1

2's complement

Unsigned 4-bit number

0-15 ← $2^4 = 16$ → (2^3-1) to (-2^3) = 7, 6, 5, ... 0, ... -7, -8.

D	8 4 2 1	-8 4 2 1	
0	0000	1000	-8
1	0001	1001	-7
2	0010	1010	-6
3	0011	1011	-5
4	0100	1100	-4
5	0101	1101	-3
6	0110	1110	-2
7	0111	1111	-1

1001

$1 \times -8 + 1 \times 1 = x$

$-8 \times 1 =$

-8

3 + (-5)

0011 +3
1011 -5

$\boxed{1110}$ -2

6 + (-2)

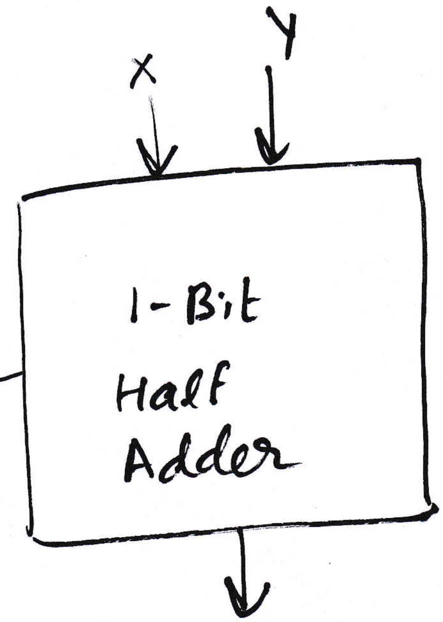
0110
1110

1 $\boxed{0100}$

Adders

- → one Bit Half Adder
- → one bit full **Adder**

X	Y	Z	Cout
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

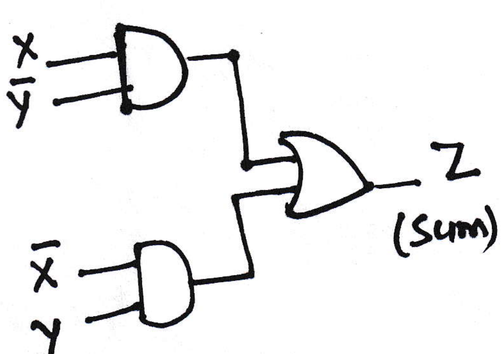


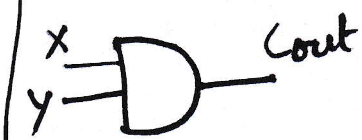
→ Logic to add 1-bit, 2-bit, 4-bit, 8-bit, ... remains the same

