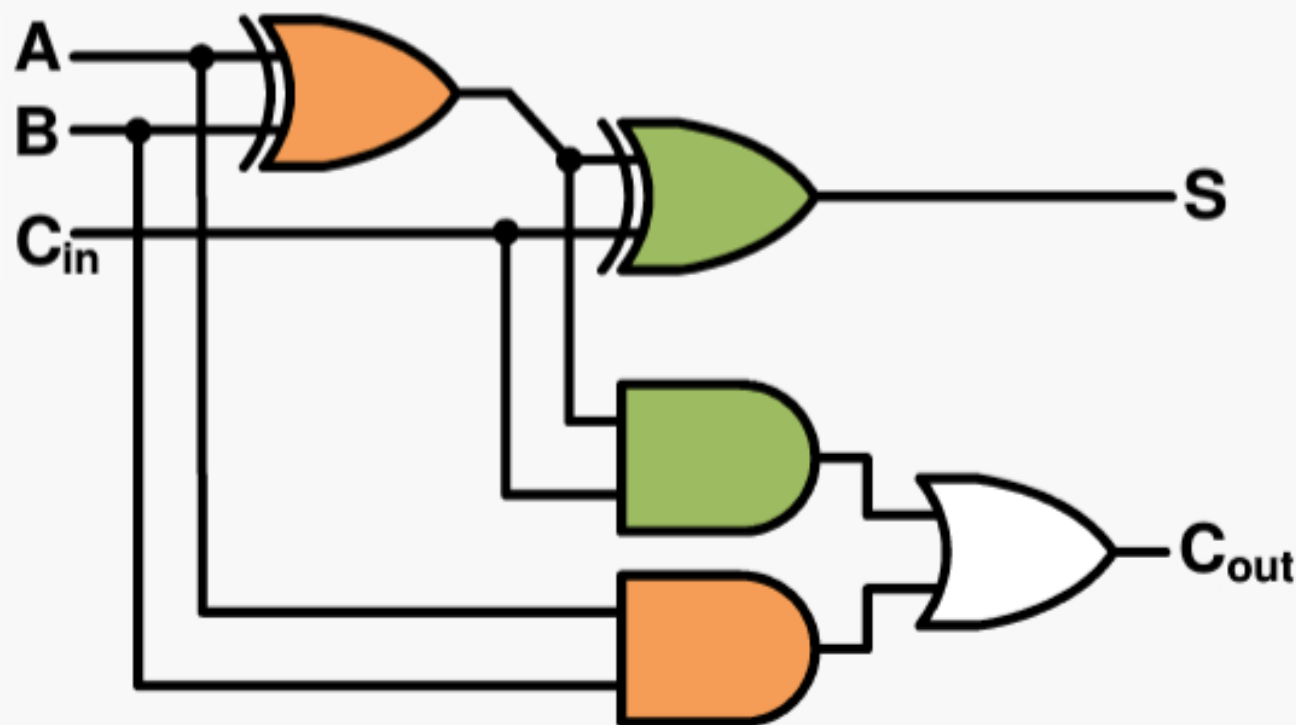


ALU设计

全加器 (Full Adder)

