Circuitos electrónicos digitales

Unidades Aritméticas Lógicas

Índice

- Introducción
- Circuitos sumadores básicos
- Sumador paralelo de n bits
- Sumador/Restador
- Unidad aritmético-lógica (ALU)

Introducción

- Los sistemas digitales poseen una gran potencia de cálculo ya que permiten ejecutar con gran velocidad operaciones aritméticas y lógicas
- Una operación aritmética en un computador puede ser realizada de dos formas:
  - hardware: existe un circuito en el procesador que realiza esa operación (gran velocidad y alto coste)
  - software: existe un algoritmo que descompone esa operación en otras más elementales que son realizadas mediante hardware
- Aritmética binaria
  - Coma fija
  - Coma flotante

Introducción

- Hardware aritmético en los procesadores:
  - Todos los procesadores poseen al menos un sumador-restador
  - Los procesadores diseñados para el cálculo numérico (coprocesadores matemáticos) poseen multiplicadores, circuitos para la división, etc.
- Software aritmético en los procesadores:
  - Los procesadores más simples poseen instrucciones para la suma y la resta.
  - A medida que aumenta la complejidad se incluyen instrucciones de multiplicación y división
  - En los más complejos se tienen operaciones más abstractas como exponenciales, logaritmos, etc.

Introducción

- Las principales diferencias entre la forma de operar manual y la de un computador digital son:
  - La base del sistema de numeración es B = 2 (binaria).
  - La forma de representar números con signo normalmente no es con signo-magnitud, sino a través de los complementos (a 2 o a 1).
  - El número de bits de los datos está acotado entonces:
    - ✓ Errores de desbordamiento, de precisión
    - ✓ Incumplimiento de propiedades algebráicas: las operaciones se vuelven no-cerradas
    - ✓ Pueden incumplirse las propiedades asociativas y distributiva).

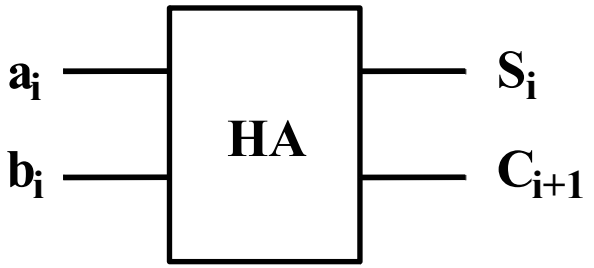
Índice

- Introducción
- Circuitos sumadores básicos
- Sumador paralelo de n bits
- Sumador/Restador
- Unidad aritmético-lógica (ALU)

Circuitos sumadores básicos

- Semisumador o Half Adder (HA)
  - Se trata del circuito que suma dos bits.
  - Obtiene como salida el bit de suma y el acarreo.

a_i	b_i	C_{i+1}	S_i
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

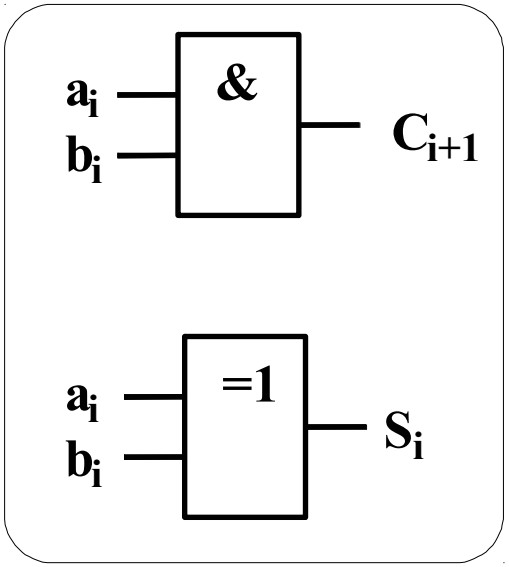


Circuitos sumadores básicos

- Semisumador o Half Adder (HA)
  - ➪ Una posible implementación mediante puertas lógicas

$$C_{i+1} = a_i \cdot b_i$$

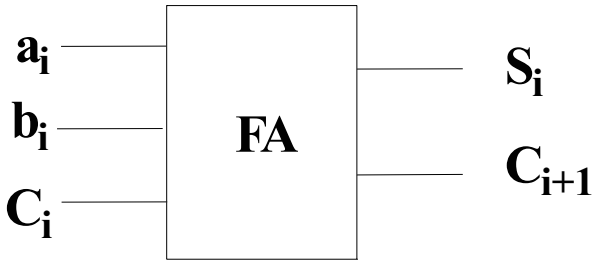
$$S_i = a_i \oplus b_i$$



Circuitos sumadores básicos

- Sumador completo Full Adder (FA)
  - Permite realizar la suma de tres bits simultáneamente.
  - Obtiene como salida el bit de suma y el acarreo.

a_i	b_i	C_i	C_{i+1}	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

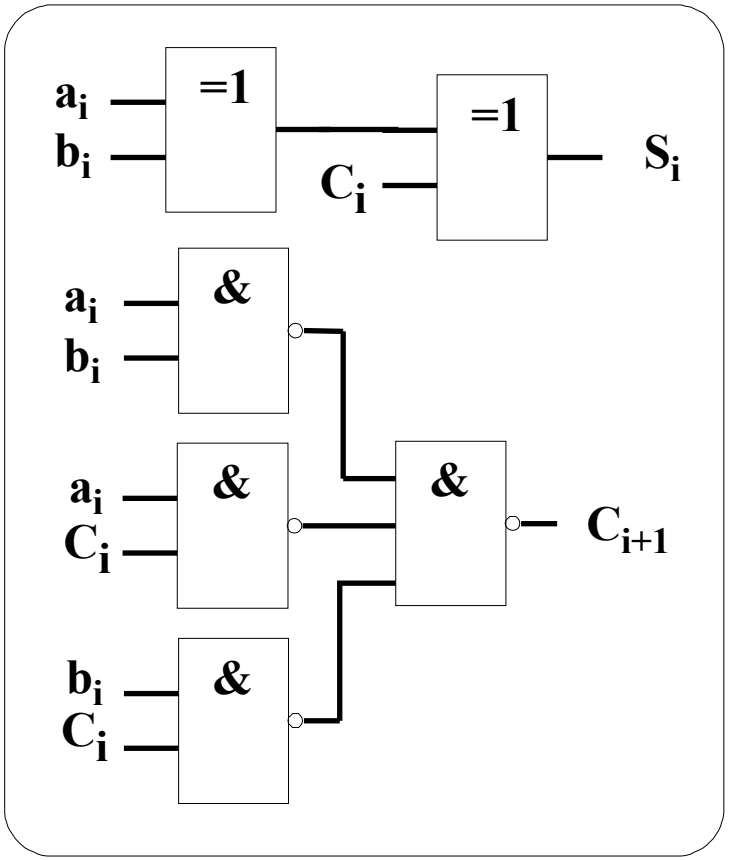


Circuitos sumadores básicos

- Sumador completo Full Adder (FA)
  - ➪ Una implementación mediante puertas lógicas