→ Only diff. is that circuit becomes bigger

$Z = X \cdot \bar{Y} + \bar{X} \cdot Y$

$C_{out} = X \cdot Y$





	3	0011
+	4	0100
	7	0111

1

0

$\boxed{1}$

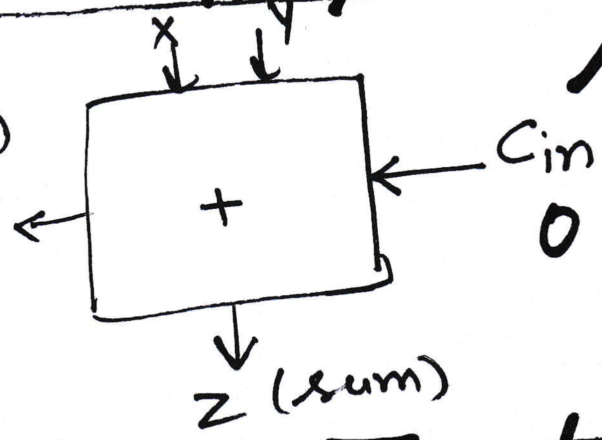
0011
0010

0101

0100
+0001

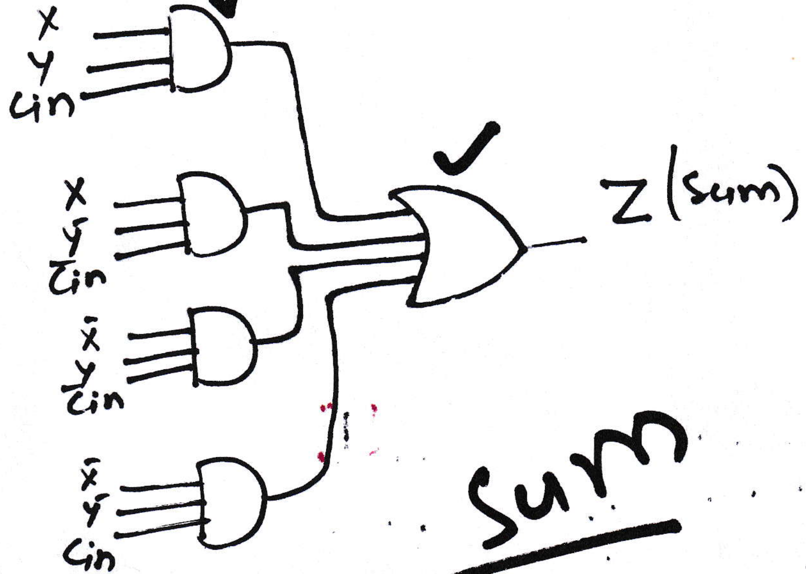
0101

One-bit Full Adder

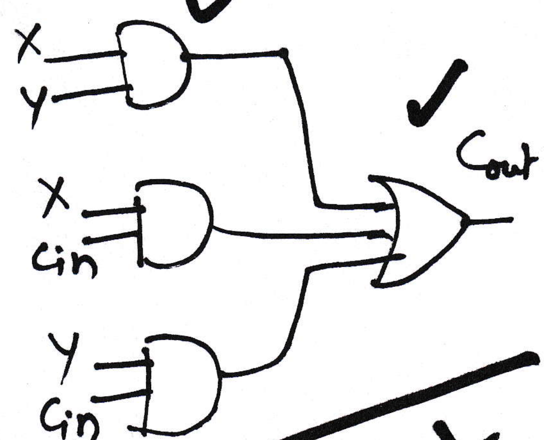


$$Z = X \cdot Y \cdot C_{in} + X \cdot \bar{Y} \cdot \bar{C}_{in} + \bar{X} \cdot Y \cdot \bar{C}_{in} + \bar{X} \cdot \bar{Y} \cdot C_{in}$$

$$C_{out} = \cancel{X} \cdot Y + X \cdot C_{in} + Y \cdot C_{in}$$



sum

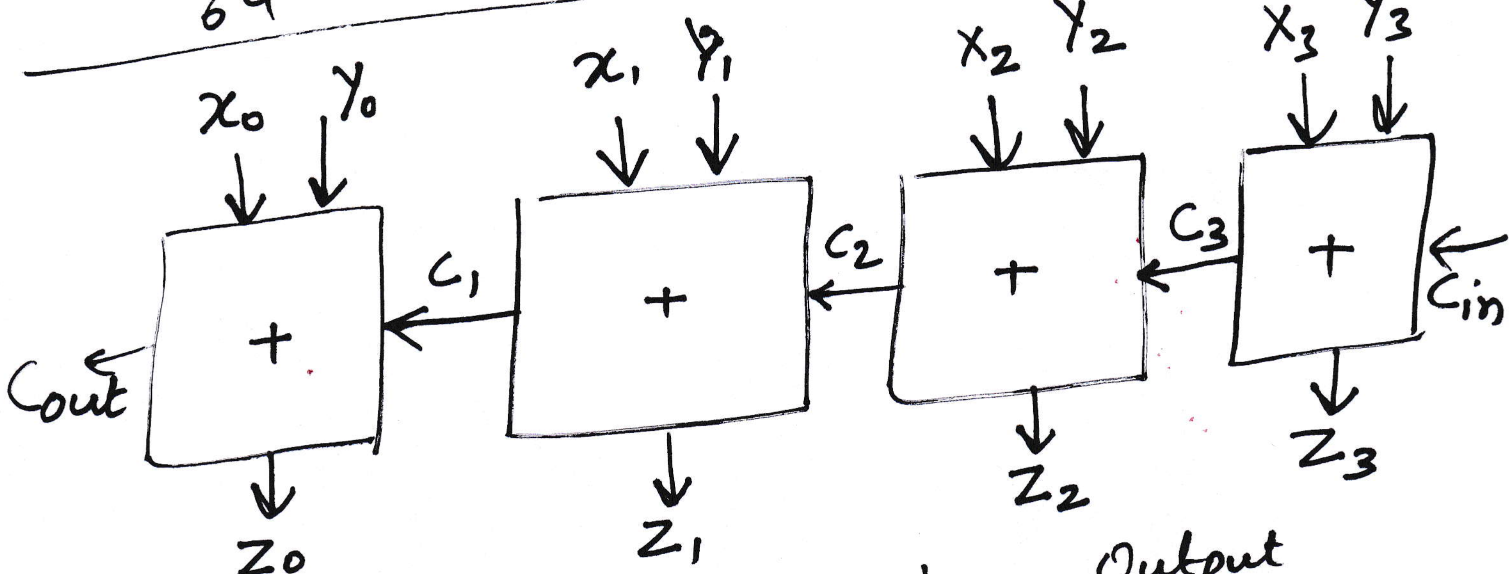


carry

①

Processor cycle

4-bit adder - 4 steps — 4
8-bit adder - 8 steps — 8
16 " " - 16 " — 16
32 " " - 32 " — 32
64 " " - 64 " — 64 [very big]



