CODIFICACIÓN DE NÚMEROS

Para representar números naturales ($\mathbb{N}$) basta con utilizar cualquier sistema posicional de base b donde no exista parte fraccionaria ($q=0$).

En un computador la base natural es $b=2$ y dado que el espacio de representación está limitado a n bits, tendremos:

- RANGO DE REPRESENTACIÓN. $\text{rango}(BIN_n) = [0, 2^n - 1]$
- MÓDULO: $\text{mod}(BIN_n) = 2^n$
- $BIN(N, n) = N_{2,n}$ es la representación de $N \in \text{rango}(BIN_n)$
- $VAL\text{-}BIN(N_{2,n}) = N$ es el valor codificado por $N_{2,n} \in BIN_n$.

OPERACIONES CON $\mathbb{N}$ EN EL COMPUTADOR

• Definición de la suma en BIN_n

Sea $a, b \in \text{rango}(BIN_n)$

$$BIN(a,n) + BIN(b,n) = \begin{cases} BIN(a+b, n) \ \ des = 0 \ \ si \ \ a+b \in \text{rango}(BIN_n) \\ BIN(a+b-2^n, n) \ \ des = 1 \\ si \ \ a+b \notin \text{rango}(BIN_n) \end{cases}$$

• Definición de la resta en BIN_n

Sea $a, b \in \text{rango}(BIN_n)$

$$BIN(a,n) - BIN(b,n) = \begin{cases} BIN(a-b, n) \ \ des = 0 \ \ si \ \ a \geq b \rightarrow a-b \in \text{rango}(BIN_n) \\ BIN((a-b, 2^n), n) \ \ des = 1 \ \ si \ \ a < b \rightarrow a-b \notin \text{rango}(BIN_n) \end{cases}$$

ARITMÉTICA EN BIN_n

Se cumple que la representación de la suma de valores es equivalente a la suma de las representaciones siempre que el resultado esté en el rango de representación

$$BIN(a+b, n) = BIN(a,n) + BIN(b,n)$$

Y lo mismo ocurriría con la resta.
En definitiva, la aritmética en BIN_n conserva el sistema de representación y permite operar con representaciones.

TEOREMA FUNDAMENTAL DE LA NUMERACIÓN EN $\mathbb{Z}$

La representación de $Z \in \mathbb{Z}$ en el sistema de base b es:

$$Z = \begin{cases} Z_b = s \ \ d_{n-1} \dots d_1 \ d_0 \\ Z = (-1)^s \sum_{i=0}^{n-1} d_i \cdot b^i \ \ \text{con} \ \ s \in \{0,1\} \end{cases}$$

• EL RANGO ES SIMÉTRICO Y CUENTA CON DOS REPRESENTACIONES PARA EL 0.

OPERADOR SM (Z, n) binario

En general, el operador SM devuelve la representación $Z_{sm,b,n}$ de un valor Z en un sistema de base b sobre n dígitos más un indicador de desbordamiento.

En particular, en el computador la base es $b = 2$.

Sea $Z \in \mathbb{Z}$ y $b=2$. El operador $SM(Z, n)$ devuelve:

$$SM(Z,n) \begin{cases} \left.\begin{array}{l} BIN(|Z|, n-1) \\ S = \text{'0'} \text{ si } Z \geq 0 \\ S = \text{'1'} \text{ si } Z < 0 \end{array}\right\} = Z_{SM,n} \\ des = \text{'0'} \text{ si } TAM(|Z|, 2) \leq n-1 \\ des = \text{'1'} \text{ si } TAM(|Z|, 2) > n-1 \end{cases}$$

OPERADORES EN SM_n

Los siguientes operadores nos permiten pasar del espacio de los valores al espacio de sus representaciones en SM_n y viceversa:

- $SM(Z, n) = Z_{sm,n}$ es la representación de $Z \in$ rango (SM_n)
- $VAL\text{-}SM(Z_{sm,n}) = Z$ es el valor codificado por $Z_{SM,n} \in SM_n$

OPERACIÓN SUMA SIGNO-MAGNITUD

Sean $a, b \in$ rango (SM_n). Sea $r = a+b$ con $r \in \mathbb{Z}$. La suma depende de los signo de a

OPERANDOS	$SM(a,n) + SM(b,n) =$
$a, b > 0$	$SM(r,n)$ des $= 0$ si $r \in$ rango (SM_n) $SM(r - 2^{n-1}, n)$ des $= 1$ si $r \notin$ rango (SM_n)

