计算机组成原理

第五讲

张展

哈尔滨工业大学计算学部

容错与移动计算研究中心

第6章 计算机的运算方法

乘法器性能提升 6.3

- 核心算法：**n**个部分积累加
- **Booth**一位乘 → **Booth**两位乘
  - 一位乘法：n个全加器，n^2个全加器时延，面积小 (Intel 8086)
  - 两位乘法： 减少相加数，速度更快，增加额外电路
- 斜向进位阵列乘法器 → 华莱士树
  - 斜向进位：（n^2-n）个全加器，n级全加器时延，面积大
  - 华莱士树： 更多全加器，$\log_2 n$ 级全加器时延，面积更大
- 主流乘法器
  - 二位**booth**算法 + 华莱士树 + 流水

四、除法运算

1. 分析笔算除法

$x = -0.1011 \quad y = 0.1101 \quad$ 求 $x \div y$

$$
\begin{array}{r|l}
 & \;0.1101 \\
\hline
0.1101 & 0.10110 \\
 & \underline{0.01101} \\
 & 0.010010 \\
 & \underline{0.001101} \\
 & 0.00010100 \\
 & \underline{0.00001101} \\
 & 0.00000111
\end{array}
$$

✓商符单独处理

? 心算上商

? 余数不动低位补“0”
减右移一位的除数

? 上商位置不固定

$x \div y = -0.1101$ 商符心算求得

余数 0.00000111

2. 笔算除法和机器除法的比较 6.3

笔算除法	机器除法
商符单独处理	符号位异或形成
心算上商	$\|x\|-\|y\|>0$ 上商 1
	$\|x\|-\|y\|<0$ 上商 0
余数 不动 低位补“0” 减右移一位 的除数	余数 左移一位 低位补“0” 减 除数
2 倍字长加法器	1 倍字长加法器
上商位置 不固定	在寄存器 最末位上商

3. 原码除法 6.3

以小数为例

$$[x]_{原} = x_0 .\, x_1 x_2 \cdots \ x_n$$

$$[y]_{原} = y_0 .\, y_1 y_2 \cdots \ y_n$$

$$\left[\frac{x}{y}\right]_{原} = (x_0 \oplus y_0) .\, \frac{x^*}{y^*}$$

式中 $x^* = 0.\, x_1 x_2 \cdots \ x_n$ 为 x 的绝对值

$y^* = 0.\, y_1 y_2 \cdots \ y_n$ 为 y 的绝对值

商的符号位单独处理 $x_0 \oplus y_0$

数值部分为绝对值相除 $\frac{x^*}{y^*}$

约定 小数定点除法 $x^* < y^*$ 整数定点除法 $x^* > y^*$

被除数不等于 0

除数不能为 0

(1) 恢复余数法

例6.24 $x = -0.1011$ $y = -0.1101$ 求 $[\frac{x}{y}]_原$

解：$[x]_原 = 1.1011$ $[y]_原 = 1.1101$ $[y^*]_补 = 0.1101$ $[-y^*]_补 = 1.0011$

① $x_0 \oplus y_0 = 1 \oplus 1 = 0$

②

被除数（余数）	商	说　明
$\boxed{0.1011}$	0.0000	
+ 1.0011		$+[-y^*]_补$
1.1110	0	余数为负，上商 0
+ 0.1101		恢复余数 $+[y^*]_补$
$\boxed{0.1011}$	0	恢复后的余数
逻辑左移 1.0110	0	←1
+ 1.0011		$+[-y^*]_补$
0.1001	01	余数为正，上商 1
逻辑左移 1.0010	01	←1
+ 1.0011		$+[-y^*]_补$

被除数（余数）	商	说　明
0 . 0 1 0 1	0 1 1	余数为正，上商 1
逻辑左移 0 . 1 0 1 0	0 1 1	← 1
+ 1 . 0 0 1 1		$+[-y^*]_{补}$
1 . 1 1 0 1	0 1 1 0	余数为负，上商 0
+ 0 . 1 1 0 1		恢复余数 $+[y^*]_{补}$
0 . 1 0 1 0	0 1 1 0	恢复后的余数
逻辑左移 1 . 0 1 0 0	0 1 1 0	← 1
+ 1 . 0 0 1 1		$+[-y^*]_{补}$
0 . 0 1 1 1	0 1 1 0 1	余数为正，上商 1

$$\frac{x^*}{y^*} = 0.1101$$

$$\therefore \left[\frac{x}{y}\right]_{原} = 0.1101$$

上商 5 次

第一次上商判溢出

移 4 次

余数为正　上商 1

余数为负　上商 0，恢复余数

(2) 不恢复余数法（加减交替法） 6.3

- 恢复余数法运算规则

余数 $R_i>0$ 上商 “1”， $2R_i-y^*$

余数 $R_i<0$ 上商 “0”， R_i+y^* 恢复余数

$$2(R_i+y^*)-y^*=2R_i+y^*$$

- 不恢复余数法运算规则

上商 “1” $2R_i-y^*$

上商 “0” $2R_i+y^*$

加减交替

例6.25 $x = -0.1011$ $y = -0.1101$ 求 $[\frac{x}{y}]_{原}$ **6.3**

解：

	0.1011	0.0000	
	+1.0011		$+[-y^*]_{补}$
	1.1110	0	余数为负，上商 0
逻辑左移	1.1100	0	$\leftarrow$ 1
	+0.1101		$+[y^*]_{补}$
	0.1001	01	余数为正，上商 1
逻辑左移	1.0010	01	$\leftarrow$ 1
	+1.0011		$+[-y^*]_{补}$
	0.0101	011	余数为正，上商 1
逻辑左移	0.1010	011	$\leftarrow$ 1
	+1.0011		$+[-y^*]_{补}$
	1.1101	0110	余数为负，上商 0
逻辑左移	1.1010	0110	$\leftarrow$ 1
	+0.1101		$+[y^*]_{补}$
	0.0111	01101	余数为正，上商 1