$$C_{i+1} = a_i \cdot b_i + a_i \cdot C_i + b_i \cdot C_i$$

$$S_i = a_i \oplus b_i \oplus C_i$$

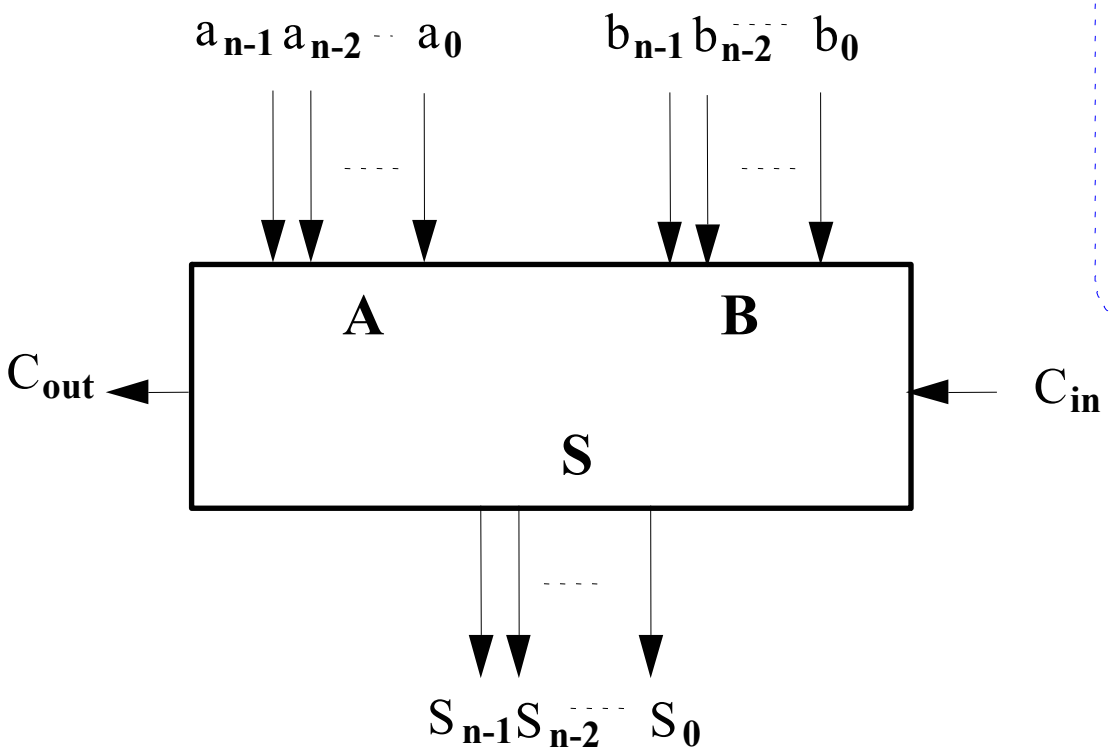


Índice

- Introducción
- Circuitos sumadores básicos
- Sumador paralelo de n bits
- Sumador/Restador
- Unidad aritmético-lógica (ALU)

Sumador paralelo de n bits

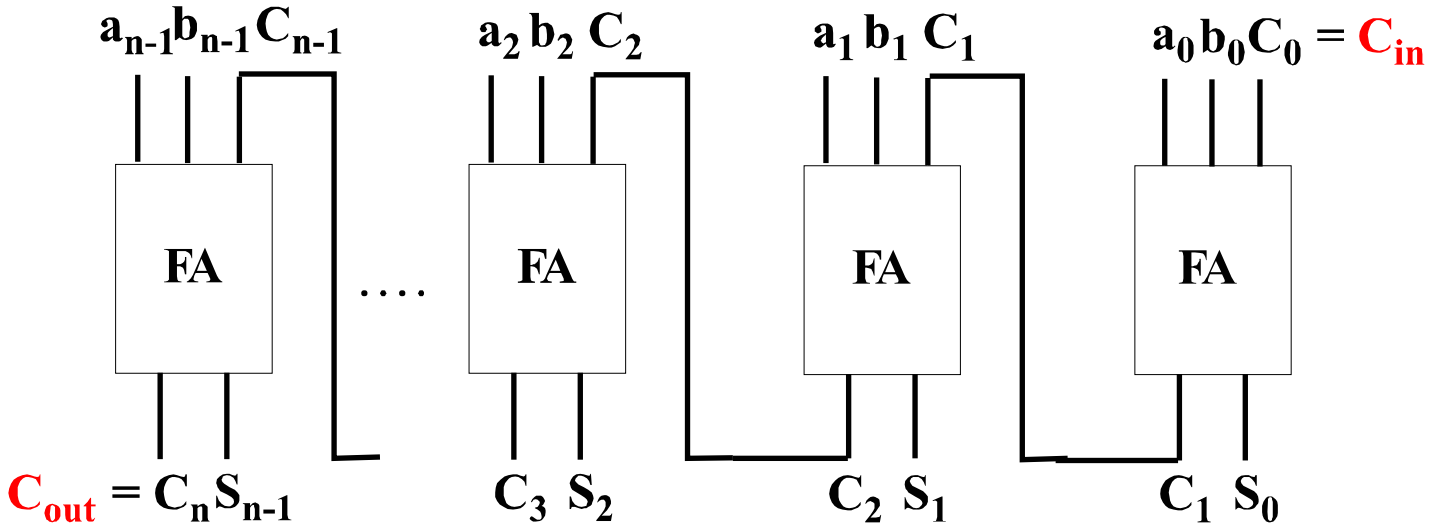
- Un sumador paralelo de n bits, es un dispositivo lógico combinacional de 2n+1 entradas y n+1 salidas que realiza la suma de dos números binarios de n bits.



$$\begin{array}{rl} C_{out}\ c_{n-1} \ldots c_2\ c_1\ c_0 = C_{in} & \\ + \quad a_{n-1} \ldots a_2\ a_1\ a_0 & \\ \quad\ b_{n-1} \ldots b_2\ b_1\ b_0 & \\ \hline \quad S_{n-1} \ldots S_2\ S_1\ S_0 & \end{array}$$

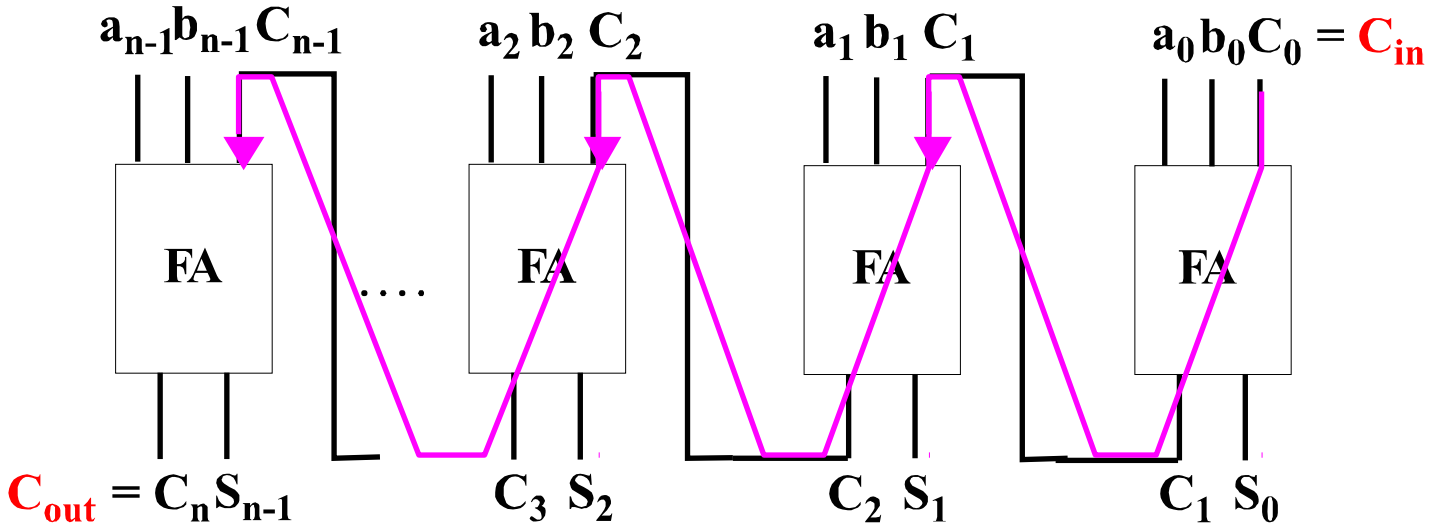
Sumador paralelo de n bits

- Sumador paralelo con acarreo serie
  - Es el más intuitivo y tiene un coste razonablemente bajo.
  - También es conocido como sumador de rizado o ripple adder
  - Se trata de un circuito modular



