Práctica 1: Representación de números

Números reales

Organización del Computador I
DC - UBA

Verano 2018

Menú del día (primera parte)

Hoy vamos a ver:

- Representaciones de números *con coma*
- Cambios de base
- Punto fijo
- Punto flotante
- Underflow
- Representación normalizada
- Representación IEEE 754

Sistema decimal *con coma*

Sistema decimal *con coma*

- **Números enteros**
  la posición de cada símbolo refiere a una potencia de 10

Sistema decimal *con coma*

- **Números enteros**
  la posición de cada símbolo refiere a una potencia de 10
- **Números fraccionarios**
  pasa lo mismo

Sistema decimal *con coma*

- **Números enteros**
  la posición de cada símbolo refiere a una potencia de 10
- **Números fraccionarios**
  pasa lo mismo

$$7 \quad 4 \quad , \quad 3 \quad 1 \quad 2$$

Sistema decimal *con coma*

- **Números enteros**
  la posición de cada símbolo refiere a una potencia de 10
- **Números fraccionarios**
  pasa lo mismo

$$\begin{array}{cccccccc} 10^1 & 10^0 & & & & & & \\ 7 & 4 & , & 3 & & 1 & & 2 \end{array}$$

Sistema decimal *con coma*

- **Números enteros**
  la posición de cada símbolo refiere a una potencia de 10
- **Números fraccionarios**
  pasa lo mismo

$$\begin{array}{cccccc} 10^{1} & 10^{0} & & 10^{-1} & 10^{-2} & 10^{-3} \\ 7 & 4 & , & 3 & 1 & 2 \end{array}$$

Sistema decimal *con coma*

- **Números enteros**
  la posición de cada símbolo refiere a una potencia de 10
- **Números fraccionarios**
  pasa lo mismo

$$\begin{array}{cccccc} 10^{1} & 10^{0} & & 10^{-1} & 10^{-2} & 10^{-3} \\ 7 & 4 & , & 3 & 1 & 2 \end{array}$$

Sistema decimal *con coma*

- **Números enteros**
  la posición de cada símbolo refiere a una potencia de 10
- **Números fraccionarios**
  pasa lo mismo

$$\begin{array}{cccccccccc} 10^1 & 10^0 & & 10^{-1} & 10^{-2} & 10^{-3} \\ 7 & 4 & , & 3 & 1 & 2 \\ 7 \times 10 & 4 \times 1 & & 3 \times 0,1 & 1 \times 0,01 & 2 \times 0,001 \end{array}$$

En general

Si b es la base,

- utilizamos b símbolos para representar números entre 0 y $b-1$
- interpretamos los numerales como:

$$(\mathtt{a}_n\mathtt{a}_{n-1}\cdots\mathtt{a}_0\ ,\ \mathtt{a}_{-1}\cdots\mathtt{a}_{-k+1}\mathtt{a}_{-k})_b = \sum_{i=-k}^{n} \mathtt{a}_i \cdot b^i$$

Ejercicios

Expresemos en base 10 (base decimal)

- $(327,752)_8$

Ejercicios

Expresemos en base 10 (base decimal)

- $(327,752)_8$
  - $(327{,}752)_8 = (3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 7 \times 8^{-1} + 5 \times 8^{-2} + 2 \times 8^{-3})_{10}$
  - $(327{,}752)_8 = (215{,}9570313)_{10}$

Ejercicios

Expresemos en base 10 (base decimal)

- $(327,752)_8$
  - $(327{,}752)_8 = (3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 7 \times 8^{-1} + 5 \times 8^{-2} + 2 \times 8^{-3})_{10}$
  - $(327{,}752)_8 = (215{,}9570313)_{10}$

- $(10,001)_2$

Ejercicios

Expresemos en base 10 (base decimal)

- $(327,752)_8$
  - $(327{,}752)_8 = (3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 7 \times 8^{-1} + 5 \times 8^{-2} + 2 \times 8^{-3})_{10}$
  - $(327{,}752)_8 = (215{,}9570313)_{10}$

- $(10,001)_2$
  - $(10{,}001)_2 = (1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3})_{10}$
  - $(10{,}001)_2 = (2{,}125)_{10}$

Ejercicios

Expresemos en base 10 (base decimal)

- $(327,752)_8$
  - $(327{,}752)_8 = (3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 7 \times 8^{-1} + 5 \times 8^{-2} + 2 \times 8^{-3})_{10}$
  - $(327{,}752)_8 = (215{,}9570313)_{10}$
- $(10,001)_2$
  - $(10{,}001)_2 = (1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3})_{10}$
  - $(10{,}001)_2 = (2{,}125)_{10}$
- $(10,001)_{11}$

Ejercicios

Expresemos en base 10 (base decimal)

- $(327,752)_8$
  - $(327{,}752)_8 = (3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 7 \times 8^{-1} + 5 \times 8^{-2} + 2 \times 8^{-3})_{10}$
  - $(327{,}752)_8 = (215{,}9570313)_{10}$
- $(10,001)_2$
  - $(10{,}001)_2 = (1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3})_{10}$
  - $(10{,}001)_2 = (2{,}125)_{10}$
- $(10,001)_{11}$
  - $(10{,}001)_{11} = (1 \times 11^1 + 0 \times 11^0 + 0 \times 11^{-1} + 0 \times 11^{-2} + 1 \times 11^{-3})_{10}$
  - $(10{,}001)_{11} = (11{,}000751314800901577 76)_{10}$

Cambiando de base

¿Cómo hacemos para escribir el $(2{,}375)_{10}$ en binario?

$$(2{,}375)_{10} = (2)_{10} + (0{,}375)_{10}$$

Entonces vamos por partes

Parte entera: lo visto hasta ahora

Parte fraccionaria: ¿qué hacemos?

