Tema 1 - Representación de números de punto flotante

Curso de Física Computacional

M. en C. Gustavo Contreras Mayén

30 de agosto de 2012

Contenido

Contenido

¿Qué es lo que pasa?

Realiza la siguiente operación en Python:

```
>>> 1 - 0.9 - 0.1
```

por qué el resultado que esperamos no es cero?

Considera el siguiente código

```
suma = 1
for i in range(1,1000001):
    suma = suma+0.0000001
print suma
```

¿Cuál es el resultado que nos devuelve el programa?

¿En dónde está el error?

Considera el siguiente código

```
suma = 1
for i in range(1,1000001):
    suma = suma+0.0000001
print suma
```

¿Cuál es el resultado que nos devuelve el programa?

¿En dónde está el error?

Para dar respuesta a las preguntas, tendremos que ahondar en la manera en que se representan los números en una computadora.

El Estándar IEE 754

El estándar IEEE 754 establece dos formatos básicos para representar a los números reales en la computadora:

- de precisión simple.
- de precisión doble.

Precisión simple

Cuando se representa un número real en precisión simple, se usan 32 bits (4 bytes) de la siguiente manera:

1. 1 bit para el signo (**s**) de número.
2. 8 bits para el exponente (**exp**)
3. 23 bits para la mantisa (**m**) que se distribuyen de la siguiente forma:

31	30 ⋯ 23	22 ⋯ 0
s	**exp**	**m**
signo	exponente	mantisa

El exponente se suele representar en notación *Exceso a* 2^{n-1}, mientras que para la mantisa, normalmente se usa el Signo y Magnitud.

La mantisa se suele normalizar colocando el punto decimal a la derecha del bit más significativo.

¿Qué es el Exceso a 2^{n-1}?

En Exceso a 2^{n-1} tenemos que si se dispone de **n** bits para representar a un número entero **N** positivo o negativo, dicho número se representa como $N + 2^{n-1}$, por lo que:

$$N_{Ex} = N + 2^{n+1}$$

Ejemplo

En Exceso a 2^{n-1} para $n = 16$, el número positivo 9503_{10} se presenta como:

$$
\begin{aligned}
9503_{10} =& (9503_{10} + (2^{16-1})_{10}) \\
=& (9503_{10} + (2^{15})_{10}) \\
=& (9503_{10} + 32768_{10}) \\
=& (42271_{10}) \\
=& (1010010100011111_2) \\
=& 1010010100011111_{\text{Exp a32768}}
\end{aligned}
$$

Ejemplo

En Exceso a 2^{n-1} para $n = 16$, el número positivo 9503_{10} se presenta como:

$$
\begin{aligned}
9503_{10} =& (9503_{10} + (2^{16-1})_{10}) \\
=& (9503_{10} + (2^{15})_{10}) \\
=& (9503_{10} + 32768_{10}) \\
=& (42271_{10}) \\
=& (1010010100011111_2) \\
=& 1010010100011111_{\text{Exp a32768}}
\end{aligned}
$$

Ejemplo

En Exceso a 2^{n-1} para $n = 16$, el número positivo 9503_{10} se presenta como:

$$
\begin{aligned}
9503_{10} &= (9503_{10} + (2^{16-1})_{10}) \\
&= (9503_{10} + (2^{15})_{10}) \\
&= (9503_{10} + 32768_{10}) \\
&= (42271_{10}) \\
&= (1010010100011111_2) \\
&= 1010010100011111_{\text{Exp a32768}}
\end{aligned}
$$

Ejemplo

En Exceso a 2^{n-1} para $n = 16$, el número positivo 9503_{10} se presenta como:

$$
\begin{aligned}
9503_{10} &= (9503_{10} + (2^{16-1})_{10}) \\
&= (9503_{10} + (2^{15})_{10}) \\
&= (9503_{10} + 32768_{10}) \\
&= (42271_{10}) \\
&= (1010010100011111_2) \\
&= 1010010100011111_{\text{Exp a32768}}
\end{aligned}
$$

Ejemplo

En Exceso a 2^{n-1} para $n = 16$, el número positivo 9503_{10} se presenta como:

$$
\begin{aligned}
9503_{10} =& (9503_{10} + (2^{16-1})_{10}) \\
=& (9503_{10} + (2^{15})_{10}) \\
=& (9503_{10} + 32768_{10}) \\
=& (42271_{10}) \\
=& (1010010100011111_2) \\
=& 1010010100011111_{\text{Exp a32768}}
\end{aligned}
$$

