

# UNIDAD 5 – CIRCUITOS SECUENCIALES

Las computadoras digitales realizan diversas operaciones aritméticas con números representados en forma binaria. En esta unidad se verán los principios usados por las computadoras para realizar operaciones aritméticas básicas.

## 5-1 Circuitos Aritméticos.

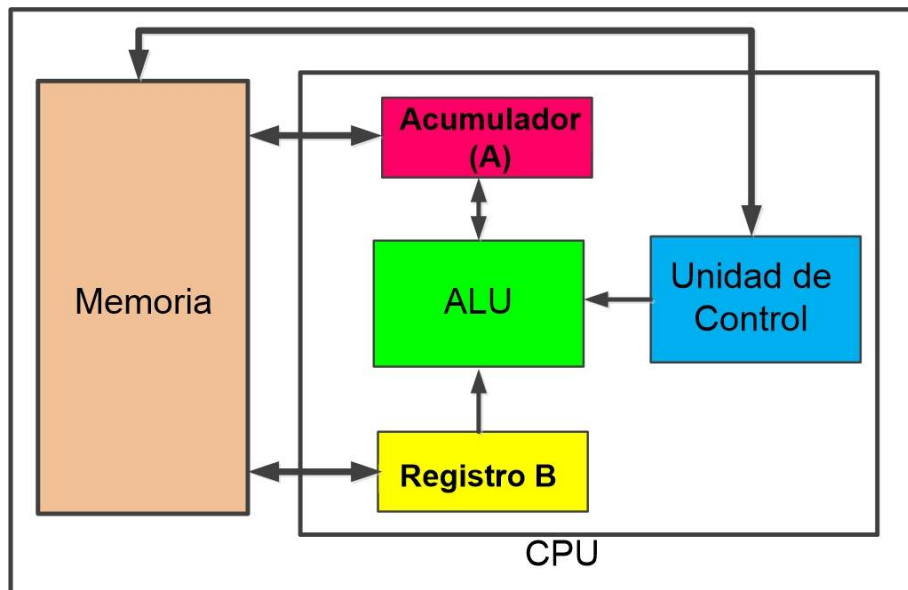
Una función de la mayoría de las computadoras es la ejecución de operaciones aritméticas.

Estas operaciones las ejecuta **la unidad aritmética/lógica (ALU) de la computadora**, compuesta por una combinación de compuertas lógicas y flip-flops para poder sumar, restar, multiplicar y dividir números binarios.

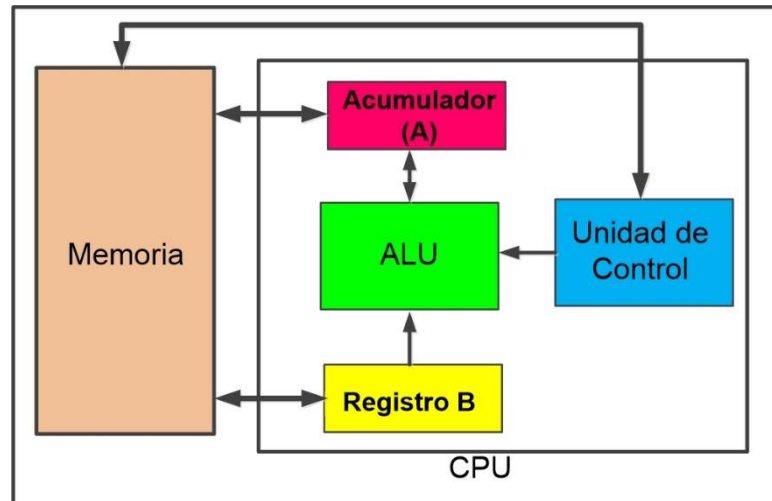
A continuación, se analizarán algunos circuitos básicos usados para realizar las operaciones aritméticas vistas anteriormente.

### La unidad aritmética/lógica.

La siguiente figura muestra un diagrama de bloques de los elementos principales de un ALU. El propósito principal del ALU es recibir datos binarios almacenados en la memoria y registros para ejecutar operaciones aritméticas y lógicas sobre estos datos, de acuerdo con las instrucciones provenientes de la unidad de control.



