

Facultad de Ingeniería Eléctrica

Laboratorio de Electrónica

"Ing. Luís García Reyes"

Materia: "Laboratorio de Electrónica Digital I"

Practica Número 2

"Aritmética Binaria"

Objetivo:

En esta práctica, se evalúa la capacidad del alumno para la realización de operaciones aritméticas en base binaria utilizando circuitos digitales que realizan operaciones aritméticas.

Introducción:

Las computadoras digitales efectúan diversas tareas de procesamiento de información. Entre estas funciones se encuentran las operaciones aritméticas más básicas.

Suma:

Entrada A	Entrada B	Acarreo	Salida
1	0	0	1
1	1	0	2
1	2	0	3
1	3	0	4
1	4	0	5
1	5	0	6
1	6	0	7
1	7	0	8
1	8	0	9
1	9	1	0

Antes de iniciar, recordemos la operación de suma de dígitos decimales, al realizar cualquier suma decimal no existe problema al realizar una suma que de como resultado un número inferior a 9.

Ejemplo:

$$1 + 8 = 9$$

Pero que ocurre al realizar una suma como $1 + 9$, el resultado es 0 unidades con una unidad de acarreo, esta unidad se coloca delante del 0 y se le denomina decena, así el resultado es diez 10

$$1 + 9 = 0(\text{unidades}) + 1(\text{acarreo}) = 10$$

Este mismo procedimiento se lleva a cabo al

utilizar cualquier sistema numérico.

En el sistema Binario la operación más simple es la suma de dos dígitos binarios. Esta suma simple de dos bits consiste en cuatro posibles operaciones elementales, las cuales se muestran en la siguiente tabla donde las entradas son A y B.

En este caso las primeras tres operaciones producen un resultado de un solo dígito, pero cuando ambos bits sumados son "1", el resultado de la suma

Entrada A	Entrada B	Acarreo	Salida
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

binaria consta de dos dígitos. En donde al bit más significativo del resultado de la suma se le denomina acarreo.

Este resultado es similar al que se da en el sistema numérico decimal.

Entonces si en el sistema decimal la "segunda" posición equivale a multiplicar el dígito por la base (10), entonces en el sistema binario equivale a multiplicar al valor del dígito por 2.

En la siguiente tabla se muestra una suma de 4 bits, en donde se realizan sumas de 0000 binario a 1111 binario, es decir en decimal de 0 a 15, al realizar las sumas se tienen resultados de 0000 a 1 1110, esto quiere decir que al realizar sumas de 4 bits es posible tener un acarreo dependiendo de los sumandos.

D	C	B	A		D	C	B	A		Acarreo	D	C	B	A
0	0	0	0		0	0	0	1		0	0	0	0	1
0	0	0	1		0	0	1	1		0	0	1	0	0
0	0	1	0		0	1	1	0		0	1	0	0	0
0	1	1	1		1	0	0	0		0	1	1	1	1
0	1	0	0		1	1	0	0		1	0	0	0	0

Resta:

La operación de resta no es "sencilla" de implementar con circuitos integrados, debido a la necesidad de utilizar memorias temporales (prestamos), así para la resta se utiliza un método alternativo que se basa en realizar la resta utilizando un sumador donde uno de los sumandos se hace "negativo", esto se puede observar en decimal al realizar la resta.

$$25 - 7 = 25 + (-7) = 18$$

En circuitos digitales así como en computadoras digitales se usan los "complementos" que equivalen a tener números negativos, para simplificar la operación de resta. Esta es la forma más conveniente de realizar la resta de números binarios utilizando circuitos integrados.

Hay dos tipos de complementos para cada sistema base, el complemento a la base y el complemento a la base disminuida.

Si utilizamos la base 2 (binaria) se tiene el complemento a "2" (complemento a la base) y el complemento a "1" (complemento a la base disminuida).

Representación de números negativos.

En el sistema de numeración de complemento a la base r , los números negativos de n dígitos se representan por la cantidad que les falta para completar r^n . Es decir, en este sistema, la cantidad $-N_r$ se representa por su complemento, es decir, como $r^n - N$ y en ocasiones se denota $[N]_r$.

Ejemplo: Caso decimal.