全加器由两个半加器构成

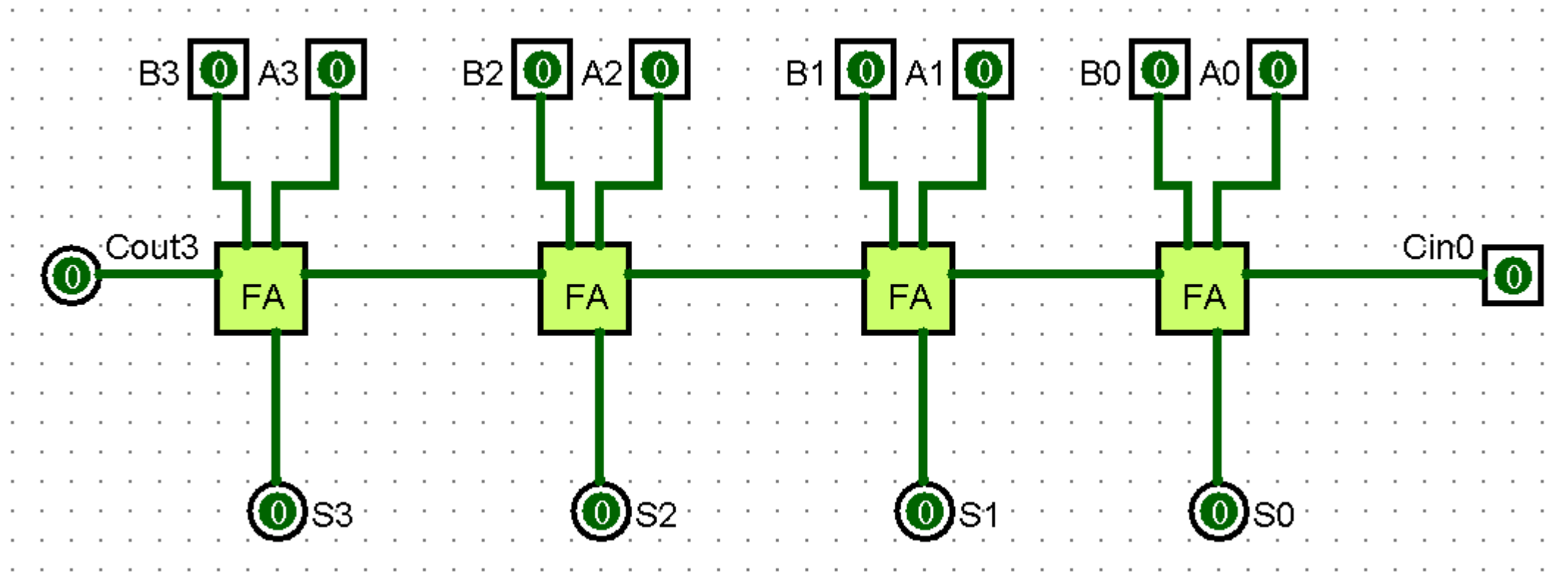
- 输入端口A、B、 C_{in} (进位输入)
- 输出端口S (和)、 C_{out} (进位输出)



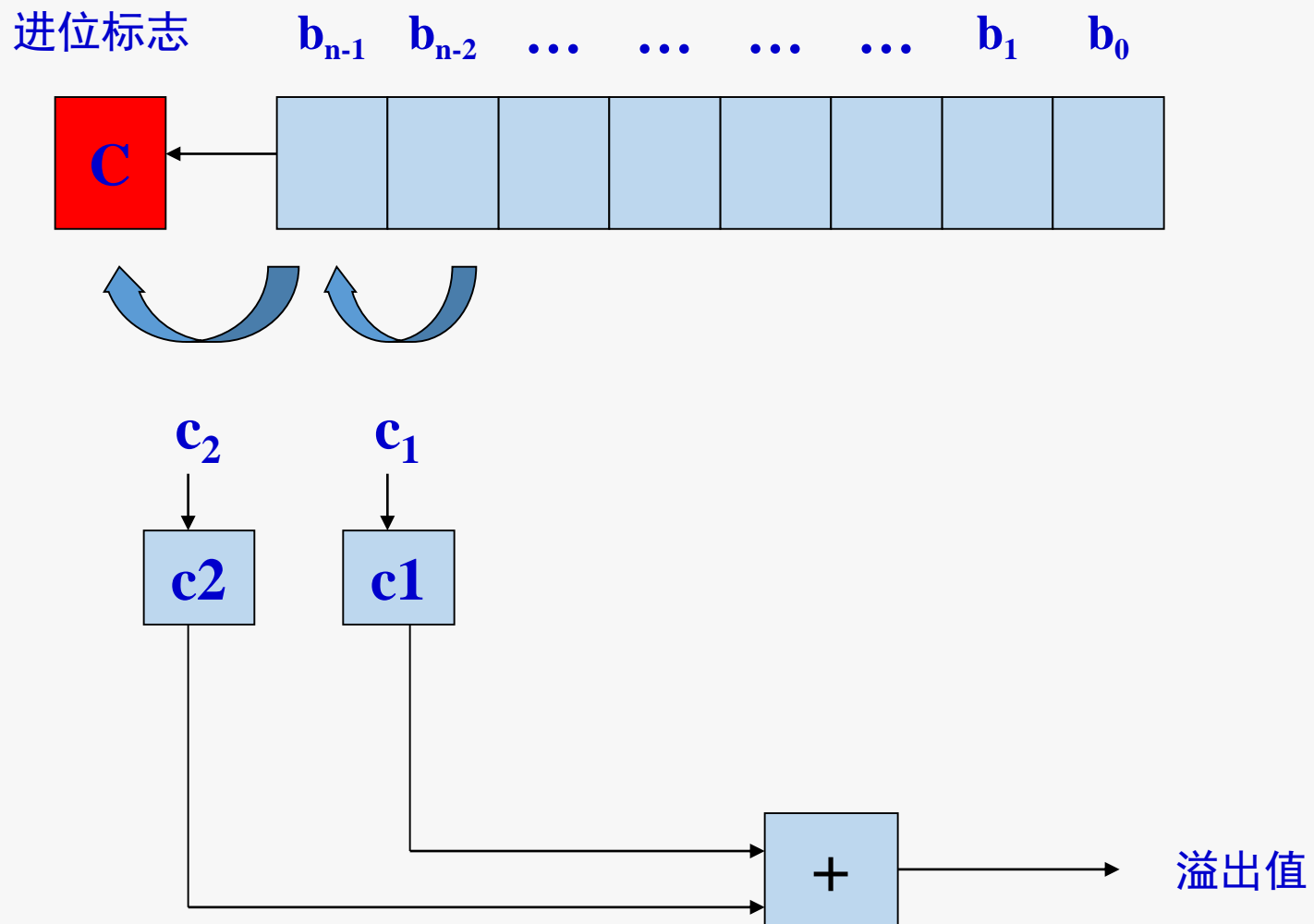
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

