75.03/95.57 Organización del Computador

U1 – Sistemas de Numeración

Agenda

Metodos de representacion de números negativos

- Bit de signo y valor absoluto
- Complemento a 1
- Complemento a la base
- Complemento a 2
- Exceso

Formatos de representación de números enteros

- Binario de punto fijo sin signo
- Binario de punto fijo con signo
- BCD Empaquetado
- Zoneado

Bit de signo y valor absoluto

Base = 2 Precisión = n

x_1	x_2	.	.	.	x_n

$_2$

Signo
0 positivo
1 negativo

Valor absoluto

Ejemplo:
Base = 2 Precisión = 4

0	0	1	1	← 3
1	0	1	1	← -3

Bit de signo y valor absoluto

Paso por paso

Base = 2   precisión = 4
Representar -6
1) Ver signo para determinar el primer bit: 1 (por ser negativo)
2) Pasar valor absoluto a base 2: $|-6_{10}| = 6_{10} = 110_2$
3) Concatenar los bits: 1110

Indicar número almacenado en 1101
1) Ver primer bit para determinar el signo: negativo (por ser 1)
2) Pasar a base 10 los bits descartando el primero: $101_2 = 5_{10}$
3) Indicar el número según signo y valor obtenidos: -5

Antes de seguir veamos B^n

$B^n = ?$

base
cantidad de
simbolos

precisión
cantidad de
dítigos

x_1	x_2	.	.	.	x_n

$B \;*\; B \;*\; \ldots\ldots \;*\; B$

$= B^n$ = # total de números representables

Ejemplos:

B = 10
n = 2

$B^n = 10^2 = 100$

$\underbrace{[00, 01, \ldots.99]}_{100}$

B = 2
n = 4

$B^n = 2^4 = 16$

$$\left.\begin{array}{l}0000\\0001\\0010\\0011\\0100\\0101\\0110\\0111\\1000\\1001\\1010\\1011\\1100\\1101\\1110\\1111\end{array}\right\}16$$

Bit de signo y valor absoluto

Rango de Representación

Minimo: $-2^{n-1}-1$
Maximo: $2^{n-1}-1$

Ventajas

- Rango simétrico

Desventajas

- Doble representación del 0
- No permite operar aritméticamente

```
  0001       1
 +1001    + -1
  ----      --
  1010    ≠  0
```

Agenda

Metodos de representacion de números negativos

- Bit de signo y valor absoluto
- Complemento a 1
- Complemento a la base
- Complemento a 2
- Exceso

Formatos de representación de números enteros

- Binario de punto fijo sin signo
- Binario de punto fijo con signo
- BCD Empaquetado
- Zoneado

Complemento a 1

Base = 2 Precisión = n

x_1	x_2	.	.	.	x_n

$]_2$

Signo
0 positivo
1 negativo

Positivo -> Valor absoluto
Negativo -> Complemento (aplicar NOT) al valor absoluto

NOT
es cambiar
1 por 0
0 por 1

Ejemplo:
Base = 2 Precisión = 4

0	0	1	1	← 3
1	1	0	0	← -3

Complemento a 1

Paso por paso

Base = 2 precisión = 4

Representar -6

1. Pasar valor absoluto a base 2: $|-6_{10}| = 6_{10} = 110_2$
2. Completar con 0 a izquierda hasta completar n: 0110
3. Si es negativo, complementar (hacer NOT): 1001

Indicar número almacenado en 1101

1. Si primer bit es 1 (es negativo), complementar (hacer NOT): 0010
2. Pasar a base 10: $0010_2 = 2_{10}$
3. Indicar el número según signo y valor obtenidos: -2

Complemento a 1

Rango de Representación

Minimo: $-2^{n-1}-1$
Maximo: $2^{n-1}-1$

Desventajas

- Doble representación del 0

Ventajas

- Rango simétrico
- Permite operar aritméticamente sumando el "end-around carry"

```
11
 0101      5
+1110    + -1
 ----      --
 0011    ≠  4
 +   1
 ----
 0100    =  4
```

