

# Digital Logic IV

Michael C. Hackett

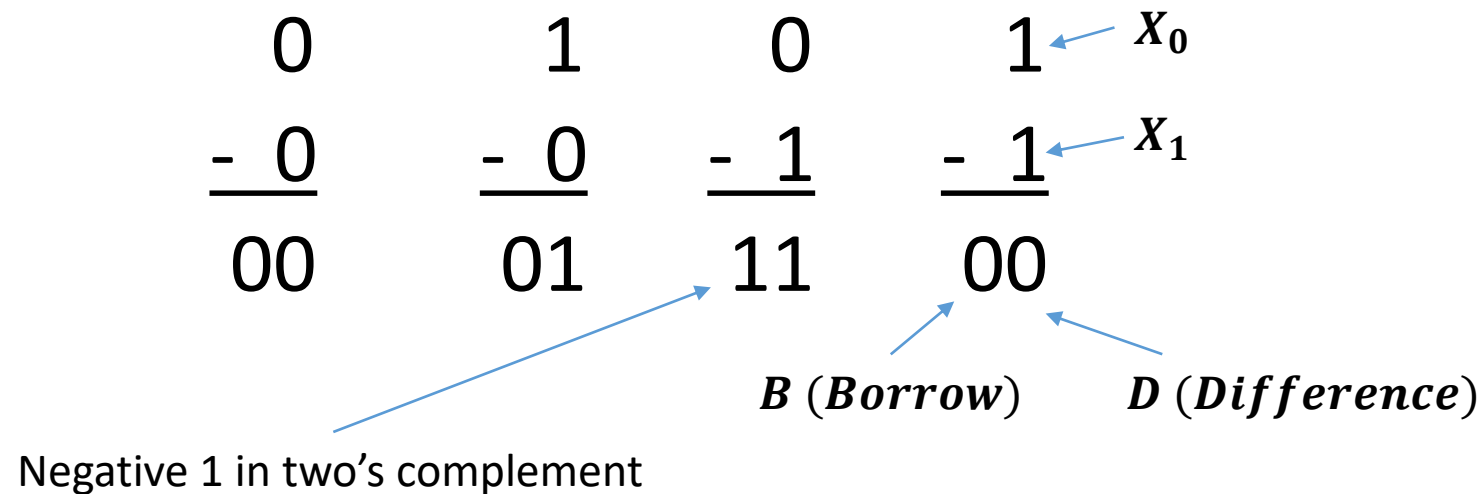
Assistant Professor, Computer Science

# Lecture Topics

- Combinational Circuits
  - Subtractors
    - Half Subtractor
    - Full Subtractor
  - Multipliers

# Subtractors

- **Subtractors** are combinational logic circuits capable of performing subtraction
- A **half subtractor** has two inputs (the two digits to subtract) and two outputs (the difference and the borrow).



# Half Subtractor

- Half subtractor truth table:

$X_1$	$X_0$	$B$	$D$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	0	0

$$\begin{array}{r} 0 \\ - 0 \\ \hline 00 \end{array} \quad \begin{array}{r} 1 \\ - 0 \\ \hline 01 \end{array} \quad \begin{array}{r} 0 \\ - 1 \\ \hline 11 \end{array} \quad \begin{array}{r} 1 \\ - 1 \\ \hline 00 \end{array}$$

$B$  (*Borrow*)       $D$  (*Difference*)

Diagram illustrating the half subtractor operation for the case  $X_1=1, X_0=1$ . The inputs are  $X_1$  and  $X_0$ . The output is the Borrow ( $B$ ) and the Difference ( $D$ ). The diagram shows the subtraction of  $1$  from  $1$  at the  $X_0$  position, resulting in a Borrow of  $1$  and a Difference of  $0$ .

# Half Subtractors

SOP Expressions:

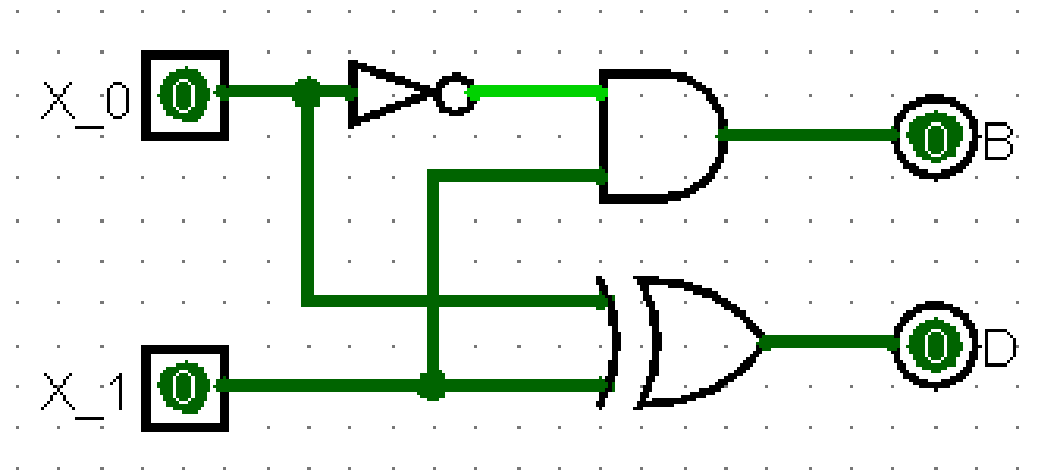
$$B = X_1 \overline{X_0}$$

$$D = \overline{X_1} X_0 + X_1 \overline{X_0} = X_1 \oplus X_0$$

$X_1$	$X_0$	$B$	$D$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	0	0

# Half Subtractor

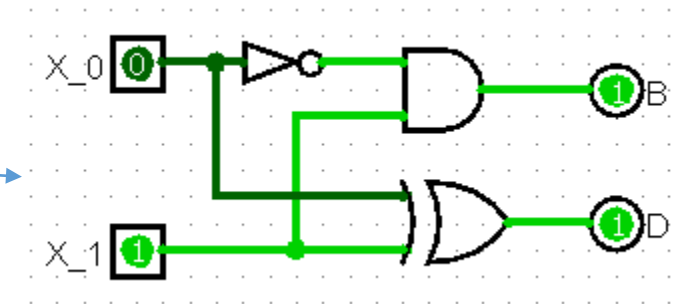
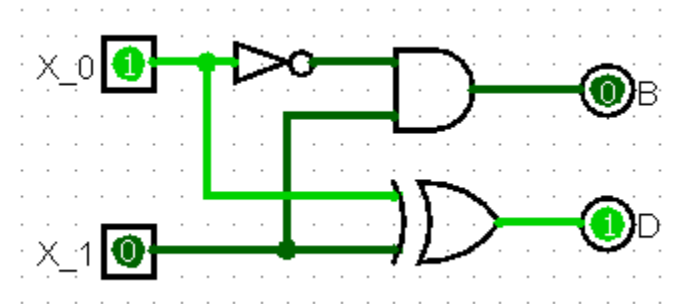
Half Subtractor Logic Circuit:



# Half Subtractor

Half Subtractor Logic Circuit:

$X_1$	$X_0$	$B$	$D$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	0	0



# Full Subtractors

- A **full subtractor** has three inputs (the two digits to subtract, plus a value *borrowed in*) and two outputs (the difference and the borrow).

The diagram shows a truth table for a full subtractor. The inputs are labeled  $X_1$  and  $X_0$  above the second and third columns of the table, respectively. The output for the difference is labeled  $D$  (Difference) to the right of the fourth column. The output for the borrow is labeled  $B_{OUT}$  (Borrow Out) above the fifth column. The input for the borrow-in is labeled  $B_{IN}$  (Borrow In) to the left of the first column. Blue arrows point from each label to its corresponding column in the table.

$B_{IN}$	$X_1$	$X_0$	$D$	$B_{OUT}$
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



# Full Subtractors

SOP Expressions:

$$B_{OUT} = \overline{B_{IN}} \overline{X_1} X_0 + \overline{B_{IN}} X_1 \overline{X_0} + \overline{B_{IN}} X_1 X_0 + B_{IN} X_1 X_0$$

$$D = \overline{B_{IN}} \overline{X_1} X_0 + \overline{B_{IN}} X_1 \overline{X_0} + B_{IN} \overline{X_1} \overline{X_0} + B_{IN} X_1 X_0$$

$B_{IN}$	$X_1$	$X_0$	$B_{OUT}$	$D$
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

# Full Subtractors

Simplifying:

$$B_{OUT} = \overline{B_{IN}} \overline{X_1} X_0 + \overline{B_{IN}} X_1 \overline{X_0} + \overline{B_{IN}} X_1 X_0 + B_{IN} X_1 X_0$$

$$B_{OUT} = \overline{B_{IN}} X_1 + \overline{B_{IN}} X_0 + X_1 X_0$$

		$X_0$	
		0	1
$B_{IN} \ X_1$	00	0 000	1 001
	01	1 010	1 011
	11	0 110	1 111
	10	0 100	0 101

# Full Subtractors

Simplifying:

$$D = \underbrace{\overline{B_{IN}} \overline{X_1} X_0 + \overline{B_{IN}} X_1 \overline{X_0}}_{\overline{B_{IN}} (X_0 \oplus X_1)} + \underbrace{B_{IN} \overline{X_1} \overline{X_0} + B_{IN} X_1 X_0}_{B_{IN} (X_0 \odot X_1)}$$

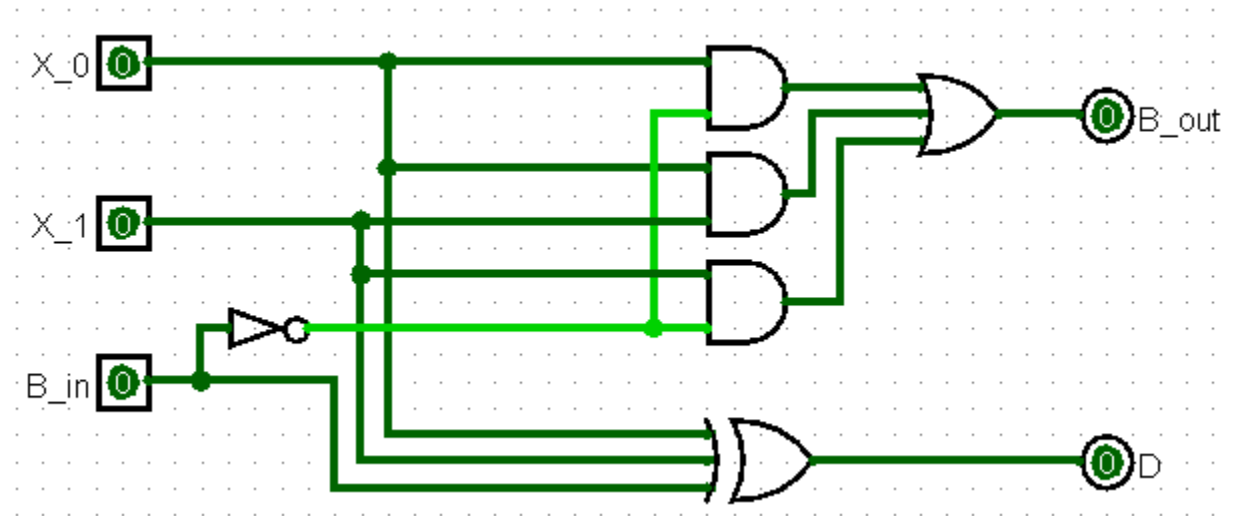
$$D = \overline{B_{IN}} (X_0 \oplus X_1) + B_{IN} (X_0 \odot X_1)$$

