Announcements

assign2 is being graded

- Feedback early next week

Textbook readings

- Very helpful for this and upcoming material

Goals for Today

Finish discussion of ints

- Signed/unsigned, larger integer types

Generalize binary polynomial to real numbers

- Fixed point representation

See the mechanics of floating point

Understand the limitations of floating point

- Epsilon (non-representable numbers)
- Arithmetic error

Code: Signed and Unsigned

Assignment copies bit pattern

- Value may change sign if not representable

Right shift different behavior

- Unsigned: fill with zero
- Signed: fill with copy of sign bit (preserve sign)

Comparison between signed and unsigned

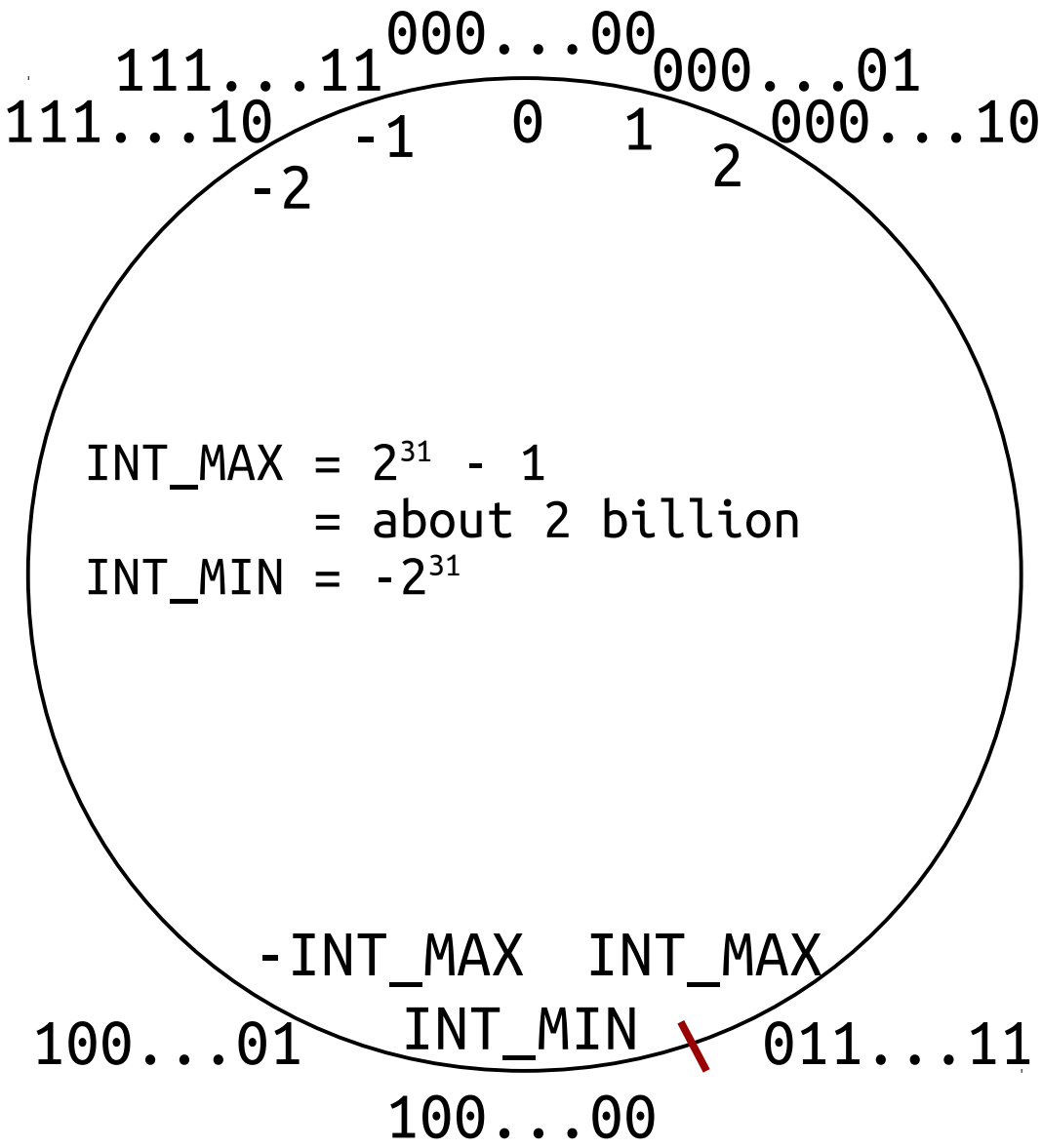
- Unsigned wins

Larger Integer Types

short (16 bits), int (32 bits), long (64 bits)

