

Sistemas de Representación de números fraccionarios: Punto Fijo

Organización de computadoras 2018

Universidad Nacional de Quilmes

Punto Fijo

El sistema numérico de punto fijo es una generalización de los sistemas enteros que permite representar números fraccionarios. En un sistema de punto fijo, se destinan una cierta cantidad de bits a la parte entera y el resto a la parte fraccionaria, considerando que existe un punto (o coma) que los separa, tal como se utiliza la coma en el sistema decimal. Sin embargo, en los sistemas binarios de punto fijo, el punto no está representado explícitamente, sino que se asume una posición determinada.

Por ejemplo, se puede construir un sistema binario sin signo con punto fijo de 8 bits, en el que 5 bits sean para la parte entera y 3 bits para la parte fraccionaria.

En relación a la notación para sistemas enteros BSS(n), SM(n) n se agrega un dato adicional que hace referencia a la cantidad de bits fraccionarios:

- BSS(n,m): Sistema de punto fijo en BSS con n bits en total, de los cuales m son fraccionarios.
- SM(n,m): Sistema de punto fijo en SM con n bits en total, de los cuales 1 es de signo, y m de parte fraccionaria. Esto implica que $n - m - 1$ corresponden a la parte entera.

Los bits fraccionarios, al igual que los enteros, tienen un peso, pero está asociado a una potencia de 2 negativa, así, el primer bit fraccionario (de izquierda a derecha) tiene un peso de 2^{-1} , el segundo de 2^{-2} , y así sucesivamente hasta el último bit fraccionario. Entonces, para interpretar una cadena en punto fijo, se debe interpretar por un lado la parte entera en el sistema correspondiente y por otro lado la parte fraccionaria.

1. Interpretación

Para **Interpretar** una cadena en punto fijo, se tienen dos mecanismos. El mecanismo *literal* (o *por partes*) requiere que se trabaje por un lado la parte entera en el sistema BSS y por otro lado la parte fraccionaria, aplicando las potencias negativas mencionadas antes.

Por ejemplo, al interpretar la cadena 10011101 se tiene una parte entera:

*mecanismo
por partes*

$$1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

y una parte fraccionaria:

$$1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

y componiendo ambas partes:

$$\begin{aligned} I_{BSS(8,3)}(10011101) &= 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ &= 16 + 2 + 1 + 0,5 + 0,125 = 19,625 \end{aligned}$$

*mecanismo
escalado*

El segundo mecanismo (mecanismo **escalado**) aprovecha la interpretación BSS, adaptando el valor que se obtiene a la posición de la **coma desplazada** (m lugares). Ese desplazamiento de coma tiene en cuenta la escala entre el sistema BSS(n) y el sistema BSS(n,m), que es justamente: 2^{-m}

$$\text{cadena} \xrightarrow{I_{bss}} \text{valor entero} \xrightarrow{\text{escala}} \text{valor fraccionario}$$

Aplicando este mecanismo al ejemplo original, la cadena 10011101 se interpreta en BSS como:

$$\begin{aligned} I_{BSS(8,3)}(10011101) &= 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 128 + 16 + 8 + 4 + 1 = 157 \end{aligned}$$

Finalmente se escala el valor entero:

$$157 \cdot 2^{-3} = \frac{157}{2^3} = \frac{157}{8} = 19,625$$

(notar que 2^{-3} es la escala del sistema, con respecto a BSS)

Pasando en limpio:

$$10011101 \xrightarrow{I_{bss}} \mathbf{157} \xrightarrow{\times 2^{-3}} \text{valor } \mathbf{19,625}$$

2. Representación y error

Tal como ocurre con la interpretación, la representación tiene dos posibles mecanismos: el *mecanismo por partes* o el mecanismo *escalado*.

*mecanismo
por partes*

El mecanismo por partes requiere partir el problema y representar por un lado la parte entera (usando la representación de BSS) y por otro lado la parte fraccionaria haciendo multiplicaciones sucesivas .

Por ejemplo, a la hora de representar el valor **3,8** en el sistema BSS(7,4):

1. Representar el **valor 3** usando los bits enteros (7 en total, 4 fraccionarios: 3 enteros), obteniendo: **011**
2. Construir la cadena fraccionaria que aproxime el valor **0,8** multiplicando por 2 tantas veces como bits fraccionarios se tienen, y en cada multiplicación separar la parte entera del resultado, que es lo que corresponde a cada nuevo bit. Veamos el ejemplo:

- a) $0,8 * 2 = 1,6 \Rightarrow b_{-1} = 1$
- b) $0,6 * 2 = 1,2 \Rightarrow b_{-2} = 1$
- c) $0,2 * 2 = 0,4 \Rightarrow b_{-3} = 0$
- d) $0,4 * 2 = 0,8 \Rightarrow b_{-4} = 0$

