DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Práctica 1: Representación de la información

Números enteros

Segundo Cuatrimestre 2020

Organización del Computador I
DC - UBA

8 bits...

¿Cuántas cosas puede contar esta máquina?

¿Cómo se representan los números?

Cada número se puede representar de **varias** maneras.

Por ejemplo: el vigésimo elemento de los números naturales sin el cero.

- Si se escribiese en la antigua Roma hubiese sido:

¿Cómo se representan los números?

Cada número se puede representar de **varias** maneras.

Por ejemplo: el vigésimo elemento de los números naturales sin el cero.

- Si se escribiese en la antigua Roma hubiese sido:

XX

- Si se escribiese acá mismo, sería:

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Cada número se puede representar de **varias** maneras.

Por ejemplo: el vigésimo elemento de los números naturales sin el cero.

- Si se escribiese en la antigua Roma hubiese sido:

XX

- Si se escribiese acá mismo, sería:

20

- Aunque... también podría haber sido:

¿Cómo se representan los números?

Cada número se puede representar de **varias** maneras.

Por ejemplo: el vigésimo elemento de los números naturales sin el cero.

- Si se escribiese en la antigua Roma hubiese sido:

$$\text{XX}$$

- Si se escribiese acá mismo, sería:

$$20$$

- Aunque... también podría haber sido:

$$10100$$

Para representar un número natural podemos usar:

En este caso, representan el número veintisiete.

¿Importa el orden de los porotos?

Para representar un número natural podemos usar:

En este caso, representan el número veintisiete.

¿Importa el orden de los porotos?

No

Para representar un número natural podemos usar también el conjunto:

$$B=\{0,1,2,3,4,5,6,7,8,9\}$$

Para representar un número natural podemos usar también el conjunto:

$$B=\{0,1,2,3,4,5,6,7,8,9\}$$

En este caso, para representar el número veintisiete:

27

Sistemas posicionales

Para representar un número natural podemos usar también el conjunto:

$$B=\{0,1,2,3,4,5,6,7,8,9\}$$

En este caso, para representar el número veintisiete:

27

¿Importa el orden de los símbolos?

Para representar un número natural podemos usar también el conjunto:

$$B=\{0,1,2,3,4,5,6,7,8,9\}$$

En este caso, para representar el número veintisiete:

27

¿Importa el orden de los símbolos?

Si

$$27 \neq 72$$

¿Qué es una base?

- El sistema *decimal* utiliza un conjunto de 10 símbolos para representar los números.

¿Qué es una base?

- El sistema *decimal* utiliza un conjunto de 10 símbolos para representar los números.
- Así, en cada numeral, la posición de cada símbolo está relacionada con una potencia de 10.

¿Qué es una base?

- El sistema *decimal* utiliza un conjunto de 10 símbolos para representar los números.
- Así, en cada numeral, la posición de cada símbolo está relacionada con una potencia de 10.

$$27 = 2 \cdot 10^1 + 7 \cdot 10^0$$

¿Qué es una base?

- El sistema *decimal* utiliza un conjunto de 10 símbolos para representar los números.
- Así, en cada numeral, la posición de cada símbolo está relacionada con una potencia de 10.

$$27 = 2 \cdot 10^1 + 7 \cdot 10^0$$

¿Y si en vez de diez símbolos tuviéramos otra cantidad?

¿Qué es una base?

- El sistema *decimal* utiliza un conjunto de 10 símbolos para representar los números.
- Así, en cada numeral, la posición de cada símbolo está relacionada con una potencia de 10.

$$27 = 2 \cdot 10^1 + 7 \cdot 10^0$$

¿Y si en vez de diez símbolos tuviéramos otra cantidad?

A esa cantidad la llamamos **base**

Representaciones e interpretaciones de números.

- ¿Qué números pueden representar las siguientes cadenas?
  10
  478
  2011
  ORGA1

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

- ¿Qué números pueden representar las siguientes cadenas?
  10
  478
  2011
  ORGA1
- Dada la representación de un número, ¿puedo saber qué base se está utilizando?

Bases.

- En base 2, usamos los símbolos 0 y 1 y escribimos los naturales: 0, 1, 10, 11, 100, 101, 110...
- En base 3, usamos los símbolos 0, 1 y 2 y escribimos los naturales: 0, 1, 2, 10, 11, 12, 20...
- ...y así...

Bases más comunes

Base	Símbolos usados
2 (binario)	0, 1
8 (octal)	0, 1, 2, 3, 4, 5, 6, 7
10 (decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16 (hexadecimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Bases.

- En base 2, usamos los símbolos 0 y 1 y escribimos los naturales: 0, 1, 10, 11, 100, 101, 110...
- En base 3, usamos los símbolos 0, 1 y 2 y escribimos los naturales: 0, 1, 2, 10, 11, 12, 20...
- ...y así...

