Universidad Centroamericana “José Simeón Cañas”
Facultad de Ingeniería y Arquitectura

Tutorías - Fundamentos de Programación Ciclo 01/2021
Tutor: Ernesto José Canales Guillén

Realización de operaciones en la ALU

Recordemos:

- **Magnitud verdadera:**

Consiste en la representación de números binarios tal cual es. Es decir, se realiza la conversión por el método conocido, se escribe en las casillas, justificado a la derecha y se coloca el bit de signo en la posición más a la izquierda. Luego las casillas intermedias se rellenan con cero. Esta es la representación que más se parece a la de los binarios naturales vista en las clases anteriores.

- **Complemento a Uno:**

Consiste en obtener primero la representación en magnitud verdadera; luego se reemplazan los unos por ceros y los ceros por unos, excepto el signo. En otras palabras, se niegan todos los bits de la magnitud.

- **Complemento a Dos:**

Consiste en obtener primero la representación en magnitud verdadera; luego se obtiene la representación en complemento a uno y finalmente, se le suma uno

- **Convenio de Complemento a dos:**

Para la representación de cantidades numéricas en las computadoras digitales actuales existe un convenio, el cual se conoce como “convenio de complemento a dos”. Este convenio nos dice que, dentro de las computadoras:

a) Las cantidades positivas se representan en magnitud verdadera.
b) Las cantidades negativas se representan en complemento a dos.

- **Conversión de regreso a base diez:**

Para convertir a base diez una cantidad representada en cualquiera de estos formatos, se procede así:

a) Si está en magnitud verdadera: simplemente se realiza la suma de las potencias de dos de las posiciones que tienen uno, excepto el signo. Luego, el signo se coloca dependiendo si es 0 u 1 en esa posición.
b) Si está en complemento a uno: se niegan los bits de magnitud para regresar a magnitud verdadera y luego se procede como indica el literal a).
c) Si está en complemento a dos: se niegan los bits, se le suma 1 y luego se procede como en el literal a).

“Para todo ejercicio de conversión a cualquiera de estas tres representaciones nos deberán indicar con cuantos bits se está trabajando.”

Ejercicios

Representar cada una de las siguientes cantidades en magnitud verdadera, complemento a uno y complemento a dos, con la cantidad de bits indicada:

a) 57 con 9 bits.
b) 80, con 10 bits.
c) 215 con 11 bits.
d) –49 con 7 bits.
e) –346 con 11 bits.

a) $(+57)_{10} = (0\ 0011\ 1001)_2$ con 9 bits

MV:

0	0	0	1	1	1	0	0	1

C1:

Negar								
0	1	1	0	0	0	1	1	0

C2:

Sumar +1								
								+1
0	1	1	0	0	0	1	1	0

R\\

0	1	1	0	0	0	1	1	1

b) $(+80)_{10} = (00\ 0101\ 0000)_2$ con 10 bits

MV:

0	0	0	1	0	1	0	0	0	0

C1:

Negar									
0	1	1	0	1	0	1	1	1	1

C2:

Sumar +1									
					1	1	1	1	+1
0	1	1	0	1	0	1	1	1	1

R\\

0	1	1	0	1	1	0	0	0	0

c) $(+215)_{10} = (000\ 1101\ 0111)_2$ con 11 bits

MV:

0	0	0	1	1	0	1	0	1	1	1

C1:

Negar										
0	1	1	0	0	1	0	1	0	0	0

C2:

Sumar +1										
										+1
0	1	1	0	0	1	0	1	0	0	0

R\\

0	1	1	0	0	1	0	1	0	0	1

d) $(-49)_{10} = (111\ 0001)_2$ con 7 bits

MV:

1	1	1	0	0	0	1

C1:

Negar						
1	0	0	1	1	1	0

C2:

Sumar +1						
						+1
1	0	0	1	1	1	0

R\\

1	0	0	1	1	1	1

e) $(-346)_{10} = (101\ 0101\ 1010)_2$ con 11 bits

MV:

1	0	1	0	1	0	1	1	0	1	0

C1:

Negar										
1	1	0	1	0	1	0	0	1	0	1

C2:

Sumar +1										
									1	+1
1	1	0	1	0	1	0	0	1	0	1

R\\

1	1	0	1	0	1	0	0	1	1	0

Dada las siguientes cantidades y el formato de representación indicado, ¿cuál es la cantidad correspondiente, en el sistema Base 10?

f) 1000101011, en magnitud verdadera.
g) 0011101, en complemento a uno.
h) 10101101, en complemento a dos.

f) $(10\ 0010\ 1011)_2 = (-43)_{10}$ es MV con 10 bits

MV: (Conversión de regreso: caso a.)

