

Circuitos combinacionales aritméticos (Parte II)

© Luis Entrena, Celia López,
Mario García, Enrique San Millán
Universidad Carlos III de Madrid

Contenidos

1. Circuitos sumadores y restadores

- **Sumadores con propagación de acarreo serie**
 - Semisumador. Sumador total. Sumador de n bits con acarreo serie
- **Sumadores con acarreo anticipado**
- **Sumador/Restador en complemento a 2**

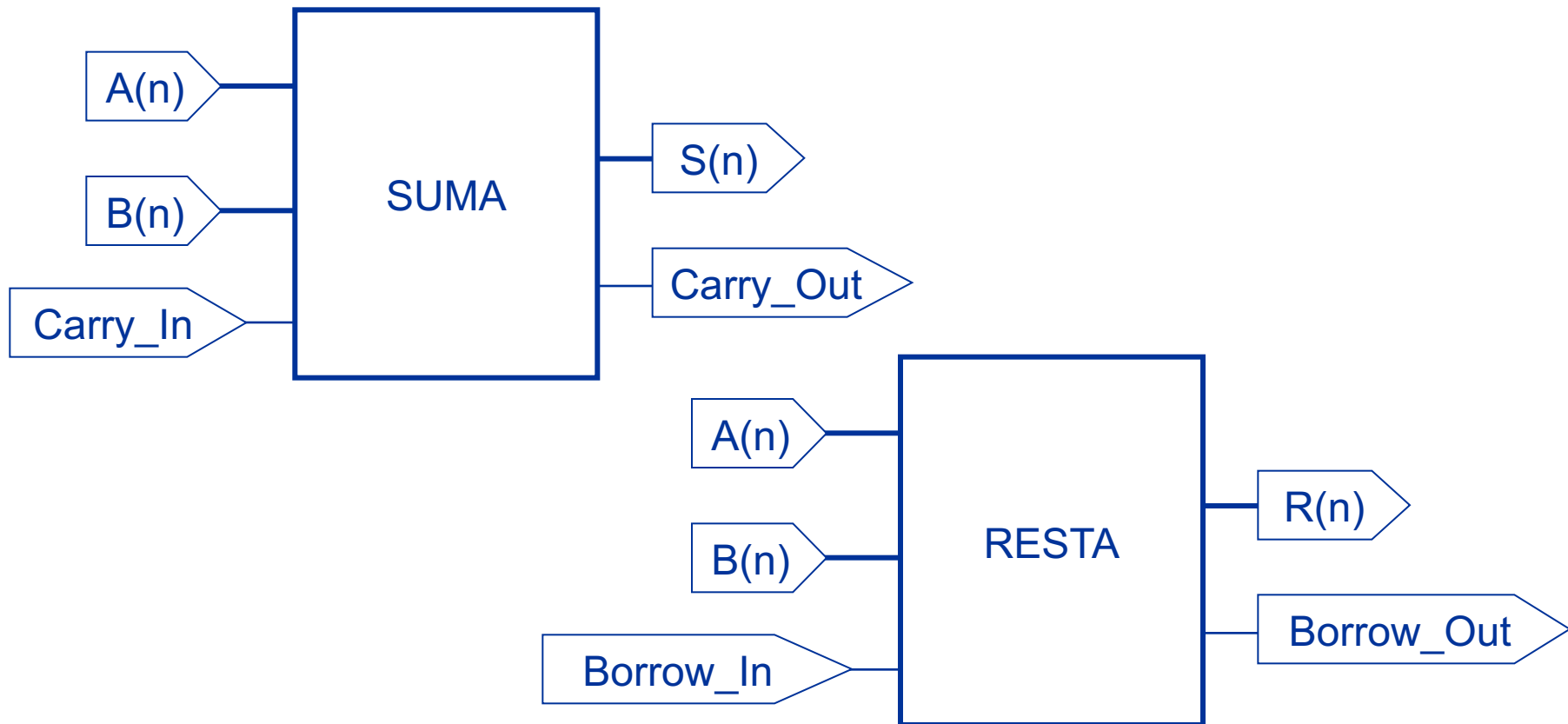