Bases más comunes

Base	Símbolos usados
2 (binario)	0, 1
8 (octal)	0, 1, 2, 3, 4, 5, 6, 7
10 (decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16 (hexadecimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

¿Es posible cambiar un número de una base a otra?

Bases.

- En base 2, usamos los símbolos 0 y 1 y escribimos los naturales: 0, 1, 10, 11, 100, 101, 110...
- En base 3, usamos los símbolos 0, 1 y 2 y escribimos los naturales: 0, 1, 2, 10, 11, 12, 20...
- ...y así...

Bases más comunes

Base	Símbolos usados
2 (binario)	0, 1
8 (octal)	0, 1, 2, 3, 4, 5, 6, 7
10 (decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16 (hexadecimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

¿Es posible cambiar un número de una base a otra?

¿Cómo escribirían un programa que lo haga?

Teorema:
Sean $a, b \in \mathbb{Z}$ con $b \neq 0$.

Existen $q, r \in \mathbb{Z}$ con $0 \leq r < |b|$

tales que $a = b \times q + r$

Además, q y r son únicos *(de a pares)*.

Teorema:
Sean $a, b \in \mathbb{Z}$ con $b \neq 0$.

Existen $q, r \in \mathbb{Z}$ con $0 \leq r < |b|$

tales que $a = b \times q + r$

Además, q y r son únicos *(de a pares)*.

¿Cómo lo usamos ?

Cambios de bases: Teorema de la división.

Teorema:
Sean $a, b \in \mathbb{Z}$ con $b \neq 0$.

Existen $q, r \in \mathbb{Z}$ con $0 \leq r < |b|$

tales que $a = b \times q + r$

Además, q y r son únicos *(de a pares)*.

¿Cómo lo usamos ?

$a = b \times q + r$

Teorema:
Sean $a, b \in \mathbb{Z}$ con $b \neq 0$.

Existen $q, r \in \mathbb{Z}$ con $0 \leq r < |b|$

tales que $a = b \times q + r$

Además, q y r son únicos *(de a pares)*.

¿Cómo lo usamos ?

$a = b \times q + r$

$a = (b_1 \times q_1 + r_1) \times q + r$

Teorema:
Sean $a, b \in \mathbb{Z}$ con $b \neq 0$.

Existen $q, r \in \mathbb{Z}$ con $0 \leq r < |b|$

tales que $a = b \times q + r$

Además, q y r son únicos *(de a pares)*.

¿Cómo lo usamos ?

$a = b \times q + r$

$a = (b_1 \times q_1 + r_1) \times q + r$

$a = [(b_2 \times q_2 + r_2) \times q_1 + r_1] \times q + r$

Teorema:
Sean $a, b \in \mathbb{Z}$ con $b \neq 0$.

Existen $q, r \in \mathbb{Z}$ con $0 \leq r < |b|$

tales que $a = b \times q + r$

Además, q y r son únicos *(de a pares)*.

¿Cómo lo usamos ?

$a = b \times q + r$

$a = (b_1 \times q_1 + r_1) \times q + r$

$a = [(b_2 \times q_2 + r_2) \times q_1 + r_1] \times q + r$

Si definimos $q = q1 = ... = q_N$, podemos continuar con la expansión hasta que $b_N < q_N$

Teorema:
Sean $a, b \in \mathbb{Z}$ con $b \neq 0$.

Existen $q, r \in \mathbb{Z}$ con $0 \leq r < |b|$

tales que $a = b \times q + r$

Además, q y r son únicos *(de a pares)*.

¿Cómo lo usamos ?

$a = b \times q + r$

$a = (b_1 \times q_1 + r_1) \times q + r$

$a = [(b_2 \times q_2 + r_2) \times q_1 + r_1] \times q + r$

Si definimos $q = q1 = ... = q_N$, podemos continuar con la expansión hasta que $b_N < q_N$

$a = \{[(b_N \times q_N + r_N) \times q_{N-1}) + r_{N-1}] \times ...\} \times q + r \times 1$

Cambios de bases: Teorema de la división.

Teorema:
Sean $a, b \in \mathbb{Z}$ con $b \neq 0$.

Existen $q, r \in \mathbb{Z}$ con $0 \leq r < |b|$

tales que $a = b \times q + r$

Además, q y r son únicos *(de a pares)*.

¿Cómo lo usamos ?

$a = b \times q + r$

$a = (b_1 \times q_1 + r_1) \times q + r$

$a = [(b_2 \times q_2 + r_2) \times q_1 + r_1] \times q + r$

Si definimos $q = q1 = ... = q_N$, podemos continuar con la expansión hasta que $b_N < q_N$

$a = \{[(b_N \times q_N + r_N) \times q_{N-1}) + r_{N-1}] \times ...\} \times q + r \times 1$

Si distribuímos:

Cambios de bases: Teorema de la división.