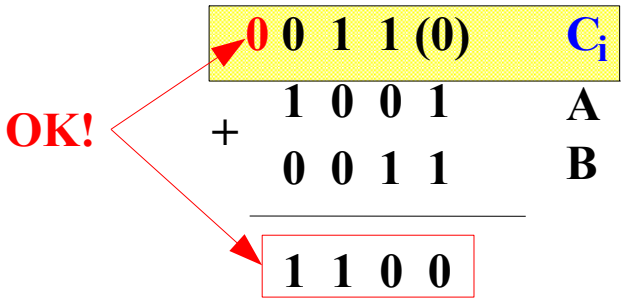
Sumador paralelo de n bits

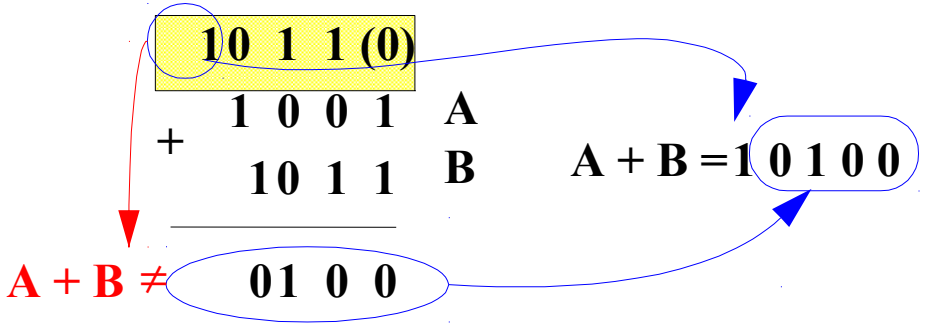
- Es lento debido a la propagación serie del acarreo
- El tiempo que tarda en realizarse una suma crece linealmente con el número de bits



Sumador paralelo de n bits

- El problema del desbordamiento en la suma de magnitudes
  - Con n bits el rango representable es [0,2n-1]
  - Si A+B > 2n-1
    - ✓ el resultado no es representable
    - ✓ hay desbordamiento (overflow)
  - Cout señala la existencia de desbordamiento
  - En caso de desbordamiento, el resultado correcto está en el número de n+1 bits (A + B = CoutSn-1 .... S0)



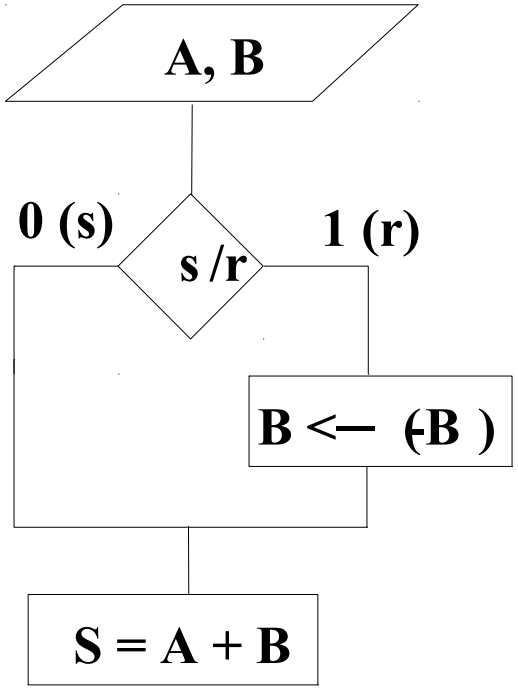


Índice

- Introducción
- Circuitos sumadores básicos
- Sumador paralelo de n bits
- Sumador/Restador
- Unidad aritmético-lógica (ALU)

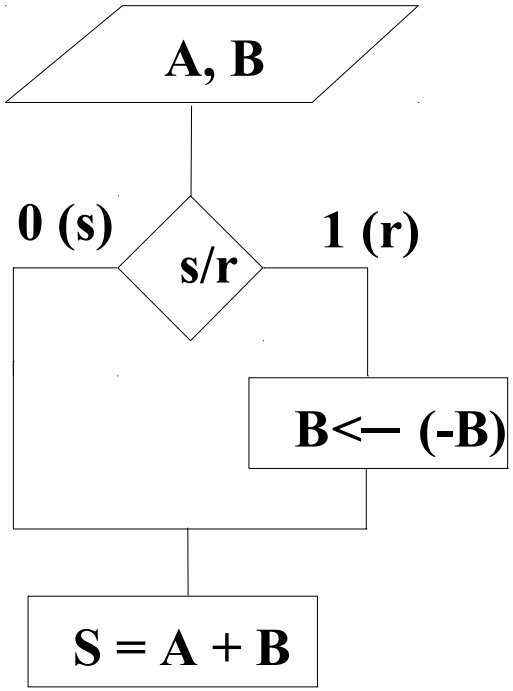
Sumador/Restador

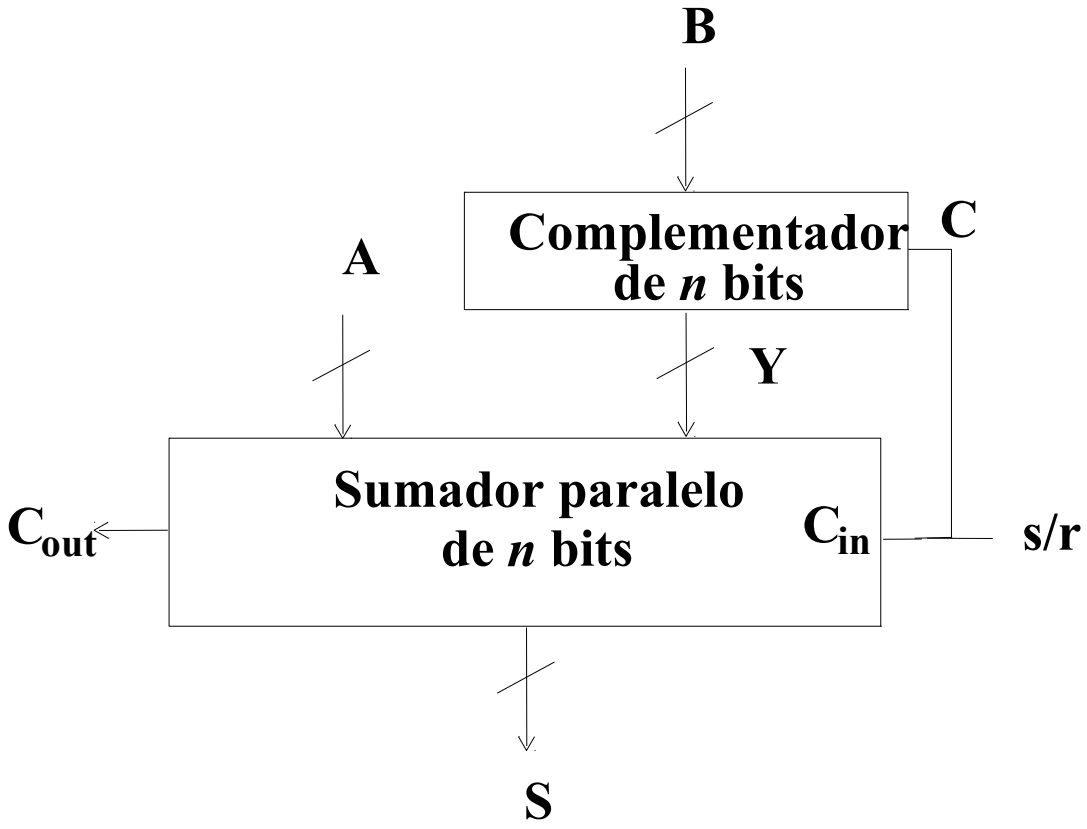
- La suma-resta de números con signo
  - Calcular la diferencia A-B es equivalente a calcular A + (-B)
    - ✓ la resta aritmética se reduce a una suma
    - ✓ implica trabajar con números con signo
  - B es (- B) en complemento a 1
  - [B + 1] es (- B) en complemento a 2



