Sistemas de Codificación Numérica

April 16, 2024

1 Notación

Asumiremos que un vector binario x tiene la siguiente forma:

x_{N-1}	x_{N-2}	$\cdots$	x_3	x_2	x_1	x_0

2 Codificación sin signo (US)

Valor numérico:

$$x = \sum_{i=0}^{N-1} x_i 2^i, \quad x_i \in \{0, 1\} \tag{1}$$

Rango: $[0, +2^N - 1]$

3 Codificación signo-magnitud (SM)

Valor numérico:

$$x = (-1)^{x_{N-1}} \left(\sum_{i=0}^{N-2} x_i 2^i \right), \quad x_i \in \{0, 1\} \tag{2}$$

Rango: $[-2^{N-1} - 1, +2^{N-1} - 1]$

Observar que el 0 tiene dos representaciones:

- $+0 = 000 \ldots 000$
- $-0 = 100 \ldots 000$

4 Codificación complemento a dos (2C)

Valor numérico:

$$x = -x_{N-1}2^{N-1} + \sum_{i=0}^{N-2} x_i 2^i, \quad x_i \in \{0,1\} \tag{3}$$

Rango: $[-2^{N-1}, +2^{N-1}-1]$

Teorema 1 (Extensión de signo para 2C)**.** *Sea $x \in \mathbb{Z}$ tal que tiene una representación 2C de N bits dada por la ec. (3). Entonces la representación 2C de $M > N$ estará data por*

$$x = -x_{N-1}2^{M-1} + \sum_{i=N-1}^{M-2} x_{N-1}2^i + \sum_{i=0}^{N-2} x_i 2^i, \quad x_i \in \{0,1\} \tag{4}$$

Es decir la nueva representación en $M > N$ bits se obtiene replicando el MSB de x por $M - N$ veces.

Proof. Basta con ver que si $x_{N-1} \neq 0$ entonces

$$\begin{aligned} -x_{N-1}2^{M-1} + \textstyle\sum_{i=N-1}^{M-2} x_{N-1}2^i &= -2^{M-1} + \left(2^{N-1} + 2^N + \ldots + 2^{M-3} + 2^{M-2}\right) \\ &= -2^{N-1}2^{M-N} + 2^{N-1}\left(1 + 2 + \ldots + 2^{M-N-1}\right) \\ &= -2^{N-1}2^{M-N} + 2^{N-1}\left(2^{M-N} - 1\right) \\ &= 2^{N-1}\left(-2^{M-N} + 2^{M-N} - 1\right) \\ &= -2^{N-1} \end{aligned} \tag{5}$$

Por el contrario, si $x_{N-1} = 0$ entonces

$$-x_{N-1}2^{M-1} + \sum_{i=N}^{M-2} x_{N-1}2^i = 0 \tag{6}$$

Luego,

$$x = -x_{N-1}2^{M-1} + \sum_{i=N-1}^{M-2} x_{N-1}2^i + \sum_{i=0}^{N-2} x_i 2^i = -2^{N-1} + \sum_{i=0}^{N-2} x_i 2^i \tag{7}$$

□

Ejemplo:

$x = -10_D$, $N = 5$, $M = 7$

Entonces

$x = -10_D = 10110_{2C\_N} = 1110110_{2C\_M}$

Teorema 2 (Propiedad del wrap around para 2C)**.** *Sean $x, y, z \in \mathbb{Z}$ tal que los tres tienen una representación 2C de N bits. Sea $w = x + y + z$ tal que también posee una representación 2C de N bits pero tal que $w' = x + y$ tiene una representación 2C de $N+1$ bits. Si el cálculo de la suma parcial $w' = x+y$ se hace con N bits en lugar de $N+1$, entonces el valor de w calculado como $w = w^* + z$, donde $w^* = x + y$ está calculado con N bits, resulta que el valor de w es el correcto en N bits.*