Complemento a 10 para 2 dígitos base =10 número de dígitos = 2 ($r=10, n=2$) por ejemplo para representar el número 15 en complemento se realiza la siguiente operación:

$-15_{10} = 10^2 - 15 = 100 - 15 = [85]_{10}$ De esta manera, una resta de por ejemplo $45 - 15$, se puede realizar mediante la suma

$$45 + 85 = (1)30$$

Despreciando el acarreo indicado entre paréntesis, ya que sólo se están usando 2 dígitos.

Caso binario.

Complemento a 2 para 4 dígitos base = 2 número de dígitos = 2($r=2, n=4$) de la misma manera para representar el número -5 base 2, se realiza la siguiente operación:

$(-0101)_2 = 2^4 - [0101]_2 = (10000 - 0101)_2 = [1011]_2$, de esta manera, una resta como $[1010]_2 - [0101]_2$, se puede realizar mediante la suma de $[1010]_2 + [1011]_2$.

$$[0101]_2 + [1011]_2 = [(1)0101]_2$$

Despreciando el acarreo indicado entre paréntesis, ya que sólo se están usando 4 dígitos.

Observación: En el sistema de complemento a dos, los números positivos se escriben sin ningún cambio.

Bit de signo (S): En el sistema de numeración de complemento a dos el MSB se denomina bit de signo y se usa para indicar el signo del número representado, de acuerdo a la siguiente convención:

S = 0 El número es positivo y el resto de los bits indica su magnitud directamente.

S = 1 El número es negativo y está en la forma complementada

Ejemplo: Expresar +5 y -5 en una palabra de 8 bits en el sistema de complemento a 2.

+ 5 es positivo y se expresará directamente por su magnitud en binario como 0000101

- 5 es negativo y estará expresado en la forma de complemento a 2 como:

$$+ 5 = (0\ 0\ 0\ 0\ 0\ 1\ 0\ 1)_2$$

$$- 5 = (1\ 1\ 1\ 1\ 1\ 0\ 1\ 1)_2 \text{ en complemento a 2}$$

Observe que de acuerdo a esta convención del sistema de complemento a dos, al aplicar el complemento a 2 a un número binario, equivale a cambiarle el signo (multiplicar por -1)

Ejemplo:

$(1\ 1\ 0\ 1\ 0\ 1\ 1)_2$ es un número de 7 bits, incluyendo el bit signo. El equivalente decimal es:

Primero el bit signo es = 1, el número es negativo y se encuentra en su forma complementada.

1 1 0 1 0 1 1 : número negativo

0 0 1 0 1 0 1 : valor absoluto (complemento a dos del número)

21 : equivalente decimal del complemento entonces: $[1 1 0 1 0 1 1]_2 = -21_{10}$

Ejemplo:

Si $(0 1 1 0 1 0 1 1)_2$ es un número de 8 bits, incluyendo el bit signo. El equivalente decimal es:

Debido a que el bit de signo es = 0, el número es positivo y el resto de los bits indican su magnitud.

número positivo: $(0 1 1 0 1 0 1 1) = 107_{10}$

valor absoluto : $(0 1 1 0 1 0 1 1)$

entonces : $(0 1 1 0 1 0 1 1)_2 = +107_{10}$

Ejemplo: Cuál es el equivalente en binario de complemento a dos de $(-535)_{10}$, para resolver este problema primero usaremos hexadecimal como paso intermedio a binario. Convirtiendo +535 a Hexadecimal:

$$535/16 = 33, \text{ residuo} = 7$$

$$33/16 = 2, \text{ residuo} = 1$$

$$2/16 = 0, \text{ residuo} = 2$$

$$\text{Resultado} = (217)_{16}$$

$$\text{Por tanto } (+535)_{10} = (217)_{16} = (010\ 0001\ 0111)_2 = 010\ 0001\ 0111$$

Así $(101\ 1110\ 1001)_2$ es complemento a dos por lo tanto $(-535)_{10} = [101\ 1110\ 1001]_2$ observe que para indicar este resultado se requieren por lo menos 11 bits

COMPLEMENTO A LA BASE DISMINUIDA Una alternativa al sistema de complemento a la base es el complemento a la base menos 1. en este sistema, un número negativo $-N$ de n dígitos en base r se representa por lo que le falta para llegar a $r^n - 1$ y se suele denotar por $[N]_{r-1}$, es decir, $[N]_{r-1} = r^n - 1 - N$. en otra palabras, $[N]_{r-1} = [N]_r - 1$.

