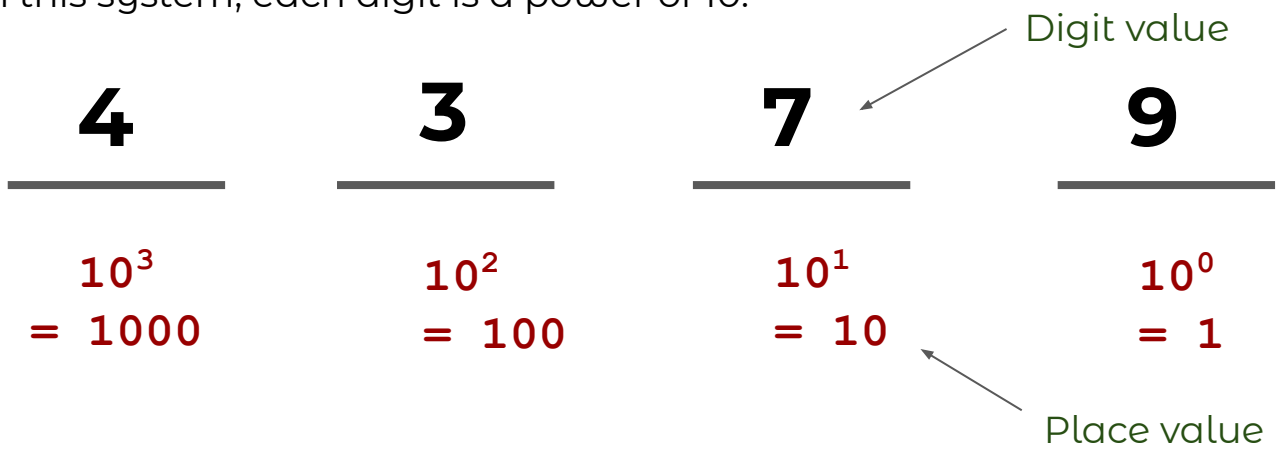
Introduction to Binary

The binary system is a way of representing data using 0s and 1s. This system is used by computers to represent all the data it works with.

```
1011110000011111
0010101000100101
0101010011111001
0100001001010100
0010010010101010
0011001011
```

Decimal System:

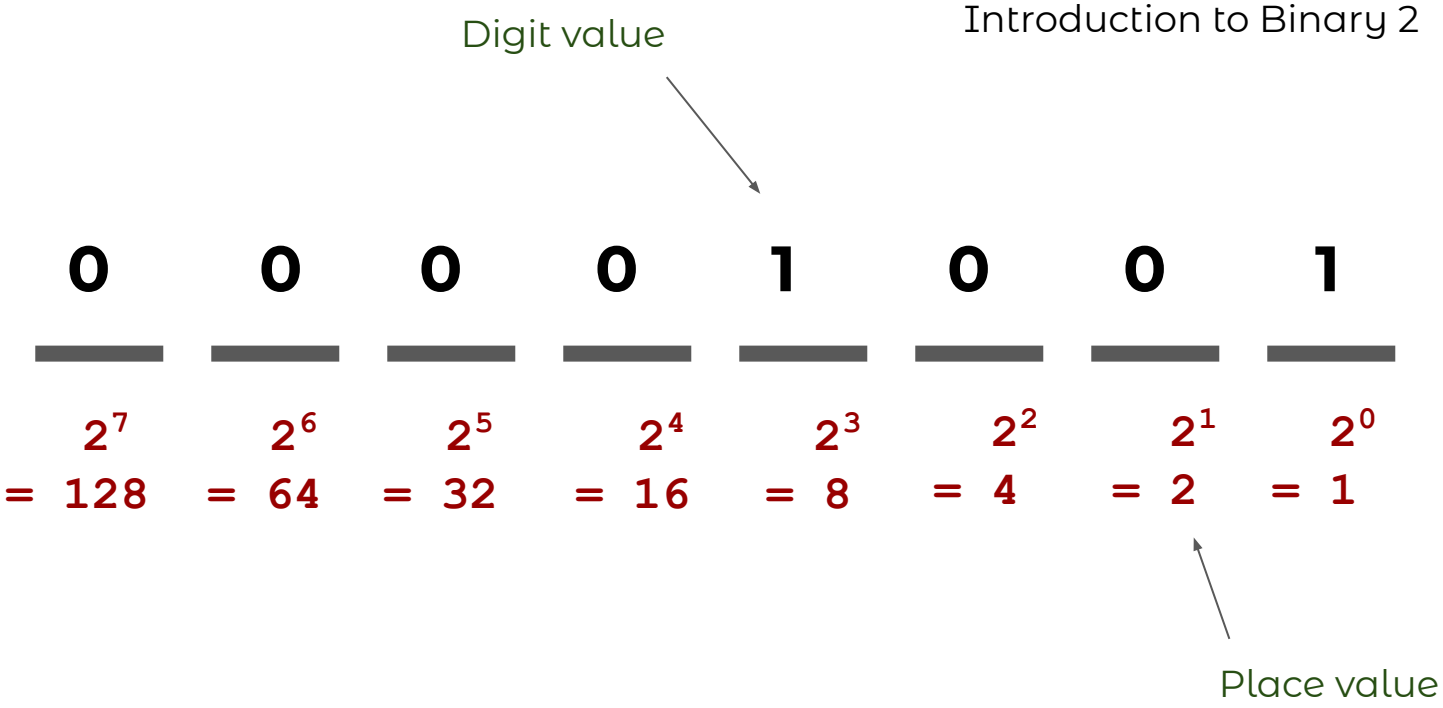
The number system we use in everyday life is called the decimal system, where we use the numbers 1,2,3,4,5,6,7,8,9 to represent all of our numbers. In this system, each digit is a power of 10.