Proof. Sea $2^{N-1} \leq w' = x + y < 2^N$, entonces $w'_{N-1} = 1$ y resulta

$$w' = 2^{N-1} + \sum_{i=0}^{N-2} w'_i 2^i \tag{8}$$

Sea $z < 0$ tal que $0 < w = x + y + z < 2^{N-1} - 1$, entonces $z_{N-1} = 1$ ya que z está expresado en 2C, es decir

$$z = -2^{N-1} + \sum_{i=0}^{N-2} z_i 2^i \tag{9}$$

Por lo tanto,

$$0 < w = (x + y) + z = w' + z = \sum_{i=0}^{N-2} (w'_i + z_i) 2^i \tag{10}$$

Sea w^* la expresión 2C en N bits de w', es decir se toman los N bits de w' y se los interpreta según fuesen 2C, resultando:

$$w^* = -2^{N-1} + \sum_{i=0}^{N-2} w'_i 2^i \tag{11}$$

Sea $w^{**} = w^* + z$, entonces

$$\begin{array}{rcl} w^{**} = w^* + z & = & -2^{N-1} + \sum_{i=0}^{N-2} w'_i 2^i - 2^{N-1} + \sum_{i=0}^{N-2} z_i 2^i \\ & = & -2^N + \sum_{i=0}^{N-2} (w'_i + z_i) 2^i \end{array} \tag{12}$$

Si se elimina el bit N de w^{**} (ya que por hipótesis del teorema, la cuenta se hace con $N-1$ bits y no con N), entonces resulta:

$$w^{***} = \sum_{i=0}^{N-2} (w'_i + z_i) 2^i = w > 0 \tag{13}$$

Observar que $w^{***}_{N-1} = w^{N-1} = 0$.

El caso de $-2^N \leq w' = x + y < -2^{N-1}$ y $z > 0$ se deja como ejercicio al lector.

□

Ejemplo: Sea $N = 5$ bits

$x = 10$, $y = 9$, $z = -7$

Resultado esperado: $w = x + y + z = 10 + 9 - 7 = (10 + 9) - 7 = 19 - 7 = 12$

Se hace la suma en 5 bits:

$$\begin{aligned} w &= (x + y) + z \\ &= (10 + 9) - 7 \\ &= (10 + 9)\rfloor_{\text{mod } 32} - 7 \\ &= -13 - 7 \\ &= (-20)\rfloor_{\text{mod } 32} = 12 \end{aligned}$$

x	$=$	10	$=$	01010	
y	$=$	9	$=$	01001	
$(x+y)$	$=$	-13	$=$	10011	
z	$=$	-7	$=$	11001	
$(x+y)+z$	$=$	12	$=$	$1\,01100$	$\rightarrow$ Se descarta el bit 5.

Teorema 3 (Inversión de signo en 2C)**.** *Sea $x \in \mathbb{Z}$ expresado en 2C con N bits. Entonces $-x$ puede ser expresado también con N bits según $-x = \overline{x} + 1$.*

Proof. Sea $x = -2^{N-1}x_{N-1} + \sum_{i=0}^{N-2} x_i 2^i$. Entonces,

$$\overline{x} = -2^{N-1}\overline{x}_{N-1} + \sum_{i=0}^{N-2} \overline{x}_i 2^i \tag{14}$$

Entonces,

$$\begin{aligned} x + \overline{x} &= -2^{N-1}x_{N-1} + \textstyle\sum_{i=0}^{N-2} x_i 2^i - 2^{N-1}\overline{x}_{N-1} + \sum_{i=0}^{N-2} \overline{x}_i 2^i \\ &= -2^{N-1}(x_{N-1} + \overline{x}_{N-1}) + \textstyle\sum_{i=0}^{N-2}(x_i + \overline{x}_i)2^i \\ &= -2^{N-1}(1) + \textstyle\sum_{i=0}^{N-2}(1)2^i \\ &= -2^{N-1} + 2^{N-1} - 1 \\ &= -1 \end{aligned} \tag{15}$$

Luego, $x = -\overline{x} - 1$. Finalmente, $-x = \overline{x} + 1$.

