Ingeniería en Computación

Clase 3.
Curso 2021

En esta clase:

- Representación binaria con signo
  - Signo y Magnitud / Signo y Módulo (SM)
  - Complemento a 1 (CA1)
  - Complemento a 2 (CA2)
- Rangos de representación.
- Ventajas y desventajas entre los tres sistemas de representación.
- adición, substracción multiplicación y división binarias en CA2.

Representación binaria con signo

- Para poder representar los números con signo utilizando solamente dos símbolos (0 y 1) es necesario codificar el signo del número:
  - Normalmente se codifica el signo con un 0 para números positivos y un 1 para números negativos
  - Se utilizan 3 diferentes sistemas de representación, que varían según las diferentes formas de codificar la magnitud:
    - Signo y magnitud (SM)
    - Complemento a 1 (CA1)
    - Complemento a 2 (CA2)

Representación binaria con signo

SIGNO Y MAGNITUD

- Dado que sólo es posible representar la **magnitud** con un número binario, el signo (+) o (-) puede mostrarse agregando un **bit de signo**.
  - El bit de signo 0 indica un número positivo
  - El bit de signo 1 indica un número negativo
  - Ejemplo:

    $25_{10} = 11001_{BIN}$ (Binario Natural)

    $+25_{10} = 011001_{SM}$ (Módulo y signo)

    $-25_{10} = 111001_{SM}$ (Módulo y signo)

Representación binaria con signo

COMPLEMENTO a 1

- Si el número es positivo:
  - El MSB es un 0 (signo)
  - El resto de los bits son la magnitud en binario natural
- Si el número es negativo:
  - El MSB es un 1 (signo)
  - El resto de los bits son el complemento (a 1) de la magnitud

Ca1(A) = \A

La operación \A Se lee como "complemento a 1 de A" o más comunmente "A negado". Significa que se invierten todos los bits de A, incluyendo el de signo.

Equivalentemente, se puede definir como (siendo n el número de bits):

Ca1(A) *= 2^n –1 – A = (2^n −1)− A =11....11− A =* ***\A***

- Ejemplos:

$+25_{10}$ = 011001 $_{Ca1}$ (Complemento a 1)

-25_{10} = 100110 $_{Ca1}$ (Complemento a 1)

Representación binaria con signo

COMPLEMENTO a 2

- Si el número es positivo:
  - El MSB es un 0 (signo)
  - El resto de los bits son la magnitud en binario natural
- Si el número es negativo:
  - El MSB es un 1 (signo)
  - El resto de los bits son el complemento a 2 de la magnitud.
  - El complemento a dos de un número es su complemento a 1, + 1

Ca2(A) = Ca1(A)+1 = \A+1

Equivalentemente, se puede definir como(siendo n el número de bits):

Ca2(A) *= 2^n – A = 2^n − A + 1 – 1= (2^n −1)+1− A =11....11− A+1 =* ***\A+1***

Ejemplos:

$+25_{10} = 011001_{Ca2}$

$-25_{10} = 100111_{Ca2}$

Representación binaria con signo

COMPLEMENTO a 2

- No deben confundirse los siguientes conceptos
- “Operación de complementar a 2”:
  - ***Ca2(A) =   Ca1(A)+1 = \A+1***
- “Representación en complemento a 2”
  - Es el valor que representa un numero dado, que se obtiene siguiendo las siguientes reglas que permiten representar dos conjuntos de números:
    - Si el número es positivo, se utiliza la representación binaria natural, si el número se va a representar con n bits, el MSB debe ser “0”.
    - Si el numero a representar es negativo, se escribe su valor positivo y se aplica la operación de complementar a 2.
- La unión de ambos conjuntos es la representación en complemento a 2 utilizando n bits.
- La representación Complemento a 2 es la más utilizada en sistemas digitales.

Representación binaria con signo

COMPLEMENTO a 2

- Otra forma de realizar la operación de complementar a 2 es la siguiente:
- Comenzando por el LSB, copiar los bits hasta encontrar el primer 1, e invertir el resto:
- EJEMPLO:

Ca2(11000110)= 00111010

Comprobación:

Ca2(11000110) = 00111001+1= 00111010

Representación binaria con signo

Extensión del número de bits

- Un mismo número puede representarse con diferente número de bits:

$+25_{10} = 011001_{SM} = 00011001_{SM} = 0000000000011001_{SM}$

$+25_{10} = 011001_{CA1} = 00011001_{CA1} = 0000000000011001_{CA1}$

$+25_{10} = \underbrace{011001_{CA2}}_{\text{6 bits}} = \underbrace{00011001_{CA2}}_{\text{8 bits}} = \underbrace{0000000000011001_{CA2}}_{\text{16 bits}}$