$$D = \underbrace{\overline{B_{IN}} (X_0 \oplus X_1) + B_{IN} (\overline{X_0 \oplus X_1})}_{\overline{B_{IN}} \oplus (X_0 \oplus X_1)} \quad \bar{X}Y + X\bar{Y} = X \oplus Y$$

$$D = B_{IN} \oplus (X_0 \oplus X_1) = B_{IN} \oplus X_0 \oplus X_1$$

# Full Subtractor

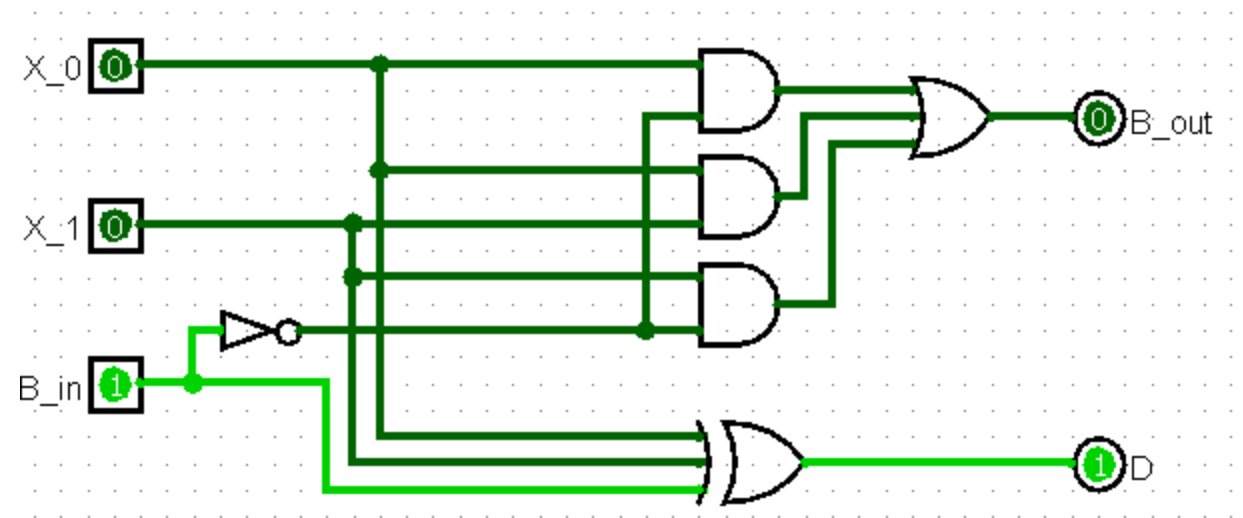
Full Subtractor Logic Circuit:



# Full Subtractor

Full Subtractor Logic Circuit:

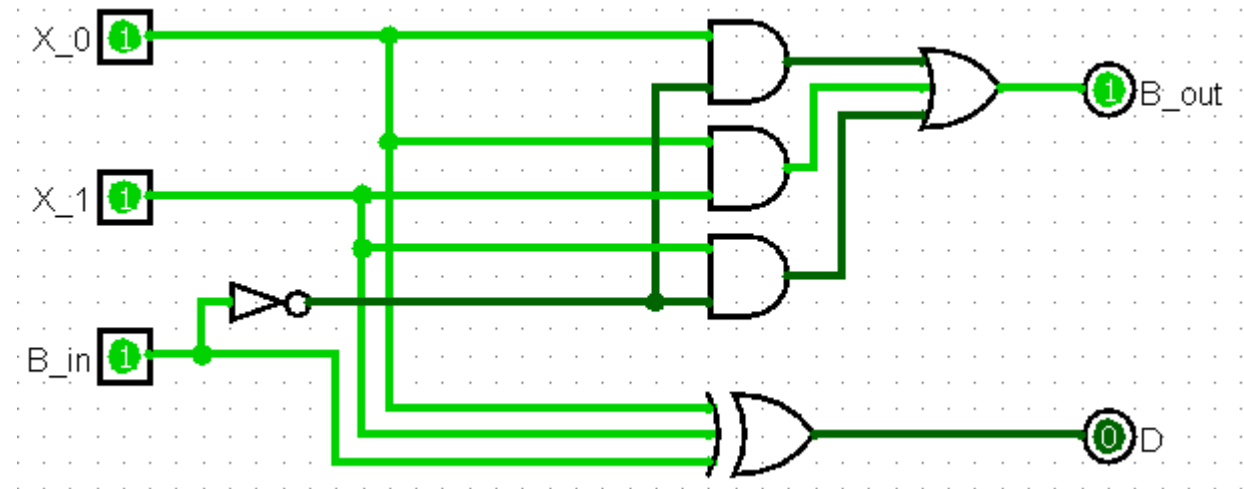
$B_{IN}$	$X_1$	$X_0$	$B_{OUT}$	$D$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



# Full Subtractor

Full Subtractor Logic Circuit:

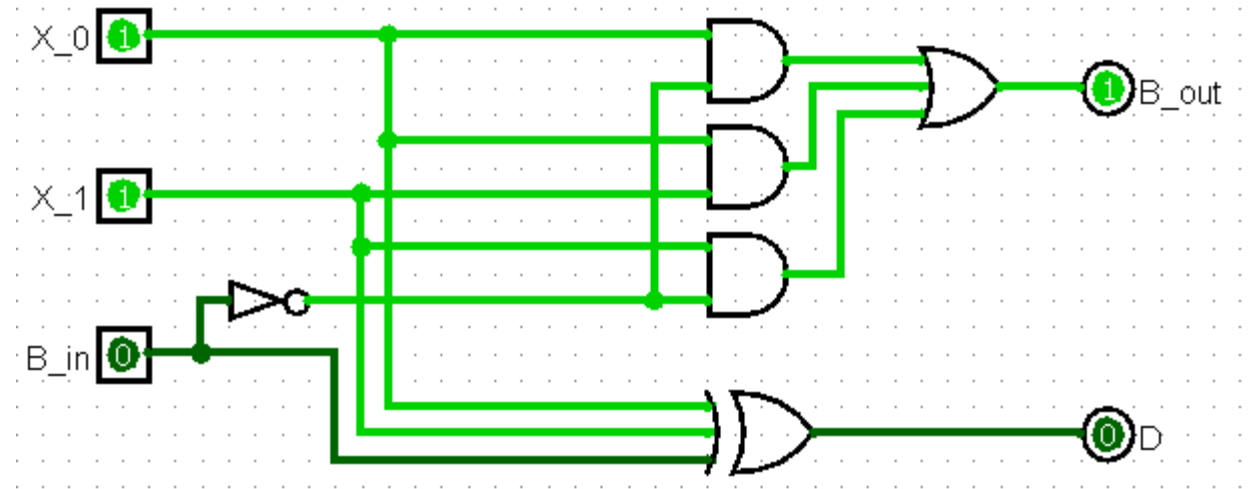
$B_{IN}$	$X_1$	$X_0$	$B_{OUT}$	$D$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



# Full Subtractor

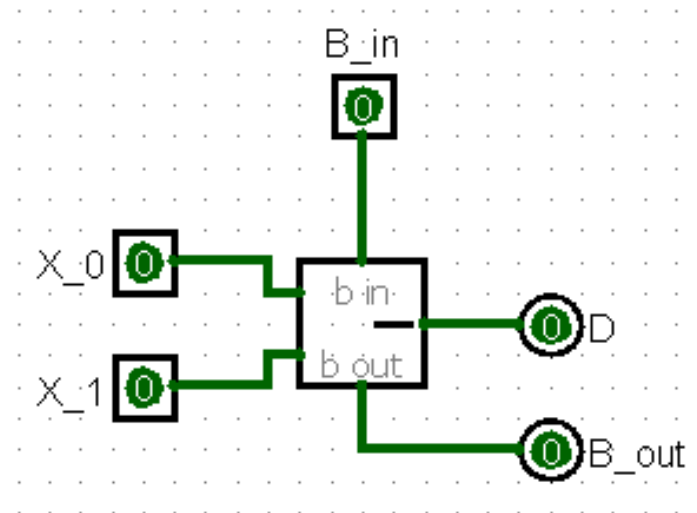
Full Subtractor Logic Circuit:

$B_{IN}$	$X_1$	$X_0$	$B_{OUT}$	$D$
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



# Full Subtractors

- Abstracted Full Subtractor:

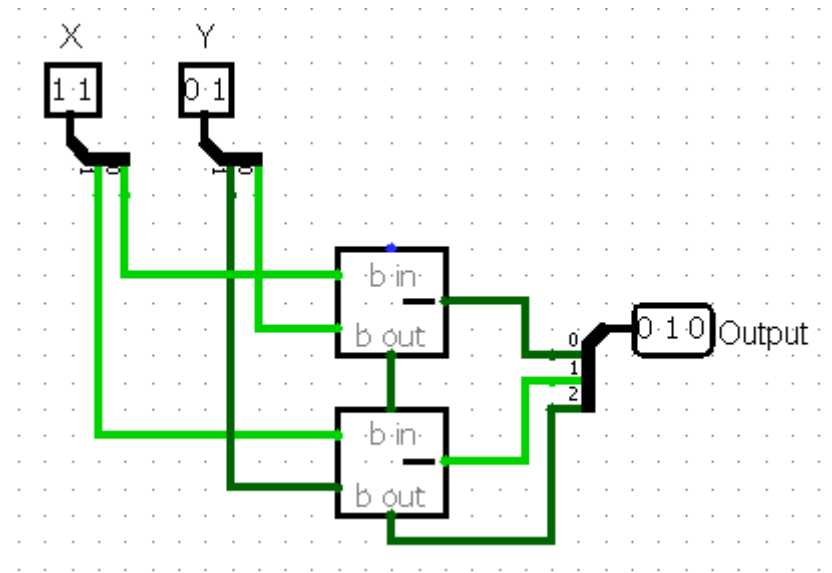




# Full Subtractors

- Like full adders, full subtractors can work together by providing the borrow out of one full subtractor as the borrow in for a second subtractor

