Revised: 5/9/2024

Digital Logic III

Michael C. Hackett

Assistant Professor, Computer Science

COMMUNITY COLLEGE OF PHILADELPHIA

Lecture Topics

- Combinational Circuits
  - Adders
    - Half Adder
    - Full Adder

Adders

- **Adders** are combinational logic circuits capable of performing addition
- A **half adder** has two inputs (the two digits to add) and two outputs (the sum and the carry).

$$
\begin{array}{rrrr}
0 & 1 & 0 & 1 \\
\underline{+\,0} & \underline{+\,0} & \underline{+\,1} & \underline{+\,1} \\
00 & 01 & 01 & 10
\end{array}
$$

X_0

X_1

$C\ (Carry)$ $S\ (Sum)$

Half Adders

- Half adder truth table:

X_1	X_0	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$\begin{array}{r} 0 \\ +\,0 \\ \hline 00 \end{array} \qquad \begin{array}{r} 1 \\ +\,0 \\ \hline 01 \end{array} \qquad \begin{array}{r} 0 \\ +\,1 \\ \hline 01 \end{array} \qquad \begin{array}{r} 1 \\ +\,1 \\ \hline 10 \end{array} \begin{array}{l} \leftarrow X_0 \\ \leftarrow X_1 \\ \\ \end{array}$$

$C\ (Carry)$ $\quad$ $S\ (Sum)$

Half Adders

SOP Expressions:

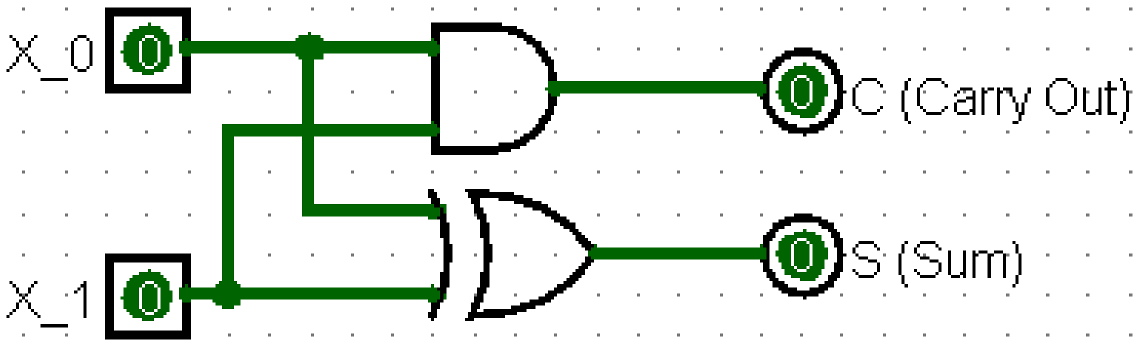
$$C = X_1 X_0$$

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

X_1	X_0	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Half Adders

Half Adder Logic Circuit:



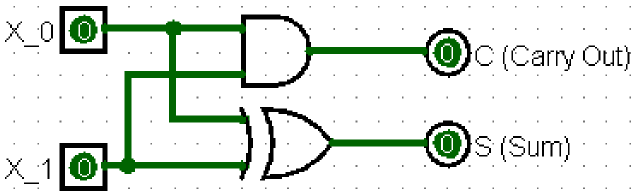
Half Adders

Half Adder Logic Circuit:

0

<u>+ 0</u>

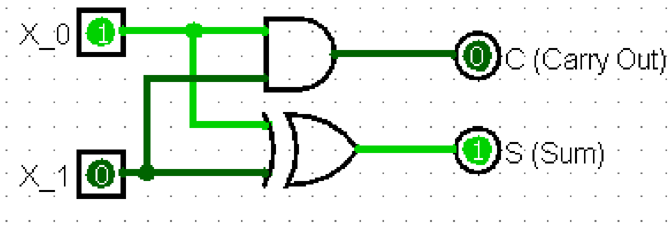
00



1

<u>+ 0</u>

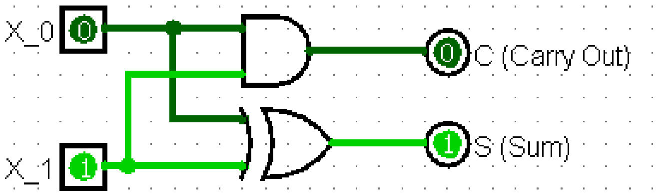
01



0

<u>+ 1</u>

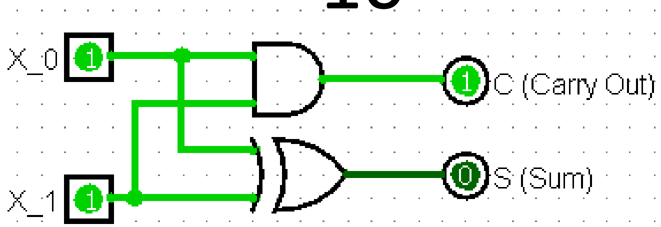
01



1

<u>+ 1</u>

10



Full Adders

- A **full adder** has three inputs (the two digits to add, plus a value *carried in*) and two outputs (the sum and the carry).

$\boldsymbol{C_{IN}(Carry\ In)}$ $\boldsymbol{X_1}$ $\boldsymbol{X_0}$ $\boldsymbol{C_{OUT}(Carry\ Out)}$ $\boldsymbol{S\ (Sum)}$

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

Full Adders

- Full adder truth table:

C_{IN}	X_1	X_0	C_{OUT}	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

Full Adders

SOP Expressions:

$$C_{OUT} = \overline{C_{IN}}X_1X_0 + C_{IN}\overline{X_1}X_0 + C_{IN}X_1\overline{X_0} + C_{IN}X_1X_0$$

$$S = \overline{C_{IN}}\,\overline{X_1}X_0 + \overline{C_{IN}}X_1\overline{X_0} + C_{IN}\overline{X_1}\,\overline{X_0} + C_{IN}X_1X_0$$

C_{IN}	X_1	X_0	C_{OUT}	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

Full Adders

Simplifying:

$$C_{OUT} = \boldsymbol{\overline{C_{IN}}X_1X_0} + C_{IN}\overline{X_1}X_0 + C_{IN}X_1\overline{X_0} + \boldsymbol{C_{IN}X_1X_0}$$

$$C_{OUT} = \boldsymbol{X_1X_0(\overline{C_{IN}} + C_{IN})} + C_{IN}\overline{X_1}X_0 + C_{IN}X_1\overline{X_0}$$

$$C_{OUT} = \boldsymbol{X_1X_0(1)} + C_{IN}\overline{X_1}X_0 + C_{IN}X_1\overline{X_0}$$

$$C_{OUT} = X_1X_0 + \boldsymbol{C_{IN}\overline{X_1}X_0} + \boldsymbol{C_{IN}X_1\overline{X_0}}$$

$$C_{OUT} = X_1X_0 + C_{IN}(\boldsymbol{\overline{X_1}X_0} + \boldsymbol{X_1\overline{X_0}})$$

$$C_{OUT} = X_1X_0 + C_{IN}(X_0 \oplus X_1)$$

Full Adders

Simplifying:

$$S = \overline{C_{IN}}\,\overline{X_1}X_0 + \overline{C_{IN}}X_1\overline{X_0} + C_{IN}\overline{X_1}\,\overline{X_0} + C_{IN}X_1X_0$$

$$S = \overline{C_{IN}}\,(X_0 \oplus X_1) + C_{IN}(X_0 \odot X_1)$$

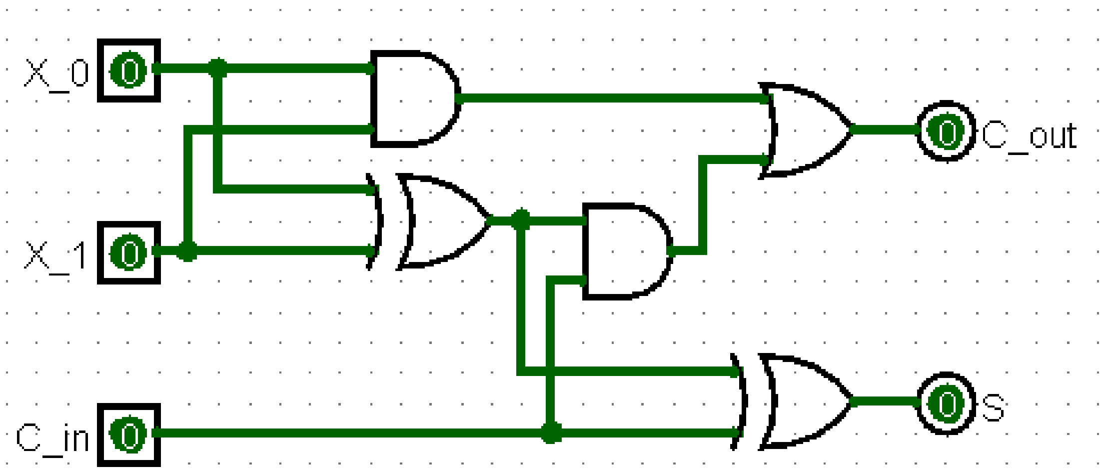
$$S = \overline{C_{IN}}\,(X_0 \oplus X_1) + C_{IN}(\overline{X_0 \oplus X_1})$$

