Organización de Computadoras – Fac. de Informática - UNLP – Prof. Jorge Runco
Curso 2020 - Práctica N° 1: Números sin signo

1)

El rango en BSS con 8 bits es desde 0 a 255.

0 = 00000000 1 = 00000001 = 2^0
127 = 01111111 = $2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$ = 64+32+16+8+4+2+1
128 = 10000000 = 2^7 256 = Fuera de rango
255 = 11111111 =$2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$ = 128+64+32+16+8+4+2+1
- 1 = Fuera de rango - 8 = Fuera de rango - 128 = Fuera de rango
137 = 10001001 = $2^7 + 2^3 + 2^0$ = 128 + 8 + 1
35 = 00100011 = $2^5 + 2^1 + 2^0$ = 32 + 2 + 1
100 = 01100100 = $2^6 + 2^5 + 2^2$ = 64 + 32 + 4
0,5 = Sólo enteros 1,25 = Sólo enteros

---

2)

00000000 = 0
11111111 = $2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$ =255
01010101 = $2^6 + 2^4 + 2^2 + 2^0$ =64 + 16 + 4 + 1 = 85
10101010 = $2^7 + 2^5 + 2^3 + 2^1$ = 128 + 32 + 8 + 2 = 170
10000000 = 2^7 = 128
01111111 = $2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$ = 127
11111110 = $2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1$ = 254
01100110 = $2^6 + 2^5 + 2^2 + 2^1$ = 64 + 32 + 4 + 2 = 102

---

3)

$$\underbrace{\_\_\ \ \_\_\ \ \_\_\ \ \_\_\ \ \_\_\ \ \_\_}_{\text{Parte entera}}\ ,\ \underbrace{\_\_\ \ \_\_\ \ \_\_\ \ \_\_}_{\text{Parte fraccionaria}}$$

Más chico = 000000,0000 = 0
Más grande = 111111,1111 = $2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4}$ =
= 32 + 16 + 8 + 4 + 2+1+ 0,5 + 0,25 + 0,125 + 0,0625 = 63, 9375

Resolución = distancia entre dos representaciones sucesivas = 0,0001 = 0,0625

---

4)

3 = 000011,0000 Representación exacta.
Error absoluto = |Numero a representar − Numero representado| = |3 − 3| = 0

5, 25

Parte entera → $5 = 0000101 = 2^2 + 2^0$

Parte fraccionaria →
$0{,}25 \times 2 = 0{,}5$
$0{,}5 \times 2 = 1{,}0$
$0{,}0 \times 2 = 0{,}0$
$0{,}25 = 0{,}0100 = 2^{-2}$

$5{,}25 = 0000101{,}\ 0100 = 2^2 + 2^0 + 2^{-2}$

Representación exacta.

Error absoluto = |Numero a representar − Numero representado| = |5,25 − 5,25| = 0

1, 2

Parte entera → $1 = 0000001 = 2^0$

Parte fraccionaria →
$0{,}2 \times 2 = 0{,}4$
$0{,}4 \times 2 = 0{,}8$
$0{,}8 \times 2 = 1{,}6$
$0{,}6 \times 2 = 1{,}2$
$0{,}0011 = 2^{-3} + 2^{-4}$

La parte fraccionaria tiene 4 bits, por eso no seguimos con las cuentas.

1,2 → $0000001{,}\ 0011 = 2^0 + 2^{-3} + 2^{-4} = 1 + 0{,}125 + 0{,}0625 = 1{,}1875$

El Nº que sigue → $0000001{,}\ 0100 = 2^0 + 2^{-2} = 1 + 0{,}25 = 1{,}25$

→ 1,0011 = 1,1875 Error = 1,2 – 1,1875 = 0,0125
1,0100 = 1,25 Error = 1,25 – 1,2 = 0,05

El error más pequeño es 0,0125 entonces 1,1875 es la representación más cercana a 1,2.

