



Práctico 1: Aritmética

Representación de Números en Signo y Magnitud

1. Representar los siguientes números utilizando el convenio Signo y Magnitud:
 - 12
 - 9
 - 27
 - -15
 - 14
 - -19

Sumas en Complemento a 2 (8 bits)

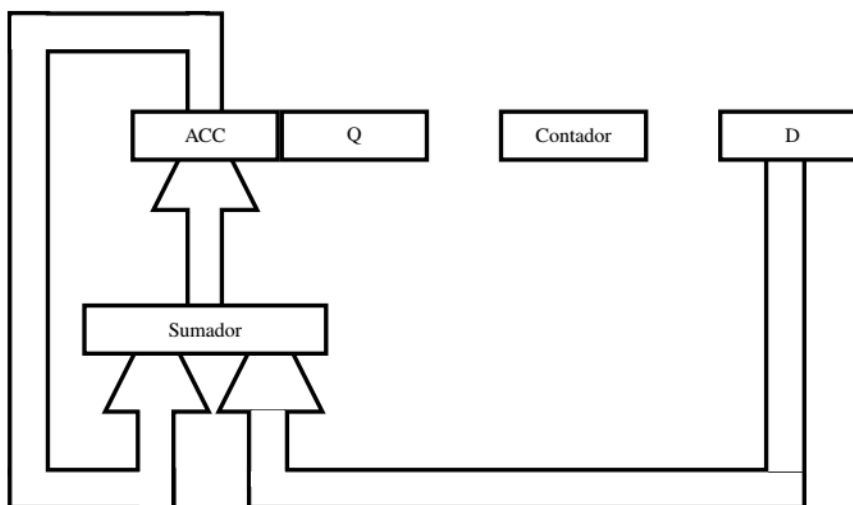
2. Realizar las siguientes sumas. Cada número debe representarse en complemento a 2 con 8 bits antes de efectuar la operación. Verificar si hay desbordamiento (overflow).
 - $12 + 9$
 - $27 + (-15)$
 - $14 + (-19)$
 - $(-7) + (-13)$
 - $23 + 10$
 - $(-20) + (-13)$

Multiplicaciones en Signo y Magnitud

3. Realizar las siguientes multiplicaciones de números binarios representados en Signo y Magnitud (4 bits de magnitud y 1 bit de signo).
 - a. $9 \times (-7)$
 - b. $-12 \times (-8)$
 - c. $-4 \times (12)$
- Describir las operaciones paso a paso con resultados parciales, en base al esquema que sigue:

ACCIÓN	REG. ACUMUL.	REG Q.	REG D.	CONTADOR

- Describir cómo se determina el signo del resultado. Realizar la tabla de verdad del dispositivo. ¿A qué función corresponde?



La imagen representa el dispositivo para realizar la multiplicación



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

El dispositivo utilizado tiene los siguientes registros:

- **Contador:** Se carga con el número de bits del multiplicador.
- **Acumulador:** Se pone a 0 inicialmente. Queda al final la parte más significativa del producto.
- **Q:** Se carga con el multiplicador. Queda la parte menos significativa del producto.
- **D:** Se carga con el multiplicando.

Instrucciones para la multiplicación en signo y magnitud

Verificar el bit menos significativo del registro Q (multiplicador):

- Si es 0, desplazar **todo el contenido** del registro acumulador y el registro Q **una posición a la derecha**, insertando un 0 en el bit más significativo del acumulador.
- Si es 1, **sumar** el registro D (multiplicando) al acumulador, luego desplazar **todo el contenido** del acumulador y el registro Q **una posición a la derecha**, insertando un 0 en el bit más significativo del acumulador.

Actualizar el contador:

- Restar 1 al contador después de cada operación.

Condición de finalización:

- La multiplicación termina cuando el contador llega a 0.

El producto (resultado) es el número formado por los bits del registro acumulador que carga los bits de mayor peso y en el registro Q, se almacenan los de menor peso.

