

# Computación – Curso 2024



Números en punto fijo

Prof. Jorge M. Runco

Lic. en Física Medica

Lic. en Física

# Representación de números enteros

---

- Sin signo
- Módulo y signo
- Complemento a uno (  $C_{a1}$  )  
Complemento a la base reducida
- Complemento a dos (  $C_{a2}$  )  
Complemento a la base
- Exceso

# Números enteros sin signo

---

- Si el número tiene  $n$  bits, puedo representar
  - $2^n =$  números distintos
- El rango va desde
  - 0 a  $(2^n - 1)$

# Números enteros sin signo

---

Ejemplo:  $n = 3$  bits

Decimal	Representación sin signo
0	000
1	001
2	010
..	....
7	111

# Números enteros sin signo

---

Ejemplo:  $n = 8$  bits

0	00000000
..	.....
128	10000000
..	.....
254	11111110
255	11111111

# Números enteros sin signo

---

- RECORDAR: la cantidad de representaciones distintas depende del número de bits

$$\text{N}^{\circ}\text{s distintos} = 2^n \quad \leftarrow$$

# Representación en BCS

---

- Con  $n$  bits, 1 bit representa al signo y  $n-1$  bits a la magnitud



- El bit  $n-1$  (extremo izquierdo) representa sólo al signo
- Los bits  $0$  a  $n-2$  la magnitud

# Binario con signo

---

- Un 0 en el bit de signo indica que el número es positivo
- Un 1 en el bit de signo indica que el número es negativo
- Los bits  $0 \rightarrow n-2$  representan el valor absoluto en binario
- El rango:  $-(2^{n-1} - 1) \rightarrow +(2^{n-1} - 1)$  con 2 ceros



# Binario con signo

---

## ► Ejemplos

$$+32_{10} = \overset{+}{\text{00100000}}$$

32

$$-32_{10} = \overset{-}{\text{10100000}}$$

32

$$+7_{10} = \text{00000111}$$

$$-7_{10} = \text{10000111}$$

$$+41_{10} = \text{00101001}$$

$$-41_{10} = \text{10101001}$$

# Binario con signo (3)

---

➤ Ejemplo:  $n=8$  bits

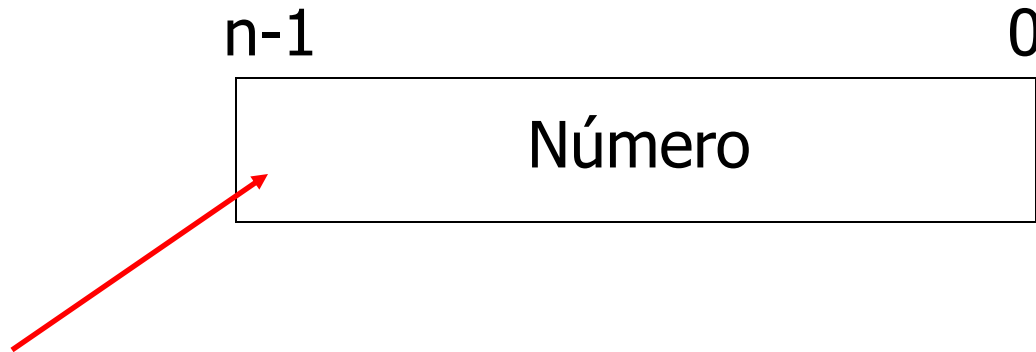
Números negativos {  $11111111 \leftarrow -(2^{n-1} - 1) = -127$   
...  
 $10000000 \leftarrow -0$

Números positivos {  $01111111 \leftarrow +(2^{n-1} - 1) = +127$   
...  
 $00000000 \leftarrow +0$

# Representación en Ca1

---

❖ Los  $n$  bits representan al número



❖ Información del signo

# Ca1

---

- Si el número es positivo, los  $n$  bits tienen la representación binaria del número (como siempre )
- Si el número es negativo, los  $n$  bits tienen el Ca1 del valor deseado.
- El Ca1 de un número en base 2 se obtiene invirtiendo todos los bits

# Ca1


---

- Los positivos empiezan con cero (0)
- Los negativos empiezan con uno (1)
- El rango va desde  
     $-(2^{n-1} - 1)$  a  $+(2^{n-1} - 1)$   
con dos ceros