2,001

Parte entera → $2 = 0000010 = 2^1$

Parte fraccionaria →
$0{,}001 \times 2 = 0{,}002$
$0{,}002 \times 2 = 0{,}004$
$0{,}004 \times 2 = 0{,}008$
$0{,}008 \times 2 = 0{,}016$
$0{,}0000 = 0{,}0$

2,001 → $0000010{,}\ 0000 = 2^1 = 2{,}0$

El Nº que sigue → $0000010{,}\ 0001 = 2^1 + 2^{-4} = 2 + 0{,}0625 = 2{,}0625$

→ 000010,0000 = 2,0 Error = 2,001 – 2,0 = 0,001
000010,0001 = 2,0625 Error = 2,0625 – 2,001 = 0,0615

El error más pequeño es 0,001 entonces 2,0 es la representación más cercana a 2,001.

23,125

Parte entera $23 = 010111 = 2^4 + 2^2 + 2^1 + 2^0 = 16 + 4 + 2 + 1$

Parte fraccionaria

$$\left.\begin{array}{l} 0,125 \times 2 = 0,25 \\ 0,25 \times 2 = 0,5 \\ 0,5 \times 2 = 1,0 \\ 0,0 \times 2 = 0,0 \end{array}\right\} \; 0,0010 = 2^{-3} = 0,125$$

23,125 $010111,0010 = 23,125$ Error = 23,125 – 23,125 = 0

La representación es exacta, error = 0.

62,0625 $111110,0001 = 62,0625$ Error = 62,0625 – 62,0625 = 0

La representación es exacta, error = 0.

1,0625 $000001,0001 = 2^0 + 2^{-4} = 1,0625$ Error = 1,0625 – 1,0625 = 0

La representación es exacta, error = 0 $2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4}$

----

35 $100011,0000 = 2^5 + 2^1 + 2^0 = 32 + 2 + 1 = 35$

Error = 1,0625 – 1,0625 = 0 La representación es exacta, error = 0

---

5) El formato es BSS con 6 bits para la parte entera y 4 bits para la fraccionaria. Debemos ubicar la coma de acuerdo al formato indicado

$000000,0000 = 0$

$010101,0101 = 2^4 + 2^2 + 2^0 + 2^{-2} + 2^{-4} = 16 + 4 + 1 + 0,25 + 0,0625 = 21,3125$

$100000,0000 = 2^5 = 32$

$111111,1110 = 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} =$
$32+16+8+4+2+1+0,5+0,25+0,125 = 63,875$

$111111,1111 = 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} =$
$32+16+8+4+2+1+0,5+0,25+0,125+0,0625 = 63,9375$

$101010,1010 = 2^5 + 2^3 + 2^1 + 2^{-1} + 2^{-3} = 32 + 8 + 2 + 0,5 + 0,125 = 42,625$

011111,1111 = $2^4 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4}$ =
16 + 8 + 4 + 2 + 1+ 0,5 + 0,25 + 0,125 + 0,0625 = 31,9375

011011,0110 = $2^4 + 2^3 + 2^1 + 2^0 + 2^{-2} + 2^{-3}$ =
16 + 8 + 2 + 1 + 0,25 + 0,125 = 27,375

---

6)

<u>Sumas</u>

a)
```
    11111
   00011101     ZNVC=0000
 + 00011011
   --------
   00111000
```

b)
```
     111
    10011101    ZNVC=0001
  + 01110010
    --------
1 ← 00001111
```

c)
```
   111
   01110110     ZNVC=0110
 + 01110001
   --------
   11100111
```

d)
```
     11   11
    10111001    ZNVC=0101
  + 11100011
    --------
1 ← 10011100
```

e)
```
    11111
   00111010     ZNVC=0000
 + 00001111
   --------
   01001001
```

f)
```
     111
    01110000    ZNVC=0001
  + 11110001
    --------
1 ← 01100001
```

g)
```
   1
   01001100     ZNVC=0110
 + 01110000
   --------
   10111100
```

h)
```
    1
    11001100    ZNVC=0101
  + 11110000
    --------
1 ← 10111100
```

