Binary Number Systems

Positional Notation

```
4     3     2     1     0
```

- Allows us to count past 10.
- Each column of a number represents a power of the base. The exponent is the order of magnitude for the column.

Positional Notation

$$10^4 \quad 10^3 \quad 10^2 \quad 10^1 \quad 10^0$$

- The Decimal system is based on the number of digits we have.

Positional Notation

10^4	10^3	10^2	10^1	10^0
10000	1000	100	10	1

- The magnitude of each column is the base, raised to its exponent.

Positional Notation

10^4	10^3	10^2	10^1	10^0
10000	1000	100	10	1
2	7	9	1	6
20000	+7000	+900	+10	+6

=27916

- The magnitude of a number is determined by multiplying the magnitude of the column by the digit in the column and summing the products.

Binary Numbers

The base in a Binary system is 2.

There are only 2 digits – 0 and 1.

Since we use the term frequently, “binary digit” can be shortened to ‘bit’.

8 bits together form a byte.

A Single Byte

7	6	5	4	3	2	1	0

A Single Byte

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

A Single Byte

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

A Single Byte

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

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

=255

is the largest decimal value that can be expressed in 8 bits.

A Single Byte

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

0 +0 +0 +0 +0 +0 + 0 + 0

=0

There is also a representation for zero, making 256 (2^8) combinations of 0 and 1 in 8 bits.

Longer Numbers

Since 255 is the largest number that can be represented in 8 bits, lager values simply require longer numbers.

For example, 27916 is represented by:

0011011010000110

Longer Numbers

Since 255 is the largest number that can be represented in 8 bits, lager values simply require longer numbers.

For example, 27916 is represented by:

Can you remember the Binary representation?

Short Forms for Binary

Because large numbers require long strings of Binary digits, short forms have been developed to help deal with them.

An early system used was called Octal.

It's based on the 8 patterns in 3 bits.

Short Forms for Binary - Octal

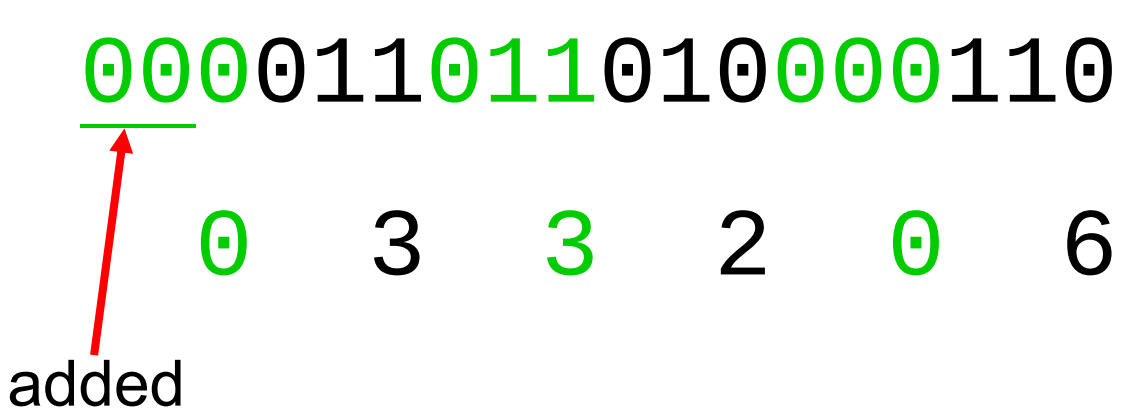
111	7
110	6
101	5
100	4
011	3
010	2
001	1
000	0

0011011010000110

can be short-formed by dividing the number into 3 bit chunks (starting from the least significant bit) and replacing each with a single Octal digit.

Short Forms for Binary - Octal

111	7
110	6
101	5
100	4
011	3
010	2
001	1
000	0



Short Forms for Binary - Hexadecimal

0111	7	1111	F
0110	6	1110	E
0101	5	1101	D
0100	4	1100	C
0011	3	1011	B
0010	2	1010	A
0001	1	1001	9
0000	0	1000	8

It was later determined that using base 16 and 4 bit patterns would be more efficient.

