《计算机系统基础（四）：编程与调试实践》

整数加减运算

整数加减运算

整数加减运算的电路

状态标志CF、ZF、SF和OF

整数加减运算结果的溢出问题

整数加减运算电路

补码加减运算公式：

$$[x+y]_{补} = [x]_{补} + [y]_{补} \quad (\bmod 2^n)$$

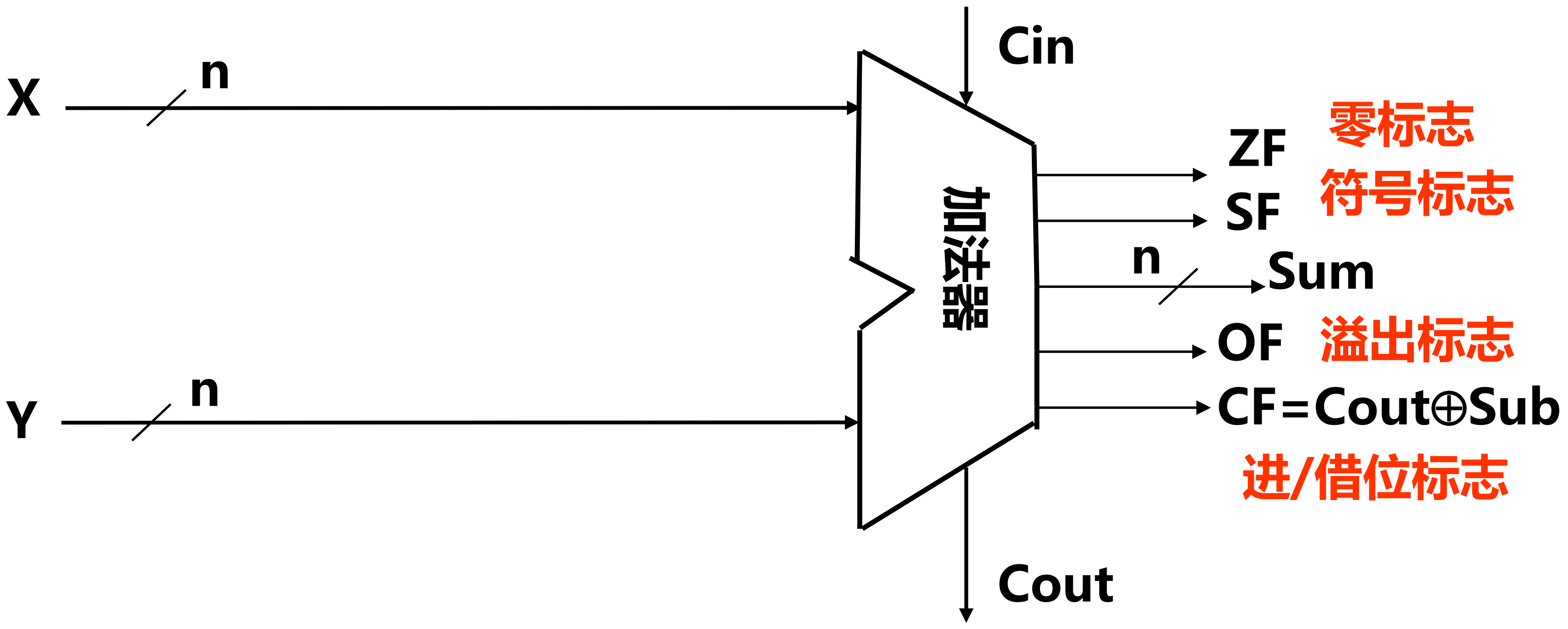
$$[x-y]_{补} = [x]_{补} + [-y]_{补} \quad (\bmod 2^n)$$

整数加减运算电路

补码加减运算公式：

$$[x+y]_{补} = [x]_{补} + [y]_{补} \quad (\bmod 2^n)$$

$$[x-y]_{补} = [x]_{补} + [-y]_{补} \quad (\bmod 2^n)$$



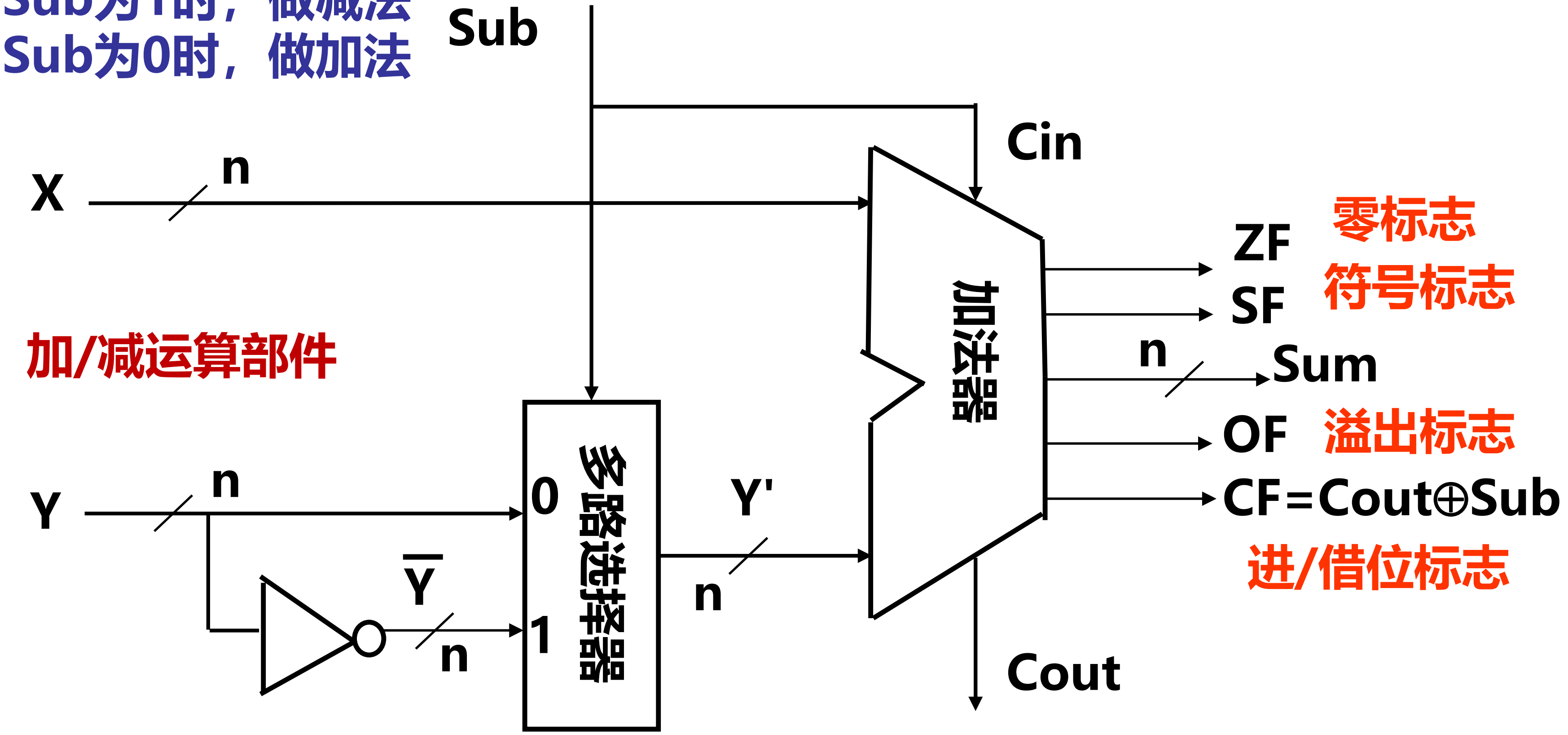
整数加减运算电路

补码加减运算公式：

$$[x+y]_{补} = [x]_{补} + [y]_{补} \quad (\bmod 2^n)$$

$$[x-y]_{补} = [x]_{补} + [-y]_{补} \quad (\bmod 2^n)$$

当Sub为1时，做减法
当Sub为0时，做加法



整数加减运算

```
#include "stdio.h"
void main( )
{   int   a=0x98765432,  b=0x87654321, c, d;
     unsigned int ua=0x98765432, ub=0x87654321, uc, ud;
     c=a+b;    uc=ua+ub;
     d=a-b;     ud=ua-ub;
     printf( “%d+(%d)=%d\n",a,b,c);
     printf("%u+%u=%u\n",ua,ub,uc);
     printf("%d-(%d)=%d\n",a,b,d);
     printf("%u-%u=%u\n",ua,ub,ud);
}
```