Ejemplo de resolución

$$8 \times (-6)$$

$$8 = 01000 \text{ (multiplicando)} \quad -6 = 10110 \text{ (multiplicador)}$$

- **Contador:** Se carga con el número de bits del multiplicador. En este caso 4
- **Acumulador:** Se pone a 0 inicialmente. Los **bits más significativos** del producto quedan almacenados en el **registro acumulador**.
- **Q:** Se carga con el multiplicador. En este caso 0110. Los **bits menos significativos** del producto quedan en el **registro Q**.
- **D:** Se carga con el multiplicando. En este caso 1000

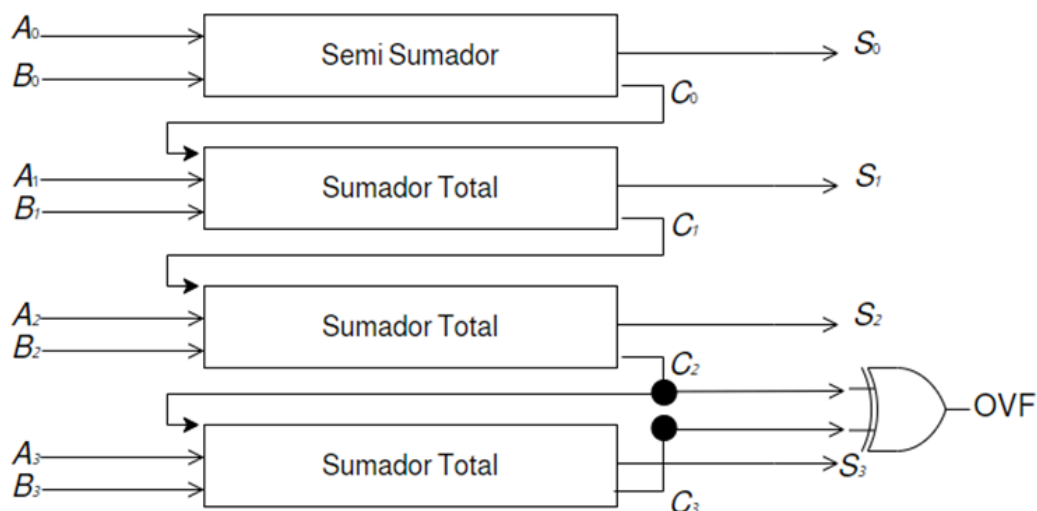
ACCIÓN	REG. ACUMUL.	REG Q.	REG D.	CONTADOR
carga	0000	0110	1000	4
→	0000	0011		3
sumo Reg. Acum + Reg. D	<u>+1000</u> 1000	0011 0011		
→	0100	0001		2
sumo Reg. Acum + Reg. D	<u>+1000</u> 1100	0001 0001		
→	0110	0000		1
→	0011	0000		0
Resultado	00110000			



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

Sumador Binario y Verificación de Resultados

4. La figura representa el circuito de un sumador binario de 4 bits. Los A_i y B_i son los sumandos; los S_i son los resultados; C_i son los acarrees y OVF es el rebalse (overflow).



Utilizando las tablas de verdad del medio sumador y del sumador total y para la representación de los números en el sistema de Complemento a 2 realizar las siguientes sumas y verificar los resultados. $S = A+B$

- $A=3$ $B=3$
- $A=5$ $B=-7$
- $A=-8$ $B=2$
- $A=7$ $B=6$
- $A=-5$ $B=-5$

A	B	Suma	Carry out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tabla de verdad semisumador

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabla de verdad Sumador Total



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

Ejemplo de resolución:

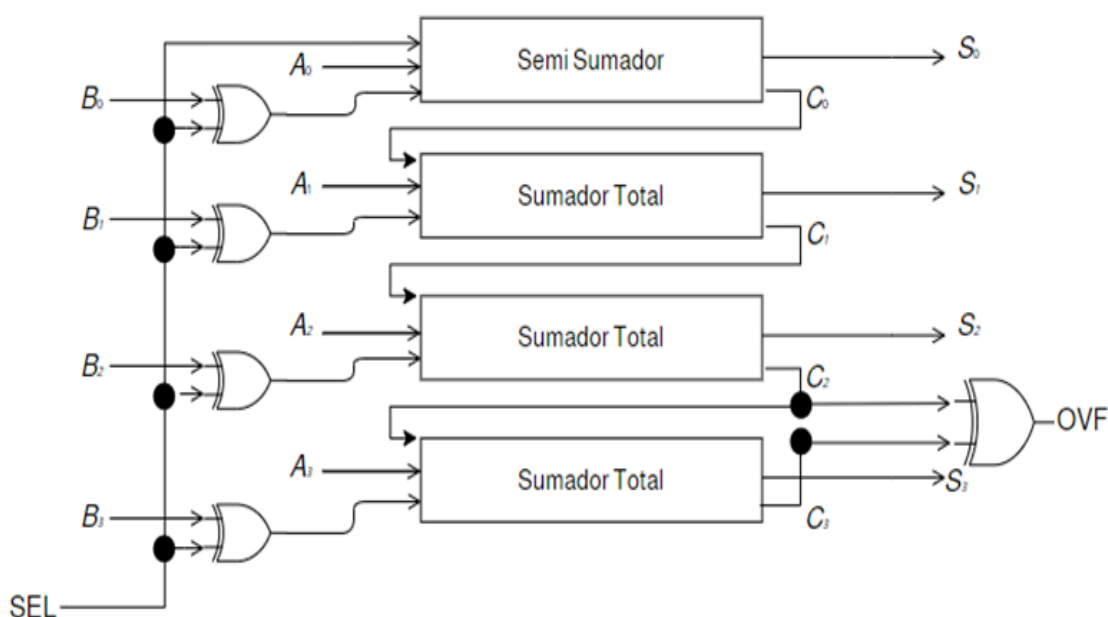
A=5 0101

B=3 0011

C_{in}	A	B	S	C_{out}	OVF
-	1	1	0	1	1
1	0	1	0	1	
1	1	0	0	1	
1	0	0	1	0	

RESULTADO DE LA SUMA: 1000 = 8

5. Analizar el siguiente circuito y explicar qué función realiza.



Para poder explicar su función, realizar las siguientes operaciones utilizando los valores correspondientes a cada una:

	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	SEL
Operación 1	0	1	0	1	0	0	1	0	0
Operación 2	0	1	0	1	0	1	0	0	1
Operación 3	1	1	1	0	0	0	1	0	1
Operación 4	1	0	1	1	1	1	0	1	0

6. Considerando los siguientes tipos de sumadores, ¿Cuál considera que es más rápido o eficiente? ¿Por qué?
- Un sumador secuencial con célula sumadora de 1 bit para realizar una suma de n bits.
 - Un sumador de acarreo serie (ripple adder).
 - Un sumador de generación de acarreo adelantado.



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

7. Una ALU de 16 bits está formada de 16 ALU de 1 bit y cada una tiene un tiempo de suma de 100 ns. Si para la propagación de una ALU a la siguiente existe una demora adicional de 10 ns. ¿Cuánto tiempo tardará en aparecer el resultado de una suma de 16 bits?
8. Se tiene una CPU en la cual la representación numérica es por medio del sistema de complemento a 2. ¿Cómo puede hacerse para utilizar el multiplicador anterior (signo, magnitud)? ¿Cuál es el costo que debe pagarse?