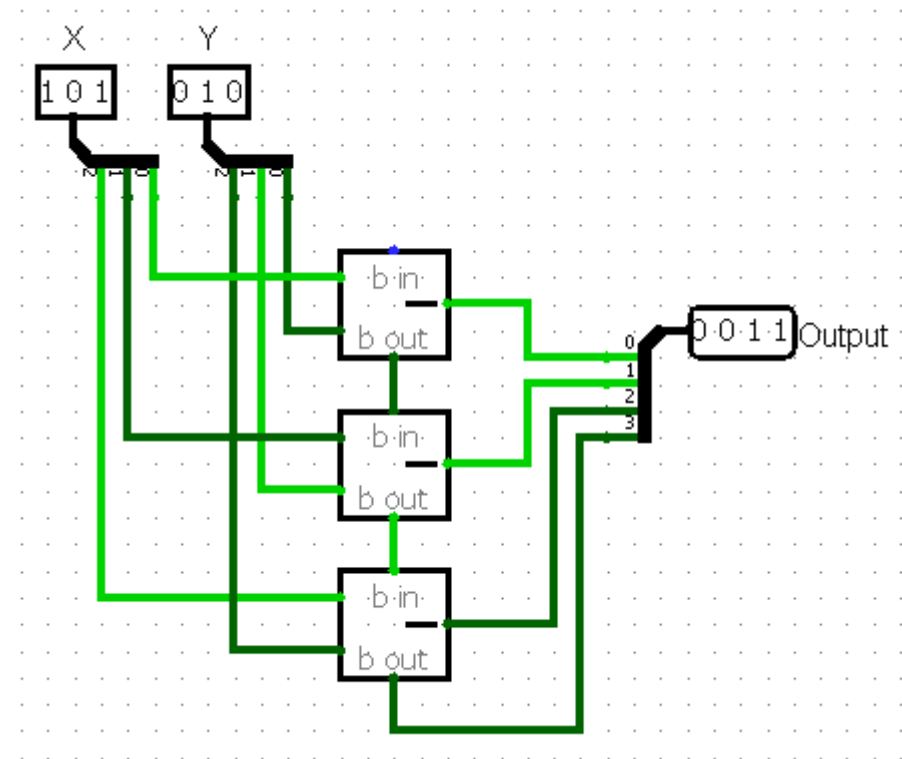
$$\begin{array}{r} 11 \\ -01 \\ \hline 010 \end{array}$$



# Full Subtractors

- A 3-bit Subtractor:

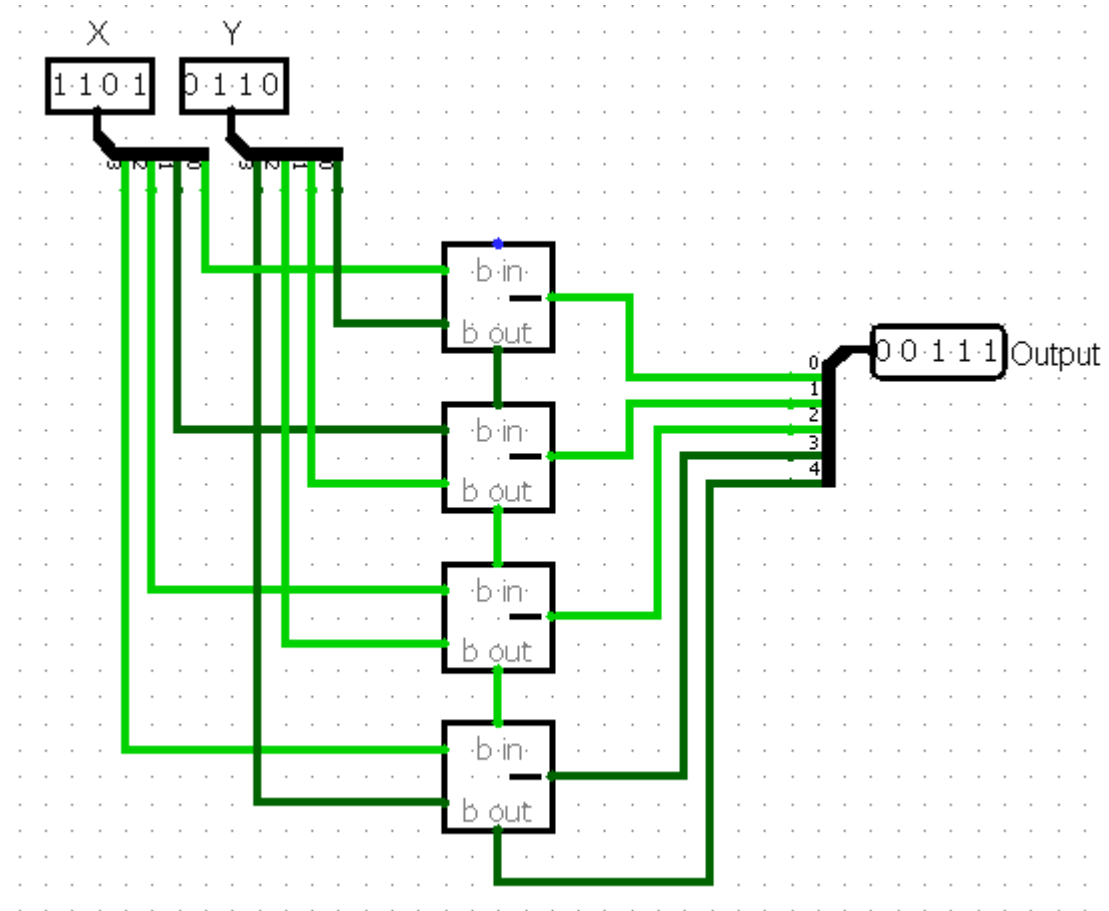
$$\begin{array}{r} 101 \\ - 010 \\ \hline 0011 \end{array}$$



# Full Subtractors

- A 4-bit Subtractor:

$$\begin{array}{r} 1101 \\ - 0110 \\ \hline 0011 \end{array}$$



# Multipliers

- **Multipliers** (not to be confused with multiplexers) are combinational logic circuits capable of performing multiplication
- Note that the multiplication of two 1-bit numbers is a simple *and* operation

$$\begin{array}{cccc}
 0 & 1 & 0 & 1 \xleftarrow{X_0} \\
 \times 0 & \times 0 & \times 1 & \times 1 \xleftarrow{Y_0} \\
 \hline
 0 & 0 & 0 & 1
 \end{array}$$

$X_0$	$Y_0$	$X_0 \times Y_0$	$X_0 \cdot Y_0$
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	0

# Multipliers

- However, addition will be required when multiplying numbers that are two or more bits.
- We will see how to construct multipliers using full and half adders.
- The largest product of multiplying two, 2-bit numbers is 9:
  - $11 \times 11 = 1001$  ( $3 \times 3 = 9$ )
  - Thus, our circuit must have 4 outputs
    - $P_0$  through  $P_3$