i)
```
    10000000    ZNVC=1011
  + 10000000
    --------
1 ← 00000000
```

j)
```
   00000000     ZNVC=0100
 + 10000000
   --------
   10000000
```

<u>Restas</u>

a)
```
   00011101     ZNVC=0000
 - 00011011
   --------
   00000010
```

b)
```
   10011101     ZNVC=0010
 - 01110010
   --------
   00101011
```

c)

```
   01110110     ZNVC=0000
 - 01110001
 ----------
   00000101
```

d)

```
1→ 10111001     ZNVC=0101
 - 11100011
 ----------
   11010110
```

e)

```
   00111010     ZNVC=0000
 - 00001111
 ----------
   00101011
```

f)

```
1→ 01110000     ZNVC=0001
 - 11110001
 ----------
   01111111
```

g)

```
1→ 01001100     ZNVC=0101
 - 01110000
 ----------
   11011100
```

h)

```
1→ 11001100     ZNVC=0101
 - 11110000
 ----------
   11011100
```

i)

```
   10000000     ZNVC=1000
 - 10000000
 ----------
   00000000
```

j)

```
1→ 00000000     ZNVC=0111
 - 10000000
 ----------
   10000000
```

---

7)
Recordar: rango en Ca2=[−128, +127] rango en BSS=[0, 255].
Si la bandera de V=1 (overflow) resultado incorrecto Ca2. Si la bandera de C=1(carry o borrow) resultado incorrecto BSS.

Sumas

	Ca2	BSS
a)	29 + 27 = 56	29 + 27 = 56
b)	- 99 + 114 = 15	157 + 114 = 15 X
c)	118 + 113 = - 25 X	118 + 113 = 231
d)	- 71 + (-29) = - 100	185 + 227 = 156 X
e)	58 + 15 = 73	58 + 15 = 73
f)	112 + (-15) = 97	112 + 241 = 97 X
g)	76 + 112 = - 68 X	76 + 112 = 188
h)	- 52 + (-16) = -68	204 + 240 = 188 X
i)	- 128 + (-128) = 0 X	128 + 128 = 0 X
j)	0 + (-128) = -128	0 + 128 = 128

a) Resultado correcto ambos sistemas
b) Resultado en Ca2 correcto.

Resultado en BSS incorrecto. Debería ser 261, fuera de rango y el resultado mostrado es 15.
c) Resultado en Ca2 incorrecto. Debería ser 118+113=231, fuera de rango y el resultado mostrado es -25.
Resultado en BSS correcto. 231 está en el rango.
d) Resultado en Ca2 correcto.
Resultado en BSS incorrecto. Debería ser 185+227=412, fuera de rango y el resultado mostrado es 156.
e) Resultado correcto en ambos sistemas.
f) Resultado en Ca2 correcto.
Resultado en BSS incorrecto. Debería ser 112+241=353, fuera de rango y el resultado mostrado es 97.
g) Resultado en Ca2 incorrecto. Debería ser 188, fuera de rango y el resultado mostrado es -68.
Resultado en BSS correcto. 188 está en el rango.
h) Resultado en Ca2 correcto.
Resultado en BSS incorrecto. Debería ser 204+240=444, fuera de rango y el resultado mostrado es 188.
i) Resultado en Ca2 incorrecto. Debería ser -128+(-128)=-256, fuera de rango y el resultado mostrado es 0.
Resultado en BSS incorrecto. Debería ser 128+128=256, fuera de rango y el resultado mostrado es 0.

<u>Restas</u>

	Ca2	BSS
a)	29 – 27 = 2	29 – 27 = 2
b)	- 99 – 114 = 43 X	157 – 114 = 43
c)	118 – 113 = 5	118 – 113 = 5
d)	- 71 – (-29) = - 42	185 – 227 = 214 X
e)	56 – 15 = 41	56 – 15 = 41
f)	112 – (-15) = 127	112 – 241 = 127 X
g)	76 – 112 = -36	76 – 112 = 220 X
h)	- 52 – (-16) = -36	204 – 240 = 220 X
i)	- 128 – (-128) = 0	128 – 128 = 0
j)	0 – (-128) = -128 X	0 – 128 = 128 X

a) Ca2 correcto. BSS correcto.
b) Ca2 incorrecto. BSS correcto.
c) Ca2 correcto. BSS correcto.
d) Ca2 correcto. BSS incorrecto.
e) Ca2 correcto. BSS correcto.
f) Ca2 correcto. BSS incorrecto.
g) Ca2 correcto. BSS incorrecto.
h) Ca2 correcto. BSS incorrecto.

i) Ca2 correcto. BSS correcto.
j) Ca2 incorrecto. BSS incorrecto.

7) 9)
Cada vez que hay V (overflow) es incorrecto el resultado en Ca2.
Cada vez que hay C (carry en la suma y borrow en la resta), es incorrecto el resultado es BSS.

En los resultados marcados con X hay condición de V ó C según corresponda.