Agenda

Metodos de representacion de números negativos

- Bit de signo y valor absoluto
- Complemento a 1
- Complemento a la base
- Complemento a 2
- Exceso

Formatos de representación de números enteros

- Binario de punto fijo sin signo
- Binario de punto fijo con signo
- BCD Empaquetado
- Zoneado

Complemento a la Base

Base = B Precisión = n

$Rep(x_b) = x_b$ si $x \geq 0$
$Rep(x_b) = Cb(|x_b|)$ si $x < 0$

Ejemplos:

$B = 10 \quad n = 2 \Rightarrow B^n = 10^2 = 100$

Rep(3) = 3
Rep(-3) = Cb(3) = 100 – 3 = 97
Rep(-1) = Cb(1) = 100 – 1 = 99
Rep(-50) = Cb(50) = 100 – 50 = 50
Rep(50) => NO SE PUEDE
=> 49 es el mayor positivo representable

¿ Pero que es Cb?

$$Cb(r) + r = B^n$$
$$\Longrightarrow Cb(r) = B^n - r$$

Notar que:
Si
k es complemento de r
=> $k + r = B^n$

=> r es complemento de k

Complemento a la Base

Rango de Representación

Mínimo: $-(B^n/2)$
Máximo: $(B^n/2) - 1$

Desventajas

- Rango asimétrico (un negativo más)

Ventajas

- Única representación del 0
- Permite operar aritméticamente

Complemento a la Base

Permite operar aritméticamente:

Sumas

```
  5 + 2   (es 7)

  5 + 2
= 5 + Cb(98)

   0
    5
  + 2
   ---
   07   (A)
```

```
  5 + (-2)   (es 3)

  5 + 98
= 5 + Cb(2)

     11
       5
    + 98
     ---
     103 (B)
```

```
  -5 + 2   (es -3)

  95 + 2
= 95 + Cb(98)

   0
   95
 +  2
  ---
  097   --> -3 (C)
```

```
  -5 + (- 2)   (es -7)

  95 + 98
= 95 + Cb(2)

   11
    95
  + 98
   ---
   193   --> -7 (D)
```

Restas

```
  5 - 2     (es
3)
= 5 + (-2)

  5 - 2
= 5 + 98
= 5 + Cb(2)
 (B)
```

```
  5 - (-2)   (es 7)
= 5 + 2

  5 - 98
= 5 + 2
= 5 + Cb(98)
       (A)
```

```
  -5 - 2   (es -7)
= -5 + (-2)

  95 - 2
= 95 + 98
= 95 + Cb(2)
    (D)
```

```
  -5 - (- 2)   (es -3)
= -5 +  2

  95 - 98
= 95 + 2
= 95 + Cb(98)
    (C)
```

Complemento a la Base

Permite operar aritméticamente: CONCLUSIÓN

A - B

Se trabaja como

A + Comp(B)

***** NO IMPORTA EL SIGNO DE B ******

Agenda

Metodos de representacion de números negativos

- Bit de signo y valor absoluto
- Complemento a 1
- Complemento a la base
- Complemento a 2
- Exceso

Formatos de representación de números enteros

- Binario de punto fijo sin signo
- Binario de punto fijo con signo
- BCD Empaquetado
- Zoneado

Complemento a 2

Base = 2 Precisión = n

$Rep(x_{10}) = x_2$ si $x \geq 0$

$Rep(x_{10}) = NOT(|x_2|) + 1$ si $x < 0$

Ejemplos:

B = 2   n = 4

Rep (3) = 0011 Rep(-3) = NOT(0011) + 1 = 1100 + 1 = 1101 Rep(-1) = NOT(0001) + 1 = 1110 + 1 = 1111	Rep(-8) = NOT(1000) + 1 = 0111 + 1 = 1000 Rep(8) => NO SE PUEDE => 7 = $0111_{\vert 2}$ es el mayor positivo representable

Veamos...