Algoritmo de Booth para Multiplicaciones en Complemento a 2

9. El algoritmo de Booth permite hacer multiplicaciones en Complemento a 2. Realizar las siguientes multiplicaciones. En cada una, definir cuántos bits debe tener la palabra para poder hacer la operación.
 - a. $9 \times (-7)$
 - b. $-12 \times (-8)$
 - c. $-4 \times (12)$
 - d. 12×10
 - e. 19×27
 - f. 6×15

Algoritmo de Booth – Reglas

(Considerando el bit de arrastre Q_{-1} , que inicia en 0)

1. Analizar los dos últimos bits (bit menos significativo del registro Q y Q_{-1}):
 - Si cambia de 0 a 1 → Restar el multiplicando (sumar su complemento a 2) al registro A y luego desplazar todo a la derecha.
 - Si cambia de 1 a 1 → Solo desplazar todo a la derecha.
 - Si cambia de 0 a 0 → Solo desplazar todo a la derecha.
 - Si cambia de 1 a 0 → Sumar el multiplicando al registro A y luego desplazar todo a la derecha.
2. Manejo del signo en el desplazamiento:
 - Si el resultado es **negativo** (bit más significativo de A = 1), antes de desplazar, insertar un 1 en el bit más significativo para mantener el signo.
 - Si el resultado es **positivo** (bit más significativo de A = 0), antes de desplazar, insertar un 0 en el bit más significativo.
3. Actualizar el contador:
 - Restar 1 al contador después de cada operación.
4. Finalización:
 - El proceso se repite hasta que el contador llegue a 0.
5. Resultado final:
 - El producto queda en el **registro A** (bits más significativos) y en el **registro Q** (bits menos significativos).

Resumen: Leyes o Reglas: (Considerando el bit de arrastre Q_{-1} que inicia en 0)

Cambio	Acción / operación
0 a 1	Reg. A + C2 (M) (multiplicando) y desplazar a la derecha →
1 a 1	desplazar a la derecha →
0 a 0	desplazar a la derecha →
1 a 0	Reg. A + M (multiplicando) y desplazar a la derecha →
	si el resultado es negativo (1 adelante) antes de desplazar introducir un 1 en el bit de mayor peso y luego desplazar (así se mantiene el signo), si no, introducir un 0



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

Ejemplo de resolución:

6 = 00110 x -9 = 10111

Acción	Reg A	Reg Q.	Q ₋₁ (bit de arrastre inicia en 0)	M	Contador
Carga inicial	00000	10111	0	00110 C2 =11010	5
0 a 1 Reg. A + C2 (M)	$\begin{array}{r} +11010 \\ -11010 \\ \hline \end{array}$	$\begin{array}{r} 10111 \\ 10111 \\ \hline \end{array}$			
→ Como el resultado es negativo mantener el signo	11101	01011	1		4
1 a 1 → Como el resultado es negativo mantener el signo	11110	10101	1		3
1 a 1 → Como el resultado es negativo mantener el signo	11111	01010	1		2
1 a 0 Reg. A + M (multiplicando)	$\begin{array}{r} +00110 \\ -01101 \\ \hline \end{array}$	$\begin{array}{r} 01010 \\ 01010 \\ \hline \end{array}$			
→	00011	10101	0		1
0 a 1 Reg. A + C2 (M)	$\begin{array}{r} +11010 \\ -11100 \\ \hline \end{array}$	$\begin{array}{r} 10101 \\ 10101 \\ \hline \end{array}$			
→ Como el resultado es negativo mantener el signo	11110	01010			0

Resultado 1111001010 = -54 (0000110110 = 54)