4位全加器

■ (1) 添加溢出标志判断

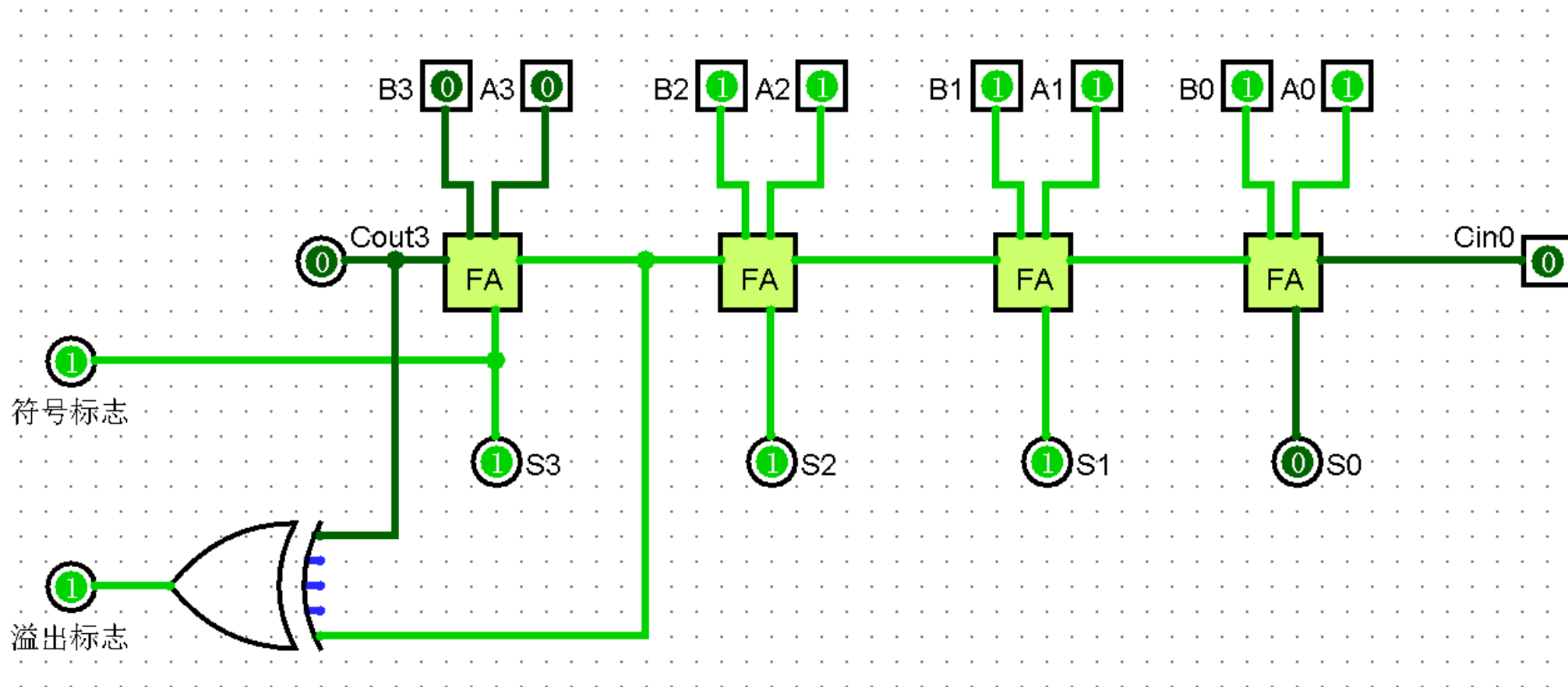


CPU怎么判断溢出?