# Ca1


---

## ➤ Ejemplos

  
 $+32_{10} = 00100000$

$$+7_{10} = 00000111$$

$$+41_{10} = 00101001$$

  
 $-32_{10} = 11011111$

$$-7_{10} = 11111000$$

$$-41_{10} = 11010110$$

# Ca1

---

➤ Ejemplo:  $n=8$  bits

Números negativos {  $11111111 \leftarrow -0$   
...  
 $10000000 \leftarrow -(2^{n-1} - 1) = -127$

Números positivos {  $01111111 \leftarrow +(2^{n-1} - 1) = +127$   
...  
 $00000000 \leftarrow +0$

# Ca1

---

- ❖ Dada una cadena de bits ¿qué número decimal representa si lo interpretamos en Ca1?
- ❖ Cuando es positivo:

$$01100000 = 1 \times 2^6 + 1 \times 2^5 = 64 + 32 = 96$$

*Como siempre*



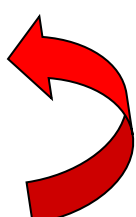
# Ca1

---

❖ Cuando es negativo:

✓ Ca1 del número y obtengo el positivo  
Ej.

$$11100000 = -31$$

$$11100000 \xrightarrow{\text{red arrow}} 00011111 = +31$$


# Resumen Ca1

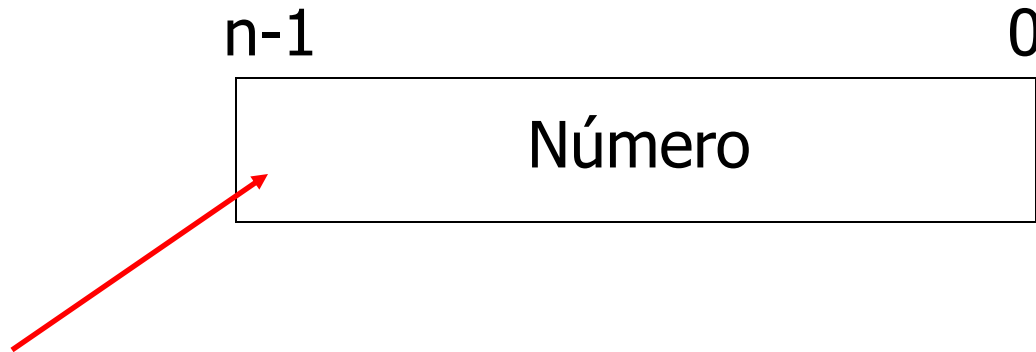
---

- ❖ El intervalo es simétrico
- ❖ Los  $n$  bits representan al número
- ❖ Los positivos empiezan con cero (0)
- ❖ Los negativos empiezan con uno (1)
- ❖ Hay dos ceros
- ❖ Números distintos  $2^n$

# Representación en Ca2

---

❖ Los  $n$  bits representan al número



❖ Información del signo

# Representación en Ca2

---

- Si el número es positivo, los  $n$  bits tienen la representación binaria del número (como siempre)
- Si el número es negativo, los  $n$  bits tienen el Ca2 del valor deseado.
- El Ca2 de un número (en base 2) se obtiene invirtiendo todos los bits (Ca1) y luego sumándole 1.

# Ca2

---

- Otra forma: “mirando” desde la derecha se escribe el número (base 2) igual hasta el primer “1” uno inclusive y luego se invierten los demás dígitos

# Ca2

---

- Los positivos empiezan con cero (0)
- Los negativos empiezan con uno (1)
- El rango es asimétrico y va desde  $-(2^{n-1})$  a  $+(2^{n-1} - 1)$
- Hay un solo cero

# Ca2

---

## ➤ Ejemplos

$$\begin{aligned} +32_{10} &= \text{00}\text{100000} \quad \leftarrow \text{"mirando"} \\ &\quad \text{desde la derecha} \\ -32_{10} &= \text{11}\text{100000} \end{aligned}$$