$a, b > 0$	'0'	$BIN(a, n-1) + BIN(b, n-1)$
$a > 0,\ b < 0$	'0' si $\lvert a\rvert \geq \lvert b\rvert$ '1' si $\lvert a\rvert < \lvert b\rvert$	$BIN(a, n-1) - BIN(\lvert b\rvert, n-1)$ $BIN(\lvert b\rvert, n-1) - BIN(a, n-1)$
$a < 0,\ b > 0$	'0' si $\lvert a\rvert \leq \lvert b\rvert$ '1' si $\lvert a\rvert > \lvert b\rvert$	$BIN(b, n-1) - BIN(\lvert a\rvert, n-1)$ $BIN(\lvert a\rvert, n-1) - BIN(b, n-1)$
$a, b < 0$	'1'	$BIN(\lvert a\rvert, n-1) + BIN(\lvert b\rvert, n-1)$

· Definición de la resta en signo-magnitud

Sean $a, b \in$ rango (SM_n). Sea $r = a - b$ con $r \in \mathbb{Z} / a = r + b$. La resta depende de los signos de a y b:

operandos	$SM(a, n) - SM(b, n) =$
$a > 0,\ b < 0$	$SM(r, n)$ des = '0' si $r \in$ rango (SM_n) $SM(r - 2^{n-1}, n)$ des = '1' si $r \notin$ rango (SM_n)
$a, b > 0$ $a, b < 0$	$SM(r, n)$ des = '0'
$a < 0,\ b > 0$	$SM(r, n)$ des = '0' si $r \in$ rango (SM_n) $SM(r + 2^{n-1}, n)$ des = '1' si $r \notin$ rango (SM_n)

· Definición de la resta en SM_n en función del operador BIN

Sean $a, b \in$ rango (SM_n). La resta $SM(r, n)$ se puede escribir en función de BIN

operandos	S	$BIN(r, n-1)$
$a > 0,\ b < 0$	'0'	$BIN(a, n-1) + BIN(b, n-1)$
$a, b > 0$	'0' si $\lvert a\rvert \geq \lvert b\rvert$ '1' si $\lvert a\rvert < \lvert b\rvert$	$BIN(a, n-1) - BIN(\lvert b\rvert, n-1)$ $BIN(\lvert b\rvert, n-1) - BIN(a, n-1)$
$a, b < 0$	'0' si $\lvert a\rvert \leq \lvert b\rvert$ '1' si $\lvert a\rvert > \lvert b\rvert$	$BIN(b, n-1) - BIN(\lvert a\rvert, n-1)$ $BIN(\lvert a\rvert, n-1) - BIN(b, n-1)$
$a < 0,\ b > 0$	'1'	$BIN(\lvert a\rvert, n-1) + BIN(\lvert b\rvert, n-1)$

• IMPLEMENTACIÓN DE OPERADORES EN SM_n

- Por un lado se procesan los signos y por otro las magnitudes de los operandos.
- Según sea el signo de los operandos, sus magnitudes se suman o restan en BIN_{n-1} con independencia de la operación original a realizar en SM_n.
- Cuando los signos de a y b son iguales en suma u opuestos en resta el desbordamiento será el propio de la operación de suma de magnitudes en BIN_{n-1}, es decir, des = '1' si $r \notin$ rango (BIN_{n-1})
- Cuando los signos de a y b son opuestos en suma o iguales en resta no hay desbordamiento (des = '0') ya que los operandos se ordenan adecuadamente -primero el mayor- y se restan las magnitudes en BIN_{n-1}.

* DIFICULTADES

La implementación del operador SM y su aritmética de suma y resta se puede basar en el operador BIN. Sin embargo, el hecho de tener que manejar por un lado los signos y por otro lado las magnitudes utilizando unas veces la operación suma y otras la resta, con la precaución de ordenar los operandos para calcular una diferencia positiva siempre dificulta la implementación de operadores aritméticos en SM_n y limita la utilidad de este sistema de representación de enteros ($\mathbb{Z}$).

* ALTERNATIVAS

A pesar de sus limitaciones, en ocasiones puede ser conveniente utilizar la representación de enteros según el sistema signo-magnitud definido anteriormente ...

... pero además podemos utilizar los sistemas complementarios aprovechando que la limitación de espacio de representación conduce a la ARITMÉTICA MODULAR.