Sumador/Restador

- La suma-resta de números en complemento a 2
  - ➪ En general: A – B = A + (-B)

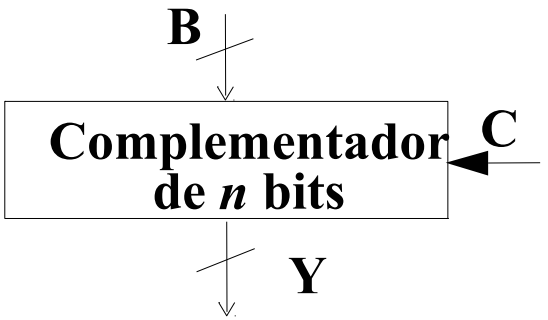


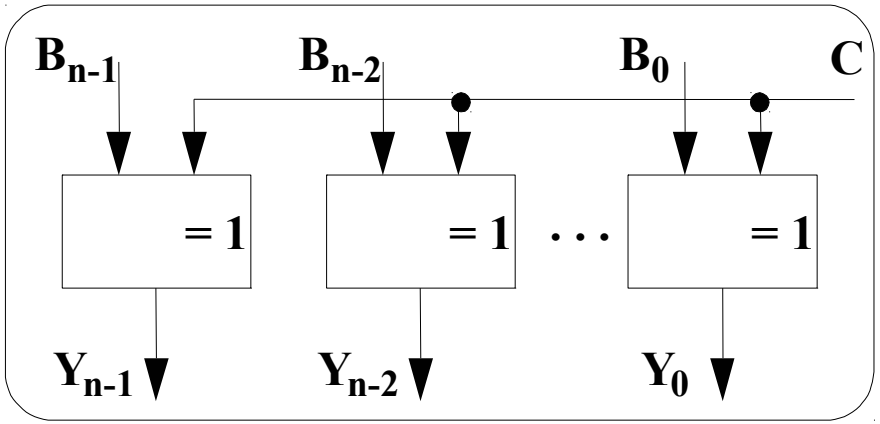


Sumador/Restador

- El complementador es simplemente una colección de puertas XOR

C	Y
0	B
1	$\overline{B}$





Sumador/Restador

- La suma-resta de números en complemento a 2
  - Utilizaremos la notación complemento a 2 para representar los números positivos y negativos

$$\begin{array}{r} 1001 = -7 \\ 0101 = +5 \\ \hline 1110 = -2 \end{array}$$

$$\begin{array}{r} 1100 = -4 \\ 0100 = +4 \\ \hline 10000 = 0 \end{array}$$

$$\begin{array}{r} 0011 = +3 \\ 0100 = +4 \\ \hline 0111 = +7 \end{array}$$

$$\begin{array}{r} 1100 = -4 \\ 1111 = -1 \\ \hline 11011 = -5 \end{array}$$

$$\begin{array}{r} 0101 = +5 \\ 0100 = +4 \\ \hline 1001 = -7 \end{array}$$

$$\begin{array}{r} 1001 = -7 \\ 1010 = -6 \\ \hline 10011 = +3 \end{array}$$

desbordamiento

Sumador/Restador

- El problema del desbordamiento en la suma-resta de números con signo
  - Se pone de manifiesto porque la magnitud ocupa un bit más y el bit de signo no es correcto
  - En caso de desbordamiento, el resultado correcto está en el número de n+1 bits A + B = CoutSn-1 ... S0
  - La detección del desbordamiento se lleva a cabo mediante una señal adicional: el bit de overflow (V)

Sumador/Restador

- Problema de desbordamiento en la suma-resta de números con signo. En la suma, el desbordamiento se produce cuando:
  - ➪ al sumar dos números positivos se obtiene uno negativo
  - ➪ al sumar dos números negativos se obtiene uno positivo

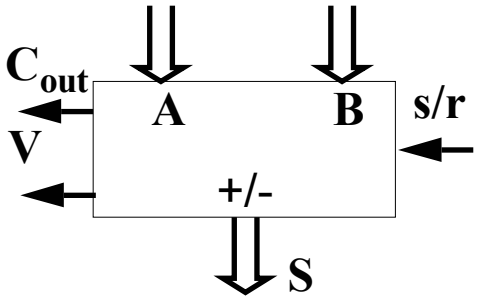
C_n	C_{n-1}			
0	1			
	0	a_{n-2}	$\cdots$	a_0
+	0	b_{n-2}	$\cdots$	b_0
	1	S_{n-2}	$\cdots$	S_0

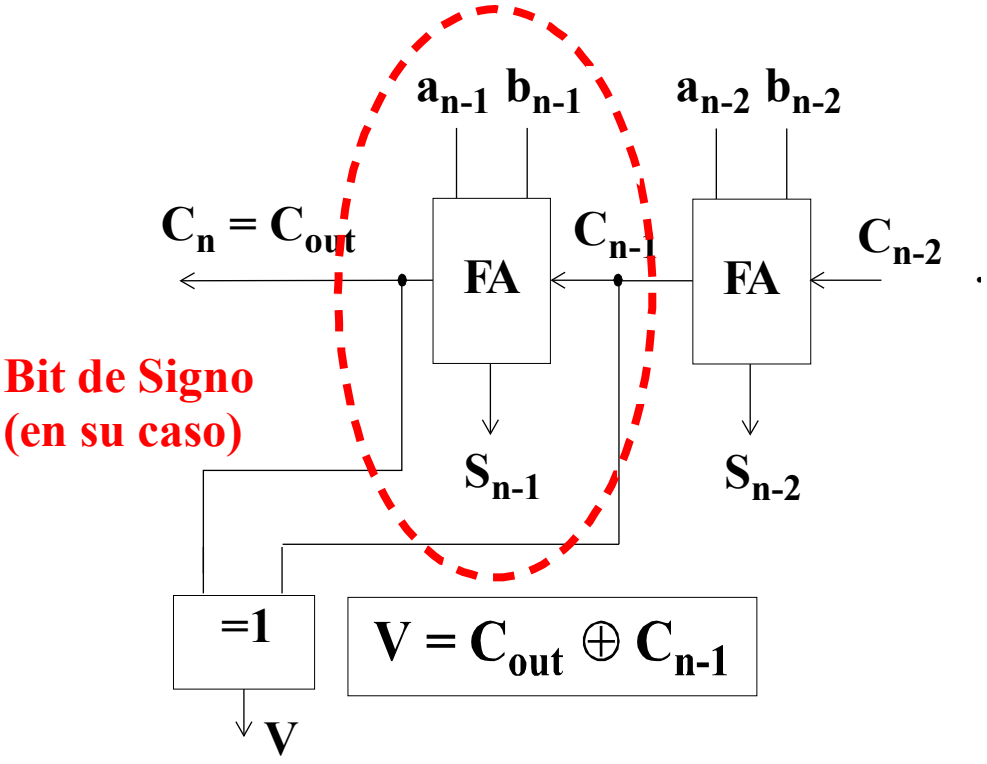
C_n	C_{n-1}			
1	0			
	1	a_{n-2}	$\cdots$	a_0
+	1	b_{n-2}	$\cdots$	b_0
	0	S_{n-2}	$\cdots$	S_0

$$V = C_{out} \oplus C_{n-1}$$

Sumador/Restador

El sumador restador quedaría:





Índice

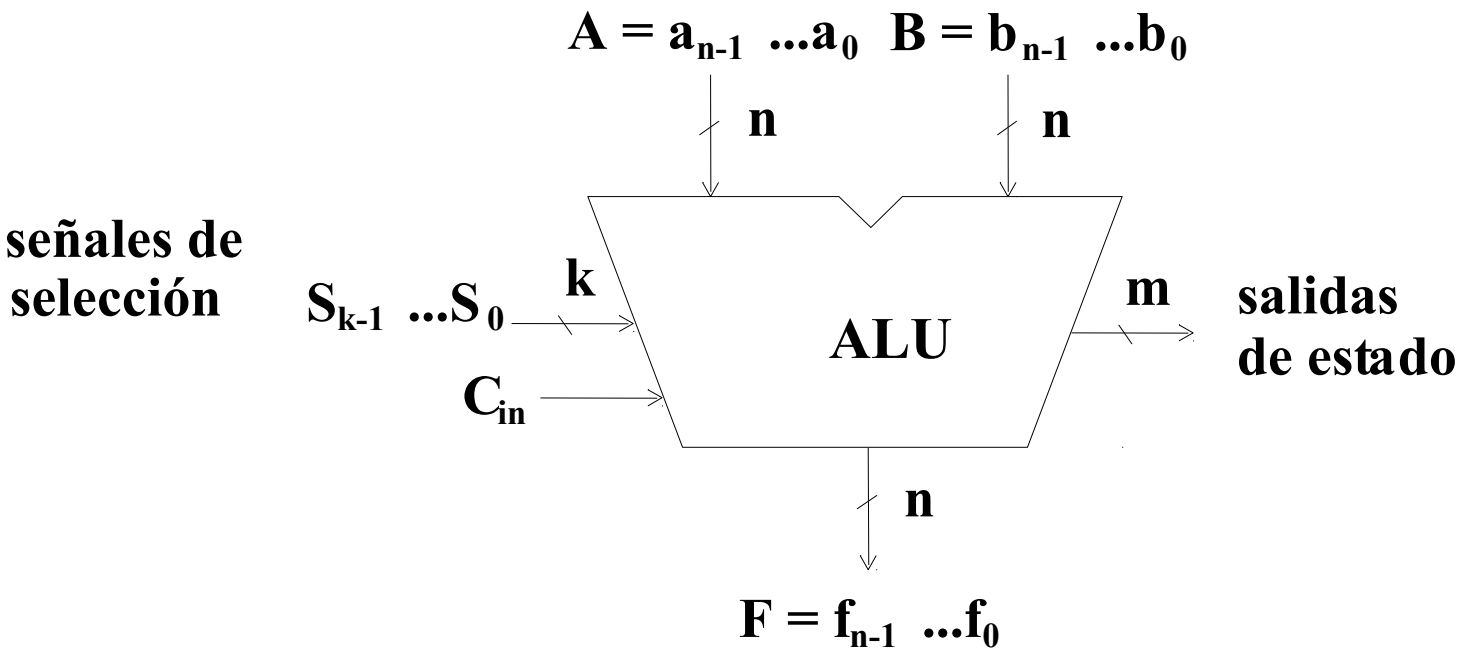
- Introducción
- Circuitos sumadores básicos
- Sumador paralelo de n bits
- Sumador/Restador
- Unidad aritmético-lógica (ALU)

Unidad aritmético-lógica (ALU)

- Es el circuito donde se realiza el procesado de datos
- Procesado: operaciones aritméticas y lógicas. Normalmente se opera sobre dos datos
- Usualmente pueden realizar diversas operaciones. Para elegirlas se incluyen unas señales de selección
- Además de las salidas que muestran el resultado de la operación, se incluyen otras salidas (flags) de estado o de condición.
- Típicamente son C_{out}, V, Z (Z=1 si el resultado es 0) y S (signo)

Unidad aritmético-lógica (ALU)

Representación gráfica de una ALU



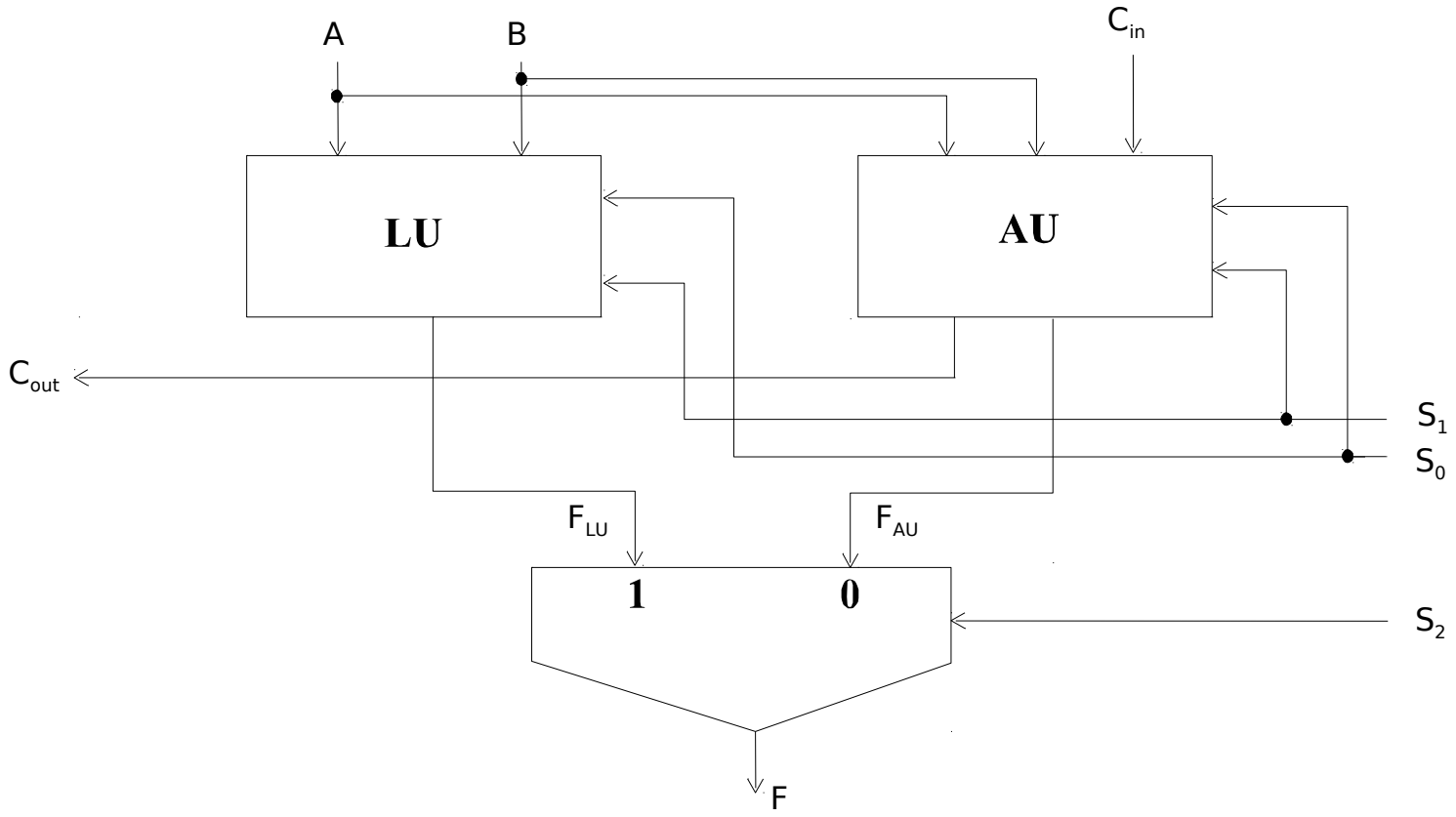
Unidad aritmético-lógica (ALU)