Same idea: MSB = sign, uses 2's complement

int Number Circle



Code: Larger Integer Types

short (16 bits), int (32 bits), long (64 bits)

- Same idea: MSB = sign, uses 2's complement

Assign to smaller type

- Keep least significant bits, could flip sign

Assign to larger type

- Fill with zeros (unsigned) or copy of sign bit (signed)
- Depends on source type

So Far

Finish discussion of ints

- Signed/unsigned, larger integer types

Generalize binary polynomial to real numbers

- Fixed point representation

See the mechanics of floating point

Understand the limitations of floating point

- Epsilon (non-representable numbers)
- Arithmetic error

Place Value: Real Numbers

In decimal: $5\ 6\ 7$

$10^2\ 10^1\ 10^0$

Place Value: Real Numbers

In decimal:

5	6	7	.	8	9
10^2	10^1	10^0		10^{-1}	10^{-2}

Place Value: Real Numbers

In decimal:

5	6	7	.	8	9
10^2	10^1	10^0		10^{-1}	10^{-2}

In binary:

0	1	0	1
2^3	2^2	2^1	2^0
8	4	2	1

Place Value: Real Numbers

In decimal:

5	6	7	.	8	9
10^2	10^1	10^0		10^{-1}	10^{-2}

In binary:

0	1	0	1	.	1	1	0	0
2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
8	4	2	1		$1/_2$	$1/_4$	$1/_8$	$1/_{16}$

Place Value: Real Numbers

In decimal:

5	6	7	.	8	9
10^2	10^1	10^0		10^{-1}	10^{-2}

In binary:

0	1	0	1	.	1	1	0	0
2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
8	4	2	1		$^1/_2$	$^1/_4$	$^1/_8$	$^1/_{16}$

$$4 + 1 + 1/2 + 1/4 = 5\ 3/4 = 5.75$$

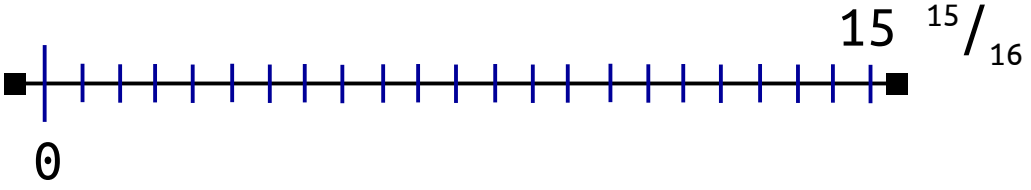
Fixed Point

8 bits: 4 for whole part, 4 for fraction

Whole part: 0 to 15

Fraction: 1/16 intervals

(No negatives for now)



Example

0	1	0	1	.	1	1	0	0
2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
8	4	2	1		$^1/_2$	$^1/_4$	$^1/_8$	$^1/_{16}$

4 + 1 + 1/2 + 1/4 = 5 3/4 = 5.75

0	0	1	0	.	1	0	0	0
2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
8	4	2	1		$^1/_2$	$^1/_4$	$^1/_8$	$^1/_{16}$

2 + 1/2 = 2.5

Adding Fixed Point

```
    1
    5.75
  + 2.5
  ------
    8.25
```

Adding Fixed Point

```
    1
    5.75        0101.1100
  + 2.5       + 0010.1000
-------     ---------------
    8.25
```

Adding Fixed Point

```
    1
    5.75        0101.1100
  + 2.5       + 0010.1000
-------     ---------------
    8.25             100
```