$[x]_{原} = 1.1011$

$[y]_{原} = 1.1101$

$[x^*]_{补} = 0.1011$

$[y^*]_{补} = 0.1101$

$[-y^*]_{补} = 1.0011$

例6.25 结果

① $x_0 \oplus y_0 = 1 \oplus 1 = 0$

② $\dfrac{x^*}{y^*} = 0.1101$

$\therefore \; [\dfrac{x}{y}]_{原} = 0.1101$

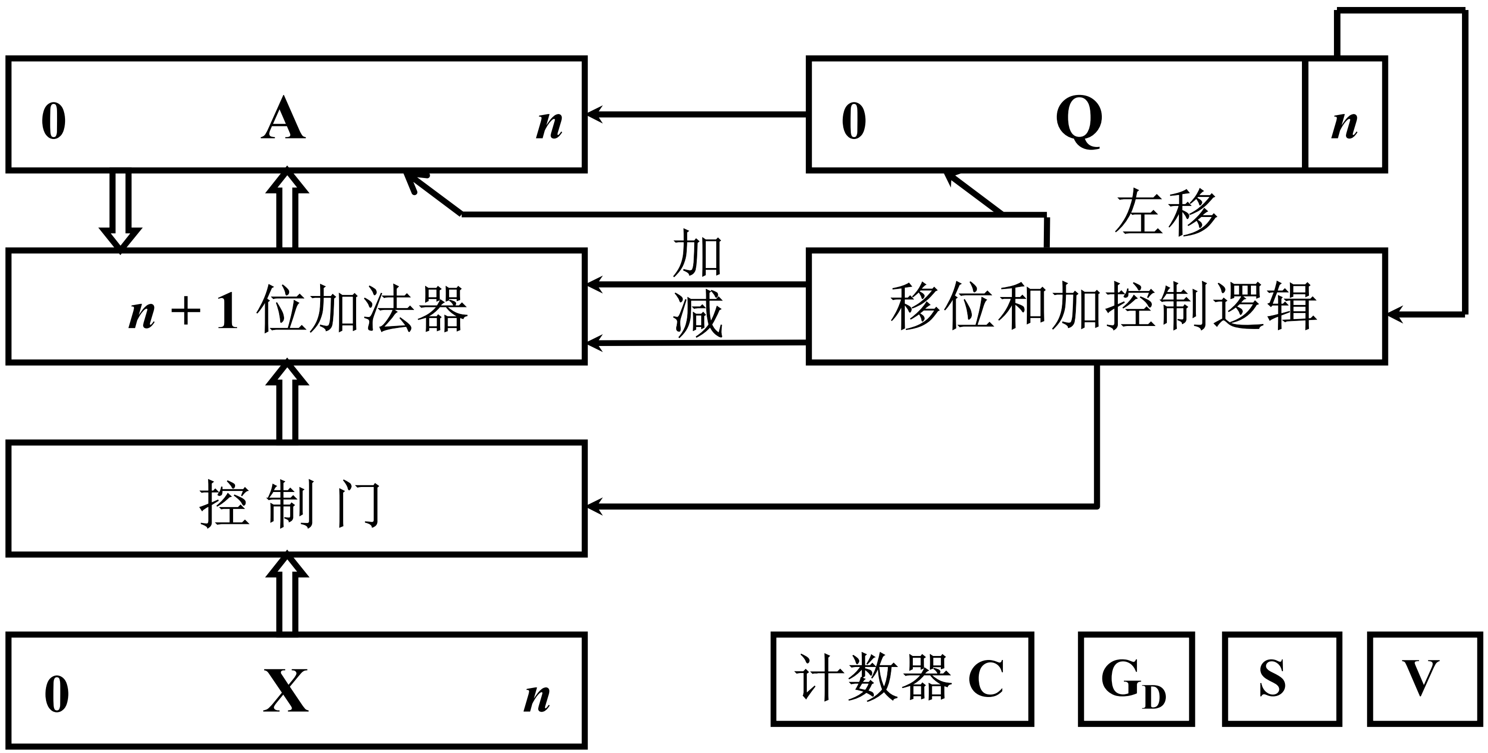
特点 上商 $n+1$ 次

第一次上商判溢出

移 n 次，加 $n+1$ 次

用移位的次数判断除法是否结束

(3) 原码加减交替除法硬件配置



A、X、Q 均 n+1 位

用 Q_n 控制加减交替

6.4 浮点四则运算

一、浮点加减运算

$$x = S_x \cdot 2^{j_x} \qquad y = S_y \cdot 2^{j_y}$$

1. 对阶

(1) 求阶差

$$\Delta j = j_x - j_y = \begin{cases} = 0 & j_x = j_y \quad \text{已对齐} \\ > 0 & j_x > j_y \begin{cases} x \text{ 向 } y \text{ 看齐} & S_x \leftarrow 1,\ j_x - 1 \\ y \text{ 向 } x \text{ 看齐} & \checkmark\ S_y \rightarrow 1,\ j_y + 1 \end{cases} \\ < 0 & j_x < j_y \begin{cases} x \text{ 向 } y \text{ 看齐} & \checkmark\ S_x \rightarrow 1,\ j_x + 1 \\ y \text{ 向 } x \text{ 看齐} & S_y \leftarrow 1,\ j_y - 1 \end{cases} \end{cases}$$

(2) 对阶原则

小阶向大阶看齐

例如 $x = 0.1101 \times 2^{01}$ $y = (-0.1010) \times 2^{11}$ 6.4

求 $x+y$

解： $[x]_补 = 00, 01; 00.1101$ $[y]_补 = 00, 11; 11.0110$

1. 对阶

① 求阶差 $[\Delta j]_补 = [j_x]_补 - [j_y]_补$

$$\begin{array}{r} = 00, 01 \\ +\quad 11, 01 \\ \hline 11, 10 \end{array}$$

阶差为负（-2） $\therefore S_x \longrightarrow 2$ $j_x + 2$

② 对阶 $[x]_补' = 00, 11; 00.0011$

2. 尾数求和

$$\begin{array}{rr} [S_x]_补' & = 00.0011 \\ +\quad [S_y]_补 & = 11.0110 \\ \hline & 11.1001 \end{array}$$

对阶后的$[S_x]_补'$

$\therefore [x+y]_补 = 00, 11; 11.1001$

3. 规格化

(1) 规格化数的定义

$$r = 2 \qquad \frac{1}{2} \le |S| < 1$$

(2) 规格化数的判断

$S>0$	规格化形式	$S<0$	规格化形式
真值	$0.1\times\times\cdots\times$	真值	$-0.1\times\times\cdots\times$
原码	$0.\boxed{1}\times\times\cdots\times$	原码	$1.\boxed{1}\times\times\cdots\times$
补码	$\boxed{0.1}\times\times\cdots\times$	补码	$\boxed{1.0}\times\times\cdots\times$
反码	$0.1\times\times\cdots\times$	反码	$1.0\times\times\cdots\times$

原码　不论正数、负数，第一数位为1

补码　符号位和第 一数位不同

特例

$$S = -\frac{1}{2} = -0.100 \cdots 0$$

$$[S]_{原} = 1.100 \cdots 0$$

$$[S]_{补} = \boxed{1.1}00 \cdots 0$$

$\therefore$ $[-\frac{1}{2}]_{补}$ 不是规格化的数

$$S = -1$$

$$[S]_{补} = \boxed{1.0}00 \cdots 0$$

$\therefore$ $[-1]_{补}$ 是规格化的数

(3) 左规

尾数左移一位，阶码减 1，直到数符和第一数位不同为止

上例 $[x+y]_{补} = 00, 11; 11.1001$

左规后 $[x+y]_{补} = 00, 10; 11.0010$

$$\therefore x + y = (-0.1110)\times 2^{10}$$

(4) 右规

当 尾数溢出（ >1）时，需 右规

即尾数出现 01. ×× ⋯× 或 10. ×× ⋯× 时

尾数右移一位，阶码加 1

例6.27 $x = 0.1101 \times 2^{10}$ $y = 0.1011 \times 2^{01}$ **6.4**

求 $x+y$（除阶符、数符外，阶码取 3 位，尾数取 6 位）

解：

$$[x]_补 = 00, 010; 00.110100$$
$$[y]_补 = 00, 001; 00.101100$$

① 对阶

$$[\Delta j]_补 = [j_x]_补 - [j_y]_补 \quad \begin{array}{r} = \ 00, 010 \\ + \ 11, 111 \\ \hline \boxed{1}00, 001 \end{array}$$

