

Representación de datos



Representación de datos

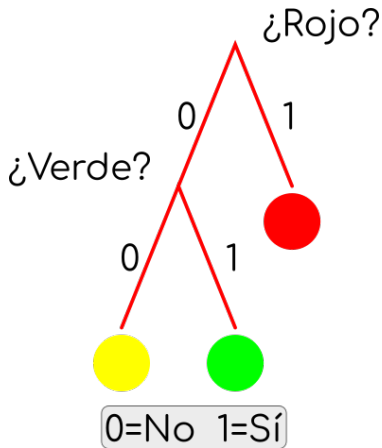
¿Cómo representamos información más compleja que un bit?

¿Cómo podríamos saber el estado de un semáforo utilizando solo preguntas con respuesta binaria *Sí/No*?



Representación de datos

¿Cómo representamos información más compleja?



Representación de datos

¿Cómo representamos información más compleja?

Podemos usar dos bits y asignar la correspondencia de forma arbitraria:

Codificación	Color
00	
01	
10	
11	-



Representación de datos

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan $\lceil \log_2(N) \rceil = k$ bits. O dicho de otra manera: *Se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.*



Representación de datos

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan $\lceil \log_2(N) \rceil = k$ bits. O dicho de otra manera: *Se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.*

Ejemplos:

- Para representar 16 colores distintos hace falta sólo:



Representación de datos

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan $\lceil \log_2(N) \rceil = k$ bits. O dicho de otra manera: *Se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.*

Ejemplos:

- Para representar 16 colores distintos hace falta sólo: **4b.**



Representación de datos

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$. Ejemplos:

- Para representar las letras $A-Z$ del ingles, solo en mayúscula, más los dígitos decimales y otros 10 símbolos auxiliares (44 símbolos en total) se necesitan:



Representación de datos

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$. Ejemplos:

- Para representar las letras $A-Z$ del ingles, solo en mayúscula, más los dígitos decimales y otros 10 símbolos auxiliares (44 símbolos en total) se necesitan: **6b**.



Representación de datos

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.

Ejemplos:

- Si queremos representar los números del -100 al 100 (201 en total), necesitamos:



Representación de datos

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.

Ejemplos:

- Si queremos representar los números del -100 al 100 (201 en total), necesitamos: **8b**.



Representación de datos

Representación de enteros sin signo (positivos)

Si tenemos k bits y queremos representar sólo enteros positivos ¿Qué rango de números podemos representar?



Representación de datos

Representación de enteros sin signo (positivos)

Si tenemos k bits y queremos representar sólo enteros positivos ¿Qué rango de números podemos representar?

- con k bits podemos representar 2^k números distintos.



Representación de datos

Representación de enteros sin signo (positivos)

Si tenemos k bits y queremos representar sólo enteros positivos ¿Qué rango de números podemos representar?

- con k bits podemos representar 2^k números distintos.
- Como queremos incluir el cero el último número no puede ser 2^k , sino $2^k - 1$.



Representación de datos

Representación de enteros sin signo (positivos)

Si tenemos k bits y queremos representar sólo enteros positivos ¿Qué rango de números podemos representar?

- Por lo tanto el *Rango de Representación* de los enteros sin signo con k bits es $[0, 2^k - 1]$.



Representación de datos

Representación de enteros sin signo (positivos)

Si tenemos k bits y queremos representar sólo enteros positivos ¿Qué rango de números podemos representar?

- Por lo tanto el *Rango de Representación* de los enteros sin signo con k bits es $[0, 2^k - 1]$.



Representación de datos

Representación de enteros positivos

Ejemplos:

- Con 2^b podemos representar:

Dec.	Bin.	2^b
0	0	00
1	1	01
2	10	10
3	11	11



Representación de enteros positivos

Rango de Representación

Ejemplos:

- Con $2b$ tenemos un rango de $[0, 3]$:
- Con $4b$ tenemos un rango de $[0, 15]$.



Representación de enteros positivos

Rango de Representación

Ejemplos:

- Con $2b$ tenemos un rango de $[0, 3]$:
- Con $4b$ tenemos un rango de $[0, 15]$.
- Con $16b$ tenemos un rango de



Representación de enteros positivos

Rango de Representación

Ejemplos:

- Con $2b$ tenemos un rango de $[0, 3]$:
- Con $4b$ tenemos un rango de $[0, 15]$.
- Con $16b$ tenemos un rango de $[0, 65535]$.
- Con $32b$ tenemos un rango de



Representación de enteros positivos

Rango de Representación

Ejemplos:

- Con $2b$ tenemos un rango de $[0, 3]$:
- Con $4b$ tenemos un rango de $[0, 15]$.
- Con $16b$ tenemos un rango de $[0, 65535]$.
- Con $32b$ tenemos un rango de $[0, 4294967295]$.