Input

$X = X_0 X_1 X_2 X_3$

$Y = Y_0 Y_1 Y_2 Y_3$

Cin → Initially zero

Output

$Z = Z_0 Z_1 Z_2 Z_3$

Cout.

$\boxed{+}$ → one bit full adder.

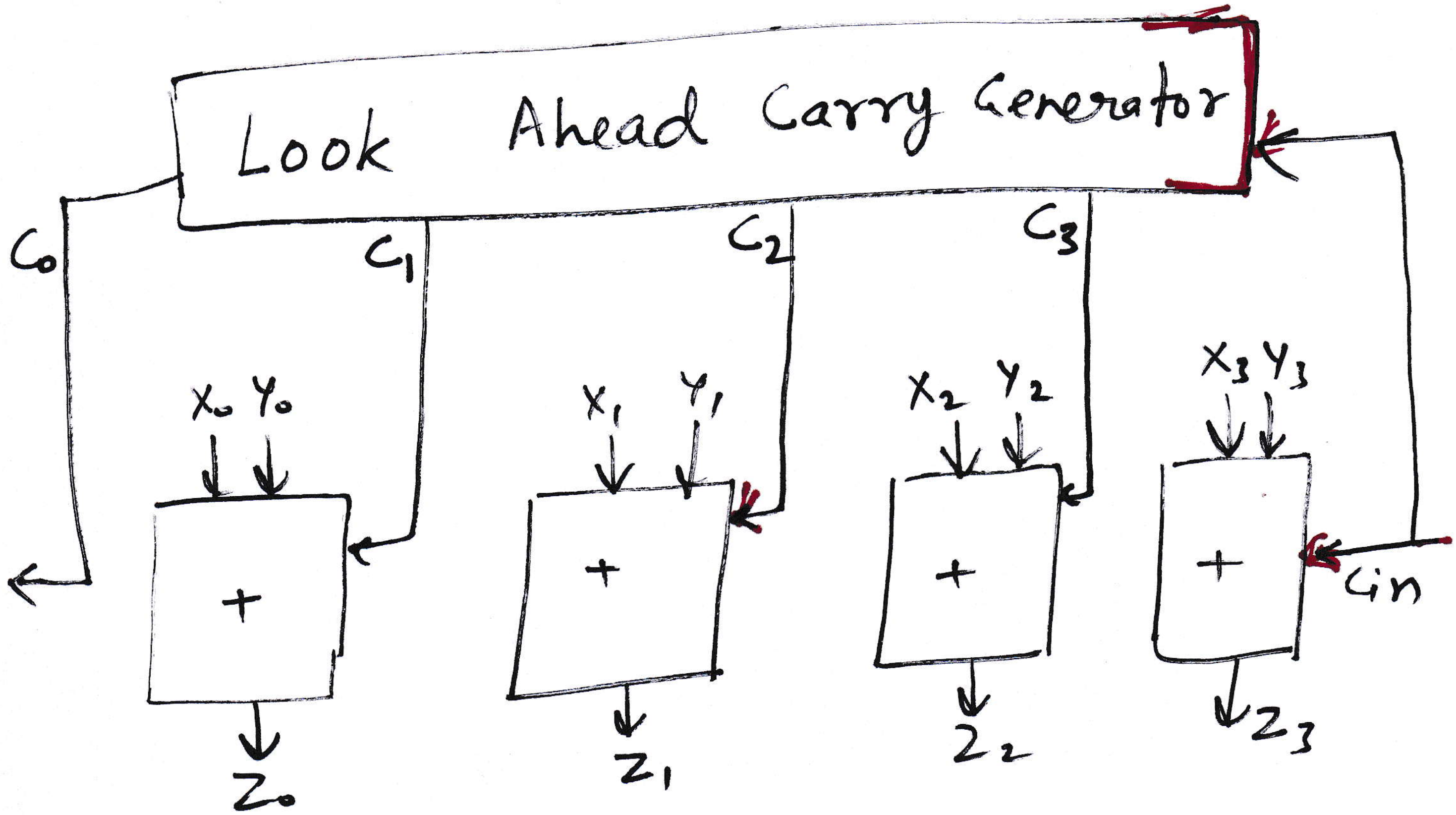
→ 3) also serial adder
4) slow adder

$$
\begin{array}{ccccc}
 & & 1 & & \\
 & 0 & 0 & 1 & 1 \\
+ & 0 & 0 & 1 & 0 \\
\hline
 & 0 & 1 & 0 & 1 \\
 & ④ & ③ & ② & ①
\end{array}
$$