□

5 Codificación signed digit (SD)

Valor numérico:

$$x = \sum_{i=0}^{N-1} x_i 2^i, \quad x_i \in \{-1, 0, 1\} \tag{16}$$

Observar que la representación no es unívoca.

Ejemplo: $3_{10} = 011 = 10\overline{1}$ (Notación $\overline{1} = -1$)

6 Codificación de Booth

Es una representación SD por lo cual:

$$x = \sum_{i=0}^{N-1} x_i 2^i, \quad x_i \in \{-1, 0, 1\} \tag{17}$$

Se utiliza para números signados. Observar que si la representación 2C es de N bits, entonces la representación de Booth tamibién.

6.1 Conversión 2C a Booth

Algoritmo: Sea $x_{-1} = 0$, se aplica la siguiente Tabla.

Table 1: Conversión 2C a Booth.

x_i^{2C}	x_{i-1}^{2C}	x_i^{BSD}
0	0	0
0	1	1
1	0	$\overline{1}$
1	1	0

Ejemplo:

Sea $x = -21_{10} = 101011_{2C}$. Entonces, $x_{Booth} = \overline{1}1\overline{1}10\overline{1}$.

7 Codificación canonical signed digit (CSD)

Es una representación SD por lo cual:

$$x = \sum_{i=0}^{N-1} x_i 2^i, \quad x_i \in \{-1, 0, 1\} \tag{18}$$

Si a una representación SD se le pone la restricción de que dos dígitos consecutivos al menos uno debe ser 0, entonces se obtiene la representación CSD. Se deonmina entonces como condición CSD a:

$$x_{i+1}.x_i = 0, \quad 0 \leq i \leq N-1 \tag{19}$$

Se puede demostrar entonces que:

$$P(|x_i| = 0) = \frac{1}{3} + \frac{1}{9}\left[1 - \left(\frac{-1}{2}\right)^N\right] \tag{20}$$

7.1 Conversión US a CSD

Ejemplo:

	1	1	1	0	1	0	1	1	1	0	1
	1	1	1	0	1	1	0	0	$\overline{1}$	0	1
	1	1	1	1	0	$\overline{1}$	0	0	$\overline{1}$	0	1
1	0	0	0	$\overline{1}$	0	$\overline{1}$	0	0	$\overline{1}$	0	1

Observar que si la codificación US de x requiere N bits, entonces, la codificación CSD requiere $N+1$ bits. Esto es debido a que para todo el rango de representación, algunas veces habrá q restar desde la potencia de dos immediatamente superior al valor de x.

Algoritmo: Sea $c_0 = 0$, se aplica la siguiente Tabla.

Table 2: Conversión US a CSD.

x_{i+1}^{US}	x_i^{US}	c_i	c_{i+1}	x_i^{CSD}
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	$\overline{1}$
1	0	0	0	1
1	0	1	1	0
1	1	0	1	$\overline{1}$
1	1	1	1	0

7.2 Conversión 2C a CSD

Observar que si la codificación 2C de x requiere N bits, entonces, la codificación CSD requiere también N bits debido a que el rango de representación en 2C será $[-2^{N-1}, +2^{N-1}-1]$.

Observar que si se define:

x_i	$\overline{x_i}$
0	0
1	$\overline{1}$
$\overline{1}$	1

Entonces para la conversión 2C a CSD, hay dos casos:

- x es positivo: se aplica el algoritmo de conversión US a CSD.
- x es negativo: de la representación 2C, $x = -x_{N-1}2^{N-1} + \sum_{i=0}^{N-2} x_i 2^i$, se puede representar $x = -2^{N-1} + x^+$, siendo $x^+ = \sum_{i=0}^{N-2} x_i 2^i \geq 0$. Entonces, se representa x^+ en CSD. Observemos que como x^+ tiene $N-1$ bits, su representación CSD tiene N bits. Por lo cual, el MSB de la representación CSD de x^+ puede ser 0 ó 1 ya que $x^+ \geq 0$. Entonces:

$$\begin{aligned} x &= -2^{N-1} + x^+ \\ &= -2^{N-1} + x^{+CSD} \\ &= -2^{N-1} + x_{N-1}^{CSD}.2^{N-1} + \sum_{i=0}^{N-2} x_i^{+CSD}.2^i \end{aligned} \tag{21}$$

  1. Si $x_{N-1}^{+CSD} = 1$ entonces Ec. (21) se convierte en $x = \sum_{i=0}^{N-2} x_i^{+CSD}.2^i$ y por lo tanto $x_{N-1}^{CSD} = 0$.
  2. Si $x_{N-1}^{+CSD} = 0$ entonces Ec. (21) se convierte en $x = -2^{N-1} + \sum_{i=0}^{N-2} x_i^{+CSD}.2^i$ y por lo tanto $x_{N-1}^{CSD} = \overline{1}$.

Ejemplo:

Sea $x = -3_{10} = 11101_{2C}$. Entonces estamos en el caso 1., $x_4^{2C} = 1$ y $x_4^{+CSD} = 1$, por lo cual $x_4^{CSD} = 0$ y resulta $x_{CSD} = 00\overline{1}01$.

Ejemplo:

Sea $x = -13_{10} = 10011_{2C}$. Entonces estamos en el caso 2., $x_4^{2C} = 1$ y $x_4^{+CSD} = 0$, por lo cual $x_4^{CSD} = \overline{1}$ y resulta $x_{CSD} = \overline{1}010\overline{1}$.

Finalmente, para lograr la conversión de 2C a CSD, la Tabla 7.1 puede utilizarse excepto para el caso de $x_{N-1}^C SD$ el cual debe calcularse de la siguiente forma:

- Si $x_{N-1}^{2C} = 1$ y $c_{N-1} = 0$, entonces $x_{N-1}^{CSD} = \overline{1}$
- Si $x_{N-1}^{2C} = 1$ y $c_{N-1} = 1$, entonces $x_{N-1}^{CSD} = 0$

8 Representación en punto fijo

- Sin signo, $uI.F$, donde $I, F \in \mathbb{Z}$, $N = I + F$ es la cantidad de bits de representación.

$$\begin{aligned} x &= \left(\sum_{i=0}^{N-1} x_i 2^i\right) 2^{-F} \\ x &= \sum_{i=0}^{N-1} x_i 2^{i-F} \end{aligned} \tag{22}$$

- Con signo, $sI.F$, donde $I, F \in \mathbb{Z}$, $N = I + F$ es la cantidad de bits de representación.

$$\begin{aligned} x &= \left(-x_{N-1}2^{N-1} + \sum_{i=0}^{N-2} x_i 2^i\right) 2^{-F} \\ &= -x_{N-1}2^{I-1} + \sum_{i=0}^{N-2} x_i 2^{i-F} \end{aligned} \tag{23}$$

Observar que $N = I + F > 0$, I, F no necesariamente son mayores a cero ambos a la vez.

Observar que el valor del LSB está dado por 2^{-F}.

Ejemplo: u8.5
$I = 8$, $F = 5$, $N = I + F = 13$
Cuánto representa el LSB? 2^{-F}, $F = 5$, entonces $LSB = 2^{-5} = 1/32 = 0.03125$

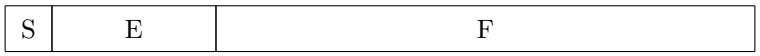
Ejemplo: $x_{u8.5} = 1011101010101$
Cuánto vale en decimal? $x_{US} = 5973_{10}$, entonces $x_{u8.5} = 5973 \times 2^{-5} = 5973/32 = 186.65625$
Rango de representación: +0 a +8191/32 = +0 a +255.96875