2. Circuitos de multiplicación

- Circuito multiplicador básico

3. Unidades Aritmético-Lógicas (ALUs)

- Concepto de ALU

Circuitos sumadores y restadores

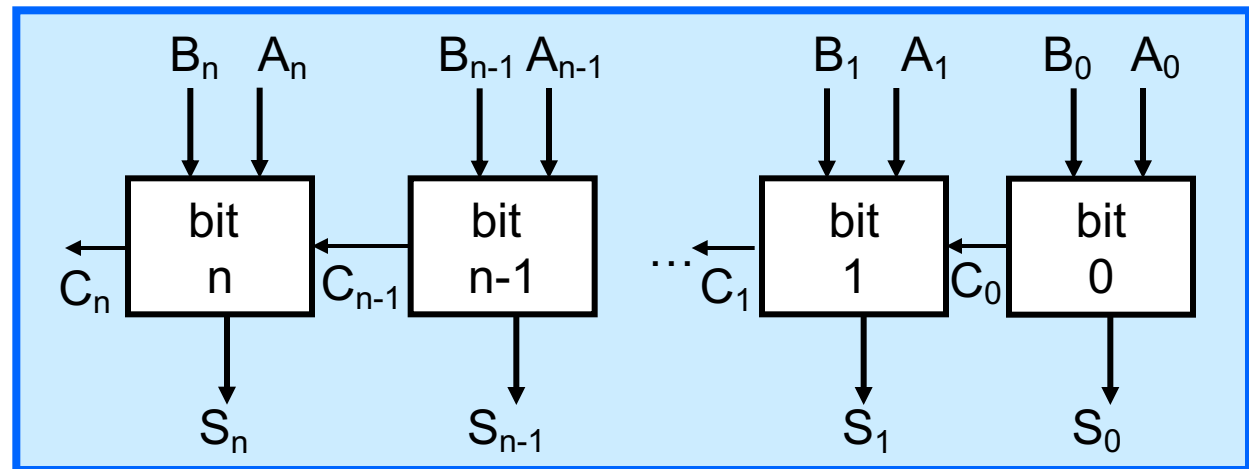


Sumador con propagación de acarreo serie.

Suma decimal y binaria

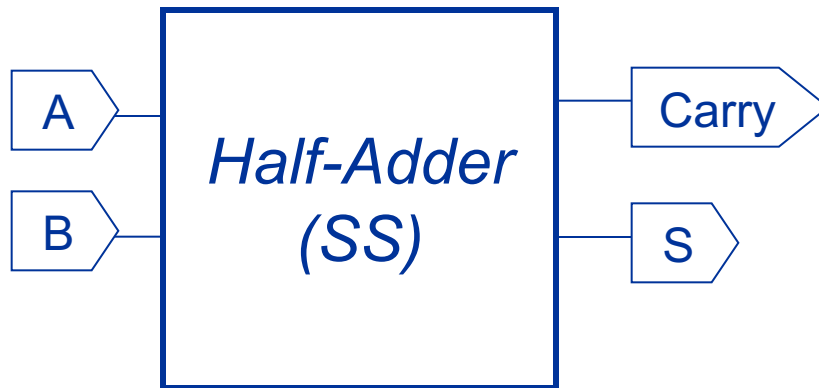
| | | |
|-------|---|-----------|
| ① ① | | |
| 86d | → | 1010110b |
| 25d | → | 0011001b |
| <hr/> | | |
| 111d | → | 11011111b |