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

$$S = C_{IN} \oplus (X_0 \oplus X_1) = C_{IN} \oplus X_0 \oplus X_1$$

Full Adders

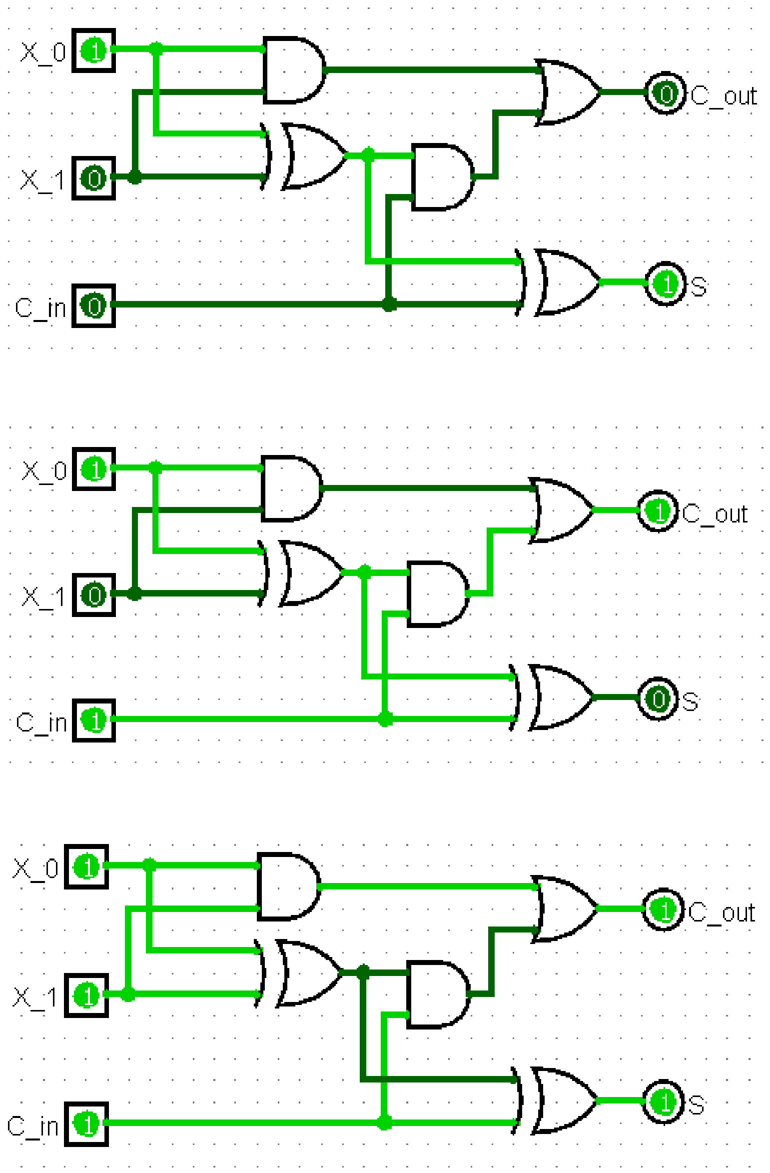
Full Adder Logic Circuit:



Full Adders

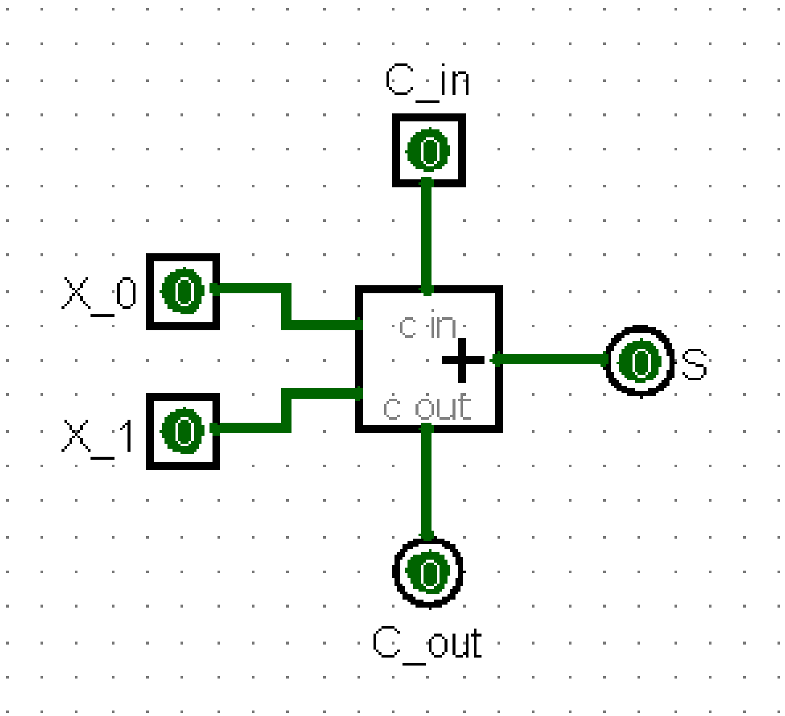
Full Adder Logic Circuit:

C_{IN}	X_1	X_0	C_{OUT}	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**



Full Adders

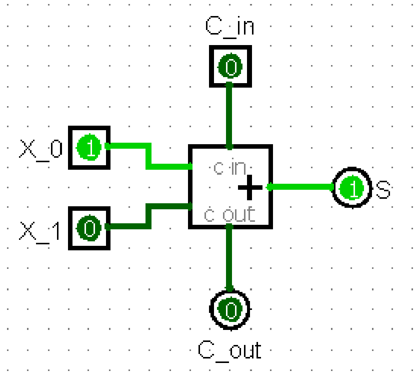
- Abstracted Full Adder:

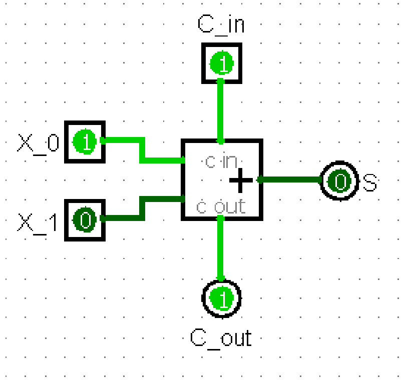


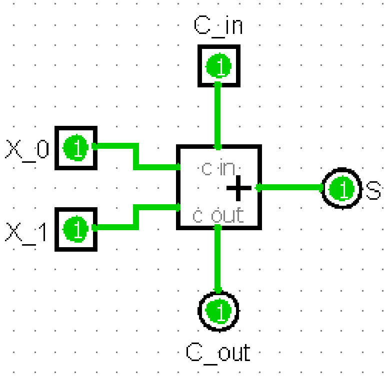
Full Adders

- Abstracted Full Adder:

C_{IN}	X_1	X_0	C_{OUT}	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**







Full Adders

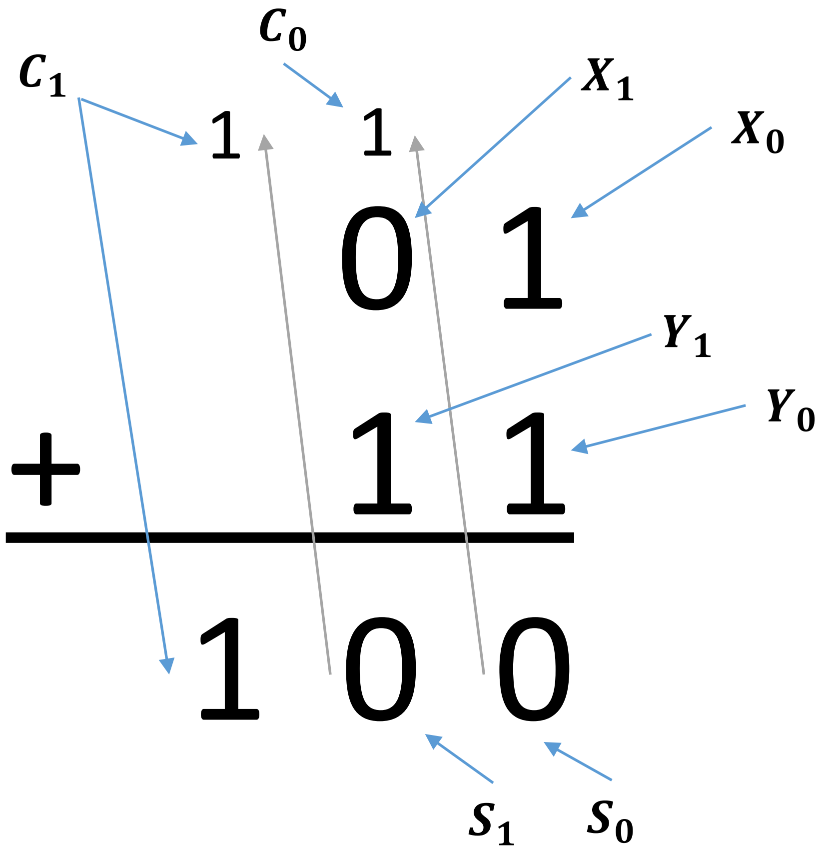
- So far, we've seen only how adders can add single digits together
  - 1+1+1 or 0+1 or 1+0+1 is no problem
  - What if we wanted to add 10+11 or 11101+10101?