■ Sistemas complementarios

- La operación de resta está parcialmente definida en $\mathbb{N}$ ya que la diferencia de dos números no puede ser negativa.
- De ahí surge la necesidad de definir los números negativos y crear $\mathbb{Z}$.
- Podemos definir cada número negativo como el resultado de la resta $0-m$ con $m \in \mathbb{N}$ obteniendo el conjunto $\mathbb{Z}^- = \{-1, -2, -3, ...\}$.
- De esta manera, en $\mathbb{Z}$ sí podemos definir la resta completamente así: $a - b = d$ con $a, b, d \in \mathbb{Z} / a = d + b$.
- Se cumple entonces que si $z \in \mathbb{Z} \Rightarrow 0 - z = (-z)$ y $z + (-z) = 0$
- En un sistema de numeración posicional limitado a n dígitos $POS_{b,n} = (S, R, n)$ sabemos que dado el número A y el número B tal que $B = A + b^n$ se cumple que sus representaciones coinciden: $A_{b,n} = B_{b,n}$
- En aritmética modular se dice: $A \equiv B \pmod{b^n}$.
- Según esto, la representación de $(-Z) = 0 - Z$ será la misma que la de $Y = b^n - Z$ en sistemas complementarios $-Z_{Cb,n} = Y_{Cb,n}$.
- En general, el valor $b^n - Z$ se conoce como complemento a la base.

* CONSECUENCIAS

- Restar es equivalente a sumar el negativo. De esta manera sólo se necesita la op aritmética suma
- El negativo de un número Z se representa por el complemento a la base de dicho número.
- La representación en complemento a la base es consistente con que $+z + (-z) = 0$:

$$Z_{Cb,n} + (-Z_{Cb,n}) = (Z + b^n - Z)_{C...} = (b^n) ...$$

· Complemento a 2

- Algoritmo de cambio de signo

Sabemos que $\sum_{i=0}^{n-1} 2^i = 2^n - 1 \rightarrow 2^n = 1 + \sum_{i=0}^{n-1} 2^i$

Sea $N \in \mathbb{N}$ tal que $N = \sum_{i=0}^{n-1} d_i \cdot 2^i \rightarrow N_{2,n} = d_{n-1} \cdots d_0$

Calculamos el complemento a 2 de N:

$$2^n - N = 1 + \sum_{i=0}^{n-1} 2^i + \sum_{i=0}^{n-1} d_i \cdot 2^i = 1 + \sum_{i=0}^{n-1} (1-d_i) \cdot 2^i$$

Como $(1-d_i)$ d_i concluimos que $2^n - N = 1 + \sum_{i=0}^{n-1} \bar{d_i} \cdot 2^i$

Si N es la codificación de un Z cualquiera de un sistema de representación en complemento a 2 $(N_{2,n} = Z_{c2,n})$ entonces:

$$\bar{N}_{2,n} + 1 = -Z_{c2,n}$$

Por otra parte, es trivial que $2^n - (2^n - N) = N$ y, por tanto:

$$\overline{-Z}_{c2,n} + 1 = (\overline{N_{2,n} + 1}) + 1 = N_{2,n} = Z_{c2,n}$$

Es decir, que para conmutar el signo de una representación en C2 basta con complementar y sumar la unidad a dicha representación:

$$-Z_{c2,n} = \bar{Z}_{c2,n} + 1$$

- Definición del operador $C2(Z,n)$

Sea $Z \in \mathbb{Z}$ y $b=2$. El operador C2 devuelve:

$$C2(Z,n) = \begin{cases} BIN(2^n + Z, n) = Z_{c2,n} \\ des = '0' \text{ si } Z \in rango\ (C2n) \\ des = '1' \text{ si } Z \notin rango\ (C2n) \end{cases}$$

O también podemos dar una definición alternativa

$$C2\ (Z,n) \begin{cases} \text{si } Z \geq 0 \quad BIN(Z,n) \\ \text{si } Z < 0 \quad BIN(2^n - |Z|, n) \end{cases} = Z_{c2,n}$$
$$\quad \begin{cases} des = '0' \text{ si } Z \in rango\ (C2n) \\ des = '1' \text{ si } Z \notin rango\ (C2n) \end{cases}$$

- Operadores en $C2_n$

Los siguientes operadores nos permiten pasar del espacio de los valores al espacio de sus representaciones en C_{2n} y viceversa:

- $C_{2n}(Z,n) = Z_{c2,n}$ es la representación de $Z \in rango\ (C2_n)$
- $VAL-C2\ (Z_{c2,n}) = Z$ es el valor codificado por $Z_{c2,n} \in C2_n$

