Plancha 2

Representación Computacional de Números Reales

2024 - Arquitectura del Computador

Licenciatura en Ciencias de la Computación

1

Introducción

Esta plancha trata los sistemas de representación de números enteros en el lenguaje de programación C y los operadores de bits para manipular estos números a bajo nivel.

2

Procedimiento

Resuelva cada ejercicio en computadora. Cree un subdirectorio dedicado para cada ejercicio, que contenga todos los archivos del mismo. Para todo ejercicio que pida escribir código, genere un programa completo, con su función `main` correspondiente; evite dejar fragmentos sueltos de programas.

Asegúrese de que todos los programas que escriba compilen correctamente con `gcc`. Se recomienda además pasar a este las opciones `-Wall` y `-Wextra` para habilitar advertencias sobre construcciones cuestionables en el código

3

Punto Flotante

1. Expresar en norma IEEE 754 simple precisión los siguientes números:

   a) 29.

   b) $0,625$.

   c) $5,75$.

   d) -138.

   e) $-15,125$.

   f) $0,1$.

Soluciones

a) $29 = (-1)^{0} \cdot (11101)_2 \cdot 2^{0} = (-1)^{0} \cdot (1,1101)_2 \cdot 2^{4}$.

- $s = 0$.
- $E = 4 + 127 = 131 = (10000011)_2$.
- $f = (0,1101)_2$.
- $29 = \left(\boxed{0}\boxed{10000011}\boxed{11010000000000000000000}\right)_{IEEE754}$.

b) $0,625 = (-1)^{0} \cdot (0,101)_2 \cdot 2^{0} = (-1)^{0} \cdot (1,01)_2 \cdot 2^{-1}$.

- $s = 0$.
- $E = -1 + 127 = 126 = (01111110)_2$.
- $f = (0,01)_2$.
- $0,625 = \left(\boxed{0}\boxed{01111110}\boxed{01000000000000000000000}\right)_{IEEE754}$.

c) $5,75 = (-1)^{0}\,(101,11)_2 \cdot 2^{0} = (-1)^{0}\,(1,0111)_2 \cdot 2^{2}$.

- $s = 0$.
- $E = 2 + 127 = 129 = (10000001)_2$.
- $f = (0,0111)_2$.
- $5,75 = \left(\boxed{0}\boxed{10000001}\boxed{01110000000000000000000}\right)_{IEEE754}$.

d) $-138 = (-1)^{1} \cdot (10001010)_2 \cdot 2^0 = (-1)^{1} \cdot (1,0001010)_2 \cdot 2^7$.
- $s = 1$.
- $E = 7 + 127 = 134 = (10000110)_2$.
- $f = (0,0001010)_2$.
- $-138 = \left(\boxed{1}\boxed{10000110}\boxed{00010100000000000000000}\right)_{IEEE754}$.

e) $-15,125 = (-1)^{1} (1111,001)_2 \cdot 2^0 = (-1)^{1} (1,111001)_2 \cdot 2^3$.
- $s = 1$.
- $E = 3 + 127 = 130 = (10000010)_2$.
- $f = (0,111001)_2$.
- $-15,125 = \left(\boxed{1}\boxed{10000010}\boxed{11100100000000000000000}\right)_{IEEE754}$.

f) $0,1 = (-1)^{0} (0,0001\overline{1100})_2 \cdot 2^0 = (-1)^{0} (1,\overline{1001})_2 \cdot 2^{-4}$.
- $s = 0$.
- $E = -4 + 127 = 123 = (01111011)_2$.
- $f = (0,\overline{1001})_2$.
- $0,1 \approx \left(\boxed{0}\boxed{01111011}\boxed{10011001100110011001101}\right)_{IEEE754}$.

2. Haga dos funciones o macros de C para extraer la fracción y el exponente de un `float` sin usar variables auxiliares.

   Sugerencia: utilice corrimientos de bits y máscaras. Luego use los tipos definidos en la cabecera `ieee754.h` para corroborar.

   Solución

```c
unsigned int signo(float f) {
    unsigned int* r = (int*) &f;
    return *r >> 31;
}

unsigned int fraccion(float f) {
    unsigned int* r = (int*) &f;
    return (*r << 9) >> 9;
}

unsigned int exponente(float f) {
    unsigned int* r = (int*) &f;
    return (*r << 1) >> 24;
}
```

3. Convierta a `double` y `float` norma IEEE 754 el número: $N = 6,225$. Realice el cálculo de manera explícita y luego corrobore el resultado mediante un programa que aproveche las herramientas provistas en el ejercicio anterior. Analizar en cada caso si se ha cometido error de representación y en caso afirmativo la magnitud del mismo.