Representación de caracteres

ASCII (1963)

- 95 símbolos imprimibles y 33 caracteres de control: 128 en total.
- Desarrollado para usar sólo 7 bits.
- El bit sobrante se utilizó para crear extensiones. Actualmente se usa en el *UTF-8*.



El Byte

ASCII (1963)

Tabla *ASCII*:

Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex	
0	00	NUL	16	10	DLE	32	20		48	30	0
1	01	SOH	17	11	DC1	33	21	!	49	31	1
2	02	STX	18	12	DC2	34	22	"	50	32	2
3	03	ETX	19	13	DC3	35	23	#	51	33	3
4	04	EOT	20	14	DC4	36	24	\$	52	34	4
5	05	ENQ	21	15	NAK	37	25	%	53	35	5



El Byte

ASCII (1963)

Representando “IC 2023” en *ASCII*:

I	49	0100	1001
C	43	0100	0011
	20	0010	0000
2	32	0011	0010
0	30	0011	0000
2	32	0011	0010
3	33	0011	0011



El bit

Para pensar:

- Si una imagen tiene una altura de $1024px$, un ancho de $720px$, y cada pixel puede tener 1 de 16 000 colores distintos ¿Cuántos bits se necesitan para representar la imagen?
- ¿Cuántos bits necesito para poder representar la edad de una persona?



Representación de datos numéricos

- Enteros
 - Sin Signo (SS)
 - Signo-Magnitud (SM).
 - Complemento a 2 (C2)
 - En exceso
- Fraccionarios.
 - Punto Fijo.
 - Punto Flotante

 Solamente se puede operar con datos en el mismo sistema de representación

Representación de datos numéricos

Representación Signo-Magnitud

- El primer bit a la izquierda representa el signo.
- Los restantes bits, el valor absoluto o magnitud.



Representación de datos numéricos

Representación Signo-Magnitud

- El primer bit a la izquierda representa el signo.
- Los restantes bits, el valor absoluto o magnitud.

¿Cuál es el RR de este sistema?



¿Consultas?



Representación de datos numéricos

- Enteros
 - Sin signo (SS)
 - Signo Magnitud (SM)
 - Complemento a 2 (C2)
 - Exceso
- Fraccionarios:
 - Punto Fijo
 - Punto Flotante



Representación de datos numéricos

- Solamente se puede operar entre datos representados con el mismo sistema de representación.
- El resultado de toda operación vuelve a estar representado en el mismo sistema.



Sistema de Signo-magnitud

- El primer bit a la izquierda representa el signo.
 - 0 -> Positivo
 - 1 -> Negativo
- Los restantes bits, el valor absoluto o magnitud



RR de Signo Magnitud

Si tenemos k bits, el RR es:
 $[-2^{k-1} - 1, (2^{k-1}) - 1]$



Limitaciones

- No utiliza eficientemente los k bits.
- No es conveniente para la aritmética (poco práctico).



Operación de Complemento a 2

- La operación de complementar a 2 consiste en obtener el opuesto de un número aritméticamente (el que tiene el mismo valor absoluto pero signo opuesto).
- $C2(a) + a = 0$



Operación de C2

Para obtener el complemento a 2 de un número:

- Se invierte cada uno de los bits (reemplazando 0 por 1 y viceversa)
- Al resultado se le suma 1.

Ejemplo: 111010



Representación en C2

Para representar un número a en complemento a 2 con k bits, comenzamos por considerar su signo:



Representación en C2

Para representar un número a en complemento a 2 con k bits, comenzamos por considerar su signo:

- Si a es positivo o cero, lo representamos como en SS(k), es decir, lo escribimos en base 2 con k bits.



Representación en C2

- Si a es negativo, tomamos su valor absoluto, lo representamos como en $SS(k)$, y le aplicamos la **operación** complemento a 2.



Representación en C2

Ejemplo:

- Representar 17 en C2 con $k = 8$ bits.



Representación en C2

Ejemplo:

- Representar 17 en C2 con $k = 8$ bits.
- $17 = 00010001(2)$ con $k = 8$ bits.



Representación en C2

Ejemplo:

- Representar 17 en C2 con $k = 8$ bits.
- $17 = 00010001(2)$ con $k = 8$ bits.
- Representar -17 en C2 con $k = 8$ bits.



