Tema 6. Representación digital de datos

6.0. Documentación

💡 Documentos Tema 6:

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f69c306f-7fda-462e-8288-fffda3faa975/U6__RepresentacionDatos.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/53435b33-473c-4d73-8fb9-0721a0bb580c/U6_Datos_Enunciados.pdf

6.1. Representación de números enteros

Un número binario natural, entero sin signo, utiliza un sistema numérico posicional. Con un número n de bits se pueden representar 2^n número diferentes en el rango $[0, 2^n - 1]$.

Cuando el número binario tiene signo, para un número de n bits, el bit más significativo señala el signo (0 positivo y 1 negativo), mientras que los $n - 1$ bits restantes la magnitud.

6.1.1. Complemento a 2

La **codificación** en complemento a 2 representa el valor de un número en un sistema binario posicional, en un número de n bits. El bit más significativo tiene el valor -2^{n-1}.

La **operación** de complemento a 2 de número entero equivale a la inversión de signo de un número. Consiste en invertir todos sus bits y sumar 1 al resultado.

6.1.2. Problemas de desbordamiento

Los sistemas digitales operan con un número fijo de bits. El resultado de una operación suma puede sobrepasar el rango de representación de los bits utilizados. Para ello se realiza la operación XOR de los do último **acarreos**, si el resultado es 0, el resultado es correcto.

6.2. Sumador binario

6.2.1. Semisumador (1 bit)

Un **semisumador** es un circuito combinacional para la suma aritmética de los dos bits de la entrada, obteniendo a la salida de un bit para la suma y un bit para el acarreo.

- Ecuación para el bit de suma: $Suma = b_1 \oplus b_2$
- Ecuación para el bit de acarreo: $C_{out} = b_1 \cdot b_2$

6.2.2. Sumador completo (1 bit)

Un **sumador completo** es un circuito combinacional para la suma de los dos bits de la entrada mas el acarreo del bit anterior.

- Ecuación para el bit de suma: $Suma = (b_1 \oplus b_2) \oplus C_{in}$
- Ecuación para el bit de acarreo: $C_{out} = b_1 \cdot b_2 + (b_1 \oplus b_2) \cdot C_{in}$

6.2.3. Circuito sumador para n bits en paralelo con acarreo en serie

El circuito completo presenta un circuito sumador elemental para cada bit, los bits del mismo peso se suman dos a dos y para obtener cada suma parcial se necesita el acarreo que se produce en la suma precedente.

6.3. Representación de números reales

En la representación de números reales el número de bits dedicados a la parte entera y a la parte fraccionaria es fijo. Los números negativos se representan en complemento a 2. Existe un problema de precisión.

6.3.1. Coma flotante

Un número se representa con tres grupos de bits: signo, exponente y mantisa; siguiente el estándar IEEE-754.

- Signo (1 bit): indica el signo del número, 0 si es positivo y 1 si es negativo.
- Exponente (8 bits): indica el exponente de una potencia en base 2.
- Mantisa (23 bits): indica el valor absoluto del número, multiplicando a la potencia.

Existen números de precisión simple (32 bits) y de precisión doble (64 bits).

6.4. Otros códigos binarios

6.4.1. Código decimal binario (BCD)

Es una representación para números enteros sin signo, en la que cada dígito decimal tiene su equivalente binario en 4 bits.

6.4.2. Representación de texto (ASCII)

ASCII es un código binario que se utiliza para representar caracteres. El código ASCII extendido utiliza 8 bits para diferentes caracteres adicionales a un alfabeto tradicional.

EJERCICIOS:

1. Escribir $-954,625$ con coma fija (16,4) en codificación signo magnitud.

Dedicamos 1 bit al signo (signo magnitud), 11 bits a la parte entera del número y 4 bits a la parte fraccionaria; 16 bits en total.

$954 = 512 + 256 + 128 + 32 + 16 + 8 + 2 = 2^9 + 2^8 + 2^7 + 2^5 + 2^4 + 2^3 + 2^1 \rightarrow 01110111010$

$0,625 * 2 = 1,250; 0,250 * 2 = 0,5; 0,5 * 2 = 1; 0,625 \rightarrow 1010$

Luego, $-954,625 \rightarrow 1\ 01110111010\ 1010$

2. Escribir $-954,625$ en codificación complemento a 2 (16,4).

Tomamos el número en binario natural: $954,625 \rightarrow 0011101110101010$

Invertimos todos los bits: 1100010001010101

Sumamos 1, porque el número es negativo: $-954,625 \rightarrow$
$1100010001010101 + 1 = 1100010001010110$

3. Escribir $-954,625$ en estándar de coma flotante con precisión simple.

Convertimos el número a binario natural: $954,625 \rightarrow 001110111010,1010$

Pasamos el número a notación científica: $001110111010,1010 =$
$1,1101110101010 \times 2^{9}$

Codificamos el exponente: $127 + 9 = 136 = 128 + 8 = 2^{7} + 2^{3} = 10001000$

La mantisa son los dígitos “detrás de la coma”: 1101110101010

Luego, $-954,625 \rightarrow 1\ 10001000\ 00000000110111010101000$.

En hexadecimal: $C46EA8004$