- Definición de la suma en $C2_n$

Sean $a, b \in$ rango $(C2_n)$. Sea $r = a + b$ con $r \in \mathbb{Z}$. La suma depende de los signos de a y b:

operandos	$C2(a,n) + C2(b,n) =$
$a, b > 0$	$C2(r,n)$ des = '0' si $r \in$ rango $(C2_n)$ $C2(r-2^n, n)$ des = '1' si $r \notin$ rango $(C2_n)$
$a>0, b<0$ $a<0, b>0$	$C2(r,n)$ des = '0'
	$C2(r,n)$ des = '0' si $r \in$ rango $(C2_n)$ $C2(r+2^n, n)$ des = '1' si $r \notin$ rango $(C2_n)$

- Definición de la suma en $C2_n$ en función del operador BIN

Sean $a, b \in$ rango$(C2_n)$. La suma $C2(r,n)$ se puede escribir en función del operador BIN:

operandos	BIN $(?, n)$
$a, b > 0$	BIN (a,n) + BIN (b,n)
$a>0, b<0$	BIN (a,n) + BIN $(2^n - \lvert b \rvert, n)$
$a<0, b>0$	BIN $(2^n - \lvert a \rvert, n)$ + BIN (b,n)
$a, b < 0$	BIN $(2^n - \lvert a \rvert, n)$ + BIN $(2^n - \lvert b \rvert, n)$

- Definición de la resta en $C2_n$

operandos	$C2(a,n) - C2(b,n) =$
$a>0, b<0$	$C2(r,n)$ des = '0' si $r \in$ rango $(C2_n)$ $C2(r-2^n, n)$ des = '1' si $r \notin$ rango $(C2_n)$
$a, b > 0$ $a, b < 0$	$C2(r,n)$ des = '0'
$a<0, b>0$	$C2(r,n)$ des = '0' si $r \in$ rango $(C2_n)$ $C2(r+2^n, n)$ des = '1' si $r \notin$ rango $(C2_n)$

- Definición de la resta en función del operador BIN

Sean $a, b \in$ rango $(C2_n)$. La resta $C2(r,n)$ se puede escribir en f. de BIN:

operandos	BIN $(?, n)$
$a, b > 0$	BIN (a,n) + BIN $(2^n - \lvert b \rvert, n)$
$a>0, b<0$	BIN (a,n) + BIN (b,n)
$a<0, b>0$	BIN $(2^n - \lvert a \rvert, n)$ + BIN $(2^n - \lvert b \rvert, n)$
$a, b < 0$	BIN $(2^n - \lvert a \rvert, n)$ + BIN (b,n)

* Al sumar o restar nº representados en C2 obtenemos nºs representados en el mismo sistemas. Es decir, la aritmética en $C2_n$ conserva el sistema de representación.

· COMPLEMENTO A 1

- Cuando la base es 2 el complemento restringido a la base se conoce como C1.
- Dado un $N \in \mathbb{N}$, el C1 de N es $2^n - 1 - N$.
- Calcular el complemento a 1 de N es tanto como calcular cuánto le queda a N para llegar al módulo 2^{n-1} tiene n bits a '1'.
- Así, sumar N con su complemento a 1 siempre vale el módulo que se toma como representación (adicional) del 0.
- En conclusión, determinar el C1 de un número se puede interpretar como cambiar de signo al nº.

- Algoritmo de "cambio de signo"

Sabemos que $\sum_{i=0}^{n-1} 2^i = 2^n - 1$

Sea $N \in \mathbb{N}$ tal que $N = \sum_{i=0}^{n-1} d_i \cdot 2^i \Rightarrow N_{2,n} = d_{n-1} \dots d_0$

Calculamos el C1 de N: $2^n - 1 - N = \sum_{i=0}^{n-1} 2^i + \sum_{i=0}^{n-1} d_i \cdot 2^i = \sum_{i=0}^{n-1} (1-d_i) \cdot 2^i$

Como $(1-d_i) = \overline{d_i}$ entonces $2^n - 1 - N = \sum_{i=0}^{n-1} \overline{d_i} \cdot 2i$

- Operador C1 (Z, n)

Sea $Z \in \mathbb{Z}$ y $b=2$. El operador $C1(Z,n)$ devuelve:

$$C1(Z,n) = \begin{cases} \left.\begin{array}{ll} \text{si } Z \geq 0 & BIN(Z,n) \\ \text{si } Z < 0 & BIN(2^n - 1 - |Z|, n) \end{array}\right\} = Z_{C1,n} \\ \\ des = \text{'0'} \text{ si } Z \in \text{rango } (C1\, n) \\ des = \text{'1'} \text{ si } Z \notin \text{rango } (C1\, n) \end{cases}$$