Solución

- $6,225 = (-1)^0 \left(110,001\overline{1100}\right)_2 \cdot 2^0 = (-1)^0 \left(1,1000\overline{11100}\right)_2 \cdot 2^2$.
- $s = 0$.
- $E_1 = 2 + 127 = 129 = (10000001)_2$.
- $f = \left(0,1000\overline{11100}\right)_2$.
- $6,225 \approx \left(\boxed{0}\boxed{10000001}\boxed{10001110011001100110011}\right)_{IEEE754}$.
- $E_2 = 2 + 1023 = 1025 = (10000000001)_2$.
- $6,225 \approx \left(\boxed{0}\boxed{10000000001}\boxed{1000111001100110011001100110011001100110011001100110}\right)_{IEEE754}$.

4. Dados los números $N_1 = (100000)_{10}$, $N_2 = (0,2)_{10}$ y $N_3 = (0,1)_{10}$:

   a) Realizar la operación $N_1 \otimes (N_2 \oplus N_3)$ en simple precisión IEEE754.

   b) Realizar la operación $(N_1 \otimes N_2) \oplus (N_1 \otimes N_3)$ en simple precisión IEEE754.

   c) Repetir para doble precisión IEEE 754.

   d) Comparar los resultados.

   Solución

   a) COMPLETAR.

   b) COMPLETAR.

   c) COMPLETAR.

   d) La propiedad distributiva del producto respecto de la suma puede no valer para valores en norma IEEE754.
   Puede apreciarse como en simple precisión, el segundo resultado es preferible.
   En doble precisión no se observan diferencias para esos valores.

5. Realice el procedimiento de suma en simple precisión del número $(1,75)_{10} \cdot 2^{-79}$ con el número $0x19d00000$ expresado en IEEE 754 simple precisión.

   Solución COMPLETAR.

6. El siguiente programa muestra algunas cualidades de los números `NaN` (*Not A Number*) y la función `isnan` de C, que indica si un flotante es `NaN`.

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
```

```
    float g = 0.0;
    float f = 0.0 / g;
    printf("f: %f\n", f);