- Full adders can work together by providing the carry out of one full adder as the carry in for a second adder
  - The technique shown next is a *ripple-carry adder*

$$\begin{array}{r} 0\ 0\ 1 \\ +\ \underline{0\ 1\ 1} \\ 1\ 0\ 0 \end{array}$$

Full Adders

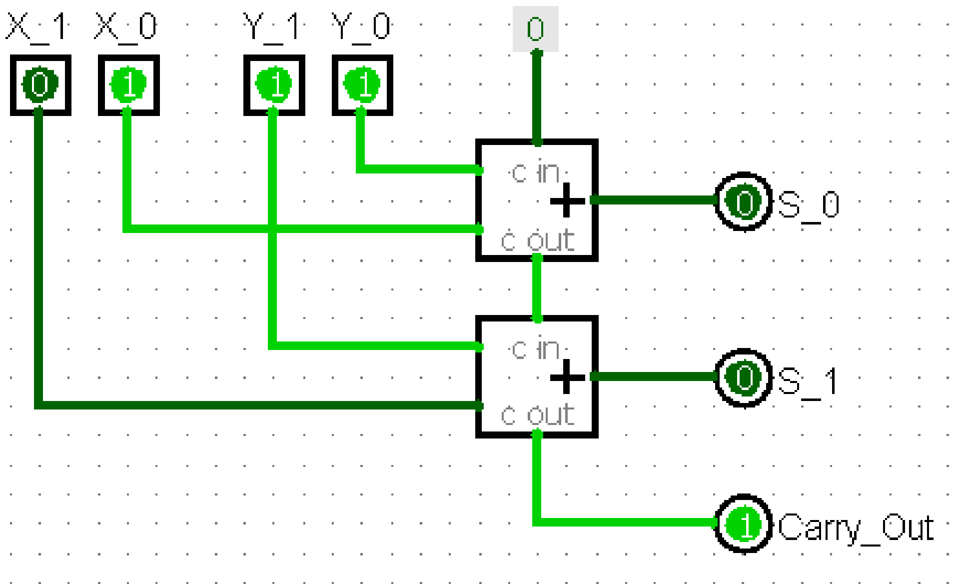
- For example:



Ripple-Carry Adders

- A 2-bit Adder:

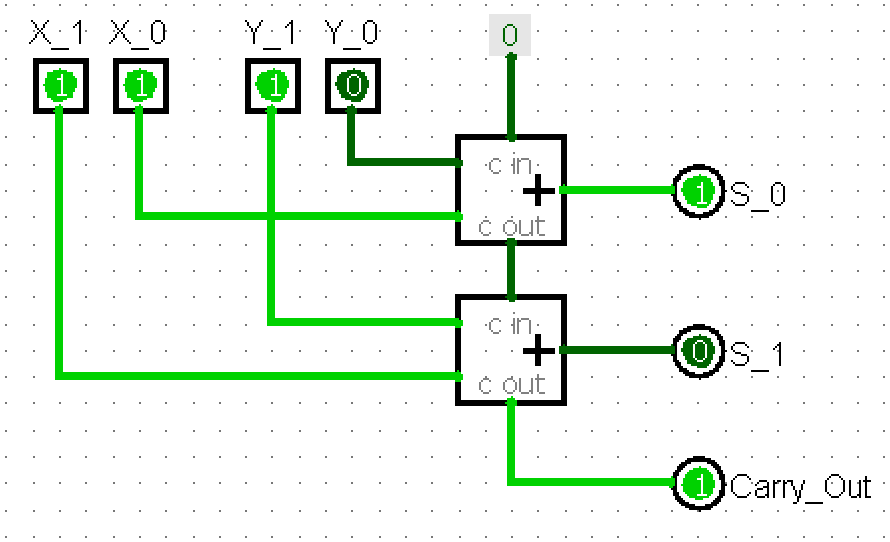
$$\begin{array}{r} 0\ 1 \\ +\quad 1\ 1 \\ \hline 1\ 0\ 0 \end{array}$$