- Operadores en $C1_n$

- $C1(Z,n) = Z_{C1,n}$ es la representación de $Z \in$ rango $(C1_n)$
- $VAL-C1(Z_{C1,n}) = Z$ es el valor codificado por $Z_{C1,n} \in C1_n$

- Definición de la <u>suma</u> en $C1_n$

Sea $a, b \in$ rango $(C1_n)$. Sea $r = a+b$ con $r \in \mathbb{Z}$. La suma depende de los signos a y operandos

operandos	$C1(a,n) + C1(b,n) =$
$a, b > 0$	$C1(r,n)$ des = '0' si $r \in$ rango $(C1_n)$ $C1(r - 2^n + 1, n)$ des = '1' si $r \notin$ rango $(C1_n)$
$a>0, b<0$ $a<0, b>0$	$C1(r,n)$ des = '0'
	$C1(r,n)$ des = '0' si $r \in$ rango $(C1_n)$ $C1(r + 2^n - 1, n)$ des = '1' si $r \notin$ rango $(C1_n)$

- Definición de la suma en C1n en función del operador BIN

Sean a, b ∈ rango (C1n). La suma C1 (r, n) se puede escribir en función del operador BIN:

operandos	BIN (?, n)
$a, b > 0$	$BIN(a, n) + BIN(b, n)$
$a > 0, b < 0$	$BIN(a, n) + BIN(2^n - 1 - \|b\|, n)$
$a < 0, b > 0$	$BIN(2^n - 1 - \|a\|, n) + BIN(b, n)$
$a, b < 0$	$BIN(2^n - 1 - \|a\|, n) + BIN(2^n - 1 - \|b\|, n)$

- Definición de la resta en C1n

Sea a, b ∈ rango (C1n). Sea $r = a - b$ con $r \in \mathbb{Z}$ / $a = r + b$. La resta depende de los signos de a y b:

operandos	$C1(a, n) - C1(b, n) =$
$a > 0, b < 0$	$C1(r, n)$ des = '0' si $r \in$ rango (C1n) $C1(r - 2^n + 1, n)$ des = '1' si $r \notin$ rango (C1n)
$a, b > 0$ / $a, b < 0$	$C1(r, n)$ des = '0'
$a < 0, b > 0$	$C1(r, n)$ des = '0' si $r \in$ rango (C1n) $C1(r + 2^n - 1, n)$ des = '1' si $r \notin$ rango (C1n)

- Definición de la resta en C1n en función del operador BIN

operandos	BIN (?, n)
$a, b > 0$	$BIN(a, n) + BIN(2^n - 1 - \|b\|, n)$
$a > 0, b < 0$	$BIN(a, n) + BIN(b, n)$
$a < 0, b > 0$	$BIN(2^n - 1 - \|a\|, n) + BIN(2^n - 1 - \|b\|, n)$
$a, b < 0$	$BIN(2^n - 1 - \|a\|, n) + BIN(b, n)$

*Al sumar o restar números representados en C1 no siempre obtenemos números representados en el mismo sistema. Es decir la aritmética en C1n **NO conserva el sistema de representación.** ⟹ lo que supone una complicación a la hora de diseñar operadores aritméticos que compensa la mayor facilidad del cálculo del negativo respecto a C2.

• EXCESO 2^{n-1} ($EX\text{-}2^{n-1}{}_n$)

- Operadores

$EX\text{-}2^{n-1}(Z, n) = Z_{EX\text{-}2^{n-1}, n}$ es la representación de $Z \in$ rango ($EX\text{-}2^{n-1}{}_n$)

$VAL\text{-}C1(Z_{EX\text{-}2^{n-1}}, n) = Z$ es el valor codificado por $Z_{EX\text{-}2^{n-1}, n} \in EX\text{-}2^{n-1}{}_n$

- Definición del operador $EX\text{-}2^{n-1}(Z, n)$

Sea $Z \in \mathbb{Z}$ y $b = 2$. El operador $EX\text{-}2^{n-1}$ devuelve

$$EX\text{-}2^{n-1}(Z, n) = \begin{cases} BIN(Z + 2^{n-1}, n) = Z_{EX\text{-}2^{n-1}, n} \\ \text{des} = \text{'0'} \quad \text{si } Z \in \text{rango } (EX\text{-}2^{n-1}, n) \\ \text{des} = \text{'1'} \quad \text{si } Z \notin \text{rango } (EX\text{-}2^{n-1}, n) \end{cases}$$