El ALU contiene al menos dos registros compuestos por flip-flops: el **registro B** y el **registro acumulador (A)**. Tiene lógica combinacional para realizar operaciones aritméticas y lógicas sobre números binarios almacenados en estos registros.



La **secuencia de operaciones** se realiza de la siguiente manera:

1. La unidad de control recibe de la memoria el código binario de una instrucción la cual indica que un número almacenado en una localidad de memoria se debe sumar a otro almacenado en el **registro acumulador**.
2. El número a sumar se transfiere de la memoria al **registro B**.
3. El número en el **registro B** y el número en el **registro acumulador** son sumados por el ALU cuando lo ordena la unidad de control. La suma resultante se envía al **acumulador** para ser almacenada en memoria.
4. El nuevo número en el **acumulador** puede permanecer ahí para que pueda sumarse a otro número, o si se terminó el proceso aritmético puede transferirse a la memoria para su almacenamiento.

Estos **pasos indican la razón del nombre del registro acumulador**.

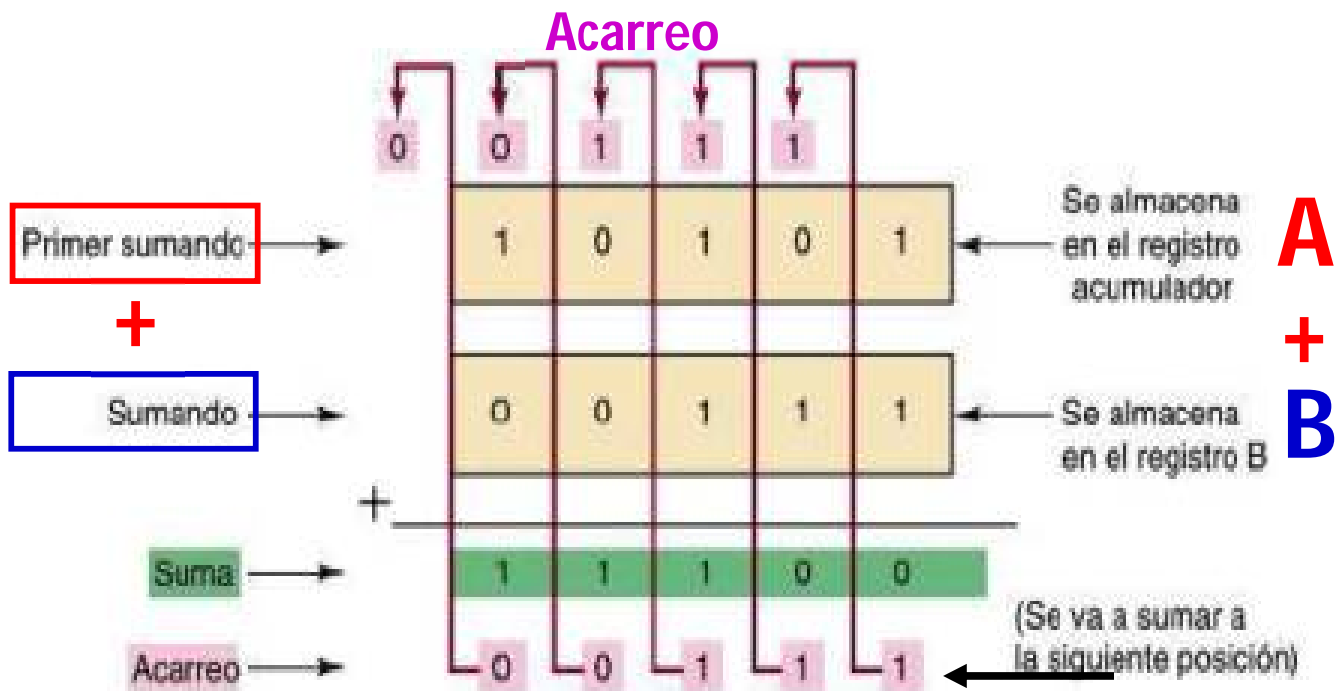
Este registro "acumula" el resultado de las sumas que ocurren cuando se realizan sumas sucesivas entre números almacenados en la memoria.