Teorema:
Sean $a, b \in \mathbb{Z}$ con $b \neq 0$.

Existen $q, r \in \mathbb{Z}$ con $0 \leq r < |b|$

tales que $a = b \times q + r$

Además, q y r son únicos *(de a pares)*.

¿Cómo lo usamos ?

$a = b \times q + r$

$a = (b_1 \times q_1 + r_1) \times q + r$

$a = [(b_2 \times q_2 + r_2) \times q_1 + r_1] \times q + r$

Si definimos $q = q1 = ... = q_N$, podemos continuar con la expansión hasta que $b_N < q_N$

$a = \{[(b_N \times q_N + r_N) \times q_{N-1}) + r_{N-1}] \times ...\} \times q + r \times 1$

Si distribuímos:

$$a = b_N \times q^{N+1} + r_N \times q^N + ... + r_1 \times q + r \times q^0$$

Ejemplo:

$$27 = 2 \cdot 10^1 + 7 \cdot 10^0 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

Ejercitación
Escribir los siguientes números en binario, octal y hexadecimal.

- diez
- quinientos doce

Sumando...

En decimal	En base 3
8 4 5	2 1 2
+ 3 4 2	+ 1 0 1
1 1 8 7	1 0 2 0

Multiplicando...

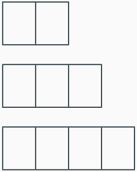
En base 3

$$
\begin{array}{r}
1\;2 \\
\times\; 2\;1 \\
\hline
1\;2 \\
1\;0\;1\;- \\
\hline
1\;0\;2\;2
\end{array}
$$

Tiras de símbolos

Si sólo podemos escribir tiras de símbolos de longitud fija:

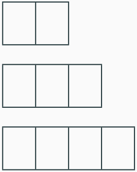
- ¿cuántos números podemos representar?
- ¿de qué depende?



Tiras de símbolos

Si sólo podemos escribir tiras de símbolos de longitud fija:

- ¿cuántos números podemos representar?
- ¿de qué depende?



Al operar con precisión fija, podemos tener **overflow**

Ocurre overflow cuando el resultado de una operación necesita una tira de símbolos más grande que la disponible

Hasta aca: números naturales

Por ahora sabemos...

- Leer naturales
- Escribir naturales
- Cambiarlos de base
- Operar con naturales (sumarlos, multiplicarlos)

¿Y qué hacemos con los enteros?

Codificando números enteros en base binaria

- **Sin signo**

Codificando números enteros en base binaria

- **Sin signo**
  solo sirve para positivos.

- **Sin signo**
  solo sirve para positivos.
- **Signo + Magnitud**

Codificando números enteros en base binaria

- **Sin signo**
  solo sirve para positivos.
- **Signo + Magnitud**
  se usa un bit para indicar el signo
- **Exceso** m

Codificando números enteros en base binaria

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

- **Sin signo**
  solo sirve para positivos.
- **Signo + Magnitud**
  se usa un bit para indicar el signo
- **Exceso** m
  Representa n como $m + n$.
  De esta manera, estamos "moviendo" el cero desde 0 hacia m.
  Luego los menores serán los negativos

- **Complemento a 2**

Codificando números enteros en base binaria

- **Complemento a 2**

  Los positivos se representan igual.

- **Complemento a 2**

  Los positivos se representan igual.
  Un número negativo n como $2^k + n$.
  Significa lo que le falta al número para llegar (el complemento) a 2^k.

- **Complemento a 2**

  Los positivos se representan igual.

  Un número negativo n como $2^k + n$.

  Significa lo que le falta al número para llegar (el complemento) a 2^k.

  Ejemplo:

  Quiero representar el número -2_{10} con $k = 3$ dígitos binarios.

- **Complemento a 2**

  Los positivos se representan igual.
  Un número negativo n como $2^k + n$.
  Significa lo que le falta al número para llegar (el complemento) a 2^k.
  Ejemplo:
  Quiero representar el número -2_{10} con $k = 3$ dígitos binarios.
  Entonces, $2^k \rightarrow 2^3 = 8_{10} = 1000_2$

Codificando números enteros en base binaria

- **Complemento a 2**

  Los positivos se representan igual.

  Un número negativo n como $2^k + n$.

  Significa lo que le falta al número para llegar (el complemento) a 2^k.

  Ejemplo:

  Quiero representar el número -2_{10} con $k = 3$ dígitos binarios.

  Entonces, $2^k \to 2^3 = 8_{10} = 1000_2$

  Luego $2 + 2_c = 1000_2$, donde 2_c es el complemento a 2 del número.

