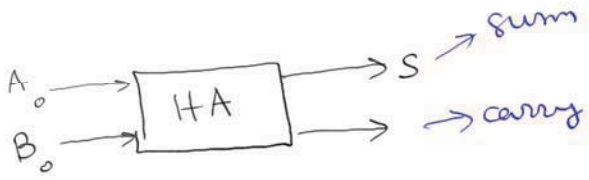
4.7 Binary Adders

Half Adder

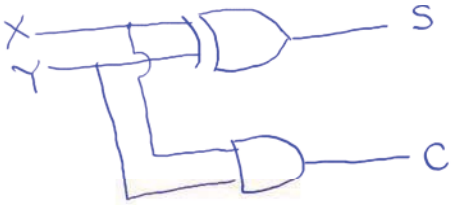


X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

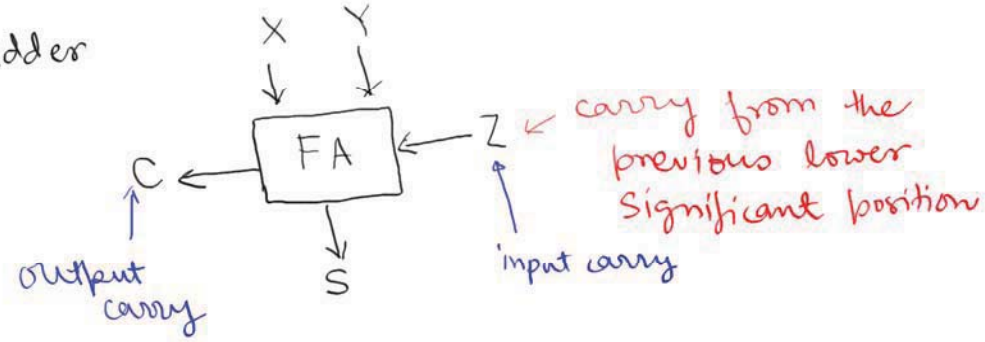
carry

$S = \bar{X}Y + X\bar{Y} = X \oplus Y$

$C = XY$



Full Adder



X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

See K maps for C & S in the book

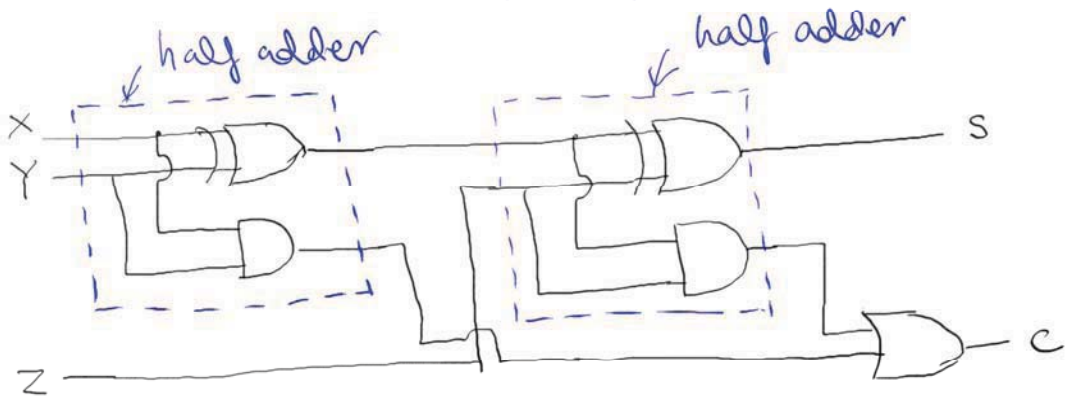
$$S = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ$$