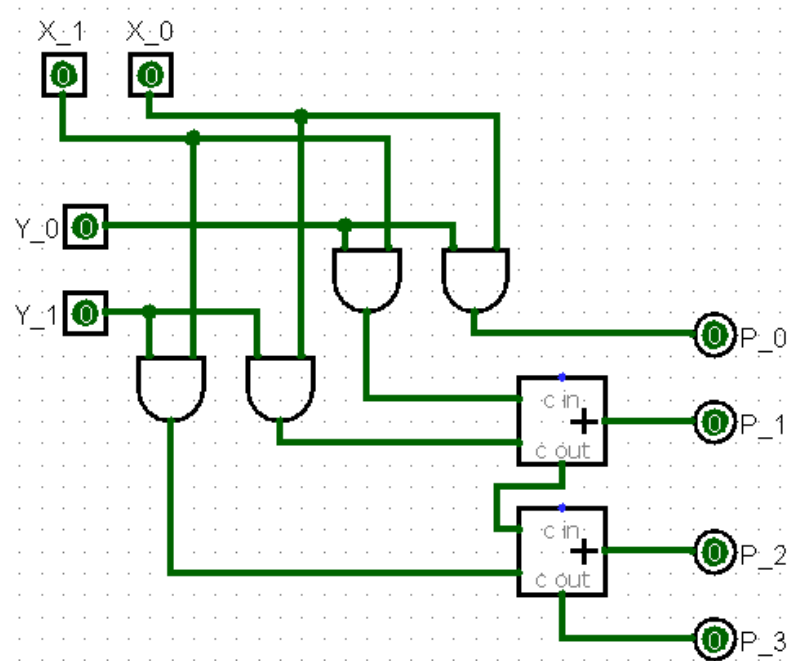
A handwritten binary multiplication diagram. The first number is 11, with its bits labeled  $X_1$  (for the left '1') and  $X_0$  (for the right '1'). The second number is 10, with its bits labeled  $Y_1$  (for the left '1') and  $Y_0$  (for the right '0'). A horizontal line separates the multiplicand from the multiplier. Below the line, the first partial product is 00, corresponding to  $X_0 \times 11$ . The second partial product is 11, corresponding to  $X_1 \times 10$ . A horizontal line separates the two partial products. The final sum is 110, with its bits labeled  $P_2$  (for the left '1'),  $P_1$  (for the middle '1'), and  $P_0$  (for the right '0').

$$\begin{array}{r} \begin{array}{cc} X_1 & X_0 \\ 1 & 1 \end{array} \\ \times \begin{array}{cc} Y_1 & Y_0 \\ 1 & 0 \end{array} \\ \hline 00 \\ + 11 \\ \hline 110 \\ \begin{array}{ccc} \nearrow & \nearrow & \nearrow \\ P_2 & P_1 & P_0 \end{array} \end{array}$$

# Multipliers

## 2-bit Multiplier Logic Circuit:

- (Uses 2 half adders)

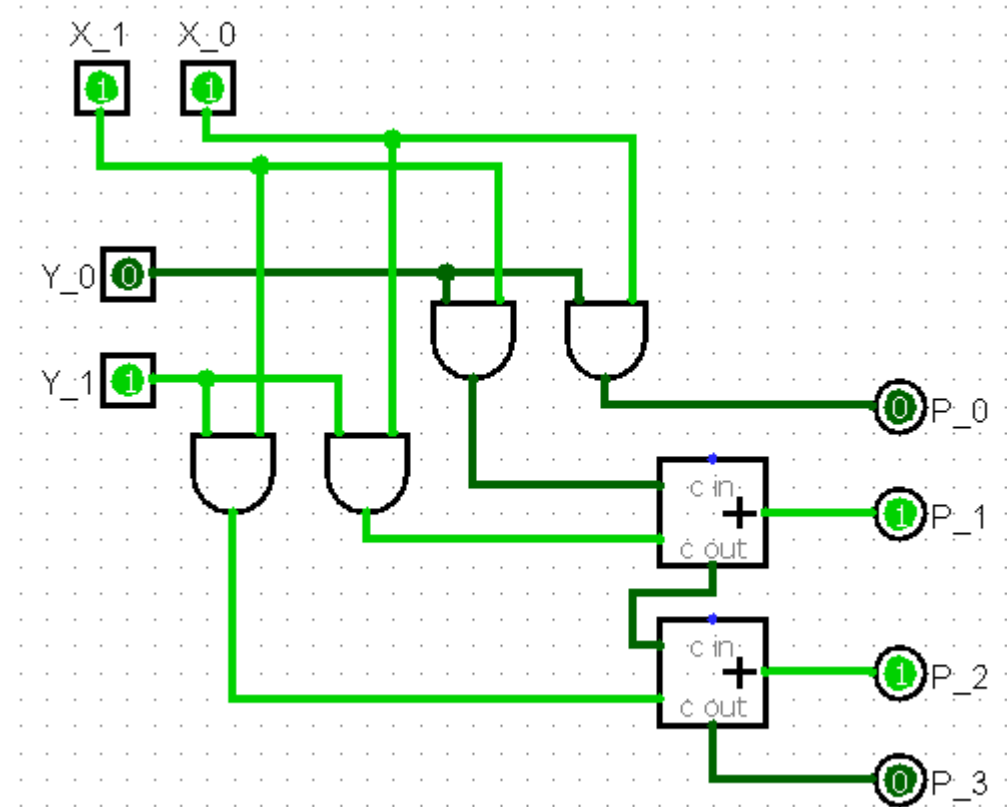


# Multipliers

2-bit Multiplier Logic Circuit:

$$\begin{array}{r} \begin{array}{cc} X_1 & X_0 \\ 1 & 1 \end{array} \\ \times \begin{array}{cc} Y_1 & Y_0 \\ 1 & 0 \end{array} \\ \hline 00 \\ + 11 \\ \hline 110 \end{array}$$