- **Complemento a 2**

  Los positivos se representan igual.

  Un número negativo n como $2^k + n$.

  Significa lo que le falta al número para llegar (el complemento) a 2^k.

  Ejemplo:

  Quiero representar el número -2_{10} con $k = 3$ dígitos binarios.

  Entonces, $2^k \rightarrow 2^3 = 8_{10} = 1000_2$

  Luego $2 + 2_c = 1000_2$, donde 2_c es el complemento a 2 del número.

  En este caso tendríamos:

  $010 + 2c = 1000 \rightarrow 1000 - 010 = 2c$.

- **Complemento a 2**

  Los positivos se representan igual.

  Un número negativo n como $2^k + n$.

  Significa lo que le falta al número para llegar (el complemento) a 2^k.

  Ejemplo:

  Quiero representar el número -2_{10} con $k = 3$ dígitos binarios.

  Entonces, $2^k \rightarrow 2^3 = 8_{10} = 1000_2$

  Luego $2 + 2_c = 1000_2$, donde 2_c es el complemento a 2 del número.

  En este caso tendríamos:

  $010 + 2c = 1000 \rightarrow 1000 - 010 = 2c$.

  Finalmente $2c = 110_2$

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	?		
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011		
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	?	
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	?
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2			
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	?		
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010		
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	?	
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	?
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8			

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	?		

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	OVERFLOW		

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	OVERFLOW	?	

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	OVERFLOW	1000	

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	OVERFLOW	1000	?

Codificando...

En base 2, numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	OVERFLOW
-2	1010	1110	1101
-8	OVERFLOW	1000	0111

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	?		
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011		
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	?	
-2			
-8			

Codificando...(con más bits)

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	?
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2			
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	?		
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010		
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	?	
-8			

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	
-8			

Codificando...(con más bits)

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	?
-8			

Codificando...(con más bits)

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8			

Codificando...(con más bits)

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	?		

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000		

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	?	

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	?

Codificando...(con más bits)

En base 2, numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	0000 0111

Codificando...(con más bits)

Similitudes entre 4 y 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	0000 0111

Codificando...(con más bits)

Similitudes entre 4 y 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	0000 0111

Extendiendo la cantidad de bits de precisión:

- Signo + Magnitud: Se extiende con 0's, pero el bit más significativo se mantiene indicando el signo.
- Complemento a 2: Se extiende con el valor del bit más significativo.
- Exceso a m: Se extiende siempre con 0's.

Resumiendo: Enteros como numerales binarios

Solo sirve para los positivos.

numeral $\rightarrow$ número que representa

$1111 \rightarrow 15_{10}$	$0111 \rightarrow 7_{10}$
$1110 \rightarrow 14_{10}$	$0110 \rightarrow 6_{10}$
$1101 \rightarrow 13_{10}$	$0101 \rightarrow 5_{10}$
$1100 \rightarrow 12_{10}$	$0100 \rightarrow 4_{10}$
$1011 \rightarrow 11_{10}$	$0011 \rightarrow 3_{10}$
$1010 \rightarrow 10_{10}$	$0010 \rightarrow 2_{10}$
$1001 \rightarrow 9_{10}$	$0001 \rightarrow 1_{10}$
$1000 \rightarrow 8_{10}$	$0000 \rightarrow 0_{10}$

Para los numerales de 4 *bits*.

Resumiendo: Enteros como numerales binarios

El primer bit es el signo, los demás son el *significado* (la magnitud del número en valor absoluto).

numeral $\rightarrow$ número que representa

$1111 \rightarrow -7_{10}$	$0111 \rightarrow 7_{10}$
$1110 \rightarrow -6_{10}$	$0110 \rightarrow 6_{10}$
$1101 \rightarrow -5_{10}$	$0101 \rightarrow 5_{10}$
$1100 \rightarrow -4_{10}$	$0100 \rightarrow 4_{10}$
$1011 \rightarrow -3_{10}$	$0011 \rightarrow 3_{10}$
$1010 \rightarrow -2_{10}$	$0010 \rightarrow 2_{10}$
$1001 \rightarrow -1_{10}$	$0001 \rightarrow 1_{10}$
$1000 \rightarrow -0_{10}$	$0000 \rightarrow 0_{10}$

Para los numerales de 4 *bits*.