Ejemplo

En Exceso a 2^{n-1} para $n = 16$, el número positivo 9503_{10} se presenta como:

$$\begin{aligned}
9503_{10} &= (9503_{10} + (2^{16-1})_{10}) \\
&= (9503_{10} + (2^{15})_{10}) \\
&= (9503_{10} + 32768_{10}) \\
&= (42271_{10}) \\
&= (1010010100011111_2) \\
&= 1010010100011111_{\text{Exp a}32768}
\end{aligned}$$

Signo y Magnitud

En la representación de un número entero en *Signo y Magnitud*, de los **n** bits de dicha representación, el más significativo representa el signo de la misma representación, por lo que se llama *bit de signo*.

El resto de los bits representan la magnitud.

$$N_{SM} = a_{n-1}a_{n-2}\ldots a_1a_0$$

donde a_{n-1} representa el signo del número y el resto de los bits, la magnitud del mismo:

Representación de Signo y Magnitud

a_{n-1}	a_{n-2}	a_{n-3}		$\cdots$	a_1	a_0

bit de signo — bits de la magnitud

Ejemplo

Para escribir el número

$$101110.01010111010000111110000011111100010011_2$$

en el estándar IEEE 754 con precisión simple, exponente en Exceso a 2^{n-1} y mantisa en Signo y Magnitud, lo primero que que hay que hacer es normalizarlo:

$$1.0111001010111010000111110000011111100010011_2 \times 2^5$$

El exponente en Exceso a 2^{n-1} será:

$$\begin{aligned}
5_{10} + (2^{8-1} - 1)_{10} &= 5_{10} + (128 - 1)_{10} = \\
&= 132_{10} = \\
&= 10000100_{\text{Exp. a127}}
\end{aligned}$$

Tomamos de la mantisa los 23 bits más significativos:

$$1.01111001010111000000111$$

El resto de los bits no se puede representar ya que no caben en la mantisa.

Cuando la mantisa se normaliza dejando el punto decimal a la derecha del bit más significativo, dicho bit siempre vale **1**, por lo que se puede prescindir de él y tomar agregar en su lugar otro bit, por lo que aumenta la precisión del número representado:

$$01111001010111000000111 1$$

Tomamos de la mantisa los 23 bits más significativos:

$$1.01111001010111000000111$$

El resto de los bits no se puede representar ya que no caben en la mantisa.

Cuando la mantisa se normaliza dejando el punto decimal a la derecha del bit más significativo, dicho bit siempre vale **1**, por lo que se puede prescindir de él y tomar agregar en su lugar otro bit, por lo que aumenta la precisión del número representado:

$$01110010101110000001111$$

Al bit omitido se le llama *bit implícito*. Por otra lado, el bit del signo vale $\mathbf{0}$, ya que es un número posi-
tivo, por tanto, el número se puede representar como:

31	30 ⋯ 23	22 ⋯ 0
0	**10000100**	**01110010101110000001111**
signo	exponente	mantisa

Precisión doble

En precisión doble, para escribir un número real, se utilizan 64 bits (8 bytes):

1. 1 bit para el signo (**s**) del número.
2. 11 bits para el exponente (**exp**)
3. 52 bits para la mantisa (**m**).