En un programa de varios pasos, el acumulador, por lo general, contiene los resultados parciales de los pasos intermedios a medida que se van completando, y el resultado final al terminar el programa.

## 5-2 El Sumador Binario en Paralelo.

Las computadoras realizan la operación de **suma sobre dos números binarios a la vez**, donde **cada número binario puede tener varios dígitos**.

La siguiente figura ilustra el **proceso de suma de dos números de cinco bits al interior del ALU**.



El primer sumando se transfiere al **registro acumulador**. El **acumulador** contiene FF que contienen los valores **10101**.

De manera similar, el **segundo sumando**, **00111**, se transfiere al registro **B**.

El proceso de adición inicia sumando los bits menos significativos (LSB) del primer y segundo sumando. De esta forma, **1+1=10**, lo cual significa que la **suma de esa posición es 0**, con un **acarreo de 1**.

El acarreo se suma a la siguiente posición con los bits del primer y segundo sumando en esa posición.

Así, en la segunda posición se suma  $1+0+1=10$ , lo cual produce un resultado de 0 y un acarreo de 1.

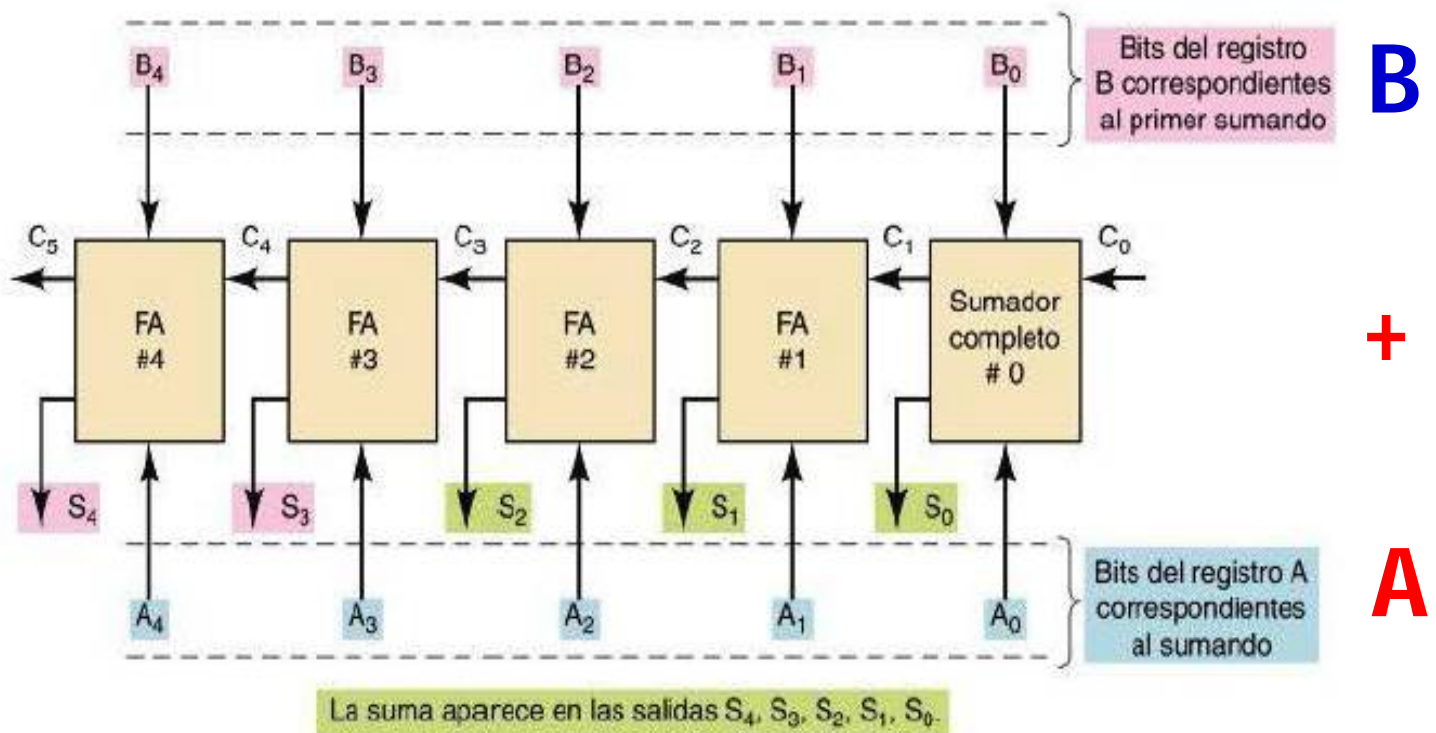
El acarreo se suma a la siguiente posición con los bits del primer y segundo sumando en esa posición, y así en lo sucesivo para el resto de las posiciones, como se muestra en la figura anterior.