Resumiendo: Enteros como numerales binarios

Los numerales que representa positivos son iguales a los anteriores

Para los negativos, dado un n negativo se representan escribiendo

$2^k + n$ en notación sin signo

cuentas $\rightarrow$ numeral $\rightarrow$ número que representa

$2^4 + (-1) = 15 \rightarrow$	$1111 \rightarrow -1_{10}$	$0111 \rightarrow 7_{10}$
$2^4 + (-2) = 14 \rightarrow$	$1110 \rightarrow -2_{10}$	$0110 \rightarrow 6_{10}$
$2^4 + (-3) = 13 \rightarrow$	$1101 \rightarrow -3_{10}$	$0101 \rightarrow 5_{10}$
$2^4 + (-4) = 12 \rightarrow$	$1100 \rightarrow -4_{10}$	$0100 \rightarrow 4_{10}$
$2^4 + (-5) = 11 \rightarrow$	$1011 \rightarrow -5_{10}$	$0011 \rightarrow 3_{10}$
$2^4 + (-6) = 10 \rightarrow$	$1010 \rightarrow -6_{10}$	$0010 \rightarrow 2_{10}$
$2^4 + (-7) = 9 \rightarrow$	$1001 \rightarrow -7_{10}$	$0001 \rightarrow 1_{10}$
$2^4 + (-8) = 8 \rightarrow$	$1000 \rightarrow -8_{10}$	$0000 \rightarrow 0_{10}$

Para los numerales de 4 *bits*.

Resumiendo: Enteros como numerales binarios

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

El número n se representa como $m + n$

cuentas $\rightarrow$ numeral $\rightarrow$ número que representa

$5 + (10) = 15$	$\rightarrow 1111 \rightarrow 10_{10}$		$5 + (2) = 7$	$\rightarrow 0111 \rightarrow 2_{10}$
$5 + (9) = 14$	$\rightarrow 1110 \rightarrow 9_{10}$		$5 + (1) = 6$	$\rightarrow 0110 \rightarrow 1_{10}$
$5 + (8) = 13$	$\rightarrow 1101 \rightarrow 8_{10}$		$5 + (0) = 5$	$\rightarrow 0101 \rightarrow 0_{10}$
$5 + (7) = 12$	$\rightarrow 1100 \rightarrow 7_{10}$		$5 + (-1) = 4$	$\rightarrow 0100 \rightarrow -1_{10}$
$5 + (6) = 11$	$\rightarrow 1011 \rightarrow 6_{10}$		$5 + (-2) = 3$	$\rightarrow 0011 \rightarrow -2_{10}$
$5 + (5) = 10$	$\rightarrow 1010 \rightarrow 5_{10}$		$5 + (-3) = 2$	$\rightarrow 0010 \rightarrow -3_{10}$
$5 + (4) = 9$	$\rightarrow 1001 \rightarrow 4_{10}$		$5 + (-4) = 1$	$\rightarrow 0001 \rightarrow -4_{10}$
$5 + (3) = 8$	$\rightarrow 1000 \rightarrow 3_{10}$		$5 + (-5) = 0$	$\rightarrow 0000 \rightarrow -5_{10}$

Para los numerales de 4 *bits* en exceso 5.

Para hacer en una noche de furia y juegos

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$28\ /\ 2 \quad = 14 \quad \text{Resto} = 0$

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$28 \;/\; 2 \quad = 14 \quad \text{Resto} = 0$

$14 \;/\; 2 \quad = 7 \quad \text{Resto} = 0$

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$28 \ / \ 2 \quad = 14 \quad \text{Resto} = 0$$
$$14 \ / \ 2 \quad = 7 \quad \text{Resto} = 0$$
$$7 \ / \ 2 \quad = 3 \quad \text{Resto} = 1$$

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$
\begin{array}{lll}
28\ /\ 2 & = 14 & \text{Resto} = 0 \\
14\ /\ 2 & = 7 & \text{Resto} = 0 \\
7\ /\ 2 & = 3 & \text{Resto} = 1 \\
3\ /\ 2 & = 1 & \text{Resto} = 1
\end{array}
$$

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$
\begin{array}{llll}
28 \;/\; 2 & = 14 & \text{Resto} = 0 \\
14 \;/\; 2 & = 7 & \text{Resto} = 0 \\
7 \;/\; 2 & = 3 & \text{Resto} = 1 \\
3 \;/\; 2 & = 1 & \text{Resto} = 1
\end{array}
$$

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$28\ /\ 2 = 14 \quad \text{Resto} = 0$$
$$14\ /\ 2 = 7 \quad \text{Resto} = 0$$
$$7\ /\ 2 = 3 \quad \text{Resto} = 1$$
$$3\ /\ 2 = 1 \quad \text{Resto} = 1$$

Quedando así el número 11100.

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$
\begin{aligned}
28\ /\ 2 &= 14 \quad \text{Resto} = 0\\
14\ /\ 2 &= 7 \quad \text{Resto} = 0\\
7\ /\ 2 &= 3 \quad \text{Resto} = 1\\
3\ /\ 2 &= 1 \quad \text{Resto} = 1
\end{aligned}
$$

Quedando así el número 11100.

Otra aproximación.