63	62 ⋯ 52	51 ⋯ 0
s	**exp**	**m**
signo	exponente	mantisa

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$19 \,\underline{\big|\; 2 \quad}$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$\begin{array}{r|l} 19 & \ 2 \\ \hline & \ 9 \end{array}$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$\begin{array}{r|l} 19 & 2 \\ \hline \mathbf{1} & 9 \end{array}$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$\begin{array}{rr|l} & 19 & 2 \\ \cline{3-3} a_0 = 1 \longleftarrow & \mathbf{1} & 9 \end{array}$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{rr|l}
 & 19 & 2 \\ \cline{3-3}
a_0 = 1 \longleftarrow & \mathbf{1} & 9 \;\big|\; \\ \cline{3-3}
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{rrr|l}
 & 19 & \multicolumn{1}{|l}{2} & \\
\cline{3-4}
a_0 = 1 \longleftarrow & \mathbf{1} & 9 & 2 \\
\cline{4-4}
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{rrr|l}
 & & 19 & \multicolumn{1}{|l}{2} \\ \cline{4-4}
a_0 = 1 \longleftarrow & & \mathbf{1} & 9 \;\; \begin{array}{|l}2 \\ \hline \end{array} \\
 & & & \;\;\;\; 4
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$\begin{array}{rr|ll}
 & 19 & 2 & \\ \cline{3-4}
a_0 = 1 \longleftarrow & \mathbf{1} & 9 & \multicolumn{1}{|l}{2} \\ \cline{4-4}
 & & \mathbf{1} & 4
\end{array}$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{rcl}
 & & 19 \;\big|\; 2 \\
a_0 = 1 & \longleftarrow & \mathbf{1} \quad 9 \;\big|\; 2 \\
a_1 = 1 & \longleftarrow & \mathbf{1} \quad 4
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$\begin{array}{lrrrr}
 & 19 & \multicolumn{1}{|c}{2} & & \\ \cline{3-4}
a_0 = 1 \longleftarrow & \mathbf{1} & 9 & \multicolumn{1}{|c}{2} & \\ \cline{4-5}
a_1 = 1 \longleftarrow & & \mathbf{1} & 4 & \multicolumn{1}{|c}{} \\ \cline{5-5}
\end{array}$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{lcl|ll}
 & & 19 & \underline{\;2\;} & \\
a_0 = 1 & \longleftarrow & \mathbf{1} & \;9 & \big|\underline{\;2\;} \\
a_1 = 1 & \longleftarrow & & \mathbf{1} & \;4\;\big|\underline{\;2\;}
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{lccccc}
 & & 19 & 2 & & \\
a_0 = 1 \longleftarrow & \mathbf{1} & & 9 & 2 & \\
a_1 = 1 \longleftarrow & & & \mathbf{1} & 4 & 2 \\
 & & & & & 2
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{lcccccc}
 & & 19 & \multicolumn{1}{|c}{2} & & & \\
\cline{4-5}
a_0 = 1 \longleftarrow & & \mathbf{1} & 9 & \multicolumn{1}{|c}{2} & & \\
\cline{5-6}
a_1 = 1 \longleftarrow & & & \mathbf{1} & 4 & \multicolumn{1}{|c}{2} & \\
\cline{6-7}
 & & & & \mathbf{0} & 2 &
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{rcl|l}
 & & 19 & 2 \\
\hline
a_0 = 1 \longleftarrow & \mathbf{1} & 9 & 2 \\
a_1 = 1 \longleftarrow & & \mathbf{1} \quad 4 & 2 \\
a_2 = 0 \longleftarrow & & \mathbf{0} & 2
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{lccccccc}
 & & & 19 & \multicolumn{1}{|c}{2} & & & \\ \cline{5-6}
a_0 = 1 \longleftarrow & \mathbf{1} & & & 9 & \multicolumn{1}{|c}{2} & & \\ \cline{6-7}
a_1 = 1 \longleftarrow & & \mathbf{1} & & & 4 & \multicolumn{1}{|c}{2} & \\ \cline{7-8}
a_2 = 0 \longleftarrow & & & \mathbf{0} & & & 2 & \multicolumn{1}{|c}{} \\ \cline{8-8}
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{lcccccc}
 & & 19 & \multicolumn{1}{|c}{2} & & & \\ \cline{4-5}
a_0 = 1 \longleftarrow & & \mathbf{1} & 9 & \multicolumn{1}{|c}{2} & & \\ \cline{5-6}
a_1 = 1 \longleftarrow & & & \mathbf{1} & 4 & \multicolumn{1}{|c}{2} & \\ \cline{6-7}
a_2 = 0 \longleftarrow & & & & \mathbf{0} & 2 & \multicolumn{1}{|c}{2} \\ \cline{7-7}
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{lllllll}
 & & 19 & \multicolumn{1}{|l}{2} & & & \\ \cline{4-5}
a_0 = 1 \longleftarrow & & \mathbf{1} & 9 & \multicolumn{1}{|l}{2} & & \\ \cline{5-6}
a_1 = 1 \longleftarrow & & & \mathbf{1} & 4 & \multicolumn{1}{|l}{2} & \\ \cline{6-7}
a_2 = 0 \longleftarrow & & & & \mathbf{0} & 2 & \multicolumn{1}{|l}{2} \\ \cline{7-7}
 & & & & & & \mathbf{1}
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{llllllll}
 & & 19 & \multicolumn{1}{|l}{2} & & & & \\ \cline{4-5}
a_0 = 1 \longleftarrow & \mathbf{1} & & 9 & \multicolumn{1}{|l}{2} & & & \\ \cline{5-6}
a_1 = 1 \longleftarrow & & \mathbf{1} & & 4 & \multicolumn{1}{|l}{2} & & \\ \cline{6-7}
a_2 = 0 \longleftarrow & & & \mathbf{0} & & 2 & \multicolumn{1}{|l}{2} & \\ \cline{7-8}
 & & & & & \mathbf{0} & & \mathbf{1}
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{lccccc}
 & 19 & 2 & & & \\
a_0 = 1 \longleftarrow & \mathbf{1} & 9 & 2 & & \\
a_1 = 1 \longleftarrow & & \mathbf{1} & 4 & 2 & \\
a_2 = 0 \longleftarrow & & & \mathbf{0} & 2 & 2 \\
a_3 = 0 \longleftarrow & & & & \mathbf{0} & \mathbf{1}
\end{array}
$$