- Operandos: **n bits**
- Resultado: **n+1 bits**

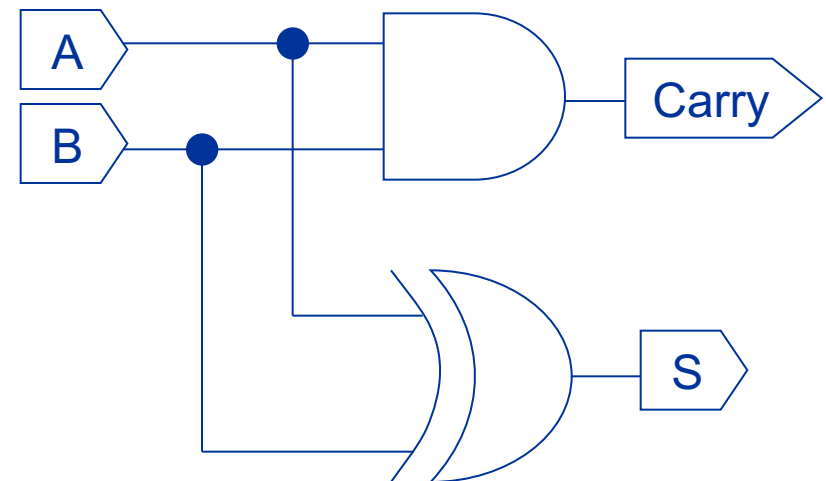


Sumador con propagación de acarreo serie.

Semisumador

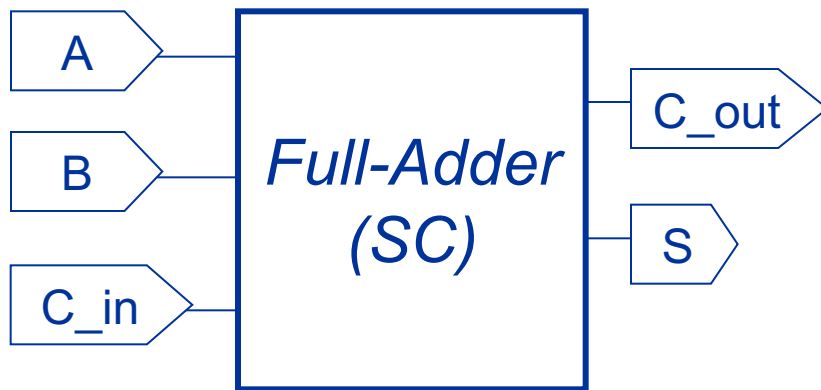


| A | B | S | Carry |
|---|---|---|-------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

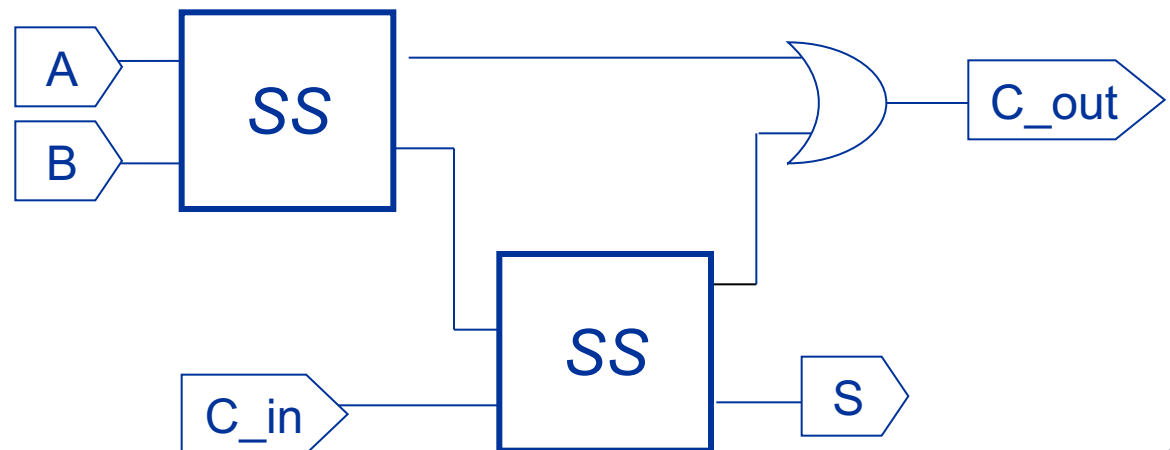


Sumador con propagación de acarreo serie.