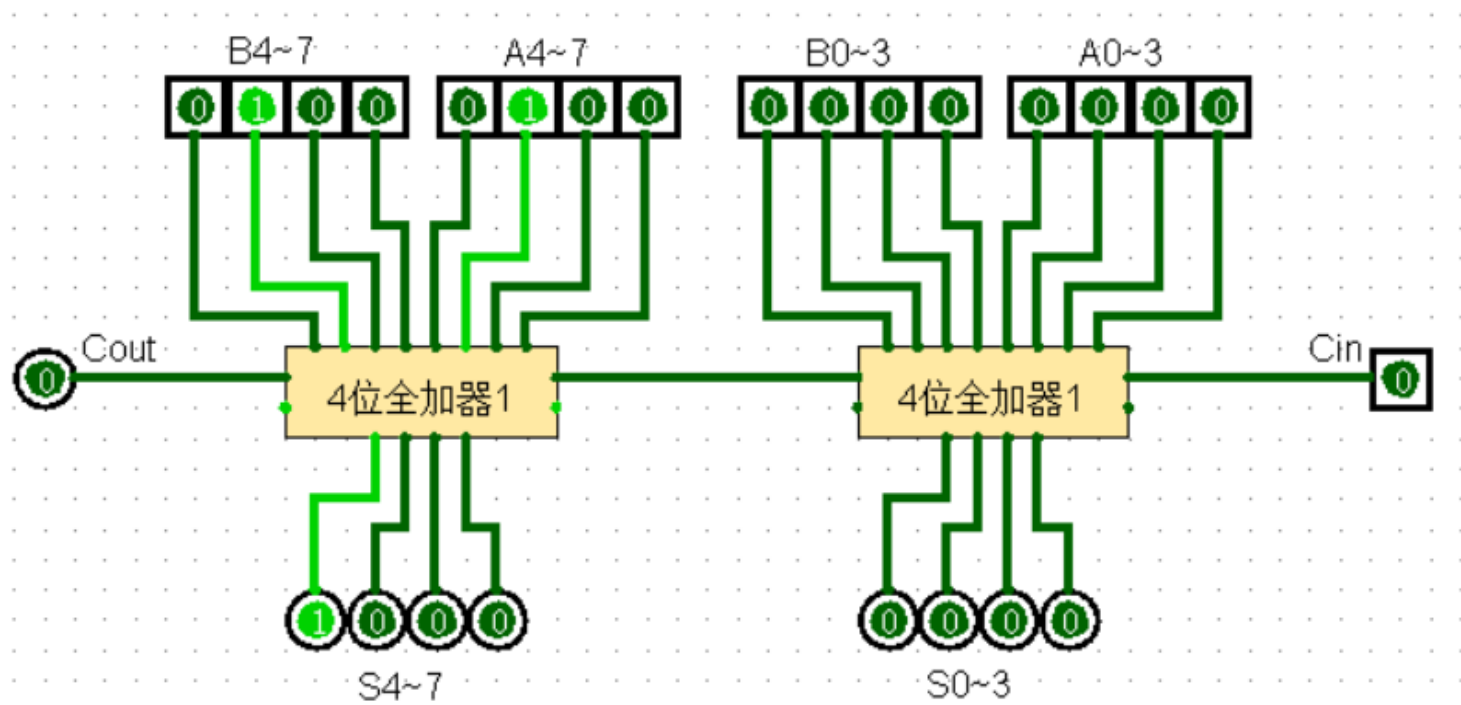
4位全加器

■ (1) 添加溢出标志判断



4位全加器

■ (2) 如何添加减法运算?



补码直接参与计算

■ $4 - 3 = 4 + (-3)$