En cada paso de este proceso se realiza la **suma de tres bits: el bit del primer sumando, el bit del segundo sumando y el bit de acarreo** proveniente de la posición anterior.

El **resultado de la suma** de estos tres bits **produce dos bits: un bit de suma y un bit de acarreo** que se suma a la siguiente posición.

Se sigue el mismo proceso para la posición de cada bit.

Si se diseña un circuito lógico que pueda realizar este proceso, se puede utilizar el mismo circuito para cada una de las posiciones de los bits, como se muestra en la siguiente figura.



Las variables  $A_4, A_3, A_2, A_1$  y  $A_0$  representan los bits del primer sumando almacenado en el acumulador, el **registro A**.

Las variables  $B_4, B_3, B_2, B_1$  y  $B_0$  representan los bits del segundo sumando almacenados en el **registro B**.

Las variables  $C_4, C_3, C_2, C_1$  y  $C_0$  representan los bits de acarreo en las posiciones correspondientes.

Las variables  $S_4, S_3, S_2, S_1, S_0$  son los bits de salida de la suma en cada posición.

Los bits correspondientes del primer y segundo sumando y el bit de acarreo proveniente de la posición anterior alimentan al circuito lógico conocido como **sumador completo o FA-Full Adder**.

Por ejemplo, los bits  $A_1$  y  $B_1$  alimentan al sumador completo 1, FA #1, conjuntamente con  $C_1$ , el bit de acarreo generado en la suma de los bits  $A_0$  y  $B_0$ .

Los bits  $A_0$  y  $B_0$  se conectan al sumador completo 0 (FA #0).  $C_0$ ,  $A_0$  y  $B_0$  son los LSB del primer y segundo sumando.

En principio,  $C_0$  será 0, ya que no puede haber acarreo hacia esa posición. Sin embargo, más adelante se analizarán situaciones en las que  $C_0$  es 1.

El circuito sumador completo en paralelo que se utiliza en cada posición tiene tres entradas: el bit A, el bit B y el bit C y genera dos salidas: el bit de suma y el bit de acarreo.

Por ejemplo, el sumador completo 0 (FA #0) tiene las entradas  $A_0$ ,  $B_0$  y  $C_0$ , y produce las salidas  $S_0$  y  $C_1$ . El sumador completo 1 (FA #1) tiene las entradas  $A_1$ ,  $B_1$  y  $C_1$ , y las salidas  $S_1$  y  $C_2$ , y así sucesivamente.

Este arreglo se repite para todos los bits del primer y segundo sumando.

Este ejemplo es para números de cinco bits, en las computadoras actuales la cantidad de bits varía de 8 hasta 64.

Al arreglo de la figura anterior se le conoce como **sumador en paralelo**, ya que todos los bits del primer y segundo sumando se alimentan en los circuitos del sumador de forma simultánea.

Esto significa que las sumas en cada posición se realizan al mismo tiempo, en paralelo, lo cual indica que la suma en paralelo es una **operación rápida**. Es diferente a la manera en que se suma en papel, donde se toma una posición a la vez, iniciando con el LSB.

### 5-3 Diseño de un Sumador Completo.

Ahora que se conoce la función del sumador completo, se puede diseñar un circuito lógico que desarrolle esta función.

Primero, se debe construir la tabla de verdad con los valores de las entradas y salidas para todos los casos posibles.

