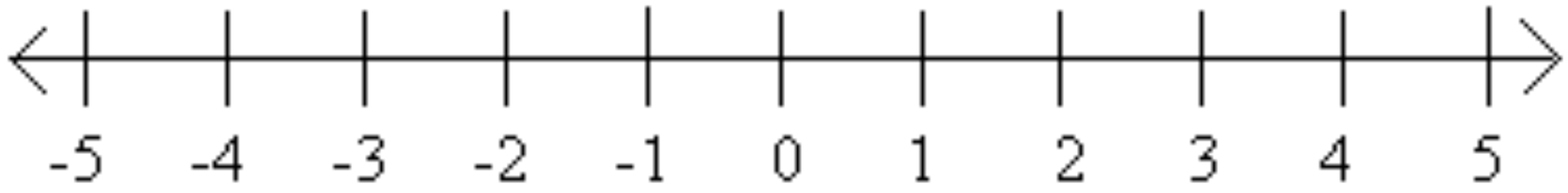
Negative Numbers



Negative Numbers

What you will be able to do after this lecture:

1. explain the need for 2s-complement
2. explain how 10's complement works in decimal
3. perform addition and subtraction with 2s complement

Integers

A whole number (not fractional) that can be positive, negative or zero.

Examples:

-5, 1, 7, 928, -2943

Not Examples:

7.32, -1.42, .09

Computers use a fixed number of bits to represent an integer (8-bit, 16-bit, 32-bit or 64-bit).

Two representations of integers:

1. **Unsigned Integers:** can represent zero and positive integers.
2. **Signed Integers:** can represent zero, positive and negative integers.

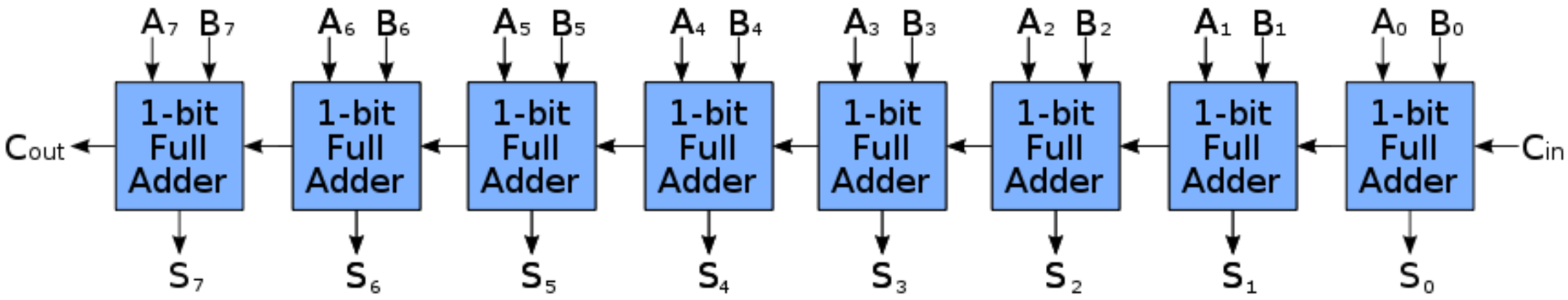
Decimal Arithmetic

Add:

34	36	34	-34	-34
23	17	-23	23	-23
———	———	———	———	———

Different strategies (algorithms) depending on signs

Binary Addition



Different ‘algorithms’ for different variants

Not good.

