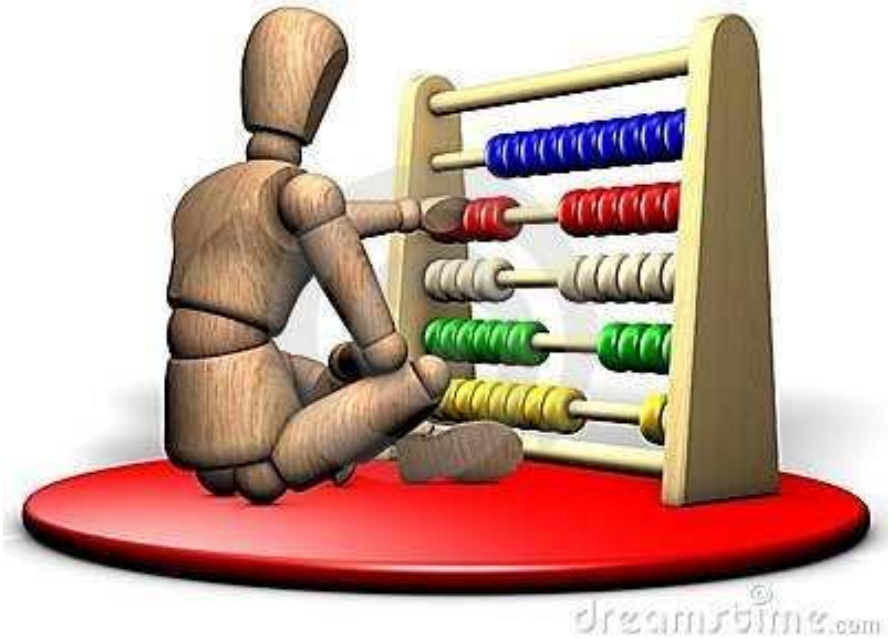
ARITMETICA DEL COMPUTADOR

NOVENA UNIDAD



Sumario

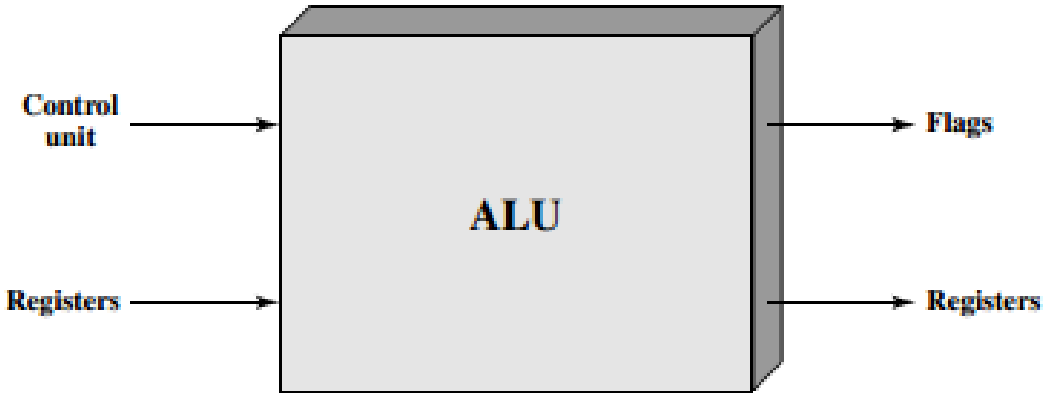
I. Unidad Aritmética Lógica
II. Representación Integrada
III. Aritmetica Integrada
IV. Representacion de Punto Flotante
V. Aritmetica de Punto Flotante

Introduccion.-

- La unidad aritmética y lógica (ALU) centra el aspecto más complejo de la aritmética del equipo.
- La aritmética computacional se realiza con frecuencia en dos tipos muy diferentes de números: enteros y de punto flotante.

I. Unidad Aritmética Lógica.-

- Parte del equipo que realiza la operaciones aritméticas y lógicas sobre los datos.
- Todos los demás elementos de la unidad de ordenador de control del sistema, los registros, memoria, E / S llevan datos a la UAL para que sean procesados y luego toma resultados de vuelta.
- Los datos se presentan a la UAL en los registros, y los resultados de una operación se almacenan en los registros. (almacenamiento temporal)
- Activar los indicadores como el resultado de una operación.
- Un indicador de desbordamiento se establece en 1 si el resultado de un cálculo supera la longitud del registro de almacenaje.



