Learning objectives

- To add two positive binary numbers
- Explain the concept of overflow

Binary Addition Rules

$$\begin{array}{rr} & 0 \\ + & 0 \\ \hline & 0 \end{array} \qquad \begin{array}{rr} & 0 \\ + & 1 \\ \hline & 1 \end{array} \qquad \begin{array}{rr} & 1 \\ + & 1 \\ \hline 1 & 0 \end{array} \qquad \begin{array}{rr} & 1 \\ & 1 \\ + & 1 \\ \hline 1 & 1 \end{array}$$

Adding 2 Binary numbers

		1		1	1	1
+	1	1	1	0	1	0
1		1		0	0	1

Worked Example

128	64	32	16	8	4	2	1
0	0	1	1	0	0	1	1
1	0	0	0	1	1	1	1
	1	1	1	1	1	1	
1	1	0	0	0	0	1	0

A quick check...

128	64	32	16	8	4	2	1
1	1	0	0	0	0	1	0

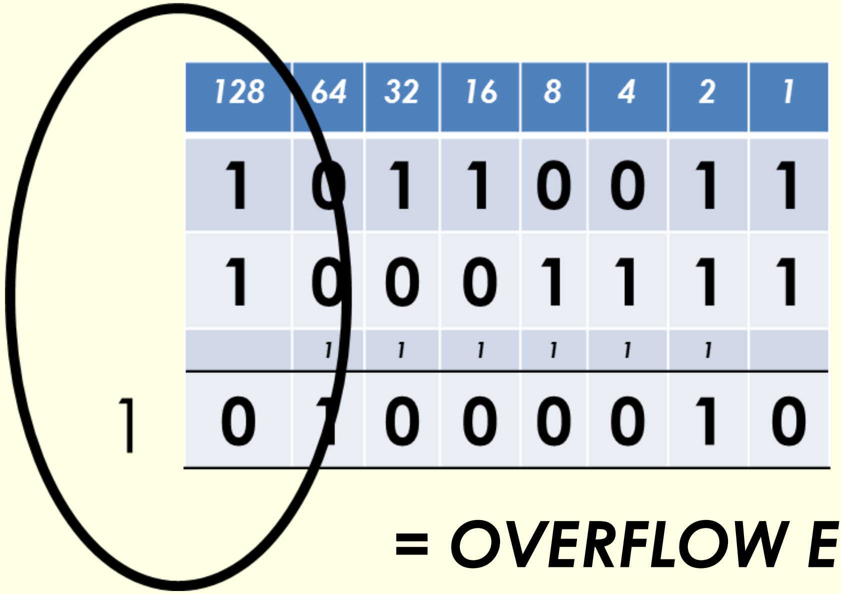
= 194

128	64	32	16	8	4	2	1
0	0	1	1	0	0	1	1
1	0	0	0	1	1	1	1
	1	1	1	1	1	1	
1	1	0	0	0	0	1	0

51 + 143 = 194

The problem with 9-bit answers

Because the largest number we can hold in a byte is 255, if we add two bytes together there is a chance that the answer will be greater than 255.



	128	64	32	16	8	4	2	1
	1	0	1	1	0	0	1	1
	1	0	0	0	1	1	1	1
		1	1	1	1	1	1	
1	0	1	0	0	0	0	1	0

= OVERFLOW ERROR!

This answer will not be able to be held in a byte and so this causes an over flow error.

In modern computers, CPU’s can hold much larger numbers so this is dealt with. *Ever heard of a 32 or 64 bit processor? – these can deal with larger binary numbers!*

However you do **need to know that overflow errors occur** when doing binary addition **when the answer is 9 bits** in length.