$$C = XY + XZ + YZ$$

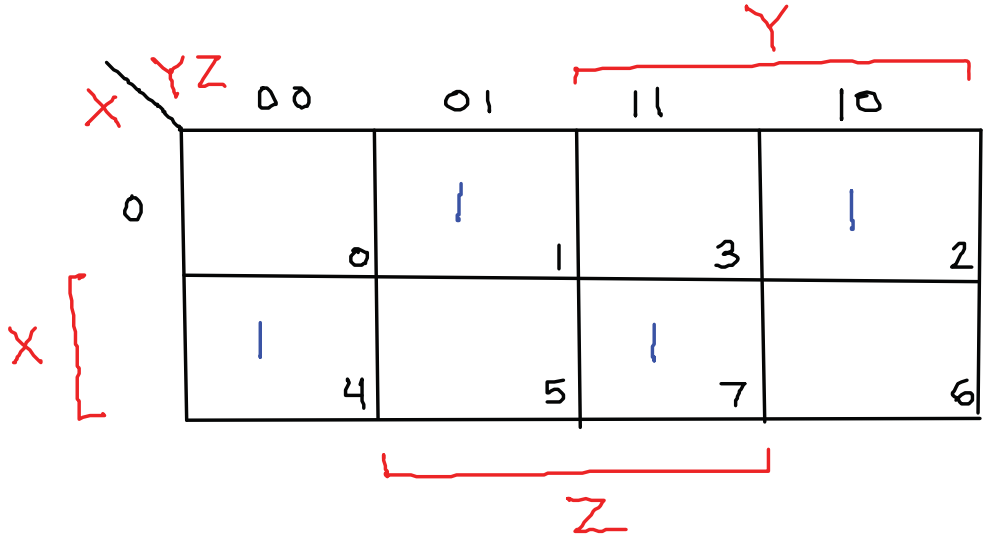
↓

$$S = X \oplus Y \oplus Z$$

$$C = XY + Z(X \oplus Y)$$

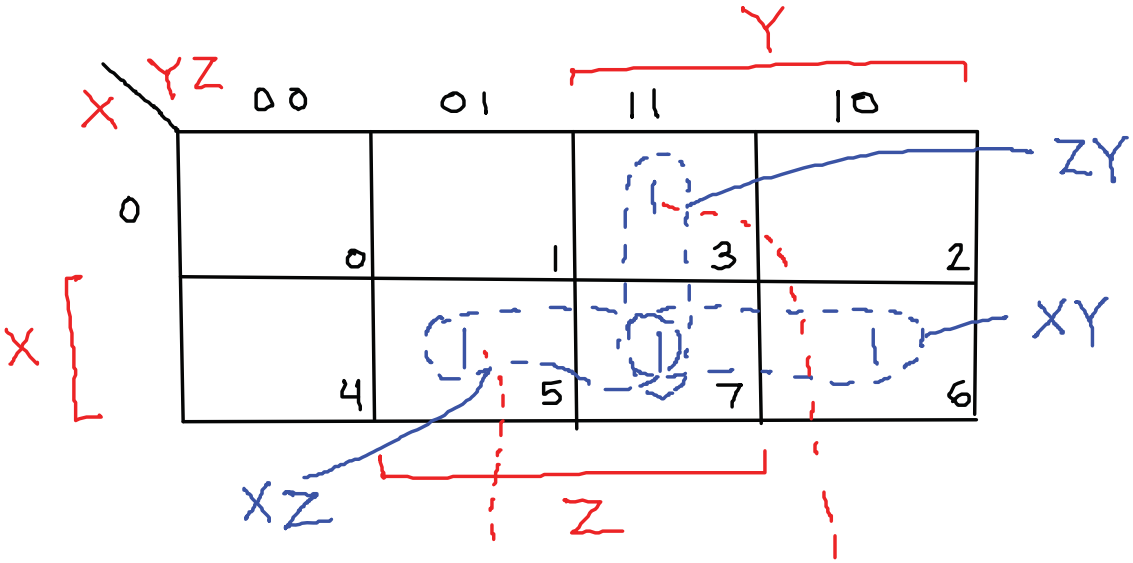


K-Maps for full adder



$$S = \Sigma(1,2,4,7) = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ$$

$$= X \oplus Y \oplus Z$$



$$C = XY + XZ + YZ$$

$$= XY + X\bar{Y}Z + \bar{X}YZ$$

