



INSTITUTO POLITECNICO NACIONAL

Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias  
Sociales y Administrativas



Alumno  
Romero Moreno Oscar Alejandro

Materia  
Arq. Y Organiz. Computadoras

Boleta  
2015081222

Secuencia  
2NM31

Profesor  
Velasco Contreras Jose Antonio

Tarea 7

27/03/2020

# Unidad Aritmética Lógica

Abordaremos los aspectos que permiten la implementación de la aritmética de un computador, atributo funcional de la Unidad Aritmética Lógica (ALU). Primero se revisará lo relacionado a la forma de representar los números como una trama de bit, para posteriormente ver los algoritmos sobre dichas codificaciones, que permiten implementar las operaciones aritméticas de forma consistente. Veremos que la selección de los esquemas de codificación está fuertemente vinculada a la eficiencia en la implementación de su aritmética.

## Representación de números enteros

Las computadoras utilizan cuatro métodos para la representación interna de números enteros (positivos y negativos); éstos son los siguientes:

- Módulo y signo
- Complemento a 1
- Complemento a 2

### Módulo y signo

En este sistema de representación, el bit que está situado más a la izquierda representa el signo, y su valor será 0 para el signo + y 1 para el signo -. El resto de bits ( $n-1$ ) representan el módulo del número. Suponemos en principio que los números no poseen parte decimal, por lo que el punto se supone implícito a la derecha.

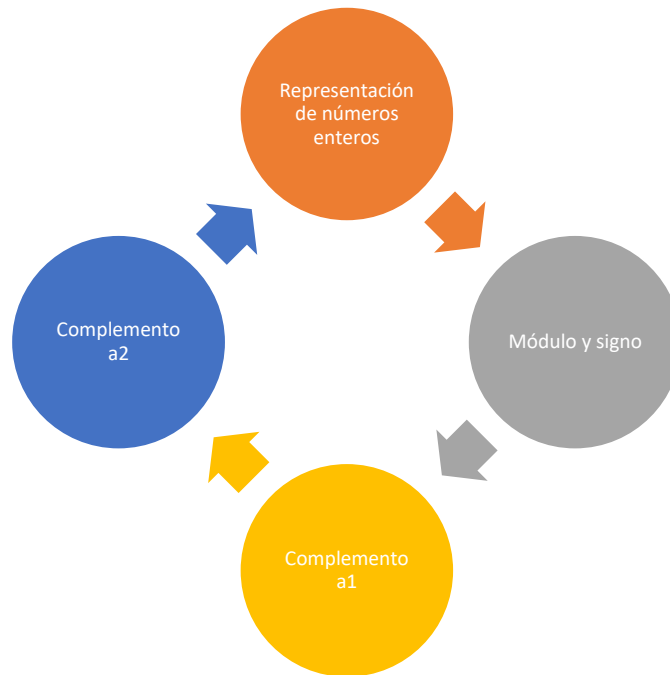
Por ejemplo, supongamos que disponemos de 8 bits, y queremos representar los números 10 y -10. Veamos cuáles son sus representaciones.

### Complemento a 1

Este sistema de representación utiliza el bit de más a la izquierda para el signo, correspondiendo el 0 para el signo + y el 1 para el signo -. Para los números positivos, los  $n-1$  bits de la derecha representan el módulo (igual que en el sistema anterior). El negativo de un número positivo se obtiene complementando todos sus dígitos (cambiando ceros por uno y viceversa) incluido el signo.

## Complemento a 2

Este sistema de representación utiliza el bit de más a la izquierda para el signo, correspondiendo el 0 para el signo + y el 1 para el signo -. Para los números positivos, los  $n - 1$  bits de la derecha representan el módulo (igual que en los dos sistemas anteriores). El negativo de un número positivo se obtiene en dos pasos:



## Representación de números en punto flotante

La representación en punto flotante está basada en la notación científica: – El punto decimal no se halla en una posición fija dentro de la secuencia de bits, sino que su posición se indica como una potencia de la base:

En todo número en punto se flotante distinguen tres componentes:

- Signo: indica el signo del número (0= positivo, 1=negativo)
- Mantisa: contiene la magnitud del número (en binario puro)
- Exponente: contiene el valor de la potencia de la base (sesgado)
- La base queda implícita y es común a todos los números, la más usada es 2. El valor de la secuencia de bits ( s,  $e_{p-1}$ , ..., e 0,  $m_{q-1}$ , ..., m 0 ) es: Dado que un mismo número

puede tener varias representaciones ( $0.110 \cdot 2^5 = 110 \cdot 2^{-2} = 0.0110 \cdot 2^6$ ) los números suelen estar normalizados: – un número está normalizado si tiene la forma  $1.xx... \cdot 2^{xx...}$  (ó  $0.1xx... \cdot 2^{xx...}$ ) – dado que los números normalizados en base 2 tienen siempre un 1 a la izquierda, éste suele quedar implícito (pero debe ser tenido en cuenta al calcular el valor de la secuencia)