1) Carry is rippled from one stage to the other
2) Ripple-carry

② ①

Parallel Adder



(3)

$$\boxed{C = X \cdot Y + X \cdot C_{in} + Y \cdot C_{in}}$$

(2)

$$C_3 = \underbrace{X_3 \cdot Y_3} + X_3 \cdot C_{in} + Y_3 \cdot C_{in}$$

=D =D =D

$$C_3 = \underbrace{X_3 \cdot Y_3}_{g_3} + C_{in} \cdot \underbrace{(X_3 + Y_3)}_{p_3}$$

generate $\leftarrow g_3$ $\qquad$ $p_3 \rightarrow$ propagate

$$\boxed{C_3 = g_3 + p_3 \cdot C_{in}} \quad \text{--- (1)}$$

$$C_2 = g_2 + p_2 \cdot C_3$$

$$\Rightarrow C_2 = g_2 + p_2 g_3 + p_2 p_3 C_{in} \quad \text{--- (2)}$$

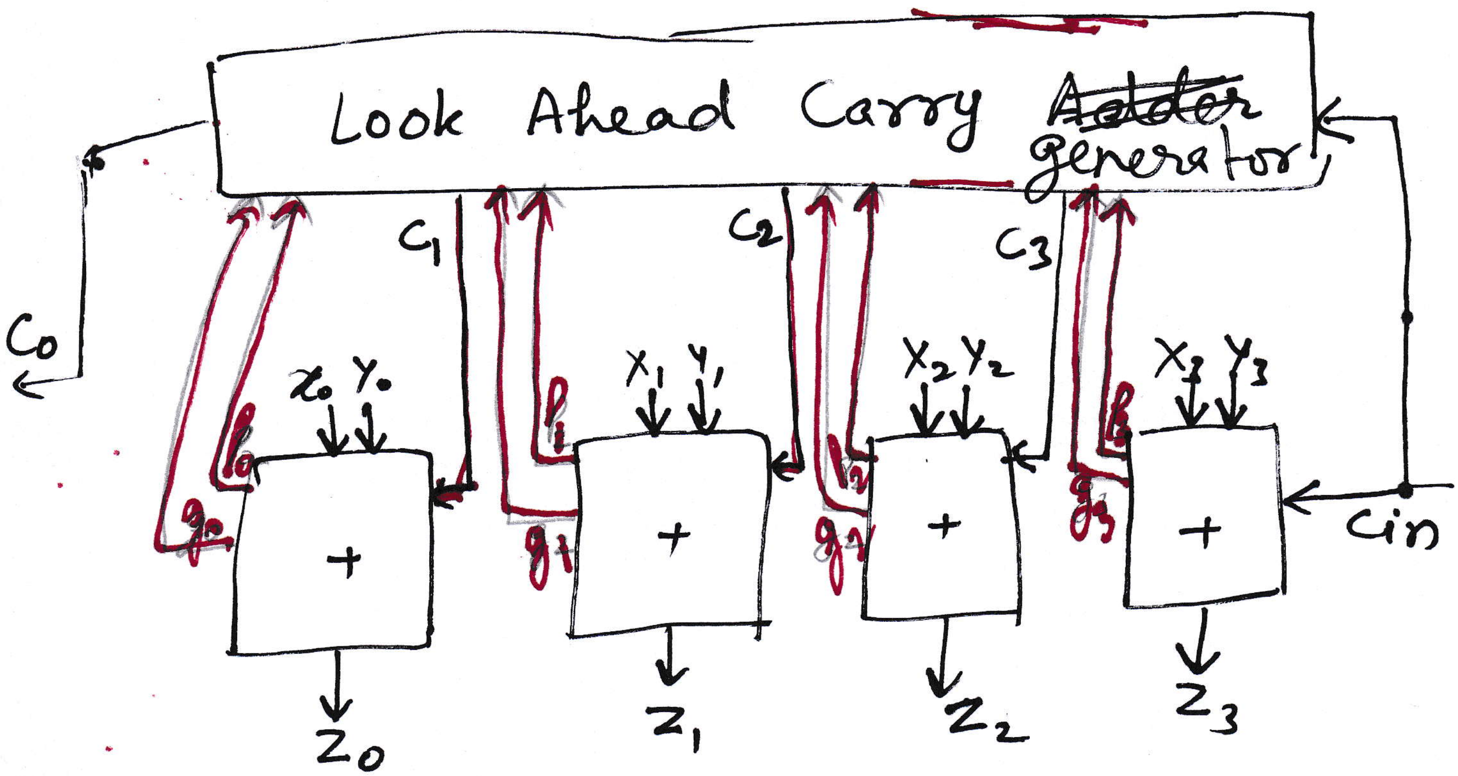
$$C_1 = g_1 + p_1 \cdot C_2$$

$$C_1 = g_1 + p_1 g_2 + p_1 p_2 g_3 + p_1 p_2 p_3 C_{in} \quad \text{--- (3)}$$

$$C_0 = g_0 + p_0 C_1$$

$$C_0 = g_0 + p_0 g_1 + p_0 p_1 g_2 + p_0 p_1 p_2 g_3 + p_0 p_1 p_2 p_3 C_{in}$$