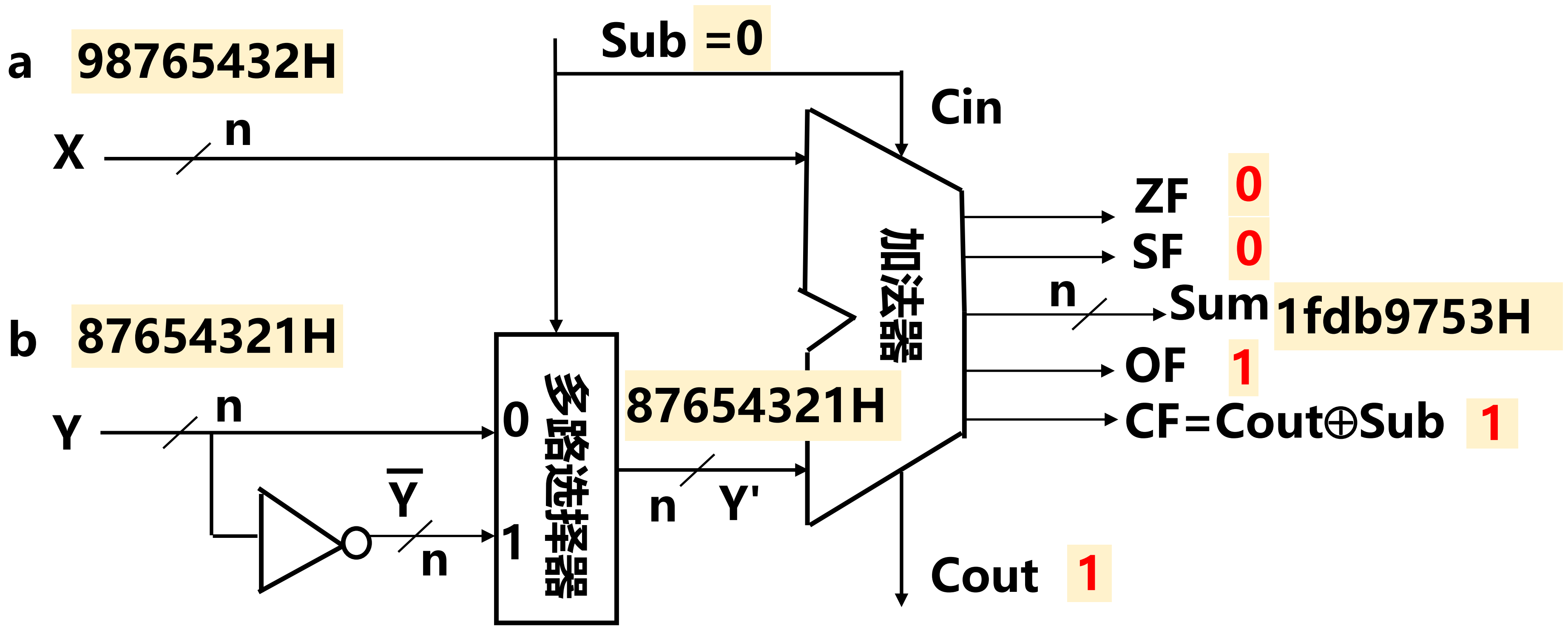
1. 运行程序，查看各变量的机器数。
2. 查看实现带符号整数加、减法运算和无符号整数加、减法运算的指令。
3. 在整数加减运算电路图上，分别标注出运算a+b、ua+ub、a-b和 ua-ub时的输入和输出内容，以及加法器的输入内容 。
1. 每次加减运算后，计算标志位OF、SF、ZF和CF的值。

整数加减运算电路

98765432H	1001 1000 0111 0110 0101 0100 0011 0010
+87654321H	+1000 0111 0110 0101 0100 0011 0010 0001
11fdb9753H	10001 1111 1101 1011 1001 0111 0101 0011

Cout=1

EFL=0000 0a07H =0……0 1010 0000 0111B

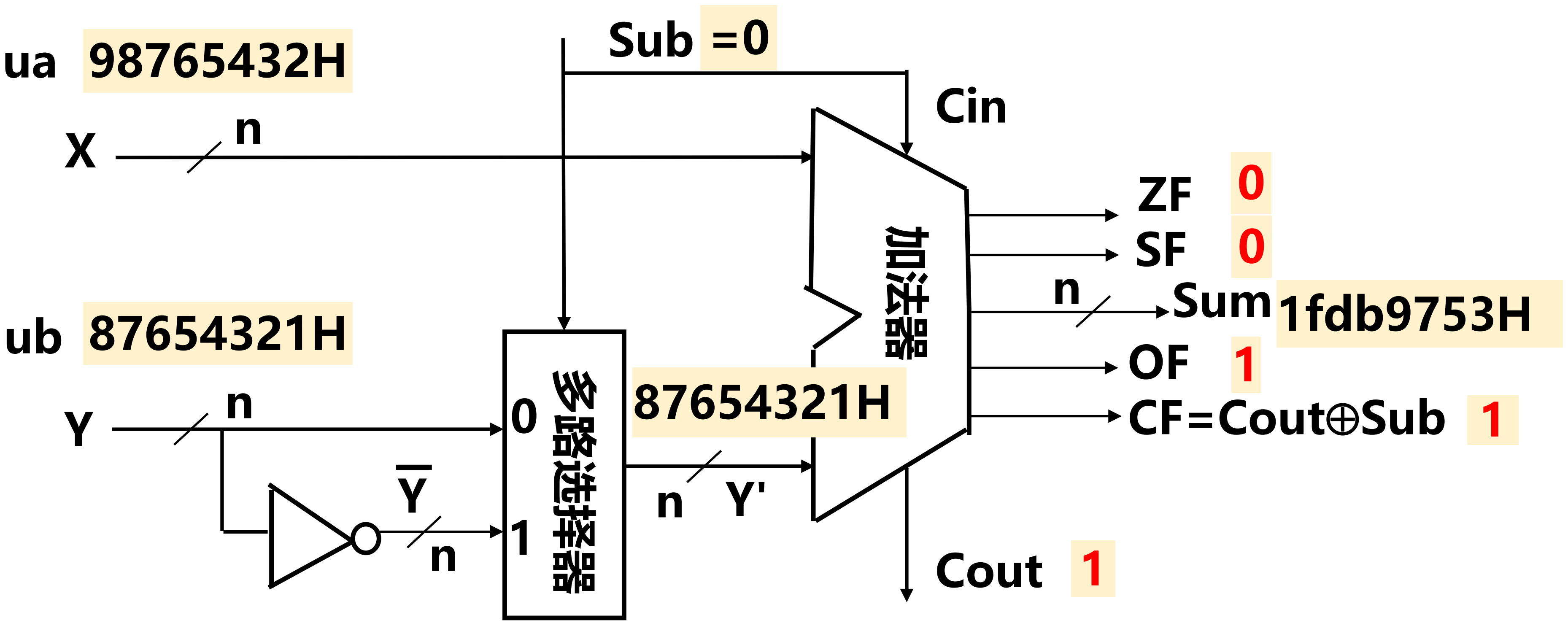


整数加减运算电路

98765432H	1001 1000 0111 0110 0101 0100 0011 0010
+87654321H	+1000 0111 0110 0101 0100 0011 0010 0001
11fdb9753H	10001 1111 1101 1011 1001 0111 0101 0011