Ripple-Carry Adders

- Another example:

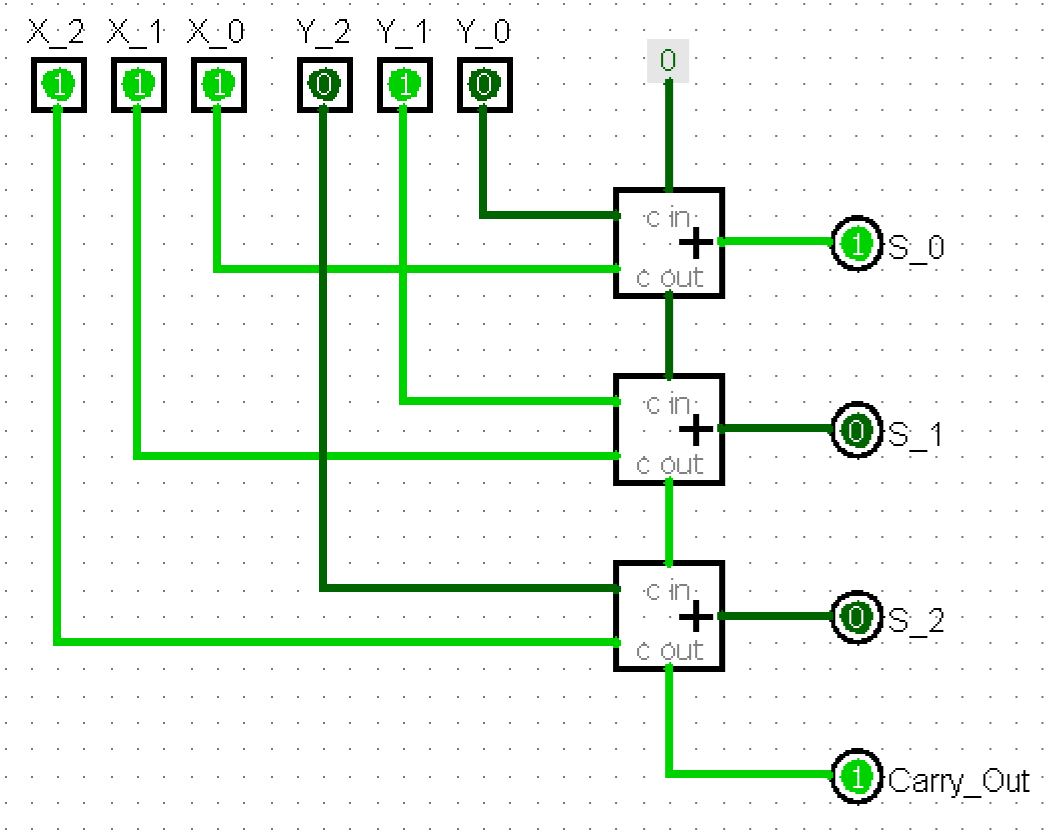
$$\begin{array}{r} 1\ 1 \\ +\quad 1\ 0 \\ \hline 1\ 0\ 1 \end{array}$$



Ripple-Carry Adders

- A 3-bit Adder:

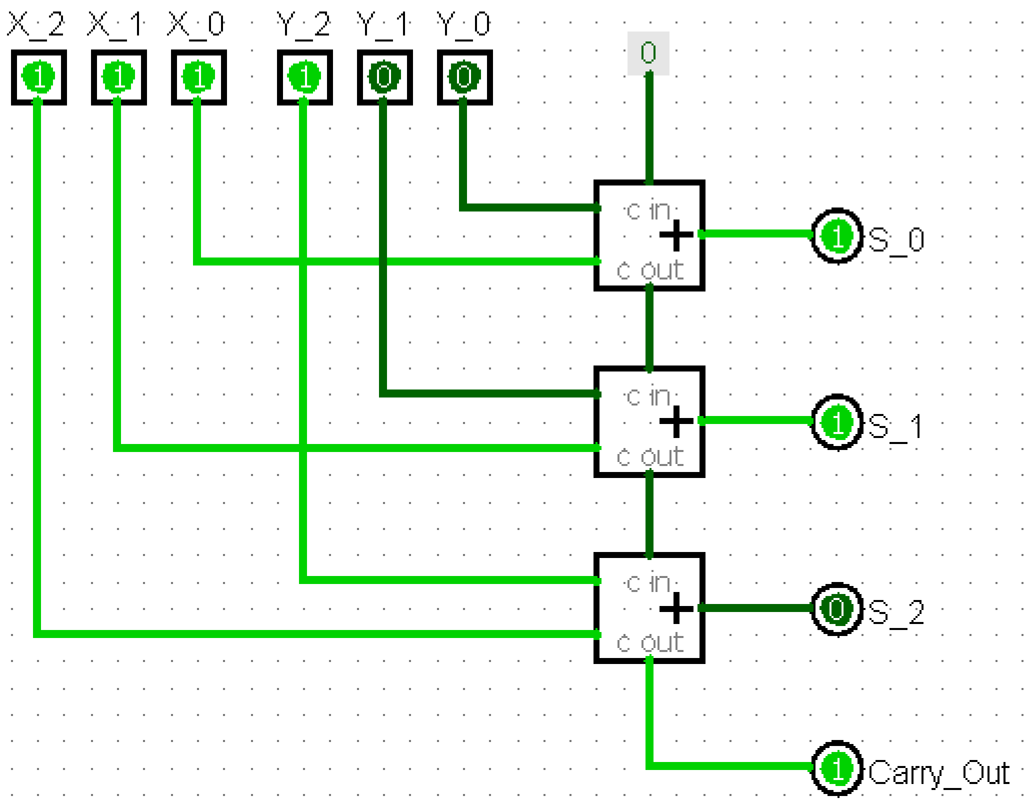
$$
\begin{array}{r}
1\ 1\ 1 \\
+ \quad 0\ 1\ 0 \\
\hline
1\ 0\ 0\ 1
\end{array}
$$



Ripple-Carry Adders

- Another example:

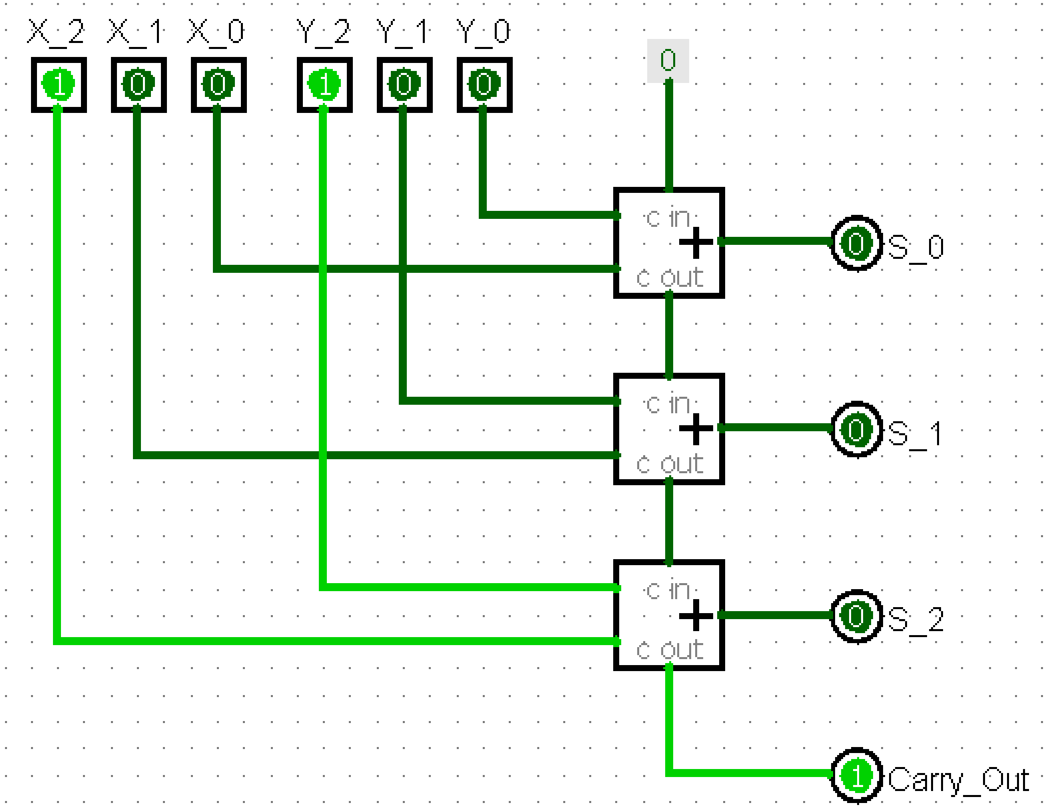
$$\begin{array}{r} 1\ 1\ 1 \\ +\quad 1\ 0\ 0 \\ \hline 1\ 0\ 1\ 1 \end{array}$$