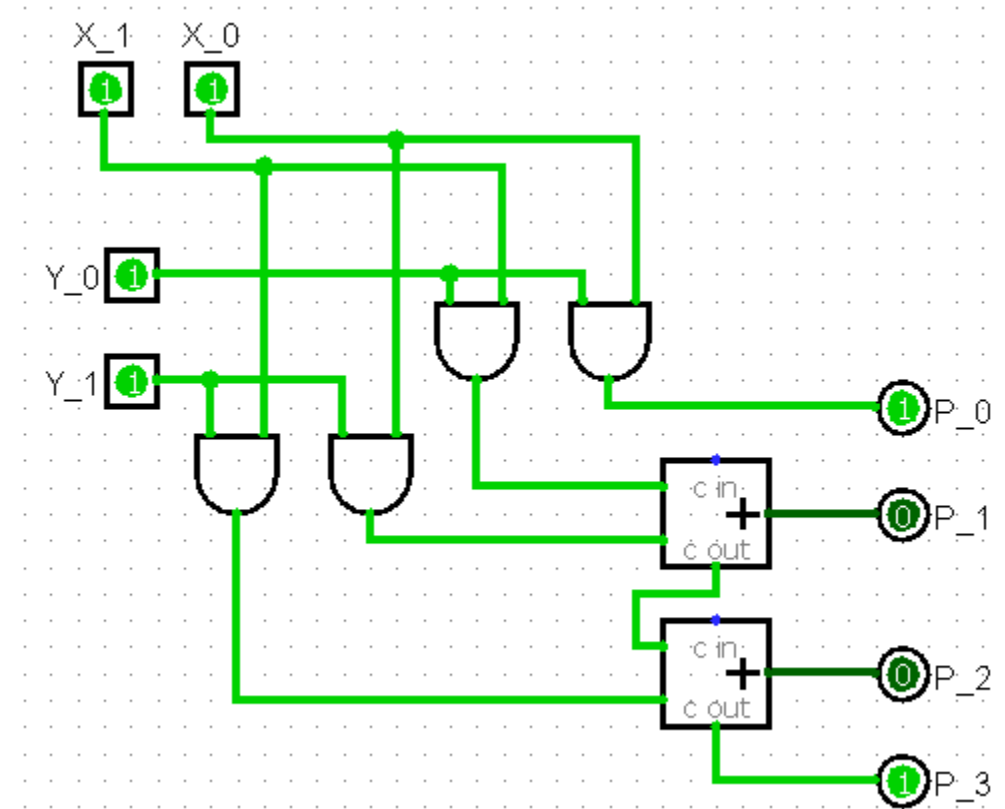
$P_2 \quad P_1 \quad P_0$



# Multipliers

2-bit Multiplier Logic Circuit:

$$\begin{array}{r} X_1 \swarrow \quad \searrow X_0 \\ Y_1 \swarrow \quad \searrow Y_0 \\ \begin{array}{r} X \ 1 \ 1 \\ \hline 1 \ 1 \ 1 \\ + \ 1 \ 1 \\ \hline 1 \ 0 \ 0 \ 1 \end{array} \\ \begin{array}{c} \nearrow P_3 \quad \nearrow P_2 \quad \nearrow P_1 \quad \nearrow P_0 \end{array} \end{array}$$