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

```
 0001
+1111
 ----
10000
```

```
 0011
+1101
 ----
10000
```

```
 1000
+1000
 ----
10000
```

NOT + 1
es Cb
con b=2

Complemento a 2

Paso por paso

Base = 2   precisión = 4

Representar -6

1. Pasar valor absoluto a base 2: $|-6_{10}| = 6_{10} = 110_2$
2. Completar con 0 a izquierda hasta completar n: 0110
3. Si es negativo, complementar (hacer NOT + 1): 1001 + 1 = 1010

Indicar número almacenado en 1101

1. Si primer bit es 1 (es negativo), complementar (hacer NOT+1): 0010 + 1 = 0011
2. Pasar a base 10 los bits: $0011_2 = 3|_{10}$
3. Indicar el número según signo y valor obtenidos: -3

Complemento a 2

Rango de Representación

Mínimo: -2^{n-1}
Máximo: $2^{n-1} - 1$

Desventajas

- Rango asimétrico (un negativo más)

Ventajas

- Única representación del 0
- Permite operar aritméticamente.
  Aplica la misma mecánica de resolver X-Y como X+Cb(Y)

Agenda

Metodos de representacion de números negativos

- Bit de signo y valor absoluto
- Complemento a 1
- Complemento a la base
- Complemento a 2
- Exceso

Formatos de representación de números enteros

- Binario de punto fijo sin signo
- Binario de punto fijo con signo
- BCD Empaquetado
- Zoneado

Exceso a la base

Base = B Precisión = n

Rep(x_b) = x_b + exceso (para todo x)

Con B = 2

```
Exceso = 2^n/2
       = 2*2 . . .* ~~2/2~~
       = 2^(n-1)
```

$$\text{Exceso} = 2^n/2 = 2 * 2 \ldots * \cancel{2/2} = \mathbf{2^{n-1}}$$

Ejemplos:

B = 10 n = 2

=> $B^n/2 = 10^2/2 = 50$

Rep (3) = 3 + 50 = 53
Rep(-3) = -3 + 50 = 47
Rep(0) = 0 + 50 = 50
Rep(-50) = -50 + 50 = 0
Rep(-51) NO SE PUEDE (da –1)
Rep (49) = 49 + 50 = 99
Rep(50) NO SE PUEDE (da 100)

¿ Pero que es el exceso?

Es $B^n/2$

Entendamos por qué:
Con B=10 y n = 1
B^n = 10 valores posibles

¿Cuál sería el rango de números a representar "más justo"?

$$\underbrace{-5 \quad -4 \quad -3 \quad -2 \quad -1 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4}_{\text{10 en total}}$$

Cuánto hay q sumar como mínimo a cada negativo para que "desaparezca" el signo?
5
Que termina siendo **$B^n/2$**