But since there are only 10 numeric digits, 6 letters were borrowed to complete the set of hexadecimal digits.

Short Forms for Binary - Hexadecimal

0111	7	1111	F
0110	6	1110	E
0101	5	1101	D
0100	4	1100	C
0011	3	1011	B
0010	2	1010	A
0001	1	1001	9
0000	0	1000	8

`0011011010000110`

can be short-formed by dividing the number into 4-bit chunks (starting from the least significant bit) and replacing each with a single Hexadecimal digit.

Short Forms for Binary - Hexadecimal

0111	7	1111	F
0110	6	1110	E
0101	5	1101	D
0100	4	1100	C
0011	3	1011	B
0010	2	1010	A
0001	1	1001	9
0000	0	1000	8

```
00110110100000110

   3    6    8    6
```

Integers

- To store integers, half the combinations are used to represent negative values.
- The MSB is used to represent the sign.
- The range for Integers in 1 byte is:

  -128 to +127

- Which value of the sign bit (0 or 1) will represent a negative number?

Excess Notation

- The notation system that uses 0 to represent negative values.
- Fixed length notation system.
- Zero is the first non-negative value:
  - 10000000
- The pattern immediately before zero is -1:
  - 01111111
- The largest value is stored as 11111111 (+127)
- The smallest value is stored as 00000000 (-128)

2's Complement Notation

- The notation system that uses 1 to represent negative values.
- Fixed length notation system.
- Zero is the first non-negative value:
  - 00000000
- The pattern immediately before zero is -1:
  - 11111111
- The largest value is stored as 01111111 (+127)
- The smallest value is stored as 10000000 (-128)

Interpretations of Binary Patterns

Binary	Decimal	Hexadecimal	Excess	2's Complement
1111	15	F	7	-1
1110	14	E	6	-2
1101	13	D	5	-3
1100	12	C	4	-4
1011	11	B	3	-5
1010	10	A	2	-6
1001	9	9	1	-7
1000	8	8	0	-8
0111	7	7	-1	7
0110	6	6	-2	6
0101	5	5	-3	5
0100	4	4	-4	4
0011	3	3	-5	3
0010	2	2	-6	2
0001	1	1	-7	1
0000	0	0	-8	0

Arithmetic in 2's Complement

(remember it's a *fixed length* system)

```
00 + 00 = 00
00 + 01 = 01
01 + 00 = 01
01 + 01 = 10
```

-1	in 2's complement	11111111
+ 1	in 2's complement	+ 00000001
0	discard the carry bit	1 00000000