We want a single/unified way to do addition

Regardless or sign

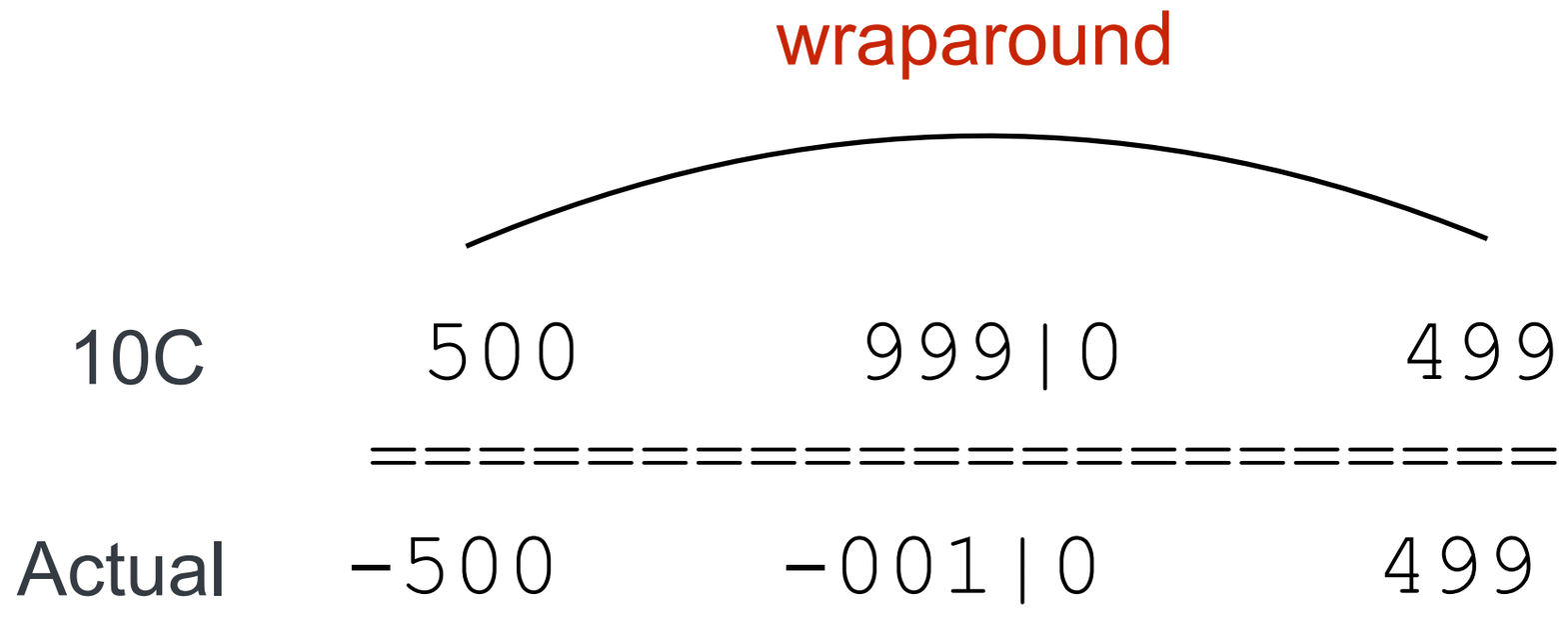
Complements

A technique to subtract one number from another using only addition of positive numbers.

10's Complement

Let's start with decimal

3 digit integers
Not 0,…,999



For negative number *x*, 10C representation is 1000 - *x, e.g.*

- *-1 ~ 999*
- *-9 ~ 991*

1000 is termed the modulus.
1000 - *x* same as (999 - *x*) + 1

Because of wraparound behaviour subtracting *x* same as adding 10C representation of *x.*

10C Examples

To convert a negative decimal number to 10C
Flip: i.e. convert each digit x to its 9C (9-x)
Add 1

3 decimal digits

```
  347
 -124
  223
```

```
-124
 875  flip     347
 876  add 1   +876
          ----
          1223
```

4 decimal digits

```
 0347
-0124
 0223
```

```
-0124
 9875  flip     0347
 9876  add 1   +9876
          -----
          10223
```

overflows out

10C Exercise

Try

```
  563
 -267
-----
  296
```

```
-267
 732 flip
 733 add 1
```

```
  563
  733
------
 1296
```

Complements were commonly used in mechanical calculators.

Signed Binary Integers

Signed Integers: can represent zero, positive and negative integers.

