2.2 定点加法、减法运算

提纲

2.2.1 补码加减法

- 如： $y=0.0111$ $[y]_{补}=0.0111$ $[-y]_{补}=1.1001$

从右边到左边，除了第一个1和右边的0保持不变以外，其它按位取反,很重要！

- $[-y]_{补}=\overline{[y]_{补}}+1$
- 补码加法

公式：$[x+y]_{补}=[x]_{补}+[y]_{补}$

- 补码减法：为了将减法转变为加法，需证明公式：

$[x-y]_{补}=[x]_{补}+[-y]_{补}$

2.2.1 补码加减法

- 例：x=-0.1011,y=0.0111

  $[x]_{补}$=1.0101　$[y]_{补}$=0.0111

  $[x+y]_{补}=[x]_{补}+[y]_{补}$=1.0101+0.0111=1.1100

  x+y=-0.0100

- 例：x=+0.01011,y=-0.00111

  $[x]_{补}$=0.01011 $[y]_{补}$=1.11001 $[-y]_{补}$=0.00111

  $[x-y]_{补}=[x]_{补}+[-y]_{补}$=0.10010

2.2.2 溢出的检测

- 可能产生溢出的情况
  - 两正数加，变负数，上溢（大于机器所能表示的最大数）
  - 两负数加，变正数，下溢（小于机器所能表示的最小数）

- 例： x = +0.1011, y = +0.1001,求x + y
- 例： x = -0.1101, y = -0.1011,求x + y

- 检测方法
  - 双符号位法
  - 单符号位法

2.2.2 溢出的检测

- ①双符号位法（参与加减运算的数采用变形补码表示）

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

- S_{f1} S_{f2}

0	0	正确（正数）
0	1	上溢
1	0	下溢
1	1	正确（负数）

- S_{f1}和S_{f2}分别为最高符号位和第二符号位
- S_{f1}表示正确的符号，逻辑表达式为$V=S_{f1} \oplus S_{f2}$,可以用异或门来实现

2.2.2 溢出的检测

- ①双符号位法（参与加减运算的数采用变形补码表示）

- 检验举例：
  - x = +0.1100, y = +0.1000,求x + y
  - x = -0.1100, y = -0.1000,求x + y
  - 结果出现了01或10的情况就为溢出

2.2.2 溢出的检测

- ①单符号位法

- C_f C_0

  0 0 正确（正数）

  0 1 上溢

  1 0 下溢

  1 1 正确（负数）

- $V=C_f \oplus C_0$ 其中C_f为符号位产生的进位,C_0为最高有效位产生的进位。

2.2.3 基本的二进制加法/减法器

- 首先我们来讨论最简单的一位全加器的结构，设定两个二进制数字A_i，B_i和一个进位输入C_i相加，产生一个和输出S_i，以及一个进位输出C_{i+1}。

$$\begin{array}{rr} & A_i \\ + & B_i\ C_i \\ C_{i+1} & S_i \end{array}$$

- 下表列出一位全加器进行加法运算的输入输出真值表。

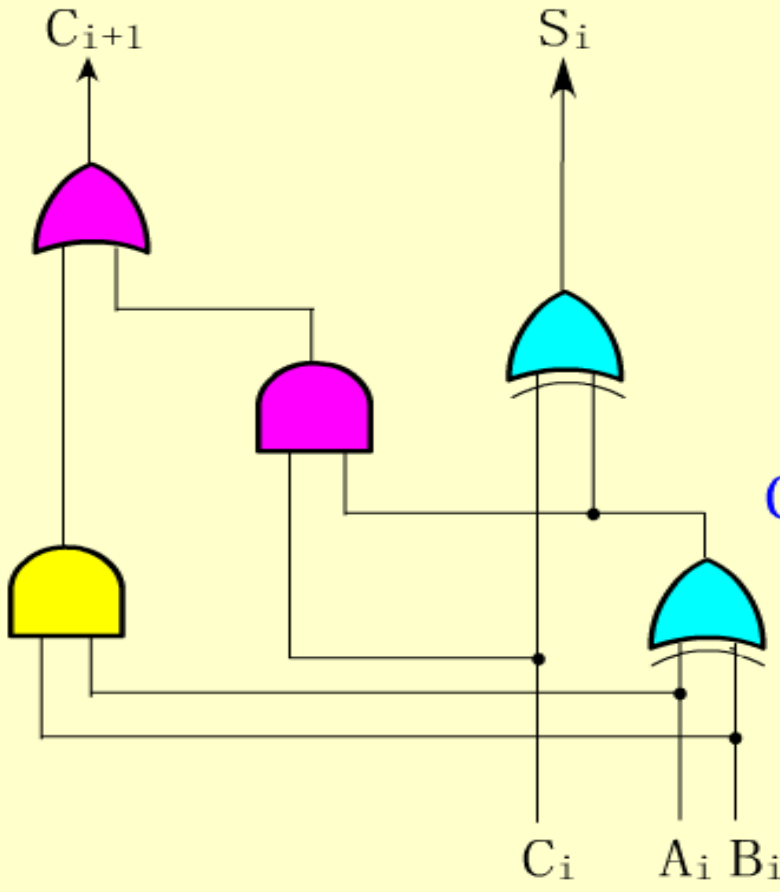
2.2.3 基本的二进制加法/减法器

- 一位全加器真值表

输入			输出	
A_i	B_i	C_i	S_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2.2.3 基本的二进制加法/减法器