- Si los números son positivos como en el ejemplo, en los tres sistemas se añaden ceros a la izquierda (en rojo) hasta completar el número de bits necesarios
- Nótese que el MSB se mantiene en 0 indicando el signo (positivo)

Representación binaria con signo

Extensión del número de bits

- En el caso de Números Negativos:
- Para representación SM, mantener el MSB que indica el signo y añadir ceros para completar el número de bits.
- Para representación Ca1 y Ca2 añadir unos a la izquierda del número hasta completar la cantidad de bits necesarios

$-25_{10} = 111001_{SM} = 10011001_{SM} = 1000000000011001_{SM}$

$-25_{10} = 100110_{CA1} = 11100110_{CA1} = 1111111111100110_{CA1}$

$-25_{10} = \underbrace{100111}_{6\text{ bits}}{}_{CA2} = \underbrace{11100111}_{8\text{ bits}}{}_{CA2} = \underbrace{1111111111100111}_{16\text{ bits}}{}_{CA2}$

Representación binaria con signo

Comparación:

- Vemos cómo la misma codificación en 3 bits representa distintos números en los tres sistemas
- Puede notarse que SM y Ca1 tienen dos representaciones del cero mientras que en Ca2 la representación del cero es única

Codificación	SM	Ca1	Ca2
000	0	0	0
001	1	1	1
010	2	2	2
011	3	3	3
100	-0	-3	-4
101	-1	-2	-3
110	-2	-1	-2
111	-3	-0	-1

Rangos de representación.

Cuál es el rango de valores que puede representarse con 8 bits?

- **Binario sin signo: $2^n = 2^8 = 256$ valores distintos (0 a 255)**

  $00000000=0_{10}$ al $11111111=255_{10}$

- **Magnitud y signo: $2^n - 1 = 2^8 - 1 = 255$ valores distintos**

  ($-127 \rightarrow -0$ y $0 \rightarrow 127$)

  $11111111 = -127_{10} \rightarrow 10000000 = -0_{10}$

  $00000000 = 0_{10} \rightarrow 01111111 = 127_{10}$

- **Ca1 : $2^n - 1 = 2^8 - 1 = 255$ valores distintos ($-127 \rightarrow -0$ y $0 \rightarrow 127$)**

  $10000000_{Ca1} = -127_{10} \rightarrow 11111111_{Ca1} = -0_{10}$

  $00000000_{Ca1} = 0_{10} \rightarrow 01111111_{Ca1} = 127_{10}$

- **Ca2 : $2^n = 2^8 = 256$ valores distintos ($-128 \rightarrow 0 \rightarrow 127$)**

  $1000000_{Ca2} = -128_{10} \rightarrow 00000000_{Ca2} = 0_{10} \rightarrow 01111111_{Ca2} = 127_{10}$

Ventajas y desventajas de los distintos sistemas de representación.

- Módulo y Signo y Ca1 tienen dos representaciones del cero.
  - Esto ocasiona que tengo dos condiciones distintas para verificar un mismo número → aumenta la complejidad
- Las operaciones de suma y resta en módulo y signo son complicadas, en particular no se verifica que x-z = x+(-z)
- Ca2 tiene una unica representación del cero,esto me permite representar un valor más.
- En Ca1 y Ca2 se cumple que x-z = x+(-z). Esto permite simplificar las unidades aritméticas

Operaciones en complemento a 2

SUMA

- Realizar la suma binaria normal de magnitudes.
  - Los bits de signo son sumados con los bits de magnitud.
- Si de la adición resulta un acarreo más alla del MSB, se descarta
  - Si el MSB es cero, entonces el resultado es positivo.
  - Si el MSB es uno el resultado es negativo y se encuentra representado en Ca2.
- **Overflow o desborde:**
  - Puede ocurrir sólo cuando se suman dos números positivos o negativos, y el resultado excede el rango de representación.
  - Cuando ocurre overflow, el signo del resultado es opuesto al signo de los operandos.

SUMA. Ejemplos

- Consideremos números representados en 5 bits

Dos números positivos. (+9) + (+4) 0 1001 (+9) + 0 0100 (+4) 0 1101 (+13)	Número positivo mayor en módulo que número negativo. (+9) + (–4) 0 1001 (+9) + 1 1100 (–4) 1|0 0101 (+5)	Dos números iguales y de signo contrario. (+9) + (–9) 0 1001 (+9) + 1 0111 (–9) 1|0 0000 (0)
Dos números negativos (–9) + (–4) 1 0111 (–9) + 1 1100 (–4) 1|1 0011 (–13)	Número positivo menor en módulo que número negativo. (+4) + (–9) 0 0100 (+4) + 1 0111 (–9) 1 1011 (–5)	OVERFLOW (+9) + (+9) 0 1001 (+9) + 0 1001 (+9) 1 0010 (-14)

Operaciones en complemento a 2

RESTA.