Parallel Adder



→ We will predict (look ahead) all the carry

→ Using all X-bits, Y-bits, and Cin, we will predict C_3, C_2, C_1, C_0 [Cout]

→ ~~This should be~~ done in

(5)

→ All work done in just 3 cycle

→ ~~One~~ One cycle will produce all g's & p's.

→ One cycle will produce all carries independently using g's, p's, and Cin.

→ Last cycle, ~~ad~~ all the adder circuits will produce the result.

⑥

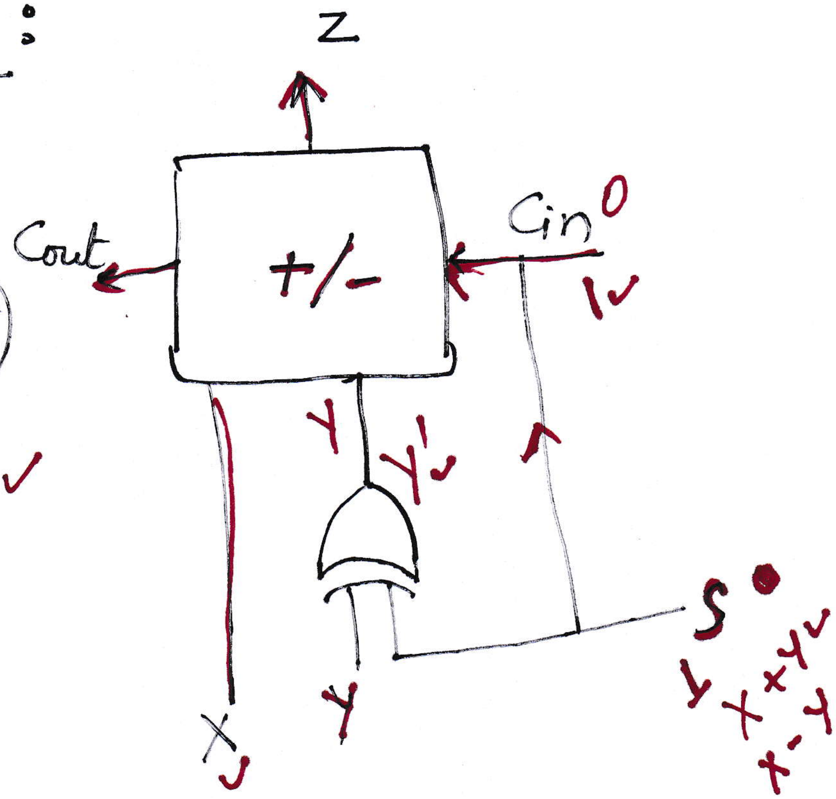
Substraction :

$X - Y$

$= X + \begin{pmatrix} \text{2's complement} \\ \text{of } y \end{pmatrix}$

$= X + \begin{pmatrix} \text{1's complement } y \\ + 1 \end{pmatrix}$

$= X + (y' + 1)$



XOR

$$\begin{array}{ll} s=0 & \left[\begin{array}{cc|c} 0 & 0 & 0 \\ 0 & 1 & 1 \end{array}\right] y \\ s=1 & \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}\right] y' \end{array}$$