Fundamentals of Computer Engineering

Module III - Unit 9
Information and Data

Teachers: Moisés Martínez (1ºA English)

Year: 2022 - 2023

Universidad Francisco de Vitoria
UFV Madrid

Grado en Ingeniería Informática
Escuela Politécnica Superior

What is Data?

What is Information?

Data and Information are equal?

Information vs Data

Data vs Information

Data is unorganised and unrefined **raw** facts.

Information is the **organization** and **interpretation** of those facts

Information vs Data

Both concepts have an important role in Computer Sciences but there are significant differences between them.

Data	Information
Data refers to raw facts that have no specific meaning.	Information refers to processed data that has a purpose and meaning.
Data is independent of the information.	Information is dependent on data.
Data or Raw Data is not enough to make a decision.	Information is usually sufficient to help make a decision in a specific context.

Universidad Francisco de Vitoria UFV Madrid

Information in computers

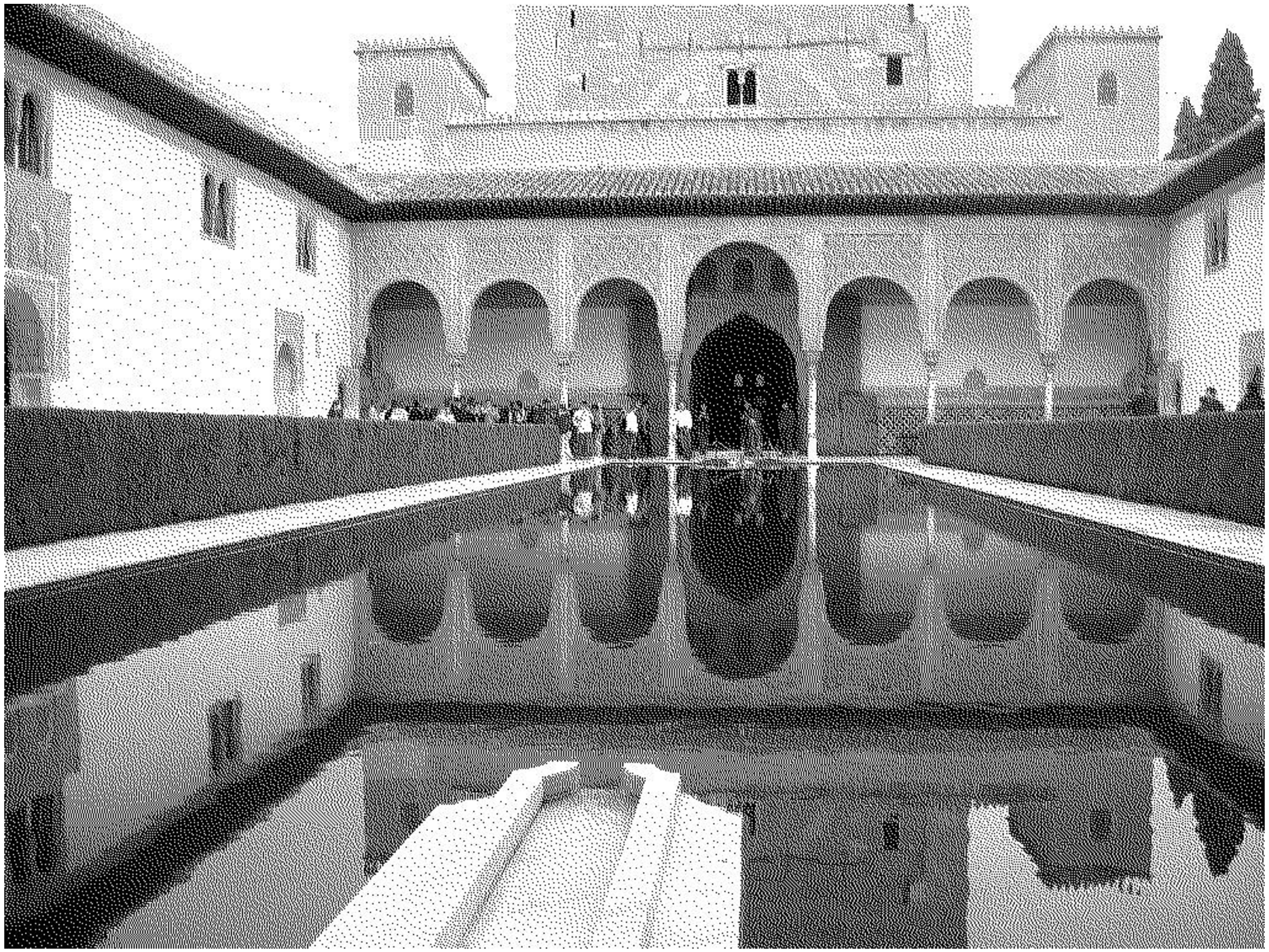
Information in computers.

A bit is a **binary digit** and it is the **smallest unit of data** on a computer. Bits can hold only one of two values: 0 or 1.

A byte is the **smallest addressable memory** in most computers that can store data that is smaller than a byte. Based on the context, it can represent different types of information:

- Letter
- Number
- Program instruction.
- Pixel in an image or part of an audio recording.

Information in computers.



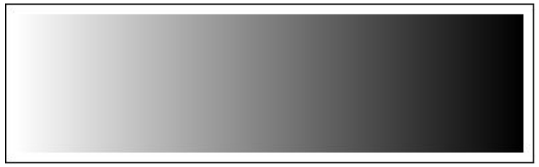
Picture created using 1 bit: 2 colors.

Universidad Francisco de Vitoria UFV Madrid

❗ OFF-EXAM CONTENT

Information in computers.

Information is modeled using a **gray scale** where each pixel can represent one shade of gray (0 - 256).



Black □ 00000000

White □ 11111111

Picture created using 8 bits: 256 colors.

Universidad Francisco de Vitoria **UFV** Madrid

❗ OFF-EXAM CONTENT

Information in computers.

Information is modeled using several color layers (Red, Green, Blue) and one for the luminosity factor.

Black □ 00000000

White □ 11111111

16.7 Million Colors

Picture created using 8 bits: 256 colors per channel.

Universidad Francisco de Vitoria UFV Madrid

❗ OFF-EXAM CONTENT

Information in computers.

Information is modeled using several color layers (Red, Green, Blue) and one for the luminosity factor.

Black □ 00000000

White □ 11111111

281 Trillion Colors

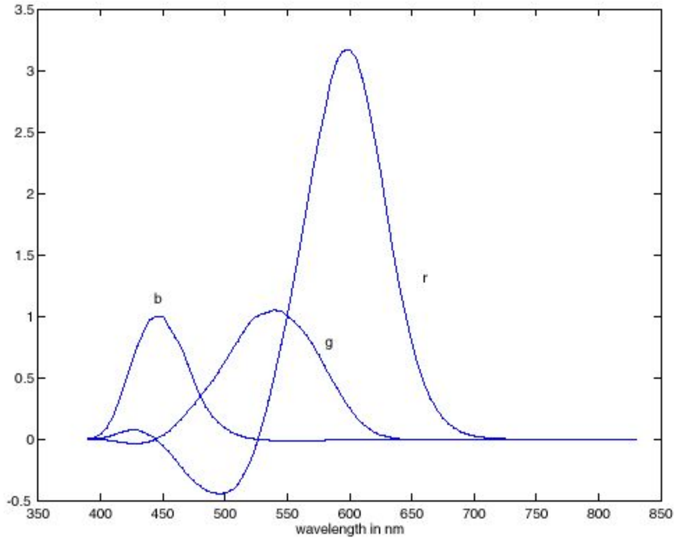
Picture created using 16 bits: 65536 colors per channel.

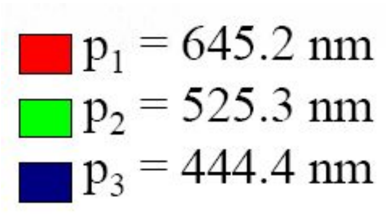
Universidad Francisco de Vitoria UFV Madrid

! OFF-EXAM CONTENT

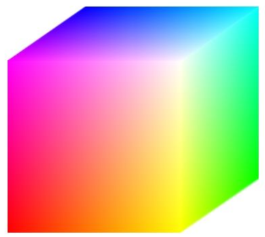
Information in computers.

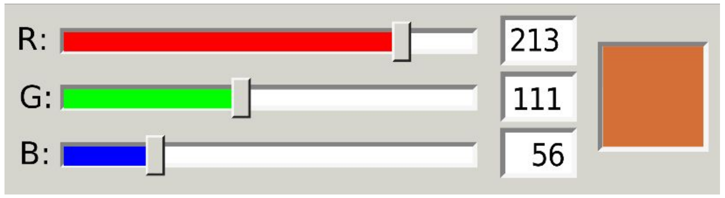
The **RGB color model** is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary/secondary colors, Red, Green, and Blue.





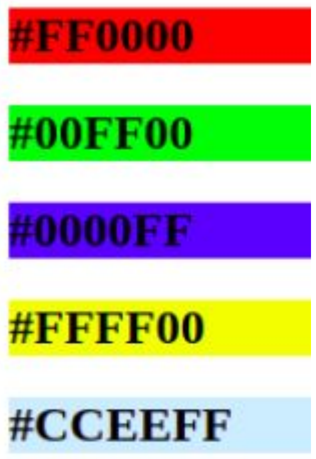
🟥 $p_1 = 645.2$ nm
🟩 $p_2 = 525.3$ nm
🟦 $p_3 = 444.4$ nm





R: 213
G: 111
B: 56

Hex representation



#FF0000
#00FF00
#0000FF
#FFFF00
#CCEEFF

Each color is represented by a **hex** number of 2 digits.

OFF-EXAM CONTENT

Information in computers.

A **numeral system or a number system** is a writing system for expressing numbers; that is, a mathematical notation for representing numbers of a given set, using digits or other symbols in a consistent manner.