Un algoritmo

Quiero expresar $(0{,}375)_{10}$ en base 2

Un algoritmo

Quiero expresar $(0{,}375)_{10}$ en base 2

1. Multiplico el número por la base

Un algoritmo

Quiero expresar $(0{,}375)_{10}$ en base 2

1. Multiplico el número por la base

   - $0{,}375 \times 2 = 0{,}75$

Un algoritmo

Quiero expresar $(0{,}375)_{10}$ en base 2

1. Multiplico el número por la base
   - $0{,}375 \times 2 = 0{,}75$
2. Multiplico la **parte fraccionaria** del resultado por la base

Un algoritmo

Quiero expresar $(0{,}375)_{10}$ en base 2

1. Multiplico el número por la base
   - $0{,}375 \times 2 = 0{,}75$
2. Multiplico la **parte fraccionaria** del resultado por la base
   - $0{,}75 \times 2 = 1{,}5$

Un algoritmo

Quiero expresar $(0{,}375)_{10}$ en base 2

1. Multiplico el número por la base
   - $0{,}375 \times 2 = 0{,}75$
2. Multiplico la **parte fraccionaria** del resultado por la base
   - $0{,}75 \times 2 = 1{,}5$
3. Si el resultado **no** tiene parte fraccionaria, termino
   Si no, repito el paso 2

Un algoritmo

Quiero expresar $(0{,}375)_{10}$ en base 2

1. Multiplico el número por la base
   - $0{,}375 \times 2 = 0{,}75$
2. Multiplico la **parte fraccionaria** del resultado por la base
   - $0{,}75 \times 2 = 1{,}5$
3. Si el resultado **no** tiene parte fraccionaria, termino
   Si no, repito el paso 2
   - En este caso tengo que repetir el resultado tiene parte fraccionaria

Un algoritmo

Quiero expresar $(0{,}375)_{10}$ en base 2

1. Multiplico el número por la base
   - $0{,}375 \times 2 = 0{,}75$
2. Multiplico la **parte fraccionaria** del resultado por la base
   - $0{,}75 \times 2 = 1{,}5$
3. Si el resultado **no** tiene parte fraccionaria, termino
   Si no, repito el paso 2
   - En este caso tengo que repetir el resultado tiene parte fraccionaria
   - $0{,}5 \times 2 = 1$

Un algoritmo

Quiero expresar $(0{,}375)_{10}$ en base 2

1. Multiplico el número por la base
   - $0{,}375 \times 2 = 0{,}75$
2. Multiplico la **parte fraccionaria** del resultado por la base
   - $0{,}75 \times 2 = 1{,}5$
3. Si el resultado **no** tiene parte fraccionaria, termino
   Si no, repito el paso 2
   - En este caso tengo que repetir el resultado tiene parte fraccionaria
   - $0{,}5 \times 2 = 1$

¡Listo! $\boxed{(0{,}375)_{10} = (0{,}011)_2}$

Un algoritmo

Quiero expresar $(0{,}375)_{10}$ en base 2

1. Multiplico el número por la base
   - $0{,}375 \times 2 = 0{,}75$
2. Multiplico la **parte fraccionaria** del resultado por la base
   - $0{,}75 \times 2 = 1{,}5$
3. Si el resultado **no** tiene parte fraccionaria, termino
   Si no, repito el paso 2
   - En este caso tengo que repetir el resultado tiene parte fraccionaria
   - $0{,}5 \times 2 = 1$

¡Listo! $\boxed{(0{,}375)_{10} = (0{,}011)_2}$

Ejercicios

- Expresar $(8{,}25)_{10}$ en base 4
- Expresar $(12{,}30)_{10}$ en base 2

Representando reales

Punto fijo: representa la parte entera y fraccionaria por separado

Punto flotante: sigue la idea de notación científica

Punto fijo

Para representar números de punto fijo

- Representamos la parte entera usando alguno de los métodos vistos en la clase pasada
  - sin signo
  - con signo
  - complemento a 2
  - exceso m
- Representamos la parte fraccionaria en la base b

Punto Fijo - Ejemplo

Defino un sistema de punto fijo:

4 bits para la parte entera, y otros 4 para la fraccionaria

Punto Fijo - Ejemplo

Defino un sistema de punto fijo:

4 bits para la parte entera, y otros 4 para la fraccionaria

Yo decido que el sistema sólo representará reales *positivos*
(codificación sin signo de 4 bits)

por lo que la representación de la parte entera no necesita poder expresar signo

Punto Fijo - Ejemplo

Defino un sistema de punto fijo:

4 bits para la parte entera, y otros 4 para la fraccionaria

Yo decido que el sistema sólo representará reales *positivos*
(codificación sin signo de 4 bits)

por lo que la representación de la parte entera no necesita poder expresar signo

Arbitrariamente decido que la parte entera se escribe primero

Punto Fijo - Ejemplo

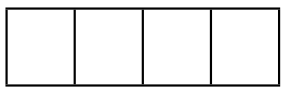
Defino un sistema de punto fijo:

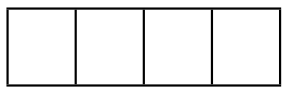
4 bits para la parte entera, y otros 4 para la fraccionaria

Yo decido que el sistema sólo representará reales *positivos*
(codificación sin signo de 4 bits)

por lo que la representación de la parte entera no necesita poder expresar signo

Arbitrariamente decido que la parte entera se escribe primero

Parte entera: 

Parte fraccionaria: 

Punto Fijo - Ejemplo

Defino un sistema de punto fijo:

4 bits para la parte entera, y otros 4 para la fraccionaria

Yo decido que el sistema sólo representará reales *positivos*
(codificación sin signo de 4 bits)

por lo que la representación de la parte entera no necesita poder expresar signo

Arbitrariamente decido que la parte entera se escribe primero

Parte entera:					Parte fraccionaria:				

Notación

Si escribo 10010001, debe entenderse:

Parte entera:	1	0	0	1	Parte fraccionaria:	0	0	0	1

Punto fijo

1. ¿Cuál es el real representable más grande? ¿y el más chico?

Punto fijo

1. ¿Cuál es el real representable más grande? ¿y el más chico?
   - $00000000 = (0{,}0)_2 = (0{,}0)_{10}$
   - $11111111 = (1111{,}1111)_2 = (15{,}9375)_{10}$
2. ¿Cuál es el mínimo real representable mayor a cero?

Punto fijo

1. ¿Cuál es el real representable más grande? ¿y el más chico?
   - $00000000 = (0{,}0)_2 = (0{,}0)_{10}$
   - $11111111 = (1111{,}1111)_2 = (15{,}9375)_{10}$
2. ¿Cuál es el mínimo real representable mayor a cero?
   - $00000001 = (0{,}0001)_2 = (0{,}0625)_{10}$
3. ¿Cuál es el máximo real representable menor a uno?

Punto fijo

1. ¿Cuál es el real representable más grande? ¿y el más chico?
   - $00000000 = (0{,}0)_2 = (0{,}0)_{10}$
   - $11111111 = (1111{,}1111)_2 = (15{,}9375)_{10}$
2. ¿Cuál es el mínimo real representable mayor a cero?
   - $00000001 = (0{,}0001)_2 = (0{,}0625)_{10}$
3. ¿Cuál es el máximo real representable menor a uno?
   - $00001111 = (0{,}1111)_2 = (0{,}9375)_{10}$
4. Muestre un número racional que esté entre el cero y el mínimo real *representable* mayor a cero.

Punto fijo

1. ¿Cuál es el real representable más grande? ¿y el más chico?
   - $00000000 = (0{,}0)_2 = (0{,}0)_{10}$
   - $11111111 = (1111{,}1111)_2 = (15{,}9375)_{10}$
2. ¿Cuál es el mínimo real representable mayor a cero?
   - $00000001 = (0{,}0001)_2 = (0{,}0625)_{10}$
3. ¿Cuál es el máximo real representable menor a uno?
   - $00001111 = (0{,}1111)_2 = (0{,}9375)_{10}$
4. Muestre un número racional que esté entre el cero y el mínimo real *representable* mayor a cero.
   - $(0{,}00001)_2 = (0{,}03125)_{10}$
5. Muestre un número no racional que sea menor al máximo representable y mayor al mínimo. ¿Se puede representar?

Underflow

¿Cómo representamos ... un número que es más pequeño **en módulo** que el menor real distinto de cero representable?

Recién encontramos uno para nuestro sistema: $(0{,}00001)_2$

No podemos representarlo exactamente

Única opción: representarlo como $(0)_2$ o como $(0{,}0001)_2$ **¿Nos importa?**

Underflow

¿Cómo representamos ... un número que es más pequeño **en módulo** que el menor real distinto de cero representable?

Recién encontramos uno para nuestro sistema: $(0{,}00001)_2$

No podemos representarlo exactamente

Única opción: representarlo como $(0)_2$ o como $(0{,}0001)_2$ **¿Nos importa?** Depende... Errores de redondeo pueden hacer que $0{,}1 + 0{,}2 = 0{,}30000000000000004$

Punto flotante

Recordemos

La *mantisa* (m) representa un número fraccionario
El *exponente* (e) es a lo cual se debe elevar la base

Si b es la base entonces el número representado es

$$m \times b^e$$

Tanto la *mantisa* como el *exponente* pueden representarse:

- con signo
- sin signo
- con notación complemento
- con notación exceso m.

Punto flotante - Ejemplo

Defino un sistema de punto flotante:

- 4 bits para el exponente y otros 3 bits para la mantisa

Punto flotante - Ejemplo

Defino un sistema de punto flotante:

- 4 bits para el exponente y otros 3 bits para la mantisa
- Representación del exponente: signo+magnitud

Punto flotante - Ejemplo

Defino un sistema de punto flotante:

- 4 bits para el exponente y otros 3 bits para la mantisa
- Representación del exponente: signo+magnitud
- Representará sólo reales *positivos*

  por lo que la representación de la mantisa no necesita poder expresar signo.