Números a representar

-5	-4	-3	-2	-1	0	1	2	3	4

Como se representan

0	1	2	3	4	5	6	7	8	9

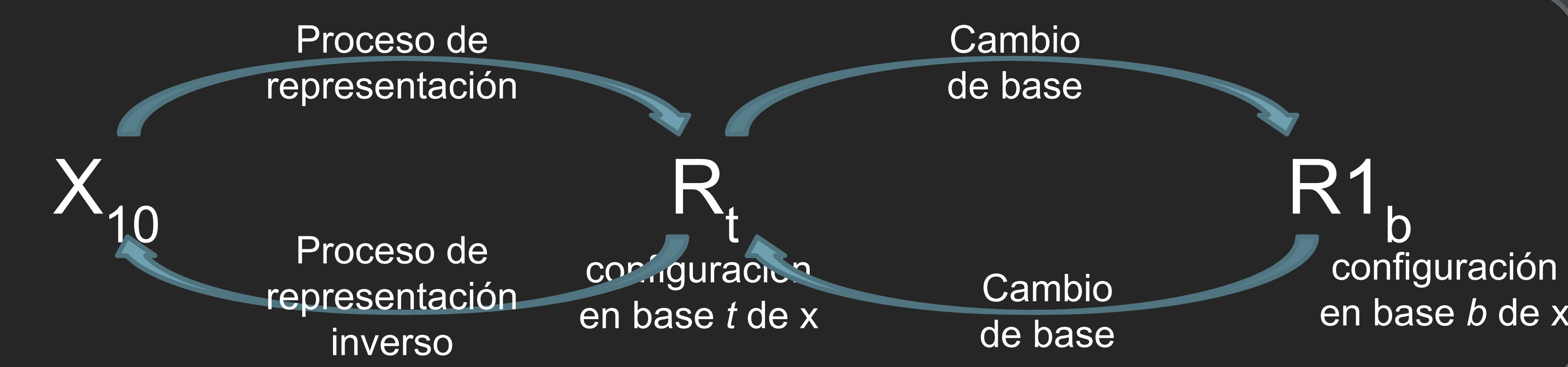
Formato y Configuración

Formato:
Representación <u>computacional</u> de un número

$X_{10} \rightarrow$ Proceso $\rightarrow R_t$

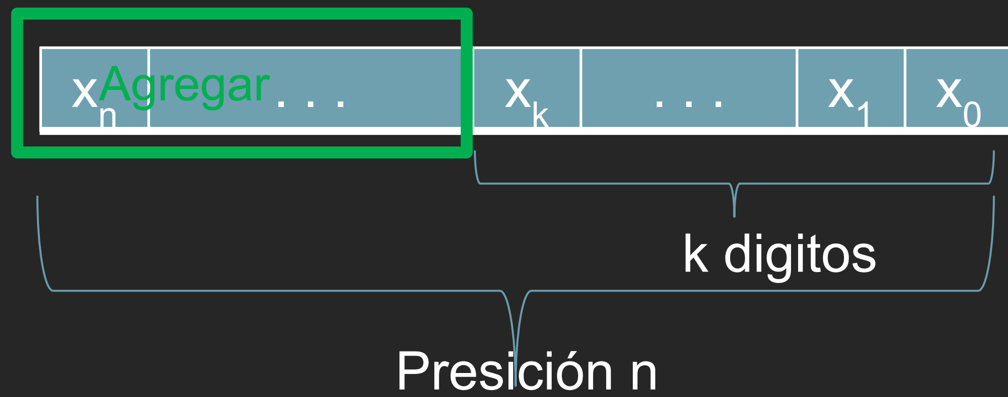
Configuración:
Expresión en una determinada base de un número en un formato

$R_t \rightarrow$ Cambio de base $\rightarrow R1_b$



Expansión y Truncamiento

Expansión:



Truncamiento:

Agenda

Metodos de representacion de números negativos

- Bit de signo y valor absoluto
- Complemento a 1
- Complemento a la base
- Complemento a 2
- Exceso

Formatos de representación de números enteros

- Binario de punto fijo sin signo
- Binario de punto fijo con signo
- BCD Empaquetado
- Zoneado

Binario de punto fijo sin signo

Base = 2   Precisión = n   Enteros positivos

Como almacenar un número
1) Pasar el nro a base 2
2) Completar con 0 a izquierda hasta alcanzar n digitos

Como recuperar un número almacenado
Pasos anteriores en orden inverso

Rango de representación

Minimo: 0
Máximo: $2^n - 1$

Agenda

Metodos de representacion de números negativos

- Bit de signo y valor absoluto
- Complemento a 1
- Complemento a la base
- Complemento a 2
- Exceso

Formatos de representación de números enteros

- Binario de punto fijo sin signo
- Binario de punto fijo con signo
- BCD Empaquetado
- Zoneado

Binario de punto fijo con signo

Base = 2   Precisión = n   Enteros positivos y negativos

Es la implementación del método complemento a 2

Como almacenar un número
1) Pasar el nro a base 2
2) Completar con 0 a izquierda hasta alcanzar n digitos
3) Si el nro es negativo, complementar usando método de "complemento a 2" (Not +1)

Como recuperar un número almacenado
1) Si el primer bit es 1 (es negativo), complementar.
2) Quitar 0 a izquierda.
3) Pasar a base 10 y colocar el signo que corresponda.