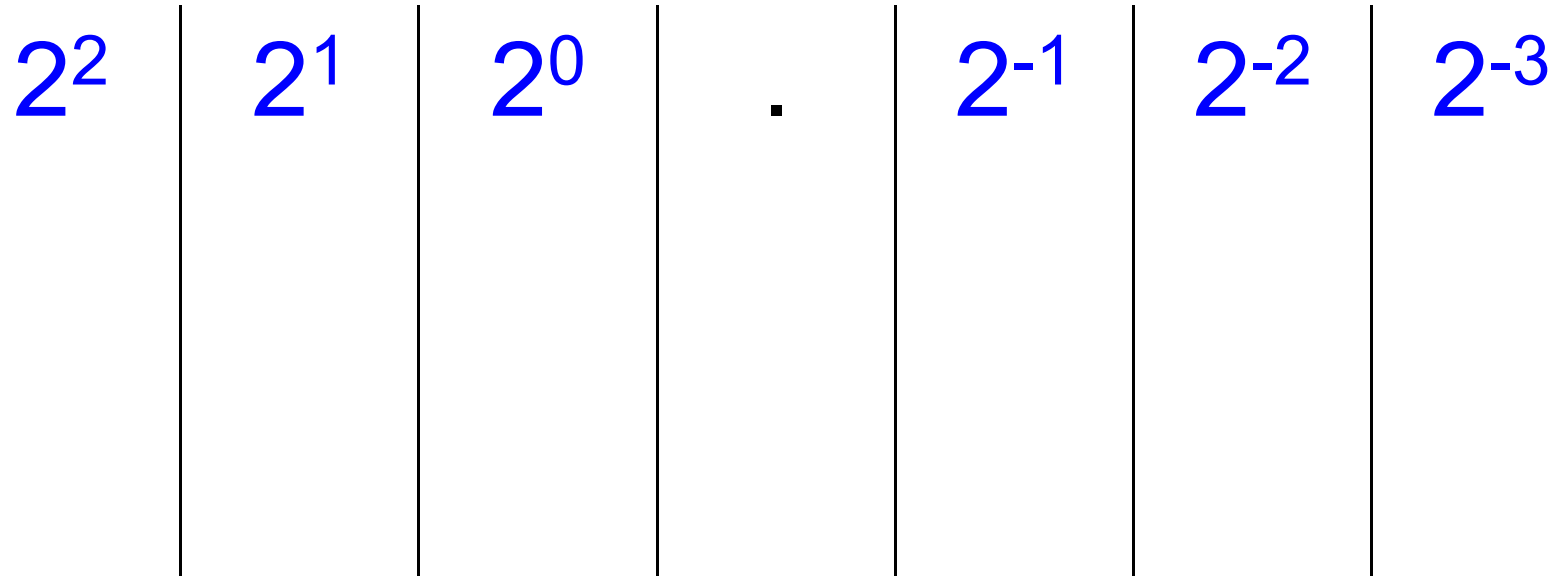
Fractions

- A radix separates the integer part from the fraction part of a number.

101.101

- Columns to the right of the radix have negative powers of 2.

Fractions



Fractions

2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}
4	2	1	.	½	¼	⅛

Fractions

2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}
4	2	1	.	½	¼	⅛
1	0	1	.	1	0	1

Fractions

2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}
4	2	1	.	½	¼	⅛
1	0	1	.	1	0	1
4	+	1	+	½	+	⅛

Fractions

2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}
4	2	1	.	½	¼	⅛
1	0	1	.	1	0	1
4	+	1	+	½	+	⅛

5⅝

Scientific Notation

Very large and very small numbers are often represented such that their order of magnitude can be compared.

The basic concept is an exponential notation using powers of 10.

$$a \times 10^{b}$$

Where b is an integer,
and a is a real number such that:

$$1 \leq |a| < 10$$

Scientific Notation

An electron's mass is about

0.000000000000000000000000000091093826 kg.

In scientific notation, this is written

$9.1093826 \times 10^{-31}$ kg.

The Earth's mass is about

5,973,600,000,000,000,000,000 kg.

In scientific notation, this is written

5.9736×10^{24} kg.

E Notation

To allow values like this to be expressed on calculators and early terminals

$$\times 10^{b}$$

was replaced by Eb

So $9.1093826 \times 10^{-31}$

becomes 9.1093826E−31

And 5.9736×10^{24}

becomes 5.9736E+24

E Notation

The ‘a’ part of the number is called the mantissa or significand.

The ‘Eb’ part is called the exponent.

Since these numbers could also be negative they would typically have a sign as well.

Floating Point Storage

In floating point notation the bit pattern is divided into 3 components:

- Sign – 1 bit (0 for +, 1 for -)
- Exponent – stored in Excess notation
- Mantissa – must begin with 1

Mantissa

- Assumes a radix point immediately left of the first digit.
- The exponent will determine how far and in which direction to move the radix.

An example in 8 bits

If the following pattern stores a floating point value, what is it?

01101001

An example in 8 bits

If the following pattern stores a floating point value, what is it?