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$28\ /\ 2 = 14 \quad \text{Resto} = 0$$
$$14\ /\ 2 = 7 \quad \text{Resto} = 0$$
$$7\ /\ 2 = 3 \quad \text{Resto} = 1$$
$$3\ /\ 2 = 1 \quad \text{Resto} = 1$$

Quedando así el número 11100.

Otra aproximación.

Busco $2^x \geq 28$

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$\begin{aligned} 28 \ / \ 2 &= 14 & \text{Resto} &= 0 \\ 14 \ / \ 2 &= 7 & \text{Resto} &= 0 \\ 7 \ / \ 2 &= 3 & \text{Resto} &= 1 \\ 3 \ / \ 2 &= 1 & \text{Resto} &= 1 \end{aligned}$$

Quedando así el número 11100.

Otra aproximación.

Busco $2^x \geq 28 \rightarrow 5$, ya que $2^5 = 32$. Necesitaré 5 dígitos

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$\begin{array}{llll} 28 \;/\; 2 & = 14 & \text{Resto} & = 0 \\ 14 \;/\; 2 & = 7 & \text{Resto} & = 0 \\ 7 \;/\; 2 & = 3 & \text{Resto} & = 1 \\ 3 \;/\; 2 & = 1 & \text{Resto} & = 1 \end{array}$$

Quedando así el número 11100.

Otra aproximación.

Busco $2^x \geq 28 \rightarrow 5$, ya que $2^5 = 32$. Necesitaré 5 dígitos

Sabemos:

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$\begin{aligned} 28 \ / \ 2 &= 14 & \text{Resto} &= 0 \\ 14 \ / \ 2 &= 7 & \text{Resto} &= 0 \\ 7 \ / \ 2 &= 3 & \text{Resto} &= 1 \\ 3 \ / \ 2 &= 1 & \text{Resto} &= 1 \end{aligned}$$

Quedando así el número 11100.

Otra aproximación.

Busco $2^x \geq 28 \rightarrow 5$, ya que $2^5 = 32$. Necesitaré 5 dígitos

Sabemos:

$x_1 * 2^4 + x_2 * 2^3 + x_3 * 2^2 + x_4 * 2^1 + x_5 * 2^0$

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$28 \;/\; 2 = 14 \quad \text{Resto} = 0$$
$$14 \;/\; 2 = 7 \quad \text{Resto} = 0$$
$$7 \;/\; 2 = 3 \quad \text{Resto} = 1$$
$$3 \;/\; 2 = 1 \quad \text{Resto} = 1$$

Quedando así el número 11100.

Otra aproximación.

Busco $2^x \geq 28 \rightarrow 5$, ya que $2^5 = 32$. Necesitaré 5 dígitos

Sabemos:

$x_1 * 2^4 + x_2 * 2^3 + x_3 * 2^2 + x_4 * 2^1 + x_5 * 2^0$

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$28 \ / \ 2 \quad = 14 \quad \text{Resto} = 0$

$14 \ / \ 2 \quad = 7 \quad \text{Resto} = 0$

$7 \ / \ 2 \quad = 3 \quad \text{Resto} = 1$

$3 \ / \ 2 \quad = 1 \quad \text{Resto} = 1$

Quedando así el número 11100.

Otra aproximación.

Busco $2^x \geq 28 \rightarrow 5$, ya que $2^5 = 32$. Necesitaré 5 dígitos

Sabemos:

$x_1 * 2^4 + x_2 * 2^3 + x_3 * 2^2 + x_4 * 2^1 + x_5 * 2^0$

1				

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$\begin{array}{lll} 28 / 2 & = 14 & \text{Resto} = 0 \\ 14 / 2 & = 7 & \text{Resto} = 0 \\ 7 / 2 & = 3 & \text{Resto} = 1 \\ 3 / 2 & = 1 & \text{Resto} = 1 \end{array}$$

Quedando así el número 11100.

Otra aproximación.

Busco $2^x \geq 28 \rightarrow 5$, ya que $2^5 = 32$. Necesitaré 5 dígitos

Sabemos:

$x_1 * 2^4 + x_2 * 2^3 + x_3 * 2^2 + x_4 * 2^1 + x_5 * 2^0$

1	1			

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$\begin{array}{lll} 28\ /\ 2 & = 14 & \text{Resto} = 0 \\ 14\ /\ 2 & = 7 & \text{Resto} = 0 \\ 7\ /\ 2 & = 3 & \text{Resto} = 1 \\ 3\ /\ 2 & = 1 & \text{Resto} = 1 \end{array}$$

Quedando así el número 11100.

Otra aproximación.