Ejemplos:

Caso decimal para 2 dígitos ($r=10, n=2$, complemento a 9) $(-15)_{10} = (10^2 - 1) - 15 = 99 - 15 = [84]_9$ de esta manera, una resta como $45 - 15$, se puede realizar mediante la suma $45 + 84 = (1) + 29$, (sumando el acarreo indicado entre paréntesis como una corrección del resultado).

Caso binario para 4 dígitos ($r=2, n=4$, complemento a 1) $(-0101)_2 = (2^4 - 1) - (0101)_2 = (10000 - 1)_2 - (0101)_2 = (1111)_2 - (0101)_2 = [1010]_1$, de esta manera, una resta como $(1010)_2 - (0101)_2 = (0101)_2$ se puede realizar mediante la suma $(1010)_2 + (1010)_2 = [(1)0100]_2 + (1)_2 = (0101)_2$ i **Sumando el acarreo indicado entre paréntesis !**

OBTENCIÓN DE COMPLEMENTOS BINARIOS SIN USAR RESTAS. La ventaja que provee el sistema de complementos al convertir una resta en una suma no sería tal si para obtener el

complemento usamos la definición, ya que esta requiere una resta. A continuación se describen dos algoritmos que permiten obtener el complemento a 2 sin usar restas:

Algoritmo 1

- 1.- Se copian los bits de LSB a MSB hasta encontrar el primer bit con valor de 1.
- 2.- A continuación se prosigue invirtiendo los bits restantes (es decir, cambiando 1's por 0's y 0's por 1's) hasta llegar al MSB.

Ejemplo.

Para obtener el complemento a 2 de $N = (0011\ 0100)_2$, se copian los primeros bits hasta el primer 1: 100 y se invierten los bits restantes: 01001 resultando finalmente: $(1100\ 1100)_2$

Algoritmo 2

- 1.- Se obtiene el complemento a 1 del número invirtiendo todos sus bits.
- 2.- Se suma 1 al resultado anterior.

Ejemplo. Para obtener el complemento a 2 de $N = 010110$, se invierten todos los bits (complemento a 1) 101001 , y se suma 1 al resultado, resultando finalmente: 101011

010110	Dato Original
101001	Se invierten los bits
+1	Se suma +1
101011	resultado

SUMAS Y RESTAS USANDO COMPLEMENTO A DOS.

A continuación se dan algunos ejemplos de restas usando complemento a dos, se introduce el concepto de sobreflujo y de bit de signo.

Sobreflujo. Se dice que en una operación (en el sistema de complemento a dos) produce un sobreflujo, cuando el resultado requiere más bits que los que se están usando en la operación. Para visualizar el concepto de *sobreflujo*, es conveniente tener en mente el **rango representable en complemento a dos con n bits**.

Ejemplo: ¿Cuál es el rango de números representables en complemento a dos con 3 bits? En este caso es simple representar todos los números e interpretarlos en decimal:

0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	-4
1	0	1	-3
1	1	0	-2
1	1	1	-1

entonces el rango es de -4 a $+3$. No es difícil concluir que en general el rango representable con n bits será de -2^{n-1} a $+2^{n-1}-1$, de acuerdo al ejemplo anterior, si intentamos sumar (por ejemplo) $3+2$ usando tres bits, el resultado (5) no podrá ser expresado con tres bits, ya que se sale del rango y se tendrá un sobreflujo.

