Realización de operaciones en la ALU

Las operaciones aritméticas en la computadora se realizan dentro de la ALU (Unidad Aritmética – Lógica), que es una de las partes que componen la CPU. Las computadoras tienen sus propias particularidades que, al momento de realizar operaciones aritméticas, las diferencian de la forma en como las realizamos sobre papel. Algunas diferencias importantes son:

a) La representación del signo, que ya sabemos que por convención es 0, para números positivos y 1 para números negativos.

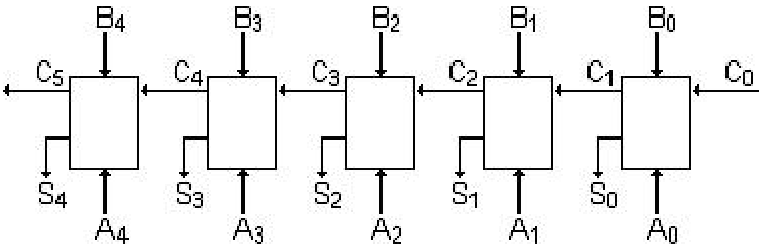
b) La cantidad limitada de bits para representar cantidades (tamaño de registro). Esto significa que, no importa qué cantidad se esté representando ,si es muy grande o muy pequeña, siempre se utiliza la misma cantidad de bits. El resultado de cualquier operación, ya sea suma, resta, producto, etc., no podrá rebasar la cantidad de bits que el dispositivo electrónico utiliza para la representación de cantidades.

c) El diseño electrónico del dispositivo para la realización de operaciones. Todas las computadoras contienen, dentro del ALU, un circuito sumador. Pero carecen de un circuito restador, el cual no se puede construir debido a su complejidad. Así que todas las operaciones, incluso las de resta, se basan en este circuito y eso implica la implementación de ciertos artificios aritméticos.

Adición en el Sistema de Complemento a Dos

La idea de la suma dentro de la computadora es exactamente igual a como la realizamos en papel: se comienza sumando desde los dígitos de la menor potencia y se van llevando los acarreos a la siguiente potencia más alta.

Gráficamente, la suma dentro de la computadora la podemos representar de la siguiente manera, donde cada cajita representa a un circuito de compuertas lógicas que se llama sumador total y la serie de cajitas conectadas es el circuito sumador:



Donde:
Ai, son los bits del sumando.
Bi, son los bits del co-sumando.
Ci, representan los acarreos.
Si, bits del resultado.

Recuerde que los números se representan en el convenio de complemento a dos, es decir, positivos en magnitud verdadera y negativos en complemento a dos. Se sumará bit a bit, comenzando desde la derecha (la menor potencia de dos). Si hay acarreo, éste participa en la suma de los bits de la izquierda. Los bits de signo también participan en la suma. Se pueden presentar cinco casos:

Caso 1: <u>Dos números positivos</u>
Esta adición es directa. Ya que ambos son positivos se espera un 0 en el bit de signo, en caso contrario, hay un error.

Ejercicios:

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

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.

Ejercicios:

a) 9	b) 220	c) 341	d) 153
+ -6	+ -82	+ -250	+ -63
3	138	91	90
Con cinco bits	Con diez bits	Con 12 bits	Con nueve bits

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.

Ejercicios:

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

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.

Ejercicios:

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

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.

Ejercicios:

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

Resolviendo el ejercicio b del caso 1:

Al convertir $(21)_{10}$ a binario se obtiene:

$(10101)_2$

Este número binario debe pasarse a magnitud verdadera con 6 bits. Se obtiene:

010101: MV (así debe quedar pues es positivo)

Al convertir $(5)_{10}$ a binario se obtiene:

$(101)_2$

Este número binario debe pasarse a magnitud verdadera con 6 bits. Se obtiene:

000101: MV (así debe quedar pues es positivo)

Al realizar la suma, se tendrá:

$$
\begin{array}{ccccccc}
 & & & {\scriptstyle 1} & & {\scriptstyle 1} & \\
 & 0 & 1 & 0 & 1 & 0 & 1 \\
+ & 0 & 0 & 0 & 1 & 0 & 1 \\
\hline
 & 0 & 1 & 1 & 0 & 1 & 0
\end{array}
$$

La respuesta es, entonces:

0 1 1 0 1 0

Este es un número positivo, como puede verse, así que está en magnitud verdadera. Al convertirlo a base diez dará 26.