Busco $2^x \geq 28 \rightarrow 5$, ya que $2^5 = 32$. Necesitaré 5 dígitos

Sabemos:

$x_1 * 2^4 + x_2 * 2^3 + x_3 * 2^2 + x_4 * 2^1 + x_5 * 2^0$

1	1	1		

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$
\begin{array}{llll}
28 \;/\; 2 & = 14 & \text{Resto} = 0 \\
14 \;/\; 2 & = 7 & \text{Resto} = 0 \\
7 \;/\; 2 & = 3 & \text{Resto} = 1 \\
3 \;/\; 2 & = 1 & \text{Resto} = 1
\end{array}
$$

Quedando así el número 11100.

Otra aproximación.

Busco $2^x \geq 28 \rightarrow 5$, ya que $2^5 = 32$. Necesitaré 5 dígitos

Sabemos:

$x_1 * 2^4 + x_2 * 2^3 + x_3 * 2^2 + x_4 * 2^1 + x_5 * 2^0$

1	1	1	0	

¿Cómo pasamos rápido de decimal a binario?

Pasar el número 28 y a binario.

Si hacemos divisiones sucesivas:

$$\begin{array}{lll} 28 \;/\; 2 & = 14 & \text{Resto} = 0 \\ 14 \;/\; 2 & = 7 & \text{Resto} = 0 \\ 7 \;/\; 2 & = 3 & \text{Resto} = 1 \\ 3 \;/\; 2 & = 1 & \text{Resto} = 1 \end{array}$$

Quedando así el número 11100.

Otra aproximación.

Busco $2^x \geq 28 \to 5$, ya que $2^5 = 32$. Necesitaré 5 dígitos

Sabemos:

$x_1 * 2^4 + x_2 * 2^3 + x_3 * 2^2 + x_4 * 2^1 + x_5 * 2^0$

1	1	1	0	0

¿Cómo pasamos rápido de binario a hexa?

Pasar el número 10101100

- ¿Cuánto elementos puedo representar con un dígito hexadecimal ?

¿Cómo pasamos rápido de binario a hexa?

Pasar el número 10101100

- ¿Cuánto elementos puedo representar con un dígito hexadecimal ?
- 16

¿Cómo pasamos rápido de binario a hexa?

Pasar el número 10101100

- ¿Cuánto elementos puedo representar con un dígito hexadecimal ?
- 16
- Luego es potencia de 2 directo, 2^4

¿Cómo pasamos rápido de binario a hexa?

Pasar el número 10101100

- ¿Cuánto elementos puedo representar con un dígito hexadecimal ?
- 16
- Luego es potencia de 2 directo, 2^4
- o sea puedo separarlos de a 4:

1010	1100
A	C

¿Cómo pasamos rápido de binario a hexa?

Pasar el número 10101100

- ¿Cuánto elementos puedo representar con un dígito hexadecimal ?
- 16
- Luego es potencia de 2 directo, 2^4
- o sea puedo separarlos de a 4:

1010	1100
A	C

¿Cómo pasamos rápido de binario a hexa?

Pasar el número 10101100

- ¿Cuánto elementos puedo representar con un dígito hexadecimal ?
- 16
- Luego es potencia de 2 directo, 2^4
- o sea puedo separarlos de a 4:

1010	1100
A	C

¿Y para pasar de hexadecimal a binario?

¿Cómo pasamos rápido de binario a hexa?

Pasar el número 10101100

- ¿Cuánto elementos puedo representar con un dígito hexadecimal ?
- 16
- Luego es potencia de 2 directo, 2^4
- o sea puedo separarlos de a 4:

1010	1100
A	C

¿Y para pasar de hexadecimal a binario?

¿Y para pasar a complemento a 2?

¿Cómo pasamos rápido de binario a hexa?

Pasar el número 10101100

- ¿Cuánto elementos puedo representar con un dígito hexadecimal ?
- 16
- Luego es potencia de 2 directo, 2^4
- o sea puedo separarlos de a 4:

1010	1100
A	C

¿Y para pasar de hexadecimal a binario?
¿Y para pasar a complemento a 2? Hay que hacer la guía...

Sobre el Overflow y el Carry...

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Algunos ejemplos para pensar **en C2**...

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2$$

Sobre el Overflow y el Carry...

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2$$

$$\begin{array}{r} 0\,1\,0\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,0\,0\,1\,0 \end{array}$$

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2$$

$$\begin{array}{r} 0\ 1\ 0\ 1 \\ +\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 1\ 0 \end{array} \quad (C \leftarrow C)$$

OK!

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2 \qquad -5 - 3 = (-5) + (-3) = (-8)$$

$$\begin{array}{r} 0\ 1\ 0\ 1 \\ +\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 1\ 0 \end{array} \quad (C \leftarrow C)$$

OK!