Three main approaches in binary:

- Sign-Magnitude representation
- 1's Complement representation
- 2's Complement representation

Significant and Insignificant

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

MSB — Most significant bit

LSB — Least significant bit

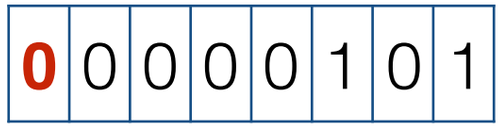
For signed integers, MSB = **sign bit**

Magnitude

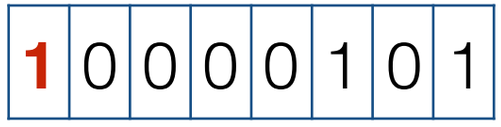
Sign and magnitude (SAM)

Use left most bit to represent sign

- 0 for positive



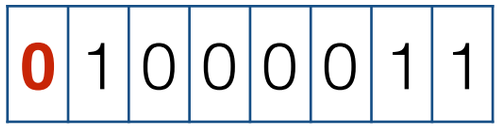
- 1 for negative



SAM range lower than unsigned binary 2^7 vs 2^8

Sign and magnitude (SAM)

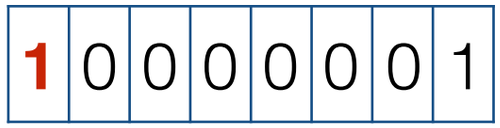
Examples



Sign bit `0` ⇒ positive

Binary value = `67D`

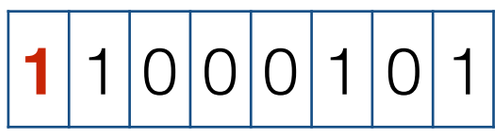
So, integer value is `+67D`



Sign bit `1` ⇒ negative

Binary value = `1D`

So, integer value is `-1D`



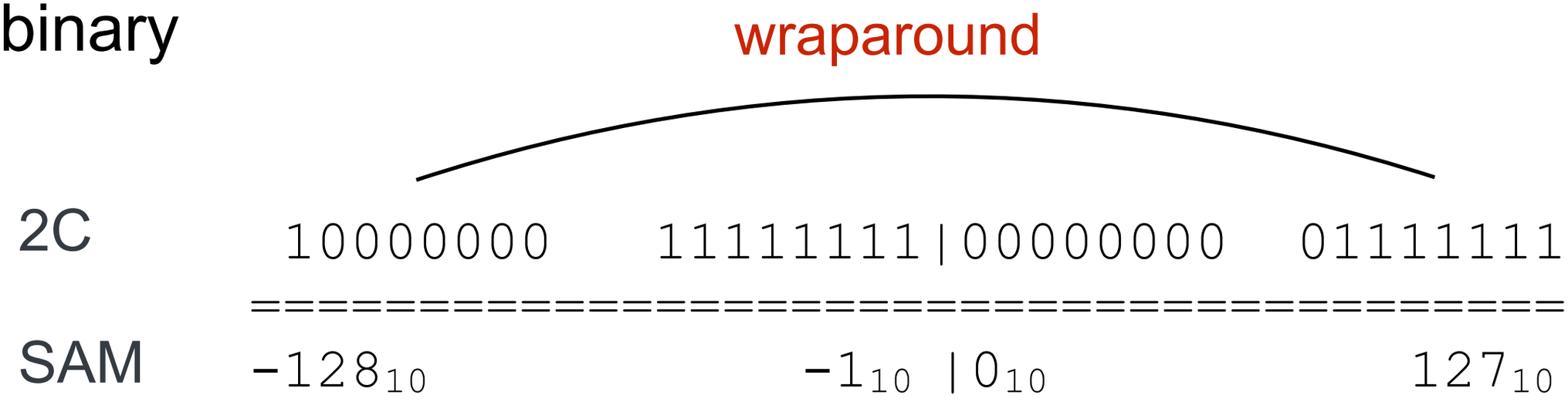
Sign bit `1` ⇒ negative

Binary value = `69D`

So, integer value is `-69D`

2's Complement

8 bit binary



For neg. number *x* 8 bit 2C representation is 1 0000 0000 - *x, e.g.*

- -1 ~ 1111 1111
- 1 0000 0000 - *x same as (*1111 1111 - *x*) + 1
- 1111 1111 - *x* : flip bits

Because of wraparound behaviour subtracting *x* same as adding 2C representation of *x.*

Twos Complement (2C)

Conversion:

- Positive numbers are the same
- Negative numbers - flip bits and add 1.