Punto flotante - Ejemplo

Defino un sistema de punto flotante:

- 4 bits para el exponente y otros 3 bits para la mantisa
- Representación del exponente: signo+magnitud
- Representará sólo reales *positivos*

  por lo que la representación de la mantisa no necesita poder expresar signo.
- Y otra definición: la interpretación debe ser

$$0, mantisa \times 2^{exponente}$$

es decir tengo un cero implícito.

Punto flotante - Ejemplo

Defino un sistema de punto flotante:

- 4 bits para el exponente y otros 3 bits para la mantisa
- Representación del exponente: signo+magnitud
- Representará sólo reales *positivos*

  por lo que la representación de la mantisa no necesita poder expresar signo.
- Y otra definición: la interpretación debe ser

$$0, mantisa \times 2^{exponente}$$

es decir tengo un cero implícito.

Notación

Si escribo 1001001, debe entenderse:

Exponente: | 1 | 0 | 0 | 1 | Mantisa: | 0 | 0 | 1 |

Punto flotante - Ejemplo

1. ¿Qué real se codifica con 1110111?

Punto flotante - Ejemplo

1. ¿Qué real se codifica con 1110111?
   - $(0{,}111 \times 10^{-110})_2 = (0{,}875 \times 2^{-6})_{10} = (0{,}013671875)_{10}$

Punto flotante - Ejemplo

1. ¿Qué real se codifica con 1110111?
   - $(0{,}111 \times 10^{-110})_2 = (0{,}875 \times 2^{-6})_{10} = (0{,}013671875)_{10}$
2. ¿Cuál es el mayor real que podemos representar?

Punto flotante - Ejemplo

1. ¿Qué real se codifica con 1110111?
   - $(0{,}111 \times 10^{-110})_2 = (0{,}875 \times 2^{-6})_{10} = (0{,}013671875)_{10}$
2. ¿Cuál es el mayor real que podemos representar?
   - 0111111 $(0{,}111 \times 10^{111})_2 = (0{,}875 \times 2^{7})_{10} = (112)_{10}$

Punto flotante - Ejemplo

1. ¿Qué real se codifica con 1110111?
   - $(0{,}111 \times 10^{-110})_2 = (0{,}875 \times 2^{-6})_{10} = (0{,}013671875)_{10}$
2. ¿Cuál es el mayor real que podemos representar?
   - 0111111 $(0{,}111 \times 10^{111})_2 = (0{,}875 \times 2^{7})_{10} = (112)_{10}$
3. Dados los reales representados como 0010000 y 0101001, ¿cuál es el mayor?

Punto flotante - Ejemplo

1. ¿Qué real se codifica con 1110111?
   - $(0{,}111 \times 10^{-110})_2 = (0{,}875 \times 2^{-6})_{10} = (0{,}013671875)_{10}$
2. ¿Cuál es el mayor real que podemos representar?
   - 0111111 $(0{,}111 \times 10^{111})_2 = (0{,}875 \times 2^{7})_{10} = (112)_{10}$
3. Dados los reales representados como 0010000 y 0101001, ¿cuál es el mayor?
   - $0010000 = (0{,}000 \times 10^{010})_2 = (0)_{10}$
   - $0101001 = (0{,}001 \times 10^{101})_2 = (0{,}125 \times 2^{5})_{10} = (4)_{10}$

Punto flotante - Ejemplo

1. ¿Qué real se codifica con 1110111?
   - $(0{,}111 \times 10^{-110})_2 = (0{,}875 \times 2^{-6})_{10} = (0{,}013671875)_{10}$
2. ¿Cuál es el mayor real que podemos representar?
   - 0111111 $(0{,}111 \times 10^{111})_2 = (0{,}875 \times 2^{7})_{10} = (112)_{10}$
3. Dados los reales representados como 0010000 y 0101001, ¿cuál es el mayor?
   - $0010000 = (0{,}000 \times 10^{010})_2 = (0)_{10}$
   - $0101001 = (0{,}001 \times 10^{101})_2 = (0{,}125 \times 2^{5})_{10} = (4)_{10}$
4. ¿Todo real representable tiene una única codificación?

Punto flotante - Ejemplo

1. ¿Qué real se codifica con 1110111?
   - $(0{,}111 \times 10^{-110})_2 = (0{,}875 \times 2^{-6})_{10} = (0{,}013671875)_{10}$
2. ¿Cuál es el mayor real que podemos representar?
   - 0111111 $(0{,}111 \times 10^{111})_2 = (0{,}875 \times 2^{7})_{10} = (112)_{10}$
3. Dados los reales representados como 0010000 y 0101001, ¿cuál es el mayor?
   - $0010000 = (0{,}000 \times 10^{010})_2 = (0)_{10}$
   - $0101001 = (0{,}001 \times 10^{101})_2 = (0{,}125 \times 2^{5})_{10} = (4)_{10}$
4. ¿Todo real representable tiene una única codificación?
   - $0000000 = (0{,}000 \times 10^{0})_2 = (0)_{10}$

Punto flotante - Ejemplo

1. ¿Qué real se codifica con 1110111?
   - $(0{,}111 \times 10^{-110})_2 = (0{,}875 \times 2^{-6})_{10} = (0{,}013671875)_{10}$
2. ¿Cuál es el mayor real que podemos representar?
   - 0111111 $(0{,}111 \times 10^{111})_2 = (0{,}875 \times 2^{7})_{10} = (112)_{10}$
3. Dados los reales representados como 0010000 y 0101001, ¿cuál es el mayor?
   - $0010000 = (0{,}000 \times 10^{010})_2 = (0)_{10}$
   - $0101001 = (0{,}001 \times 10^{101})_2 = (0{,}125 \times 2^{5})_{10} = (4)_{10}$
4. ¿Todo real representable tiene una única codificación?
   - $0000000 = (0{,}000 \times 10^{0})_2 = (0)_{10}$
   - $0110000 = (0{,}000 \times 10^{110})_2 = (0)_{10}$

Representación Normalizada ejemplo

Representación de punto flotante

- Base 2,
- Exponente 2 dígitos, exceso a 2
- Mantisa de 2 dígitos
- Normalizado 1.