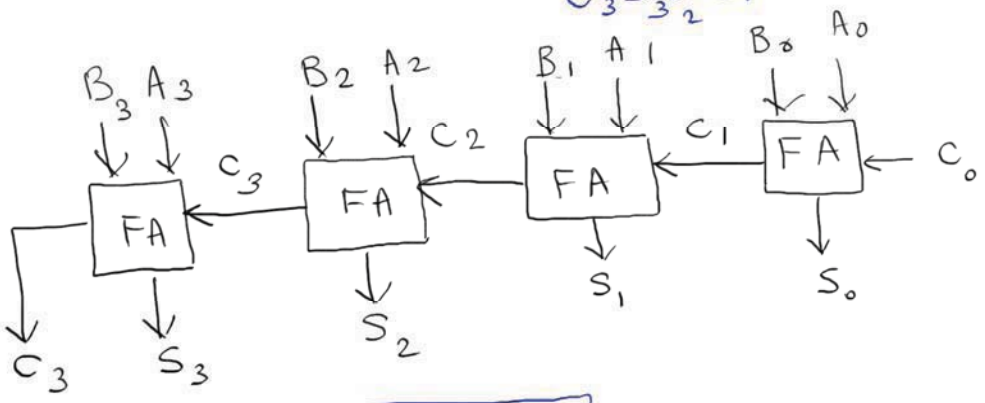
$$= XY + Z(X\bar{Y} + \bar{X}Y)$$

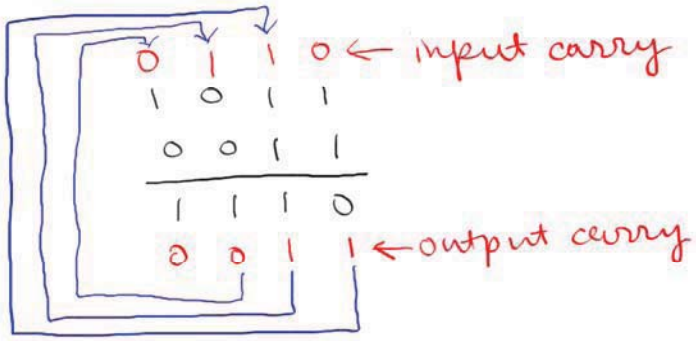
$$= XY + Z(X \oplus Y)$$

Binary Ripple Carry Adder

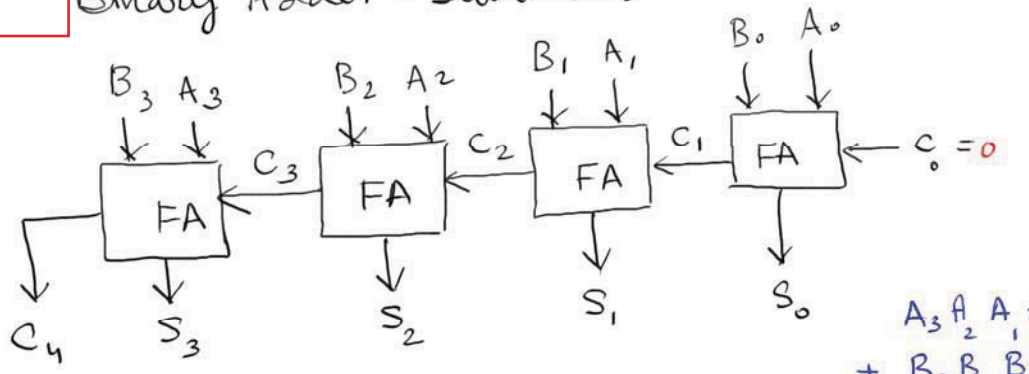
Four bit adder

$A_3 A_2 A_1 A_0$
$+ \; B_3 B_2 B_1 B_0$
$C_3 S_3 S_2 S_1 S_0$



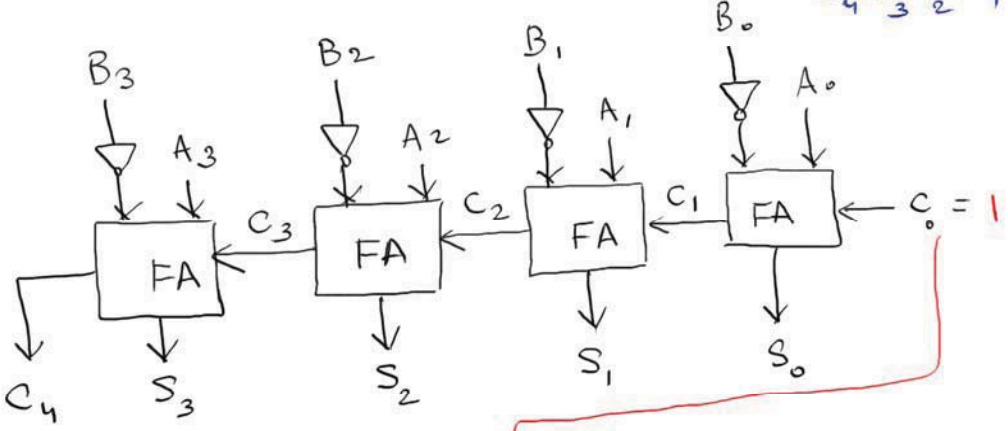