To compute a number in this system, you would multiply the digit value by the place value, then add them all together. So to represent the number 4379, you would have to compute :

(4*1000)+(3*100)+(7*10)+(9*1)=4379.

The binary system follows a similar structure, except for each digit, you only have 2 values, 0 and 1.



To compute a number in this system, you would multiply the digit value by the place value, then add them all together. So to translate the binary number 1001, you would have to compute:

(1*8)+(0*4)+(0*2)+(1*1) = 9

To convert this number back to binary, find the largest power of 2 and subtract until your total is 0

9 - (1 * $\mathbf{2^3}$) = 9 - **8** = 1	so 4th digit is 1
$\mathbf{2^2 = 4}$ > 1	so 3rd digit is 0
$\mathbf{2^1 = 2}$ > 1	so 2nd digit is 0
1 - (1 * $\mathbf{1^0}$) = 1 - **1** = 0	so 1st digit is 1

So the resulting number is 1001.

Let's try a few!

Binary Practice Problems

Digit value

Clue!

If the decimal value is odd, then the 1st digit must be a 1

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

Place value

Refer to this for help!

Convert the following decimal values into binary:

1) 8 _ _ _ _ _ _ _ _

2) 17 _ _ _ _ _ _ _ _

3) 11 _ _ _ _ _ _ _ _

4) 36 _ _ _ _ _ _ _ _

5) 70 _ _ _ _ _ _ _ _

6) 132 _ _ _ _ _ _ _ _

Binary Practice Problems

Digit value

Clue!

If the 1st digit is a 1, then the decimal value must be odd

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

Place value

Refer to this for help!

Convert the following binary values into decimal:

1) 0 0 1 0 _____

2) 1 1 1 0 _____

3) 1 0 1 1 _____

4) 0 0 1 0 0 0 0 1 _____

5) 0 0 0 1 1 0 1 0 _____

6) 1 0 1 0 1 0 1 0 _____

ANSWER KEY

Binary Practice Problems

Digit value

Clue!

If the decimal value is odd, then the 1st digit must be a 1

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

Place value

Refer to this for help!

Convert the following decimal values into binary:

1) 8 — 0 0 0 0 1 0 0 0

2) 17 — 0 0 0 1 0 0 0 1

3) 11 — 0 0 0 0 1 0 1 1

4) 36 — 0 0 1 0 0 1 0 0

5) 70 — 0 1 0 0 0 1 1 0

6) 132 — 1 0 0 0 0 1 0 0

ANSWER KEY

Binary Practice Problems

Digit value

Clue!

If the 1st digit is a 1, then the decimal value must be odd

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

Place value

Refer to this for help!

Convert the following binary values into decimal:

1) 0 0 1 0 — 2

2) 1 1 1 0 — 14

3) 1 0 1 1 — 11

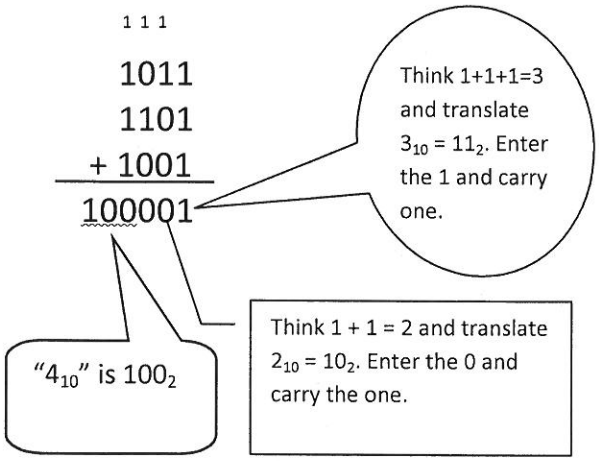
4) 0 0 1 0 0 0 0 1 — 33

5) 0 0 0 1 1 0 1 0 — 26

6) 1 0 1 0 1 0 1 0 — 170

Adding Binary Numbers

Example: Find the sum of 1011_2, 1101_2, and 1001_2.