Ejemplo de una ALU

S_2 S_1 S_0	Función ALU	
	$C_{in} = 0$	$C_{in} = 1$
0 0 0	F = A	F = A + 1
0 0 1	F = A + B	F = A + B + 1
0 1 0	F = A + $\overline{B}$	F = A + $\overline{B}$ + 1
0 1 1	F = A – 1	F = A
1 0 0	F = A AND B	
1 0 1	F = A OR B	
1 1 0	F = NOT A	
1 1 1	F = A XOR B	

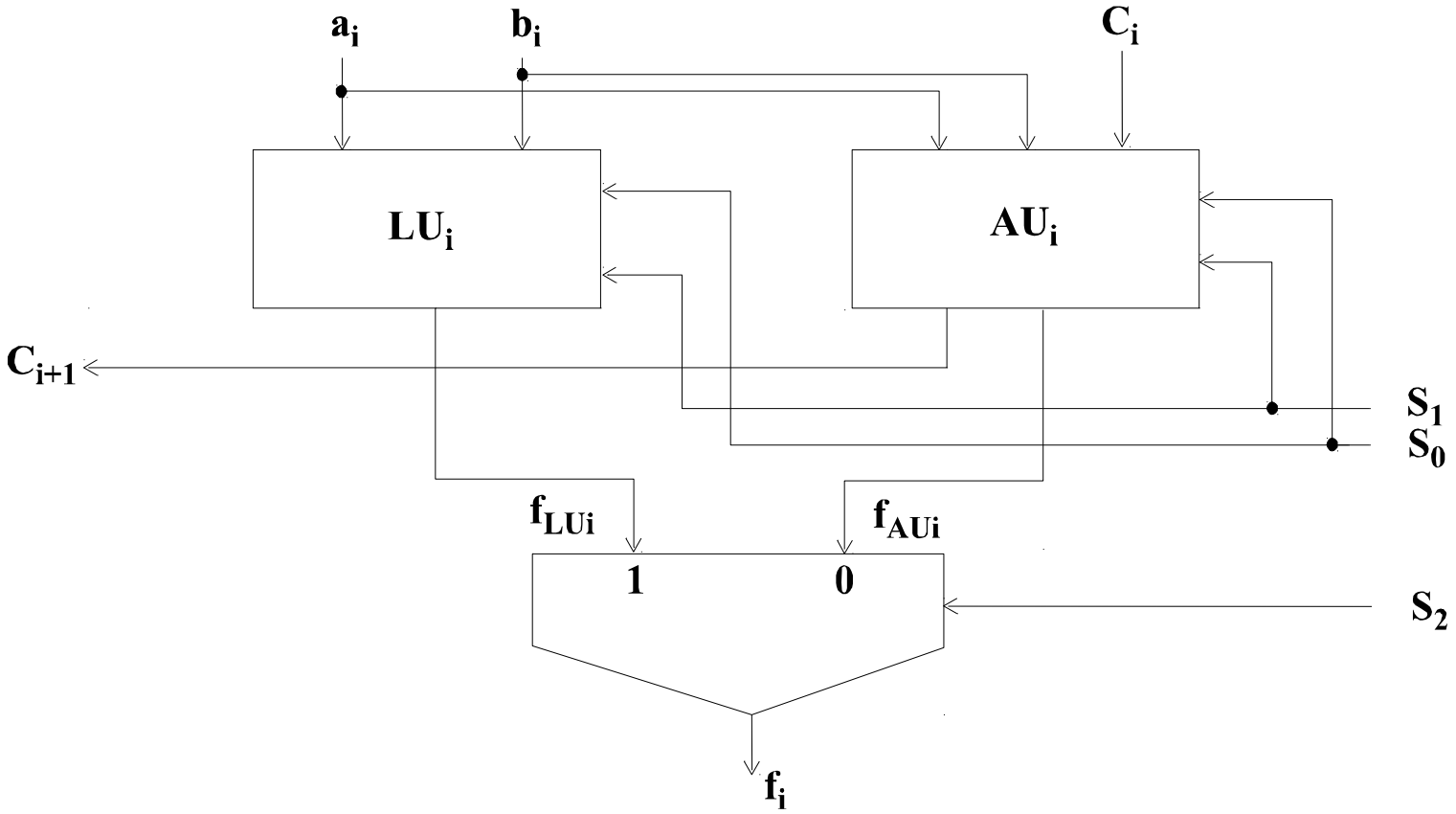
Unidad aritmético-lógica (ALU)

- Realización de una ALU
  - ➪ Se separan las partes aritmética (AU) y lógica (LU).



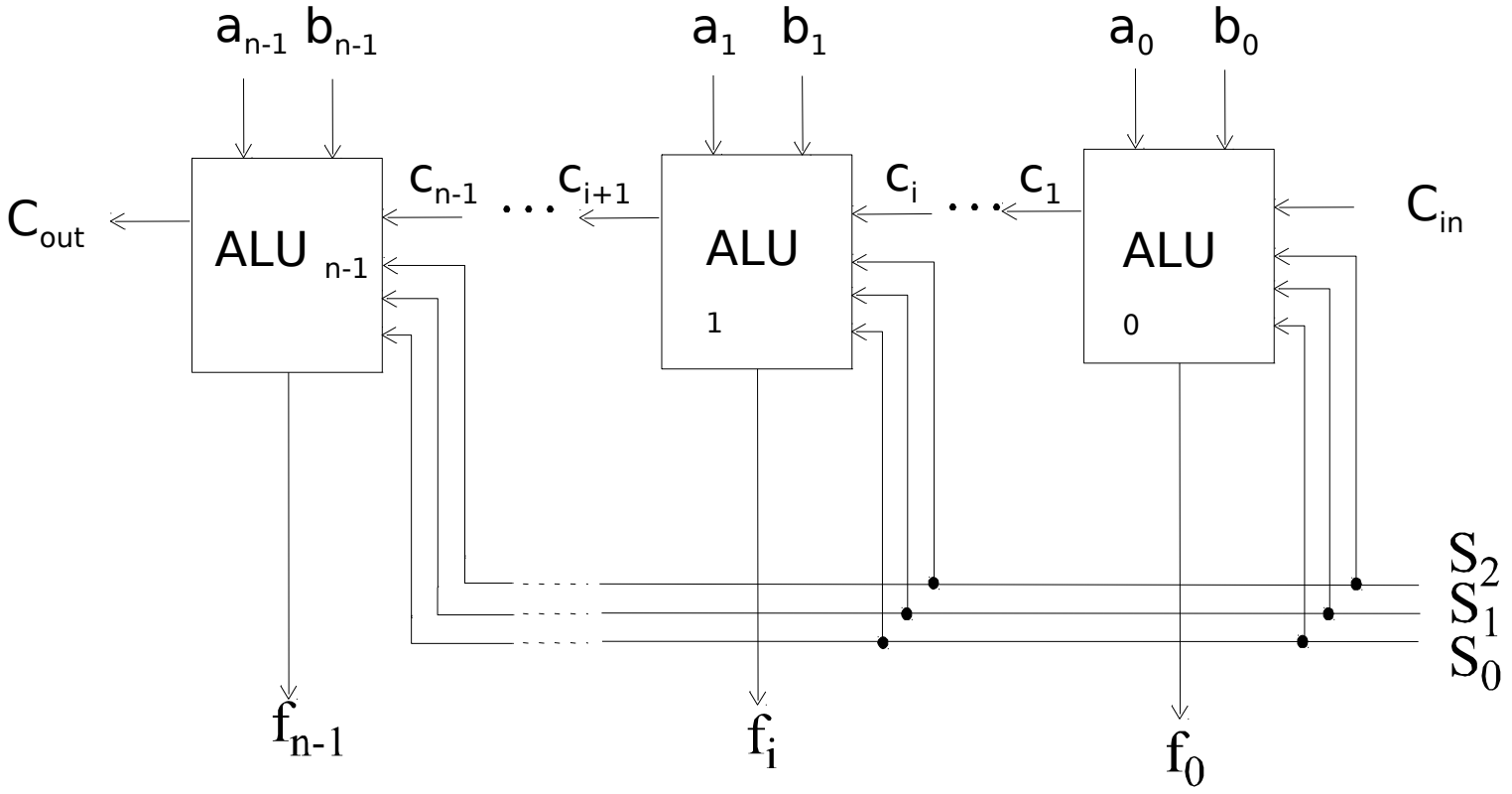
Unidad aritmético-lógica (ALU)