Adding Fixed Point

```
    1              1
    5.75       0101.1100
  + 2.5      + 0010.1000
-------    ---------------
    8.25           .0100
```

Adding Fixed Point

```
    1          1111
    5.75       0101.1100
  + 2.5      + 0010.1000
-------    ------------
    8.25       1000.0100
```

Adding Fixed Point

1	1111
5.75	0101.1100
+ 2.5	+ 0010.1000
8.25	1000.0100

1	0	0	0	.	0	1	0	0
2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
8	4	2	1		$^1/_2$	$^1/_4$	$^1/_8$	$^1/_{16}$

8 + 1/4 = 8.25

Limitations of Fixed Point

Numbers not exactly representable

(In decimal) 1/3 = 0.333333...

Limitations of Fixed Point

Numbers not exactly representable

(In decimal) 1/3 = 0.333333...

Different in binary

0.4 = 1/4 + 0.15 = 1/4 + 1/8 + 0.025

= 1/4 + 1/8 + 1/64 + 0.009375 = ...

Limitations of Fixed Point

Numbers not exactly representable

(In decimal) 1/3 = 0.333333...

Different in binary

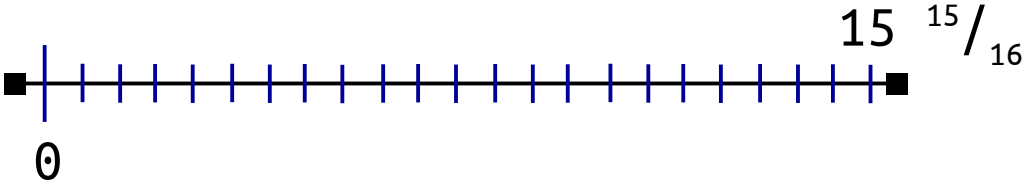
0.4 = 1/4 + 0.15 = 1/4 + 1/8 + 0.025

= 1/4 + 1/8 + 1/64 + 0.009375 = ...

Specific to fixed point

Narrow range of values (0 to 15.9375)

Constant resolution (1/16) for all numbers



So Far

Finish discussion of ints

- Signed/unsigned, larger integer types

Generalize binary polynomial to real numbers

- Fixed point representation

See the mechanics of floating point

Understand the limitations of floating point

- Epsilon (non-representable numbers)
- Arithmetic error

Different Magnitudes

3.5

35

3500

0.0000000035

3500000000000

Scientific Notation

$3.5 = 3.5 \cdot 10^{0}$

$35 = 3.5 \cdot 10^{1}$

$3500 = 3.5 \cdot 10^{3}$

$0.0000000035 = 3.5 \cdot 10^{-9}$

$3500000000000 = 3.5 \cdot 10^{12}$

Scientific Notation

$3.5 = 3.5 \cdot 10^{0}$

$35 = 3.5 \cdot 10^{1}$

$3500 = 3.5 \cdot 10^{3}$

$0.0000000035 = 3.5 \cdot 10^{-9}$

$3500000000000 = 3.5 \cdot 10^{12}$

Separate number into two pieces

- The magnitude (exponent)
- The significant figures

IEEE Floating Point

`float`: 32 bits

- 1 sign bit (sign + magnitude)
- 8 exponent bits
- 23 significand bits

$\pm 1.xxx \cdot 2^{yyy}$

IEEE Floating Point

`float`: 32 bits

- 1 sign bit (sign + magnitude)
- 8 exponent bits
- 23 significand bits

$\pm 1.xxx \cdot 2^{yyy}$

Range and resolution

- From 2^{-126} (10^{-38}) to 2^{127} (10^{38})
- 2^{23} bit patterns between each power of 2