Este mecanismo establece un criterio de corte basado en la cantidad de bits disponibles, pero como en el caso del sistema decimal, esto puede obtener una cadena que no aproxima de la mejor manera. Por ejemplo, si se tiene sistema decimal con sólo 2 dígitos fraccionarios para aproximar el valor 0,009, existe una gran diferencia entre aproximarlos con 0,00 que con 0,01. En el último caso se llevo a cabo un **redondeo** mientras que en el primer caso se truncó la cadena, ocasionando mas pérdida.

redondeo

Para evitar que la cadena de bits quede truncada, se hace una paso mas (m+1 pasos en total) a los fines del redondeo, y si el bit obtenido en este último paso es 1 entonces se suma 1 a la cadena de bits fraccionaria. siguiendo el ejemplo:

$$0,8 * 2 = 1,6 \Rightarrow b_{-5} = 1$$

Entonces:

$$1100 + 1 = 1101$$

3. Finalmente ambas subcadenas se componen en la cadena: 011 1101

Una vez finalizada la representación es oportuno controlar el trabajo realizado aplicando la interpretación (cualquiera de los mecanismos) sobre la cadena obtenida. En el ejemplo:

$$I_{bss(7,4)}(0111101) = 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-4} = 2 + 1 + 0,5 + 0,25 + 0,0625 = 3,8125$$

Sin embargo, se quería representar el valor **3,8**. Esto ocurre porque no todos los números del rango son representables en el sistema, por lo tanto puede haber un **error de representación**, que es el valor absoluto de la diferencia entre el número que se deseaba representar y el número que efectivamente se logró representar. En este ejemplo, la diferencia es:

error de representación

$$|3,8 - 3,8125| = 0,0125$$

El **mecanismo escalado** requiere llevar el valor a representar a un entero para luego aplicar la representación de BSS (R_{bss}). Dicho entero se obtiene al escalar y redondear el valor original, teniendo en cuenta, como en la interpretación, que el valor de escala es 2^{-m} .

Mecanismo escalado

$$\text{valor fraccionario} \xrightarrow{\text{escala}} \text{valor entero} \xrightarrow{R_{bss}} \text{cadena}$$

Siguiendo el ejemplo, representar el valor **3,14** en BSS(7,4)

1. Escalar: $3,8 * 2^4 = 60,8$
2. Redondeo: $60,8 \approx 61$

$$3. R_{bss(7)}(61) = 0111101$$

$$\text{valor } 3,14 \xrightarrow{\times 4} 60,8 \xrightarrow{\text{redondeo}} 61 \xrightarrow{R_{bss}} 0111101$$

Es posible observar que con este mecanismo se obtuvo la misma cadena que antes.

3. Rango y Resolución

Al igual que en los sistemas enteros (BSS, SM y CA2), el rango en Punto Fijo está determinado por el intervalo de números representables. Por ejemplo, el rango del sistema BSS(2,1) está dado por:

$$\text{Mínimo} : I_{bss(2,1)}(00) = 0$$

$$\text{Máximo} : I_{bss(2,1)}(11) = 2^0 + 2^{-1} = 1 + 0,5 = 1,5$$

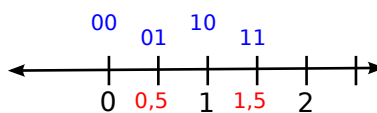
Y el del sistema sm (4,2) por:

$$\text{Mínimo} : I_{sm(4,2)}(1111) = -(2^0 + 2^{-1} + 2^{-2}) = -(1 + 0,5 + 0,5) = -1,75$$

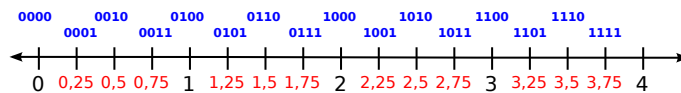
$$\text{Máximo} : I_{sm(4,2)}(0111) = 2^0 + 2^{-1} + 2^{-2} = 1 + 0,5 + 0,5 = 1,75$$

Sin embargo, el rango se refiere a todos los números representables en el intervalo. En los sistemas enteros esto es trivial (todos los enteros del intervalo), pero en punto fijo, para determinar exactamente qué números están representados dentro del intervalo es necesario el concepto de **resolución**: *la resolución es la distancia entre dos números representables consecutivos*.

Por ejemplo, en el sistema bss(2,1) la resolución es 0,5 y por lo tanto los números representables son:



Y en el sistema bss (4,2) la resolución es 0,25, y los números representables son:



Referencias

- [1] Williams Stallings, *Computer Organization and Architecture*, octava edición, Editorial Prentice Hall, 2010. **Capítulo 8: Aritmética del Computador**, subcapítulo 8.2: Representación en coma fija.