- ✓ Los dígitos en rojo se copiaron igual
- ✓ Los dígitos en azul se invirtieron

## Ca2 (otra forma )

---

$$+32_{10} = 00100000$$

1111

11011111 invierto todos los bits

+ 1 le sumo 1

$$-32_{10} = 11100000 \quad \leftarrow \text{ en Ca2}$$



# Ca2

---

➤ Ejemplo :  $n=8$  bits

Números negativos {  
11111111 ← -1  
...  
10000000 ←  $-(2^{n-1}) = -128$

Números positivos {  
01111111 ←  $+(2^{n-1} - 1) = +127$   
...  
00000000 ← +0

# Ca2

---

- ❑ Dada una cadena de bits ¿qué número decimal representa si lo interpretamos en Ca2?
- ❑ Cuando es positivo:

$$01100000 = 1 \times 2^6 + 1 \times 2^5 = 64 + 32 = 96$$

*Como siempre*

# Ca2

---

❑ Cuando es negativo:

✓ Ca2 el número y obtengo el positivo

Ej.

$$11100000 = -32$$



$$11100000 \longrightarrow 00100000 = +32$$

# Resumen Ca2

---

- ❑ El intervalo es asimétrico, hay un - más
- ❑ Los  $n$  bits representan al número
- ❑ Los positivos empiezan con cero (0)
- ❑ Los negativos empiezan con uno (1)
- ❑ Hay un solo cero
- ❑ Números distintos  $2^n$

Decimal	Magnitud con signo	Complemento a 1	Complemento a dos
+0	0 000	0 000	0 000
+1	0 001	0 001	0 001
+2	0 010	0 010	0 010
+3	0 011	0 011	0 011
+4	0 100	0 100	0 100
+5	0 101	0 101	0 101
+6	0 110	0 110	0 110
+7	0 111	0 111	0 111
-0	1 000	1 111	-
-1	1 001	1 110	1 111
-2	1 010	1 101	1 110
-3	1 011	1 100	1 101
-4	1 100	1 011	1 100
-5	1 101	1 010	1 011
-6	1 110	1 001	1 010
-7	1 111	1 000	1 001
-8	-	-	1 000

# Exceso

---

- Representar números en exceso: partimos del número en  $Ca_2$  y le sumamos una cantidad fija. El número resultante es la representación en exceso.
- La cantidad fija en general es:  $10000\dots$ . Sin importar la cantidad de bits.
- En el caso de IEEE754 que es un estándar, la cantidad fija es de la forma (para 8 bits):  
 $01111111$ .

# Ejemplos en exceso (positivo)

---

- Representar en exceso +3 con 8 bits
- 1) +3 en Ca2 = 00000011
- 2) sumamos la cantidad fija

$$\begin{array}{r} + \quad 00000011 \\ \quad 10000000 \\ \hline \quad 10000011 \end{array} \quad +3 \text{ en exceso}$$

# Ejemplos en exceso (negativo)

---

□ Representar en exceso -3 con 8 bits

□ 1) +3 en Ca2 = 00000011

-3 en Ca2 = 11111101

□ 2) sumamos la cantidad fija

$$\begin{array}{r} 11111101 \\ + \\ 10000000 \\ \hline 1 \leftarrow 01111101 \end{array} \quad - 3 \text{ en exceso}$$



# Conclusiones

---

- El rango en exceso es el mismo que en  $\text{Ca}^{2+}$ .
- Cambian las representaciones de + y - con respecto a  $\text{Ca}^{2+}$ .
- Los positivos empiezan con 1.
- Los negativos empiezan con 0.

# Exceso IEEE754 (positivo)

---

- Representar en exceso IEEE754 +3 con 8 bits
- 1) +3 en  $C_{a2}$  = 00000011
- 2) sumamos la cantidad fija

$$\begin{array}{r} + \quad 00000011 \\ \quad 01111111 \\ \hline 10000010 \end{array} \quad +3 \text{ en exceso IEEE}$$

# Exceso IEEE754 (negativo)

---

□ Representar en exceso IEEE754 -3 con 8 bits

□ 1) +3 en Ca2 = 00000011

-3 en Ca2 = 11111101

□ 2) sumamos la cantidad fija

$$\begin{array}{r} + \quad 11111101 \\ \quad 01111111 \\ \hline 1 \leftarrow 01111100 - 3 \text{ en exceso IEEE} \end{array}$$

# Para recordar...

---

- Representar un número en exceso significa sumarle una cantidad determinada (exceso) a una representación en  $Ca2$ .
- En general el exceso es: 1000.... Un 1 en el bit mas significativo y 0s en el resto.
- Otro exceso que vimos es el de la norma IEEE754 que es: 01111..... Un 0 en el bit mas significativo y 1s en el resto.