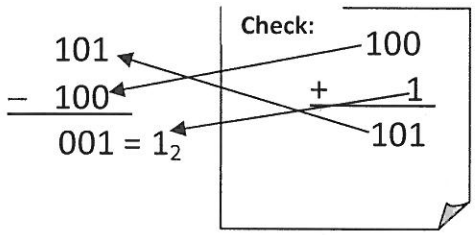


Now you try one:

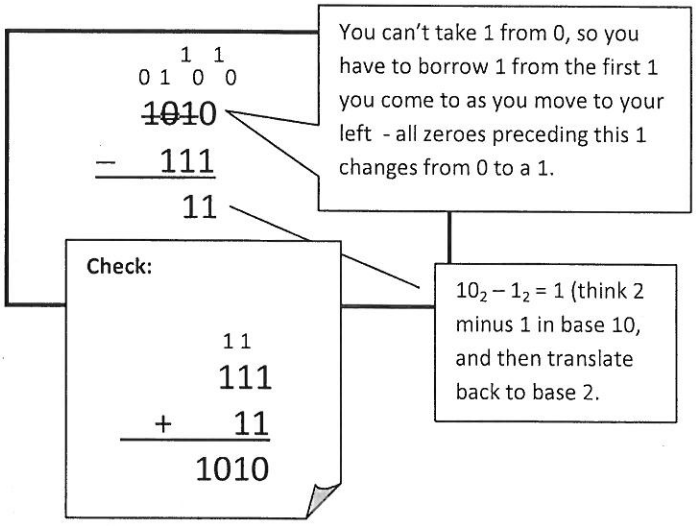
$$\begin{array}{r} 10101 \\ 1111 \\ +\ 1010 \\ \hline \end{array}$$

Subtracting Binary Numbers

Example 1: Find the difference between 101_2 and 100_2.



Example 2: Find the difference of 1010_2 and 111_2.



Example 3: Here is another example where borrowing is necessary:

$$\begin{array}{r} {\scriptstyle 0\ 1\ 1\ 1\ 1\ \overset{1}{0}} \\ \cancel{100000} \\ -\quad\ 1 \\ \hline 011111 \end{array}$$

Now you try one:

$$\begin{array}{r} 10010 \\ -\quad 101 \\ \hline \end{array}$$

Rules of Binary Addition

- `0 + 0 = 0`
- `0 + 1 = 1`
- `1 + 0 = 1`
- `1 + 1 = 0`, and carry 1 to the next more significant bit

For example,

`00011010 + 00001100 = 00100110`

			1	*1*					*carries*
	0	0	0	1	1	0	1	0	= $26_{(\text{base } 10)}$
+	0	0	0	0	1	1	0	0	= $12_{(\text{base } 10)}$
	0	0	1	0	0	1	1	0	= $38_{(\text{base } 10)}$

`00010011 + 00111110 = 01010001`

		1	*1*	*1*	*1*	*1*			*carries*
	0	0	0	1	0	0	1	1	= $19_{(\text{base } 10)}$
+	0	0	1	1	1	1	1	0	= $62_{(\text{base } 10)}$
	0	1	0	1	0	0	0	1	= $81_{(\text{base } 10)}$

Note: The rules of binary addition (without carries) are the same as the truths of the **XOR** gate.

Rules of Binary Subtraction

- `0 - 0 = 0`
- `0 - 1 = 1`, and borrow 1 from the next more significant bit
- `1 - 0 = 1`
- `1 - 1 = 0`

For example,

`00100101 - 00010001 = 00010100`

			0						*borrows*
	0	0	~~1~~	^{1}0	0	1	0	1	= $37_{(\text{base } 10)}$
-	0	0	0	1	0	0	0	1	= $17_{(\text{base } 10)}$
	0	0	0	1	0	1	0	0	= $20_{(\text{base } 10)}$

`00110011 - 00010110 = 00011101`

			0	^{1}*0*	*1*				*borrows*
	0	0	~~1~~	~~1~~	~~0~~	^{1}0	1	1	= $51_{(\text{base } 10)}$
-	0	0	0	1	0	1	1	0	= $22_{(\text{base } 10)}$
	0	0	0	1	1	1	0	1	= $29_{(\text{base } 10)}$