- Se utiliza la propiedad $x - z = x + (-z)$
- Si se utiliza representación Ca2 de n bits, entonces:

  $-z_{Ca2} = Ca2(z_{Ca2})$ dónde $Ca2(z)$ es la operación Ca2.

  por lo tanto $x_{Ca2} - z_{Ca2} = x_{Ca2} + Ca2(z_{Ca2})$

  o bien $x_{Ca2} - z_{Ca2} = x_{Ca2} + \backslash z_{Ca2} + 1$

  ($\backslash z_{Ca2}$ es la operación complemento a 1 o negación)

- En los sistemas digitales modernos se representan los números con signo en Ca2 y se implementa la resta a través de sumadores y negadores.

Operaciones en complemento a 2

MULTIPLICACIÓN

- Se realiza de manera similar a la multiplicación en base 10.
- En el caso de Números con signo representados en complemento a 2:
  - Si los dos números son positivos, se realiza la operación y se añade un bit de signo 0 (el resultado es positivo)
  - Si los dos números son negativos, se realiza la operación Ca2 sobre ambos números, se multiplican y al resultado se le añade un bit de signo 0 (el resultado es positivo).
  - Si uno de los números es negativo se hace la operación de Ca2 en ese número, se multiplican y al resultado se le vuelve a hacer la operación de Ca2 (el resultado es negativo). Si es necesario se extiende el signo.
  - Si los operandos están representados en n bits, el producto podrá llegar a necesitar 2n bits para su representación.

Operaciones en complemento a 2

MULTIPLICACIÓN (multiplicando y multiplicador, números positivos representados en 4 bits. Producto representado en 8 bits)

Ejemplo; 9 x 10

```
      1001  (9)
    x 1010 (10)
-----------------
      0000
     1001
    0000
   1001
 ---------------
  01011010  (90)
```

Ejemplo; 1 x 1

```
      0001  (1)
    x 0001 (1)
-----------------
      0001
     0000
    0000
   0000
 ---------------
  00000001  (1)
```

Ejemplo; 15 x 15

```
      1111 (15)
    x 1111 (15)
-----------------
      1111
     1111
    1111
   1111
 ---------------
  11100001 (225)
```

Operaciones en complemento a 2

MULTIPLICACIÓN

- Multiplicando y multiplicador, números positivos y negativos representados en Ca2 de 4 bits

Ejemplo: un factor positivo y el otro negativo: –3 x 6

$-3=1101_{Ca2} \rightarrow +3=0011_{Ca2}$

```
      0011  (+3)
    x 0110  (+6)
   ____________
      0000
     0011
    0011
   0000
   ____________
   00010010  (+18)
```

Aplico:
Ca2(00010010)=
$= 11101110_{Ca2} = -18$

Ejemplo: dos factores negativos (–1) x (–2)

$-1=1111_{Ca2} \rightarrow +1=0001_{Ca2}$
$-2=1110_{Ca2} \rightarrow +2=0010_{Ca2}$

```
      0001  (+1)
    x 0010  (+2)
   ____________
      0000
     0001
    0000
   0000
   ____________
   00000010  (+2)
```

Operaciones en complemento a 2

DIVISIÓN o COCIENTE

- Tambien puede realizarse de la forma que se realiza para el sistema decimal
- Por ejemplo hagamos 25 / 4 y utilicemos 6 bits para representar los operandos.
- 25_{10} =011001 4_{10} =000100
- Dividendo → 11001 /100 → Divisor

```
Dividendo →  11001  /100 → Divisor
            – 100    000110 → Cociente
              01000
             –100
              00001
             – 000
             00001→ Resto
```

Se obtiene un cociente 6 y un resto 1

Operaciones en complemento a 2

DIVISIÓN o COCIENTE

- Para las operaciones de división también valen las consideraciones que se tomaron en cuenta para la multiplicación respecto a los signos de los operandos y resultados
- Si dividendo y divisor son del mismo signo, entonces el cociente resultará positivo
- Si son de distinto signo, el cociente resultará negativo.
- Si el dividendo tiene m bits y el divisor tiene n bits, entonces el cociente tendrá (m-n) bits y el resto tendrá a lo sumo n bits.

Resumen

- Se vieron tres formas de representar números binarios con signo:
  - Signo y Módulo (SM)
  - Complemento a 1 (Ca1)
  - Complemento a 2 (Ca2)

  Se enumeraron ventajas y desventajas de las distintas representaciones.
- Se vieron como se efectuan las operaciones básicas en el sistema más utilizado que es Ca2.