- Realización de una ALU
  - ➪ Implica la realización de la ALU para cada pareja de bits entrantes (etapa típica)



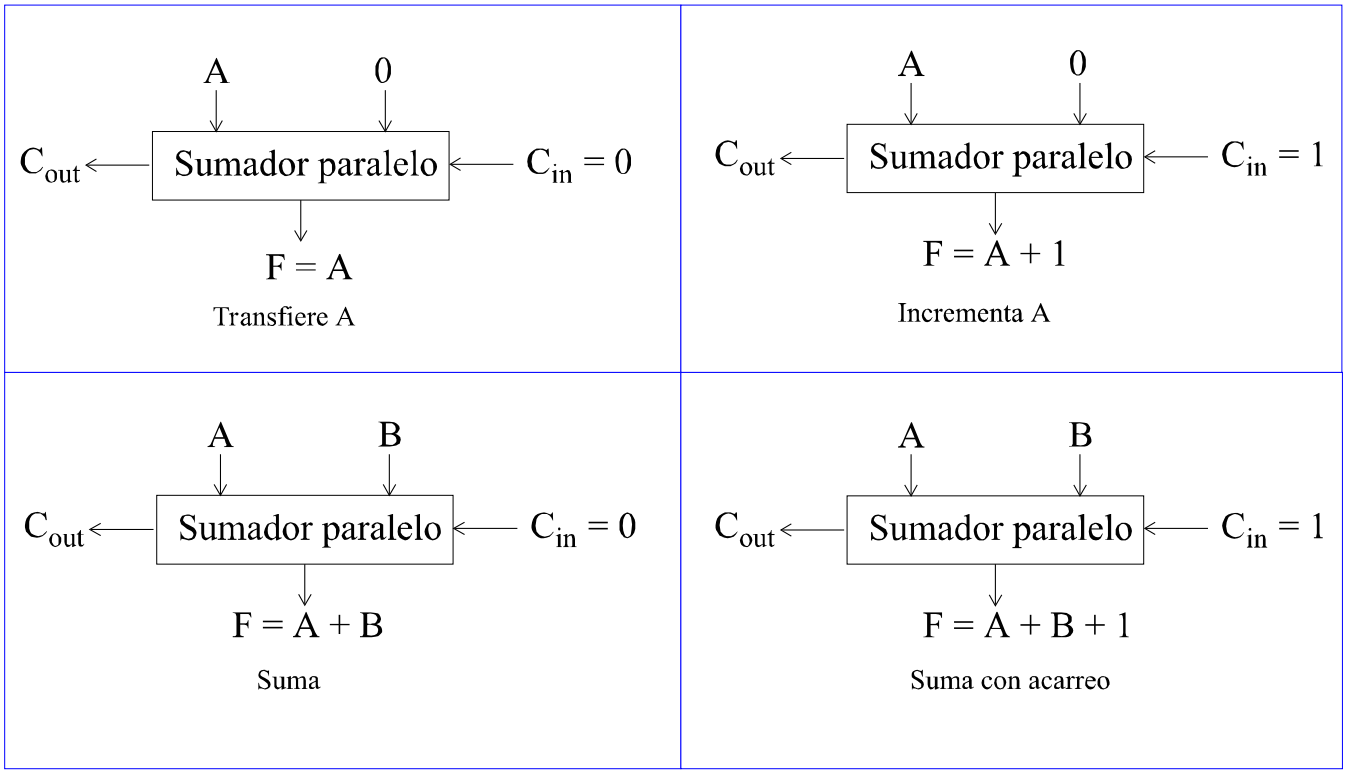
Unidad aritmético-lógica (ALU)

- Realización de una ALU
  - La ALU de n bits se implementa como cascada de n módulos de 1 bit (n etapas típicas):



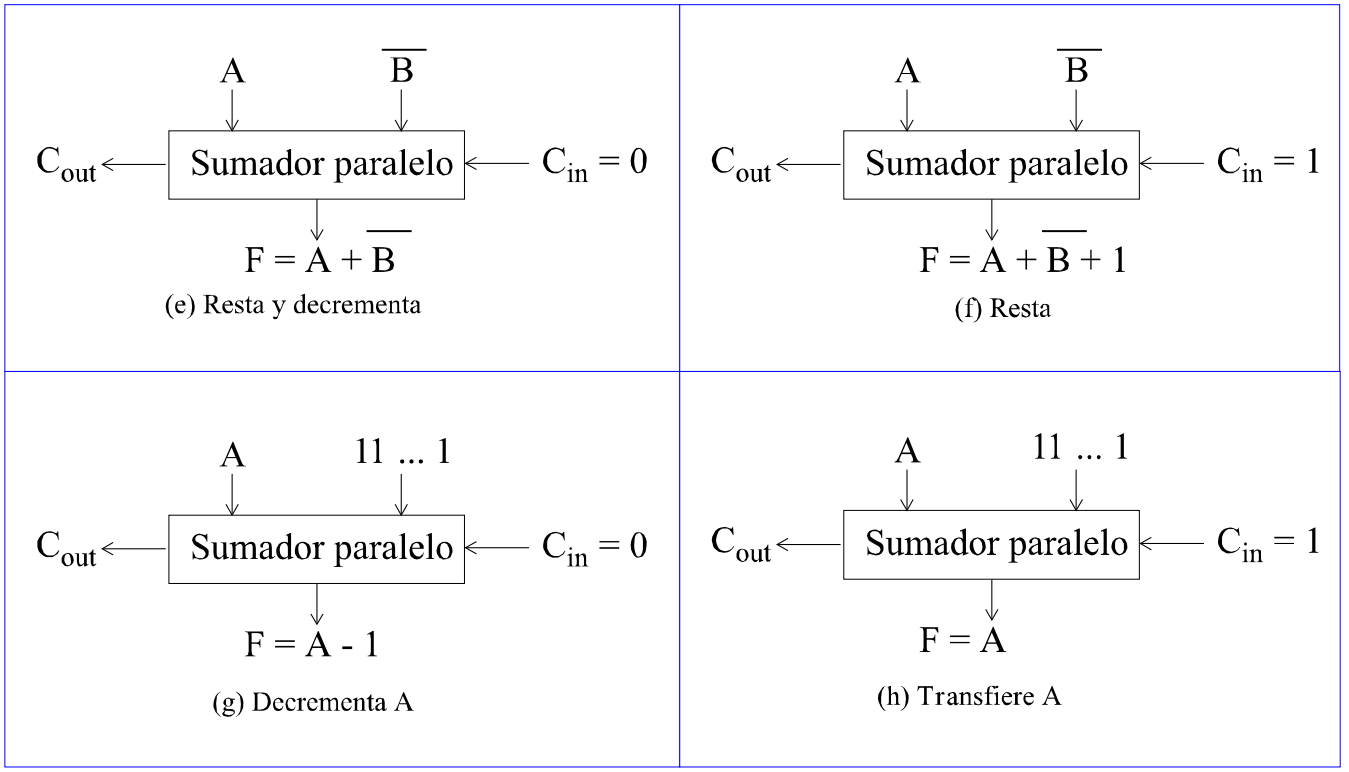
Unidad aritmético-lógica (ALU)