Cout=1

EFL=0000 0a07H =0……0 1010 0000 0111B

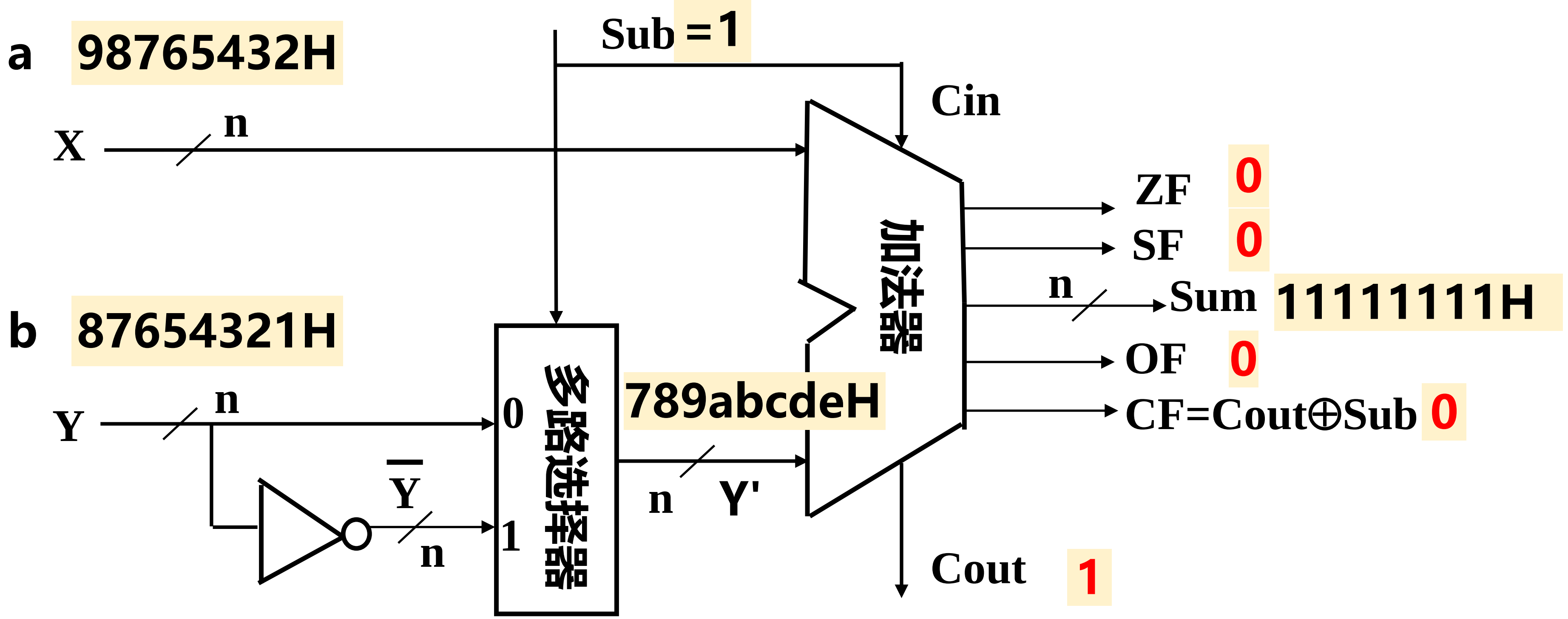


整数加减运算电路

	98765432H	1001 1000 0111 0110 0101 0100 0011 0010
	789abcdeH	0111 1000 1001 1010 1011 1100 1101 1110
+	1	+ 1
	11111 1111H	10001 0001 0001 0001 0001 0001 0001 0001

Cout=1

EFL=0000 0206H =0……0 0010 0000 0110B

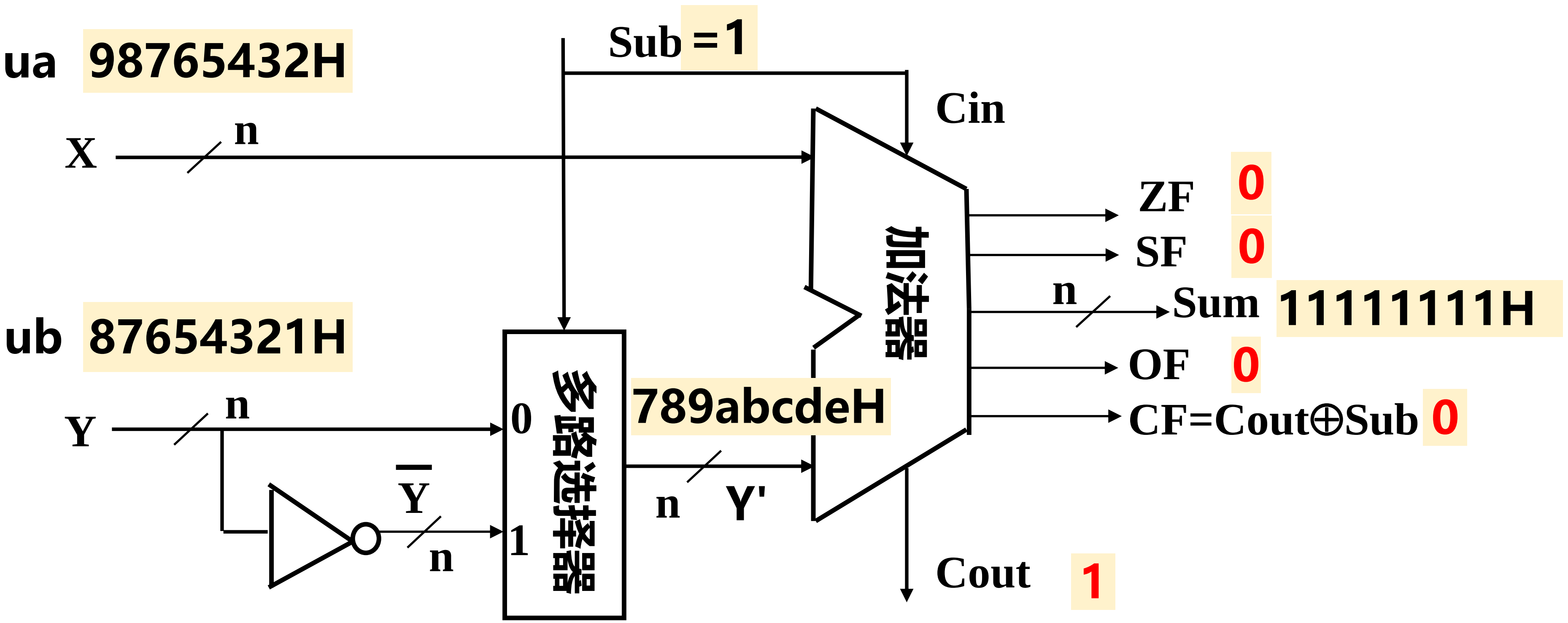


整数加减运算电路

98765432H		1001 1000 0111 0110 0101 0100 0011 0010
789abcdeH		0111 1000 1001 1010 1011 1100 1101 1110
+	1	+ 1
11111 1111H		10001 0001 0001 0001 0001 0001 0001 0001