La figura siguiente muestra la tabla de verdad con tres entradas (A, B y  $C_{ENT}$ ) y dos salidas (S y  $C_{SAL}$ ).

Bit de entrada del primer sumando	Bit de entrada del sumando	Bit de entrada del acarreo	Bit de salida de la suma	Bit de salida del acarreo
A	B	$C_{ENT}$	S	$C_{SAL}$
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

Considérese el caso  $A=1$ ,  $B=0$  y  $C_{ENT}=1$ . El sumador completo debe sumar estos bits para producir la suma (S) de 0 y un acarreo ( $C_{SAL}$ ) de 1.

A continuación, se diseñarán los circuitos para las dos salidas, iniciando con S.

La tabla de verdad muestra que existen cuatro casos en los que S es 1. Si se usa el método de suma de productos se puede escribir la expresión de S de la siguiente manera:

$$S = \bar{A} \bar{B} C_{ENT} + \bar{A} B \bar{C}_{ENT} + A \bar{B} \bar{C}_{ENT} + A B C_{ENT} \quad (6-1)$$



Ahora, se puede tratar de **simplificar esta expresión factorizando**.

$$S = \bar{A} \bar{B} C_{ENT} + \bar{A} B \bar{C}_{ENT} + A \bar{B} \bar{C}_{ENT} + A B C_{ENT}$$

Ninguno de los términos en la expresión tiene dos variables en común con alguno de los otros. No obstante, se puede **factorizar  $\bar{A}$**  en los primeros dos términos y **A** en los dos últimos:

$$S = \bar{A}(\bar{B} C_{ENT} + B \bar{C}_{ENT}) + A(\bar{B} \bar{C}_{ENT} + B C_{ENT})$$

El **primer término** entre paréntesis es la combinación de **OR exclusivo** de **B** y **C<sub>ENT</sub>**, la cual puede escribirse como  **$B \oplus C_{ENT}$** .

El **segundo término** entre paréntesis debe es el NOR exclusivo entre **B** y **C<sub>ENT</sub>**, que puede escribirse como  **$\overline{B \oplus C_{ENT}}$** .

Así, la expresión para **S** se **convierte** en la siguiente:

$$S = \bar{A}(B \oplus C_{ENT}) + A(\overline{B \oplus C_{ENT}})$$

Si se toma que  **$X = B \oplus C_{ENT}$** , se puede escribir esto como:

$$S = \bar{A} \cdot X + A \cdot \bar{X} = A \oplus X$$

que es el **OR exclusivo** entre **A** y **X**. Si se **sustituye la expresión** para **X**, se tiene que:

$$S = A \oplus [B \oplus C_{ENT}] \quad (6-2)$$

Considérese ahora la salida  $C_{SAL}$  en la tabla de verdad.

Bit de entrada del primer sumando	Bit de entrada del sumando	Bit de entrada del acarreo	Bit de salida de la suma	Bit de salida del acarreo
A	B	$C_{ENT}$	S	$C_{SAL}$
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

Se puede escribir la expresión de suma de productos para  $C_{SAL}$  de la siguiente manera:

$$C_{SAL} = \overline{A}BC_{ENT} + A\overline{B}C_{ENT} + AB\overline{C}_{ENT} + ABC_{ENT}$$

Se simplifica esta expresión a través de la factorización, escribiendo el cuarto término  $ABC_{ENT}$  tres veces, debido a que tiene factores en común con cada uno de los otros términos, de tal forma que la expresión resultante es la siguiente:

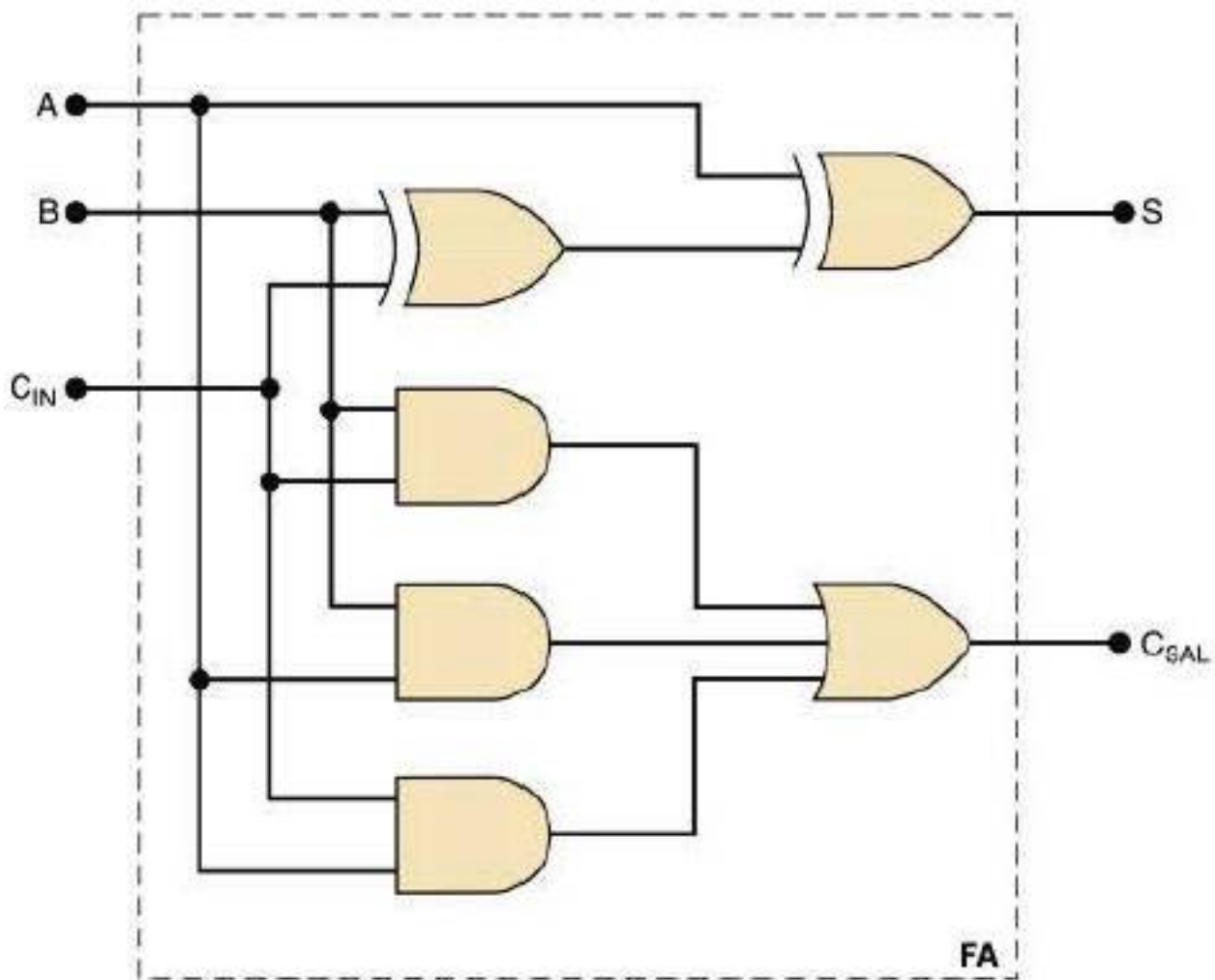
$$\begin{aligned} C_{SAL} &= BC_{ENT}(\overline{A} + A) + AC_{ENT}(\overline{B} + B) + AB(\overline{C}_{ENT} + C_{ENT}) \\ &= BC_{ENT} + AC_{ENT} + AB \end{aligned} \quad (6-3)$$

Esta expresión ya no puede simplificarse más.

Las expresiones (6-2) y (6-3) de **S** y **C<sub>SAL</sub>** pueden implantarse como se muestra en la siguiente figura.

$$S = A \oplus [B \oplus C_{ENT}] \quad (6-2)$$

$$= BC_{ENT} + AC_{ENT} + AB \quad (6-3)$$



El circuito completo con las entradas A, B, C<sub>ENT</sub> y las salidas S y C<sub>SAL</sub> representa el sumador completo. Cada uno de los FA está compuesto por el circuito de la figura anterior.