- Represent a useful set of numbers.
- Give every number represented a unique representation (or at least a standard representation).
- Reflect the algebraic and arithmetic structure of the numbers.

The **cardinal** (number of elements) of a set of numbers is called the **base** of a numerical system.

Decimal system: base = 10, digits = {0,1,2,3,4,5,6,7,8,9}

Information in computers.

Standard positional representation of a number N in base b is written as follow:

$$N = (a_n a_{n-1} a_{n-2} \ldots a_1 a_0\ a_{-1} \ldots a_{-m})_b$$

where:

- $a_i \rightarrow$ digits constituting the number (between 0 and r-1)
- $n \rightarrow$ number of integer digits
- $m \rightarrow$ number of fractional digits
- $a_n \rightarrow$ most significant digit
- $a_{-m} \rightarrow$ least significant digit
- $r^i \rightarrow$ weight of digit i
- $a_i * r^i \rightarrow$ value of digit i

Universidad Francisco de Vitoria UFV Madrid

Information in computers.

The value of a number N is given by:

$$(N)_b = a_{n-1} \times b^{n-1} + a_{n-2} \times b^{n-2} \ldots + a_1 \times b^1 + a_0 \times b^0 + a_{-1} \times b^{-1} \ldots + a_{-m} \times b^{-m}$$

where “b” is the base of the number system (e.g 2, 8, 10 or 16) and “a” is a digit that range from 0 to b-1.

$$(352.45)_{10} = 3 \times 10^2 + 5 \times 10^1 + 2 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

$$= 3 \times 100 + 5 \times 10 + 2 \times 0 + 4 \times 0{,}1 + 5 \times 0{,}01$$

Universidad Francisco de Vitoria UFV Madrid

Information in computers.

Base conversion is the process of convert a number N from one base number to another base number. It is enough to express the number to be converted in **polynomial notation**, expressing the digits and weights in base s, and operate in base s:

$$N = (10101)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (21)_{10}$$

$$M = (14)_{16} = 1 \times 16^1 + 4 \times 16^0 = (20)_{10}$$

Universidad Francisco de Vitoria UFV Madrid

Information in computers.

A positional (numeral) system is a system for representation of numbers by an ordered set of numerals symbols (called digits) in which the value of a numeral symbol depends on its position.

- Binary system represents information using digits from 0 to 1.

N =	1	0	1	1
B =	3	2	1	0

Information in computers.

A positional (numeral) system is a system for representation of numbers by an ordered set of numerals symbols (called digits) in which the value of a numeral symbol depends on its position.

- Binary system represents information using digits from 0 to 1.

N =	1	0	1	1
B =	3	2	1	0

- Decimal system represents information using digits from 0 to 9.

N =	4	2	1	4
B =	3	2	1	0

Information in computers.

A positional (numeral) system is a system for representation of numbers by an ordered set of numerals symbols (called digits) in which the value of a numeral symbol depends on its position.

- Binary system represents information using digits from 0 to 1.

N =	1	0	1	1
B =	3	2	1	0

- Decimal system represents information using digits from 0 to 9.

N =	4	2	1	4
B =	3	2	1	0

- Hexadecimal system represents information using digits from 0 to 9 and letter from A to F.

N =	A	F	7	1
B =	3	2	1	0

Universidad
Francisco de Vitoria
UFV Madrid

From Binary to Decimal

From Binary to Decimal

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

How can we number n_2 convert from a base 2 to base 10?

$$n_2 = 11001010$$

From Binary to Decimal

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

$$\left.\begin{array}{c}1\\1\\0\\0\\1\\0\\1\\0\end{array}\right\} \mathbf{n = 8}$$

First we count the number of digits in our binary number.

Universidad Francisco de Vitoria UFV Madrid

From Binary to Decimal

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

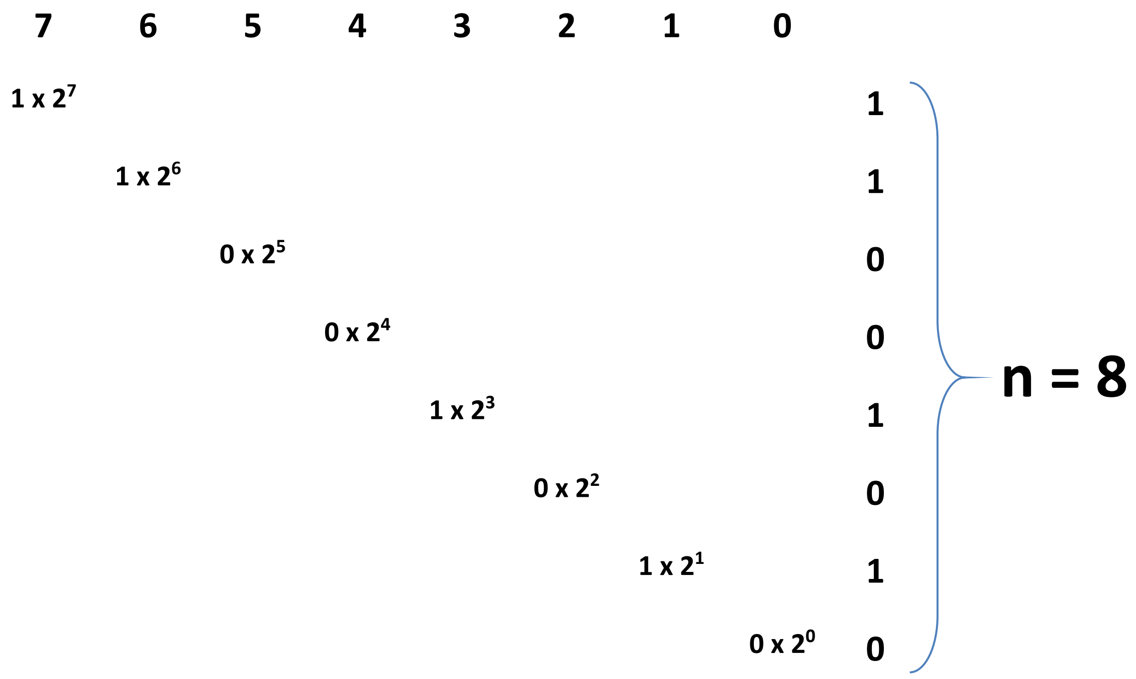
$$\left.\begin{array}{c}1\\1\\0\\0\\1\\0\\1\\0\end{array}\right\} \mathbf{n = 8}$$

Next, we multiply from left to right each digit by the power of two that corresponds to it, starting with 2^{n-1}

From Binary to Decimal

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal



Next, we multiply from left to right each digit by the power of two that corresponds to it, starting with 2^{n-1}

From Binary to Decimal

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

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

n = 8

Finally, we add all those values.

From Binary to Decimal

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

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

n = 8

= 202

Universidad
Francisco de Vitoria
UFV Madrid

From Binary to Octal

From Binary to Octal

Positional (numeral) systems

- Binary
- Decimal
- Octal

How can we number n_2 convert from a base 2 to base 8?

$$n_2 = 100111011101111$$

Universidad Francisco de Vitoria UFV Madrid

From Binary to Octal

Positional (numeral) systems

$$\begin{tikzcd}
\bullet\ \text{Binary} \arrow[dd, bend left] \\
\bullet\ \text{Decimal} \\
\bullet\ \text{Octal}
\end{tikzcd}$$

$$\underbrace{100}_{}\underbrace{111}_{}\underbrace{011}_{}\underbrace{101}_{}\underbrace{111}_{}$$

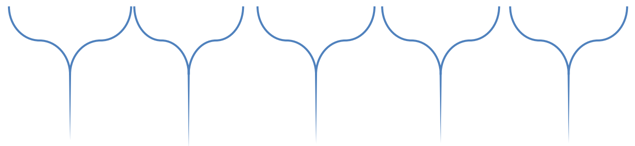
We divide bits into groups of 3 from right to left. Why?

From Binary to Octal

Positional (numeral) systems

- Binary
- Decimal
- Octal

100111011101111



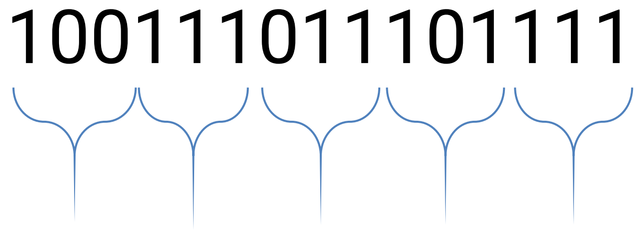
We divide bits into groups of 3 from right to left. Why?

We need 3 bits to represent numbers from 0 to 7.

From Binary to Octal

Positional (numeral) systems

- Binary
- Decimal
- Octal



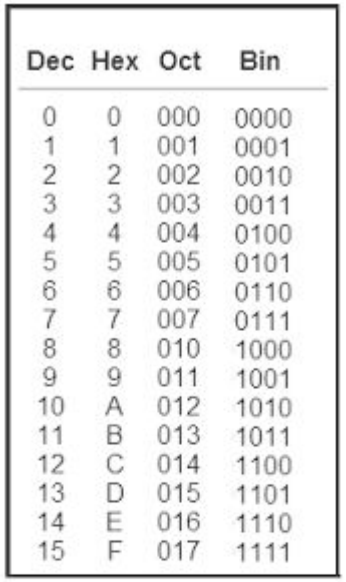
We convert each triplet to its single-digit octal equivalent.

From Binary to Octal

Positional (numeral) systems

- Binary
- Decimal
- Octal

$$\underbrace{100}\underbrace{111}\underbrace{011}\underbrace{101}\underbrace{111}$$



Dec	Hex	Oct	Bin
0	0	000	0000
1	1	001	0001
2	2	002	0010
3	3	003	0011
4	4	004	0100
5	5	005	0101
6	6	006	0110
7	7	007	0111
8	8	010	1000
9	9	011	1001
10	A	012	1010
11	B	013	1011
12	C	014	1100
13	D	015	1101
14	E	016	1110
15	F	017	1111

We convert each triplet to its single-digit octal equivalent.