# Suma binaria

---

- Para sumar números binarios, seguimos las reglas utilizadas para la suma de números decimales. La única diferencia es que el sistema binario consta sólo de dos caracteres (0 y 1).

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10 (*)$$

- Hay sólo 4 posibles combinaciones en la suma binaria

- (\*) Aparece “me llevo 1”

## Suma binaria: ejemplo

$$\begin{array}{r}
 1111 \\
 0010100 \\
 + 0111101 \\
 \hline
 1010001
 \end{array}$$

# Resta binaria

---

- Para restar números binarios, seguimos las 4 reglas mostradas abajo. Prestar atención al caso 0-1.

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 10 - 1 = 1 (*)$$

- Hay sólo 4 posibles combinaciones en la resta binaria

- (\*) Aparece “pido 1 prestado”

# Resta binaria: ejemplo

---

$$\begin{array}{r} \phantom{0} 0 \phantom{1} 0 \\ 1 \cancel{0} 1 \cancel{0} 1 1 \\ - \phantom{0} 1 0 1 0 1 0 1 \\ \hline 0 0 1 0 1 1 0 \end{array}$$



# Bits de condición (banderas)

---

- ❑ Son bits que el procesador establece de modo automático acorde al resultado de cada operación realizada.
- ❑ Sus valores permitirán tomar decisiones como:
  - ❑ Realizar o no una transferencia de control.
  - ❑ Determinar relaciones entre números (mayor, menor, igual).

# Banderas aritméticas

---

- ❑ Z (cero): vale 1 si el resultado de la operación son todos bits 0.
- ❑ C (carry): en la suma vale 1 si hay acarreo del bit más significativo; en la resta vale 1 si hay 'borrow' hacia el bit más significativo.
  - ❑ Cuando la operación involucra números sin signo,  $C=1$  indica una condición fuera de rango.

# Banderas aritméticas

---

- ❑ N (negativo): igual al bit más significativo del resultado.
  - Es 1 si el resultado es negativo
  
- ❑ V (overflow): en 1 indica una condición de fuera de rango (desborde) en  $\text{Ca}2$ .
  - El resultado no se puede expresar con el número de bits utilizado.

## Suma en Ca2

---

- Para sumar dos números en Ca2 se suman los  $n$  bits directamente.
- Si sumamos dos números  $+$  y el resultado es  $-$  ó si sumamos dos  $-$  y el resultado es  $+$  **hay overflow**, en otro caso no lo hay.
- Si los N<sup>o</sup>s son de distinto signo nunca puede haber **overflow**.

## Resta en Ca2

---

- Para restar dos números en Ca2, se restan los  $n$  bits directamente. También se puede Ca2 el sustraendo y transformar la resta en suma.
- Si a un  $N^0 +$  le restamos un  $N^0 -$  y el resultado es  $-$  ó si a un  $N^0 -$  le restamos un  $+$  y el resultado es  $+$  **hay borrow** en la resta.
- Si son del mismo signo nunca hay borrow

# Bits de condición para la suma (no está más....)

---

Operación	NZVC	Ca2	Sin signo
-----------	------	-----	-----------

$$\begin{array}{r} + 0100 \\ 0010 \\ \hline 0110 \end{array}$$

0000

$$\begin{array}{r} + +4 \\ + +2 \\ \hline +6 \end{array}$$

$$\begin{array}{r} + +4 \\ + +2 \\ \hline +6 \end{array}$$

✓ Los dos resultados son correctos.

# Bits de condición para la suma (no está más...)

---

Operación	NZVC	Ca2	Sin signo
$\begin{array}{r} + 1001 \\ + 1100 \\ \hline 1 \leftarrow 0101 \end{array}$	$0011$	$\begin{array}{r} + -7 \\ + -4 \\ \hline V +5 \end{array}$	$\begin{array}{r} + 9 \\ + 12 \\ \hline C 5 \end{array}$

✓ Los dos resultados son incorrectos.