Ejemplo: $x_{u8.-3}$
$I = 8$, $F = -3$, $N = I + F = 5$
Cuánto representa 1 LSB? $LSB = 2^{-F} = 2^{+3} = 8$

Ejemplo: $x_{u8.-3} = 10101$
Cuánto vale en decimal? $x_{US} = 21$, entonces $x_{u8.-3} = 21_{10} \times 2^{+3} = 21 * 8 = 168$
Rango de representación: +0 a 31*8 = +0 a +248

Ejemplo: Qué sucede si $F > I$?
Pongamos un caso: $sI.F = s - 5.10$
$F = 10$, $I = -5$, $N = I + F = 5$
Sea $x_{s-5.10} = 10101$
Cuánto vale en decimal? $x_{US} = -11$, entonces $x_{s-5.10} = -11 * 2^{-10} = -11/1024 = 0.0107421875$
Cuánto representa 1 LSB? $LSB = 2^{-F} = 2^{-10} = 1/1024 = 0.0009765625$
Rango de representación: -16/1024 a 15/1024 = -0.015625 a +0.0146484375

9 Representación en punto flotante



- Número de bits del campo S: N_S
- Número de bits del campo E: N_E
- Número de bits del campo F: N_F

Ejemplo: IEEE 754 presición simple (32 bits)

- $N_S = 1$
- $N_E = 8$
- $N_F = 23$

Ejemplo: IEEE 754 presición doble (64 bits)

- $N_S = 1$
- $N_E = 11$
- $N_F = 52$

9.1 Interpretación numérica

Se define lo siguiente:

- E se interpreta como US, es decir $E = \sum_{i=0}^{N_E-1} E_i.2^i$
- $Exc = 2^{N_E-1} - 1$, Ejemplo: si $N_E = 8$, entonces $Exc = 2^{8-1} - 1 = 2^7 - 1 = 128 - 1 = 127$
- $exp = E - Exc$, Ejemplo si $E = 00100001_{US} = 33$, $exp = E - Exc = 33 - 127 = -94$
- F se interpreta como US, es decir $F = \sum_{i=0}^{N_F-1} E_i.2^i$

Interpretación numérica:

- Si los campos E y F son ambos todos '00...00', entonces se adopta que el valor numérico representado es 0.000000000000. Observar que hay dos representaciones del 0: +0 (S=0) y -0 (S=1).

- Si $Ex = 000\ldots000$ y $F \neq 00\ldots00$ entonces (denormales):

$$x = -1^{S_X} \times 2^{1-EXC} \times F/2^{N_F} \tag{24}$$

  Observar que en particular si $Fx = 000\ldots000$ then $x = -1^{S_X} \times 0$ por lo cual +/- 0 es un caso particular de un denormal.

  Observar que para el caso de denormales la mantisa vale $m = F/2^{N_F}$

- Si $Ex = 111\ldots111$ entonces (especiales):
  - Si $F_X = 000\ldots000$ entonces:

$$x = -1^{Sx} \times \infty \quad (+/\text{- Infinito}) \tag{25}$$

  - Si $F_X \neq 000\ldots000$ entonces:

$$x = -1^{Sx} \times \text{NaN} \quad (+/\text{- Not a Number}) \tag{26}$$

- Si $Ex \neq 11\ldots11$ y $Ex \neq 00\ldots00$ entonces (normales):

$$x = -1^{Sx} \times 2^{E-EXC} \times \left(1 + \frac{F}{2^{N_F}}\right) \tag{27}$$

  Observar que para los normales la mantisa vale $m = 1 + F/2^{N_F}$

10 Otros sistemas de representación numérica que no vamos a estudiar (queda el nombre para el interesado)

- Logarithmic number system (LNS)
- Residual number system (RNS)
- Universal numbers (UNUM)
  - Tipo I
  - Tipo II
  - Tipo III (Posit)
- Radix 10 Floating Point IEEE-754 2008 y IEEE-754 2019