10)

```
   111 11                            111
  00011,101   ZNVC=0000             10011,101   ZNVC=0001
+ 00011,011                       + 01110,010
  ---------                         ---------
  00111,000                     1 ← 00001,111
```

La cuenta y los flags son iguales al ej. anterior. Cambia el rango y resolución, no la cantidad de números distintos representables

---

11)

a) ___ ___ ___ , ___

000,0 = 0
111,1 = 7,5

Resolución = 000,1 = 0,5 - Números distintos = 2^4 = 16 separados 0,5
0 – 0,5 – 1 – 1,5 – 2 – 2,5 – 3 – 3,5 – 4 – 4,5 – 5 – 5,5 – 6 – 6,5 – 7 – 7,5

___ ___ , ___ ___

00,00 = 0
11,11 = 3,75

Resolución = 00,01 = 0,25 - Números distintos = 2^4 = 16 separados 0,25
0 – 0,25 – 0,5 – 0,75 – 1 – 1,25 – 1,5 – 1,75 – 2 – 2,25 – 2,5 – 2,75 – 3 – 3,25 – 3,5 – 3,75

___ , ___ ___ ___

0,000 = 0
1,111 = 1 + 0,5 + 0,25 + 0,125 = 1,875

Resolución = 0,001 = 0,125 - Números distintos = 2^4 = 16 separados 0,125
0 – 0,125 – 0,25 – 0,375 – 0,5 – 0,625 – 0,75 – 0,875 – 1 – 1,125 – 1,25 – 1,375 – 1,5 –
1,5 – 1,625 – 1,75 – 1,875

---

12)
BCD desempaquetado

0 = 11110000 = F0
1 = 11110001 = F1
9 = 11111001 = F9
10 = 11110001 11110000 = F1F0
11 = 11110001 11110001 = F1F1
1220 = 11110001 11110010 11110010 11110000 = F1F2F2F0
20 = 11110010 11110000 = F2F0
34 = 11110011 11110100 = F3F4
99 = 11111001 11111001 = F9F9
100 = 11110001 11110000 11110000 = F1F0F0

BCD empaquetado

0 = 00000000 = 00
1 = 00000001 = 01
9 = 00001001 = 09
10 = 00010000 = 10
1220 = 00010010 00100000 = 1220
20 = 00100000 = 20
34 = 00110100 = 34
99 = 10011001 = 99
100 = 00000001 00000000 = 1000

Suma en BCD

$$\begin{array}{rr} & 20 \\ + & 34 \\ \hline & 54 \end{array} \qquad \begin{array}{rr} & 0010\ \ 0000 \\ + & 0011\ \ 0100 \\ \hline & \underbrace{0101}_{5}\ \ \underbrace{0100}_{4} \end{array}$$

$$\begin{array}{rr} & 1 \\ & 34 \\ + & 99 \\ \hline & 133 \end{array} \qquad \begin{array}{rr} & 11\ \ \\ & 0011\ \ 0100 \\ + & 1001\ \ 1001 \\ \hline & 1100\ \ 1101 \end{array} \qquad \begin{array}{rr} & 1\ \ 1 \\ & 1100\ \ 1101 \\ + & 0110 \\ \hline & 1101\ \ 0011 \end{array} \qquad \begin{array}{rr} & 1 \\ & 1101\ \ 0011 \\ + & 0110 \\ \hline & 1\ 0011\ \ 0011 \end{array}$$

$$\begin{array}{rr} & 1220 \\ + & 880 \\ \hline & 2100 \end{array} \qquad \begin{array}{rr} & 0001\ \ 0010\ \ 0010\ \ 0000 \\ + & 1000\ \ 1000\ \ 0000 \\ \hline & 0001\ \ 1010\ \ 1010\ \ 0000 \end{array} \qquad \begin{array}{rr} & 1\ \ 11 \\ & 0001\ \ 1010\ \ 1010\ \ 0000 \\ + & 0110 \\ \hline & 0001\ \ 1011\ \ 0000\ \ 0000 \end{array}$$

$$\begin{array}{rr} & 1\ \ 11 \\ & 0001\ \ 1011\ \ 0000\ \ 0000 \\ + & 0110 \\ \hline & 0010\ \ 0001\ \ 0000\ \ 0000 \end{array}$$