- - 9 (SAM)

1	0	0	0	1	0	0	1

- -9 (2C)

1	1	1	1	0	1	1	1

-42(SAM)

1	0	1	0	1	0	1	0

-42 (2C)

1	1	0	1	0	1	1	0

Conversion from decimal to 2's C

Convert -48 decimal to -48 in binary 2's complement

48D

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

Flip bits

1	1	0	0	1	1	1	1

Add one to get -48 in 2C

1	1	0	1	0	0	0	0

Conversion from decimal 2's Complement

Try to convert -38 decimal to 2's complement

38D

SIGN	64s	32s	16s	8s	4s	2s	1s
0	0	1	0	0	1	1	0

Flip bits

1	1	0	1	1	0	0	1

Add one to get -38 in 2C

1	1	0	1	1	0	1	0

Two's Complement Addition

Same process as ordinary addition

+52 (2C)	**0**	0	1	1	0	1	0	0
-42 (2C)	**1**	1	0	1	0	1	1	0

+37 (2C)	**0**	0	1	0	0	1	0	1
-22 (2C)	**1**	1	1	0	1	0	1	0

+10 (2C)	**0**	0	0	0	1	0	1	0

+15 (2C)	**0**	0	0	0	1	1	1	1

Negative Numbers

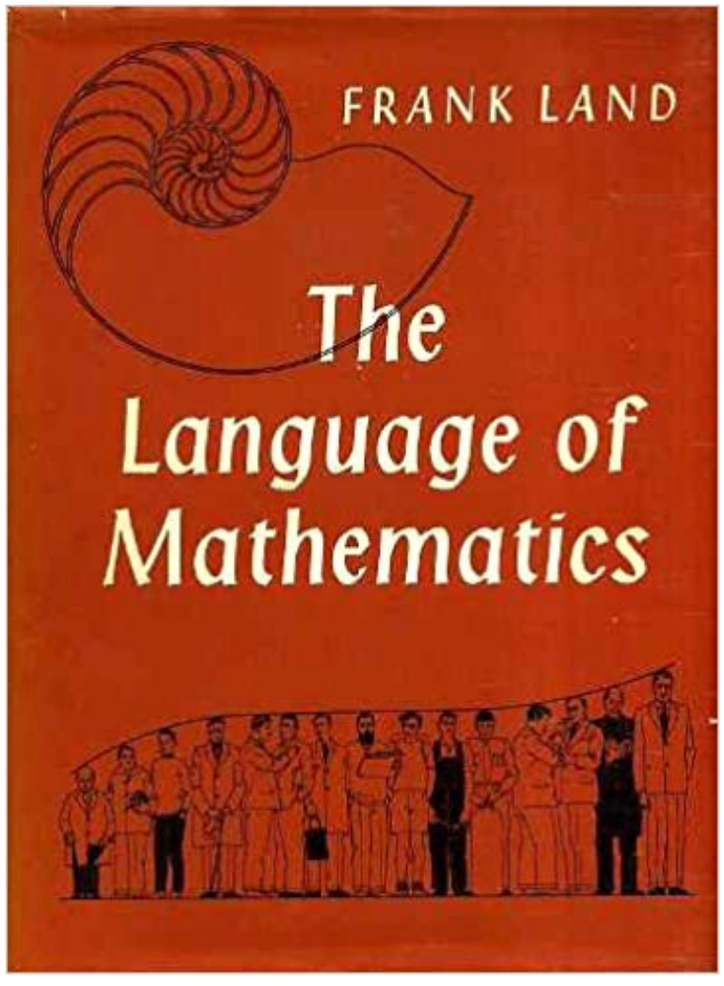
What you will be able to do after this lecture:

1. explain the need for 2s-complement
2. explain how 10's complement works in decimal
3. perform addition and subtraction with 2s complement

References

see Chapter 5 in The Architecture of Computer Hardware….Englander 4th Ed.

for more in depth reading on this see The Language of Mathematics



Computerphile: binary - plusses and minuses
https://www.youtube.com/watch?v=lKTsv6iVxV4