P_1 C_0$$

其中 $G_i = G_i, P_i = P_i$, 表达式与串行进位方式相同。

10. 已知 $X = 0.a_1 a_2 a_3 a_4 a_5 a_6$ (a_i 为0或1), 讨论下列几种情况时 a_i 各取何值:

$$(1) X > \frac{1}{2}; (2) X \geq \frac{1}{8}; (3) \frac{1}{4} \geq X > \frac{1}{16}.$$

答:

(1) 若要 $X > 1/2$, 只要 $a_1 = 1$, $a_2 \sim a_6$ 不全为0即可 ($a_2 \text{ or } a_3 \text{ or } a_4 \text{ or } a_5 \text{ or } a_6 = 1$);

(2) 若要 $X \geq 1/8$, 只要 $a_1 \sim a_3$ 不全为0即可 ($a_1 \text{ or } a_2 \text{ or } a_3 = 1$), $a_4 \sim a_6$ 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 ($a_5 \text{ or } a_6 = 1$); 若 $a_3 = 1$, 则 $a_4 \sim a_6$ 可任取0或1;

当 $a_2 = 1$ 时, $a_3 \sim a_6$ 取0。

11. 设机器字长为16位 (含1位符号位), 若一次移位需1us, 一次加法需1us, 试问原码一位乘、补码一位乘、原码加减交替除和补码加减交替除法最多各需多少时间?

答:

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

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

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

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

12. 画出实现 Booth 算法的运算器框图。

要求如下:

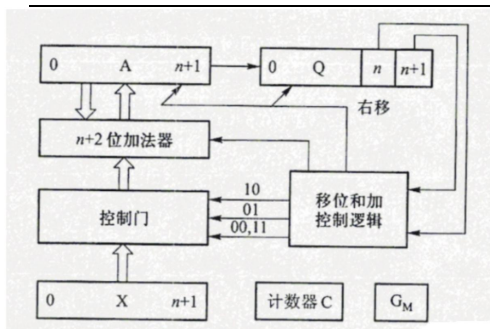
(1) 寄存器和全加器均用方框表示, 指出寄存器和全加器的位数。

(2) 说明加和移位的次数。

(3) 详细画出最低位全加器的输入电路。

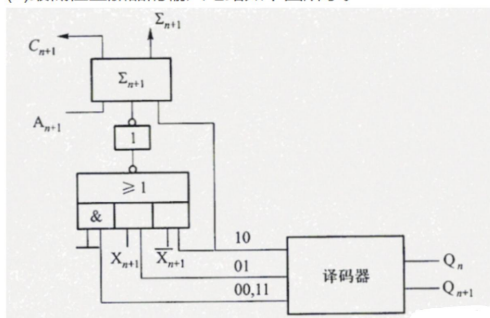
(4) 描述Booth算法重复加和移位的过程。

答: (1) 实现补码Booth算法的运算器框图如下图所示。图中全加器和寄存器均为 $n+2$ 位, 其中A寄存器存放部分积, 含两位符号位, 初态为“0”; X寄存器存放被乘数的补码, 含两位符号位; Q寄存器存放乘数的补码, 含1位符号位, 最末位为附加位, 初态为“0”。最终乘积的高位在寄存器A中, 乘积的低位在Q寄存器中。计数器C用来控制移位次数, 判断乘法是否结束。 G_M 为乘法标记。



(2) Booth算法共做n次移位，最多做n+1次加法。

(3) 最低位全加器的输入电路如下图所示。



(4) 加的过程受Q寄存器末两位控制，当它们同时为0(记做 $\overline{Q_n} \overline{Q_{n+1}}$)或同时为1(记做 $Q_n Q_{n+1}$)时，部分积(在A中)不变;当末两位为01(记做 $\overline{Q_n} Q_{n+1}$)时，部分积加上被乘数(记做 $A+X$);