Rango de representación

RR de C2

En rango de representación del sistema de representación complemento a dos, con k bits, es: $[-2^{k-1}, (2^{k-1}) - 1]$



Ventajas del sistema de representación C2 sobre SM

- El cero tiene una única representación, lo que facilita las comparaciones.
- No requiere comprobaciones de signo antes de operar. Se opera directamente.
- El mecanismo de cálculo es eficiente y fácil de implementar en hardware.



Error MUY frecuente

Para representar cualquier número en C2...
lo tengo que complementar a 2.... **NO!**



Representación / Operación

Al *representar* en C2, la *operación* de complementar a 2 únicamente se aplica cuando queremos **obtener el opuesto de un número**.



Aritmética en C2

- Una gran ventaja que aporta el sistema Ca2 es que no es necesario implementar algoritmos de resta.
- Cuando se necesita efectuar una resta, se complementa el sustraendo y luego se lo suma al minuendo.
- Las computadoras no restan: siempre suman.



Desbordamiento (overflow)

- Decimos que hay *overflow*, cuando el resultado de una operación aritmética **cae fuera del RR** del sistema.
- Es importante que la computadora lo detecte, para informar al proceso, y que pueda descartar el resultado.



Desbordamiento en C2

- Puede ocurrir tanto al sumar dos números positivos como dos negativos.
- Es fácil detectarlo en este sistema, observando los dos últimos bits de acarreo (carry).



Detectar desbordamiento en C2

Se observan los dos últimos bits de acarreo (carry).

- Son diferentes? Hay overflow.
- Son iguales? No hay overflow.



Detectar desbordamiento en C2

Ejemplo: $23 - 9$ (extender el signo para llegar a los k bits)

- $00010111 +$
- 11110111



Detectar desbordamiento en C2

	1	1	1	1	0	1	1	1
23	0	0	0	1	0	1	1	1
+								
-9	1	1	1	1	0	1	1	1
	0	0	0	0	1	1	1	0



Detectar desbordamiento en C2

Ejemplo: $123 + 9$

- 01111011 +
- 00001001



Detectar desbordamiento en C2

	0	1	1	1	1	0	1	1
123	0	1	1	1	1	0	1	1
+								
9	0	0	0	0	1	0	0	1
	1	0	0	0	0	1	0	0



Notación en exceso

- Se determina el intervalo de enteros a representar.
- Se determina la cantidad de bits necesarios para representar todos los enteros del intervalo.



Notación en exceso

- Un intervalo de enteros, $[a, b]$, comprende n valores, donde $n = b - a + 1$
- ¿Cuántos bits necesitamos para representar n valores?
- Por ejemplo, para un intervalo $[150, 157]$.



Notación en exceso

Notemos que tanto a como b pueden ser negativos. Así, podemos representar intervalos de enteros arbitrarios, lo que nos vuelve a dar un sistema de representación con signo.



Notación en exceso

Conversión de decimal a exceso

Para calcular la secuencia binaria que corresponde a un valor decimal d , a d le restamos a y luego convertimos el resultado (que será no negativo) a $SS(k)$, es decir, a binario sin signo sobre k bits.



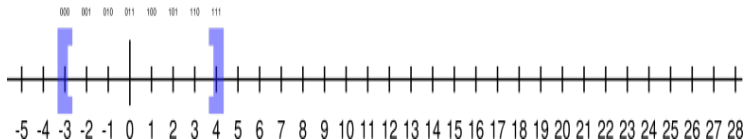
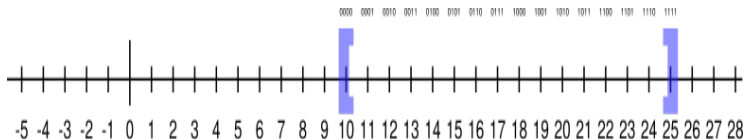
Notación en exceso

Conversión de exceso a decimal

Para calcular el valor decimal d representado por un binario en exceso, convertimos el binario a decimal como en SS(k), y al resultado (que será no negativo) le sumamos el valor de a .



Notación en exceso



Ejemplo

- Representar en el sistema de representación en exceso el intervalo $[10, 25]$. Cuántos bits serán necesarios?
- Calcular la secuencia de bits que corresponde al número 20.



Ejemplo

- Representar en el sistema de representación en exceso el intervalo $[10, 25]$. Cuántos bits serán necesarios?
- Calcular la secuencia de bits que corresponde al número 20.
- Respuesta: 1010
- Calcular el valor decimal que está representando la secuencia 1011.



¿Consultas?