$exp \backslash mant$	00	01	10	11
00				
01				
10				
11				

Representación Normalizada ejemplo

Representación de punto flotante

- Base 2,
- Exponente 2 dígitos, exceso a 2
- Mantisa de 2 dígitos
- Normalizado 1.

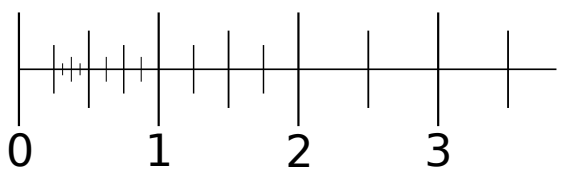
$exp \backslash mant$	00	01	10	11
00	0,25	0,3125	0,375	0,4375
01	0,5	0,625	0,75	0,875
10	1	1,25	1,5	1,75
11	2	2,5	3	3,5

Representación Normalizada ejemplo

Representación de punto flotante

- Base 2,
- Exponente 2 dígitos, exceso a 2
- Mantisa de 2 dígitos
- Normalizado 1.

$exp \backslash mant$	00	01	10	11
00	0,25	0,3125	0,375	0,4375
01	0,5	0,625	0,75	0,875
10	1	1,25	1,5	1,75
11	2	2,5	3	3,5



Representación Normalizada ejemplo

Representación de punto flotante

- Base 2,
- Exponente 2 dígitos, exceso a 2
  Excepción: cuando exponente= 00, se *denormaliza* y se toma como exponente -1
- Mantisa de 2 dígitos
- Normalizado a 1. ,salvo cuando exponente= 00.

$exp \backslash mant$	00	01	10	11
00				
01				
10				
11				

Representación Normalizada ejemplo

Representación de punto flotante

- Base 2,
- Exponente 2 dígitos, exceso a 2
  Excepción: cuando exponente$=00$, se *denormaliza* y se toma como exponente -1
- Mantisa de 2 dígitos
- Normalizado a 1. ,salvo cuando exponente$=00$.

$exp \backslash mant$	00	01	10	11
00				
01	0,5	0,625	0,75	0,875
10	1	1,25	1,5	1,75
11	2	2,5	3	3,5

Representación Normalizada ejemplo

Representación de punto flotante

- Base 2,
- Exponente 2 dígitos, exceso a 2
  Excepción: cuando exponente$=00$, se *denormaliza* y se toma como exponente -1
- Mantisa de 2 dígitos
- Normalizado a 1. ,salvo cuando exponente$=00$.

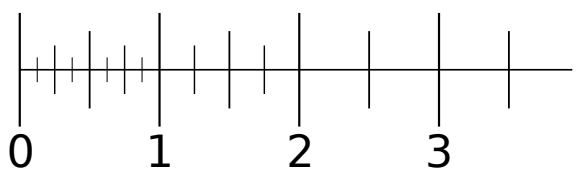
$exp \backslash mant$	00	01	10	11
00	0	0,125	0,25	0,375
01	0,5	0,625	0,75	0,875
10	1	1,25	1,5	1,75
11	2	2,5	3	3,5

Representación Normalizada ejemplo

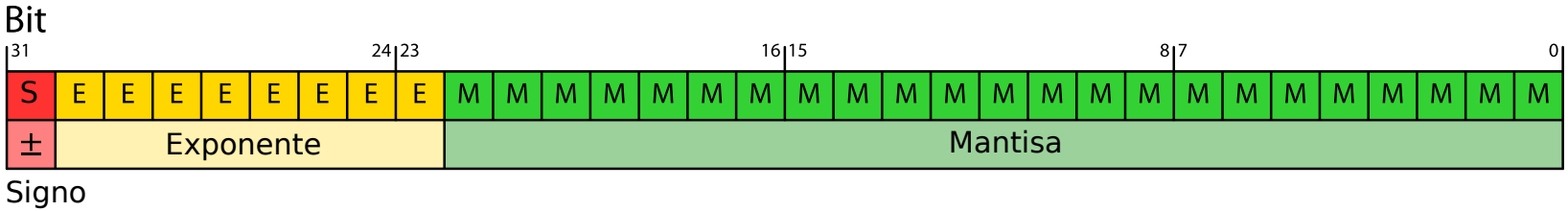
Representación de punto flotante

- Base 2,
- Exponente 2 dígitos, exceso a 2
  Excepción: cuando exponente= 00, se *denormaliza* y se toma como exponente -1
- Mantisa de 2 dígitos
- Normalizado a 1. ,salvo cuando exponente= 00.