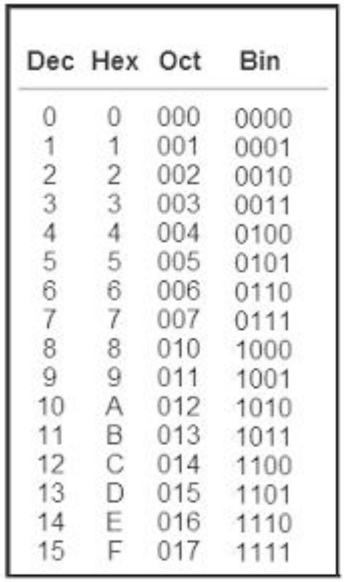
Universidad Francisco de Vitoria UFV Madrid

From Binary to Octal

Positional (numeral) systems

- Binary
- Decimal
- Octal

$$\underbrace{100}\underbrace{111}\underbrace{011}\underbrace{101}\underbrace{111}_{7}$$



Dec	Hex	Oct	Bin
0	0	000	0000
1	1	001	0001
2	2	002	0010
3	3	003	0011
4	4	004	0100
5	5	005	0101
6	6	006	0110
7	7	007	0111
8	8	010	1000
9	9	011	1001
10	A	012	1010
11	B	013	1011
12	C	014	1100
13	D	015	1101
14	E	016	1110
15	F	017	1111

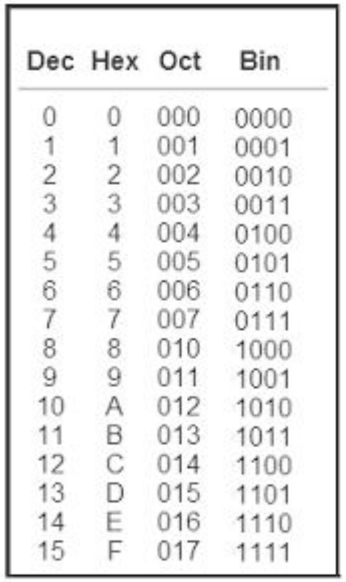
Universidad Francisco de Vitoria UFV Madrid

From Binary to Octal

Positional (numeral) systems

- Binary
- Decimal
- Octal

$$\underbrace{100}_{4}\underbrace{111}_{7}\underbrace{011}_{3}\underbrace{101}_{5}\underbrace{111}_{7}$$



Dec	Hex	Oct	Bin
0	0	000	0000
1	1	001	0001
2	2	002	0010
3	3	003	0011
4	4	004	0100
5	5	005	0101
6	6	006	0110
7	7	007	0111
8	8	010	1000
9	9	011	1001
10	A	012	1010
11	B	013	1011
12	C	014	1100
13	D	015	1101
14	E	016	1110
15	F	017	1111

Universidad
Francisco de Vitoria
UFV Madrid

From Binary to Hexadecimal

From Binary to Hexadecimal

Positional (numeral) systems

$$\begin{tikzcd}
\text{Binary} \arrow[dd, bend left] \\
\text{Decimal} \\
\text{Hexadecimal}
\end{tikzcd}$$

How can we number n_2 convert from a base 2 to base 16?

$$n_2 = 1001110111101111$$

Universidad Francisco de Vitoria UFV Madrid

From Binary to Hexadecimal

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

$$\underbrace{1001}\underbrace{1101}\underbrace{1110}\underbrace{1111}$$

We divide bits into groups of 4. Why?

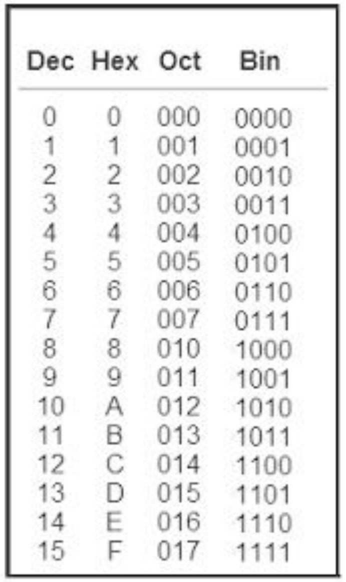
Universidad Francisco de Vitoria UFV Madrid

From Binary to Hexadecimal

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

$$\underbrace{1001}\underbrace{1101}\underbrace{1110}\underbrace{1111}$$



Dec	Hex	Oct	Bin
0	0	000	0000
1	1	001	0001
2	2	002	0010
3	3	003	0011
4	4	004	0100
5	5	005	0101
6	6	006	0110
7	7	007	0111
8	8	010	1000
9	9	011	1001
10	A	012	1010
11	B	013	1011
12	C	014	1100
13	D	015	1101
14	E	016	1110
15	F	017	1111

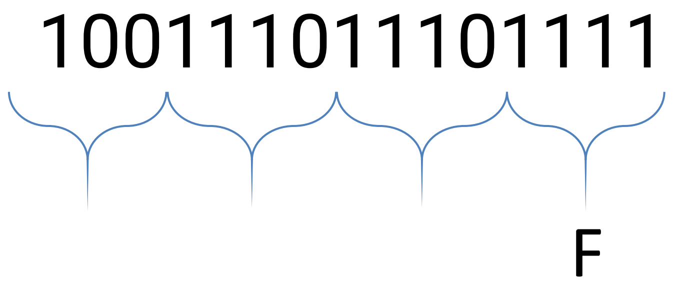
We divide bits into groups of 4. Why?

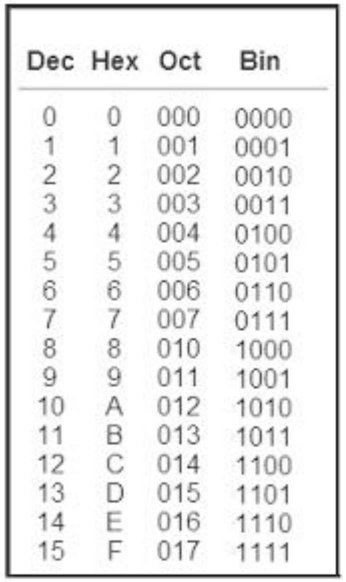
We need 4 bits in binary to represent 16 values in Hexadecimal.

From Binary to Hexadecimal

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal





Dec	Hex	Oct	Bin
0	0	000	0000
1	1	001	0001
2	2	002	0010
3	3	003	0011
4	4	004	0100
5	5	005	0101
6	6	006	0110
7	7	007	0111
8	8	010	1000
9	9	011	1001
10	A	012	1010
11	B	013	1011
12	C	014	1100
13	D	015	1101
14	E	016	1110
15	F	017	1111

We divide bits into groups of 4. Why?

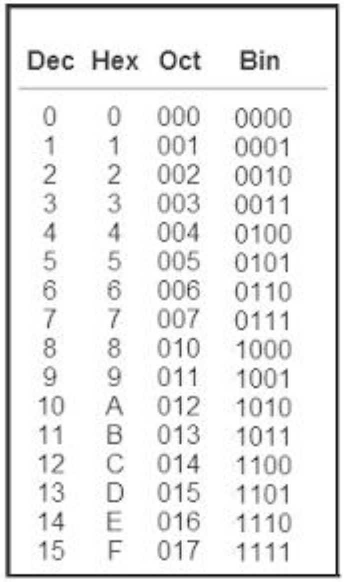
We need 4 bits in binary to represent 16 values in Hexadecimal.

From Binary to Hexadecimal

Positional (numeral) systems

$$\begin{tikzcd}
\bullet\ \text{Binary} \arrow[dd, bend left] \\
\bullet\ \text{Decimal} \\
\bullet\ \text{Hexadecimal}
\end{tikzcd}$$

$$\underbrace{100}_{4}\underbrace{1110}_{E}\underbrace{1110}_{E}\underbrace{1111}_{F}$$



Dec	Hex	Oct	Bin
0	0	000	0000
1	1	001	0001
2	2	002	0010
3	3	003	0011
4	4	004	0100
5	5	005	0101
6	6	006	0110
7	7	007	0111
8	8	010	1000
9	9	011	1001
10	A	012	1010
11	B	013	1011
12	C	014	1100
13	D	015	1101
14	E	016	1110
15	F	017	1111

We convert each group of four bits to a Hexadecimal.

Universidad Francisco de Vitoria
UFV Madrid

Universidad
Francisco de Vitoria
UFV Madrid

From Decimal to Binary

From Binary to Binary

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

$$\begin{tikzcd}\text{Decimal} \arrow[r, bend right, "\text{}"] & \text{Binary}\end{tikzcd}$$

How can we number n_1 convert from a base 10 to base 2?

$$n_1 = 233$$

Universidad Francisco de Vitoria UFV Madrid

From Binary to Binary

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

$$\begin{tikzcd}
\text{Binary} \\
\text{Decimal} \arrow[u, bend right]
\end{tikzcd}$$

base

233 / $\boxed{\mathbf{2}}$

Universidad Francisco de Vitoria UFV Madrid

From Binary to Binary

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

base

233 / $\boxed{2}$

$\boxed{1}$ 116

remainder

We divide by the **base** we want to transform to and store the **remainder**.

From Binary to Binary

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

233								/ 2
1	116							/ 2
	0	58						/ 2
		0	29					/ 2
			1	14				/ 2
				0	7			/ 2
					1	3		/ 2
						1	1	/ 2
							1	

From Binary to Binary

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

233								/ 2
1	116							/ 2
	0	58						/ 2
		0	29					/ 2
			1	14				/ 2
				0	7			/ 2
					1	3		/ 2
						1	1	/ 2
							1	
1	0	0	1	0	1	1	1	

10010111 is 233 in decimal base?

10010111 is 233 in decimal base?