    // ADVERTENCIA: 'NAN' es una extension de GCC.
    if (f == NAN) printf("Es NAN\n");
    if (isnan(f)) printf("isNaN dice que si\n");
}
```

a) El programa muestra que comparar con `NaN` retorna siempre falso y para saber si una operación dio `NaN` se puede usar `isnan`. Utilizando las funciones de los ejercicios anteriorer, implemente una función `myisnan` que haga lo mismo que la función `isnan` de C.

b) Implemente otra función, `myisnan2`, que haga lo mismo pero utilizando solo una comparación y sin operaciones de bits.

c) ¿Ocurre lo mismo con $+\infty$?

d) ¿Qué pasa si se suma un valor a $+\infty$?

Solución

a)
```
int myisnan(float f) {
    return (exponente(f) == (1 << 8) - 1) && fraccion(f);
}
```

b)
```
int myisnan2(float f) {
    return *((unsigned int*) &f) > 0x7F800000;
}
```

c) COMPLETAR.

d) COMPLETAR.

7. Dados los siguientes números representados en punto flotante IEEE 754 simple precisión, indicar a qué número en formato decimal corresponden y analizar si son números normalizados:

a) $N_1 = (11000010111011010100000000000000)_{IEEE754}$.

b) $N_2 = (0x40600000)_{IEEE754}$.

c) $N_3 = (0x00600000)_{IEEE754}$.

Soluciones

a) $N_1 = \left(\boxed{1}\boxed{10000101}\boxed{11011010100000000000000}\right)_{IEEE754}$:

- $(10000101)_2 = 133$.
- $e = 133 - 127 = 6$.
- $(1, 110110101)_2 = 2^0 + 2^{-1} + 2^{-2} + 2^{-4} + 2^{-5} + 2^{-7} + 2^{-9} = 1,853515625$
- $N_1 = -1,853515625 \cdot 2^6 = -118,625$.

b) $N_2 = \left(\boxed{0}\boxed{10000000}\boxed{10000000000000000000000}\right)_{IEEE754}$:

- $(10000000)_2 = 128$.
- $e = 128 - 127 = 1$.
- $(1, 11)_2 = 2^0 + 2^{-1} + 2^{-2} = 1,75$.
- $N_2 = 1,75 \cdot 2^1 = 3,5$.

c) $N_3 = \left(\boxed{0}\boxed{00000000}\boxed{10000000000000000000000}\right)_{IEEE754}$:

- $(00000000)_2 = 0$.
- Como N_3 es un numero denormalizado ($E = 0$) entonces $e = -126$.
- $(0, 11) = 2^{-1} + 2^{-2} = 0,75$.
- $N_3 = 0,75 \cdot 2^{-126} = 8,816207631167156309765524 \cdot 10^{-39}$.

8. Determinar la representación de punto flotante de π utilizando 4 dígitos. Utilizar la norma IEEE 754 simple precisión.

   Solución Redondeamos $\pi \approx 3,1416$, luego observemos que $(3)_{10} = (11)_2$ y $(0,1416)_{10} \approx (0,00100100001111111111001)_2$, luego:

   - $\pi \approx (11,00100100001111111111001)_2 \cdot 2^0 = (1,10010010000111111111001)_2 \cdot 2^1$.
   - $s = 0$.
   - $E = 1 + 127 = 128 = (10000000)_2$.
   - $f = (0,10010010000111111111001)_2$.
   - $3,1416 \approx \left(\boxed{0}\boxed{10000000}\boxed{10010010000111111111001}\right)_{IEEE754}$.

9. Realizar la suma $0,1 + 0,2$ utilizando aritmética en punto flotante con norma IEEE 754 simple precisión. Luego realizar la suma $0,1 + 0,4$. ¿Qué se puede observar?

   Ayuda: Al realizar las conversiones se puede reducir la cantidad de operaciones observando la periodicidad de los resultados.

Solución Observemos que:

- $0,1 \approx (0\,01111011\,10011001100110011001101)_{IEEE754} =$
  $= (-1)^0\, 2^{-4}\, (1,10011001100110011001101)_2 \approx$
  $\approx (-1)^0\, 2^{-3}\, (0,11001100110011001100110)_2 \approx$
  $\approx (-1)^0\, 2^{-2}\, (0,01100110011001100110011)_2$
- $0,2 \approx (0\,01111100\,10011001100110011001101)_{IEEE754} =$
  $= (-1)^0\, 2^{-3}\, (1,10011001100110011001101)_2$

luego

$$\begin{array}{cccccccccccccccccccccccccc} & 0, & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ + & 1, & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline 1 & 0, & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{array}$$

por lo que

$$\begin{aligned} 0,1 + 0,2 &\approx (-1)^0\, 2^{-3}\, (10,01100110011001100110011) = \\ &= (-1)^0\, 2^{-2}\, (1,00110011001100110011001) \approx \\ &\approx \left(\boxed{0}\,\boxed{01111101}\,\boxed{00110011001100110011010}\right)_{IEEE754} = \\ &= (0,30000001192092895507812 5)_{10} \end{aligned}$$

Además

- $(0,4)_{10} \approx (0\,01111101\,10011001100110011001101)_{IEEE754} \approx$
  $\approx (-1)^0\, 2^{-2}\, (1,10011001100110011001101)_2$

