DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
ESCUELA DE INGENIERÍA
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

IIC2343 - Arquitectura de Computadores (I/2025)

Ayudantía 1

Ayudantes: Daniela Ríos (danielaarp@uc.cl), Joaquín Peralta (jperaltaperez@uc.cl)

Pregunta 1: Representación de números

(a) Convierta los siguientes números decimales a binario:

1. 78_{10}
2. 182_{10}

Solución:

Para pasar los números a binario usaremos la siguiente tabla de potencias de 2.

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
512	256	128	64	32	16	8	4	2	1	0.5	0.25	0.125	0.0625

1. 78_{10}:

   Buscamos la potencia de 2 más grande que sea menor o igual a 78.

   $$78 - 2^6 = 78 - 64 = 14$$

   Ahora hacemos lo mismo pero con 14, y repetimos hasta llegar a 0.

   $$14 - 2^3 = 14 - 8 = 6$$
   $$6 - 2^2 = 6 - 4 = 2$$
   $$2 - 2^1 = 2 - 2 = 0$$

   Entonces, 78 se puede escribir como:

   $$78 = 2^6 + 2^3 + 2^2 + 2^1$$
   $$78 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

Esto en binario,

$$78_{10} = 1001110_2$$

2. 182_{10}:

   La mayor potencia de 2 menor o igual a 182 es 2^7.

$$182 - 2^7 = 182 - 128 = 54$$
$$54 - 2^5 = 54 - 32 = 22$$
$$22 - 2^4 = 22 - 16 = 6$$
$$6 - 2^2 = 6 - 4 = 2$$
$$2 - 2^1 = 2 - 2 = 0$$

   Entonces, 182 se puede escribir como:

$$182 = 2^7 + 2^5 + 2^4 + 2^2 + 2^1$$
$$182 = 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

   Esto en binario,

$$182_{10} = 10110110_2$$

(b) Reescriba los siguientes números según su representación signo-magnitud y complemento de 2. Después sume el segundo con el tercero.

1. -6_{10}
2. $-A3_{16}$
3. 123_{10}
4. -1_{10}
5. 0_n

Solución:

1. -6_{10}:

   Pasamos el número sin signo a binario,

   $$6_{10} = 110_2$$

   Completamos el byte (8 bits) con 0's.

   $$110_2 = 00000110_2$$

   Para signo-magnitud debemos cambiar el bit más significativo por el signo, como en este caso es negativo debemos cambiarlo por 1.

   $$-6_{10} = 10000110_2 \quad \text{(signo-magnitud)}$$

   Para complemento de 2 debemos invertir los bits y después sumar 1.

   $$00000110 \to 11111001 \to 11111010$$

   $$-6_{10} = 11111010_2 \quad \text{(complemento de 2)}$$

2. $-A3_{16}$:

   Pasamos el número sin signo a binario,

   $$A3_{16} = 10100011_2$$

   Como el número ya usa un byte, vamos a tener que agregar otro byte entero.

   $$10100011_2 = 0000000010100011_2$$

   Para signo-magnitud debemos cambiar el bit más significativo por el signo, como en este caso es negativo debemos cambiarlo por 1.

   $$-A3_{16} = 1000000010100011_2 \quad \text{(signo-magnitud)}$$

   Para complemento de 2 debemos invertir los bits y después sumar 1.

   $$0000000010100011 \to 1111111101011100 \to 1111111101011101$$

   $$-A3_{16} = 1111111101011101_2 \quad \text{(complemento de 2)}$$

3. 123_{10}:

   Pasamos el número sin signo a binario,

$$123_{10} = 1111011$$

   Completamos el byte (8 bits) con 0's.

$$1111011_2 = 01111011_2$$

   Para signo-magnitud debemos cambiar el bit más significativo por el signo, como en este caso es positivo no debemos cambiar nada porque el MSB ya es 0.

$$123_{10} = 01111011_2 \quad \text{(signo-magnitud)}$$

   Para representar el número en complemento de 2 no debemos hacer nada porque es un número positivo y su MSB es 0.

$$123_{10} = 01111011_2 \quad \text{(complemento de 2)}$$

4. -1_{10}:

   Pasamos el número sin signo a binario,

$$1_{10} = 1_2$$

   Completamos el byte (8 bits) con 0's.

$$1_2 = 00000001_2$$

   Para signo-magnitud debemos cambiar el bit más significativo por el signo, como en este caso es negativo debemos cambiarlo por 1.

$$-1_{10} = 10000001_2 \quad \text{(signo-magnitud)}$$

   Para complemento de 2 debemos invertir los bits y después sumar 1.

$$00000001 \to 11111110 \to 11111111$$

$$-1_{10} = 11111111_2 \quad \text{(complemento de 2)}$$

5. 0_n:

   Pasamos el número sin signo a binario,

$$0_n = 0_2$$

   Completamos el byte (8 bits) con 0's.

$$0_2 = 00000000_2$$

En signo-magnitud vamos a tener 2 representaciones para el 0; la positiva y la negativa.

$$00000000_2 = 10000000_2 = 00000000_2 \quad \text{(signo-magnitud)}$$

En complemento de 2 solo hay una representación.

$$0_{10} = 00000000_2 \quad \text{(complemento de 2)}$$

6. $-A3_{16} + 123_{10}$: Ya sabemos que:

$$-A3_{16} = 1111111101011101_2$$

$$123_{10} = 01111011_2$$

Entonces la suma es:

$$1111111101011101_2 + 0000000001111011_2$$

$$1111111111011000_2$$

(c) ¿Qué ocurre si sumamos 48_{10} con 23_{10} usando solo 7 bits y representándolos en complemento 2?

Solución: Partimos transformando ambos números a complemento 2:

$$48_{10} = 0110000_2$$

$$23_{10} = 0010111_2$$

Después realizamos la suma de estos:

$$0110000_2 + 0010111_2 = 1000111_2$$

Entonces el resultado es:

$$1000111_2 = -57_{10}$$

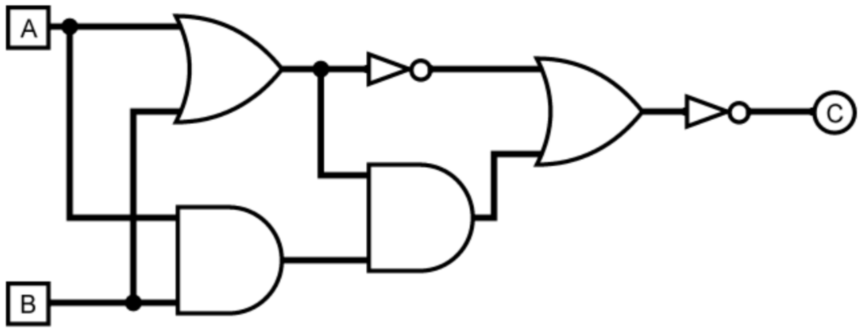
Pero claramente este resultado es **incorrecto**.

Esto ocurre porque necesitamos 7 bits para representar el resultado real (71), pero necesitamos uno de los bits para usar complemento 2, por lo tanto en verdad tenemos 6 bits para representar números, lo cuál significa que el número mas grande que se puede representar es $0111111_2 = 63_{10}$.

Al necesitar 8 bits para representar el resultado ocurre un **overflow**.

Pregunta 2: Obtener tabla de verdad a partir de un circuito [I1 2022-2]

Obtenga una fórmula de lógica booleana que represente el circuito de la figura. Utilice solo los conectivos $\wedge$, $\vee$ y $\neg$ ¿Es posible construir un circuito equivalente al anterior, usando menos compuertas y no necesariamente de las mismas?

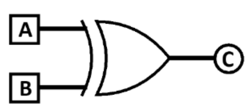


Solución:

La tabla de verdad del circuito es:

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Se puede ver que la tabla corresponde a la de XOR. La formula de lógica booleana que satisface lo requerido es la siguiente: $(\neg A \wedge B) \vee (A \wedge \neg B)$. El circuito que satisface a la lógica es el siguiente:



1. Feedback ayudantía

Escanee el QR para entregar feedback sobre la ayudantía.