■ $4 - (-3) = 4 + 3$

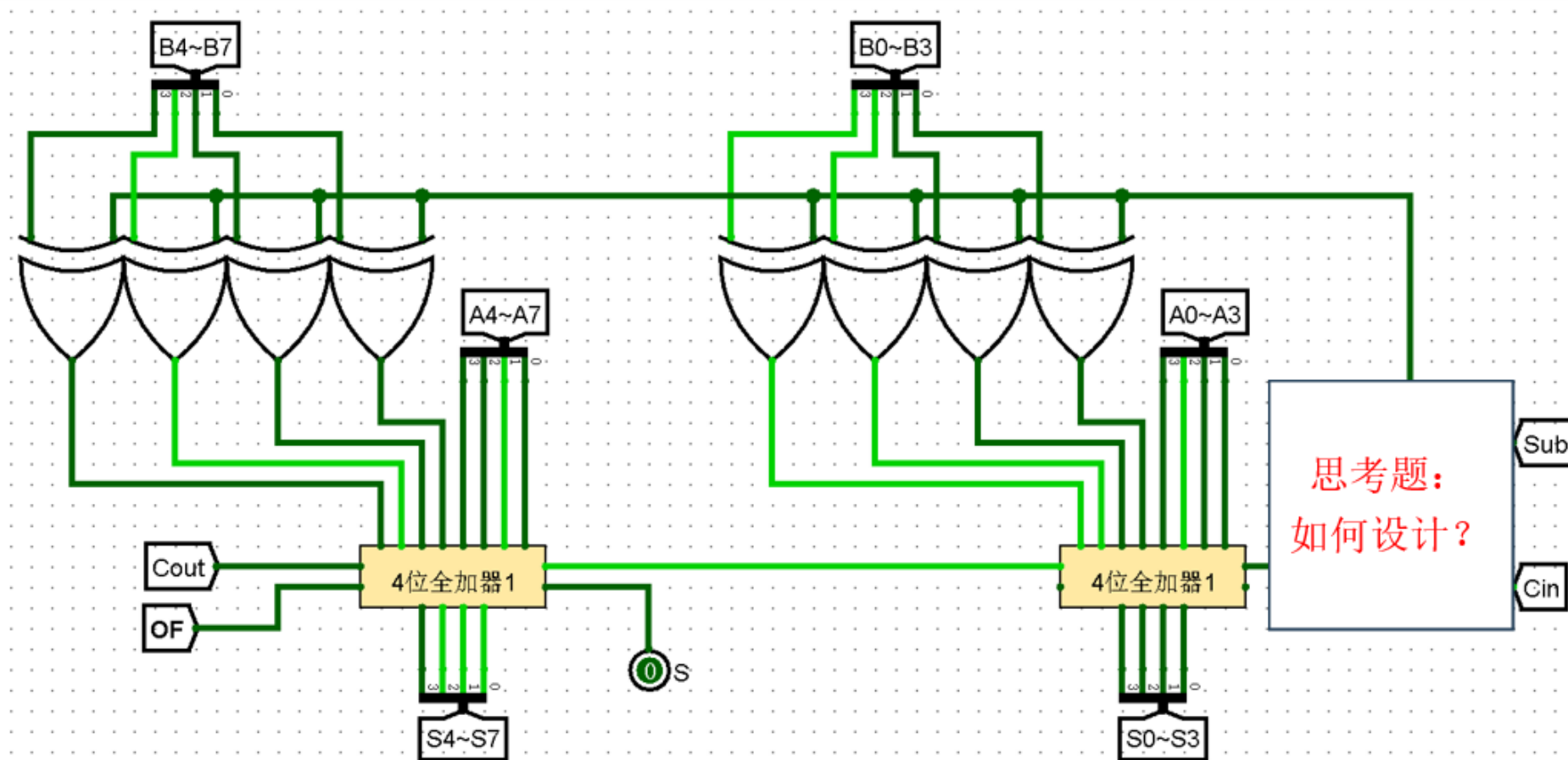
+3	00000011	
	11111100	+1
-3	11111101	
	00000010	+1