NO, 10010111 is 151.

From Binary to Binary

Positional (numeral) systems

- Binary
- Decimal
- Hexadecimal

233								**/ 2**
1	**116**							**/ 2**
	0	**58**						**/ 2**
		0	**29**					**/ 2**
			1	**14**				**/ 2**
				0	**7**			**/ 2**
					1	**3**		**/ 2**
						1	**1**	**/ 2**
							1	
1	**0**	**0**	**1**	**0**	**1**	**1**	**1**	

We need to reverse the remainder.

Universidad
Francisco de Vitoria
UFV Madrid

Binary addition

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

	1	0	1	0	0
+	1	1	1	1	0

$\longleftarrow$

Binary addition goes from right to left.

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

```
     1   0   1   0   0
+    1   1   1   1   0
----------------------
                     0
```

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

$$\begin{array}{cccccc} & 1 & 0 & 1 & 0 & 0 \\ + & 1 & 1 & 1 & 1 & 0 \\ \hline & & & & 1 & 0 \end{array}$$

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

Universidad Francisco de Vitoria UFV Madrid

OFF-EXAM CONTENT

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

Carry

		1			
	1	0	1	0	0
+	1	1	1	1	0
			0	1	0

The carry is saved for the next digit.

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

Carry

		1			
	1	0	1	0	0
+	1	1	1	1	0
		1	0	1	0

The carry is saved for the next digit.

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

Carry

		1			
	1	0	1	0	0
+	1	1	1	1	0
		1	0	1	0

We must add the carry of the previous operation.

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

		Carry			
		1			
	1	0	1	0	0
+	1	1	1	1	0
		1 + 1	0	1	0

We must add the carry of the previous operation.

OFF-EXAM CONTENT

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

Carry

```
       1    1
       1    0    1    0    0
  +    1    1    1    1    0
  ----------------------------
            0    0    1    0
```

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

Carry

1	1	1			
	1	0	1	0	0
+	1	1	1	1	0
	0	0	0	1	0

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

Carry

1	1	1			
	1	0	1	0	0
+	1	1	1	1	0
	0 + 1	0	0	1	0

We must add the carry of the previous operation.

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

Carry

1	1	1			
	1	0	1	0	0
+	1	1	1	1	0
	1	0	0	1	0

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

1	1	1			
	1	0	1	0	0
+	1	1	1	1	0
1	1	0	0	1	0

The last carry is added if it is 1. This produce an increment of the number of bits.

OFF-EXAM CONTENT

Binary addition

The binary addition is similar to the decimal addition, but we use zeros and ones in this case.

0	+	0	0
1	+	0	1
0	+	1	1
1	+	1	0 and carry 1.

1 + 1 = 2, the number 2 in binary is represented by two bits = 10.

1	1	1				
	1	0	1	0	0	= 20
+	1	1	1	1	0	= 30
1	1	0	0	1	0	= 50

Universidad Francisco de Vitoria UFV Madrid

OFF-EXAM CONTENT

Positional (numeral) systems

Some important questions ….

How many values can be represented by n bits?

Positional (numeral) systems

Some important questions ….

How many values can be represented by n bits? 2^n

Positional (numeral) systems

Some important questions ….

How many values can be represented by n bits? 2^n

How many bits are needed to represent m values?

Positional (numeral) systems

Some important questions ….

How many values can be represented by n bits? 2^n

How many bits are needed to represent m values? $Log_2(n)$ by excess

$Log_2(91) = 6.50779 = 7$

Positional (numeral) systems

Some important questions ….

How many values can be represented by n bits? 2^n

How many bits are needed to represent m values? $Log_2(n)$ by excess
$Log_2(91) = 6.50779 = 7$

Universidad Francisco de Vitoria UFV Madrid

Positional (numeral) systems

Some important questions ….

How many values can be represented by n bits? 2^n

How many bits are needed to represent m values? $\mathrm{Log}_2(n)$ by excess
$\mathrm{Log}_2(91) = 6.50779 = 7$

If we use n bits, if the minimum representable value corresponds

to the number 0, what is the maximum representable numerical value?

Positional (numeral) systems

Some important questions ….

How many values can be represented by n bits? 2^n

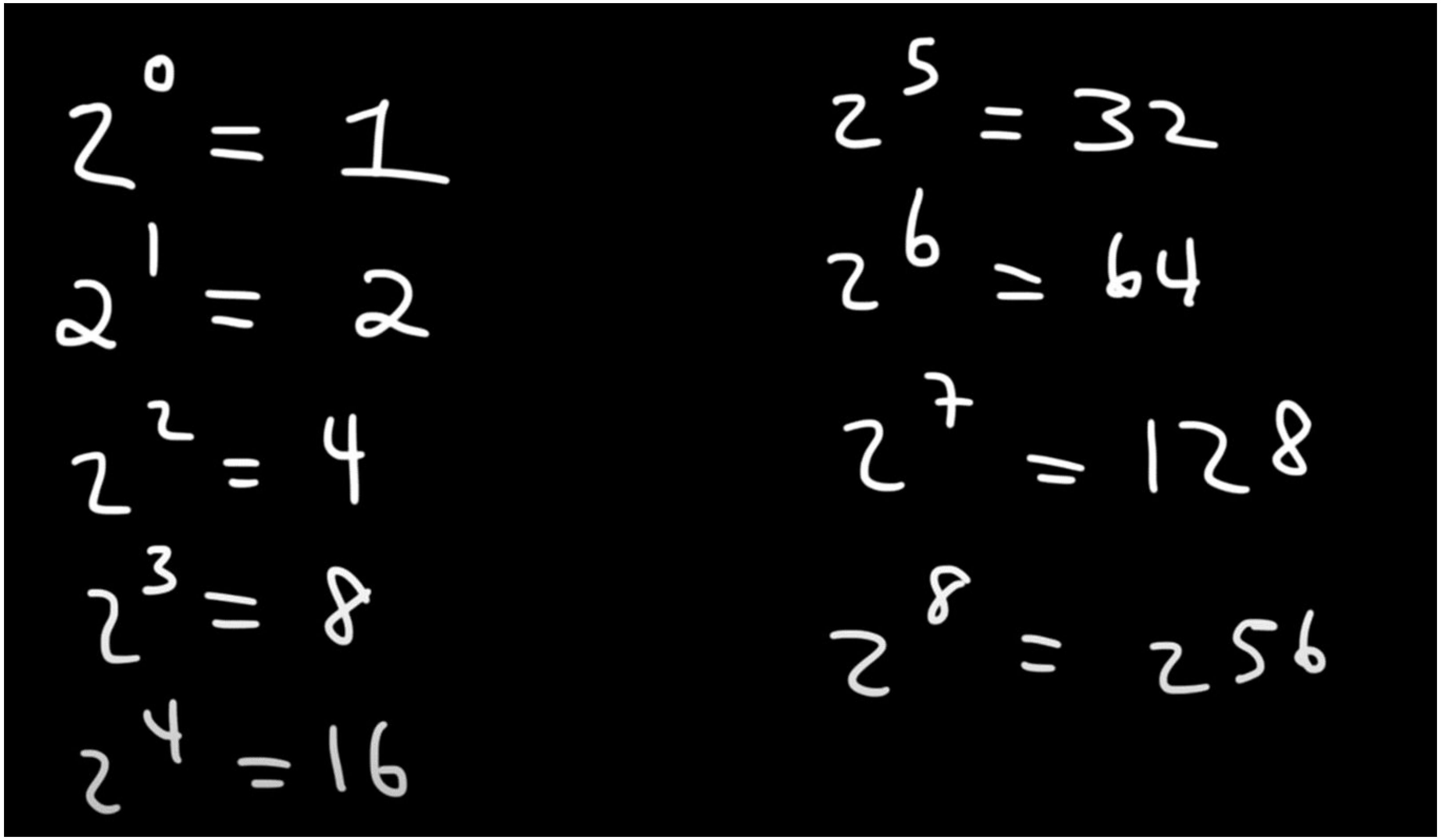
How many bits are needed to represent m values? $\text{Log}_2(n)$ by excess
$\text{Log}_2(91) = 6.50779 = 7$

If we use n bits, if the minimum representable value corresponds to the number 0, what is the maximum representable numerical value? $2^n - 1$

Universidad Francisco de Vitoria
UFV Madrid

Positional (numeral) systems

No calculator can be used in the exam.



$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$2^6 = 64$

$2^7 = 128$

$2^8 = 256$

Positional (numeral) systems

Decimal	Binary	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Universidad Francisco de Vitoria UFV Madrid

Number representation (Integers)

Number representation (Integers)

Integer numbers are represented in computers by a fixed number of bits. The representable range of number depends on the width and the representation convention (i.e. not all numbers can be represented, only those that are within the range).

Representable range of natural numbers (unsigned integers): $[0, 2^n-1]$, where n is the width.

Example: If we use a width n = 4, the natural numbers in the range [0, 15] can be represented.

Number representation (Integers)

Sign and magnitude convention

The decimal system defines the sign by adding a symbol to the magnitude to represent the sign of the number.

In binary, the sign is represented by a bit: `0 (+), 1 (-)`

$(+16)_{10}$ = 00010000 in binary with 8 bits

$(-16\ )_{10}$ = 10010000 in binary with 8 bits

Range of numbers representable with n bits: [-2n-1+1, 2n-1-1].

If we use 8 bits, we can be represented Integers numbers in the range [-127, +127].

Universidad Francisco de Vitoria UFV Madrid

Number representation (Integers)

Sign and magnitude convention

To calculate the integer value of a number with sign in binary, we must follow the next steps:

1. Convert the magnitude to base 10 using the n-1 less significant bits.
2. Add the sign: + (if it starts with 0) or - (if it starts with 1).