阶差为 +1 $\quad \therefore S_y \longrightarrow 1, \ j_y+1$

$$\therefore [y]_{补'} = 00, 010; 00.010110$$

② 尾数求和

$$\begin{array}{rr} & [S_x]_补 = 00.110100 \\ + & [S_y]_{补'} = 00.010110 \\ \hline & 01.001010 \end{array}$$

对阶后的$[S_y]_{补'}$

尾数溢出需右规

③ 右规

$$[x+y]_{补} = 00, 010; 01.\ 001010$$

右规后

$$[x+y]_{补} = 00, 011; 00.\ 100101$$

$$\therefore\ x+y = 0.\ 100101 \times 2^{11}$$

4. 舍入

在 对阶 和 右规 过程中，可能出现 尾数末位丢失

引起误差，需考虑舍入

(1) 0 舍 1 入法

(2) 恒置 “1” 法

例 6.28 $x = (-\frac{5}{8}) \times 2^{-5}$ $y = (\frac{7}{8}) \times 2^{-4}$

求 $x-y$（除阶符、数符外，阶码取 3 位，尾数取 6 位）

解： $x = (-0.101000) \times 2^{-101}$ $y = (0.111000) \times 2^{-100}$

$[x]_{补} = 11,011;\ 11.011000$ $[y]_{补} = 11,100;\ 00.111000$

① 对阶

$$[\Delta j]_{补} = [j_x]_{补} - [j_y]_{补} \quad \begin{array}{r} = \ 11,011 \\ +\ 00,100 \\ \hline 11,111 \end{array}$$

阶差为 -1 $\therefore S_x \longrightarrow 1,\ \ j_x+1$

$\therefore [x]_{补'} = 11,100;\ 11.101100$

② 尾数求和

$$
\begin{array}{rl}
[S_x]_{补'} = & 11.101100 \\
+\ [-S_y]_{补} = & 11.001000 \\
\hline
& 110.110100
\end{array}
$$