II. Representación de Enteros

- En el sistema binario, los números arbitrarios se representan con dígitos cero y uno, el signo menos, y el período o punto de base.
- Facilita el almacenamiento y procesamiento.
- Dificulta la representacion de números enteros negativos.

An 8-bit word can represent the numbers from 0 to 255, including

00000000 = 0
00000001 = 1
00101001 = 41
10000000 = 128
11111111 = 255

$$A = \sum_{i=0}^{n-1} 2^i a_i$$

II. Representación de Enteros

a. Representación de Signo - Magnitud.-

- Para representar enteros negativos y positivos, el bit más significativo (a la izquierda) se usa como un bit de signo.
- Si el bit de signo es 0, el número es positivo, si el bit de signo es 1, el número es negativo.

$$
\begin{aligned}
+\,18 &= 00010010 \\
-\,18 &= 10010010 \quad \text{(sign magnitude)}
\end{aligned}
$$

$$
A = \begin{cases} \sum_{i=0}^{n-2} 2^i a_i & \text{if } a_{n-1} = 0 \\ -\sum_{i=0}^{n-2} 2^i a_i & \text{if } a_{n-1} = 1 \end{cases}
$$

$$
\begin{aligned}
+0_{10} &= 00000000 \\
-0_{10} &= 10000000 \quad \text{(sign magnitude)}
\end{aligned}
$$

II. Representación de Enteros

b. Dos Representaciones Complementarias.-

Range	-2^{n-1} through $2^{n-1} - 1$
Number of Representations of Zero	One
Negation	Take the Boolean complement of each bit of the corresponding positive number, then add 1 to the resulting bit pattern viewed as an unsigned integer.
Expansion of Bit Length	Add additional bit positions to the left and fill in with the value of the original sign bit.
Overflow Rule	If two numbers with the same sign (both positive or both negative) are added, then overflow occurs if and only if the result has the opposite sign.
Subtraction Rule	To subtract B from A, take the twos complement of B and add it to A.

$$A = \sum_{i=0}^{n-2} 2^i a_i \qquad \text{for } A \geq 0$$

$$A = -2^{n-1} a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

II. Representación de Enteros
b. Dos Representaciones Complementarias.-

Decimal Representation	Sign-Magnitude Representation	Twos Complement Representation	Biased Representation
+8	—	—	1111
+7	0111	0111	1110
+6	0110	0110	1101
+5	0101	0101	1100
+4	0100	0100	1011
+3	0011	0011	1010
+2	0010	0010	1001
+1	0001	0001	1000
+0	0000	0000	0111
−0	1000	—	—
−1	1001	1111	0110
−2	1010	1110	0101
−3	1011	1101	0100
−4	1100	1100	0011
−5	1101	1011	0010
−6	1110	1010	0001
−7	1111	1001	0000
−8	—	1000	—

−128	64	32	16	8	4	2	1

(a) An eight-position twos complement value box

−128	64	32	16	8	4	2	1
1	0	0	0	0	0	1	1

−128 +2 +1 = −125

(b) Convert binary 10000011 to decimal

−128	64	32	16	8	4	2	1
1	0	0	0	1	0	0	0

−120 = −128 +8

(c) Convert decimal −120 to binary

II. Representación de Enteros

c. Conversión entre diferentes longitudes de Bit.-

•En un número entero de n bits y lo almacenan en M bits, donde el bit de signo en la posición más a la izquierda.

+18	=	00010010	(sign magnitude, 8 bits)
+18	=	0000000000010010	(sign magnitude, 16 bits)
−18	=	10010010	(sign magnitude, 8 bits)
−18	=	1000000000010010	(sign magnitude, 16 bits)

•Este procedimiento no va a trabajar para complementar los grupos de dos números enteros negativos.

+18	=	00010010	(twos complement, 8 bits)
+18	=	0000000000010010	(twos complement, 16 bits)
−18	=	11101110	(twos complement, 8 bits)
−32,658	=	1000000001101110	(twos complement, 16 bits)

II. Representación de Enteros

d. Representación de un punto fijo.-

- El punto de base (punto binario) es fijo y supone que esta a la derecha del ultimo dígito de la derecha.
- El programador puede utilizar la misma representación de fracciones binarias escalando los números de modo que el punto binario está implícitamente coloca en otro lugar

III. Aritmética de Enteros

a. Negación.-

- En representación de signo magnitud, la regla para la formación de la negación de un número entero es simple: invertir el bit de signo. En la notación de complemento a dos, la negación de un número entero se puede formar con las siguientes reglas:

  1. Tomar el complemento booleano de cada bit del entero (incluyendo el bit de signo). Establecer cada 1 a 0 y cada 0 a 1.
  2. Tratar el resultado como un entero binario sin signo, agregue 1.

