

REPRESENTACIÓN DE NÚMEROS ENTEROS

Programación en Fortran

Valentín Moreno

1

REPRESENTACIÓN BASADA EN UN NÚMERO FIJO DE BITS

- Los ordenadores digitales generalmente emplean cuatro métodos distintos para representar números enteros:
 - Módulo y signo: MS.
 - Complemento a Uno: C-1.
 - Complemento a Dos: C-2.
 - Exceso a Dos Elevado a $(N-1)$.
- Emplean el sistema binario y tienen un número limitado de dígitos o bits (palabra) para cada dato numérico, que es potencia de dos, que depende de cada ordenador y se representa por N .

MÓDULO Y SIGNO

- El bit situado a la izquierda indica el signo del número (no tiene valor, sólo indica el signo del número). Signo:
 - 0 : número positivo.
 - 1 : número negativo.
- Los (N-1) bits restantes se usan para el módulo del número.
- Problema: El cero tiene dos representaciones (0000 0000 y 1000 0000, para $N = 8$).
- Ejemplo: Para $N = 8$ representar -10 y 10.

COMPLEMENTO A UNO: C-1

- El bit de la izquierda indica el signo del número pero sí tiene valor.
- Representación:
 - Número Positivo: Los $(N-1)$ bits restantes representan el módulo del número (como en MS).
 - Número Negativo: se escribe el número positivo correspondiente y se complementan todos sus dígitos, incluido el bit del signo.
- Problema: el cero tiene dos representaciones (0000 0000 y 1111 1111, para $N = 8$).
- Ejemplo: Para $N = 8$ representar -10 y 10

COMPLEMENTO A DOS: C-2

- El bit de la izquierda representa el signo del número.
- Representación:
 - Número Positivo: Los $(N-1)$ bits restantes representan el módulo del número (como en MS).
 - Número Negativo: Se complementan todos los bits del número positivo (como en C-1) y luego se le suma 1 bit en binario, despreciándose el último acarreo si lo hay.
- Ventaja: Existe una única representación del cero (0000 0000, para $N = 8$).
- Ejemplo: Para $N = 8$ representar -10 y 10

EXCESO A DOS ELEVADO A (N-1)

- No hay ningún bit reservado para el signo.
- Todos los bits representan un módulo correspondiente al número representado más el exceso a $2^{(N-1)}$.
- Ventaja: El cero tiene una única representación (1000 000, para $N=8$).
- Ejemplo: Para $N = 8$ representar -10 y 10.

EJERCICIOS

- Representar en MS, C-1, C-2 y Exceso a $2^{(N-1)}$ para $N=8$:
 - 45 y -45
 - 110 y -110

REPRESENTACIÓN BASADA EN NÚMERO VARIABLE DE BYTES

- Los bytes a utilizar dependerán de la cantidad de dígitos que tenga el entero.
- Siempre existe un máximo predeterminado de bytes (determina el número de mayor valor absoluto).
- Existen dos representaciones:
 - Decimal desempquetado
 - Decimal empaquetado

DECIMAL DESEMPAQUETADO

- Cada dígito hexadecimal se almacena en un byte (8 bits).
- Bits de dígito o cuarteto de dígito: De cada byte se utilizan sólo los 4 bits de la derecha para representar la cifra correspondiente en *decimal codificado en binario* (BCD).
- Bits de zona o cuarteto de zona: De cada byte, el cuarteto de la izquierda se rellenan con una combinación prefijada de bits (1111 o F, en hexadecimal).
- El cuarteto de la izquierda de la última cifra representa el signo: 1100 o C significa positivo y 1101 o D negativo.

DECIMAL EMPAQUETADO

- Cada dígito se representa en un cuarteto (sin bits de zona).
- El primer cuarteto por la derecha lleva el signo con los valores que conocemos:
 - 1100 (C, en hexadecimal), para los positivos.
 - 1101 (D, en hexadecimal), para los negativos.
- Si es necesario se completa el primer byte con un cuarteto de ceros (0000).

EJERCICIOS

- Representar en hexadecimal empaquetado y desempaquetado:
 - 1995 y -1995
 - 30001 y -30001