Example: 00111 = 7,   11010 = -10

- **Addition and subtraction operations are more complicated in binary, since the signs and the magnitudes must be taken into account separately.**
- **Zero has two representations [−0, +0].**

Number representation (Integers)

Sign and magnitude convention

Two's complement is a mathematical operation to reversibly convert a positive binary number into a negative binary number with equivalent (but negative) value, using the binary digit with the greatest place value to indicate whether the binary number is positive or negative (the sign).

$$C_b(N) = b^n - N$$

The total of positive numbers will be $2^{n-1}-1$ and the total of negatives will be 2^{n-1} where n is the **maximum number of bits**. The 0 would count separately.

$$\text{If we use 4 digits} \rightarrow C_{10}(0129) = 10^4 - 0129 = 9871$$

$$9871 + 0129 = 10000 = 10^4$$

Universidad Francisco de Vitoria UFV Madrid

Number representation (Integers)

From binary to one’s complement

One's complement is a mathematical operation to reversibly convert a positive binary number into a negative binary number with equivalent (but negative) value. Binary numbers are represented by the transpose of the binary number representation of their equivalent positive numbers.

1. Remove the sign and use the positive number
2. Convert the decimal number into a binary number.

Number representation (Integers)

From binary to one’s complement

Two's complement is a mathematical operation to reversibly convert a positive binary number into a negative binary number with equivalent (but negative) value, using the binary digit with the greatest place value to indicate whether the binary number is positive or negative (the sign).

1. Remove the sign and use the positive number
2. Convert the decimal number into a binary number (Positive representation).
3. Transpose all digits: zeros into ones and ones into zeros.
4. Add 1 if you want to get the negative representation.

Number representation (Integers)

Convert from binary to one’s complement

Convert 229 from decimal to binary?

Number representation (Integers)

Convert from binary to one’s complement

Convert 229 from decimal to binary?

How many bits I need to represent 229 in binary? **8 bits**

229 = 11100101

If we use 8 bits, we can be represented Integers numbers in the range [0, +255].

Universidad Francisco de Vitoria UFV Madrid

Number representation (Integers)

Convert from binary to one's complement

Convert 229 from decimal to binary?

How many bits I need to represent 229 in binary?	**8 bits**
How many bits I need to represent 229 in binary one's complement?	**9 bits**

229 = 11100101

If we use 9 bits, we can be represented Integers numbers in the range [-256, +255].

Number representation (Integers)

Convert from binary to one’s complement

Convert 229 from decimal to binary?

How many bits I need to represent 229 in binary?	**8 bits**
How many bits I need to represent 229 in binary one’s complement?	**9 bits**

$$229 = 11100101 \rightarrow 011100101 = C1_2(11100101)$$

Universidad Francisco de Vitoria UFV Madrid

If we use 9 bits, we can be represented Integers numbers in the range [-256, +255].

Number representation (Integers)

Convert from binary to one's complement

Convert 229 from decimal to binary?

How many bits I need to represent 229 in binary?	**8 bits**
How many bits I need to represent 229 in binary one's complement?	**9 bits**

$$229 = 11100101 \rightarrow 011100101 = C1_2(11100101)$$

$C1_2 \longrightarrow$ **0 1 1 1 0 0 1 0 1** $\longleftarrow$ Transpose all digits

$C_2 \longrightarrow$ **1 0 0 0 1 1 0 1 0**

If we use 9 bits, we can be represented Integers numbers in the range [-256, +255].

Number representation (Integers)

Convert from binary to one's complement

Convert 229 from decimal to binary?

How is -229 in binary two's complement?

$$
\begin{array}{lr l}
C1_2 \longrightarrow & \mathbf{0\,1\,1\,1\,0\,0\,1\,0\,1} & \longleftarrow \text{We must flip all bits.} \\
 & \mathbf{1\,0\,0\,0\,1\,1\,0\,1\,0} & \\
\mathbf{+} & \mathbf{1} & \longleftarrow \text{We add 1.} \\
\hline
 & \mathbf{1\,0\,0\,0\,1\,1\,0\,1\,1} & \longleftarrow \textbf{- 229 in two's complement}
\end{array}
$$

Number representation (Integers)

Convert from binary to one's complement

Decimal signed number	Positive binary	Negative binary
0	0000	0000
1	0001	1110
2	0010	1101
3	0011	1100
4	0100	1011
5	0101	1010
6	0110	1001
7	0111	1000
8		1111

If we use 4 bits, we can be represented Integers numbers in the range [-8, +7].

Universidad Francisco de Vitoria UFV Madrid

Number representation (Integers)

Convert from binary to one's complement

Convert 229 from decimal to binary?

How is -229 in binary two's complement?

$$\mathbf{C1_2} \longrightarrow \mathbf{0\;1\;1\;1\;0\;0\;1\;0\;\boxed{1}} \longleftarrow$$

We find the most significant bit. The first 1 starting on the right.

$$\mathbf{1\;0\;0\;0\;1\;1\;0\;1\;1} \longleftarrow$$

We must flip all bits after the most significant one.

- 229 in two's complement

Number representation (Integers)

Sign and magnitude convention (**Tips**)

n = 4 bits C_2 (1100) = 0100

n = 8 bits C_2 (11011100) = 00100100

n = 8 bits C_2 (11001010) = 00110110

Universidad Francisco de Vitoria **UFV** Madrid

Number representation (Integers)

Sign and magnitude convention (**Tips**)

n = 4 bits	C_2 (1100)	= 0100
n = 8 bits	C_2 (11011100)	= 00100100
n = 8 bits	C_2 (11001010)	= 00110110

$$\mathbf{N + C_b(N) = b^n}$$

n = 4 digits, N = 7500, $C_{10}(N) = 2500$

7500 + 2500 = 10000 with n = 4 digits, the result is 0000

N and $C_b(N)$ are opposites $\rightarrow C_b(N) \sim -N$

Universidad Francisco de Vitoria **UFV** Madrid

Number representation (Integers)

Sign and magnitude convention

- Positive numbers are represented as a magnitude and sign (therefore starting with 0).
- Negative numbers are represented as the 2's complement of the corresponding positive number (starting with 1).

Range of representable numbers using n bits: [-2^{n-1}, $2^{n-1}-1$]

If we use 8 bits, we can be represented Integers numbers in the range [-127, +127].

Standard positional representation of a number N in base b using **two's complement** is written as follow:

$$N = (a_{n-1}a_{n-2} \ldots a_1a_0\, a_{-1} \ldots a_{-m})_b$$

Universidad Francisco de Vitoria UFV Madrid

Number representation (Integers)

Convert -68 to binary in two’s complement

How many bits I need to represent -68 in binary?

Number representation (Integers)

Convert -68 to binary in two's complement

How many bits I need to represent -68 in binary? **8 bits**

I need 7 bits to represent 68 but I need another bit more to represent -68 to get a range between [-128 + 127].

Number representation (Integers)

Convert -68 to binary in two’s complement

How many bits I need to represent -68 in binary? **8 bits**

I need 7 bits to represent 68 but I need another bit more to represent -68 to get a range between [-128 + 127].

68							/	2
0	34						/	2
	0	17					/	2
		1	8				/	2
			0	4			/	2
				0	2		/	2
					0	1	/	2
							/	2
0	0	1	0	0	0	1	/	2

Universidad Francisco de Vitoria UFV Madrid

68 = 1000100

Number representation (Integers)

Convert -68 to binary in two’s complement

How many bits I need to represent -68 in binary? **8 bits**

I need 7 bits to represent 68 but I need another bit more to represent -68 to get a range between [-128 + 127].

68							/	2
0	34						/	2
	0	17					/	2
		1	8				/	2
			0	4			/	2
				0	2		/	2
					0	1	/	2
							/	2
0	0	1	0	0	0	1	/	2

68 = 01000100 $\leftarrow$ Add extra 0 to have 8 bits.

Universidad Francisco de Vitoria UFV Madrid

Number representation (Integers)

Convert -68 to binary in two’s complement

How many bits I need to represent -68 in binary? **8 bits**

I need 7 bits to represent 68 but I need another bit more to represent -68 to get a range between [-128 + 127].

$$68 = 01000\boxed{1}00$$

We find the most significant bit. The first 1 starting on the right.

$$68 = \boxed{01000}100$$

We flip all bits after the most significant bit.

$$-68 = 10111100$$

Universidad
Francisco de Vitoria
UFV Madrid

Number representation (Float)

Number representation (Float)

Real numbers are represented in computers approximately using IEEE-754 standard, using an integer with a fixed precision, called the significand, scaled by an integer exponent of a fixed base (scientific notation).

Example: If we want to represent 14.345 as floating point number in base 10:

$$14.345 = \underbrace{14345}_{\text{significad}} \times \underbrace{10}_{\text{base}}{}^{\overbrace{-3}^{\text{exponent}}}$$

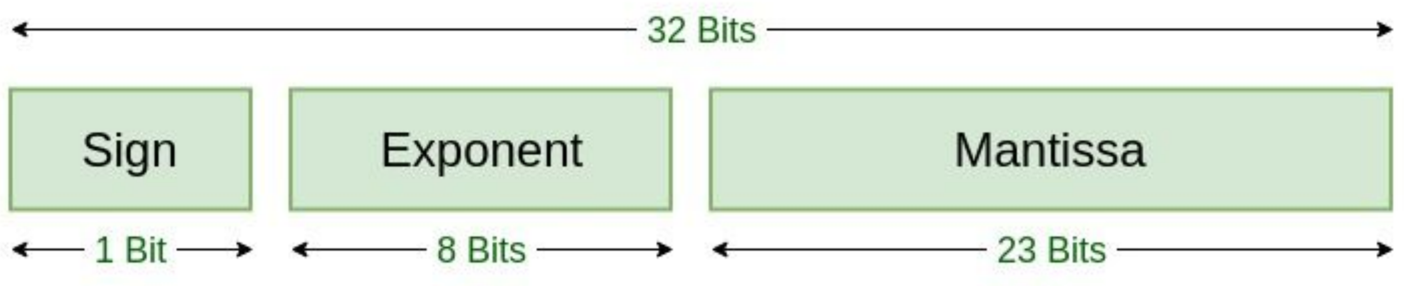
Number representation (Float)

Floating point numbers are represented in computers by a finite collection of bits composed of three parts:

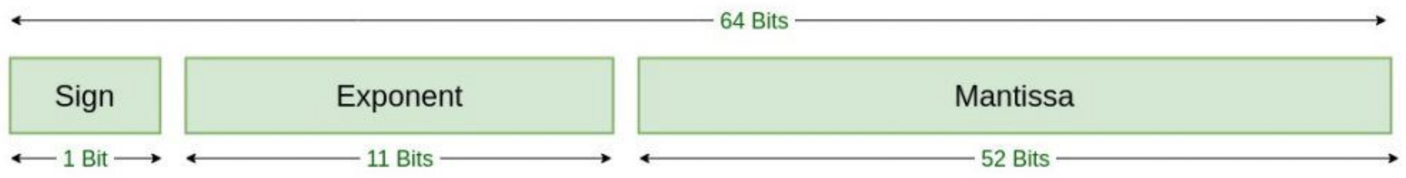
- Sign (1 single bit): The sign-bit is 1 if a number is negative and 0 if the number is negative, like integers.
- Mantissa or significand or fraction (23 bits in single precision floating point): The mantissa are the significant digits of the number in floating-point representation.
- Exponent (8 bits single precision floating point): The exponent is the radix is raised in determining the value of that floating-point representation.