Sumador completo

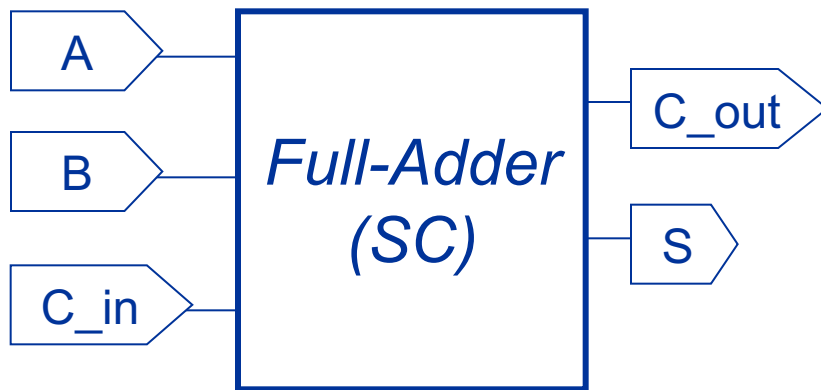


| A | B | C_in | S | C_out |
|---|---|------|---|-------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |



Sumador con propagación de acarreo serie.

Sumador completo



| A | B | C _{in} | S | C _{out} |
|---|---|-----------------|---|------------------|
| 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 |

S

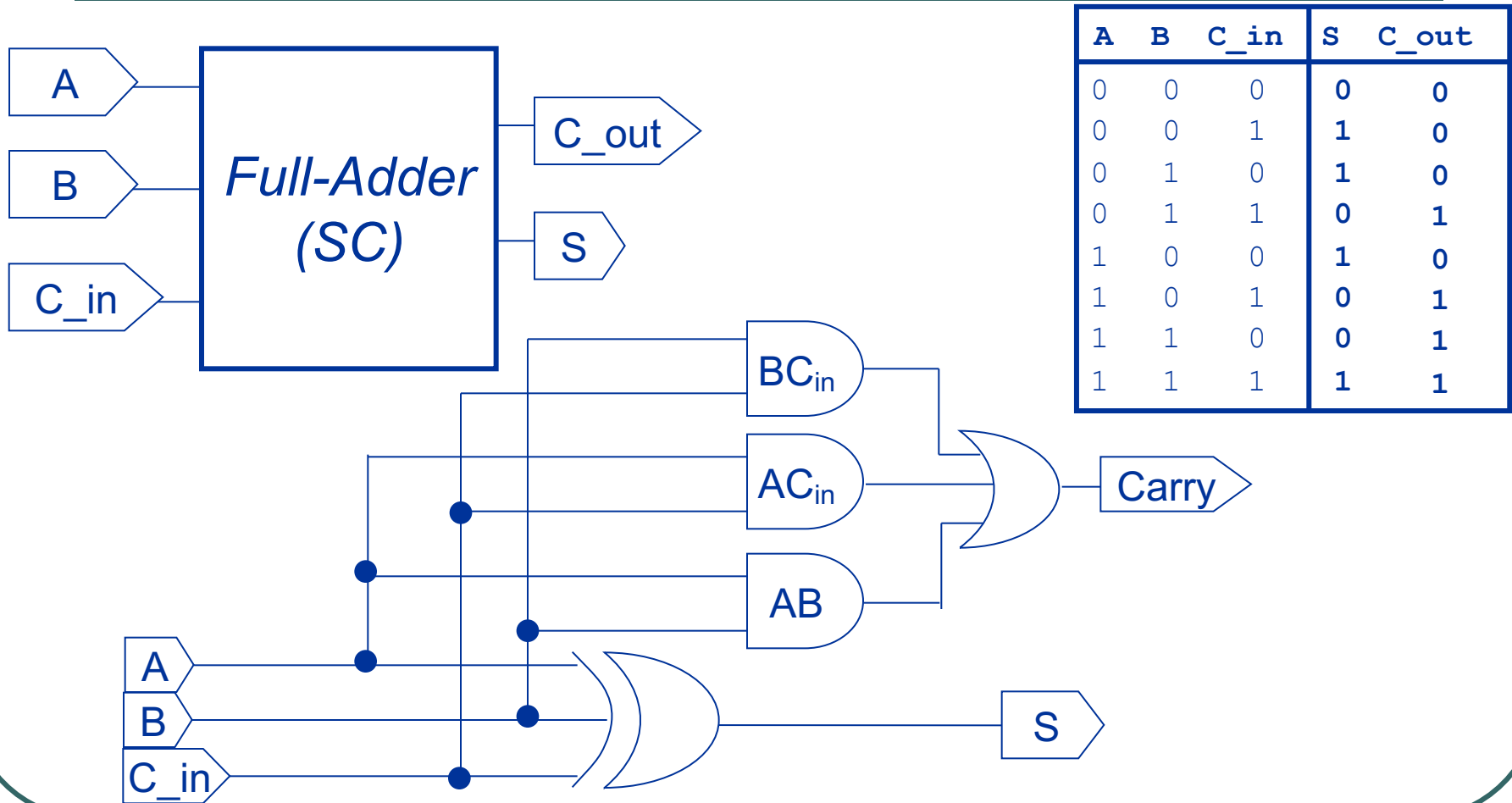
| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |

C_{out}

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 |

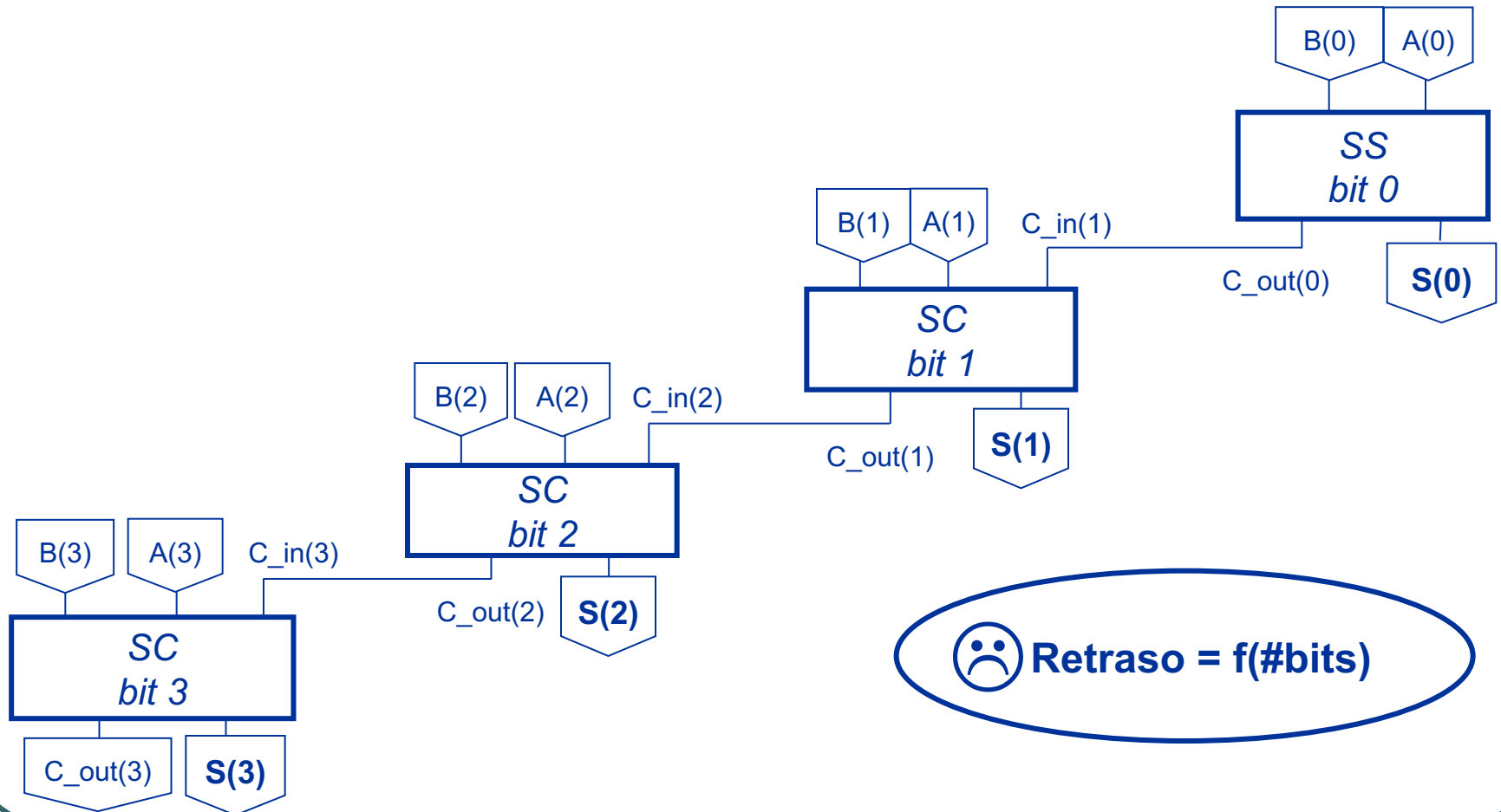
Sumador con propagación de acarreo serie.