③ 右规

$$[x-y]_{补} = 11, 100; 10.110100$$

右规后

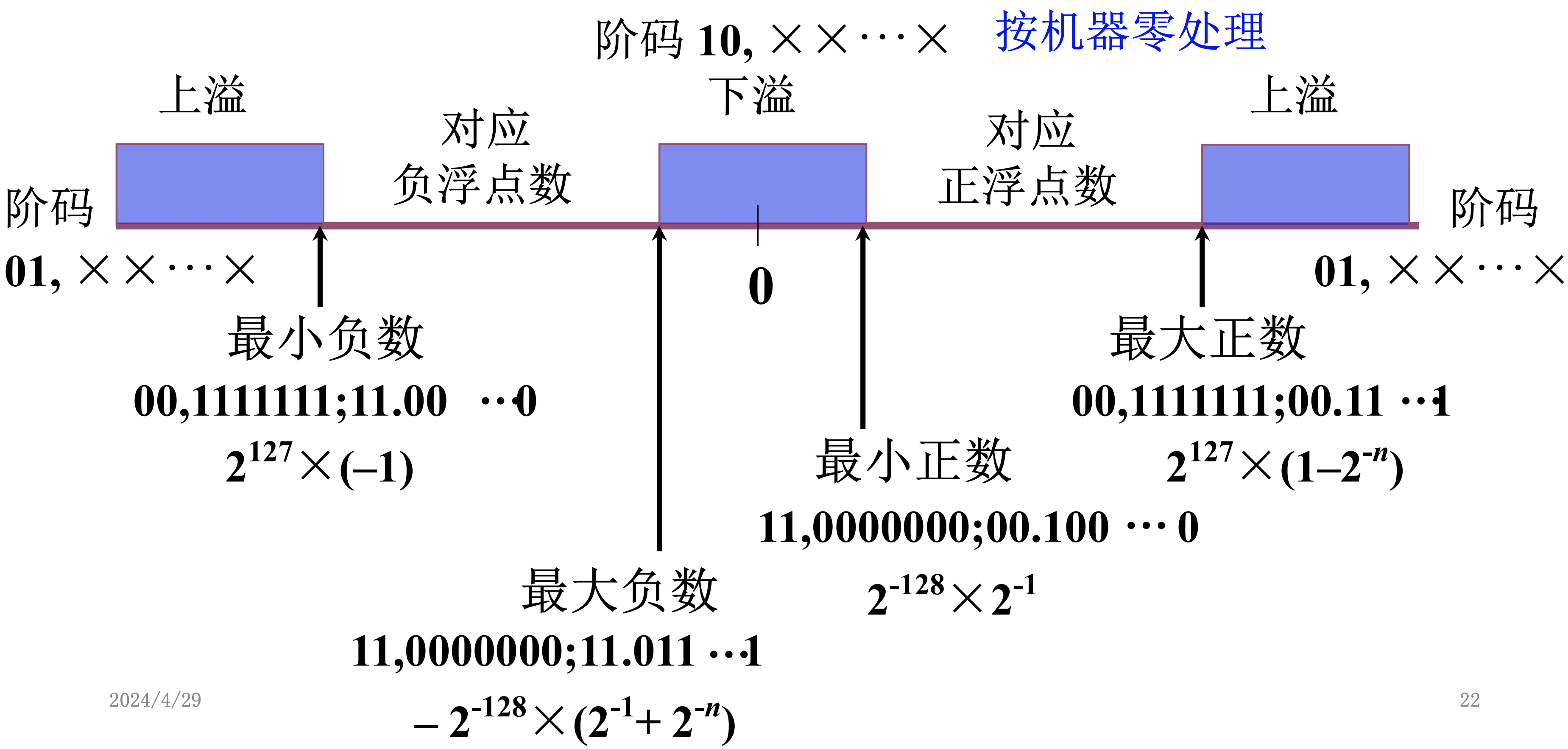
$$[x-y]_{补} = 11, 101; 11.011010$$

$$\therefore\ x-y = (-0.100110) \times 2^{-11}$$

$$= (-\frac{19}{32}) \times 2^{-3}$$

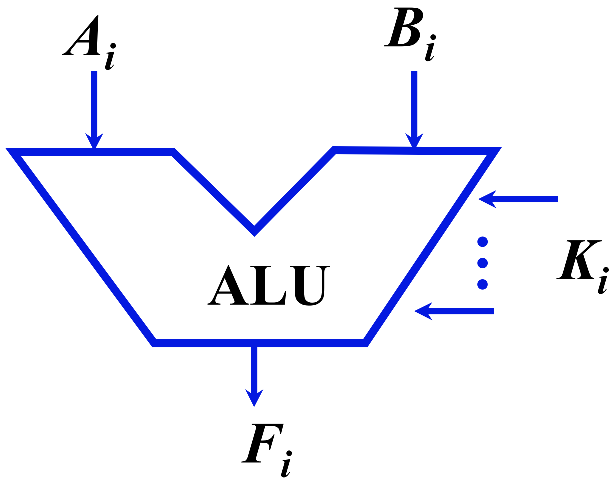
5. 溢出判断

设机器数为补码，尾数为 规格化形式，并假设阶符取 2 位，阶码的数值部分取 7 位，数符取 2 位，尾数取 n 位，则该 补码 在数轴上的表示为



6.5 算术逻辑单元

一、ALU 电路



组合逻辑电路

K_i 不同取值

F_i 不同

四位 ALU　74181

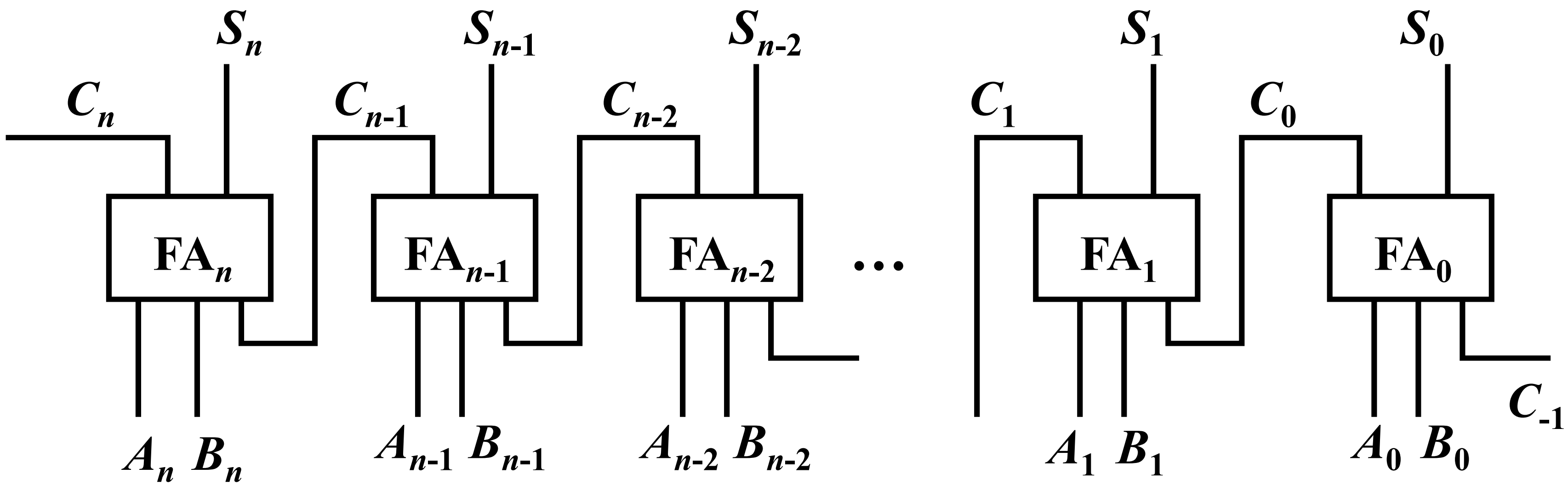
$M = 0$　　算术运算

$M = 1$　　逻辑运算

$S_3 \sim S_0$　　不同取值，可做不同运算

二、快速进位链

1. 并行加法器



$$S_i = \overline{A_i}\,\overline{B_i}\,C_{i-1} + \overline{A_i}\,B_i\,\overline{C_{i-1}} + A_i\,\overline{B_i}\,\overline{C_{i-1}} + A_i\,B_i\,C_{i-1}$$

$$C_i = \overline{A_i}\,B_i\,C_{i-1} + A_i\,\overline{B_i}\,C_{i-1} + A_i\,B_i\,\overline{C_{i-1}} + A_i\,B_i\,C_{i-1}$$

$$= A_i\,B_i + (A_i + B_i)C_{i-1}$$

$d_i = A_i\,B_i$　本地进位　　$t_i = A_i + B_i$　传送条件

则 $C_i = d_i + t_i C_{i-1}$

2. 串行进位链

进位链　　　　传送进位的电路

串行进位链　　进位串行传送

以 4 位全加器为例，每一位的进位表达式为

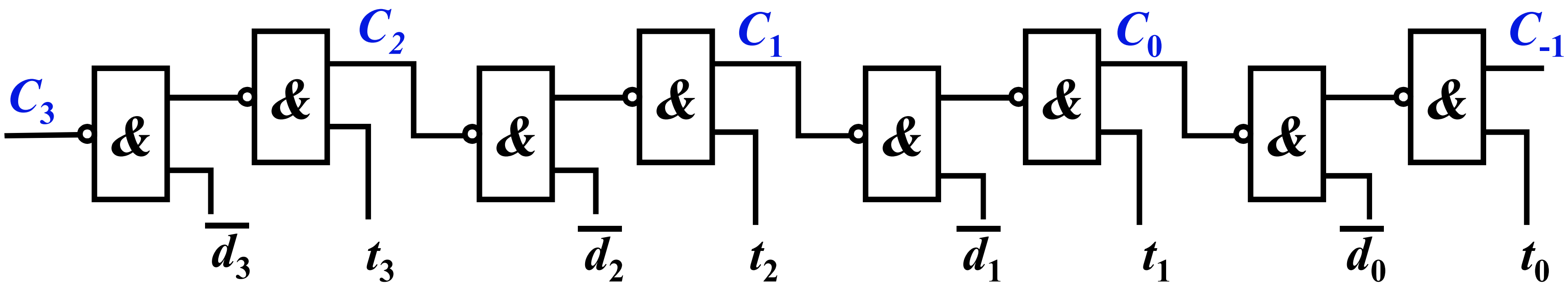
$$C_0 = d_0 + t_0 C_{-1} = \overline{\overline{d_0} \cdot \overline{t_0 C_{-1}}}$$