Cout=1

EFL=0000 0206H =0……0 0010 0000 0110B

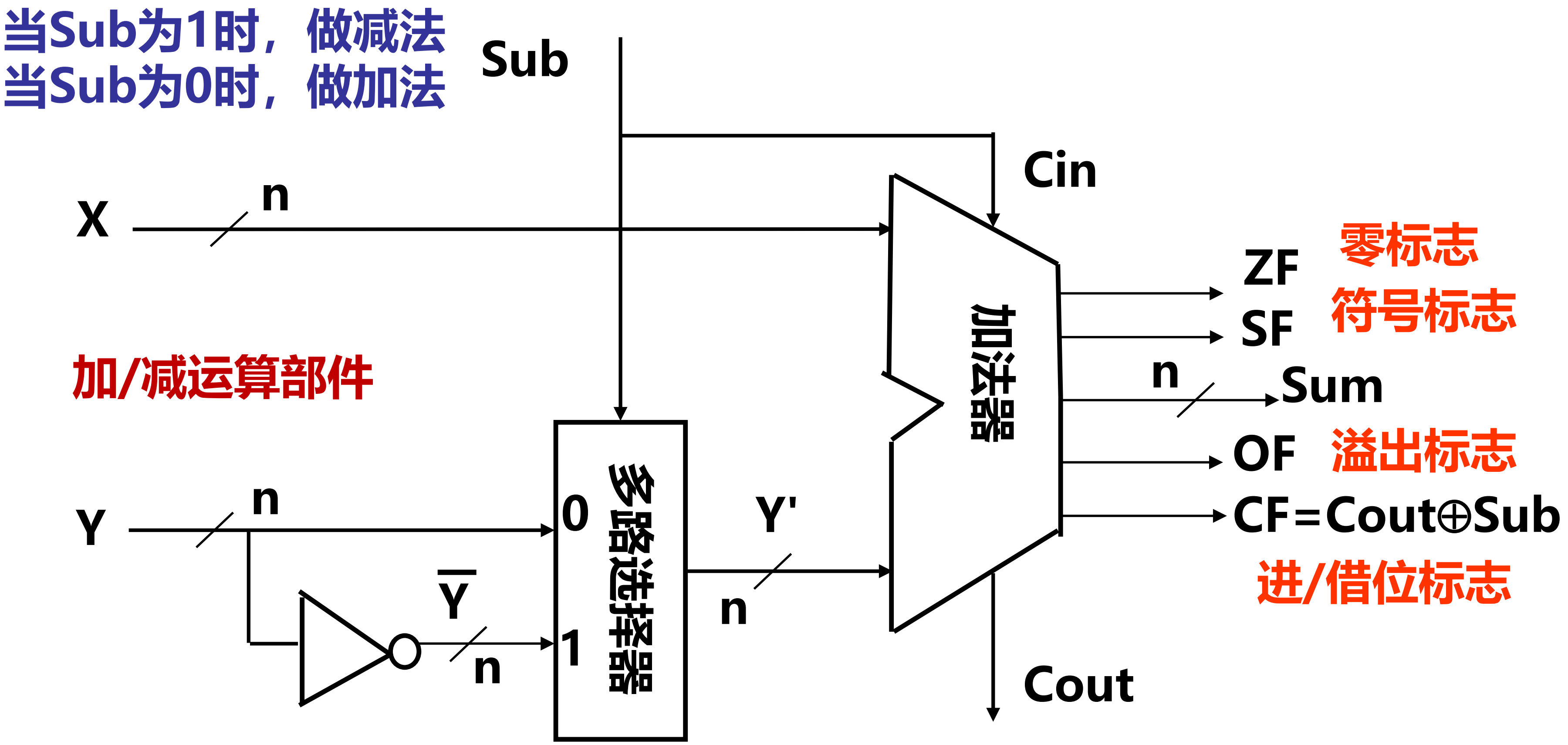


整数加减运算电路

补码加减运算公式：

$[x+y]_{补} = [x]_{补} + [y]_{补} \quad (\bmod 2^n)$

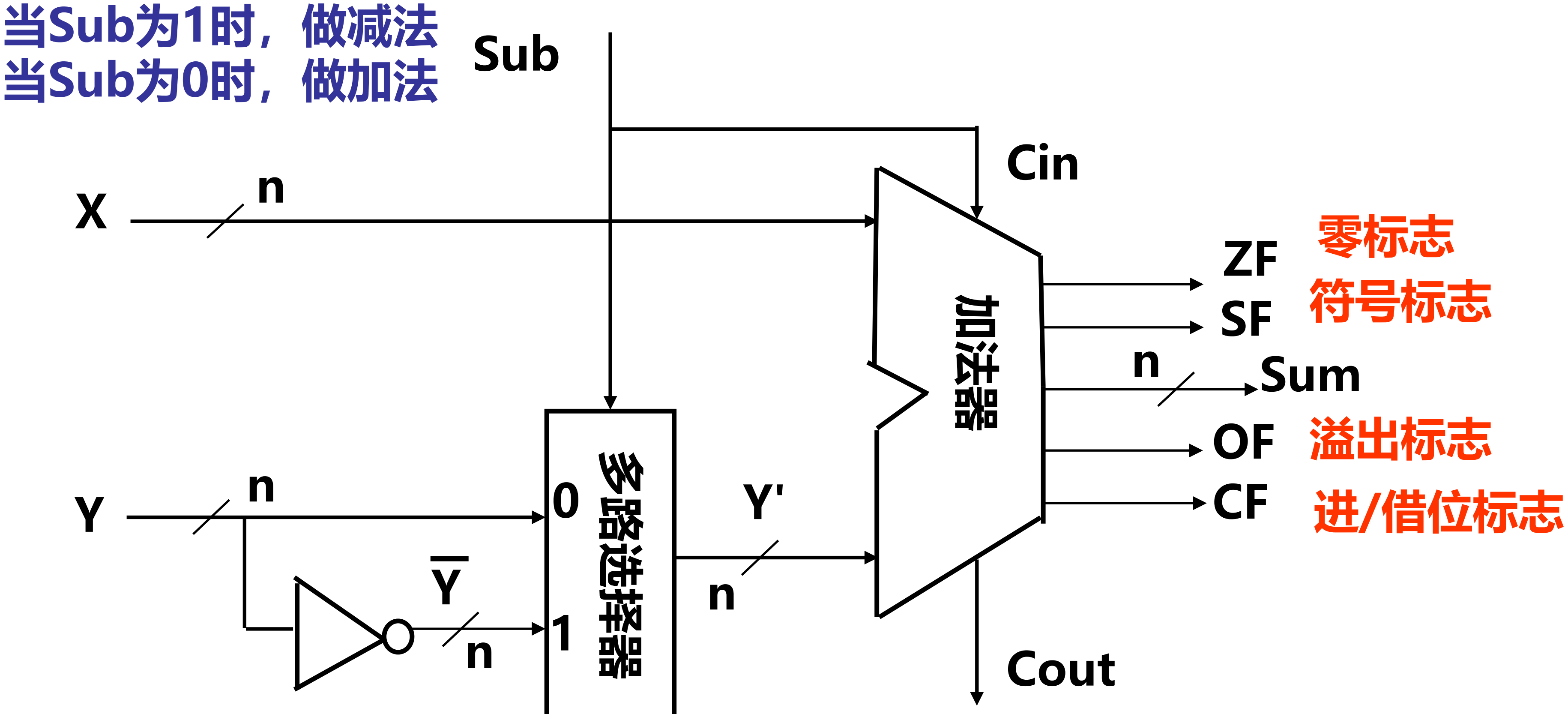
$[x-y]_{补} = [x]_{补} + [-y]_{补} \quad (\bmod 2^n)$



状态标志CF、ZF、SF和OF

$$OF = X_{n-1}\,Y'_{n-1}\,\overline{Sum_{n-1}} + \overline{X_{n-1}}\;\overline{Y'_{n-1}}\,Sum_{n-1}$$

$$CF = Cout \oplus Sub$$



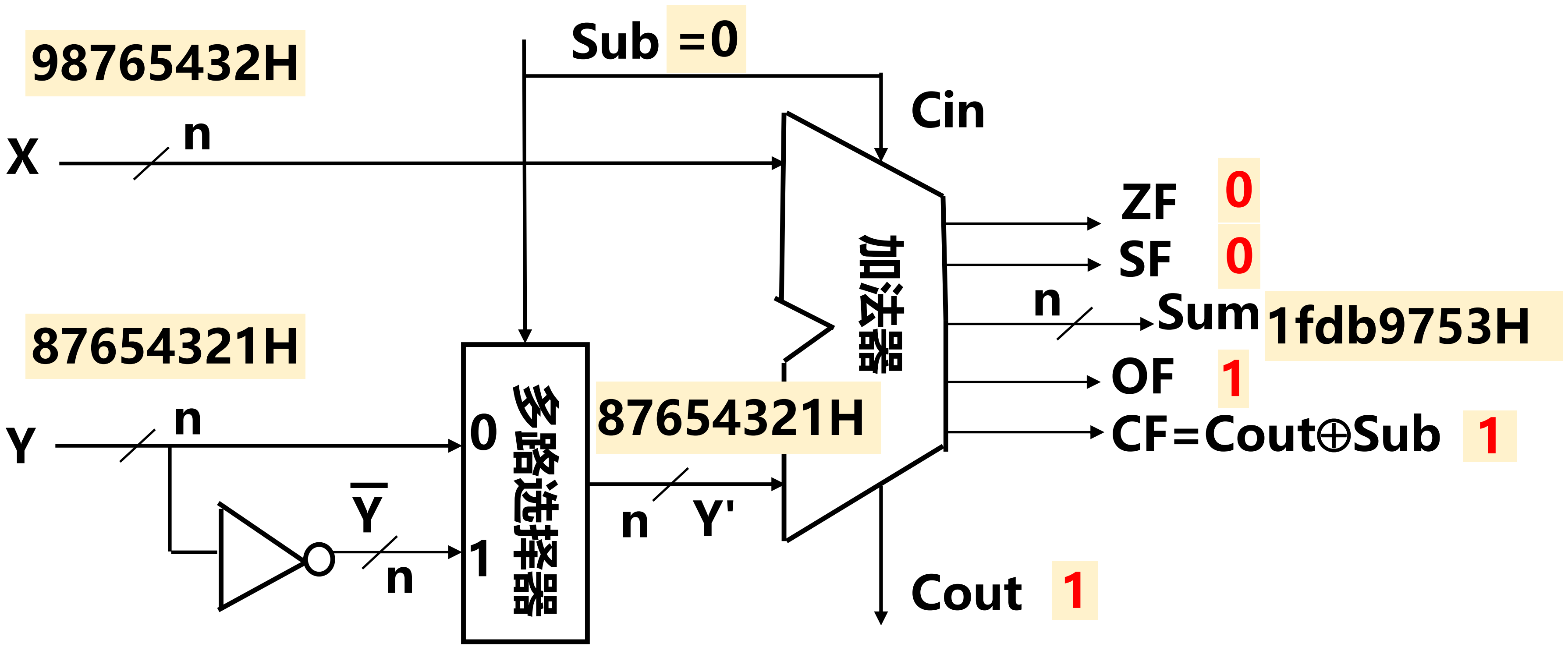
状态标志CF、ZF、SF和OF

98765432H		1001 1000 0111 0110 0101 0100 0011 0010
+87654321H		+1000 0111 0110 0101 0100 0011 0010 0001
11fdb9753H		10001 1111 1101 1011 1001 0111 0101 0011