luego

$$\begin{array}{cccccccccccccccccccccccccc} & 0, & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ + & 1, & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline 1 & 0, & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

por lo que

$$\begin{aligned} (0,1)_{10} + (0,4)_{10} &= (-1)^0\, 2^{-2}\, (10) = (-1)^0\, 2^{-1}\, (1,0) = 0,5 = \\ &= (0\,01111110\,00000000000000000000000)_{IEEE754} \end{aligned}$$

10. Efectuar los siguientes cálculos utilizando aritmética en punto flotante con norma IEEE 754 simple precisión, siendo $a = 12345$, $b = 0,0001$, $c = 45,5$:

- $(a \oplus b) \oplus c$
- $a \oplus (b \oplus c)$

Al comparar los resultados en cada una de las operaciones, ¿qué puede concluirse?

Solución Observemos que:

- $(12345)_{10} \approx (01000110010000001110010000000000)_{IEEE754} \approx$
  $\approx (-1)^0 2^{13} (1,1000000111001)_2$
- $(0,0001)_{10} \approx (00111000110100011011011100010111)_{IEEE754} \approx$
  $\approx (-1)^0 2^{-14} (1,10100011011011100010111)_2 \approx$
  $\approx (-1)^0 2^5 (0,00000000000000000011010)_2 \neq$
  $\neq (-1)^0 2^{13} (0,00000000000000000000000000)_2$
- $(45,5)_{10} \approx (01000010001101100000000000000000)_{IEEE754} =$
  $= (-1)^0 2^5 (1,011011)_2 = (-1)^0 2^{13} (0,00000001011011)_2$

luego:

- $(a \oplus b) \oplus c = a \oplus c = (01000110010000011001101000000000)_{IEEE754} =$ $(12390,5)_{10} \approx (a+b)+c = (12390{,}5001)_{10}$
- $a \oplus (b \oplus c) = (12345)_{10} \oplus (01000010001101100000000000011010)_{IEEE754} =$
  $= (-1)^0 2^{13} (1,1000000111001)_2 + (-1)^0 2^5 (1,01101100000000000011010)_2 \approx$
  $\approx (-1)^0 2^{13} (1,1000000111001)_2 + (-1)^0 2^{13} (0,00000001011011000000000)_2 =$
  $= (-1)^0 2^{13} (1,10000011001101)_2 = (12390,5)_{10} \approx a + (b+c) =$ $(12390{,}5001)_{10}$

11. Repetir el ejercicio anterior pero ahora utilizando doble precisión.

Ayuda No es necesario volver a hacer todo el procedimiento para hacer las conversiones.

Solución Puede apreciarse que en este caso, ambas operaciones dan el resultado exacto.

12. Dados los números $A = (24)_{10}$, $B = (30)_{10}$ y $C = (15,75)_{10}$:
    a) Realizar la suma $S = A \oplus B \oplus C$ en norma IEEE754 simple precisión.
    b) Expresar el resultado en hexadecimal.
    c) Convertir el resultado a doble precisión y expresar el resultado en hexadecimal.

Solución

a) COMPLETAR.

b) COMPLETAR.

c) COMPLETAR.

13. Considere el conjunto de números de punto flotante $\mathbb{F}(2,3,-1,2)$:

a) Determinar $x_{\text{mín}}, x_{\text{máx}}, \epsilon_M$ y el número de elementos de $\mathbb{F}$.

b) Determinar los números de punto flotante positivos del conjunto $\mathbb{F}$.

c) Graficar sobre la recta real los números de puntos flotantes determinados en el punto anterior.

Ayuda Recordar la notación $\mathbb{F}(\beta, t, L, U)$, donde β es la base, t es la cantidad de dígitos significativos de la mantisa, L es el valor mínimo del exponente y U es el valor máximo del exponente.

Soluciones

a) Para $\mathbb{F}(2,3,-1,2)$ resulta:

- $x_{\text{mín}} = 2^{-1} \cdot 2^{-1} = 2^{-2} = 1/4 = 0,25$.
- $x_{\text{máx}} = 2^2\left(1 - 2^{-3}\right) = 4\left(1 - 1/8\right) = 28/8 = 7/2 = 3,5$.
- $\epsilon_M = 2^{1-t} = 2^{1-3} = 2^{-2} = 1/4 = 0,25$.
- $|\mathbb{F}(2,3,-1,2)| = 2(2-1)2^{3-1}(2-(-1)+1)+1 = 33$.

b) COMPLETAR.

c) COMPLETAR.

14. (*Opcional*) Determinar si el número $(2,89)_{10} \cdot 10^4$ es representable en un formato de coma flotante de 16 bits, con mantisa normalizada de la forma $(1, b_{-1}b_{-2}\dots)_2$, bit implícito y 5 bits para el exponente.

Solución Observemos que

$$\begin{aligned}(2,89)_{10} \cdot 10^4 &= (1110000011100100)_2 \cdot 2^0 \\ &= (1,11000011100100)_2 \cdot 2^{14}\end{aligned}$$

luego como tenemos reservados 5 bits para el exponente el rango es $[-14, 15]$ con sesgo 15 por lo que:

$$(2,89)_{10} \cdot 10^{10} \approx \left(\boxed{0}\,\boxed{11101}\,\boxed{1100001110}\right)_{IEEE754}$$