+18	=	00010010	(twos complement)
bitwise complement	=	11101101	
		+ 1	
		11101110	= −18
−18	=	11101110	(twos complement)
bitwise complement	=	00010001	
		+ 1	
		00010010	= +18

III. Aritmética de Enteros

b. Adición y Sustracción.-

Se procede como si los dos números eran enteros sin signo.

Si el resultado de la operación es positivo, se obtiene un número positivo en forma de complemento a dos, que es el mismo que en forma entero sin signo.

Si el resultado de la operación es negativo, se obtiene un número negativo en forma de complemento a dos.

Hay un bit de acarreo más allá del final de la palabra (indicado por el sombreado), que se ignora.Esta condición se llama desbordamiento.

III. Aritmética de Enteros

b. Adición y Sustracción.-

REGLA DE DERRAME: Si dos números se suman, y ambos son positivos o negativos, a continuación, ocurre un desbordamiento si y sólo si el resultado tiene el signo contrario.

1001 = −7 +0101 = 5 1110 = −2 (a) (−7) + (+5)	1100 = −4 +0100 = 4 10000 = 0 (b) (−4) + (+4)
0011 = 3 +0100 = 4 0111 = 7 (c) (+3) + (+4)	1100 = −4 +1111 = −1 11011 = −5 (d) (−4) + (−1)
0101 = 5 +0100 = 4 1001 = Overflow (e) (+5) + (+4)	1001 = −7 +1010 = −6 10011 = Overflow (f) (−7) + (−6)

III. Aritmética de Enteros

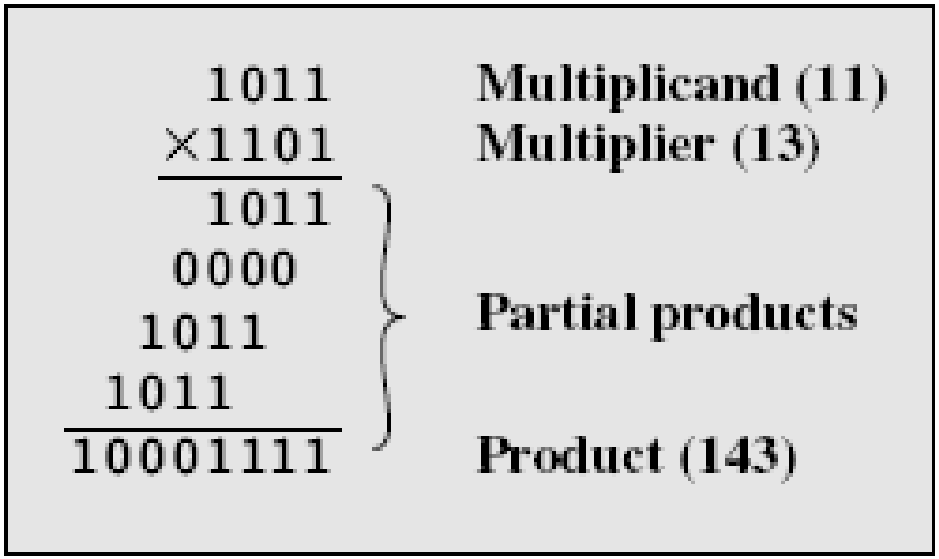
b. Adición y Sustracción.-

Regla de la Resta.- Para restar el número uno (sustraendo) de otro (minuendo), tomar el complemento a dos (negación) del sustraendo y agregarlo al minuendo

0010 = 2 +1001 = −7 1011 = −5 (a) M = 2 = 0010 S = 7 = 0111 −S = 1001	0101 = 5 +1110 = −2 10011 = 3 (b) M = 5 = 0101 S = 2 = 0010 −S = 1110
1011 = −5 +1110 = −2 11001 = −7 (c) M = −5 = 1011 S = 2 = 0010 −S = 1110	0101 = 5 +0010 = 2 0111 = 7 (d) M = 5 = 0101 S = −2 = 1110 −S = 0010
0111 = 7 +0111 = 7 1110 = Overflow (e) M = 7 = 0111 S = −7 = 1001 −S = 0111	1010 = −6 +1100 = −4 10110 = Overflow (f) M = −6 = 1010 S = 4 = 0100 −S = 1100