$exp \backslash mant$	00	01	10	11
00	0	0,125	0,25	0,375
01	0,5	0,625	0,75	0,875
10	1	1,25	1,5	1,75
11	2	2,5	3	3,5



IEEE 754 - Precisión simple



- Signo: 1 bit
- Mantisa: 23 bits (sin signo, ya que hay un bit a tal efecto)
- Exponente: 8 bits, exceso 127; $e_{min} = -126$, $e_{max} = 127$

	Exponente (e)	Mantisa (f)	Representación
Normalizados	$e_{min} \leq e \leq e_{max}$	- - -	$\pm\ 1.f\ \times 2^{e}$
Ceros	$e = e_{min} - 1$	$f = 0$	? 0
Denormalizados	$e = e_{min} - 1$	$f\ \neq\ 0$	$\pm\ 0.f\ \times 2^{e_{min}}$
Infinitos	$e = e_{max} + 1$	$f = 0$	$\pm\ \infty$
NaNs	$e = e_{max} + 1$	$f \neq 0$	*NaN*

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	1000 0010	1111 0 . . . 0

¿Qué número real representa?

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	1000 0010	1111 0 . . . 0

¿Qué número real representa?

- Signo: 1 (negativo)

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	1000 0010	1111 0 ... 0

¿Qué número real representa?

- Signo: 1 (negativo)
- Exponente: $(10000010)_2 = (130)_{10}$.

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	1000 0010	1111 0 ... 0

¿Qué número real representa?

- Signo: 1 (negativo)
- Exponente: $(10000010)_2 = (130)_{10}$. La codificación es exceso a 127 de 8 bits. Por lo tanto, el numeral $(130)_{10}$ representa el número entero 3.

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	1000 0010	1111 0 ... 0

¿Qué número real representa?

- Signo: 1 (negativo)
- Exponente: $(10000010)_2 = (130)_{10}$. La codificación es exceso a 127 de 8 bits. Por lo tanto, el numeral $(130)_{10}$ representa el número entero 3.
- Mantisa: $(0{,}1111)_2 = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} = (0{,}9375)$

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	1000 0010	1111 0 ... 0

¿Qué número real representa?

- Signo: 1 (negativo)
- Exponente: $(10000010)_2 = (130)_{10}$. La codificación es exceso a 127 de 8 bits. Por lo tanto, el numeral $(130)_{10}$ representa el número entero 3.
- Mantisa: $(0{,}1111)_2 = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} = (0{,}9375)$

Por lo tanto (dado el 1. ímplicito), el número que representa es:

$$-1{,}9375 * 2^3 = -15{,}5$$

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	0000 0000	1011 10 . . . 0

¿Qué número real representa?

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	0000 0000	1011 10 . . . 0

¿Qué número real representa?

- Signo: 1 (negativo)

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	0000 0000	1011 10 ... 0

¿Qué número real representa?

- Signo: 1 (negativo)
- Exponente: $(00000000)_2 = (0)_{10}$.

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	0000 0000	1011 10 ... 0

¿Qué número real representa?

- Signo: 1 (negativo)
- Exponente: $(00000000)_2 = (0)_{10}$. La codificación es exceso a 127 de 8 bits. Por lo tanto, el numeral $(0)_{10}$ representa el número entero -127. Es un caso especial! (e == $e_{min} - 1$)

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	0000 0000	1011 10 ... 0

¿Qué número real representa?

- Signo: 1 (negativo)
- Exponente: $(00000000)_2 = (0)_{10}$. La codificación es exceso a 127 de 8 bits. Por lo tanto, el numeral $(0)_{10}$ representa el número entero -127. Es un caso especial! (e == $e_{min} - 1$)
- Mantisa: $(0{,}10111)_2$ Es distinto de cero! ($f \neq 0$)

IEEE 754

Sea el siguiente numeral:

Signo	Exponente	Mantisa
1	0000 0000	1011 10 ... 0

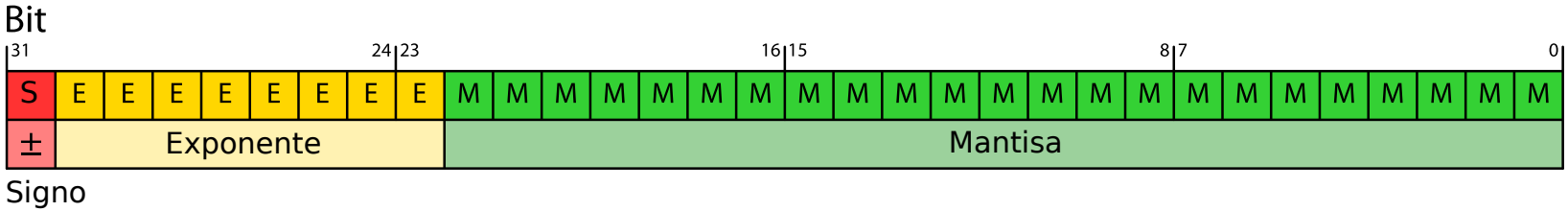
¿Qué número real representa?

- Signo: 1 (negativo)
- Exponente: $(00000000)_2 = (0)_{10}$. La codificación es exceso a 127 de 8 bits. Por lo tanto, el numeral $(0)_{10}$ representa el número entero -127. Es un caso especial! (e == $e_{min} - 1$)
- Mantisa: $(0{,}10111)_2$ Es distinto de cero! ($f \neq 0$)

Por lo tanto se trata de un caso especial. El número que representa es $0, f \ \times 2^{e_{min}}$:

$$-(0{,}10111)_2 * 2^{-126} \approx -8{,}448865646535191463525293861284 9 * 10^{-39}$$

IEEE 754



¿Pero cómo convertimos a este formato?