Binary Adder - Subtractors



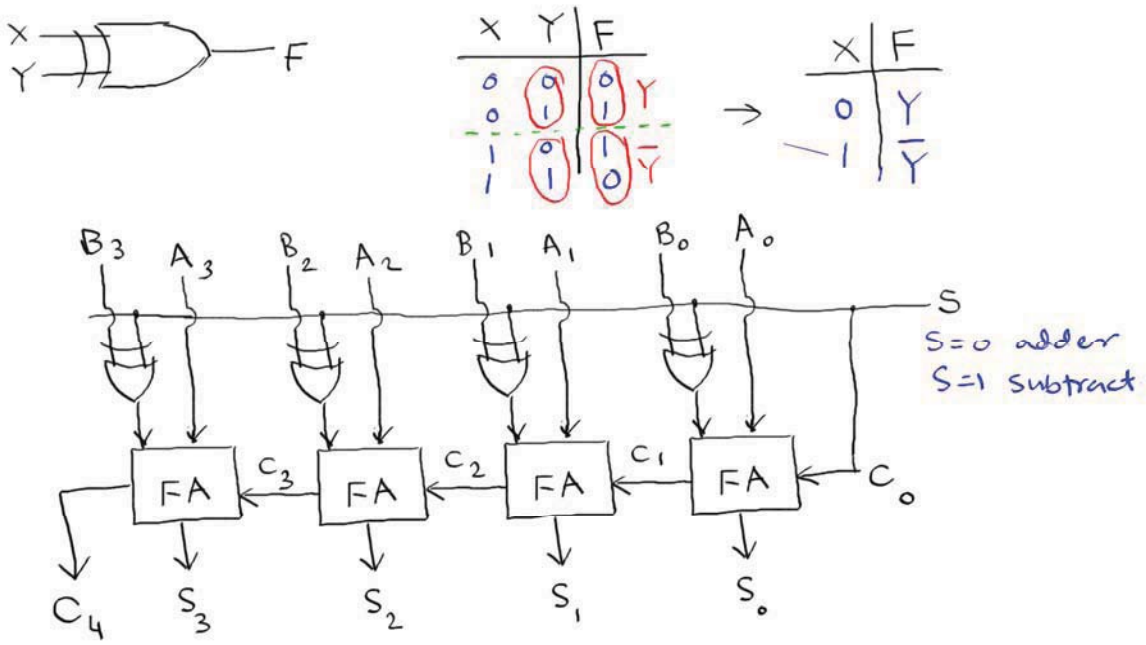
4 Bit Adder

$$\begin{array}{r} A_3\ A_2\ A_1\ A_0 \\ +\ B_3\ B_2\ B_1\ B_0 \\ \hline C_4\ S_3\ S_2\ S_1\ S_0 \end{array}$$



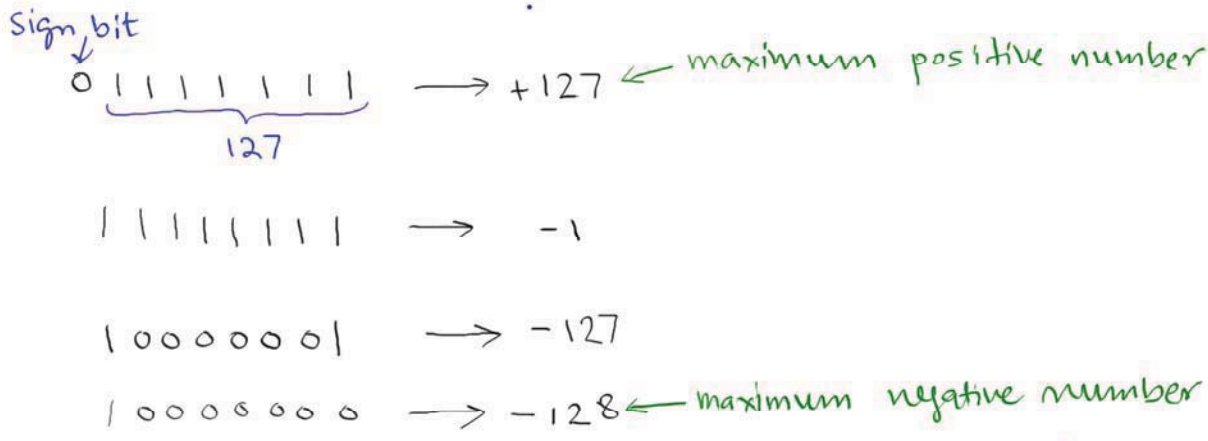
$\bar{B}_3\ \bar{B}_2\ \bar{B}_1\ \bar{B}_0 + 1$ = 2's complement of $B_3\ B_2\ B_1\ B_0$