Cout=1

$$OF=X_{n-1}\,Y'_{n-1}\,\overline{Sum}_{n-1}+\overline{X}_{n-1}\,\overline{Y'}_{n-1}\,Sum_{n-1} \quad =1$$

EFL=0000 0a07H =0……0 1010 0000 0111B



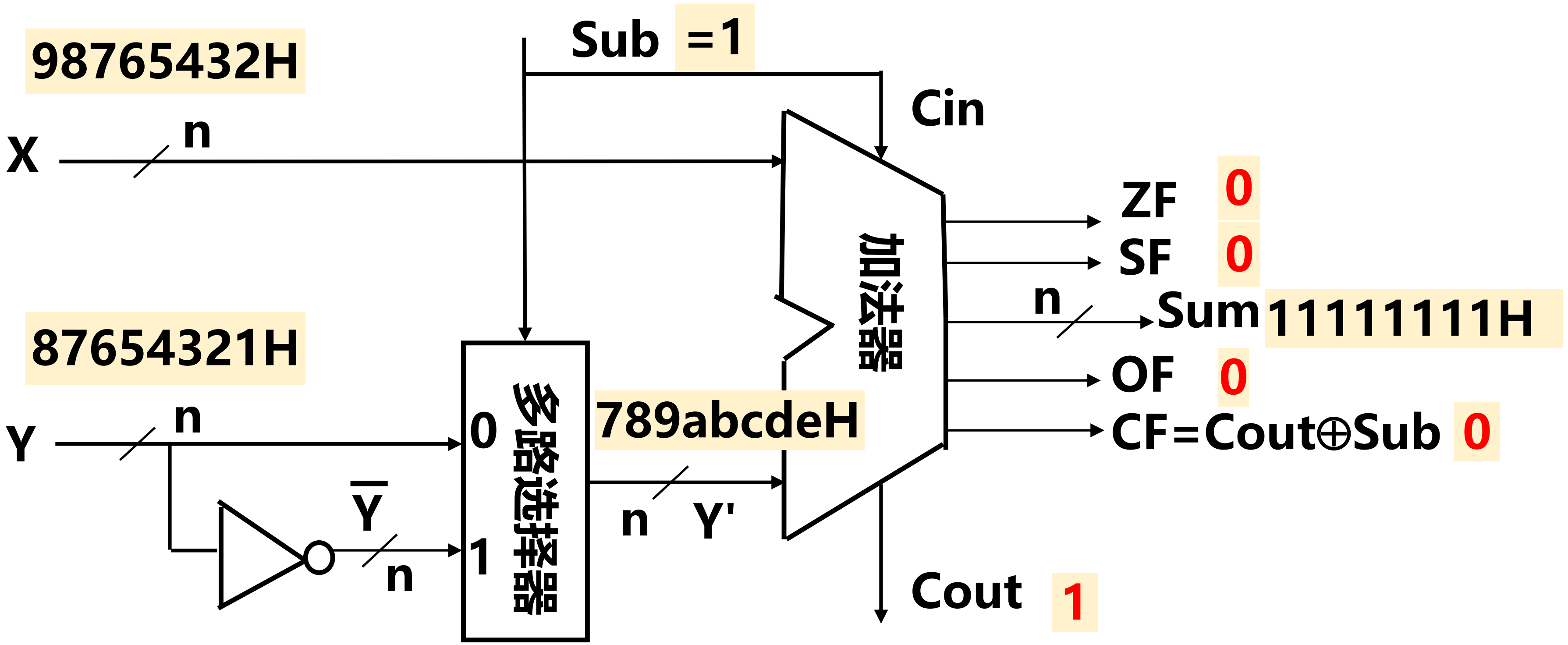
状态标志CF、ZF、SF和OF

98765432H	1001 1000 0111 0110 0101 0100 0011 0010
789abcdeH	0111 1000 1001 1010 1011 1100 1101 1110
+ 1	+ 1
11111 1111H	10001 0001 0001 0001 0001 0001 0001 0001

Cout=1

$OF=X_{n-1}\,Y'_{n-1}\,\overline{Sum_{n-1}}+\overline{X_{n-1}}\,\overline{Y'_{n-1}}\,Sum_{n-1}=0$

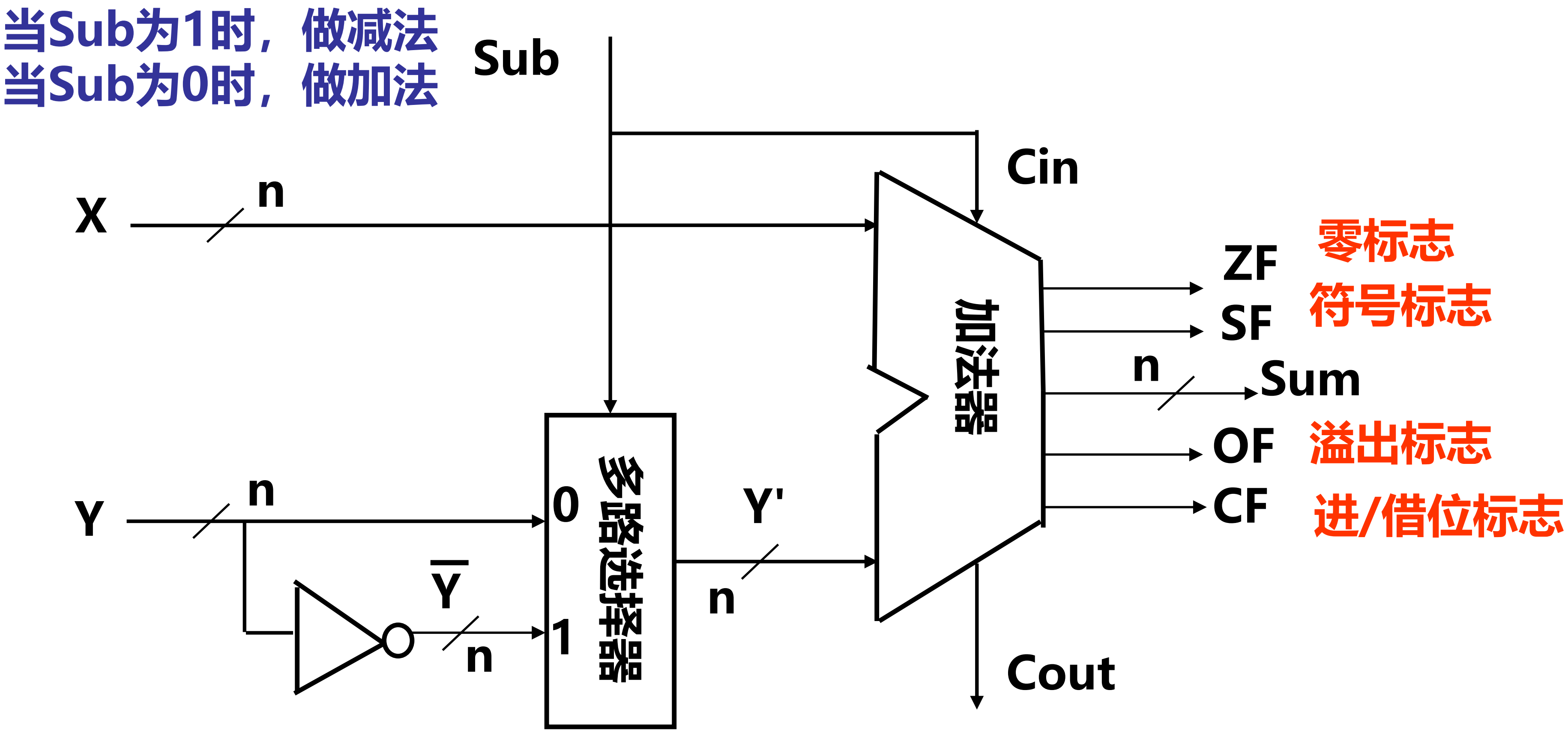
EFL=0000 0206H =0……0 0010 0000 0110B



状态标志CF、ZF、SF和OF

$$OF = X_{n-1}\,Y'_{n-1}\,\overline{Sum}_{n-1} + \overline{X}_{n-1}\,\overline{Y'}_{n-1}\,Sum_{n-1}$$