$$C_1 = d_1 + t_1 C_0$$

$$C_2 = d_2 + t_2 C_1$$

$$C_3 = d_3 + t_3 C_2$$

设与非门的级延迟时间为t_y



4位 全加器产生进位的全部时间为 $8t_y$

n 位全加器产生进位的全部时间为 $2nt_y$

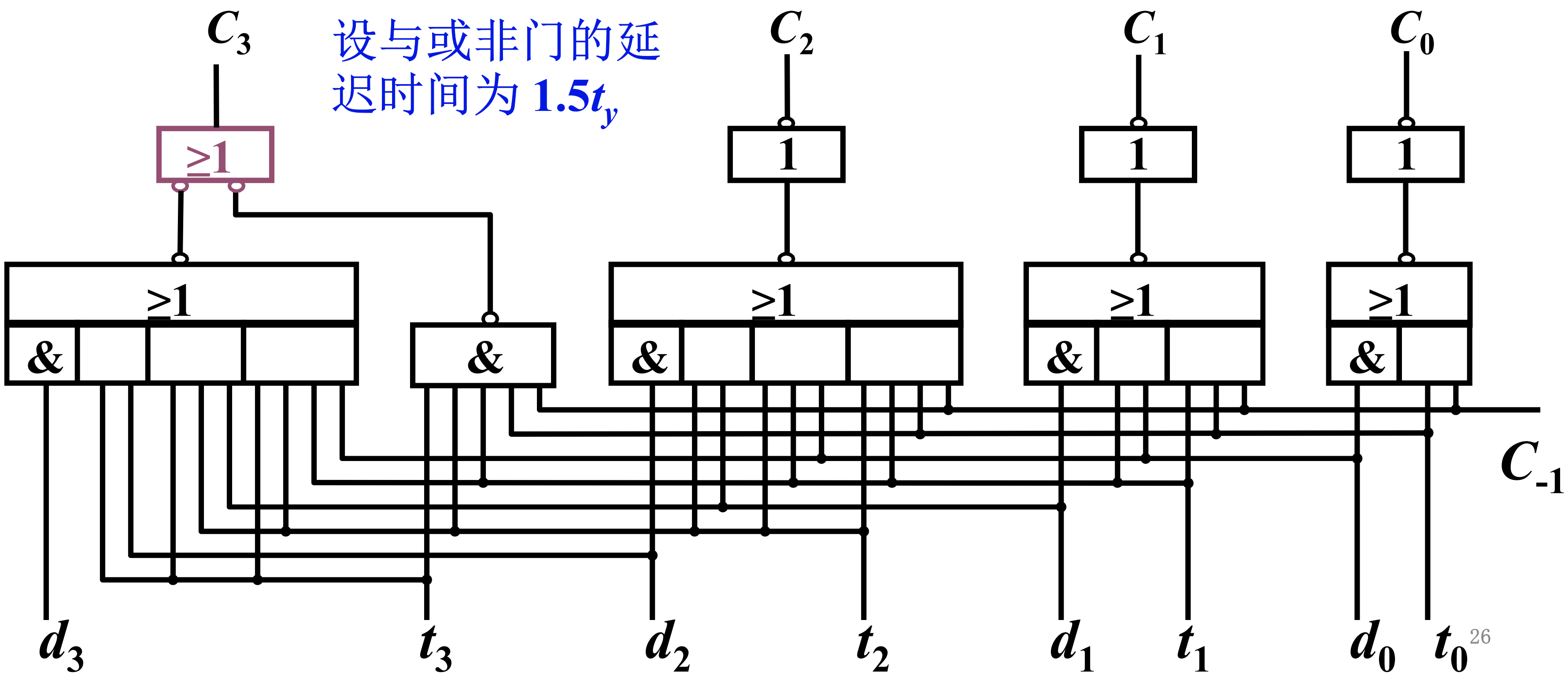
3. 并行进位链（先行进位，跳跃进位）

n 位加法器的进位同时产生　　以 4 位加法器为例

$$C_0 = d_0 + t_0 C_{-1}$$
$$C_1 = d_1 + t_1 C_0 = d_1 + t_1 d_0 + t_1 t_0 C_{-1}$$
$$C_2 = d_2 + t_2 C_1 = d_2 + t_2 d_1 + t_2 t_1 d_0 + t_2 t_1 t_0 C_{-1}$$
$$C_3 = d_3 + t_3 C_2 = d_3 + t_3 d_2 + t_3 t_2 d_1 + t_3 t_2 t_1 d_0 + t_3 t_2 t_1 t_0 C_{-1}$$