R\\

1	0	0	0	1	0	1	0	1	1

g) $(001\ 1101)_2 = (+34)_{10}$ es C1 con 7 bits

C1: (Conversión de regreso: caso b.)

0	0	1	1	1	0	1

R\\

Negar = MV						
0	1	0	0	0	1	0

h) (1010 1101)2 = (-83)10 es C2 con 8 bits

C2: (Conversión de regreso: caso c.)

1	0	1	0	1	1	0	1

Negar							
1	1	0	1	0	0	1	0

Sumar +1 = MV							
							+1
1	1	0	1	0	0	1	0

R\\

1	1	0	1	0	0	1	1

Al realizar cálculos en la ALU se pueden presentar 5 casos:

Caso 1: Dos números positivos:

Esta adición es directa. Ya que ambos son positivos se espera un 0 en el bit de signo, en caso contrario, hay un error.

a)	b)	c)	d)
4	21	341	480
+ 6	+ 5	+ 250	+ 640
10	26	591	1120
Con cinco bits	Con seis bits	Con 12 bits	Con catorce bits

a) (4 + 6 = 10) con 5 bits

$(4)_{10} = (0\ 0100)_2$ $(6)_{10} = (0\ 0110)_2$ $(10)_{10} = (0\ 1010)_2$

MV: 4

0	0	1	0	0

MV: 6

0	0	1	1	0

Sumar: 4 + 6

			1				
	0	0	0	1	0	0	MV
+	0	0	0	1	1	0	MV
R\\	0	0	1	0	1	0	MV

b) (21+ 5 = 26) con 6 bits

$(21)_{10} = (01\ 0101)_2$ $(5)_{10} = (00\ 0101)_2$ $(26)_{10} = (01\ 1010)_2$

MV: 21

0	1	0	1	0	1

MV: 5

0	0	0	1	0	1

Sumar: 21 + 5

				1		1		
	0	0	1	0	1	0	1	MV
+	0	0	0	0	1	0	1	MV
R\\	0	0	1	1	0	1	0	MV

c) (341+ 250 = 591) con 12 bits

$(341)_{10} = (0001\ 0101\ 0101)_2$ $(250)_{10} = (0000\ 1111\ 1010)_2$ $(591)_{10} = (0010\ 0100\ 1111)_2$

MV: 341

0	0	0	1	0	1	0	1	0	1	0	1

MV: 250

0	0	0	0	1	1	1	1	1	0	1	0

Sumar: 341 + 250

				1	1	1	1	1						
	0	0	0	0	1	0	1	0	1	0	1	0	1	MV
+	0	0	0	0	0	1	1	1	1	1	0	1	0	MV
R\\	0	0	0	1	0	0	1	0	0	1	1	1	1	MV

d) (480+ 640 = 1120) con 14 bits

$(480)_{10}$ = (00 0001 1110 0000)$_2$ $(640)_{10}$ = (00 0010 1000 0000)$_2$ $(1120)_{10}$ = (00 0100 0110 0000)$_2$

MV: 480

0	0	0	0	0	1	1	1	1	0	0	0	0	0

MV: 640

0	0	0	0	1	0	1	0	0	0	0	0	0	0

Sumar: 480 + 640

					1	1	1									
	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	MV
+	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	MV
R\\	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	MV

Caso 2: Un número positivo y un número negativo de módulo menor:

Se espera un 0 en el bit de signo. Se generará un acarreo a la izquierda del bit de signo, que será despreciado. El resultado es un número positivo en el formato de magnitud verdadera

a) 9 + -6 = 3 Con cinco bits

b) 220 + -82 = 138 Con diez bits

c) 341 + -250 = 91 Con 12 bits

d) 153 + -63 = 90 Con nueve bits

a) (9 + (-6) = 3) con 5 bits

$(9)_{10} = (0\ 1001)_2$ $(-6)_{10} = (1\ 0110)_2$ $(3)_{10} = (0\ 0011)_2$

MV: 9

0	1	0	0	1

MV: -6

1	0	1	1	0

C1: -6

Negar				
1	1	0	0	1

C2: -6

Sumar +1				
			1	+1
1	1	0	0	1

1	1	0	1	0

Sumar: 9 + (-6)