1. Convertir a binario
2. Normalizar
3. Codificar el signo
4. Codificar la mantisa
5. Codificar el exponente

Ejemplo

Ejemplo

Convirtamos $(-15{,}32)_{10}$ al formato IEEE 754 de *precisión simple*

Ejemplo

Convirtamos $(-15{,}32)_{10}$ al formato IEEE 754 de *precisión simple*

1. Primero tenemos que *expresar el número en binario*

Ejemplo

Convirtamos $(-15{,}32)_{10}$ al formato IEEE 754 de *precisión simple*

1. Primero tenemos que *expresar el número en binario*

$$(-15{,}32)_{10} = (-1111\ ,\ 01\overbrace{0100011110101100001}^{periodo})_2$$

Ejemplo

Convirtamos $(-15{,}32)_{10}$ al formato IEEE 754 de *precisión simple*

1. Primero tenemos que *expresar el número en binario*

$$(-15{,}32)_{10} = (-1111\ ,\ 01\overbrace{0100011110101100001}^{periodo})_2$$

2. Ahora tenemos que *normalizarlo*

Ejemplo

Convirtamos $(-15{,}32)_{10}$ al formato IEEE 754 de *precisión simple*

1. Primero tenemos que *expresar el número en binario*

$$(-15{,}32)_{10} = (-1111\ ,\ 01\overbrace{0100011110101100001}^{periodo})_2$$

2. Ahora tenemos que *normalizarlo*

$$(-15{,}32)_{10} = (-1\ ,\ 11101\overbrace{0100011110101100001}^{periodo} \times 10^{11})_2$$

Ejemplo

Convirtamos $(-15{,}32)_{10}$ al formato IEEE 754 de *precisión simple*

1. Primero tenemos que *expresar el número en binario*

$$(-15{,}32)_{10} = (-1111\ ,\ 01\overbrace{0100011110101100001}^{periodo})_2$$

2. Ahora tenemos que *normalizarlo*

$$(-15{,}32)_{10} = (-1\ ,\ 11101\overbrace{0100011110101100001}^{periodo} \times 10^{11})_2$$

3. Próximo paso, llenar la *plantilla* de la IEEE.

- ¿Cómo representamos el exponente?

- ¿Cómo representamos el exponente?
  IEEE dice que tenemos que utilizar **exceso 127**

- ¿Cómo representamos el exponente?
  IEEE dice que tenemos que utilizar **exceso 127**
  $(11)_2 = (3)_{10}$ entonces lo que necesitamos es pasar a binario el número $(127 + 3)_{10}$, o sea $(130)_{10}$

- ¿Cómo representamos el exponente?
  IEEE dice que tenemos que utilizar **exceso 127**
  $(11)_2 = (3)_{10}$ entonces lo que necesitamos es pasar a binario el número $(127 + 3)_{10}$, o sea $(130)_{10}$

  $(130)_{10} = (10000010)_2$ Representado en 8 bits, el espacio que tenemos en la plantilla de precisión simple de IEEE
- ¿Cómo representamos la mantisa?

- ¿Cómo representamos el exponente?
  IEEE dice que tenemos que utilizar **exceso 127**
  $(11)_2 = (3)_{10}$ entonces lo que necesitamos es pasar a binario el número $(127 + 3)_{10}$, o sea $(130)_{10}$

  $(130)_{10} = (10000010)_2$ Representado en 8 bits, el espacio que tenemos en la plantilla de precisión simple de IEEE
- ¿Cómo representamos la mantisa?
  IEEE dice que en precisión simple tenemos 23 bits para la mantisa. Y no nos olvidemos del **uno implícito**

- ¿Cómo representamos el exponente?
  IEEE dice que tenemos que utilizar **exceso 127**
  $(11)_2 = (3)_{10}$ entonces lo que necesitamos es pasar a binario el número $(127 + 3)_{10}$, o sea $(130)_{10}$

  $(130)_{10} = (10000010)_2$ Representado en 8 bits, el espacio que tenemos en la plantilla de precisión simple de IEEE
- ¿Cómo representamos la mantisa?
  IEEE dice que en precisión simple tenemos 23 bits para la mantisa. Y no nos olvidemos del **uno implícito**
  Teníamos nuestro número normalizado

$$(-1\ ,\ \underbrace{11101010001111010111000}_{bits}01\ \ldots\ \times\ 10^{11})_2$$

- ¿Cómo representamos el exponente?
  IEEE dice que tenemos que utilizar **exceso 127**
  $(11)_2 = (3)_{10}$ entonces lo que necesitamos es pasar a binario el número $(127 + 3)_{10}$, o sea $(130)_{10}$

  $(130)_{10} = (10000010)_2$ Representado en 8 bits, el espacio que tenemos en la plantilla de precisión simple de IEEE
- ¿Cómo representamos la mantisa?
  IEEE dice que en precisión simple tenemos 23 bits para la mantisa. Y no nos olvidemos del **uno implícito**
  Teníamos nuestro número normalizado

$$(-1\ ,\ \underbrace{11101010001111010111000}_{bits}01\ \ldots\ \times\ 10^{11})_2$$

  El uno va a ser implícito.