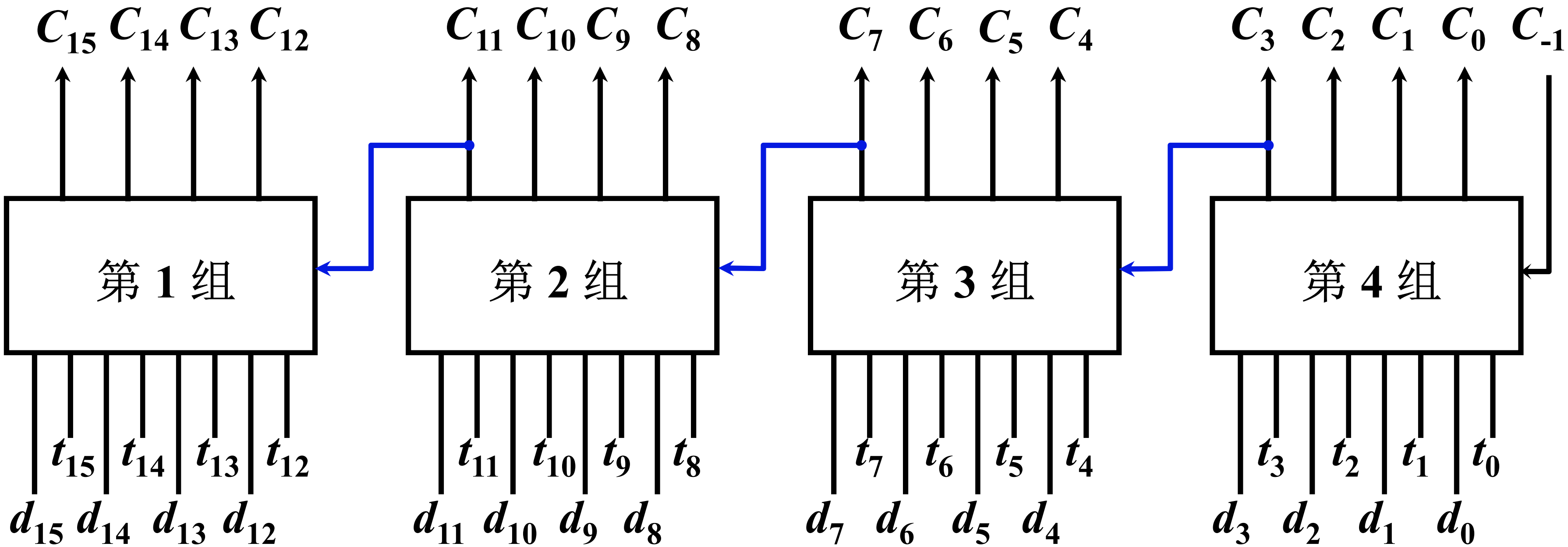
当 $d_i t_i$ 形成后，只需 $2.5t_y$ 产生全部进位

设与或非门的延迟时间为 $1.5t_y$



(1) 单重分组跳跃进位链

n 位全加器分若干小组，小组中的进位同时产生，
小组与小组之间采用串行进位　　以 $n = 16$ 为例



当 d_i、t_i 形成后　经 $2.5\ t_y$　　产生 $C_3 \sim C_0$

$5\ t_y$　　产生 $C_7 \sim C_4$

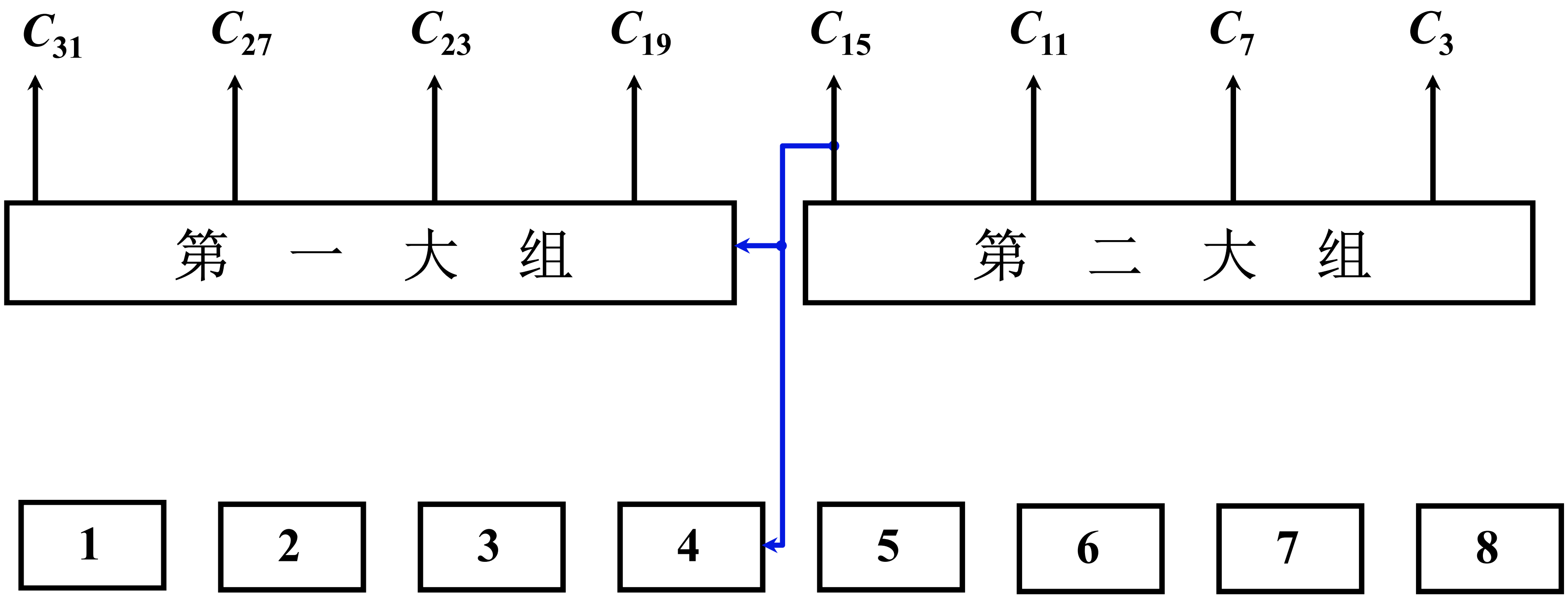
$7.5\ t_y$　　产生 $C_{11} \sim C_8$

$10\ t_y$　　产生 $C_{15} \sim C_{12}$

(2) 双重分组跳跃进位链 6.5

n 位全加器分若干大组，大组中又包含若干小组。每个大组中小组的最高位进位同时产生。大组与大组之间采用串行进位。

以 $n = 32$ 为例



(3) 双重分组跳跃进位链 大组进位分析

以第 8 小组为例

$$C_3 = d_3 + t_3C_2 = \underbrace{d_3 + t_3d_2 + t_3t_2d_1 + t_3t_2t_1d_0}_{D_8} + \underbrace{t_3t_2t_1t_0}_{T_8}C_{-1}$$

$$= D_8 + T_8C_{-1}$$

D_8 小组的本地进位 与外来进位无关

T_8 小组的传送条件 与外来进位无关 传递外来进位

同理 第 7 小组 $C_7 = D_7 + T_7C_3$

第 6 小组 $C_{11} = D_6 + T_6C_7$

第 5 小组 $C_{15} = D_5 + T_5C_{11}$

进一步展开得

$$C_3 = D_8 + T_8C_{-1}$$

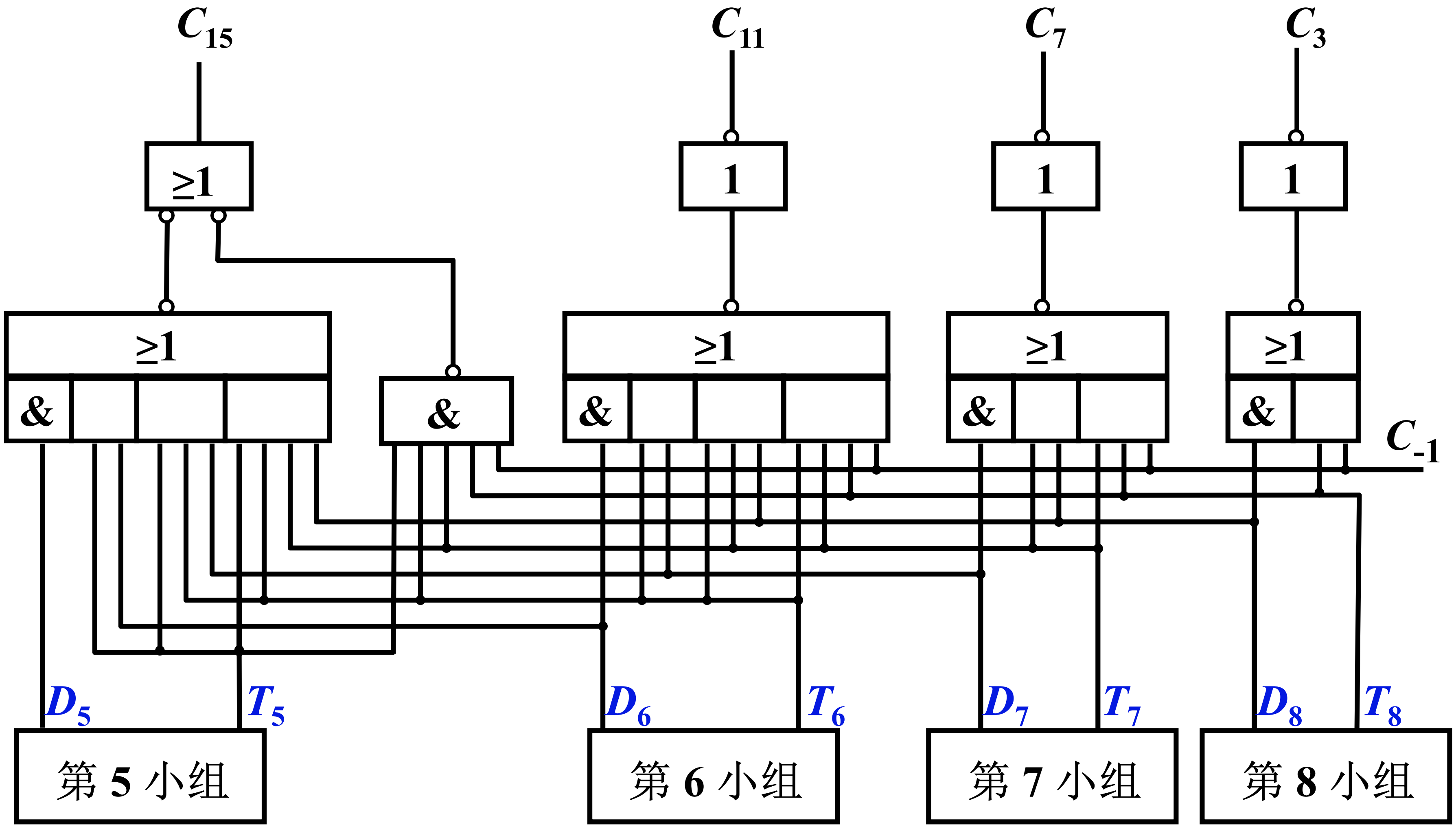
$$C_7 = D_7 + T_7C_3 = D_7 + T_7D_8 + T_7T_8C_{-1}$$