Number representation (Float)

Float numbers are divided into two based on the above three components: single precision (32 bits) and double precision (64 bits).



single precision



double precision

Universidad Francisco de Vitoria UFV Madrid

Number representation (Float)

To convert decimal number into IEEE 754 Floating Point Representation:

1. Choose precision representation: single or double.
2. Separate the whole and the decimal part of the number.
3. Convert the decimal number into binary.
4. Convert the decimal portion into binary.
5. Combine the two parts of the number that have been converted into binary.
6. Identify the sign: 0 for positive numbers and 1 for negative numbers.

Number representation (Float)

To convert decimal number into IEEE 754 Floating Point Representation:

7. Convert the binary number into base 2 scientific notation.
    - To convert the number into base 2 scientific notation, we must move the decimal point over to the left until it is to the right of the first bit to create the normalized mantissa.
8. Compute the exponent based on precision.

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$$\begin{tikzcd} \boxed{\mathbf{88}} \arrow[dr] & \mathbf{.} & \boxed{\mathbf{125}} \arrow[r] & \text{Decimal portion} \\ & \text{Decimal number} & & \end{tikzcd}$$

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$\boxed{\mathbf{88}}\mathbf{.}\boxed{\mathbf{125}} \longrightarrow$ Decimal portion

$\longrightarrow$ Decimal number

88							/	2
0	44						/	2
	0	22					/	2
		0	11				/	2
			1	5			/	2
				1	2		/	2
					0	1	/	2
							/	2
0	0	0	1	1	0	1		

Universidad Francisco de Vitoria **UFV** Madrid

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$\boxed{\mathbf{88}}\mathbf{.}\boxed{\mathbf{125}} \rightarrow$ Decimal portion

$\hookrightarrow$ Decimal number

88							/	2
0	44						/	2
	0	22					/	2
		0	11				/	2
			1	5			/	2
				1	2		/	2
					0	1	/	2
							/	2
0	0	0	1	1	0	1		

Universidad Francisco de Vitoria UFV Madrid

88 = 1011000

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$\boxed{\mathbf{88}}.\boxed{\mathbf{125}} \rightarrow$ Decimal portion

$\hookrightarrow$ Decimal number

0.125				*	**2**
0	**0.25**			*	**2**
	0	**0.5**		*	**2**
		1	**1.0**	*	**2**

0.125 = 001

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

88.125

$\mathbf{1011000.001 \times 2^{0}}$

Move decimal signal 6 places to left to let one 1 there.

$\boxed{\mathbf{1.011000001}} \mathbf{\times 2^{0+6}}$

There are set **biases** for both single and double precision. The exponent bias for single precision is 127, which means we must add the base 2 exponent found previously to it. Thus, the exponent you will use is 127 + 6 which is 133.

127 + 6 = 133

Universidad
Francisco de Vitoria
UFV Madrid

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

127 + 6 = 133

133								/	2
1	66							/	2
	0	33						/	2
		1	16					/	2
			0	8				/	2
				0	4			/	2
					0	2	2	/	2
						0	1	/	2
1	0	1	0	0	0	0	1		

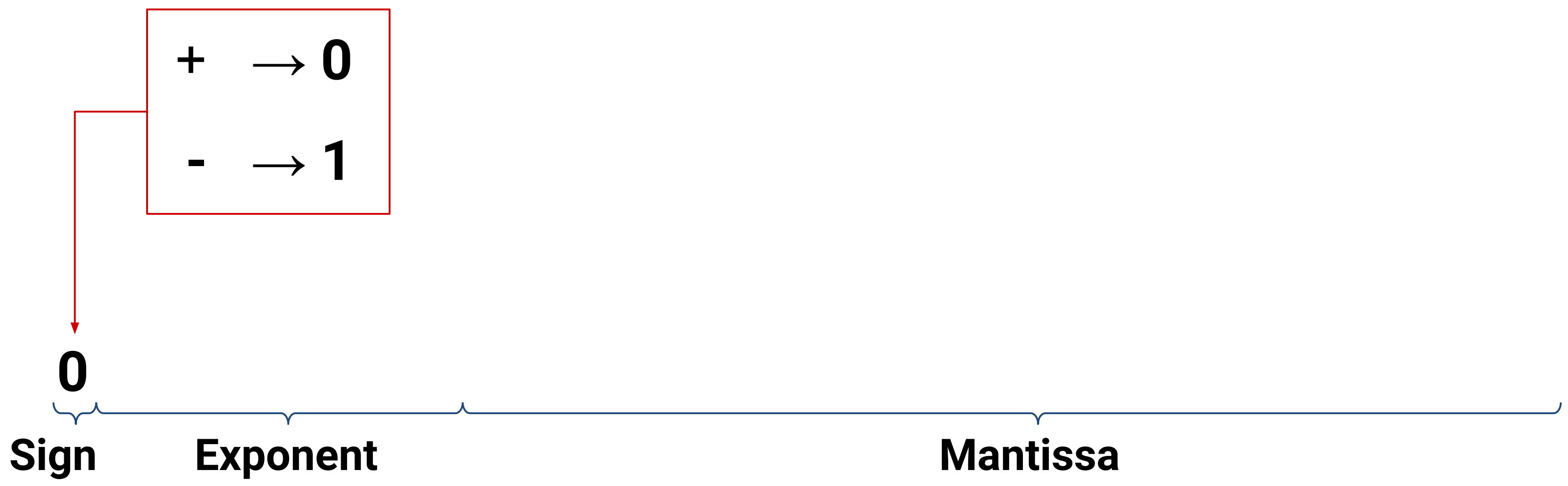
133 = 10000101

Universidad Francisco de Vitoria UFV Madrid

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**



Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$$\boxed{1\,0\,0\,0\,0\,1\,0\,1}$$

$$\downarrow$$

$$\underbrace{0}_{\textbf{Sign}}\;\underbrace{1\,0\,0\,0\,0\,1\,0\,1}_{\textbf{Exponent}}\;\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\textbf{Mantissa}}$$

Universidad Francisco de Vitoria UFV Madrid

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$$\boxed{1.011000001 \times 2^{6}}$$

$$\underbrace{0}_{\text{Sign}}\;\underbrace{1\;0\;0\;0\;0\;1\;0\;1}_{\text{Exponent}}\;\underbrace{0\;1\;1\;0\;0\;0\;0\;0\;1\qquad\qquad\qquad\qquad}_{\text{Mantissa}}$$

we just drop the 1 on the left and copy the decimal portion of the number that is being multiplied by 2.

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$$\boxed{1.011000001 \times 2^{6}}$$

$$\underbrace{0}_{\text{Sign}}\ \underbrace{1\ 0\ 0\ 0\ 0\ 1\ 0\ 1}_{\text{Exponent}}\ \underbrace{0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}_{\text{Mantissa}}$$

We complete with zeros.

Universidad Francisco de Vitoria UFV Madrid

Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use?

Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$$\begin{tikzcd} \boxed{\mathbf{25}}\mathbf{.}\boxed{\mathbf{025}} \arrow[r] \arrow[d] & \text{Decimal portion} \\ \text{Decimal number} & \end{tikzcd}$$

Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$\boxed{\textbf{25}}\textbf{.}\boxed{\textbf{025}} \rightarrow$ Decimal portion

$\hookrightarrow$ Decimal number

25					/	2
1	12				/	2
	0	6			/	2
		0	3		/	2
			1	1	/	2
1	0	0	1	1		

Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$\boxed{\mathbf{25}}.\boxed{\mathbf{025}} \rightarrow$ Decimal portion

$\rightarrow$ Decimal number

25					/	**2**
1	12				/	**2**
	0	6			/	**2**
		0	3		/	**2**
			1	1	/	**2**
1	0	0	1	1		

25 = 11001

Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$\boxed{25}.\boxed{025} \rightarrow$ Decimal portion

$\rightarrow$ Decimal number

0.025								*	**2**
0	**0.05**							*	**2**
	0	**0.1**						*	**2**
		0	**0.2**					*	**2**
			0	**0.4**				*	**2**
				1	**0.6**			*	**2**
					1	**0.2**		*	**2**
						0	**0.4**	*	**2**

Universidad
Francisco de Vitoria
UFV Madrid

Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$\boxed{\mathbf{25}}\mathbf{.}\boxed{\mathbf{025}} \rightarrow$ Decimal portion

$\hookrightarrow$ Decimal number

0.4									*	**2**
0	**0.8**								*	**2**
	1	**0.6**							*	**2**
		1	**0.2**						*	**2**
			0	**0.4**					*	**2**
				0	**0.8**				*	**2**
					1	**0.6**			*	**2**
						1	**0.2**		*	**2**
							0	**0.4**	*	**2**

Universidad Francisco de Vitoria **UFV** Madrid

Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$\boxed{\mathbf{25}}.\boxed{\mathbf{025}} \rightarrow$ Decimal portion

$\rightarrow$ Decimal number

0.4									*	2
0	0.8								*	2
	1	0.6							*	2
		1	0.2						*	2
			0	0.4					*	2
				0	0.8				*	2
					1	0.6			*	2
						1	0.2		*	2
							0	0.4	*	2

I can keep dividing to infinity, but I stop when I have 24 bits.

Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

25.025

$$11001.00000110011001100110 \times 2^{0}$$

Move decimal signal 4 places to left to let one 1 there.

$$\boxed{1.10010000011001100110011 0}\ \times 2^{0+4}$$

127 + 4 = 131

There are set **biases** for both single and double precision. The exponent bias for single precision is 127, which means we must add the base 2 exponent found previously to it. Thus, the exponent you will use is 127 + 4 which is 131.

Universidad Francisco de Vitoria UFV Madrid

Number representation (Float)

Convert 88.125 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

127 + 4 = 131

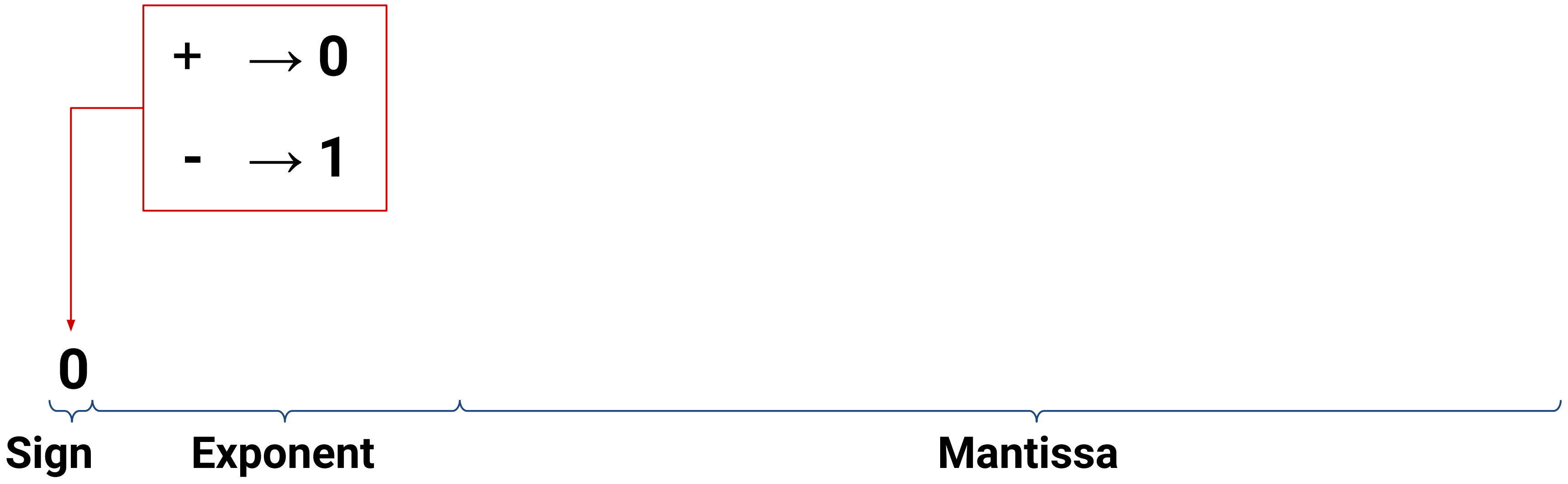
131								/	2
1	65							/	2
	1	32						/	2
		0	16					/	2
			0	8				/	2
				0	4			/	2
					0	2	2	/	2
						0	1	/	2
1	1	0	0	0	0	0	1		

131 = 10000011

Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

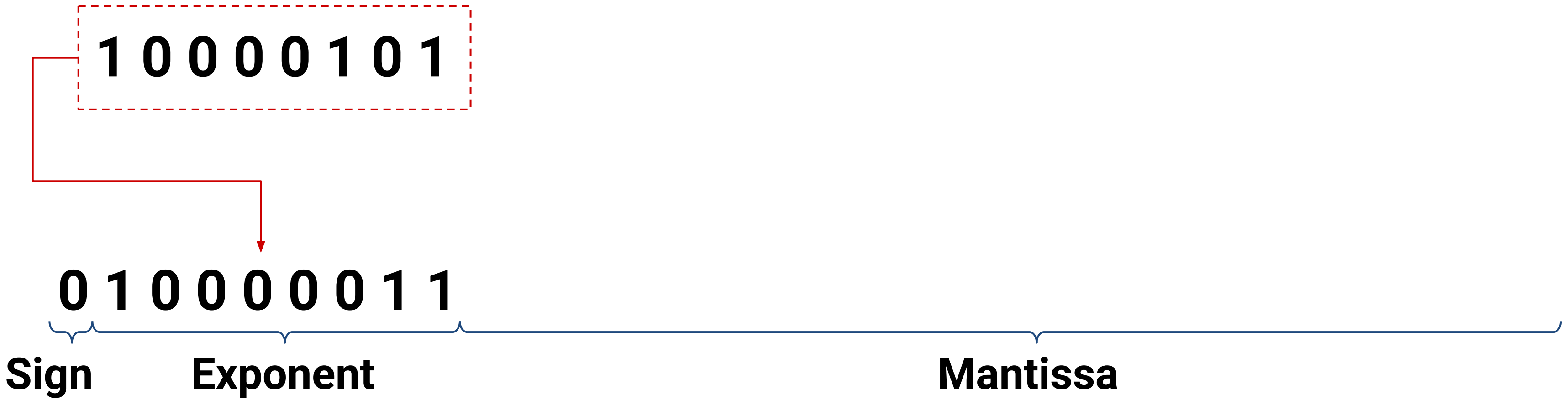
Which precision we must use? **single (32 bits)**



Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**



Number representation (Float)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

Which precision we must use? **single (32 bits)**

$$\boxed{1.10010000011001100110011001100110 \times 2^{6}}$$

$$\underbrace{0}_{\textbf{Sign}}\ \underbrace{1\,0\,0\,0\,0\,0\,1\,1}_{\textbf{Exponent}}\ \underbrace{1\,0\,0\,1\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1}_{\textbf{Mantissa}}$$

we just drop the 1 on the left and copy the decimal portion of the number that is being multiplied by 2.

Universidad Francisco de Vitoria UFV Madrid

Number representation (Float)

Convert the number represented in single IEEE 754 Floating Point to decimal?

11000010100010000000000000000000

Number representation (Float)

Convert the number represented in single IEEE 754 Floating Point to decimal?

11000010100010000000000000000000

$$\underbrace{1}_{\textbf{Sign}}\ \underbrace{1\ 0\ 0\ 0\ 0\ 1\ 0\ 1}_{\textbf{Exponent}}\ \underbrace{0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}_{\textbf{Mantissa}}$$

We split the number in its parts (sign, exponent and mantissa).

Number representation (Float)

Convert the number represented in single IEEE 754 Floating Point to decimal?

11000010100010000000000000000000

The number is negative.

$$\underbrace{\boxed{1}}_{\textbf{Sign}}\ \underbrace{1\ 0\ 0\ 0\ 0\ 1\ 0\ 1}_{\textbf{Exponent}}\ \underbrace{0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}_{\textbf{Mantissa}}$$

We identity the sign: 0 → positive and 1 → negative.

Number representation (Float)

Convert the number represented in single IEEE 754 Floating Point to decimal?

11000010100010000000000000000000

We transform the exponent to decimal to find the normalization of the mantissa.

$$\underbrace{1}_{\textbf{Sign}}\ \underbrace{\boxed{1\ 0\ 0\ 0\ 0\ 1\ 0\ 1}}_{\textbf{Exponent}}\ \underbrace{0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}_{\textbf{Mantissa}}$$

We calculate the exponent → 1 0 0 0 0 1 0 1 = 133

133 - 127 = 6

Number representation (Float)

Convert the number represented in single IEEE 754 Floating Point to decimal?

11000010100010000000000000000000

We compute the number denormalizing the mantissa and converting to decimal.

$$\underbrace{1}_{\textbf{Sign}}\ \underbrace{1\ 0\ 0\ 0\ 0\ 1\ 0\ 1}_{\textbf{Exponet}}\ \underbrace{\boxed{0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}}_{\textbf{Mantissa}}$$

1. 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

We move decimal signal 6 places to right.

You must always insert 1 on the left.

Universidad Francisco de Vitoria **UFV** Madrid

Number representation (Float)

Convert the number represented in single IEEE 754 Floating Point to decimal?

11000010100010000000000000000000

We compute the number denormalizing the mantissa and converting to decimal.

$$\underbrace{1}_{\textbf{Sign}}\ \underbrace{1\ 0\ 0\ 0\ 0\ 1\ 0\ 1}_{\textbf{Exponet}}\ \underbrace{0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}_{\textbf{Mantissa}}$$

$$\boxed{1\ 0\ 0\ 0\ 1\ 0\ 0}.\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$$

We convert to decimal.

$$\left.\begin{array}{l}\textbf{1000100 = 68}\\ \textbf{0000000000000000 = 0}\end{array}\right\}\ \textbf{- 68.0}$$

Universidad Francisco de Vitoria UFV Madrid

Universidad
Francisco de Vitoria
UFV Madrid

Encoding representation

Encoding representation

Decimal and alphanumeric encodings

An **encoding** is a **set of n-bit** strings over which a convention is established whereby each string represents a number or other type of information.