Divisiones en Signo y Magnitud

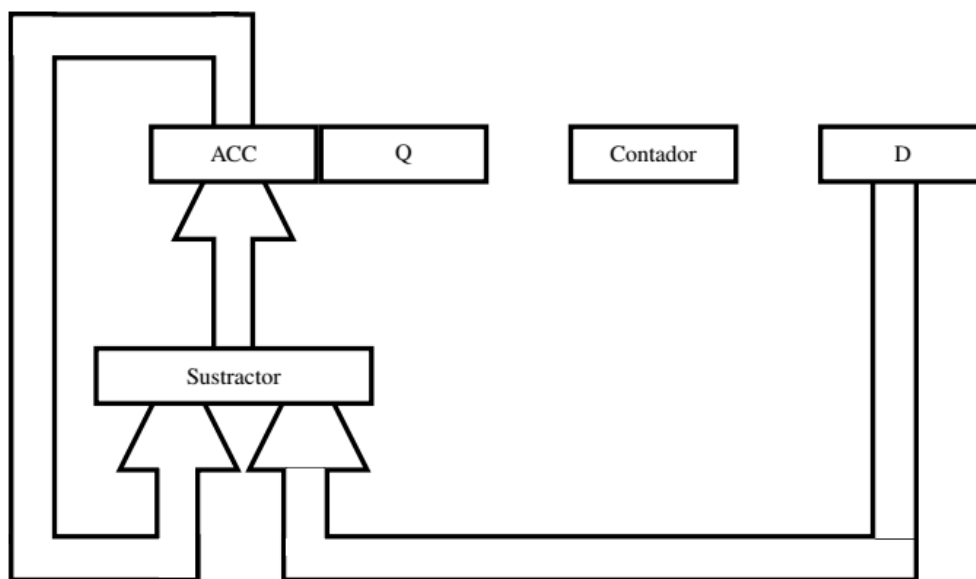
10. La división para el sistema Signo y Magnitud puede implementarse en forma similar a la multiplicación usando un par de registros ACC-Q, un registro D, un contador y un sustractor.

La función de cada registro es:

- **Contador:** Se carga con el número de bits del divisor y lleva la cuenta de los desplazamientos realizados.
- **Acumulador:** Se carga con la parte más significativa del dividendo. Va almacenando los resultados.
- **Q:** Se carga con la parte menos significativa del dividendo. Al finalizar el algoritmo queda almacenado el cociente.
- **D:** Se carga con el divisor. (es el sustraendo de la resta).



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras



Procedimiento para la división en Signo y Magnitud

1. Preparación inicial:
 - Antes de comenzar las operaciones, desplazar todo el contenido del registro acumulador y el registro Q, una posición a la izquierda.
2. Realizar la resta:
 - Restar el divisor (Registro D) del registro acumulador. Alternativamente, esto se puede hacer sumando el complemento a 2 del divisor.
3. Evaluar el resultado:
 - Si el resultado es positivo
 - Colocar un 1 en el bit menos significativo del registro Q (Q_0).
 - Desplazar todo el contenido del registro acumulador y el registro Q una posición a la izquierda.
 - Si el resultado es negativo
 - Restaurar el valor del registro acumulador sumando el divisor (D) nuevamente.
 - Colocar un 0 en el bit menos significativo de Q (Q_0).
 - Desplazar todo el contenido del registro acumulador y el registro Q una posición a la izquierda.
4. Repetir el proceso hasta que se haya realizado el número de iteraciones correspondiente al tamaño del divisor (cantidad de bits).
5. Interpretación del resultado:
 - El cociente se almacena en el registro Q.
 - El resto queda en el registro acumulador.

Resumen Simplificado:

1. Desplazar a la izquierda antes de operar.
2. Restar el divisor (D) del acumulador.
3. Si el resultado es positivo, poner 1 en Q_0 y desplazar.
4. Si el resultado es negativo, restaurar el acumulador, poner 0 en Q_0 y desplazar.
5. Repetir hasta completar todas las iteraciones.
6. El cociente queda en Q, el resto en el acumulado



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

Realizar las siguientes divisiones (en magnitud y signo)

- $100101101 \div 01010 = (-45) \div 10$
- $001100100 \div 10111 = 100 \div (-7)$
- $12 \div 3$
- $15 \div 2$

Describir las operaciones paso a paso con los resultados parciales en forma análoga a la que se hizo con la multiplicación.

Ejemplo de resolución:

12/3 12: 01100 3: 00011 -3: 11101

En las divisiones hay que dividir un número de $2n$ bits, con otro de n bits, siendo el resultado y el resto de n bits.

