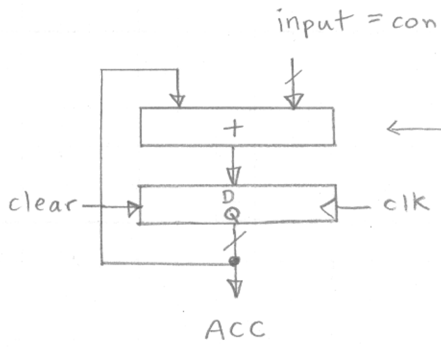
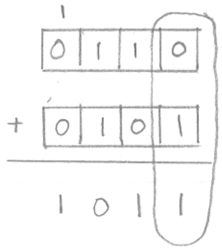
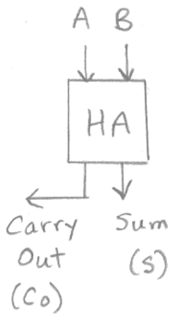
Consider making a counter that can count in arbitrary increments (basically an accumulator with the same input every cycle):



← How do we build this adder circuit?



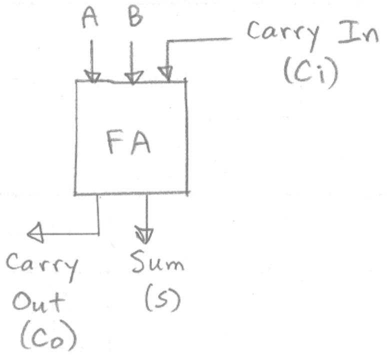
Looking at one slice (bit slice), we need a circuit which computes a sum and a carry for two input bits:



Half Adder

A	B	S	Co
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

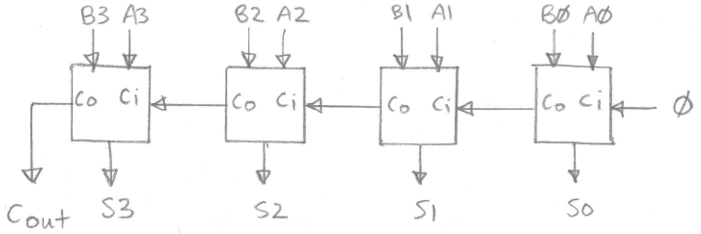
We also need to handle carry in bits for higher order bits in the word:



Full Adder

A	B	Ci	S	Co	Answer
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	0	1	2
1	0	0	1	0	1
1	0	1	0	1	2
1	1	0	0	1	2
1	1	1	1	1	3

Ex: 4-bit adder



Cout needed for large inputs (4-bit unsigned numbers: 0-15)

Ex:
$$\begin{array}{r} 1\,1\,0\,1 \\ +\ 1\,0\,0\,0 \\ \hline \underset{\smile}{1}\,0\,1\,0\,1 \end{array} = \begin{array}{r} 13_{10} \\ +\ 8_{10} \\ \hline 21_{10} \end{array}$$

Ex:
$$\begin{array}{r} 15_{10} \\ +\ 0_{10} \\ \hline 15_{10} \end{array} = \begin{array}{r} 1\,1\,1\,1 \\ +\ 0\,0\,0\,0 \\ \hline 1\,1\,1\,1 \end{array} \qquad \begin{array}{r} 15_{10} \\ +\ 1_{10} \\ \hline 16_{10} \end{array} = \begin{array}{r} 1\,1\,1 \\ 1\,1\,1\,1 \\ +\ 0\,0\,0\,1 \\ \hline 1\,0\,0\,0\,0 \end{array}$$

← carry signal "ripples" one bit at a time through the adder.

This is the slowest case (longest delay before output settles to final value). This adder structure is called a ripple-carry adder.

Delay is proportional to word size. If word bit width is N, speed is O(N) or "Order N"

Faster adder structures speed up operation.