Ripple-Carry Adders

- Another example:

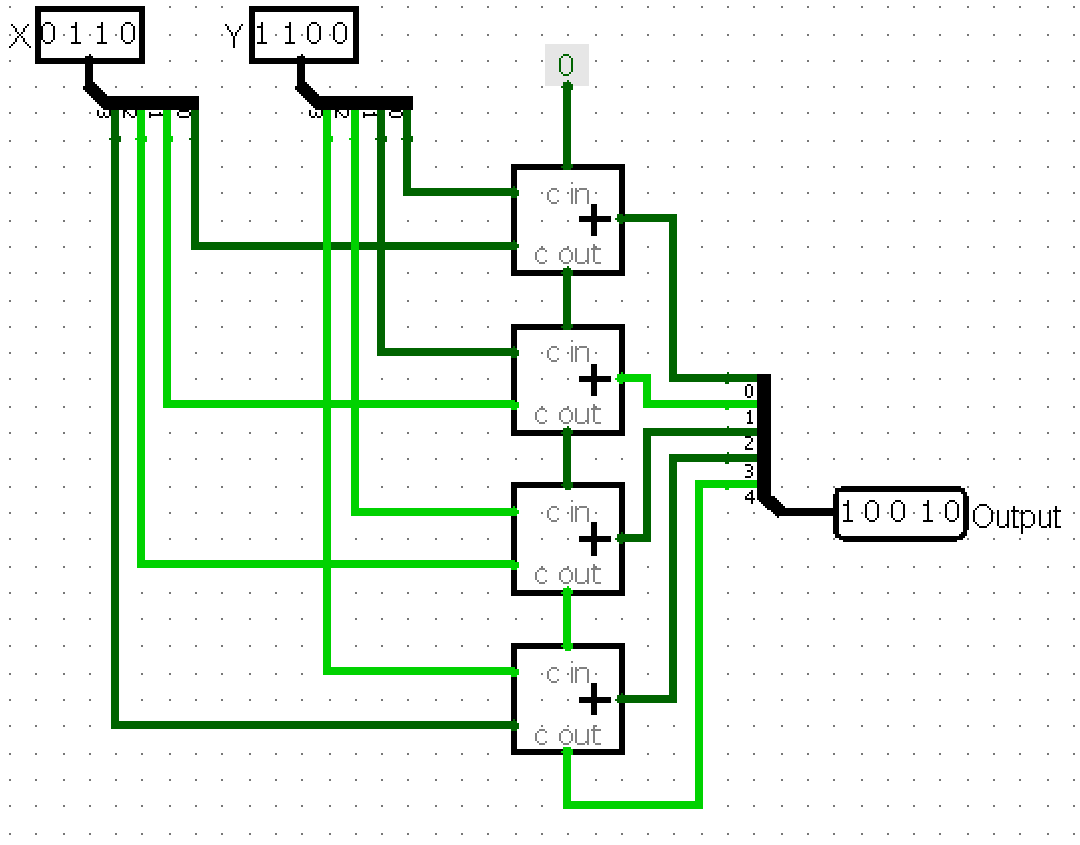
$$\begin{array}{r} 1\ 0\ 0 \\ +\quad 1\ 0\ 0 \\ \hline 1\ 0\ 0\ 0 \end{array}$$



Ripple-Carry Addersv

- A 4-bit Adder (with buses):

$$
\begin{array}{r}
0\ 1\ 1\ 0 \\
+\quad 1\ 1\ 0\ 0 \\
\hline
1\ 0\ 0\ 1\ 0
\end{array}
$$



Ripple-Carry Adders

- Another example

$$\begin{array}{r} 1\ 0\ 1\ 0 \\ +\quad 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 1\ 1\ 1 \end{array}$$