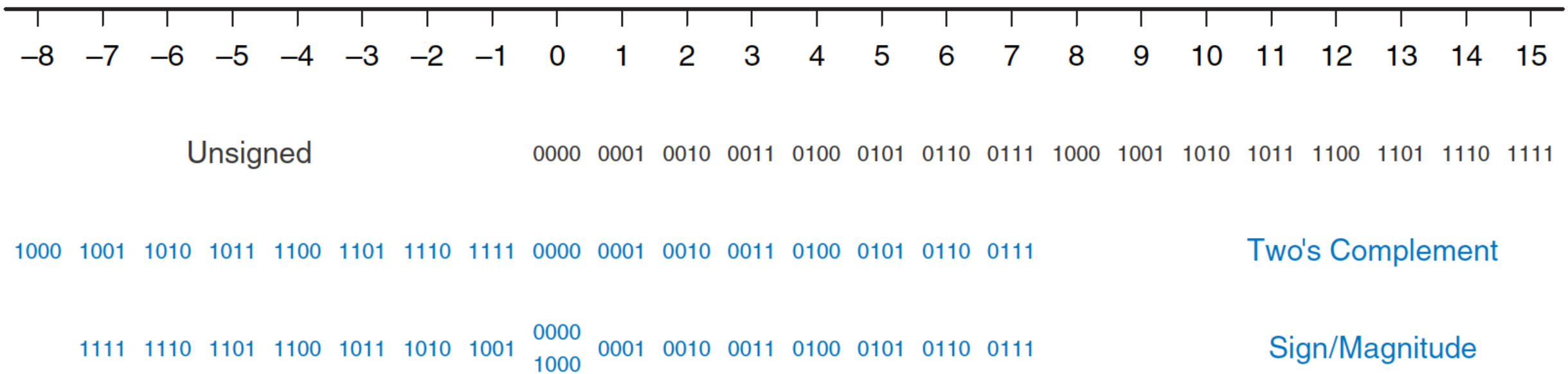
Overflow

Based on slides by Jared Moore

System	Range
Unsigned	$[0, 2^N - 1]$
Sign/Magnitude	$[-2^{N-1} + 1, 2^{N-1} - 1]$
Two's Complement	$[-2^{N-1}, 2^{N-1} - 1]$



Overflow in unsigned addition

Digital systems have a fixed number of bits.

Carry bit exceeds the number of bits.

Leads to incorrect results!

$$\begin{array}{r} 1100_2 \\ +\ 1001_2 \\ \hline \end{array}$$

Overflow in unsigned addition

Overflow: When the result of an operation cannot be represented in the given system.

4-bit system can represent [0, 15]. Any operands that lead to a sum greater than 15 will cause overflow in this system.

How to detect overflow in unsigned addition in a circuit?

Check for a carry on the most significant column.

When can two's complement addition overflow?

How can we detect overflow in two's complement?

Overflow means the same thing for unsigned and two's complement – that the result is outside the representable range

However, **we use different methods to detect overflow in unsigned and two's complement arithmetic**

Two's Complement Overflow

Overflow occurs when adding two N-bit positive numbers or negative numbers if result is greater than $2^{N-1} – 1$ or less than -2^{N-1}

Adding a positive and negative will **never** cause an overflow.

Remember: Overflow detection is not the same as unsigned numbers.

What if I really want to use unsigned overflow detection for two's complement?

Don't.

	Unsigned (overflows)	**Signed** (no overflow)
0100_2	4	4
$+\ 1110_2$	$+\ 14$	$+\ -2$
$1\ 0010_2$	2	2

How can we detect overflow in two's complement?

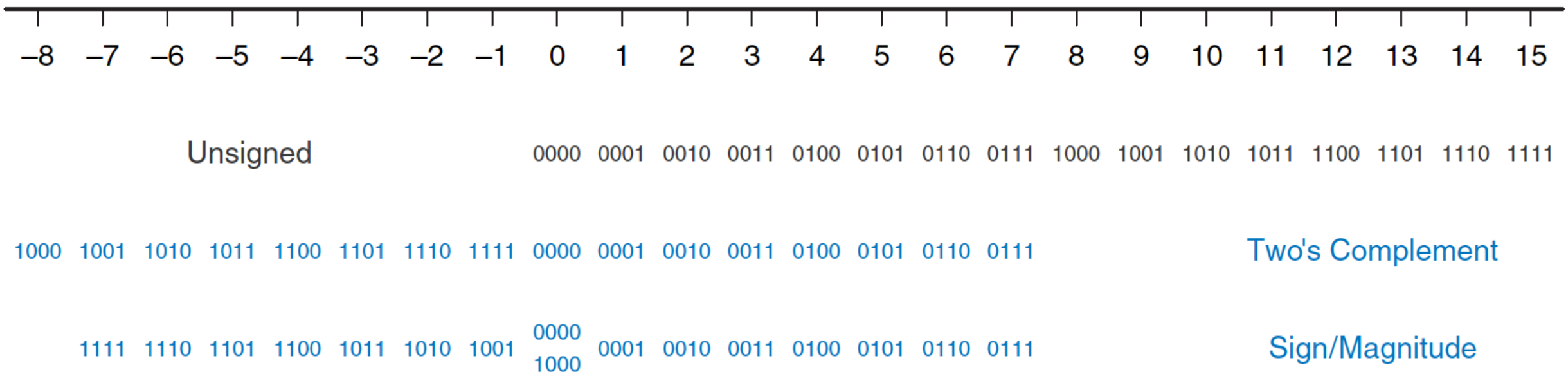
How can we detect overflow in two's complement?

Two numbers being added have the same sign

- Not possible for positive and negative to overflow!

Result has the opposite sign.

System	Range
Unsigned	$[0, 2^N - 1]$
Sign/Magnitude	$[-2^{N-1} + 1, 2^{N-1} - 1]$
Two's Complement	$[-2^{N-1}, 2^{N-1} - 1]$



Important to remember the distinction between what overflow *is* vs how we *detect it*

Remember – detecting overflow in *signed* arithmetic follows different rules from detecting overflow in *unsigned* arithmetic