

计算机组织结构

6 二进制编码的十进制数运算

刘博涵

2022年10月13日



南京大學
NANJING UNIVERSITY

回顾：二进制编码的十进制数表示

- 使用原因
 - 精度限制
 - 转换成本高
- 编码方式：自然BCD码 (NBCD, 8421 码)
 - 0 ~ 9: 0000 ~ 1001
 - 符号
 - 正: 1100 / 0
 - 负: 1101 / 1



回顾：补码表示整数的加减法

- 加法

- $[X+Y]_c = [X]_c + [Y]_c \pmod{2^n}$
- 溢出判定

- 减法

- $[X-Y]_c = [X]_c + [-Y]_c \pmod{2^n}$
- 硬件实现



加法

$$25 + 31 = 56$$

$$\begin{array}{r} 0010\ 0101 \\ +\ 0011\ 0001 \\ \hline 0101\ 0110 \end{array}$$

$$28 + 39 = 67$$

$$\begin{array}{r} 0010\ 1000 \\ +\ 0011\ 1001 \\ \hline 0101\ 0001 \\ +\ \quad \quad 1 \\ \hline \quad \quad 0110 \\ \hline 0110\ 0111 \end{array}$$

$$98 + 79 = 177$$

$$\begin{array}{r} 1001\ 1000 \\ +\ 0111\ 1001 \\ \hline 0000\ 0001 \\ +\ 1\quad \quad 1 \\ \hline \quad \quad 0110\ 0110 \\ \hline 0001\ 0111\ 0111 \end{array}$$

$$25 + 39 = 64$$

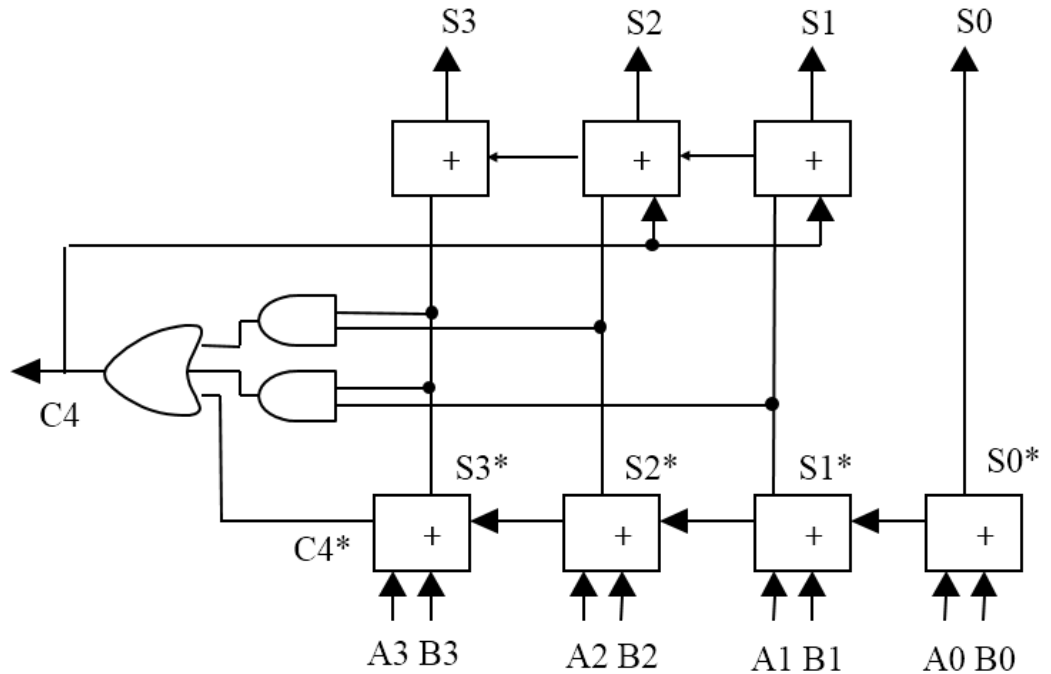
$$\begin{array}{r} 0010\ 0101 \\ +\ 0011\ 1001 \\ \hline 0101\ 1110 \\ +\ \quad \quad 0110 \\ \hline 0110\ 0100 \end{array}$$



加法 (续)

- 硬件实现：当值在[10,19]范围内时，需要对结果进行调整

S_3^*	S_2^*	S_1^*	S_0^*	C_4^*
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1



$$C_4 = (S_1^* \cdot S_3^*) + (S_2^* \cdot S_3^*) + C_4^*$$

减法

- 思路：参照补码减法，避免借位

$$\begin{aligned} N_1 - N_2 &= N_1 + (10^n - N_2) - 10^n \\ &= N_1 + (99 \dots 9 - N_2 + 1) - 10^n \end{aligned}$$

“反转” 每一个数字，最后一位加1

- “反转” 数字

- 按位反转，并添加 “1010”
- 添加 “0110” ，并按位反转

- 结果调整

- 如果有进位，舍弃进位
- 如果没有进位，对结果按位反转后加1，并将结果符号设为负

数字	反转	数字	反转
0000	1001	0001	1000
0010	0111	0011	0110
0100	0101	0101	0100
0110	0011	0111	0010
1000	0001	1001	0000



减法 (续)

$$\begin{aligned} & 309 - 125 \\ = & 309 + 874 + 1 - 1000 \\ = & 184 \end{aligned}$$

	0011	0000	1001
+	1000	0111	0101
<hr/>			
	1011	0111	1110
+	0110		0110
<hr/>			
1	0001	1000	0100

$$\begin{aligned} & 125 - 309 \\ = & 125 + 690 + 1 - 1000 \\ = & 816 - 999 - 1 \\ = & -184 \end{aligned}$$

$$\begin{array}{r}
 \begin{array}{rrr}
 & 0001 & 0010 & 0101 \\
 + & 0110 & 1001 & 0001 \\
 \hline
 & 0111 & 1011 & 0110 \\
 + & & 0110 & \\
 \hline
 & 1000 & 0001 & 0110 \\
 & & \downarrow & \\
 - & 0001 & 1000 & 0100
 \end{array}
 \end{array}$$



总结

- 加法
 - 结果调整：进位引起加0110
 - 硬件实现
- 减法
 - 思路：参照补码进行数字反转，以避免借位
 - 反转数字
 - 结果调整



谢谢

bohanliu@nju.edu.cn



南京大學
NANJING UNIVERSITY