III. Aritmética de Enteros

a. Multiplicación.-

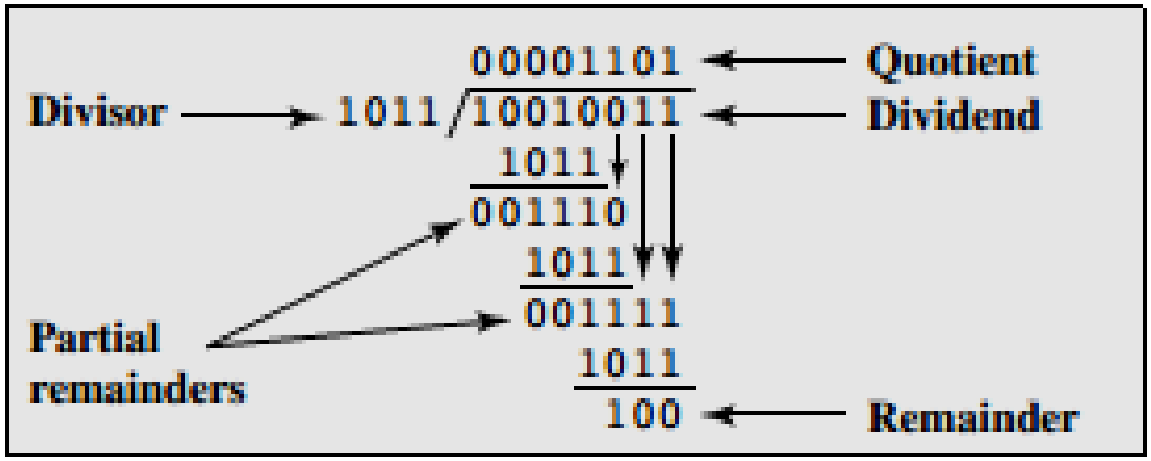


1001 (9)	1001 (−7)
× 0011 (3)	× 0011 (3)
00001001 1001 × 2^0	11111001 (−7) × 2^0 = (−7)
00010010 1001 × 2^1	11110010 (−7) × 2^1 = (−14)
00011011 (27)	11101011 (−21)

Si multiplicamos por un numero negativo se completa con unos los productos parciales

III. Aritmética de Enteros

d. División.-



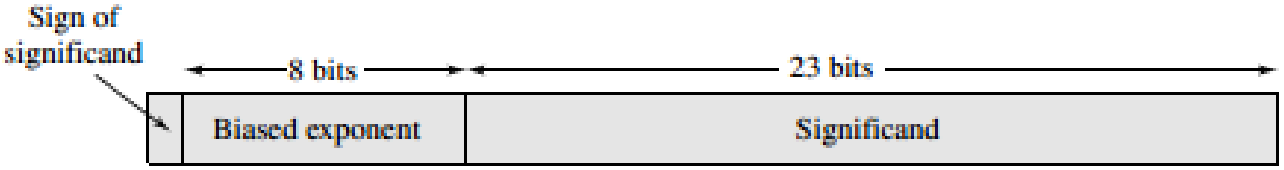
IV. Representación de Punto Flotante

a. Principios.-

Thus, 976,000,000,000,000 can be represented as 9.76×10^{14}, and 0.000000000000000976 can be represented as 9.76×10^{-14}. What we have done, in effect, is dynamically to

$$\pm S \times B^{\pm E}$$

- Sign: plus or minus
- Significand S
- Exponent E



(a) Format

$1.1010001 \times 2^{10100} = 0\ 10010011\ 10100010000000000000000 = 1.6328125 \times 2^{20}$
$-1.1010001 \times 2^{10100} = 1\ 10010011\ 10100010000000000000000 = -1.6328125 \times 2^{20}$
$1.1010001 \times 2^{-10100} = 0\ 01101011\ 10100010000000000000000 = 1.6328125 \times 2^{-20}$
$-1.1010001 \times 2^{-10100} = 1\ 01101011\ 10100010000000000000000 = -1.6328125 \times 2^{-20}$