## Simplificación mediante un mapa K.

Se han simplificado las expresiones para **S** y **C<sub>SAL</sub>** usando métodos algebraicos, también puede usarse el método del mapa K.

La siguiente figura muestra el mapa K para la salida **S**.

Bit de entrada del primer sumando	Bit de entrada del sumando	Bit de entrada del acarreo	Bit de salida de la suma	Bit de salida del acarreo
A	B	C <sub>ENT</sub>	S	C <sub>SAL</sub>
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

	$\overline{C_{ENT}}$	$C_{ENT}$
AB	0	1
AB	1	0
AB	0	1
AB	1	0

Mapa K para S

$$S = ABC_{ENT} + ABC_{ENT} + ABC_{ENT} + ABC_{ENT}$$

Este mapa no tiene 1's adyacentes por lo que no existen pares o cuádruples.

En consecuencia, no se puede simplificar la expresión para **S** mediante el mapa K.

Esto muestra una limitación del método del mapa K en comparación con el método algebraico.

Se puede simplificar la expresión para **S** por medio de la factorización y del uso de las operaciones **XOR** y **XNOR**.

El mapa K para la salida  $C_{SAL}$  se muestra en la siguiente figura.

Bit de entrada del primer sumando	Bit de entrada del sumando	Bit de entrada del acarreo	Bit de salida de la suma	Bit de salida del acarreo
A	B	$C_{ENT}$	S	$C_{SAL}$
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

	$\overline{C_{ENT}}$	$C_{ENT}$
AB	0	0
AB	0	1
AB	1	1
AB	0	1

Mapa K para  $C_{SAL}$

$$C_{SAL} = BC_{ENT} + AC_{IN} + AB$$

Los tres pares producirán la misma expresión que se obtuvo mediante el método algebraico.

En una computadora, los números a sumar se almacenan en registros o localidades de memoria compuestas por FF.

El diagrama muestra un acumulador de 4 bits con los siguientes componentes:

- Registro B:** Consiste en cuatro registros de 1 bit ( $B_3, B_2, B_1, B_0$ ) que reciben datos de la memoria y tienen una línea de carga común.
- Sumador de 4 bits:** Una fila de cuatro Full Adders (FA) que calcula la suma de los bits de B y A. Las salidas de acarreo son  $C_4, C_3, C_2, C_1, C_0$ .
- Registro A:** Consiste en cuatro registros de 1 bit ( $A_3, A_2, A_1, A_0$ ) que almacenan los operandos. Cada bit tiene una línea de carga ( $CLK$ ) y una línea de borrado ( $CLR$ ).
- Control:** Las líneas **CARGAR** y **BORRAR** controlan las operaciones de carga y borrado en los registros. La línea **TRANSFERIR** controla la salida de los datos del acumulador.
- Salidas:** Las salidas de los bits  $A_3, A_2, A_1, A_0$  son las **Salidas del acumulador**.

312

El contenido del registro A (el número binario almacenado en  $A_3-A_0$ ) se suma al contenido del registro B a través de los cuatro FA y la suma se produce en las salidas  $S_3-S_0$ .  $C_4$  es el acarreo del cuarto FA y puede usarse como la entrada de acarreo para un quinto FA, o como bit de desbordamiento para indicar que la suma excedió a 1111.

Obsérvese que las salidas de la suma están conectadas a las entradas D del registro A.

Esto permitirá que el resultado de la suma se transfiera en paralelo hacia el registro A en la transición de subida (PGT) del pulso TRANSFERIR.

Obsérvese además que las entradas D del registro B provienen de la memoria de la computadora, por lo que los números binarios de la memoria se transferirán en paralelo hacia el registro B en la PGT del pulso CARGAR.

En la mayoría de las computadoras también se realiza la transferencia en paralelo de números binarios de la memoria hacia el acumulador (registro A).

Por último, obsérvese que las salidas del registro A están disponibles para realizar transferencias hacia otro registro de la computadora o a la memoria.

Esto permite que el circuito sumador se libere y esté disponible para un nuevo conjunto de números.