	1	1					
	0	0	1	0	0	1	MV
+	0	1	1	0	1	0	C2
R\\	1	0	0	0	1	1	MV

b) (220 + (-82) = 138) con 10 bits

$(220)_{10} = (00\ 1101\ 1100)_2$ $(-82)_{10} = (10\ 0101\ 0010)_2$ $(138)_{10} = (00\ 1000\ 1010)_2$

MV: 220

0	0	1	1	0	1	1	1	0	0

MV: -82

1	0	0	1	0	1	0	0	1	0

C1: -82

Negar									
1	1	1	0	1	0	1	1	0	1

C2: -82

Sumar +1									
								1	+1
1	1	1	0	1	0	1	1	0	1

1	1	1	0	1	0	1	1	1	0

Sumar: 220 + (-82)

	1	1	1	1	1	1	1	1				
	0	0	0	1	1	0	1	1	1	0	0	MV
+	0	1	1	1	0	1	0	1	1	1	0	C2
R\\	1	0	0	1	0	0	0	1	0	1	0	MV

c) (341 + (-250) = 91) con 12 bits

$(341)_{10} = (0001\ 0101\ 0101)_2$ $(-250)_{10} = (1000\ 1111\ 1010)_2$ $(91)_{10} = (0000\ 0101\ 1011)_2$

MV: 341

0	0	0	1	0	1	0	1	0	1	0	1

MV: -250

1	0	0	0	1	1	1	1	1	0	1	0

C1: -250

Negar											
1	1	1	1	0	0	0	0	0	1	0	1

C2: -250

Sumar +1											
										1	+1
1	1	1	1	0	0	0	0	0	1	0	1

1	1	1	1	0	0	0	0	0	1	1	0

Sumar: 341 + (-250)

	1	1	1	1						1				
	0	0	0	0	1	0	1	0	1	0	1	0	1	MV
+	0	1	1	1	1	0	0	0	0	0	1	1	0	C2
R\\	1	0	0	0	0	0	1	0	1	1	0	1	1	MV

d) (153 + (-63) = 90) con 9 bits

$(153)_{10} = (0\ 1001\ 1001)_2$ $(-63)_{10} = (1\ 0011\ 1111)_2$ $(90)_{10} = (0\ 0101\ 1010)_2$

MV: 153

0	1	0	0	1	1	0	0	1

MV: -63

1	0	0	1	1	1	1	1	1

C1: -63

Negar								
1	1	1	0	0	0	0	0	0

C2: -63

Sumar +1								
								+1
1	1	1	0	0	0	0	0	0

1	1	1	0	0	0	0	0	1

Sumar: 153 + (-63)

	1	1							1		
	0	0	1	0	0	1	1	0	0	1	MV
+	0	1	1	1	0	0	0	0	0	1	C2
R\\	1	0	0	1	0	1	1	0	1	0	MV

Caso 3: Un número positivo y un número negativo de módulo mayor:

Se espera un 1 en el bit de signo. El resultado es un número negativo en complemento a dos.

a)	b)	c)
27	187	61
+ -43	+ -380	+ -86
-16	-193	-25
Con ocho bits	Con once bits	Con 10 bits

a) (27 + (-43) = -16) con 8 bits

$(27)_{10} = (0001\ 1011)_2$ $(-43)_{10} = (1010\ 1011)_2$ $(-16)_{10} = (1001\ 0000)_2$

MV: 27

0	0	0	1	1	0	1	1

MV: -43

1	0	1	0	1	0	1	1

C1: -43

Negar							
1	1	0	1	0	1	0	0

C2: -43

Sumar +1							
							+1
1	1	0	1	0	1	0	0

1	1	0	1	0	1	0	1

Sumar: 27 + (-43)

				1	1	1	1	1		
	0	0	0	0	1	1	0	1	1	MV
+	0	1	1	0	1	0	1	0	1	C2
	0	1	1	1	1	0	0	0	0	C2

Conversión de regreso: C2 a MV, caso c.

Negar							
1	0	0	0	1	1	1	1

Sumar +1							
			1	1	1	1	+1
1	0	0	0	1	1	1	1

R\\

1	0	0	1	0	0	0	0

b) (187 + (-380) = -193) con 11 bits

$(187)_{10} = (000\ 1011\ 1011)_2$ $(-380)_{10} = (101\ 0111\ 1100)_2$ $(-193)_{10} = (100\ 1100\ 0001)_2$

MV: 187

0	0	0	1	0	1	1	1	0	1	1

MV: -380

1	0	1	0	1	1	1	1	1	0	0

C1: -380

Negar										
1	1	0	1	0	0	0	0	0	1	1

C2: -380

Sumar +1										
								1	1	+1
1	1	0	1	0	0	0	0	0	1	1

1	1	0	1	0	0	0	0	1	0	0

Sumar: 187 + (-380)

				1									
	0	0	0	0	1	0	1	1	1	0	1	1	MV
+	0	1	1	0	1	0	0	0	0	1	0	0	C2
	0	1	1	1	0	0	1	1	1	1	1	1	C2

Conversión de regreso: C2 a MV, caso c.