01101001

Separate it into its components:

An example in 8 bits

If the following pattern stores a floating point value, what is it?

01101001

Separate it into its components:

- sign
- exponent
- mantissa

An example in 8 bits

If the following pattern stores a floating point value, what is it?

0 110 1001

Separate it into its components:

- sign
- exponent
- mantissa

An example in 8 bits

0 110 1001

A sign bit of 0 means the number is…?

An example in 8 bits

0 110 1001

A sign bit of 0 means the number is positive.

110 in Excess Notation converts to …?

An example in 8 bits

0 110 1001

A sign bit of 0 means the number is positive.

110 in Excess Notation converts to +2.

Place the radix in the mantissa …

An example in 8 bits

0 110 1001

A sign bit of 0 means the number is positive.

110 in Excess Notation converts to +2.

Place the radix in the mantissa .1001

Put it all together …

An example in 8 bits

0 110 1001

A sign bit of 0 means the number is positive.

110 in Excess Notation converts to +2.

Place the radix in the mantissa .1001

Put it all together …

$+ .1001 * 2^{2}$

An example in 8 bits

$$+ .1001 * 2^{2}$$

Multiplying a binary number by 2 shifts the bits left (moves the radix to the right) one position.

So this exponent tells us to shift the radix **2** positions right.

$$+ 10.01$$

$$= 2\frac{1}{4}$$

Normal Form

- The first bit of the mantissa must be 1 to prevent multiple representations of the same value. Otherwise…

`0 100 1000`	0	`.1000`	`.1000`
`0 101 0100`	1	`.0100`	`.1000`
`0 110 0010`	2	`.0010`	`.1000`
`0 111 0001`	3	`.0001`	`.1000`

Sample Test 1 Question

A pattern of binary digits can be interpreted in several different ways.

Show how the pattern **01011010** translates using each of the following interpretations. [1 point each]

Octal short form	
Hexadecimal short form	
Unsigned integer	
2's complement	
Excess 128 notation	
Floating Point notation	

Sample Test 1 Question

A pattern of binary digits can be interpreted in several different ways.

Show how the pattern **01011010** translates using each of the following interpretations. [1 point each]

Octal short form	**132**
Hexadecimal short form	
Unsigned integer	
2's complement	
Excess 128 notation	
Floating Point notation	

Sample Test 1 Question

A pattern of binary digits can be interpreted in several different ways.

Show how the pattern **01011010** translates using each of the following interpretations. [1 point each]

Octal short form	**132**
Hexadecimal short form	**5A**
Unsigned integer	
2's complement	
Excess 128 notation	
Floating Point notation	

Sample Test 1 Question

A pattern of binary digits can be interpreted in several different ways.

Show how the pattern **01011010** translates using each of the following interpretations. [1 point each]

Octal short form	**132**
Hexadecimal short form	**5A**
Unsigned integer	**90**
2's complement	
Excess 128 notation	
Floating Point notation	

Sample Test 1 Question

A pattern of binary digits can be interpreted in several different ways.

Show how the pattern **01011010** translates using each of the following interpretations. [1 point each]

Octal short form	**132**
Hexadecimal short form	**5A**
Unsigned integer	**90**
2's complement	**+90**
Excess 128 notation	
Floating Point notation	

Sample Test 1 Question

A pattern of binary digits can be interpreted in several different ways.

Show how the pattern **01011010** translates using each of the following interpretations. [1 point each]

Octal short form	**132**
Hexadecimal short form	**5A**
Unsigned integer	**90**
2's complement	**+90**
Excess 128 notation	**-38**
Floating Point notation	

Sample Test 1 Question

A pattern of binary digits can be interpreted in several different ways.

Show how the pattern **01011010** translates using each of the following interpretations. [1 point each]

Octal short form	**132**
Hexadecimal short form	**5A**
Unsigned integer	**90**
2's complement	**+90**
Excess 128 notation	**-38**
Floating Point notation	**+1¼**