(b) Examples

IV. Representación de Punto Flotante

a. Principios.-

Convertir 3.75_{10} a binario y hallar su representación en IEEE precisión simple

i	Num_{10}	d_i=parte entera(Num_{10})	NuevoNum_{10}=(Num_{10} - d_i) * b
0	3.75	d_0=3	1.50 = (3.75-3) * 2
1	1.50	d_1=1	1.00 = (1.50-1) * 2
2	1.00	d_2=1	0.00 = (1.00-1) * 2

$3.75_{10} = d_0 . d_1 d_2 = 11.11_2 = 1.111 \times 2^1$

3.75_{10}

(3.75-3) * 2 = 1.50 d0=3

(1.50-1) * 2 = 1.00 d_1=1

(1.00-1) * 2 = 0.00 d_2=1

$3.75_{10} = 11.11_2 = 1.111 \times 2^1$

IV. Representación de Punto Flotante

a. Principios.-

Convertir 0.3_{10} a binario y hallar su representación en IEEE precisión simple

0.3

$(0.3-0) * 2 = 0.6$ $d_0=0$

$(0.6-0) * 2 = 1.2$ $d_1=0$

$(1.2-1) * 2 = 0.4$ $d_2=1$

$(0.4-0) * 2 = 0.8$ $d_3=0$

$(0.8-0) * 2 = 1.6$ $d_4=0$

$(1.6-1) * 2 = 1.2$ $d_5=1$

$(1.2-1) * 2 = 0.4$ $d_6=1$

$(0.4-0) * 2 = 0.8$ $d_7=0$

$0.3_{10} = 0.01001101001..._2 = 1.001101001... \times 2^{-2}$

IV. Representación de Punto Flotante

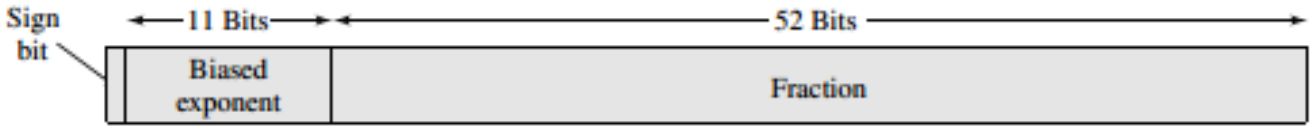
a. Principios.-

Decimal Representation	Sign-Magnitude Representation	Twos Complement Representation	Biased Representation
+8	—	—	1111
+7	0111	0111	1110
+6	0110	0110	1101
+5	0101	0101	1100
+4	0100	0100	1011
+3	0011	0011	1010
+2	0010	0010	1001
+1	0001	0001	1000
+0	0000	0000	0111
−0	1000	—	—
−1	1001	1111	0110
−2	1010	1110	0101
−3	1011	1101	0100
−4	1100	1100	0011
−5	1101	1011	0010
−6	1110	1010	0001
−7	1111	1001	0000
−8	—	1000	—