IEEE Floating Point

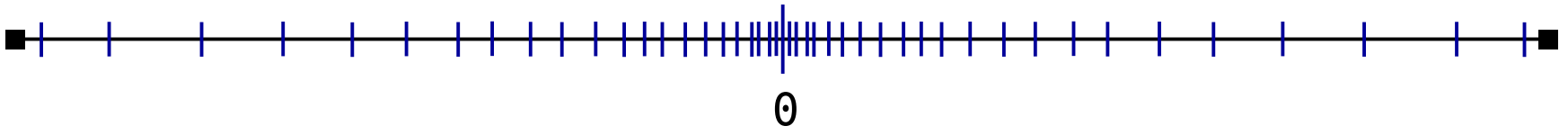
`float`: 32 bits

- 1 sign bit (sign + magnitude)
- 8 exponent bits
- 23 significand bits

$\pm 1.xxx \cdot 2^{yyy}$

Range and resolution

- From 2^{-126} (10^{-38}) to 2^{127} (10^{38})
- 2^{23} bit patterns between each power of 2



minifloat

`minifloat`: 8 bits (not a real C type!)

- 1 sign bit
- 4 exponent bits
- 3 significand bits

$\pm 1.xxx \cdot 2^{yyy}$

Parts of minifloat

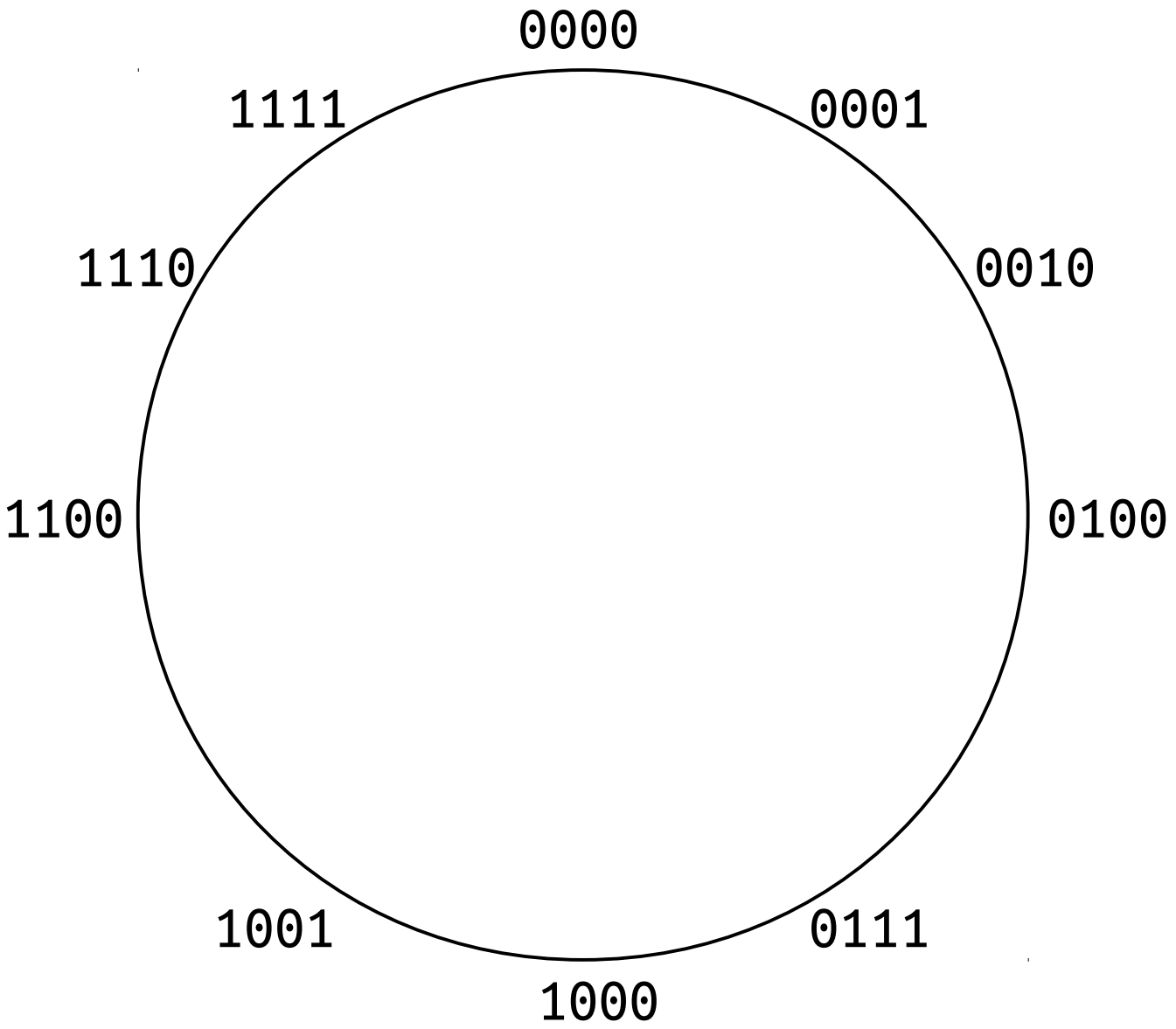
Sign: 1 = negative, 0 = positive

- x ⇒ -x: flip sign bit (not 2's complement)