# Bits de condición para la resta (no está más...)

---

Operación	NZVC	Ca2	Sin signo
$\begin{array}{r} 1 \rightarrow 0101 \\ \underline{\phantom{0101}} \\ 0111 \\ \hline 1110 \end{array}$	$1001$	$\begin{array}{r} +5 \\ +7 \\ \hline -2 \end{array}$	$\begin{array}{r} 5 \\ 7 \\ \hline \text{B } 14 \end{array}$

✓ Ca2 correcto, sin signo incorrecto.



# Bits de condición para la resta (no está más....)

---

Operación	NZVC	Ca2	Sin signo
-----------	------	-----	-----------

$$\begin{array}{r} \text{—} 1001 \\ 0100 \\ \hline 0101 \end{array}$$


0010

$$\begin{array}{r} \text{—} -7 \\ +4 \\ \hline \text{V } +5 \end{array}$$

$$\begin{array}{r} \text{—} 9 \\ 4 \\ \hline 5 \end{array}$$

✓ Ca2 incorrecto, sin signo correcto.

## Práctica 3: Ej. 9) (no está más....)

Operación	ZNVC	Ca2	Sin signo
11			
+ 01110000	0110	+ 112	+ 112
00101111		+ 47	47
<hr/> 10011111		 <hr/> -97 V	<hr/> 159 😊

✓ Ca2 incorrecto, sin signo correcto.

## Práctica 3: Ej. 9) (no está más....)

Operación	ZNVC	Ca2	Sin signo
1			
$\begin{array}{r} + 01000000 \\ + 01000000 \\ \hline 10000000 \end{array}$	0110	$\begin{array}{r} + 64 \\ + 64 \\ \hline -128 \end{array}$	$\begin{array}{r} + 64 \\ + 64 \\ \hline 128 \end{array}$

✓ Ca2 incorrecto, sin signo correcto.

## Práctica 3: Ej. 9) (no está más....)

Operación	ZNVC	Ca2	Sin signo
$\begin{array}{r} 11111111 \\ + 11111111 \\ 00000001 \\ \hline 1 \leftarrow 00000000 \end{array}$	$\begin{array}{r} 1001 \end{array}$	$\begin{array}{r} -1 \\ + 1 \\ \hline 0 \text{ 😊} \end{array}$	$\begin{array}{r} 255 \\ + 1 \\ \hline 0 \text{ ⚡ C} \end{array}$

✓ Ca2 correcto, sin signo incorrecto.

## Práctica 3: Ej. 9) (no está más....)

Operación	ZNVC	Ca2	Sin signo
$\begin{array}{r} 1111111 \\ + 01111111 \\ \hline 00000001 \\ \hline 10000000 \end{array}$	$\begin{array}{r} 0110 \end{array}$	$\begin{array}{r} 127 \\ + 1 \\ \hline -128 \end{array}$	$\begin{array}{r} 127 \\ + 1 \\ \hline 128 \end{array}$



V



✓ Ca2 incorrecto, sin signo correcto.

## Práctica 3: Ej. 9) (no está más....)

Operación	ZNVC	Ca2	Sin signo
$\begin{array}{r} + \quad 01111110 \\ \quad 00000001 \\ \hline 01111111 \end{array}$	0000	$\begin{array}{r} + \quad 126 \\ \quad 1 \\ \hline 127 \end{array} \text{ 😊}$	$\begin{array}{r} + \quad 126 \\ \quad 1 \\ \hline 127 \end{array} \text{ 😊}$

✓ Ca2 correcto, sin signo correcto.

## Práctica 3: Ej. 9) (no está más....)

Operación	ZNVC	Ca2	Sin signo
$\begin{array}{r} 111111 \\ + 11111111 \\ 11111110 \\ \hline 1 \leftarrow 11111101 \end{array}$	$\begin{array}{r} 0101 \end{array}$	$\begin{array}{r} -1 \\ + -2 \\ \hline -3 \end{array}$	$\begin{array}{r} 255 \\ + 254 \\ \hline 253 \end{array}$

✓ Ca2 correcto, sin signo incorrecto.