Binario de punto fijo con signo

Validación Overflow en operaciones aritméticas

B=2 n=4

Resolver 7 + 1

$7_{|10} = 0111_{|2}$

$1_{|10} = 1_{|2}$

```
  0111
  0111
+ 0001
  ----
  1000
```

Ultimos 2 acarreos distintos
=> OVERFLOW

Resolver 7 - 1

$7_{|10} = 0111_{|2}$

$1_{|10} = 1_{|2}$

Hallo C(1) para hacer 7+C(1)

```
NOT(0001) = 1110
          +    1
            ----
            1111
```

Ahora sumo

```
  1111
  0111
+ 1111
  ----
  0110
```

Ultimos 2 acarreos iguales
=> VALIDO

Agenda

Metodos de representacion de números negativos

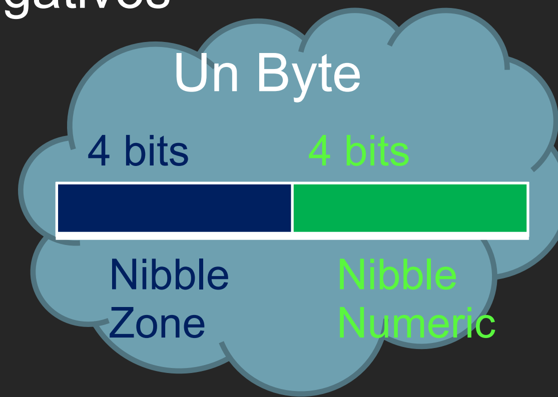
- Bit de signo y valor absoluto
- Complemento a 1
- Complemento a la base
- Complemento a 2
- Exceso

Formatos de representación de números enteros

- Binario de punto fijo sin signo
- Binario de punto fijo con signo
- BCD Empaquetado
- Zoneado

BCD Empaquetado

Base = 16   Precisión = n   Enteros positivos y negativos



Como almacenar un número

1) Pasar el nro a base 10
2) Colocar c/digito en los nibbles dejando libre el último (el de la derecha)
3) Colocar en el último nibble el signo siendo
   - C, A, F o E para positivos
   - B o D para negativos

Ej. n=3 $+123_{|10}$ --> $00123A_{|16}$ $-456_{|10}$ --> $00456B_{|16}$

Como recuperar un número almacenado

1) Tomar cada digito de los nibbles (excepto el último) para armar la cadena en base 10
2) Colocar el signo según el dígito del último nibble

Agenda

Metodos de representacion de números negativos

- Bit de signo y valor absoluto
- Complemento a 1
- Complemento a la base
- Complemento a 2
- Exceso

Formatos de representación de números enteros

- Binario de punto fijo sin signo
- Binario de punto fijo con signo
- BCD Empaquetado
- Zoneado

Zoneado

Base = 16   Precisión = n   Enteros positivos y negativos

Como almacenar un número
1) Pasar el nro a base 10
2) Colocar c/digito en los nibbles numeric
3) Colocar una F en cada nibble zone excepto en el último
3) Colocar en el último nibble zone el signo siendo
    C, A, F o E para positivos
    B o D para negativos

Ej. n=4 $+123_{|10}$ --> $F0F1F2A3_{|16}$ $-456_{|10}$ --> $F0F4F5B6_{|16}$

Como recuperar un número almacenado
1) Tomar cada digito de los nibbles (excepto el último) para armar la cadena en base 10
2) Colocar el signo según el dígito del último nibble

Preguntas

?