Ejemplo: Realizar las siguientes sumas en binario de complemento a dos, usando 4 bits. Indicar el signo del resultado y en donde se produce sobreflujo:

a) $4+3$

$$\begin{array}{r} 0\ 1\ 0\ 0 \\ +\ 0\ 0\ 1\ 1 \\ \hline 0\ 1\ 1\ 1 \end{array}$$

El resultado es positivo y no hay sobreflujo

b) $5-4$

$$\begin{array}{r} 0\ 1\ 0\ 1 \\ +\ 1\ 1\ 0\ 0 \\ \hline 1\ 0\ 0\ 0\ 1 \end{array}$$

El resultado es positivo y no hay sobreflujo

c) $-4 - 5$

$$\begin{array}{r} 1\ 1\ 0\ 0 \\ +\ 1\ 0\ 1\ 1 \\ \hline 1\ 0\ 1\ 1\ 1 \end{array}$$

El resultado es positivo, lo cual no puede ser pues se están sumando dos datos negativos por lo que hay sobreflujo.

d) $6 + 5$

$$\begin{array}{r} 0\ 1\ 1\ 0 \\ +\ 0\ 1\ 0\ 1 \\ \hline 0\ 1\ 0\ 1\ 1 \end{array}$$

El resultado es negativo, lo cual no puede ser pues se están sumando dos datos positivos por lo tanto hay sobreflujo, ya que el resultado requiere de otro bit adicional para representar el resultado de la suma, que en este caso es el bit de signo.

e) $9 + 2$

Esta operación NO se puede realizar con 4 bits, ya que el 9 requiere al menos 5 bits para ser expresado en el sistema de complemento a 2

Interpretación del acarreo en el resultado de sumas en complemento a 2

"Si los signos de los números que se suman son diferentes, cualquier acarreo obtenido es falso y deberá de ser ignorado".

Ejemplo: Ver inciso (b) del ejemplo anterior.

"Si los signos de los números que se suman son iguales, el acarreo es verdadero y actúa como el bit de signo. En ausencia de acarreo se considera que este es 0, así la respuesta es positiva".

Ejemplos ver incisos (a), (c) y (d) del ejemplo anterior.

Ejemplo: Efectúe las siguientes operaciones y exprese el resultado en decimal con su signo (MSB = bit de signo)

a).-

$$\begin{array}{r}
 0\ 1\ 1\ 1\ 1\ 0\ 1\ 1 \text{ Signo positivo} \\
 + 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0 \text{ Signo negativo} \\
 \hline
 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1 \text{ El acarreo se ignora}
 \end{array}$$

bit signo = 0 = positivo

resultado = (+)00101101 = (+)45₁₀

b).-

$$\begin{array}{r}
 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1 \text{ Signo negativo} \\
 + 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0 \text{ Signo negativo} \\
 \hline
 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 1 \text{ El acarreo es el bit de signo}
 \end{array}$$

los signos son iguales, el acarreo es el bit de signo del resultado

acarreo = bit signo = 1 = (-)

resultado = (-) complemento a 2 de 00111101 = (-)11000011 = - 195₁₀

Requisitos:

Antes de realizar la práctica se requiere resolver las siguientes sumas binarias:

Suma de 2 bits con acarreo

A		B		Acarreo	Suma
0	+	0	=		
0	+	1	=		
1	+	0	=		
1	+	1	=		

Realizar las siguientes operaciones de suma en decimal y pasarlas a binario expresada en 4 bits.

A	B	Binario A	Binario B	Resultado
0	+ 1	=		
1	+ 2	=		
2	+ 3	=		
3	+ 0	=		

Realizar las siguientes operaciones de resta en formato decimal.

Para realizar la resta binaria considere utilizar 4 bits y el uso de complementos.