Ejemplo

Se quiere expresar el número 19.5625_{10} en el estándar IEEE 754 con precisión doble, por lo que tendremos que realizar:

1. Cambiar 19.5625_{10} a base 2, iniciando con la parte entera:

$$
\begin{array}{llllll}
 & & 19 & \multicolumn{1}{|l}{2} & & \\ \cline{4-4}
a_0 = 1 \longleftarrow & \mathbf{1} & 9 & \multicolumn{1}{|l}{2} & & \\ \cline{4-4}
a_1 = 1 \longleftarrow & \mathbf{1} & 4 & \multicolumn{1}{|l}{2} & & \\ \cline{4-4}
a_2 = 0 \longleftarrow & \mathbf{0} & 2 & \multicolumn{1}{|l}{2} & & \\ \cline{4-4}
a_3 = 0 \longleftarrow & \mathbf{0} & \mathbf{1} & \longrightarrow a_4 = 1 & &
\end{array}
$$

2. Ahora la parte fraccionaria

$$0.5625 \times 2 = \mathbf{1}.125$$

2. Ahora la parte fraccionaria

$$0.5625 \times 2 = \mathbf{1}.125 \longrightarrow a_{.1} = 1$$

2. Ahora la parte fraccionaria

$$
\begin{aligned}
0.5625 \times 2 &= \mathbf{1}.125 \longrightarrow a_{.1} = 1 \\
0.125 \times 2 &= \mathbf{0}.25
\end{aligned}
$$

2. Ahora la parte fraccionaria

$$
\begin{aligned}
0.5625 \times 2 &= \mathbf{1}.125 \longrightarrow a_{.1} = 1 \\
0.125 \times 2 &= \mathbf{0}.25 \quad \longrightarrow a_{.2} = 0
\end{aligned}
$$

2. Ahora la parte fraccionaria

$$\begin{aligned} 0.5625 \times 2 &= \mathbf{1}.125 \longrightarrow a_{.1} = 1 \\ 0.125 \times 2 &= \mathbf{0}.25 \longrightarrow a_{.2} = 0 \\ 0.25 \times 2 &= \mathbf{0}.5 \end{aligned}$$

2. Ahora la parte fraccionaria

$$\begin{aligned} 0.5625 \times 2 &= \mathbf{1}.125 \longrightarrow a_{.1} = 1 \\ 0.125 \times 2 &= \mathbf{0}.25 \longrightarrow a_{.2} = 0 \\ 0.25 \times 2 &= \mathbf{0}.5 \longrightarrow a_{.3} = 0 \end{aligned}$$

2. Ahora la parte fraccionaria

$$\begin{aligned}
0.5625 \times 2 &= \mathbf{1}.125 \longrightarrow a_{.1} = 1 \\
0.125 \times 2 &= \mathbf{0}.25 \longrightarrow a_{.2} = 0 \\
0.25 \times 2 &= \mathbf{0}.5 \longrightarrow a_{.3} = 0 \\
0.5 \times 2 &= \mathbf{1}
\end{aligned}$$

2. Ahora la parte fraccionaria

$$\begin{aligned}
0.5625 \times 2 &= \mathbf{1}.125 \longrightarrow a_{.1} = 1 \\
0.125 \times 2 &= \mathbf{0}.25 \longrightarrow a_{.2} = 0 \\
0.25 \times 2 &= \mathbf{0}.5 \longrightarrow a_{.3} = 0 \\
0.5 \times 2 &= \mathbf{1} \longrightarrow a_{.4} = 1
\end{aligned}$$