Algunos ejemplos para pensar **en C2**...

$$5-3=5+(-3)=2$$

$$\begin{array}{r} 0\,1\,0\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,0\,0\,1\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5-3=(-5)+(-3)=(-8)$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,1\,0\,0\,0 \end{array}$$

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2$$

$$\begin{array}{r} 0\ 1\ 0\ 1 \\ +\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 1\ 0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 3 = (-5) + (-3) = (-8)$$

$$\begin{array}{r} 1\ 0\ 1\ 1 \\ +\ 1\ 1\ 0\ 1 \\ \hline 1\ 1\ 0\ 0\ 0 \end{array} \quad (C \leftarrow C)$$

OK!

Sobre el Overflow y el Carry...

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2$$

$$\begin{array}{r} 0\ 1\ 0\ 1 \\ +\ 1\ 1\ 0\ 1 \\ \hline \color{orange}{1}\ 0\ 0\ 1\ 0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 3 = (-5) + (-3) = (-8)$$

$$\begin{array}{r} 1\ 0\ 1\ 1 \\ +\ 1\ 1\ 0\ 1 \\ \hline \color{orange}{1}\ 1\ 0\ 0\ 0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 4 = (-5) + (-4) = -9$$

Sobre el Overflow y el Carry...

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2$$

$$\begin{array}{r} 0\,1\,0\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,0\,0\,1\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 3 = (-5) + (-3) = (-8)$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,1\,0\,0\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 4 = (-5) + (-4) = -9$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\,1\,1\,0\,0 \\ \hline 1\,0\,1\,1\,1 \end{array}$$

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2$$

$$\begin{array}{r} 0\,1\,0\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,0\,0\,1\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 3 = (-5) + (-3) = (-8)$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,1\,0\,0\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 4 = (-5) + (-4) = -9$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\,1\,1\,0\,0 \\ \hline 1\,0\,1\,1\,1 \end{array} \quad (C \leftarrow \overline{C})$$

OVERFLOW!

Sobre el Overflow y el Carry...

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2$$

$$\begin{array}{r} 0\,1\,0\,1 \\ +\ 1\,1\,0\,1 \\ \hline 1\ 0\,0\,1\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 3 = (-5) + (-3) = (-8)$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\ 1\,1\,0\,1 \\ \hline 1\ 1\,0\,0\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 4 = (-5) + (-4) = -9$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\ 1\,1\,0\,0 \\ \hline 1\ 0\,1\,1\,1 \end{array} \quad (C \leftarrow \overline{C})$$

OVERFLOW!

$$5 + 4 = 9$$

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2$$

$$\begin{array}{r} 0\,1\,0\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,0\,0\,1\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 3 = (-5) + (-3) = (-8)$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,1\,0\,0\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 4 = (-5) + (-4) = -9$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\,1\,1\,0\,0 \\ \hline 1\,0\,1\,1\,1 \end{array} \quad (C \leftarrow \overline{C})$$

OVERFLOW!

$$5 + 4 = 9$$

$$\begin{array}{r} 0\,1\,0\,1 \\ +\,0\,1\,0\,0 \\ \hline 0\,1\,0\,0\,1 \end{array}$$

Sobre el Overflow y el Carry...

Algunos ejemplos para pensar **en C2**...

$$5 - 3 = 5 + (-3) = 2$$

$$\begin{array}{r} 0\,1\,0\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,0\,0\,1\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 3 = (-5) + (-3) = (-8)$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\,1\,1\,0\,1 \\ \hline 1\,1\,0\,0\,0 \end{array} \quad (C \leftarrow C)$$

OK!

$$-5 - 4 = (-5) + (-4) = -9$$

$$\begin{array}{r} 1\,0\,1\,1 \\ +\,1\,1\,0\,0 \\ \hline 1\,0\,1\,1\,1 \end{array} \quad (C \leftarrow \overline{C})$$

OVERFLOW!

$$5 + 4 = 9$$

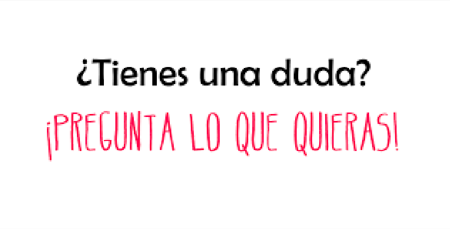
$$\begin{array}{r} 0\,1\,0\,1 \\ +\,0\,1\,0\,0 \\ \hline 0\,1\,0\,0\,1 \end{array} \quad (\overline{C} \leftarrow C)$$

OVERFLOW!

Próximas clases

- ¡Ya se puede hacer la práctica 1 completa!.
- Circuitos Combinatorios.
- Circuitos Secuenciales.

¿ Dudas ?


¿Tienes una duda?
¡PREGUNTA LO QUE QUIERAS!


YOUR ATTENTION
I THANK YOU
FOR