当末两位为10(记做 $Q_n \overline{Q_{n+1}}$)时，部分积减去被乘数，即与求补后的被乘数相加(记做 $A + \overline{X} + 1$)。

则Booth 算法的重复加过程可描述为:

$$(\overline{Q_n} \overline{Q_{n+1}} + Q_n Q_{n+1})A + \overline{Q_n} Q_{n+1}(A + X) + Q_n \overline{Q_{n+1}}(A + \overline{X} + 1) \rightarrow A$$

移位时A、Q两个寄存器串接(A//Q)，一起右移一位(算术移位)，记 $L(A//Q) \rightarrow R(A//Q)$ 。

13. 在三进制数系统中三个数字：0、1和2。下图定义了一个三进制半加法器。试设计一个电路，用二进制编码的信号来实现这个半加法器，例如可以将每个三进制数用2个比特表示。设 $A = a_1 a_0$, $B = b_1 b_0$, $\text{Sum} = s_1 s_0$ ，注意进位 Carry 只是一个二进制信号。使用以下编码方式：00 = (0)₃, 01 = (1)₃, 10 = (2)₃。你需要最小化电路的成本。

AB	Carry	Sum
00	0	0
01	0	1
02	0	2
10	0	1
11	0	2
12	1	0
20	0	2
21	1	0
22	1	1

答案：三进制半加法器可以使用二进制编码信号定义如下

A		B		Carry	Sum	
a_1	a_0	b_1	b_0	c_{out}	s_1	s_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
1	0	0	0	0	1	0
1	0	0	1	1	0	0
1	0	1	0	1	0	1

我们不用考虑 16 个估值中的其余 7 个，其中 $a_0 = a_1 = 1$ 或 $b_0 = b_1 = 1$ ，则最小代价表达式为

$$\begin{aligned}
 c_{out} &= a_0 b_1 + a_1 b_1 + a_1 b_0 \\
 s_1 &= a_0 b_0 + \bar{a}_1 \bar{a}_0 b_1 + a_1 \bar{b}_1 \bar{b}_0 \\
 s_0 &= a_1 b_1 + \bar{a}_1 \bar{a}_0 b_0 + a_0 \bar{b}_1 \bar{b}_0
 \end{aligned}$$

14. 试用上题中描述的方法设计一个三进制全加法器电路。

答案：三进制全加法器由真值表定义

c_{in}	A	B	c_{out}	Sum
0	0	0	0	0
0	0	1	0	1
0	0	2	0	2
0	1	0	0	1
0	1	1	0	2
0	1	2	1	0
0	2	0	0	2
0	2	1	1	0
0	2	2	1	1
1	0	0	0	1
1	0	1	0	2
1	0	2	1	0
1	1	0	0	2
1	1	1	1	0
1	1	2	1	1
1	2	0	1	0
1	2	1	1	1
1	2	2	1	2

我们不用考虑 32 个估值中的其余 14 个，其中 $a_0 = a_1 = 1$ 或 $b_0 = b_1 = 1$ ，则最小代价表达式为

$$c_{out} = a_0b_1 + a_1b_0 + a_1b_1 + a_1c_{in} + b_1c_{in} + a_0b_0c_{in}$$

$$s_1 = a_0b_0\bar{c}_{in} + \bar{a}_1\bar{a}_0b_1\bar{c}_{in} + a_1\bar{b}_1\bar{b}_0\bar{c}_{in} + a_1b_1c_{in} + \bar{a}_1\bar{a}_0b_0c_{in} + a_0\bar{b}_1\bar{b}_0c_{in}$$

$$s_0 = a_1b_1\bar{c}_{in} + \bar{a}_1\bar{a}_0b_0\bar{c}_{in} + a_0\bar{b}_1\bar{b}_0\bar{c}_{in} + a_1b_0c_{in} + a_0b_1c_{in} + \bar{a}_1\bar{a}_0\bar{b}_1\bar{b}_0c_{in}$$