## Práctica 3: Ej. 9) (no está más....)

Operación	ZNVC	Ca2	Sin signo
$\begin{array}{r} 1 \\ + \\ 11000000 \\ 11000000 \\ \hline 1 \leftarrow 10000000 \end{array}$	0101	$\begin{array}{r} -64 \\ + \\ -64 \\ \hline -128 \end{array}$ 😊	$\begin{array}{r} + \\ 192 \\ 192 \\ \hline 128 \end{array}$ ⚡ C

✓ Ca2 correcto, sin signo incorrecto.

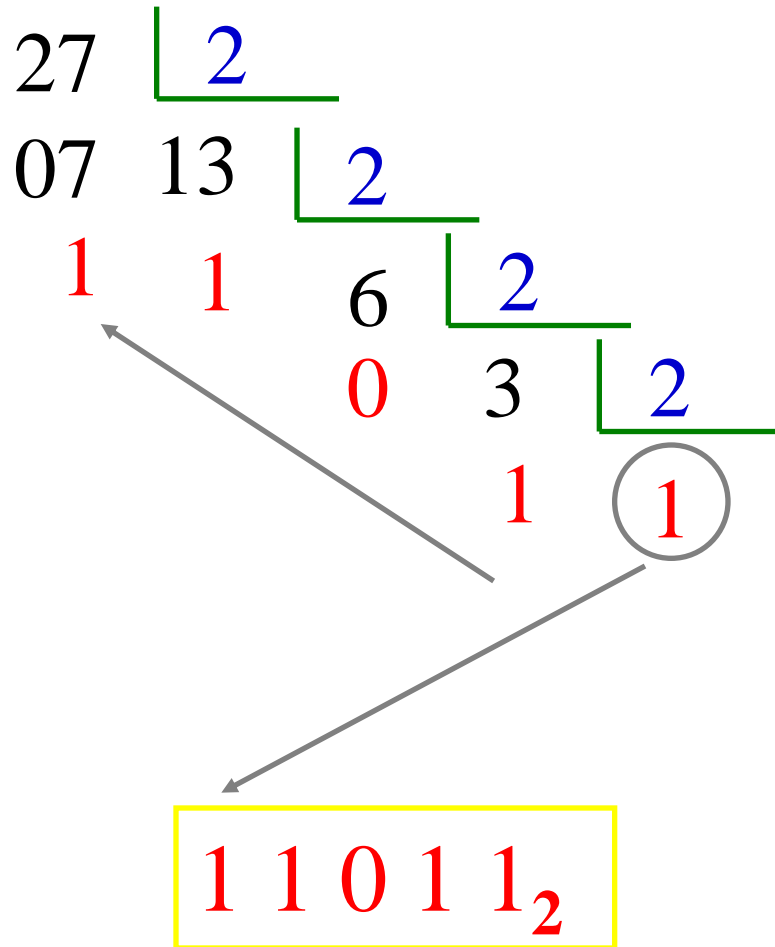


## Práctica 3: Ej. 9) (no está más....)

Operación	ZNVC	Ca2	Sin signo
$\begin{array}{r} 1 \rightarrow 101111111 \\ - 111111111 \\ \hline 100000000 \end{array}$	$\begin{array}{r} 0111 \end{array}$	$\begin{array}{r} 127 \\ - 1 \\ \hline -128 \end{array} \text{ V}$	$\begin{array}{r} 127 \\ - 255 \\ \hline 128 \end{array} \text{ C}$

✓ Ca2 incorrecto, sin signo incorrecto.

# Ejemplo



$$\underbrace{1}_1 \underbrace{1011}_B = 1B_{16}$$

# Ejemplo

---

$$54_{10} \longrightarrow \begin{array}{cccccc} 1 & 1 & 0 & 1 & 1 & 0_2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array} =$$

$$= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$= 32 + 16 + 4 + 2 = 54$$

$$\begin{array}{cccccc} 1 & 1 & 0 & 1 & 1 & 0 \\ \underbrace{\hspace{1.5cm}}_{36} & \underbrace{\hspace{1.5cm}}_{16} & & \underbrace{\hspace{1.5cm}}_{6} & & \\ & 3 & & & & \end{array} \longrightarrow$$