Combining addition & subtraction in one circuit



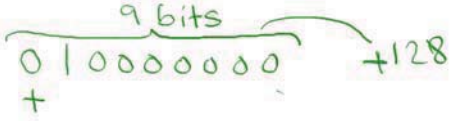
overflow in 2's complement representation

Assume 8 bits



Note: complementing the max. negative number → overflow
2's complement of 10000000 should give +128

but +128 needs more than 8 bits!



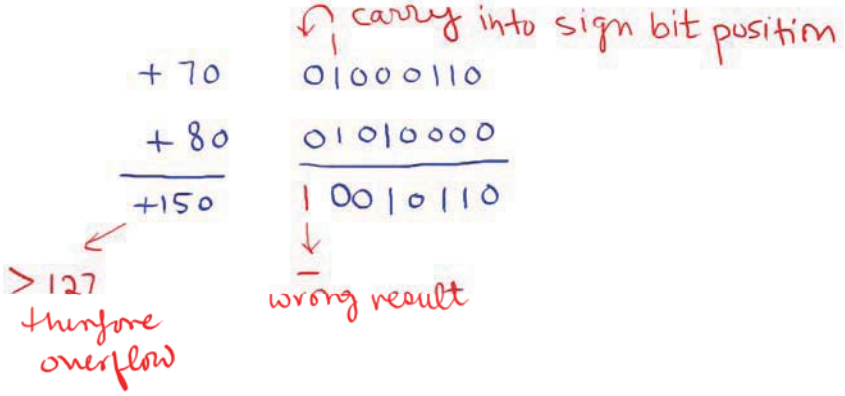
keep this "special case" in mind when converting the maximum negative number

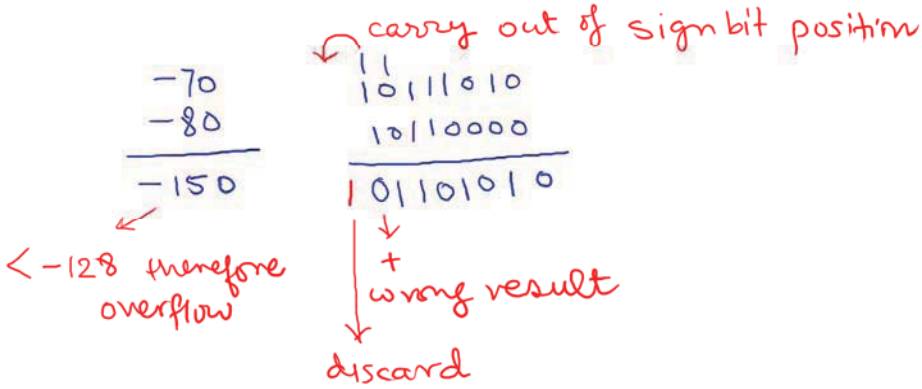
8 bits can store a number if $-128 \leq \text{number} \leq +127$

otherwise there will be overflow

Overflow in 2' complement representation

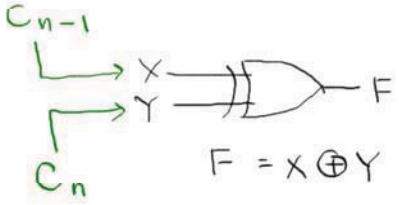
Addition: overflow can happen only if the two numbers have the same sign





overflow if

carry into sign bit position ≠ carry out of sign bit pos.