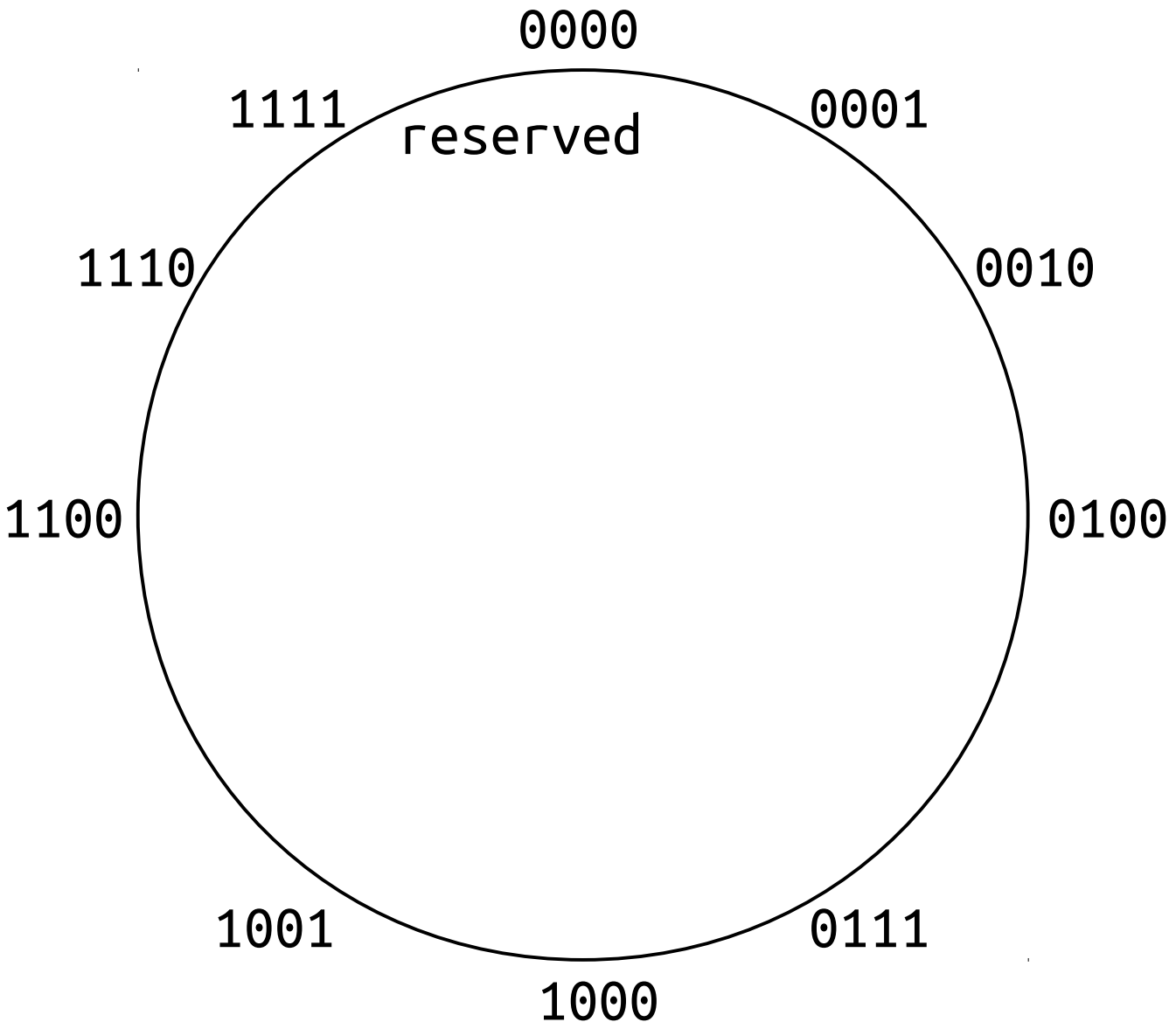
Exponent: from -6 to 7

- All 0s and all 1s reserved
- Biased

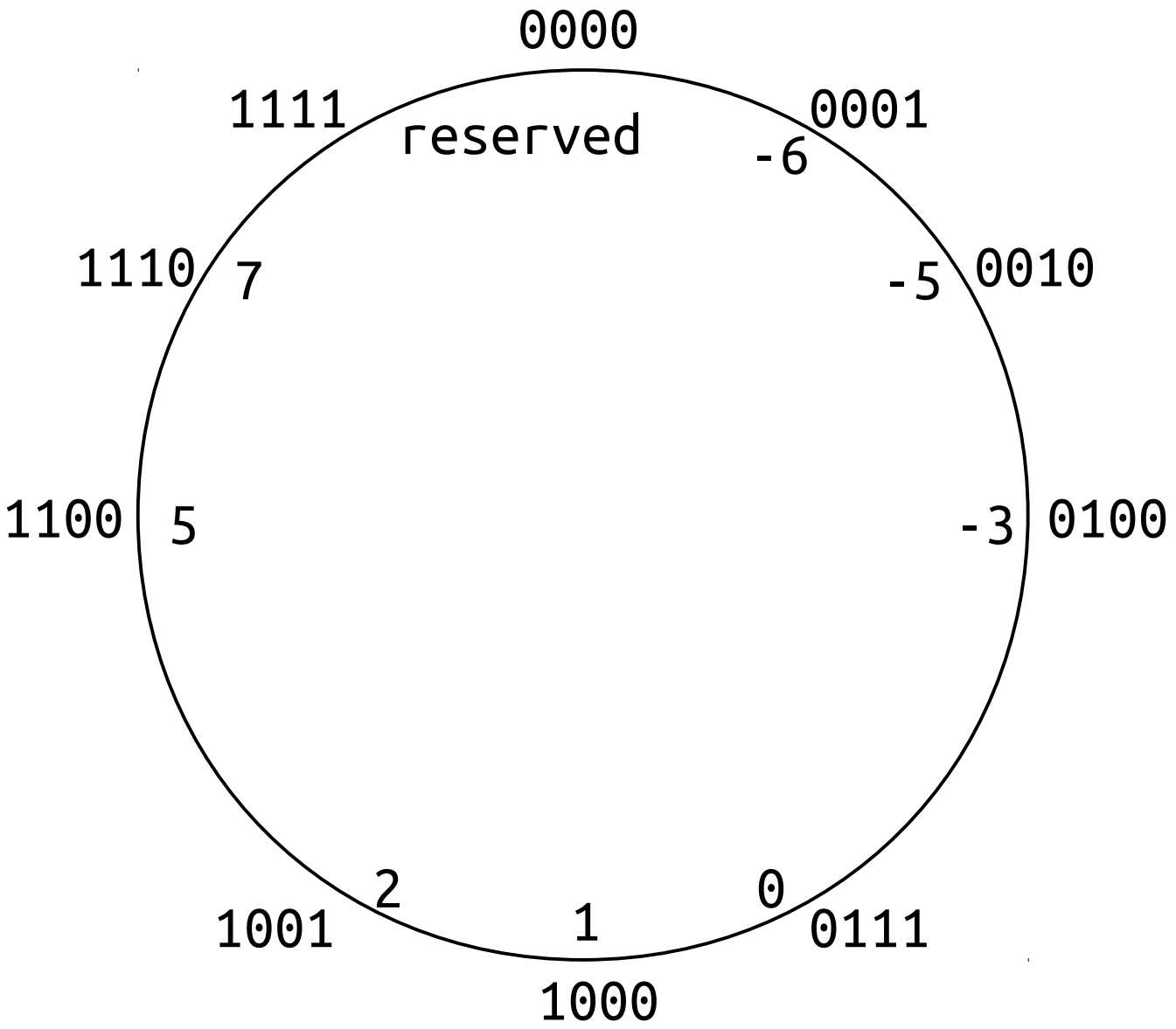
Biased Exponent



Biased Exponent



Biased Exponent



Parts of minifloat

Sign: 1 = negative, 0 = positive

x ⇒ -x: flip sign bit (not 2's complement)

Exponent: from -6 to 7

All 0s and all 1s reserved

Biased; bias = $2^{k-1} - 1$, where k = num exp. bits

E.g. bias of `minifloat` = $2^{4-1} - 1 = 7$

Parts of minifloat

Sign: 1 = negative, 0 = positive

x ⇒ -x: flip sign bit (not 2's complement)

Exponent: from -6 to 7

All 0s and all 1s reserved

Biased; bias = $2^{k-1} - 1$, where k = num exp. bits

E.g. bias of `minifloat` = $2^{4-1} - 1 = 7$

Significand: always 1.xxx (in binary)

Leading 1 not written

Parts of minifloat

Sign: 1 = negative, 0 = positive

x ⇒ -x: flip sign bit (not 2's complement)

Exponent: from -6 to 7

All 0s and all 1s reserved

Biased; bias = $2^{k-1} - 1$, where k = num exp. bits