$$0 \xrightarrow{+1} 1$$

$$1 \xrightarrow{+1} 0$$

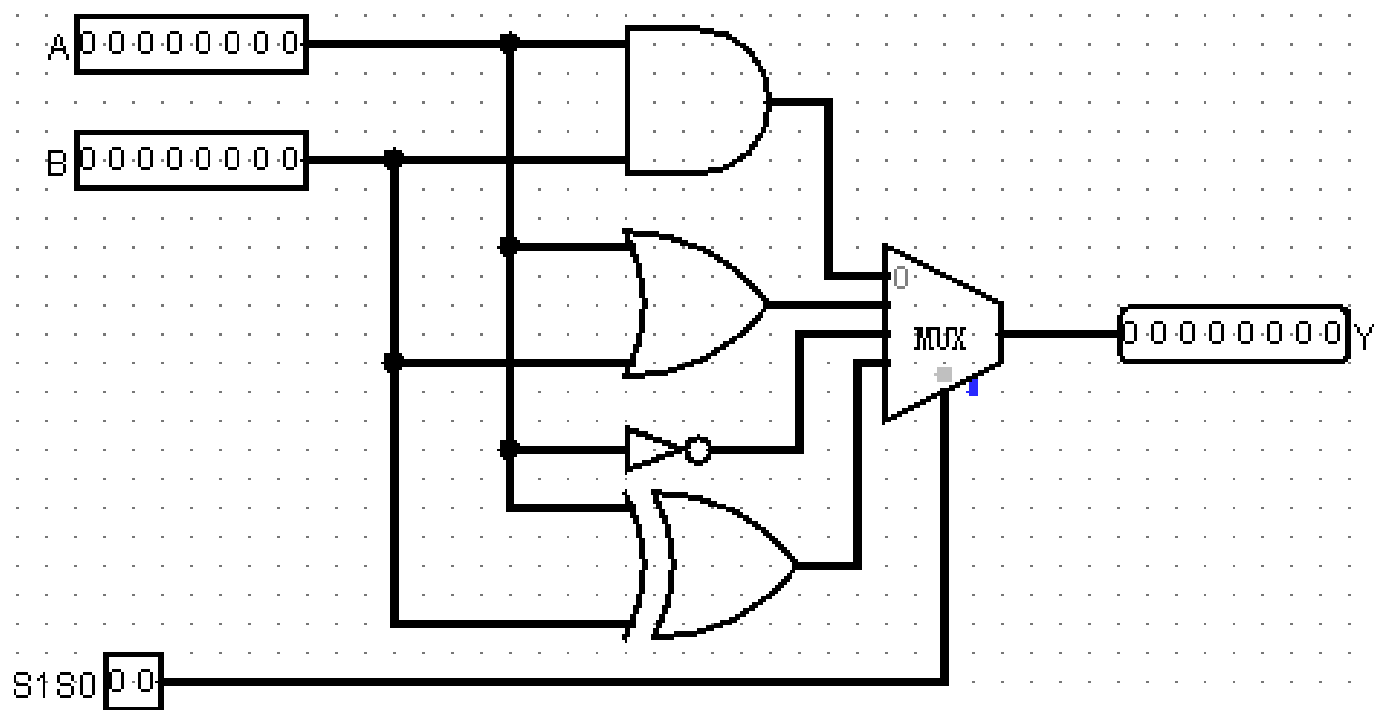
8位可控加减运算器

- Sub为减运算标志，OF为溢出标志，S为运算结果的最高位，即符号标志。
- Sub=0, $S=A+B$; Sub=1, $S=A-B$



逻辑运算

- 增加 8 位数据的基本逻辑运算包括：与、或、非、异或运算，并实现通过 4 选 1 选择器选择具体操作。



移位运算

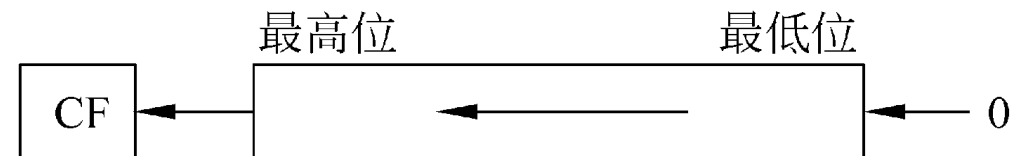
(a)算术左移

(b)逻辑左移

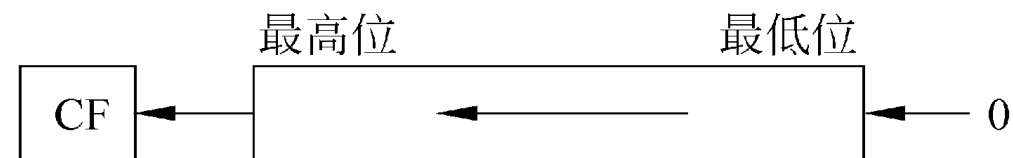
(c)算术右移

(d)逻辑右移

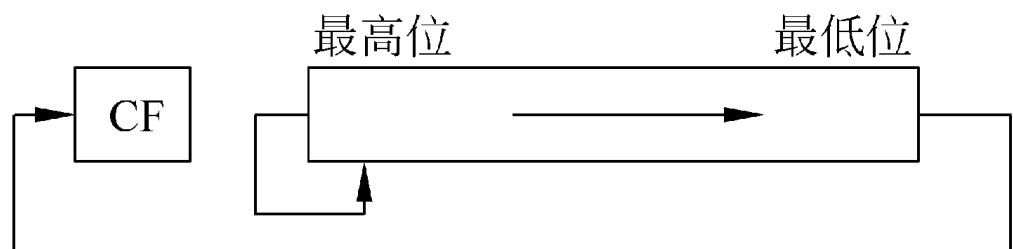
算术移位是对**有符号数**进行移位，在移位过程中必须保持符号不变；