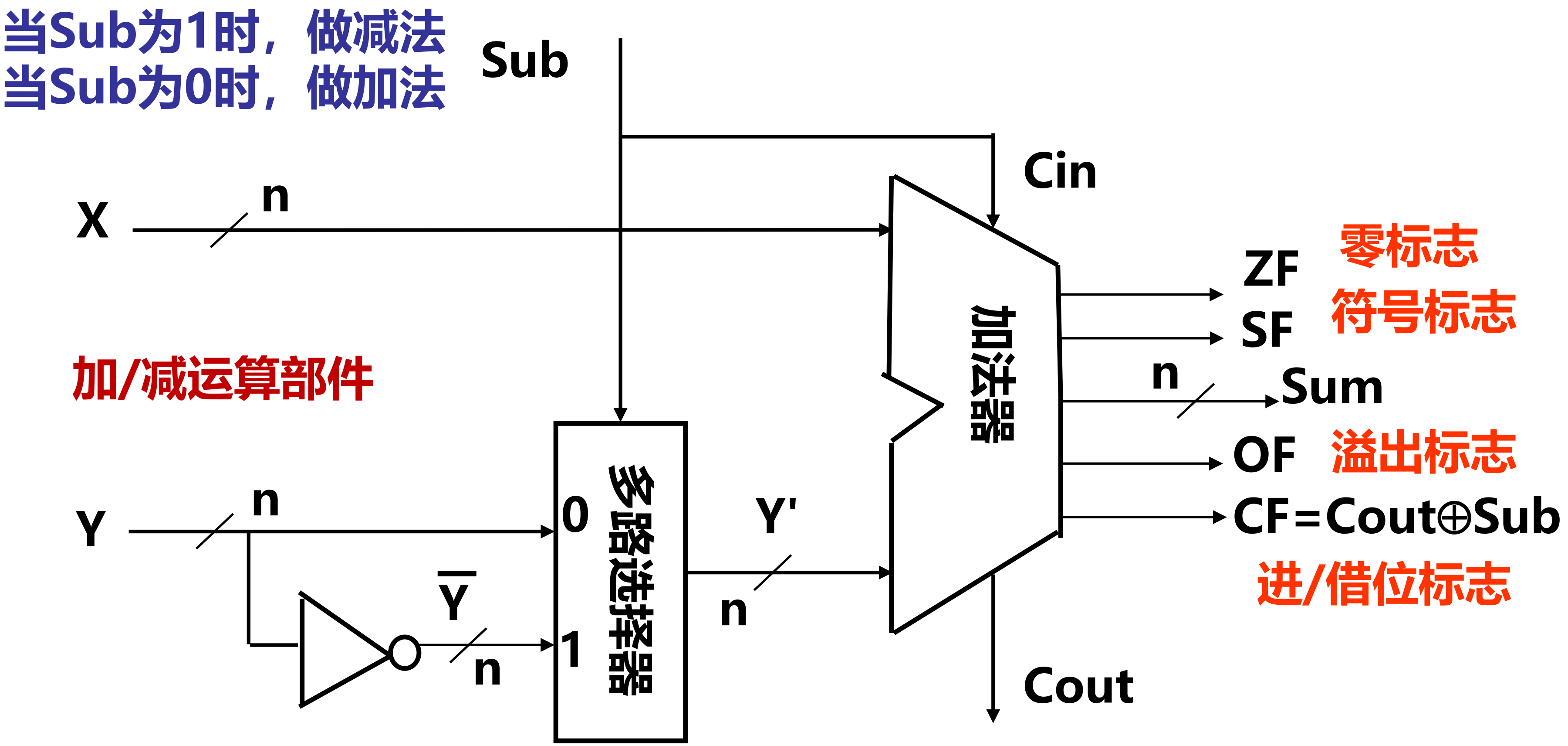
$$CF = Cout \oplus Sub$$



整数加减运算结果的溢出问题

无符号整数：用CF状态表示加减运算后是否有进位或借位，
OF值无意义。

带符号整数：用OF状态表示加减运算后结果是否溢出，
CF值无意义。



整数加减运算结果的溢出问题

FFFF FFFFH	1111 1111 1111 1111 1111 1111 1111 1111
+ 1H	0000 0000 0000 0000 0000 0000 0000 0001
10000 0000H	1 0000 0000 0000 0000 0000 0000 0000 0000

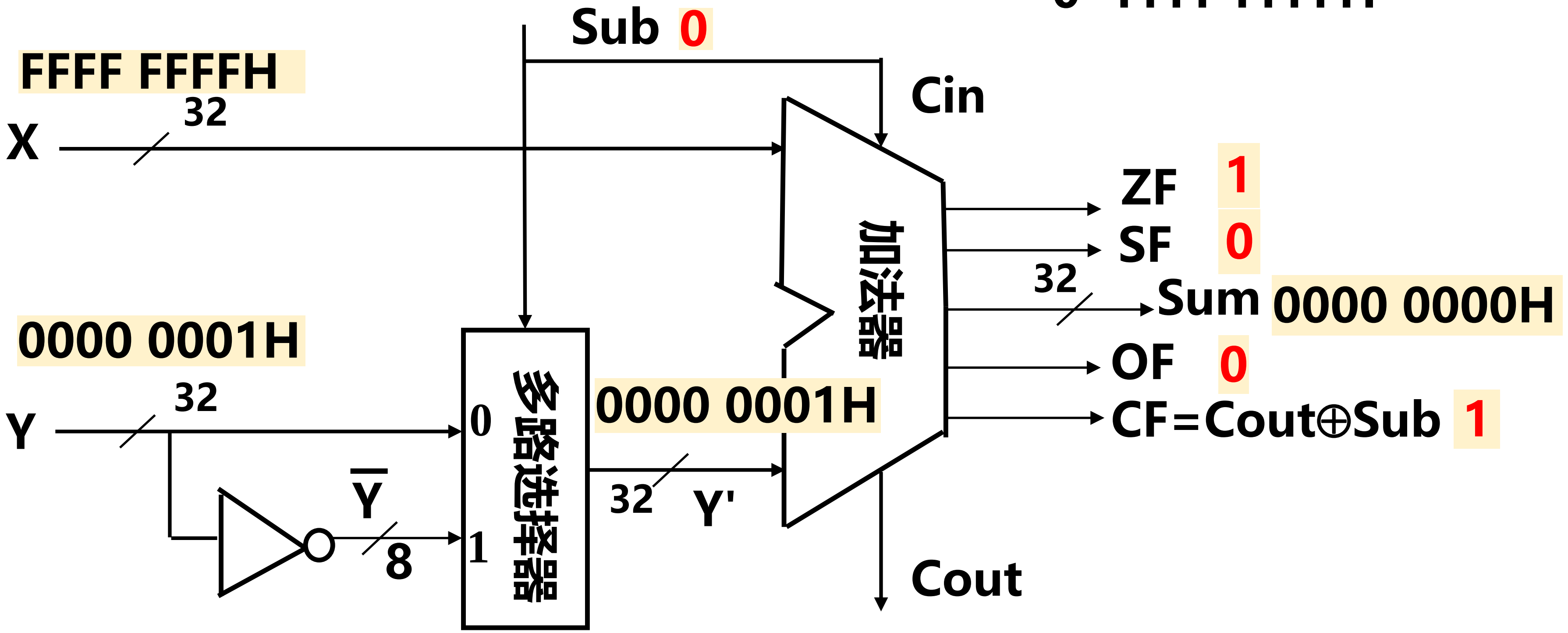
Cout=1

CF=Cout⊕Sub=1

无符号整数加法运算中，用CF表示进位，CF=Cout

当n=32时，无符号整数的表示范围是:

0~FFFF FFFFH



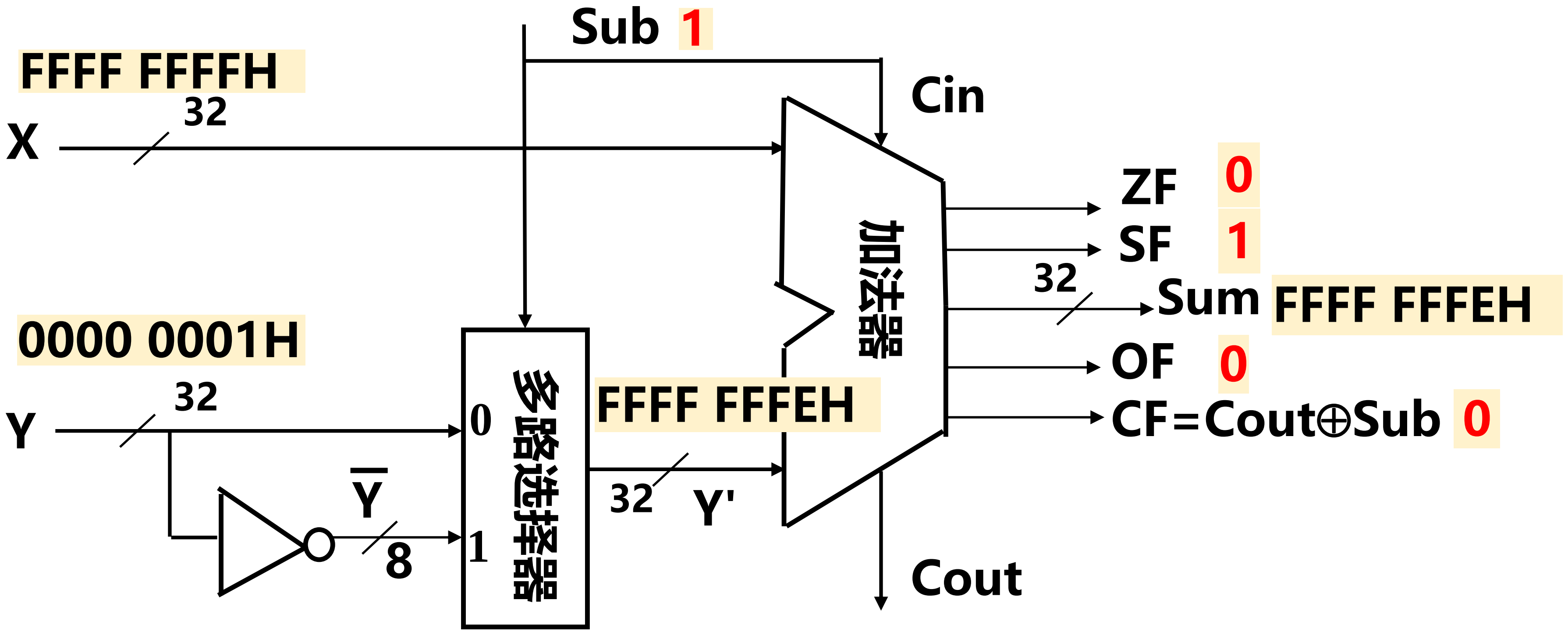
整数加减运算结果的溢出问题

```
  FFFF FFFFH          FFFF FFFFH
-         1H          FFFF FFFEH
0 FFFF FFFEH        +          1
                    1 FFFF FFFEH
```

FFFF FFFFH-1H
=FFFF FFFFH+（-1H+2^{32}）
因为够减，所以2^{32}没使用到，
成为Cout值

Cout=1　　CF=Cout⊕Sub=0

无符号整数减法运算中，用CF表示借位，CF=$\overline{\text{Cout}}$



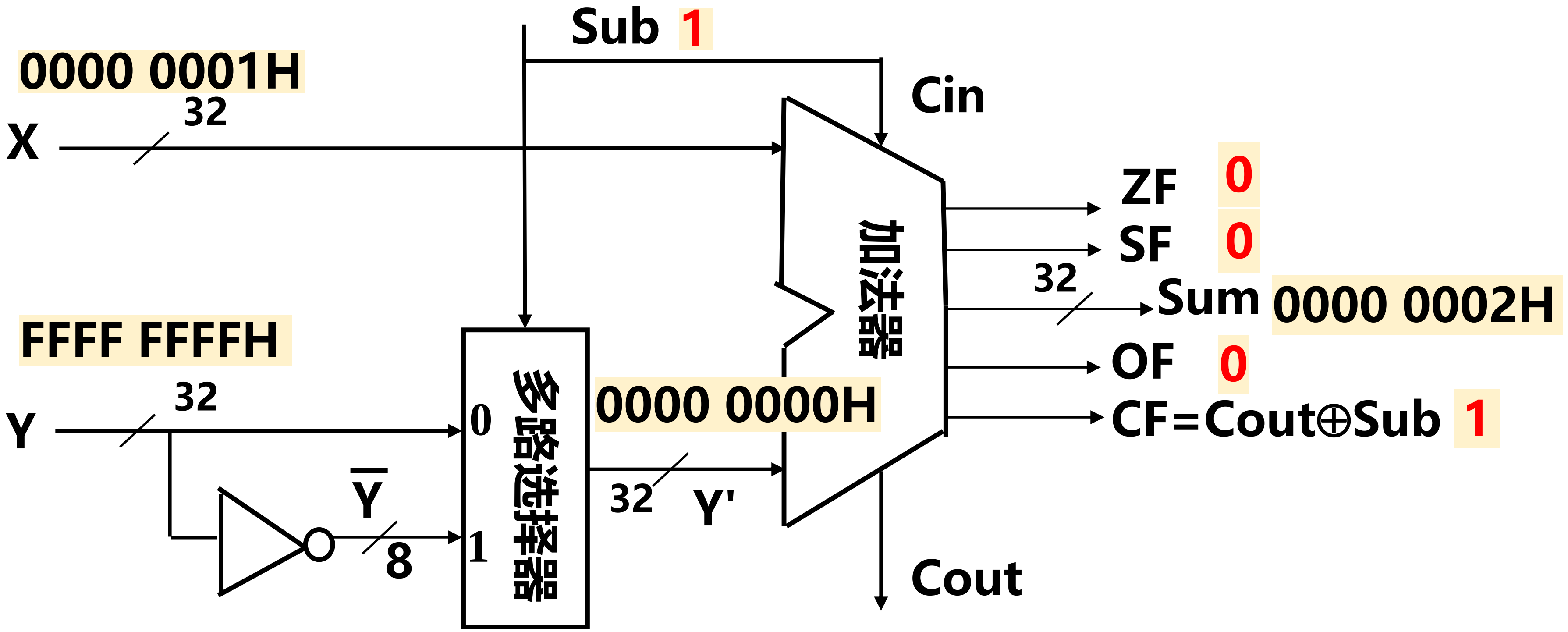
整数加减运算结果的溢出问题

```
  0000 0001H        0000 0001H
 -FFFF FFFFH        0000 0000H
 -----------      +          1
  FFFF FFFEH      ------------
                    0000 0002H
```

1H-FFFF FFFFH
=1H+（-FFFF FFFFH+2^{32}）
因为不够减，所以2^{32}被用于借位

Cout=0　　CF=Cout⊕Sub=1

无符号整数减法运算中，用CF表示借位，CF=$\overline{\text{Cout}}$



整数加减运算结果的溢出问题

无符号整数加减运算的总结：

无符号整数的加法运算中，CF表示进位，$CF = Cout \oplus \overset{0}{Sub} = Cout$

无符号整数的减法运算中，CF表示借位，$CF = Cout \oplus \underset{1}{Sub} = \overline{Cout}$

CF对带符号整数的运算无意义。

当A和B都是无符号整数时，标志位信息的应用

	CF	ZF	说明
A-B		1	A=B
A-B	1	0	A<B
A-B	0	0	A>B

整数加减运算结果的溢出问题

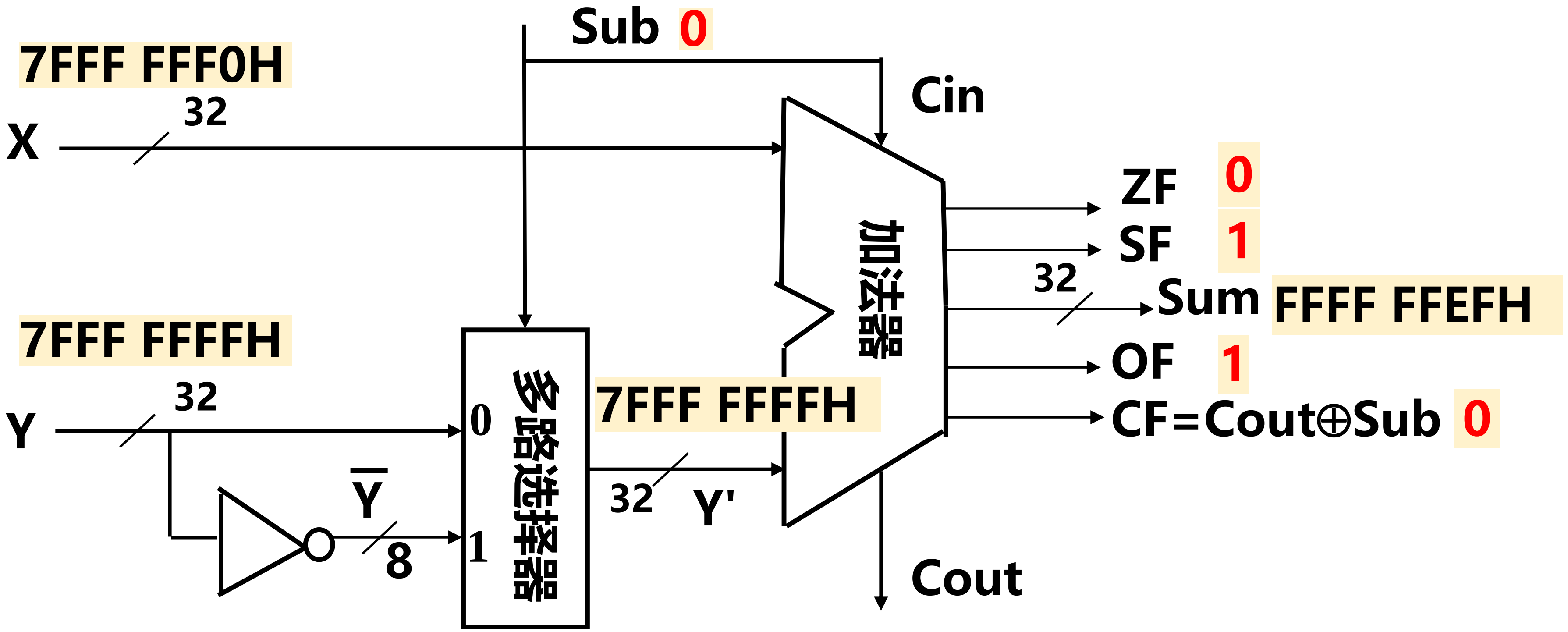
7FFF FFF0H	0111 1111 1111 1111 1111 1111 1111 0000
+7FFFFFFFH	+0111 1111 1111 1111 1111 1111 1111 1111
FFFF FFEFH	1111 1111 1111 1111 1111 1111 1110 1111