E.g. bias of `minifloat` = $2^{4-1} - 1 = 7$

Significand: always 1.xxx (in binary)

Leading 1 not written

value: $\pm 1.SSS_2 \cdot 2^{EEEE-bias}$ bits: NEEEESSS bias = 7

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE-bias}$ bits: N EEEE SSS bias = 7

0 0111 000

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7}$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

0 0111 000 $\Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

0 0111 000 $\Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

0 1001 001

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

0 0111 000 $\Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

0 1001 001 $\Rightarrow 1.001_2 \cdot 2^{9-7}$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2 = 4.5$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2 = 4.5$

-5.0

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2 = 4.5$

$-5.0 = -101_2$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2 = 4.5$

$-5.0 = -101_2 = -1.01_2 \cdot 2^2 = -1.010_2 \cdot 2^{9-7}$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2 = 4.5$

$-5.0 = -101_2 = -1.01_2 \cdot 2^2 = -1.010_2 \cdot 2^{9-7} \Rightarrow 1\ 1001\ 010$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2 = 4.5$

$-5.0 = -101_2 = -1.01_2 \cdot 2^2 = -1.010_2 \cdot 2^{9-7} \Rightarrow 1\ 1001\ 010$

0.875

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2 = 4.5$

$-5.0 = -101_2 = -1.01_2 \cdot 2^2 = -1.010_2 \cdot 2^{9-7} \Rightarrow 1\ 1001\ 010$

$0.875 = 0.111_2$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2 = 4.5$

$-5.0 = -101_2 = -1.01_2 \cdot 2^2 = -1.010_2 \cdot 2^{9-7} \Rightarrow 1\ 1001\ 010$

$0.875 = 0.111_2 = 1.11_2 \cdot 2^{-1} = 1.110_2 \cdot 2^{6-7}$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2 = 4.5$

$-5.0 = -101_2 = -1.01_2 \cdot 2^2 = -1.010_2 \cdot 2^{9-7} \Rightarrow 1\ 1001\ 010$

$0.875 = 0.111_2 = 1.11_2 \cdot 2^{-1} = 1.110_2 \cdot 2^{6-7} \Rightarrow 0\ 0110\ 110$