- 根据真值表，三个输入端和两个输出端可按如下逻辑方程进行联系：

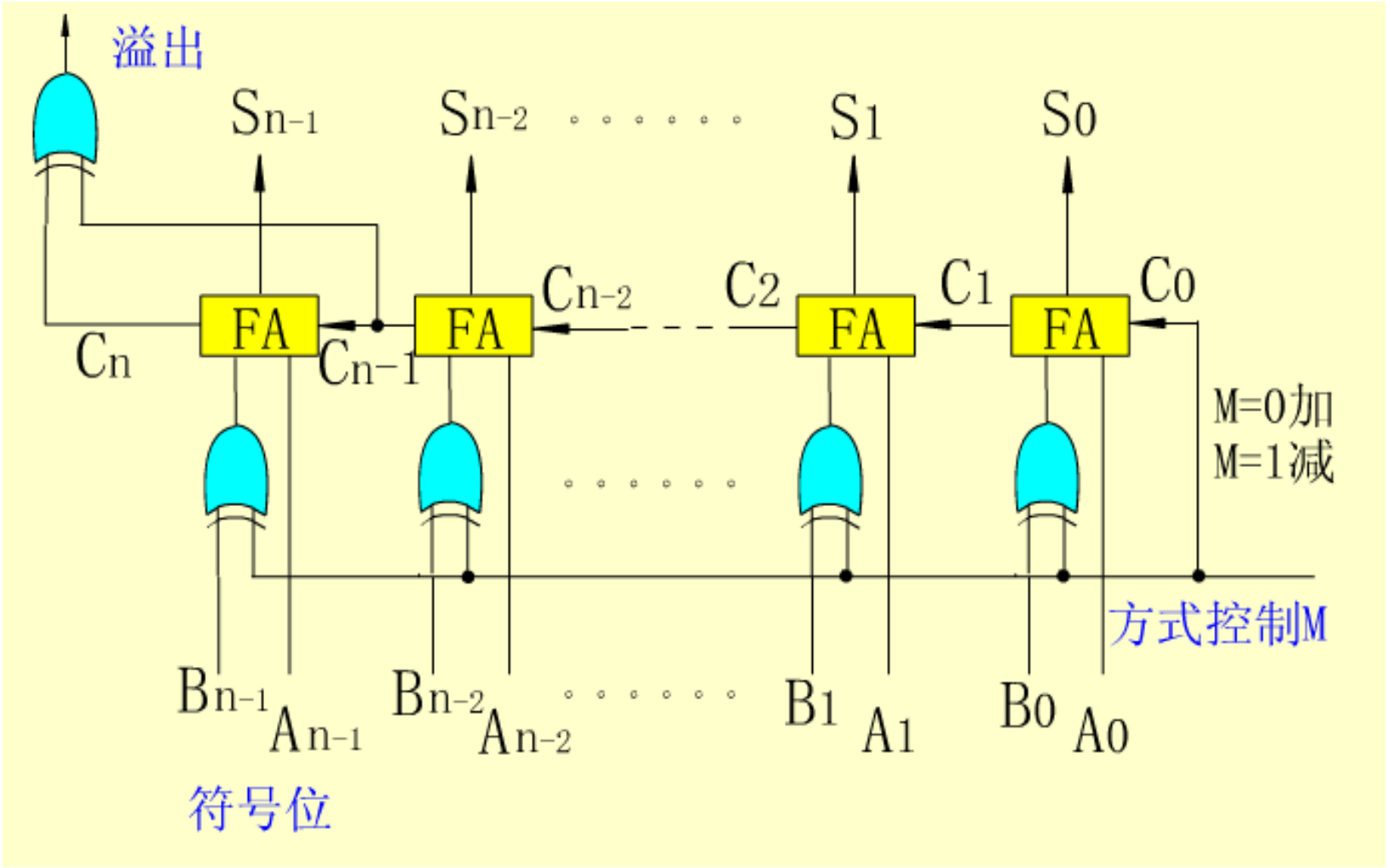


$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = A_iB_i + B_iC_i + C_iA_i$$
$$= A_iB_i + (A_i \oplus B_i)C_i$$

2.2.3 基本的二进制加法/减法器

- 行波进位的补码加法/减法器



2.2.3 基本的二进制加法/减法器

- *n*个1位的全加器(FA)可级联成一个*n*位的行波进位加减器
- M为方式控制输入线，当M = 0时，作加法(A + B)运算；当M = 1时，作减法(A - B)运算，在后一种情况下，A - B运算转化成$[A]_{补} + [-B]_{补}$运算，求补过程由$\overline{[B]_{补}} + 1$来实现
- 因此图中最右边的全加器的起始进位输入端被连接到功能方式线M上，作减法时M = 1，相当于在加法器的最低位上加1
- 图中左边还表示出单符号位法的溢出检测逻辑；当$C_n = C_{n-1}$时，运算无溢出；而当$C_n \neq C_{n-1}$时，运算有溢出，经异或门产生溢出信号

2.2.3 基本的二进制加法/减法器

- 异或门延迟3T，“与”门 或“或”门的时间延迟为T
- 对一位全加器(FA)来说，S_i的时间延迟为6T，C_{i+1}的时间延迟为2T，其中T被定义为相应于单级逻辑电路的单位门延迟

- 假如采用图2.2(a)所示的一位全加器并考虑溢出检测，那么n位行波进位加法器的延迟时间t_a为

$$t_a = 3T + 3T + n \cdot 2T + 3T = (2n + 9)T$$

- 第1个3T为计算所有B_i+M的时间（B_i与M异或），第2个3T为计算所有$A_i \oplus B_i$的时间（因为C_1要等到$A_1 \oplus B_1$计算完成之后才能计算出），2T为每级进位链的延迟时间，最后1个3T为溢出“异或”门的总时间
- S_i的异或C_i可以在溢出检测的3T计算