De modo que el número queda expresado como:

$$19.5625_{10} = 10011.1001_2$$

3. Se normaliza el número en binario que se obtuvo, dejando el punto decimal a la derecha del bit más significativo:

$$10011.1001_2 = 1.00111001 \times 2^4$$

De modo que el número queda expresado como:

$$19.5625_{10} = 10011.1001_2$$

3. Se normaliza el número en binario que se obtuvo, dejando el punto decimal a la derecha del bit más significativo:

$$10011.1001_2 = 1.00111001 \times 2^4$$

4. Se escribe el exponente en Exceso a $2^{n-1}-1$

$$\begin{aligned} 4_{10} + (2^{11-1} - 1)_{10} =& 4_{10} + (2^{10} - 1)_{10} \\ =& 4_{10} + (1024 - 1)_{10} \\ =& 1027_{10} \\ =& 10000000011_{\text{Exp. a 1023}} \end{aligned}$$

5. Se deja la mantisa utilizando el bit implícito: se eligen los ocho bits que están a la derecha de la coma (**00111001**) y el resto de la mantisa se rellena con ceros:

$$001110010000000\ldots000000$$

El número expresado en el estándar IEEE 754 con precisión doble es:

63	62 ··· 52	51 ··· 0
0	**10000000011**	**0011100100000000000000 . . . 000000000**
signo	exponente	mantisa

Casos especiales del IEEE 754

Tanto en los tipos de datos de precisión simple como de precisión doble, existen algunos casos especiales que dependen de los valores del signo, del exponente y de la mantisa.

Signo (s)	Exponente (exp)	Mantisa (m)	Significado
Positivo (0)	Todos unos (111...11)	Todos ceros (000...00)	Más infinito ($+\infty$)
Negativo (1)	Todos unos (111...11)	Todos ceros (000...00)	Menos infinito ($-\infty$)
0 ó 1	Todos unos (111...11)	Distinta de todo ceros	No es un número (Not a Number, **NaN**)
0 ó 1	Todos ceros (000...00)	Todos ceros (000...00)	Representa al cero (0)
0 ó 1	Todos ceros (000...00)	Distinta de todo ceros	Número muy pequeño cercano al cero

¿Y si la representación del número binario es periódica?

Hagamos la conversión de 0.4 a binario:

$$0.4 \times 2 = 0.8 \quad \rightarrow a_{.1} = 0$$
$$0.8 \times 2 = 1.60 \quad \rightarrow a_{.2} = 1$$
$$0.6 \times 2 = 1.20 \quad \rightarrow a_{.3} = 1$$
$$0.2 \times 2 = 0.4 \quad \rightarrow a_{.4} = 0$$
$$0.4 \times 2 = 0.8 \quad \rightarrow a_{.5} = 0$$
$$0.8 \times 2 = 1.60 \quad \rightarrow a_{.6} = 1$$
$$0.6 \times 2 = 1.20 \quad \rightarrow a_{.7} = 1$$
$$0.2 \times 2 = 0.4 \quad \rightarrow a_{.8} = 0$$
$$0.4 \times 2 = 0.8 \quad \rightarrow a_{.9} = 0$$

$$0.8 \times 2 = 1.60 \quad \rightarrow a_{.10} = 1$$
$$0.6 \times 2 = 1.20 \quad \rightarrow a_{.11} = 1$$
$$0.2 \times 2 = 0.4 \quad \rightarrow a_{.12} = 0$$
$$0.4 \times 2 = 0.8 \quad \rightarrow a_{.13} = 0$$

Como vemos, al expresar el valor de 0.4 a binario, tenemos un número periódico, en donde 0011 se repite indefinidamente. En el ejemplo se llegó hasta 0.0110011001100 que es igual a 39.990234375.

Cuanto más se repita el cálculo, más nos acercamos a 0.40, pero como tenemos un número binario periódico, no importa la cantidad de dígitos fraccionarios, siempre se acercará a 0.40, por lo que se redondea luego de cierto número de dígitos.

Veamos la conversión a decimal:

$$\begin{aligned}0.0110011001100 &= (0 \times 2^{0}) + (0 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3}) + \\ &+ (0 \times 2^{-4}) + (0 \times 2^{-5}) + (1 \times 2^{-6}) + (1 \times 2^{-7}) + \\ &+ (0 \times 2^{-8}) + (0 \times 2^{-9}) + (1 \times 2^{-10}) + (1 \times 2^{-11}) + \\ &+ (0 \times 2^{-12}) + (0 \times 2^{-13}) = \\ &= 0.39990234375\end{aligned}$$

Que se puede redondear a 0.4