$$C_{11} = D_6 + T_6C_7 = D_6 + T_6D_7 + T_6T_7D_8 + T_6T_7T_8C_{-1}$$

$$C_{15} = D_5 + T_5C_{11} = D_5 + T_5D_6 + T_5T_6D_7 + T_5T_6T_7D_8 + T_5T_6T_7T_8C_{-1}$$

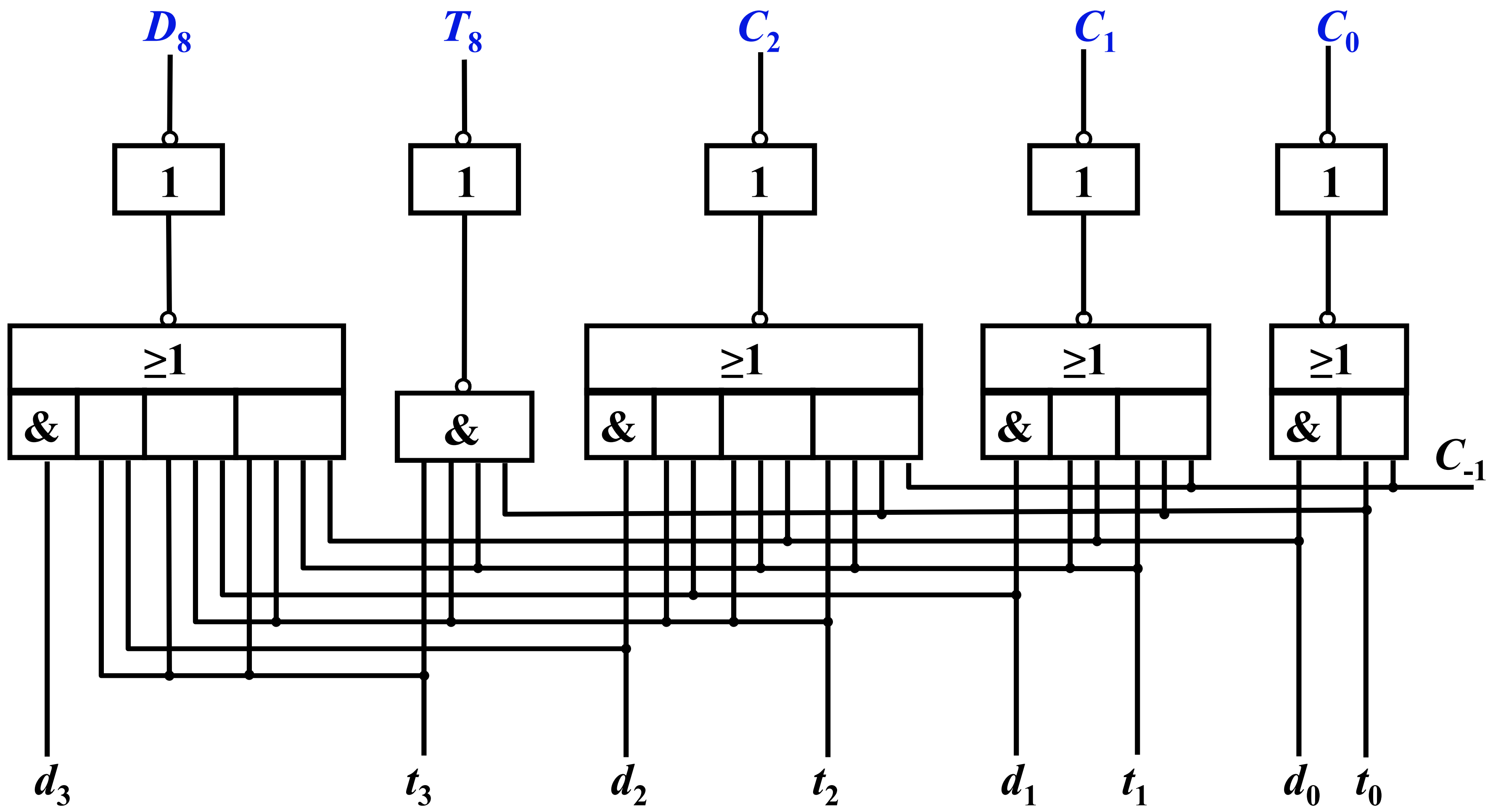
(4) 双重分组跳跃进位链的 大组 进位线路 6.5

以第 2 大组为例



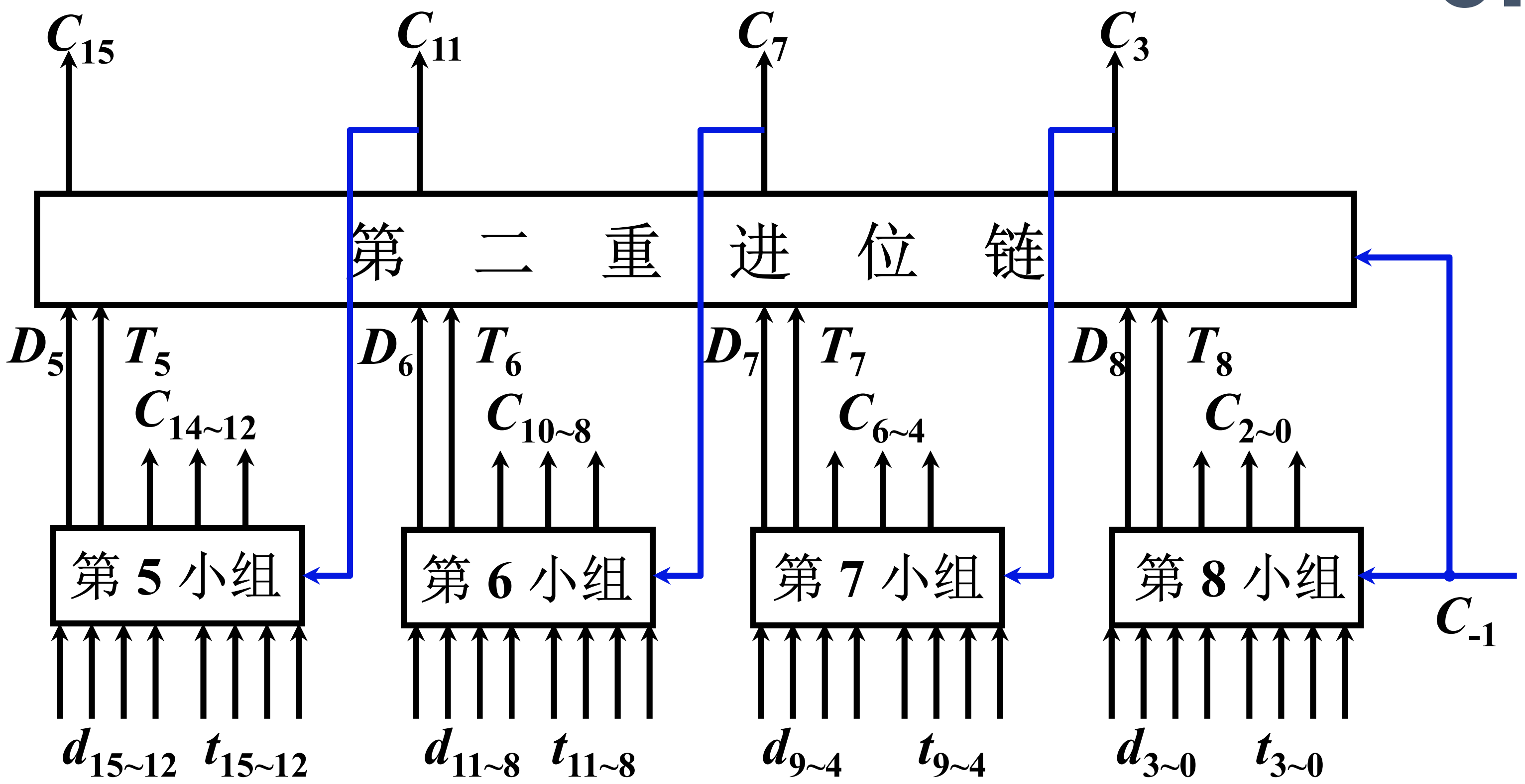
(5) 双重分组跳跃进位链的 小组 进位线路 6.5

以第 8 小组为例　　只产生 低 3 位 的进位和 本小组的 D_8 T_8



(6) n =16 双重分组跳跃进位链

6.5



当 d_i、t_i 和 C_{-1} 形成后　经 2.5 t_y　产生 C_2、C_1、C_0、$D_5 \sim D_8$、$T_5 \sim T_8$