ACCIÓN	REG. ACUMUL.	REG Q.	REG D.	CONTADOR
	00000	01100	00011	15
← desplazo	00000	11000		
resto D (sumo el C2 (D)) es <0	11101 11101	11000 11000		
restauró sumo D ←	00011 00000 00001	11000 11000 10000		4
resto D (sumo el C2 (D)) es <0	11101 11110	10000		3
restauró sumo D ←	00011 00001 00011	10000 10000 00000		2
resto D (sumo el C2 (D)) ←	11101 00000 00000	00000 00001		1
resto D (sumo el C2 (D)) es <0	11101 11101	00010		
restauró sumo D ←	00011 00000 00000	00010 00100		0

11. Se tiene una CPU en la cual la representación numérica es por medio del sistema de complemento a 2. ¿Cómo puede hacerse para utilizar el divisor anterior (signo, magnitud)? ¿Cuál es el costo que debe pagarse?



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

Representación IEEE 754 y Cálculo de Errores

12. Usando notación científica ($n = m \times r^E$, con $r=10$ y $m= 2$ dígitos) hacer la suma:

$$0,10 + 3,8 + 9,9 + 15,2 + 39,6 + 65,3$$

- de menor a mayor
- de mayor a menor

Analizar en cada caso el error que se comente. ¿La suma es asociativa?

13. Representar usando la normalización IEEE en simple precisión los siguientes números. Verificar usando la PC. [Conversor IEEE 754](#)

- 1,3
- 72,5
- 0,125

Calcular el error porcentual cometido en la transformación.

Ejemplo de Resolución:

F.4 \rightarrow 1111.01

Normalización: 1.11101×2^3

Signo: +

Exp= $3+127 = 130$

Mantisa normalizada: 1.11101 (no escribimos el 1 antes de la coma porque es el bit implícito)

0	1	0	0	0	0	0	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Cómo calcular el error:

1° pasamos el número obtenido a decimal

$$(1 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-5}) \times 2^3 = 15.25$$

$$\text{Error relativo} = |\text{Valor obtenido} - \text{Valor real}| = |15,25 - 15,25| = 0$$

14. Encontrar los números representados usando la normalización IEEE en simple precisión. Verificar usando la PC.

- BFA66666
- 00008B61

15. Realizar las siguientes operaciones con números en punto flotante en el formato IEEE 754 (en las que las partes significativas se truncan a cuatro dígitos decimales). Indique los resultados en forma normalizada).

- C30C0000 + C1500000
- 3B370000 + 39F68000

16. Realizar las siguientes operaciones en IEEE 754 de simple precisión. Utilizar cuando sea necesario el modo de redondeo al más cercano:

- $1,5 + 2^{-22}$
- $1,0 - (1,5 + 2^{-22})$
- $2^{10} / (2 - 2^{-22})$
- $8,04^{21} + 6.33^{18}$
- $5.4^{-16} / 2.89^7$



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

f. $65.58^{12} \times 44.8792^8$

17. En algunas máquinas se aprovecha que en un número en punto flotante normalizado a fracción pura el primer bit es "1". Entonces no es necesario su almacenamiento en memoria. Se utiliza la idea de bit implícito. ¿Qué ventajas y desventajas trae?
18. En el formato de representación IEEE se usa una técnica para mejorar la precisión de los números muy cercanos a cero. ¿En qué consiste? ¿Qué complicación implica? ¿Qué inconveniente encontramos en la representación de números reales en punto fijo?
19. Suponer que necesita trabajar con una precisión de 50 dígitos decimales en una PC convencional. ¿Existe algún problema? ¿En ese caso cómo lo solucionaría?
20. Considerar las siguientes representaciones en coma flotante:
 - i. 32 bits, que emplea 10 bits para el exponente y 21 para la mantisa
 - ii. 16 bits, que emplea 5 bits para el exponente y 10 para la mantisa.

Indicar:

- a) El menor y el mayor número no normalizado representable.
- b) El menor y el mayor normalizado representable.
- c) Sacar conclusiones