- Diseño de la Unidad Aritmética
  - El bloque aritmético consta básicamente de un sum. paralelo
  - Para obtener las diferentes operaciones se ha de modificar los datos de entrada al sumador



Unidad aritmético-lógica (ALU)

- Diseño de la Unidad Aritmética
  - El bloque aritmético consta básicamente de un sum. paralelo
  - Para obtener las diferentes operaciones se ha de modificar los datos de entrada al sumador



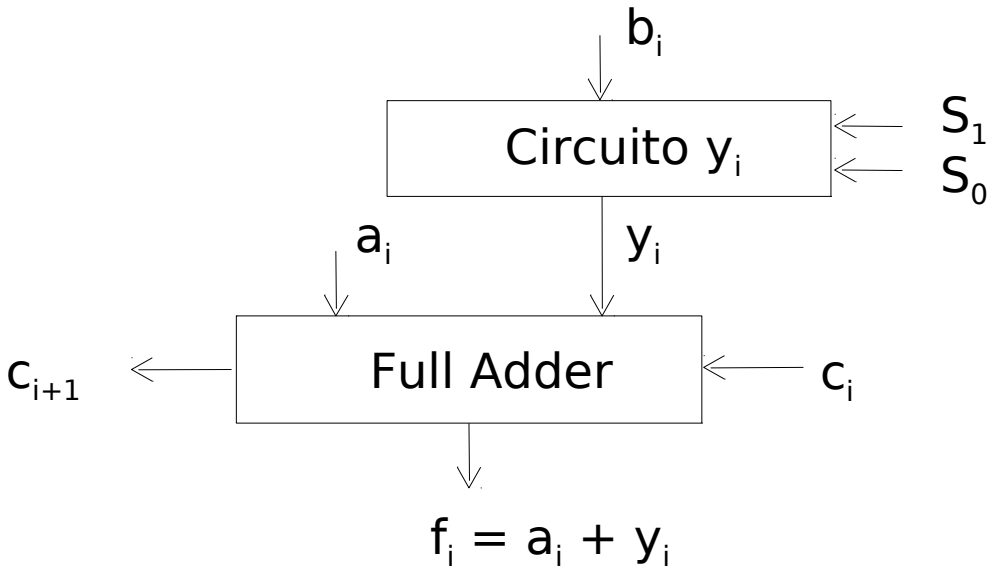
Unidad aritmético-lógica (ALU)

- Diseño de la Unidad Aritmética
  - El bloque aritmético consta básicamente de un sum. paralelo
  - Para obtener las diferentes operaciones se ha de modificar los datos de entrada al sumador

S_2 S_1 S_0	Función ALU					
	$C_{in} = 0$			$C_{in} = 1$		
0 0 0	F = A	A	0	F = A + 1	A	0
0 0 1	F = A + B	A	B	F = A + B + 1	A	B
0 1 0	F = A + $\overline{B}$	A	NOT (B)	F = A + $\overline{B}$ + 1	A	NOT (B)
0 1 1	F = A – 1	A	11…1	F = A	A	11….1

Unidad aritmético-lógica (ALU)

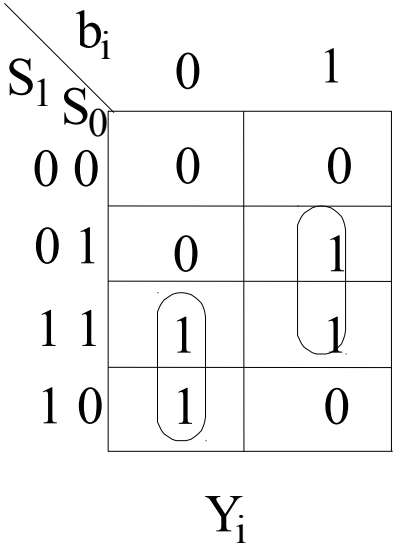
- Diseño de la Unidad Aritmética
  - El bloque aritmético consta básicamente de un sum. paralelo
  - Para obtener las diferentes operaciones se ha de modificar los datos de entrada al sumador

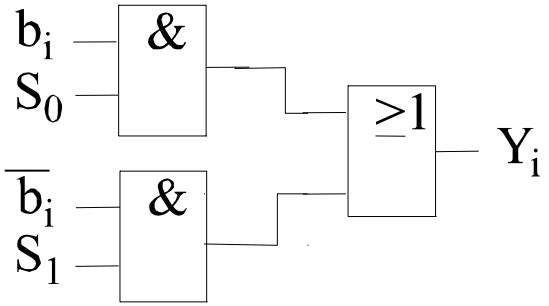


Unidad aritmético-lógica (ALU)

- Diseño del “circuito yi“ con puertas lógicas

$S_1 S_0$	Y_i
0 0	0
0 1	b_i
1 0	$\overline{b_i}$
1 1	1

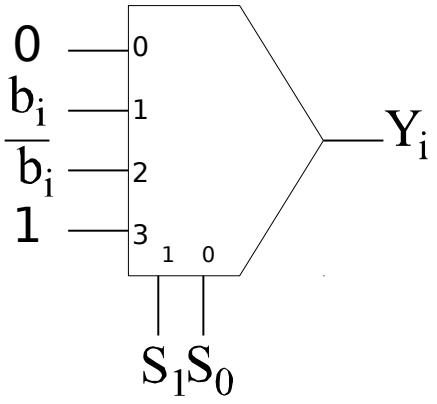




Unidad aritmético-lógica (ALU)

- Diseño del “circuito yi“ con un multiplexor

$S_1 S_0$	Y_i
0 0	0
0 1	b_i
1 0	$\overline{b_i}$
1 1	1



Unidad aritmético-lógica (ALU)

- El acarreo de salida nos puede dar una información muy importante

S_1	S_0	C_{in}		Operación	C_{out} =1 si	Comentario
0	0	0	F=A	Transferir A	--------	C_{out} =0 siempre
0	0	1	F=A+1	Incrementar A	$A=2^n-1$	Si C_{out} =1 , F=0
0	1	0	F=A+B	Sumar A+B	$A+B\geq 2^n$	Overflow si C_{out} =1
0	1	1	F=A+B+1	Incrementar A+B	$A+B\geq 2^n-1$	Overflow si C_{out} =1
1	0	0	F=A+B’	Restar A-B en Ca1	A>B	Si C_{out} =0 → A≤B y F=Ca1(B-A)
1	0	1	F=A+B’+1	Restar A-B en Ca2	A≥B	Si C_{out} =0 → A<B y F=Ca1(B-A)
1	1	0	F=A-1	Decrementar A	A≠0	Si C_{out} =0 → A=0
1	1	1	F=A	Transferir A	-------	C_{out} =1 siempre

Unidad aritmético-lógica (ALU)

- Diseño de la Unidad Lógica
  - Diseño de la etapa típica con un multiplexor y puertas lógicas

S_2	S_1	S_0	Función ALU
1	0	0	F = A AND B
1	0	1	F = A OR B
1	1	0	F = NOT A
1	1	1	F = A XOR B