X	Y	F
0	0	0
0	1	0
1	0	0
1	1	1 → overflow

1 0 0 0 0 0 0 0 ⟶ −128

2's complement = 1's complement + 1

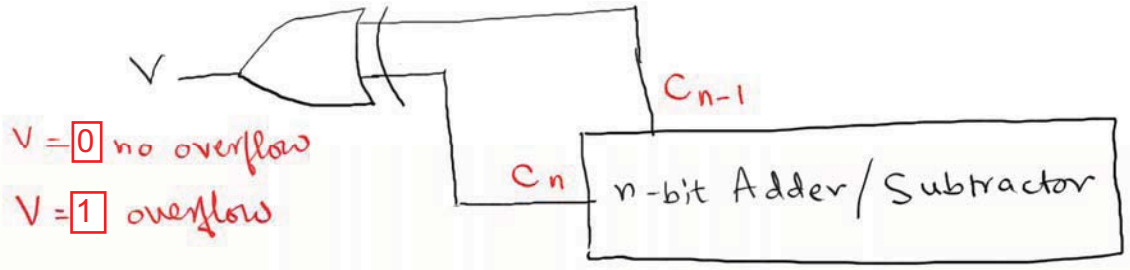
↱ → carry into signed bit position.

0 1 1 1 1 1 1 1 ⟶ 1's complement

+ 1

1 0 0 0 0 0 0 0

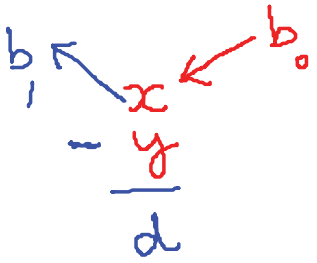
overflow

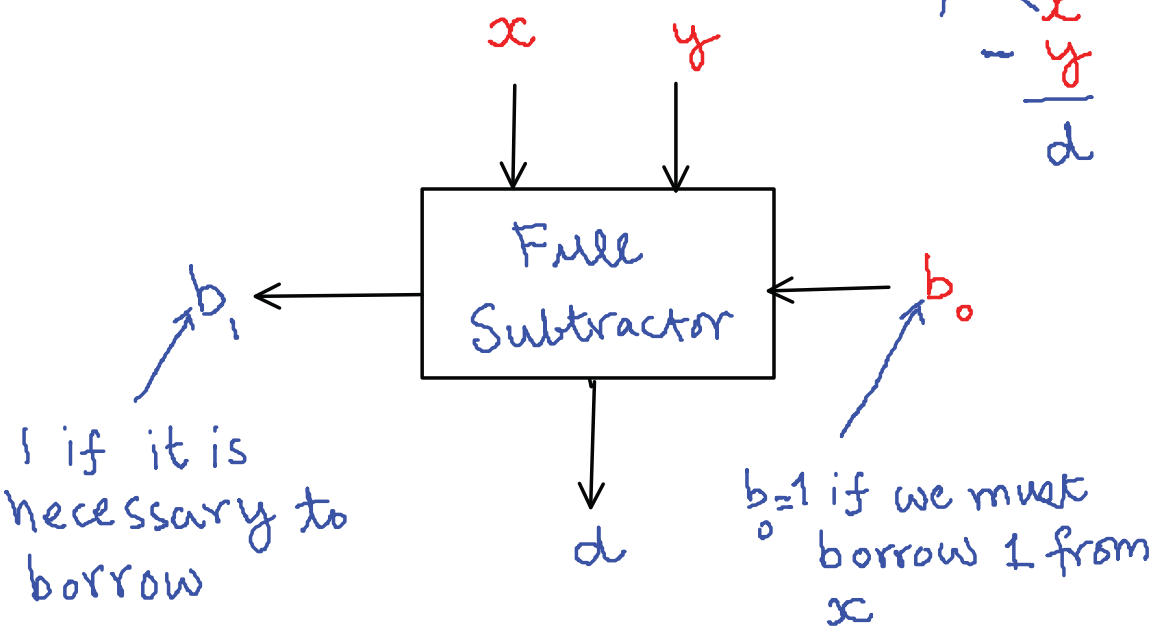


V = 0 no overflow

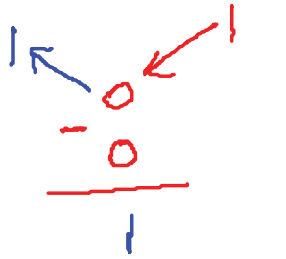
V = 1 overflow

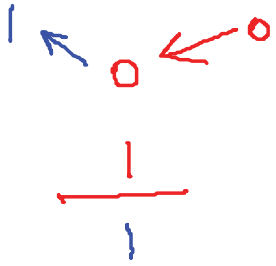
Full Subtractor

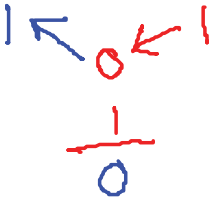




x	y	b_0	b_1	d
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1







Subtractor