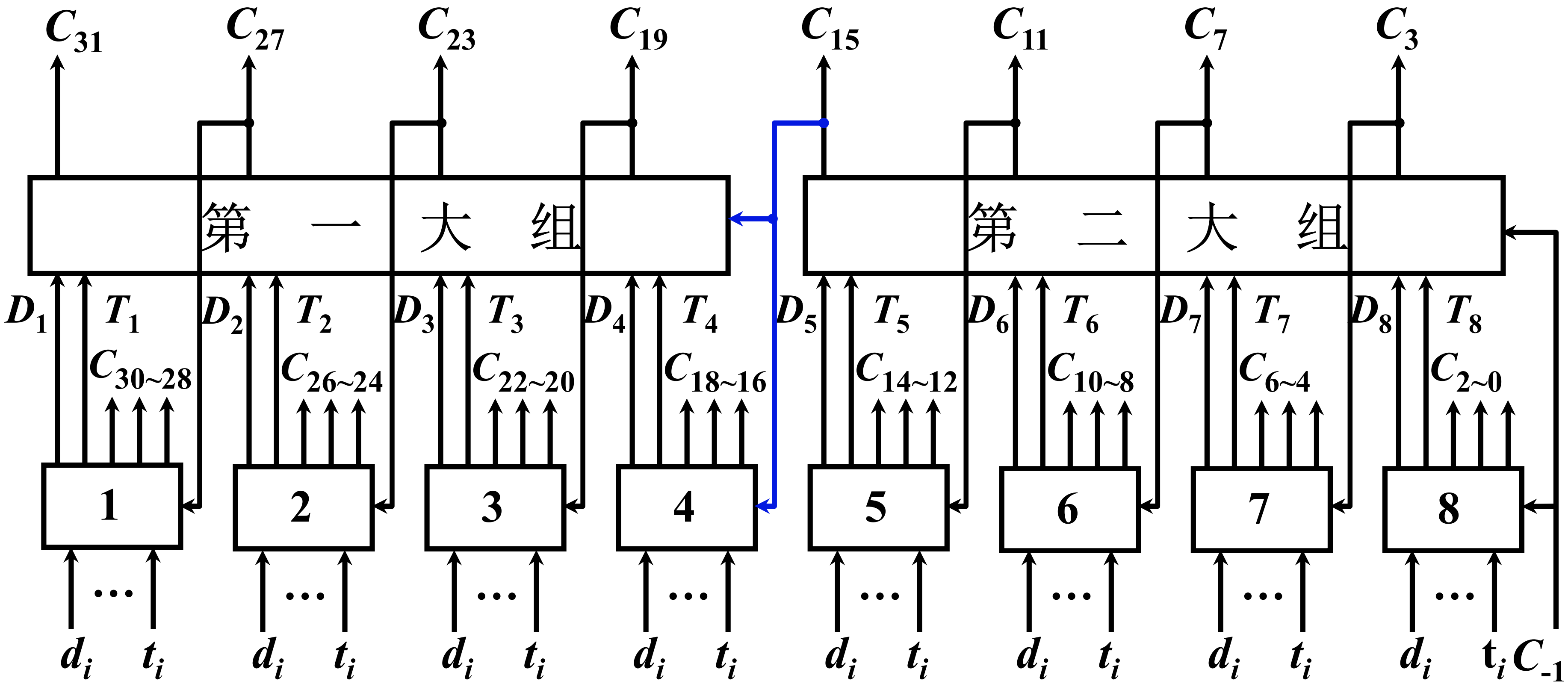
经 5 t_y　产生 C_{15}、C_{11}、C_7、C_3

经 7.5 t_y　产生 $C_{14}\sim C_{12}$、$C_{10}\sim C_8$、$C_6\sim C_4$

串行进位链　经 3 2 t_y　产生 全部进位

单重分组跳跃进位链　经 1 0 t_y　产生 全部进位

(7) n =32 双重分组跳跃进位链 6.5



当 d_i、t_i形成后 经 2.5 t_y 产生 C_2、C_1、C_0、$D_1 \sim D_8$、$T_1 \sim T_8$

5 t_y 产生 C_{15}、 C_{11}、 C_7、 C_3

7.5 t_y 产生 C_{18}~C_{16}、 C_{14}~C_{12}、 C_{10}~C_8、 C_6~C_4
C_{31}、 C_{27}、 C_{23}、 C_{19}

1 0 t_y 产生 C_{30}~C_{28}、 C_{26}~C_{24}、 C_{22}~C_{20}