$$OF=\overline{X_{n-1}}\,\overline{Y'_{n-1}}Sum_{n-1}+X_{n-1}Y'_{n-1}\overline{Sum_{n-1}}=1$$

带符号整数加法运算中，用OF表示溢出。

int类型整数的表示范围是：

-0x8000 0000~0x7FFF FFFF



整数加减运算结果的溢出问题

(-7FFF FFF0H)-7FFF FFFFH

8000 0010H	1000 0000 0000 0000 0000 0000 0001 0000
8000 0000H	1000 0000 0000 0000 0000 0000 0000 0000
+ 1	+ 1
1 0000 0011H	10000 0000 0000 0000 0000 0000 0001 0001

$OF=\overline{X_{n-1}}\,\overline{Y'_{n-1}}Sum_{n-1}+X_{n-1}Y'_{n-1}\overline{Sum_{n-1}}=1$

带符号整数加法运算中，用OF表示溢出。

int类型整数的表示范围是：

-0x8000 0000~0x7FFF FFFF

(-7FFF FFF0H)-7FFF FFFFH 结果的补码应该是：$-(7FFF\ FFFFH+7FFF\ FFFFH)+2^{32}$

但加法器上计算的是：

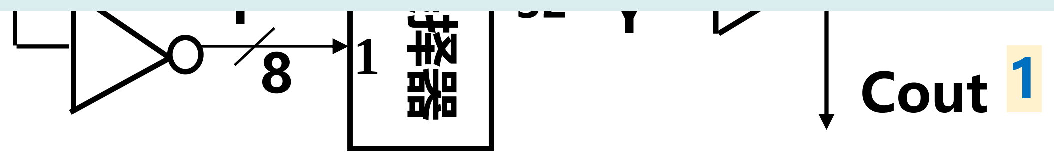
$(-7FFF\ FFF0H+2^{32})+(-7FFF\ FFFFH+2^{32})=-(7FFF\ FFFFH+7FFF\ FFFFH)+2^{32}+2^{32}$

带符号整数运算时，Cout值与溢出无关，与运算的结果也无关

$-8000\ 0000H+2^{32}=8000\ 0000H$

-(7FFF FFFFH+7FFF FFFFH) <-8000 0000H 时，超出了最小负数表示范围

$-(7FFF\ FFFFH+7FFF\ FFFFH)+2^{32}=0$ ••••••B



整数加减运算结果的溢出问题

带符号整数加减运算总结：

1. 带符号整数的加减法运算中，用OF判断溢出

   $$OF=\overline{X}_{n-1}\,\overline{Y'}_{n-1}Sum_{n-1}+X_{n-1}\,Y'_{n-1}\overline{Sum}_{n-1}=1$$

2. 从加减运算的角度来看：

   加法时，同号相加，和与两加数异号，则溢出；

   减法时，异号相减，差与被减数异号，则溢出。

当A和B都是带符号整数时，标志位信息的应用

	SF	OF	ZF	说明
A-B			1	A=B
A-B	1	0	0	$A<B$
A-B	1	1	0	$A>B$
A-B	0	0	0	$A>B$
A-B	0	1	0	$A<B$
A-B	SF!=OF and ZF==0			$A<B$
A-B	SF!=OF OR ZF==1			$A<=B$
A-B	SF==OF and ZF==0			$A>B$
A-B	SF==OF OR ZF==1			$A>=B$

总 结

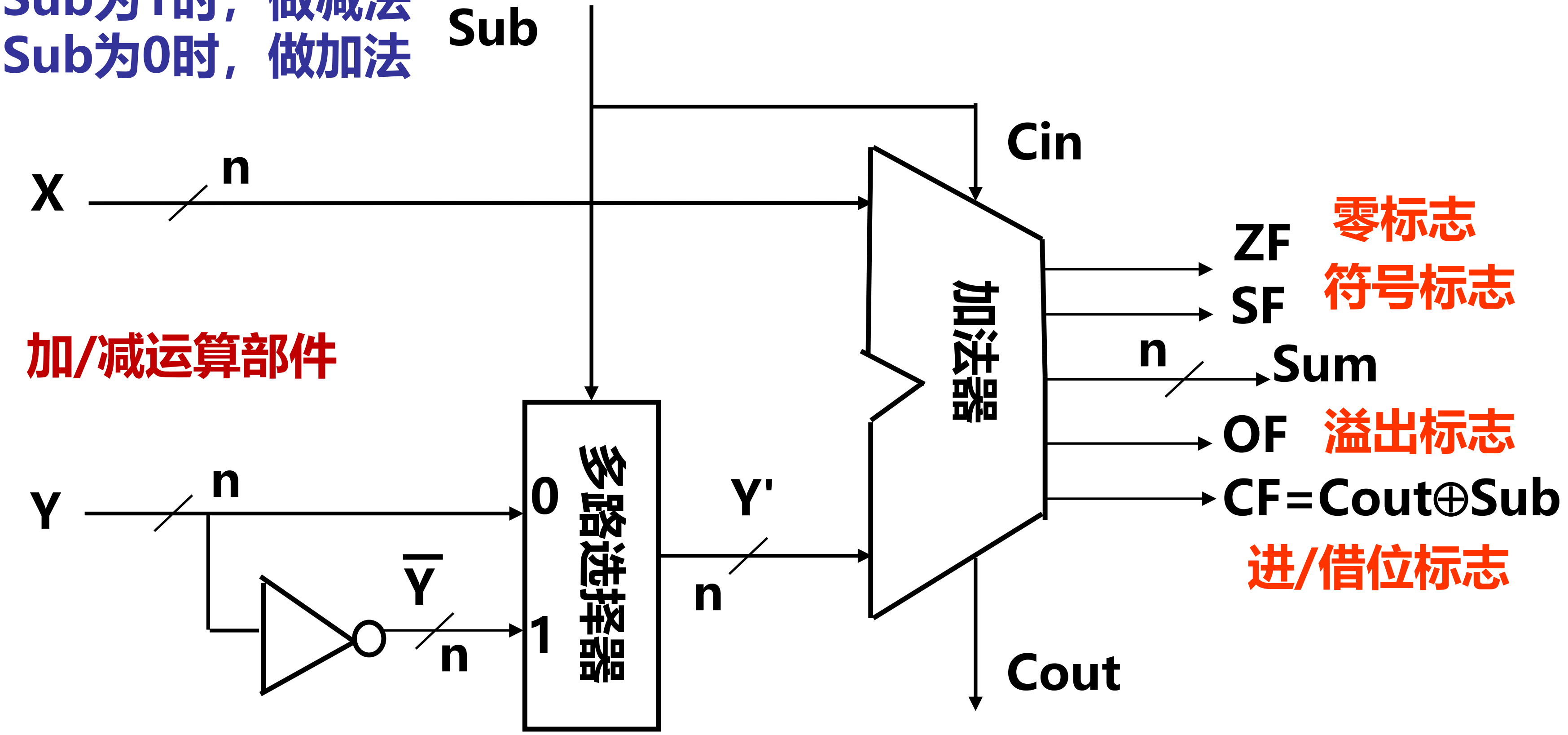
补码加减运算公式：

$$[x+y]_{补} = [x]_{补} + [y]_{补} \quad (\bmod 2^n)$$

$$[x-y]_{补} = [x]_{补} + [-y]_{补} \quad (\bmod 2^n)$$

当Sub为1时，做减法
当Sub为0时，做加法

加/减运算部件



谢谢！