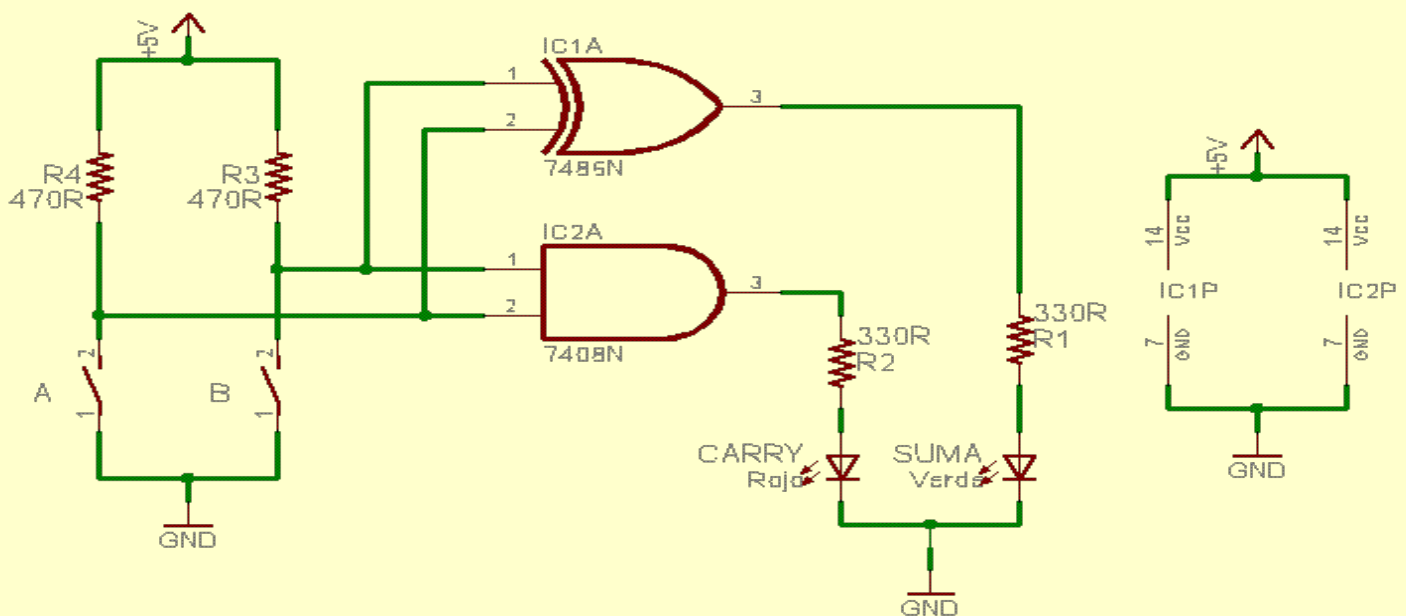
A	B	Binario A	Binario B Complemento	Resultado
3	- 1	=		
2	- 1	=		
2	- 3	=		
3	- 0	=		

A	B	Binario A	Binario B Complemento	Resultado
1	- 3	=		
2	- 1	=		
3	- 2	=		
0	- 3	=		

Desarrollo:

Parte 1

La practica consiste en implementar los circuitos de suma y resta de números binarios y comprobar las operaciones solicitadas en los prerrequisitos. La primera parte consiste de un sumador de 1 bit.



estos interruptores agrupados se les agrupa de A3 a A0. En la parte inferior se encuentra otro grupo de 4 interruptores que representan los 4 bits a sumar o restar de los bits superiores, a estos bits se les identifica como B3 a B0.

En estos interruptores B0 identifica al bit menos significativo y B3 identifica al bit más significativo.

La salida del circuito se puede observar en los led's de salida conectados. De igual manera el Led A representa al bit menos significativo y el bit D corresponde al bit más significativo, adicionalmente se tiene el bit de acarreo el cual se activa al realizar sumas que generen un acarreo en la suma.

En caso de realizar operaciones de resta, se activa el interruptor RESTA/SUMA, este interruptor tiene la función de invertir las entradas de cada uno de los interruptores utilizando las compuertas XOR (7486) y agregar el bit de acarreo al sumador.

- Utilizando este circuito, comprobar cada una de las sumas que se realizaron en los requisitos de la práctica.
- Comprobar con una punta de prueba lógica, el estado de las entradas del circuito al realizar las sumas
- Comprobar con la punta de prueba lógica el estado de las entradas del circuito al realizar las restas.

Para comprobar estos resultados se debe de llenar la siguiente tabla **por cada una de las sumas y restas**

Operación					Dato B				Dato A					Resultado						
A	B	=	R		A3	A2	A1	A0		CO	B3	B2	B1	B0		CA	D	C	B	A
					0	1	0	1		0	0	0	0	1						
5	+	1	=	6					+						=	off	off	on	on	off
					Pin	Pin	Pin	Pin		Pin	Pin	Pin	Pin	Pin						
					1	3	8	10		13	16	4	7	11						

Donde en los espacios en blanco se deben de tomar los valores lógicos que se encuentran directamente en los pines del sumador.

Reportar:

Funcionamiento de la compuerta 7486

Funcionamiento de la compuerta 7408

Funcionamiento del circuito 7483

Tablas obtenidas en la Práctica.