# Ejemplo

---

Base 10	Base 2	Base 16
108	1101100	6C
542	1000011110	21E
1084	10000111100	43C
2009	11111011001	7D9
2168	100001111000	878

# Ejemplo

$$108_{10} \xrightarrow{\text{red}} \begin{array}{ccccccc} 1 & 1 & 0 & 1 & 1 & 0 & 0_2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array} =$$

$$= 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$= 64 + 32 + 8 + 4 = 108$$

$$\underbrace{1 \ 1 \ 0}_6 \underbrace{1 \ 1 \ 0 \ 0}_C \xrightarrow{\text{red}} 6C_{16}$$

# Ejemplo

$$542_{10} \longrightarrow \begin{array}{cccccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 2^9 & 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array} 2$$

$$= 1 \times 2^9 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1$$

$$= 512 + 16 + 8 + 4 + 2 = 542$$

$$\begin{array}{ccccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \underbrace{\hspace{1cm}} & \underbrace{\hspace{2cm}} & \underbrace{\hspace{2cm}} & & & & & & \\ 2 & 1 & & & & & E & & \end{array} \longrightarrow 21E$$

# Ejemplo

$$1084_{10} \xrightarrow{\text{red}} \begin{array}{cccccccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 2^{10} & 2^9 & 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array} =$$

$$= 1 \times 2^{10} + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2$$

$$= 1024 + 32 + 16 + 8 + 4 = 1084$$

$$\underbrace{1\ 0\ 0}_{4} \underbrace{0\ 0\ 1\ 1}_3 \underbrace{1\ 1\ 0\ 0}_C \xrightarrow{\text{red}} 43C$$

# Ejemplo

---

$$\begin{array}{cccccccccccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0_2 = \\ \downarrow & & & & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & & \downarrow & \\ 2^{12} & & & & 2^8 & 2^7 & 2^6 & 2^5 & & 2^3 & & 2^1 & \end{array}$$

$$= 1 \times 2^{12} + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^1$$

$$= 4096 + 256 + 128 + 64 + 32 + 8 + 2 = 4586 \quad \leftarrow$$



# Ejemplo

---

$$\begin{array}{cccccccccccccccc} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0_2 = \\ \downarrow & & \downarrow & & & \downarrow & \downarrow & \downarrow & & & \downarrow & \downarrow & \downarrow & \downarrow & & & \\ 2^{16} & & 2^{14} & & & 2^{11} & 2^{10} & 2^9 & & & 2^6 & 2^5 & 2^4 & 2^3 & & & \end{array}$$

$$= 2^{16} + 2^{14} + 2^{11} + 2^{10} + 2^9 + 2^6 + 2^5 + 2^4 + 2^3 =$$

$$= 65536 + 16384 + 2048 + 1024 + 512 + 64 + 32 + 16 + 8 =$$

$$= 85624 \quad \leftarrow$$

# Ejemplo

---

$$\begin{array}{cccc} \text{F} & \text{E} & \text{C} & \text{B}_{16} = \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 16^3 & 16^2 & 16^1 & 16^0 \end{array}$$

$$= \text{F} \times 16^3 + \text{E} \times 16^2 + \text{C} \times 16^1 + \text{B} \times 16^0 =$$

$$= 15 \times 16^3 + 14 \times 16^2 + 12 \times 16^1 + 11 \times 16^0 =$$

$$= 15 \times 4096 + 14 \times 256 + 12 \times 16 + 11 \times 1 =$$

$$= 61440 + 3584 + 192 + 11 = 65227 \quad \leftarrow$$

# Ejemplo

---

$$\begin{array}{cccc} 1 & B & 2 & C_{16} = \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 16^3 & 16^2 & 16^1 & 16^0 \end{array}$$

$$= 1 \times 16^3 + B \times 16^2 + 2 \times 16^1 + C \times 16^0 =$$

$$= 1 \times 16^3 + 11 \times 16^2 + 2 \times 16^1 + 12 \times 16^0 =$$

$$= 1 \times 4096 + 11 \times 256 + 2 \times 16 + 12 \times 1 =$$

$$= 4096 + 2816 + 32 + 12 = 6956$$