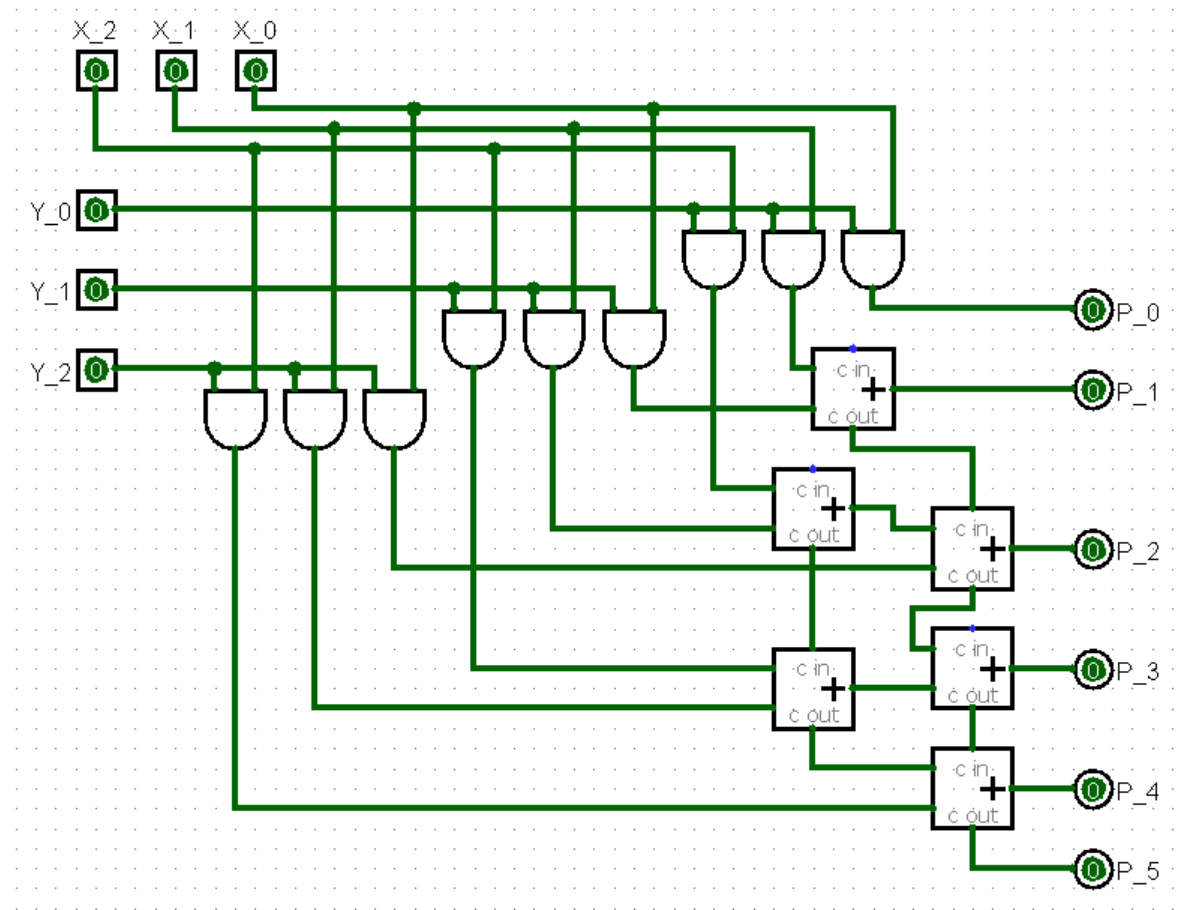




# Multipliers

## 3-bit Multiplier Logic Circuit:

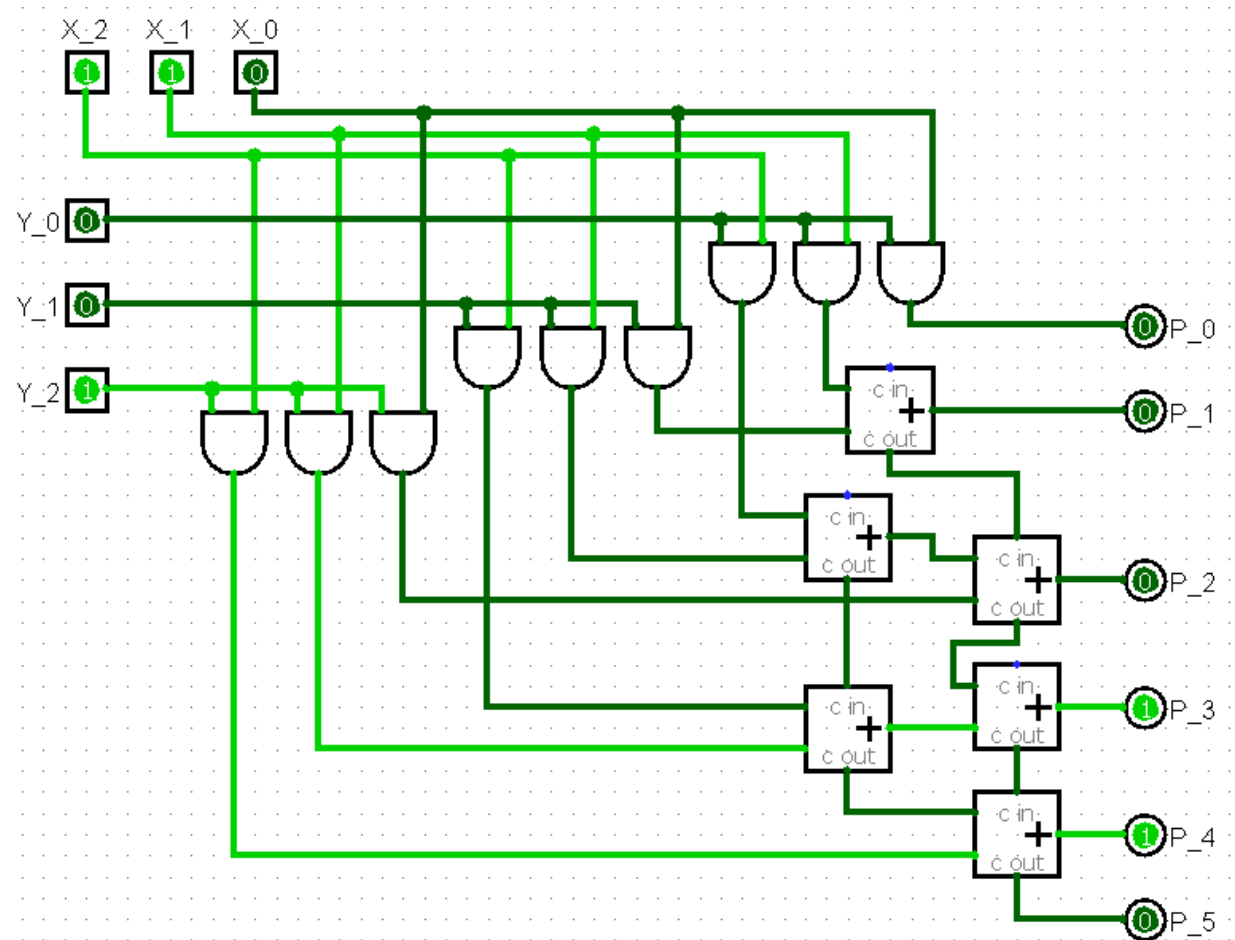
- (Uses 2 half adders)
- (Uses 3 full adders)



# Multipliers

3-bit Multiplier Logic Circuit:

$$\begin{array}{r} 110 \\ \times 100 \\ \hline 000 \\ 000 \\ + 110 \\ \hline 011000 \end{array}$$



# Multipliers

3-bit Multiplier Logic Circuit:

$$\begin{array}{r} 111 \\ \times 111 \\ \hline 111 \\ 111 \\ 111 \\ + 111 \\ \hline 110001 \end{array}$$

