



Sistemas numéricos



Introducción

- El número es una abstracción, combinación de símbolos para representar un valor
- Sistemas numéricos posicionales, definidos por la base que utilizan (cantidad de símbolos distintos).
- El valor de un número se obtiene sumando los aportes de cada cifra.
- El aporte de cada cifra se obtiene multiplicando el valor de la cifra por el peso (base elevada a la posición empezando en cero a la izquierda de la coma).



Bases más importantes

- Decimal (10) uso cotidiano
- Binaria (2) uso en electrónica y computación
- Hexadecimal (16) comprime la notación de un número binario



Rango, precisión y error

- Rango: entorno definido por el menor y el mayor número representable, diferencia entre el número mayor y el menor
- Precisión: distancia entre dos números consecutivos
- Error: mitad de la diferencia entre dos números consecutivos



Errores

- Se pueden distinguir distintos tipos de errores pero la causa primaria de todos ellos es la misma, tratar de representar un conjunto infinito como el de los números con un conjunto finito
- Errores por rango finito: este tipo de error se genera cuando se intenta representar un número que esta fuera del rango representable.
- Errores por precisión finita: este tipo de error se genera cuando se intenta representar un número que es más pequeño que la precisión utilizada.



Punto fijo

- Separación entre parte entera y parte fraccionaria en una posición fija.
- Precisión constante en todo el rango.
- Mayor simplicidad de uso en operaciones



Comparación de representaciones

Decimal	Binario	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



Números negativos

- Valor absoluto y signo
- Complemento a 1 C1
- Complemento a 2 C2
- Exceso



Valor absoluto y signo

- Forma sencilla de planteo
- Primer bit de signo
- Resto de bits valor absoluto del número
- Doble representación del cero
- Problemas con comparaciones
- Problemas con sumas y restas
- Se agranda la representación agregando bits '0' entre el bit de signo y los bits de valor absoluto



Complemento a 1 C1

- Complemento bit a bit el nro en valor absoluto ($-x) = x'$
- Primer bit de signo
- Doble representación del cero
- Mejora las comparaciones
- Bastante bueno en sumas y restas, si hay acarreo sumo arrastre circular
- Mejora las comparaciones
- Se agranda la representación extendiendo el bit de signo



Complemento a 2 C2

- $C1 + 1 \rightarrow (-x) = x' + 1$
- Primer bit de signo
- Única representación del cero
- Mayor rango para números negativos
- Bueno para las comparaciones
- Ideal para sumar y restar, se descarta el acarreo
- Se agranda la representación extendiendo el bit de signo



Exceso

- Desplazamiento de la escala
- Numero mas chico (mas negativo) se representa 0000..0
- El cero se representa a mitad de escala (generalmente) 10000..0
- Números ordenados
- Ideal para comparaciones
- En suma y resta se debe tener en cuenta el exceso
- Se agranda la representación sumando la diferencia de excesos



Comparación de representaciones

Numero	VA y S	C1	C2	Exceso
8				
7	0111	0111	0111	1111
6	0110	0110	0110	1110
5	0101	0101	0101	1101
4	0100	0100	0100	1100
3	0011	0011	0011	1011
2	0010	0010	0010	1010
1	0001	0001	0001	1001
+0	0000	0000	0000	1000
-0	1000	1111		
-1	1001	1110	1111	0111
-2	1010	1101	1110	0110
-3	1011	1100	1101	0101
-4	1100	1011	1100	0100
-5	1101	1010	1011	0011
-6	1110	1001	1010	0010
-7	1111	1000	1001	0001
-8			1000	0000



Aritmética Positivos

- Suma/resta: Similar a base 10
- Si hay acarreo en la última cifra hay desborde
- Multiplicación/división: Similar a base 10
- Para multiplicar o dividir por 2 (o múltiplos) en base 2 puedo usar corrimientos



Aritmética Negativos

- Suma/resta: Similar a base 10
- Acarreo en la última cifra en C2 se pierde
- Acarreo en la última cifra en C1 se suma a la cifra menos significativa
- Detección de desborde necesaria
- Multiplicación/división: Similar a base 10, se pasan los operandos a valor absoluto y signo, se opera con la parte numérica y luego con el signo



Desborde en suma C2

Caso	Descripción
$a > 0, b > 0$	Acarreo se pierde Hay desborde si $c < 0$
$a > 0, b < 0$	Acarreo se pierde No hay desborde
$a < 0, b > 0$	Acarreo se pierde No hay desborde
$a < 0, b < 0$	Acarreo se pierde Hay desborde si $c > 0$



Punto Flotante

- Separación en signo, exponente, mantisa para aprovechar mejor la cantidad de bits
- Normalización para asegurar una sola representación de cada número
- Compromiso entre rango y precisión
- Precisión variable
- Cerca de 0 más precisión absoluta
- Cerca de infinito menor precisión absoluta
- Mayor complejidad de uso en operaciones



Norma IEEE754

- Simple precisión 32 bits
 - 1 bit signo
 - 8 bits exponente en base 2
 - 23 bits mantisa
- Doble precisión 64 bits
 - 1 bit signo
 - 11 bits exponente en base 2
 - 52 bits mantisa



Interpretación 32 bits

- Exponente 00000000
 - Mantisa = 0 -> 0
 - Mantisa \neq 0 -> cero sucio, número denormalizado, exponente = -126
- Exponente 00000001 - 11111110
 - Número normalizado
 - Exponente en exceso 127
 - Mantisa con primer bit implícito
- Exponente 11111111
 - Mantisa = 0 -> infinito
 - Mantisa \neq 0 -> NaN



Aritmética

- Suma: Llevar el número más pequeño al mismo exponente que el más grande, sumar las mantisas y normalizar
- Resta Llevar el número más pequeño al mismo exponente que el más grande, restar las mantisas y normalizar
- Producto: Sumar los exponentes (restar el exceso), multiplicar las mantisas, normalizar y calcular el signo