逻辑移位是对**无符号数**进行移位，总是用0来填补已空出的位；



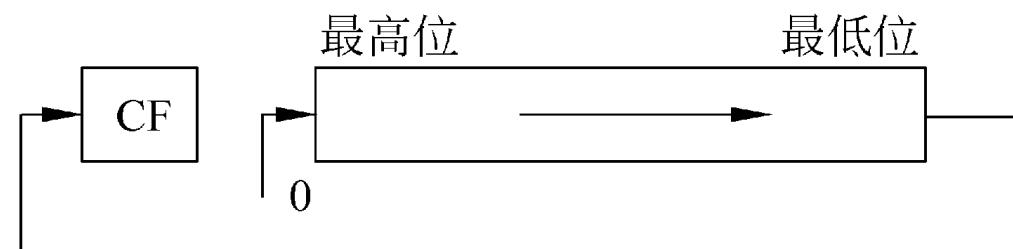
(a)



(b)



(c)



(d)

移位运算

■ 0000 0110

- 左移两位, 0001 1000
- 右移一位, 0000 0011 再右移一位, 0000 0001

■ 1000 0110

- 左移两位, 0001 1000
- 右移一位, 1100 0011 再右移一位, 1110 0001

移位运算

- 左移 n 位与乘以 2^n 等效

例如：

$$6 \times 2^2 = 00000110\text{B} \times 100\text{B} = 00011000\text{B}$$

- 同样，右移 n 位与除以 2^n 等效。若是无符号数，则右移使用逻辑右移SHR；如果是有符号数，则右移使用算术右移SAR。