Negar										
1	0	0	1	1	0	0	0	0	0	0

Sumar +1										
										+1
1	0	0	1	1	0	0	0	0	0	0

R\\

1	0	0	1	1	0	0	0	0	0	1

c) (61 + (-86) = -25) con 10 bits

$(61)_{10} = (00\ 0011\ 1101)_2$ $(-86)_{10} = (10\ 0101\ 0110)_2$ $(-25)_{10} = (10\ 0001\ 1001)_2$

MV: 61

0	0	0	0	1	1	1	1	0	1

MV: -86

1	0	0	1	0	1	0	1	1	0

C1: -86

Negar									
1	1	1	0	1	0	1	0	0	1

C2: -86

Sumar +1									
								1	+1
1	1	1	0	1	0	1	0	0	1

1	1	1	0	1	0	1	0	1	0

Sumar: 61 + (-86)

					1	1	1					
	0	0	0	0	0	1	1	1	1	0	1	MV
+	0	1	1	1	0	1	0	1	0	1	0	C2
	0	1	1	1	1	1	0	0	1	1	1	C2

Conversión de regreso: C2 a MV, caso c.

Negar									
1	0	0	0	0	1	1	0	0	0

Sumar +1									
									+1
1	0	0	0	0	1	1	0	0	0

R\\

1	0	0	0	0	1	1	0	0	1

Caso 4: Dos números negativos:

Se espera un 1 en el bit de signo. Se generará un acarreo a la izquierda del bit de signo, que será despreciado. El resultado será un número negativo en complemento a dos.

a)	b)	c)
-31	-63	-328
+ -17	+ -105	+ -230
-48	-168	-558
Con ocho bits	Con once bits	Con 12 bits

a) (-31 + (-17) = -48) con 8 bits

$(-31)_{10} = (1001\ 1111)_2$ $(-17)_{10} = (1001\ 0001)_2$ $(-48)_{10} = (1011\ 0000)_2$

MV: -31

1	0	0	1	1	1	1	1

C1: -31

Negar							
1	1	1	0	0	0	0	0

C2: -31

Sumar +1							
							+1
1	1	1	0	0	0	0	0

1	1	1	0	0	0	0	1

MV: -17

1	0	0	1	0	0	0	1

C1: -17

Negar							
1	1	1	0	1	1	1	0

C2: -17

Sumar +1							
							+1
1	1	1	0	1	1	1	0

1	1	1	0	1	1	1	1

Sumar: -31 + (-17)

	1	1	1		1	1	1	1		
	0	1	1	1	0	0	0	0	1	C2
+	0	1	1	1	0	1	1	1	1	C2
	0	1	1	0	1	0	0	0	0	C2

Conversión de regreso: C2 a MV, caso c.

Negar							
1	0	1	0	1	1	1	1

Sumar +1							
			1	1	1	1	+1
1	0	1	0	1	1	1	1

R\\

1	0	1	1	0	0	0	0

b) (-63 + (-105) = -168) con 11 bits

$(-63)_{10} = (100\ 0011\ 1111)_2$ $(-105)_{10} = (100\ 0110\ 1001)_2$ $(-168)_{10} = (100\ 1010\ 1000)_2$

MV: -63

1	0	0	0	0	1	1	1	1	1	1

C1: -63

Negar										
1	1	1	1	1	0	0	0	0	0	0

C2: -63

Sumar +1										
										+1
1	1	1	1	1	0	0	0	0	0	0

1	1	1	1	1	0	0	0	0	0	1

MV: -105

1	0	0	0	1	1	0	1	0	0	1

C1: -105

Negar										
1	1	1	1	0	0	1	0	1	1	0

C2: -105

Sumar +1										
										+1
1	1	1	1	0	0	1	0	1	1	0

1	1	1	1	0	0	1	0	1	1	1

Sumar: -63 + (-105)

	1	1	1	1					1	1	1		
	0	1	1	1	1	1	0	0	0	0	0	1	C2
+	0	1	1	1	1	0	0	1	0	1	1	1	C2
	1	1	1	1	0	1	0	1	1	0	0	0	C2

Conversión de regreso: C2 a MV, caso c.