- ¿Cómo representamos el exponente?
  IEEE dice que tenemos que utilizar **exceso 127**
  $(11)_2 = (3)_{10}$ entonces lo que necesitamos es pasar a binario el número $(127 + 3)_{10}$, o sea $(130)_{10}$

  $(130)_{10} = (10000010)_2$ Representado en 8 bits, el espacio que tenemos en la plantilla de precisión simple de IEEE
- ¿Cómo representamos la mantisa?
  IEEE dice que en precisión simple tenemos 23 bits para la mantisa. Y no nos olvidemos del **uno implícito**
  Teníamos nuestro número normalizado

$$(-1\ ,\ \underbrace{11101010001111010111000}_{bits}01\ \ldots\ \times\ 10^{11})_2$$

  El uno va a ser implícito. Así que necesitamos los primeros 23 numerales después de la coma

Listo! Tenemos todo lo que necesitamos

Listo! Tenemos todo lo que necesitamos

- Signo: 1

Listo! Tenemos todo lo que necesitamos

- Signo: 1
- Exponente: 10000010

Listo! Tenemos todo lo que necesitamos

- Signo: 1
- Exponente: 10000010
- Mantisa: 11101010001111010111000

Listo! Tenemos todo lo que necesitamos

- Signo: 1
- Exponente: 10000010
- Mantisa: 11101010001111010111000

La tira de bits que buscábamos es:

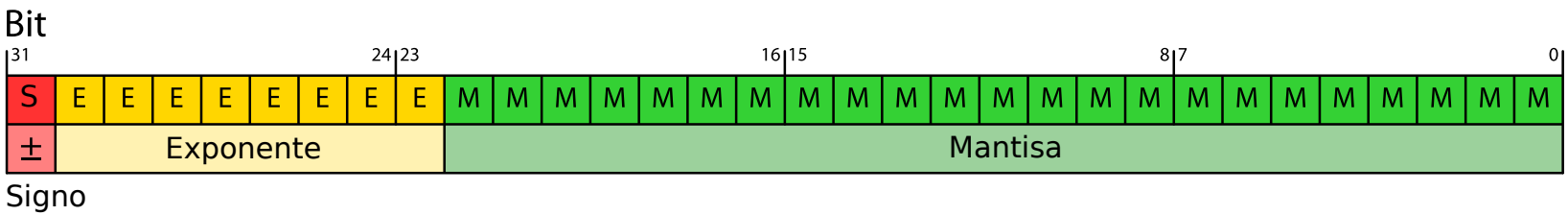
Listo! Tenemos todo lo que necesitamos

- Signo: 1
- Exponente: 10000010
- Mantisa: 11101010001111010111000

La tira de bits que buscábamos es:

$$1100000101110101000111101011100\color{red}{0}$$

donde el marcado en rojo es el *bit de la posición cero*



Redondeo y Truncamiento

- Si queremos escribir $101,010011_2 = 5,296875_{10}$ en punto fijo, con cuatro bits

Redondeo y Truncamiento

- Si queremos escribir $101,010011_2 = 5,296875_{10}$ en punto fijo, con cuatro bits
- Truncar es tomar los cuatro bits fraccionarios más significativos
- $101,0100_2 = 5,25_{10}$

Redondeo y Truncamiento

- Si queremos escribir $101,010011_2 = 5,296875_{10}$ en punto fijo, con cuatro bits
- Truncar es tomar los cuatro bits fraccionarios más significativos
- $101,0100_2 = 5,25_{10}$
- Redondear es cambiar el último de acuerdo al siguiente
- si es cero, dejo el que estaba; si es uno, lo cambio

Redondeo y Truncamiento

- Si queremos escribir $101,010011_2 = 5,296875_{10}$ en punto fijo, con cuatro bits
- Truncar es tomar los cuatro bits fraccionarios más significativos
- $101,0100_2 = 5,25_{10}$
- Redondear es cambiar el último de acuerdo al siguiente
- si es cero, dejo el que estaba; si es uno, lo cambio
- $101,0101_2 = 5,31225_{10}$

Redondeo y Truncamiento

- Si queremos escribir $101,010011_2 = 5,296875_{10}$ en punto fijo, con cuatro bits
- Truncar es tomar los cuatro bits fraccionarios más significativos
- $101,0100_2 = 5,25_{10}$
- Redondear es cambiar el último de acuerdo al siguiente
- si es cero, dejo el que estaba; si es uno, lo cambio
- $101,0101_2 = 5,31225_{10}$
- Redondear suele resultar en una representación más cercana que truncar

Hasta ahora

- Operaciones de cambio de base en números no enteros
- Punto fijo y punto flotante
- Distintas formas de representación de reales con bits

Bibliografía

Linda Null - Julia Lobur. "Essentials of Computer Organization and Architecture". Jones and Bartlett Publishers, Inc.
Capítulo 2

Yates, Randy. "Fixed-point arithmetic: An introduction." Digital Signal Labs 81.83 (2009): 198.
`http://personal.atl.bellsouth.net/y/a/yatesc/fp.pdf`

Conversor IEEE 754
`http://www.zator.com/Cpp/E2_2_4a1.htm`

[Opcional] What Every Computer Scientist Should Know About Floating-Point Arithmetic
`http://docs.sun.com/source/806-3568/ncg_goldberg.html`