Examples

value: $\pm 1.SSS_2 \cdot 2^{EEEE-bias}$ bits: N EEEE SSS bias = 7

$0\ 0111\ 000 \Rightarrow 1.000_2 \cdot 2^{7-7} = 1 \cdot 2^0 = 1.0$

$0\ 1001\ 001 \Rightarrow 1.001_2 \cdot 2^{9-7} = 1.125 \cdot 2^2 = 4.5$

$-5.0 = -101_2 = -1.01_2 \cdot 2^2 = -1.010_2 \cdot 2^{9-7} \Rightarrow 1\ 1001\ 010$

$0.875 = 0.111_2 = 1.11_2 \cdot 2^{-1} = 1.110_2 \cdot 2^{6-7} \Rightarrow 0\ 0110\ 110$

Notice: value of bit depends on exponent

$1.010_2 \cdot 2^2$

$2^0 = 1$

$1.110_2 \cdot 2^{-1}$

$2^{-3} = 1/8$

Exercise

value: $\pm 1.SSS_2 \cdot 2^{EEEE-bias}$ bits: N EEEE SSS bias = 7

0 1010 101

2.25

Exercise

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

0 1010 101 ⇒ $1.101_2 \cdot 2^{10-7} = 1.625 \cdot 2^3 = 13.0$

2.25

Exercise

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$0\ 1010\ 101 \Rightarrow 1.101_2 \cdot 2^{10-7} = 1.625 \cdot 2^3 = 13.0$

$2.25 = 10.01_2 = 1.001_2 \cdot 2^1 = 1.001_2 \cdot 2^{8-7} \Rightarrow 0\ 1000\ 001$

Limitation

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$8.5 = 1000.1_2 = 1.0001_2 \cdot 2^3$

Epsilon

value: $\pm 1.SSS_2 \cdot 2^{EEEE\text{-}bias}$ bits: N EEEE SSS bias = 7

$8.5 = 1000.1_2 = 1.0001_2 \cdot 2^3 \Rightarrow 1.000_2 \cdot 2^{10-7} = 8.0$

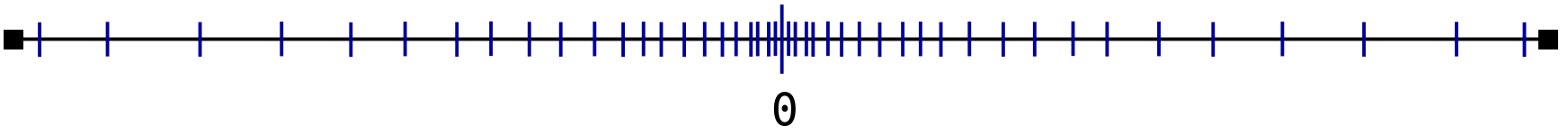
lost

Epsilon: distance to next representable value

Epsilon of 8.0 ($1.000_2 \cdot 2^3$) is 1.0 (neighbor is 9.0 ($1.001_2 \cdot 2^3$))

Cannot represent 8.5 (or 8.25, 8.75, ...)

Rounded to nearest



Denorms and Exceptional

Denormalized number: exp bits all 0

No implicit 1, exponent = `1 - bias`

$\pm 0.SSS_2 \cdot 2^{1\text{-bias}}$

Exceptional: exp bits all 1

Infinity (significand all 0)

NaN (anything else)

Floating Point Types

`float`: 32 bits

- 1 sign bit
- 8 exponent bits (bias = 127)
- 23 significand bits

`double`: 64 bits

- 1 sign bit
- 11 exponent bits (bias = 1023)
- 52 significand bits

So Far

Finish discussion of ints

- Signed/unsigned, larger integer types

Generalize binary polynomial to real numbers

- Fixed point representation

See the mechanics of floating point

Understand the limitations of floating point

- Epsilon (non-representable numbers)
- Arithmetic error

Code: FP Errors

Representation issues

- Can't represent many decimals

Arithmetic issues

- Not associative
- Add/subtract numbers of different magnitude

Epsilon issues

- Gaps get very large as number increases

Summary

Finish discussion of ints

- Signed/unsigned, larger integer types

Generalize binary polynomial to real numbers

- Fixed point representation

See the mechanics of floating point

Understand the limitations of floating point

- Epsilon (non-representable numbers)
- Arithmetic error