- Numeric encoding or code represents numerical information.
- Alphanumeric encoding or code represents numbers, letters and punctuation marks.
- Error encoding or code information so that certain errors in storing, retrieving or transmitting information can be detected and corrected.

Encoding representation

Binary Coded Decimal

Binary Coded Decimal (BCD) is used for the binary representation of numbers in decimal base. Each decimal digit is represented by a combination of 4 bits and each number as a string of digits.

734 = 0111 0011 0100

22 = 0010 0010

Decimal	Binay (BCD) 8 4 2 1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Encoding representation

Alphanumeric Codes

Alphanumeric codes are used to represent text where a code (bit string) is assigned to each character. Characters are usually grouped into 5 categories:

- Alphanumeric characters: A, B, C, ….Z, a, b, c, …., z
- Numeric characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Special characters: ( ) + - , & > < Ñ ñ # Ç ç SP …
- Geometric and graphical characters: | ⌘¶
- Control characters: enter, space, …

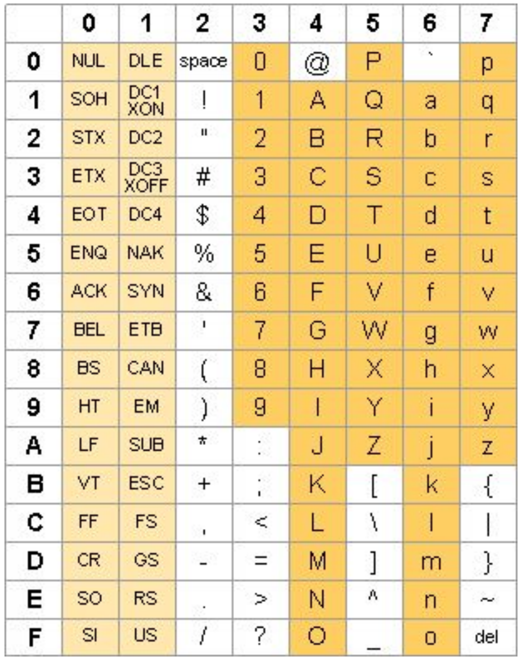
Encoding representation

Ascii Code

ASCII (American Standard Code for Information Interchange) code is one of the oldest (1968). It was created to represent the characters and symbols of the English language. **The basic ASCII code uses 7 bits (each character or symbol is represented by 7 bits).**

C/Vega,7

C	/	V	e	g	a	,	7
1000011	0101111	1010110	1100101	1100111	1100001	0101100	0110111



	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

It corresponds to the ANSI x 3.4 - 1968 or ISO 646 standardisation.

Universidad Francisco de Vitoria UFV Madrid

Encoding representation

Ascii Code

There are different extended versions of the ascii code using 8 bits.

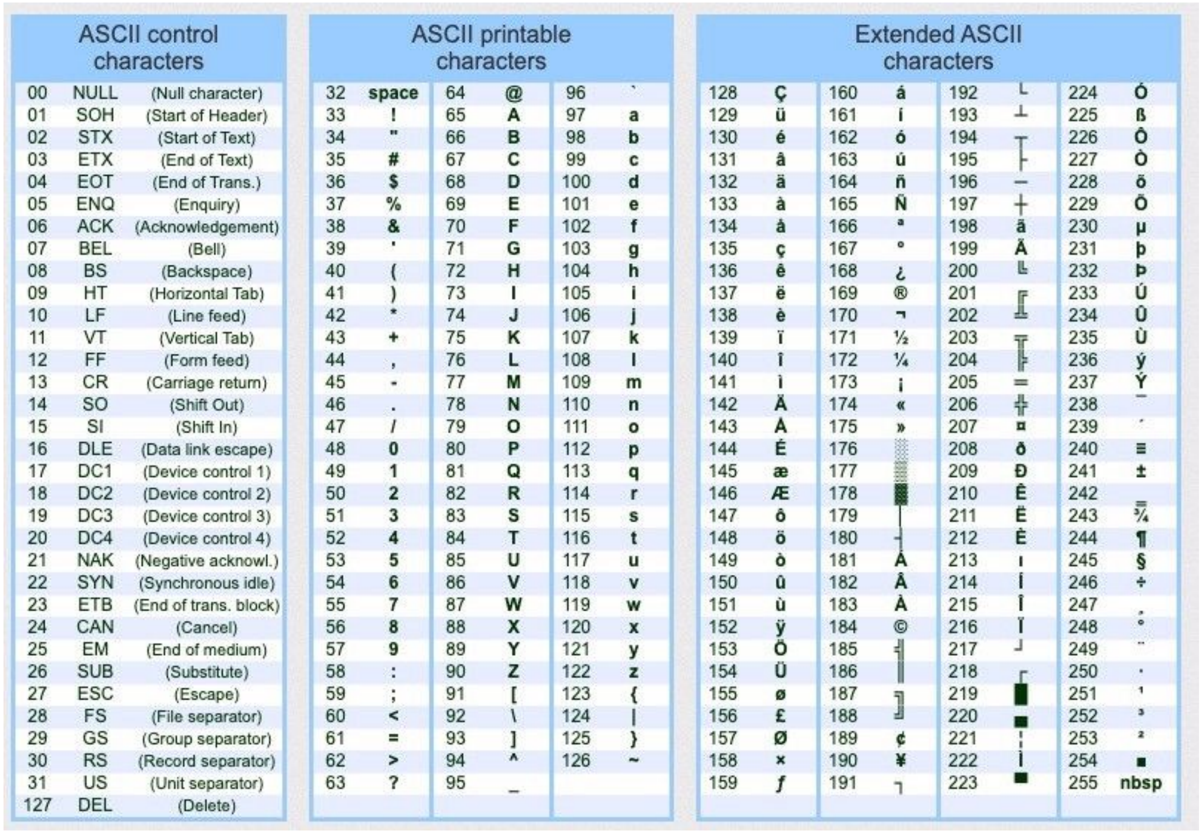
Name	ISO family	Geographical area
Latin-1	ISO 8859-1	Western and Eastern Europe
Latin-2	ISO 8859-2	Central and Eastern Europe
Latin-3	ISO 8859-3	Southern Europe, Maltese and Esperanto
Latin-4	ISO 8859-4	North europe
Latin/cyrillic	ISO 8859-5	Slavic languages
Latin/arabic	ISO 8859-6	Arabic languages
Latin/greek	ISO 8859-7	Modern greek

ISO/IEC 8859 is a joint ISO and IEC series of standards for 8-bit character encodings.

Encoding representation

Extended Ascii Code

Extended ASCII uses a eight-bit character encoding that includes (most of) the seven-bit ASCII characters, plus additional characters.



ASCII control characters			ASCII printable characters						Extended ASCII characters							
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	á	192	└	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ü	161	í	193	┴	225	ß
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	┬	226	Ô
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ú	195	├	227	Ò
04	EOT	(End of Trans.)	36	$	68	D	100	d	132	ä	164	ñ	196	─	228	õ
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	┼	229	Õ
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	å	166	ª	198	ã	230	µ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	Ã	231	þ
08	BS	(Backspace)	40	(	72	H	104	h	136	ê	168	¿	200	╚	232	Þ
09	HT	(Horizontal Tab)	41	)	73	I	105	i	137	ë	169	®	201	╔	233	Ú
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	¬	202	╩	234	Û
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	ï	171	½	203	╦	235	Ù
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	¼	204	╠	236	ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ì	173	¡	205	═	237	Ý
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ä	174	«	206	╬	238	¯
15	SI	(Shift In)	47	/	79	O	111	o	143	Å	175	»	207	¤	239	´
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	░	208	ð	240	≡
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	▒	209	Ð	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	▓	210	Ê	242	‗
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ô	179	│	211	Ë	243	¾
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ö	180	┤	212	È	244	¶
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	ò	181	Á	213	ı	245	§
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	û	182	Â	214	Í	246	÷
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ù	183	À	215	Î	247	¸
24	CAN	(Cancel)	56	8	88	X	120	x	152	ÿ	184	©	216	Ï	248	°
25	EM	(End of medium)	57	9	89	Y	121	y	153	Ö	185	╣	217	┘	249	¨
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ü	186	║	218	┌	250	·
27	ESC	(Escape)	59	;	91	[	123	{	155	ø	187	╗	219	█	251	¹
28	FS	(File separator)	60	<	92	\	124	|	156	£	188	╝	220	▄	252	³
29	GS	(Group separator)	61	=	93	]	125	}	157	Ø	189	¢	221	¦	253	²
30	RS	(Record separator)	62	>	94	^	126	~	158	×	190	¥	222	Ì	254	■
31	US	(Unit separator)	63	?	95	_			159	ƒ	191	┐	223	▀	255	nbsp
127	DEL	(Delete)														

Encoding representation

Unicode Code

The **Unicode standard** is an information standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems. This code was designed to follow this main properties:

- Universality: it covers most existing written languages.
- Uniqueness: Each symbol has a unique code.
- Uniformity: Each character is represented by 8 or 16 bits depending of the unicode version.

Encoding representation

Unicode Code

Codes are divided into 4 groups or zones, as shown in the table below.

Zone	Codes (HEX)	Symbols	Characters
A	0000 - 3FFF	Basic Latin (ASCII), Latin-1 and other Latin characters, Greek, Cyrillic, Armenian, Hebrew, Arabic, Syrian, Chinese, Japanese and Korean phonetic characters	8192
I	4000 - 9FFF	Chinese, Japanese and Korean ideograms	24576
O	A000 - DFFF	Not assigned	16384
R	E000 - FFFF	Local and user-specific characters	8192

Operations

Base change
https://youtu.be/5WtLFbriEEE

Integer numbers
https://youtu.be/B7SpmkW0lTs

Two’s complement
https://youtu.be/UTVuROxztuQ

IEEE (Floating point)
https://youtu.be/HcjXH9WGmAU

Some tips
https://youtu.be/5TlUWLxOWzU