Resolviendo el ejercicio d del caso 2:

Al convertir $(153)_{10}$ a binario se obtiene:

$(10011001)_2$

Este número binario debe pasarse a magnitud verdadera con 9 bits. Se obtiene:

010011001: MV (así debe quedar pues es positivo)

Al convertir $(-63)_{10}$ a binario se obtiene:

$(-111111)_2$

Este número binario debe pasarse a magnitud verdadera con 9 bits. Se obtiene:

100111111: MV

Luego pasarlo a complemento a uno:

111000000: C1

Luego pasarlo a complemento a dos:

$$
\begin{array}{cccccccccc}
 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
+ & & & & & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}
$$

111000001: C2 (así debe quedar pues es negativo)

Al realizar la suma, se tendrá:

$$
\begin{array}{cccccccccc}
 & {\scriptstyle 1} & & & & & & & {\scriptstyle 1} & \\
 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
+ & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
\cancel{1} & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0
\end{array}
$$

La respuesta es, entonces:

0 0 1 0 1 1 0 1 0

Este es un número positivo, como puede verse, así que está en magnitud verdadera. Al convertirlo a base diez dará 90.

Resolviendo el ejercicio c del caso 3:

Al convertir $(61)_{10}$ a binario se obtiene:

$(111101)_2$

Este número binario debe pasarse a magnitud verdadera con 10 bits. Se obtiene:

0000111101: MV (así debe quedar pues es positivo)

Al convertir $(-86)_{10}$ a binario se obtiene:

$(-1010110)_2$

Este número binario debe pasarse a magnitud verdadera con 10 bits. Se obtiene:

1001010110: MV

Luego pasarlo a complemento a uno:

1110101001: C1

Luego pasarlo a complemento a dos:

$$
\begin{array}{ccccccccccc}
 & & & & & & & & & {\scriptstyle 1} & \\
 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
+ & & & & & & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0
\end{array}
$$

1110101010: C2 (así debe quedar pues es negativo)

Al realizar la suma, se tendrá:

$$
\begin{array}{ccccccccccc}
 & & & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & & & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
+ & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1
\end{array}
$$

La respuesta es, entonces:

1 1 1 1 1 0 0 1 1 1

Este es un número negativo, como puede verse, así que está en complemento a dos. Para convertirlo a base diez habrá que pasarlo a magnitud verdadera y después sumar las potencias de dos de las posiciones que tienen 1 (excepto el signo). El resultado dará -25.

Resolviendo el ejercicio a del caso 4:

Al convertir $(-31)_{10}$ a binario se obtiene:

$(-11111)_2$

Este número binario debe pasarse a magnitud verdadera con 8 bits. Se obtiene:

10011111: MV

Luego pasarlo a complemento a uno:

11100000: C1

Luego pasarlo a complemento a dos:

$$\begin{array}{cccccccc c}
 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
+ & & & & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}$$

11100001: C2 (así debe quedar pues es negativo)

Al convertir $(-17)_{10}$ a binario se obtiene:

$(-10001)_2$

Este número binario debe pasarse a magnitud verdadera con 8 bits. Se obtiene:

10010001: MV

Luego pasarlo a complemento a uno:

11101110: C1

Luego pasarlo a complemento a dos:

$$\begin{array}{ccccccccc}
 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
+ & & & & & & & & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1
\end{array}$$

11101111: C2 (así debe quedar pues es negativo)

Al realizar la suma, se tendrá:

$$\begin{array}{ccccccccc}
 & {\scriptstyle 1} & {\scriptstyle 1} & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & \\
 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
+ & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
\hline
\cancel{1} & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}$$

La respuesta es, entonces:

1 1 0 1 0 0 0 0

Este es un número negativo, como puede verse, así que está en complemento a dos. Para convertirlo a base diez habrá que pasarlo a magnitud verdadera y después sumar las potencias de dos de las posiciones que tienen 1 (excepto el signo). El resultado dará -48.

Resolviendo el ejercicio a del caso 5:

Al convertir $(27)_{10}$ a binario se obtiene:

$(11011)_2$

Este número binario debe pasarse a magnitud verdadera con 7 bits. Se obtiene:

0011011: MV (así debe quedar pues es positivo)

El (-27)10 será:

1011011: MV

Luego pasarlo a complemento a uno:

1100100: C1

Luego pasarlo a complemento a dos:

$$\begin{array}{cccccccc}
 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
+ & & & & 1 & & & 1 \\
\hline
 & 1 & 1 & 0 & 0 & 1 & 0 & 1
\end{array}$$

1100101: C2 (así debe quedar pues es negativo)

Al realizar la suma se tendrá:

$$\begin{array}{cccccccc}
{\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & \\
 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
+ & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
\hline
\cancel{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

La respuesta es, entonces:

0 0 0 0 0 0 0