IV. Representación de Punto Flotante

b. Representación Punto Flotante Binario (IEEE).-

(a) Single format

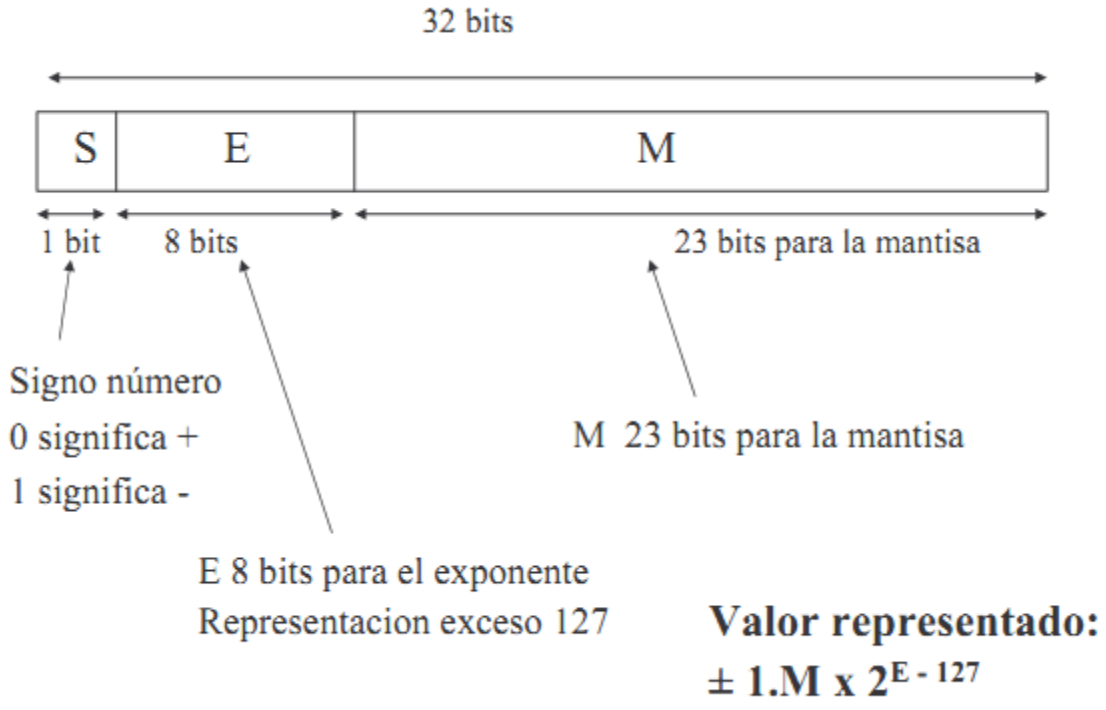


(b) Double format

Parameter	Format			
	Single	**Single Extended**	**Double**	**Double Extended**
Word width (bits)	32	≥43	64	≥79
Exponent width (bits)	8	≥11	11	≥15
Exponent bias	127	unspecified	1023	unspecified
Maximum exponent	127	≥1023	1023	≥16383
Minimum exponent	−126	≤−1022	−1022	≤−16382
Number range (base 10)	$10^{-38}, 10^{+38}$	unspecified	$10^{-308}, 10^{+308}$	unspecified
Significand width (bits)*	23	≥31	52	≥63
Number of exponents	254	unspecified	2046	unspecified
Number of fractions	2^{23}	unspecified	2^{52}	unspecified
Number of values	1.98×2^{31}	unspecified	1.99×2^{63}	unspecified

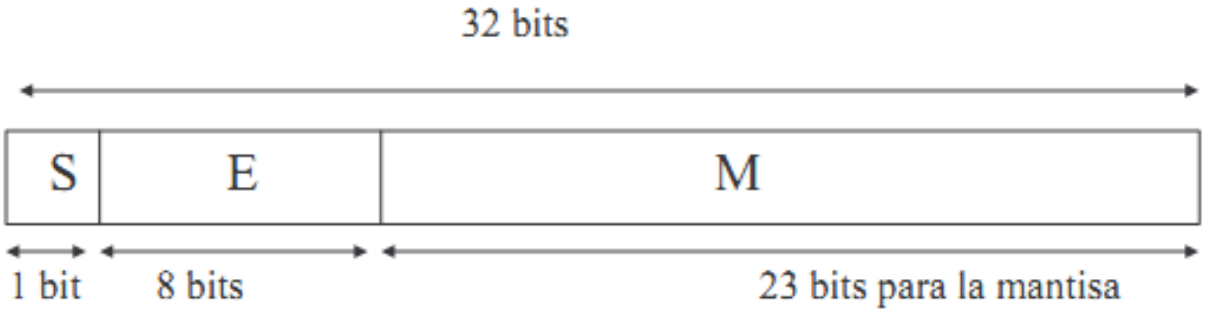
IV. Representación de Punto Flotante

b. Representación Punto Flotante Binario (IEEE).-



IV. Representación de Punto Flotante

b. Representación Punto Flotante Binario (IEEE).-



$7_{10} = 111_2$

Normalizamos el número y nos queda $1.11_2 \times 2^2$

El signo es positivo por lo que el campo de S queda con valor 0

Calculamos el exponente con exceso 127

$2 + 127 = 129 = 1000\ 0001_2$

El número 7_{10} en el estándar IEEE es representado como:

0	1000 0001	11000000000000000000000

V. Aritmética del Punto Flotante
   a. Adición y Sustracción.-

Entender y evaluación oral

V. Aritmética del Punto Flotante

b. Multiplicación y División.-

Entender y evaluación oral

V. Aritmética del Punto Flotante

c. Consideraciones de Precisión.-

Indicar recomendaciones

V. Aritmética del Punto Flotante

d. Aritmética Binaria del Punto Flotante (IEEE).-

Descripción