Negar										
1	0	0	1	0	1	0	0	1	1	1

Sumar +1										
							1	1	1	+1
1	0	0	1	0	1	0	0	1	1	1

R\\

1	0	0	1	0	1	0	1	0	0	0

c) (-328 + (-230) = -558) con 12 bits

$(-328)_{10} = (1001\ 0100\ 1000)_2$ $(-230)_{10} = (1000\ 1110\ 0110)_2$ $(-558)_{10} = (1010\ 0010\ 1110)_2$

MV: -328

1	0	0	1	0	1	0	0	1	0	0	0

C1: -328

Negar											
1	1	1	0	1	0	1	1	0	1	1	1

C2: -328

Sumar +1											
								1	1	1	+1
1	1	1	0	1	0	1	1	0	1	1	1

1	1	1	0	1	0	1	1	1	0	0	0

MV: -230

1	0	0	0	1	1	1	0	0	1	1	0

C1: -230

Negar											
1	1	1	1	0	0	0	1	1	0	0	1

C2: -230

Sumar +1											
										1	+1
1	1	1	1	0	0	0	1	1	0	0	1

1	1	1	1	0	0	0	1	1	0	1	0

Sumar: -328 + (-230)

	1	1	1				1	1	1					
	0	1	1	1	0	1	0	1	1	1	0	0	0	C2
+	0	1	1	1	1	0	0	0	1	1	0	1	0	C2
	1	1	1	0	1	1	1	0	1	0	0	1	0	C2

Conversión de regreso: C2 a MV, caso c.

Negar											
1	0	1	0	0	0	1	0	1	1	0	1

Sumar +1											
										1	+1
1	0	1	0	0	0	1	0	1	1	0	1

R\\

1	0	1	0	0	0	1	0	1	1	1	0

Caso 5: Números de igual módulo, pero de signo opuesto:

Se espera obtener cero exacto. Se generará un acarreo a la izquierda del bit de signo, que será despreciado.

a)	b)	c)
27	-56	-241
+ -27	+ 56	+ 241
0	0	0
Con siete bits	Con ocho bits	Con 10 bits

a) (27 + (-27) = 0) con 7 bits

$(27)_{10} = (001\ 1011)_2$ $(-27)_{10} = (101\ 1011)_2$ $(0)_{10} = (000\ 0000)_2$

MV: 27

0	0	1	1	0	1	1

MV: -27

1	0	1	1	0	1	1

C1: -27

Negar						
1	1	0	0	1	0	0

C2: -27

Sumar +1						
						+1
1	1	0	0	1	0	0

1	1	0	0	1	0	1

Sumar: 27 + (-27)

	1	1	1	1	1	1	1		
	0	0	0	1	1	0	1	1	MV
+	0	1	1	0	0	1	0	1	C2
R\\	1	0	0	0	0	0	0	0	MV

b) (-56 + 56 = 0) con 8 bits

$(-56)_{10} = (1011\ 1000)_2$ $(56)_{10} = (0011\ 1000)_2$ $(0)_{10} = (0000\ 0000)_2$

MV: -56

1	0	1	1	1	0	0	0

C1: -56

Negar							
1	1	0	0	0	1	1	1

C2: -56

Sumar +1							
				1	1	1	+1
1	1	0	0	0	1	1	1

1	1	0	0	1	0	0	0

MV: 56

0	0	1	1	1	0	0	0

Sumar: -56 + 56

	1	1	1	1	1					
	0	1	1	0	0	1	0	0	0	C2
+	0	0	0	1	1	1	0	0	0	MV
R\\\	1	0	0	0	0	0	0	0	0	MV

c) (-241 + 241 = 0) con 10 bits

$(-241)_{10} = (10\ 1111\ 0001)_2$ $(241)_{10} = (00\ 1111\ 0001)_2$ $(0)_{10} = (00\ 0000\ 0000)_2$

MV: -241

1	0	1	1	1	1	0	0	0	1

C1: -241

Negar									
1	1	0	0	0	0	1	1	1	0

C2: -241

Sumar +1									
									+1
1	1	0	0	0	0	1	1	1	0

1	1	0	0	0	0	1	1	1	1

MV: 241

0	0	1	1	1	1	0	0	0	1

Sumar: -241 + 241

	1	1	1	1	1	1	1	1	1	1		
	0	1	1	0	0	0	0	1	1	1	1	C2
+	0	0	0	1	1	1	1	0	0	0	1	MV
R\\	1	0	0	0	0	0	0	0	0	0	0	MV