Sumador completo



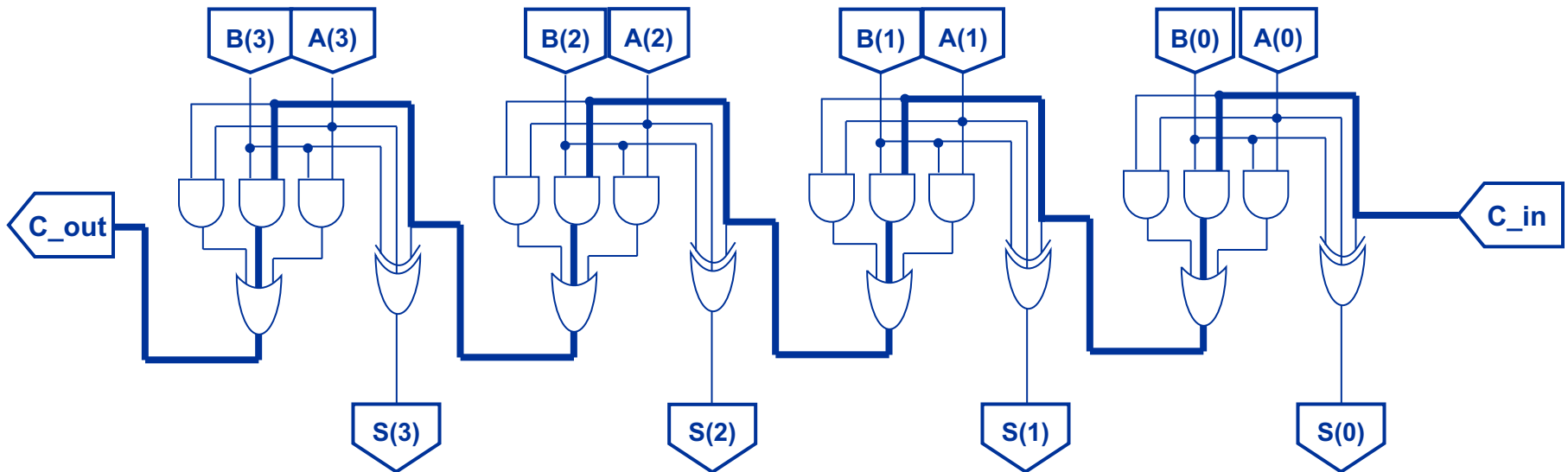
Sumador con propagación de acarreo serie.

Sumador de varios bits

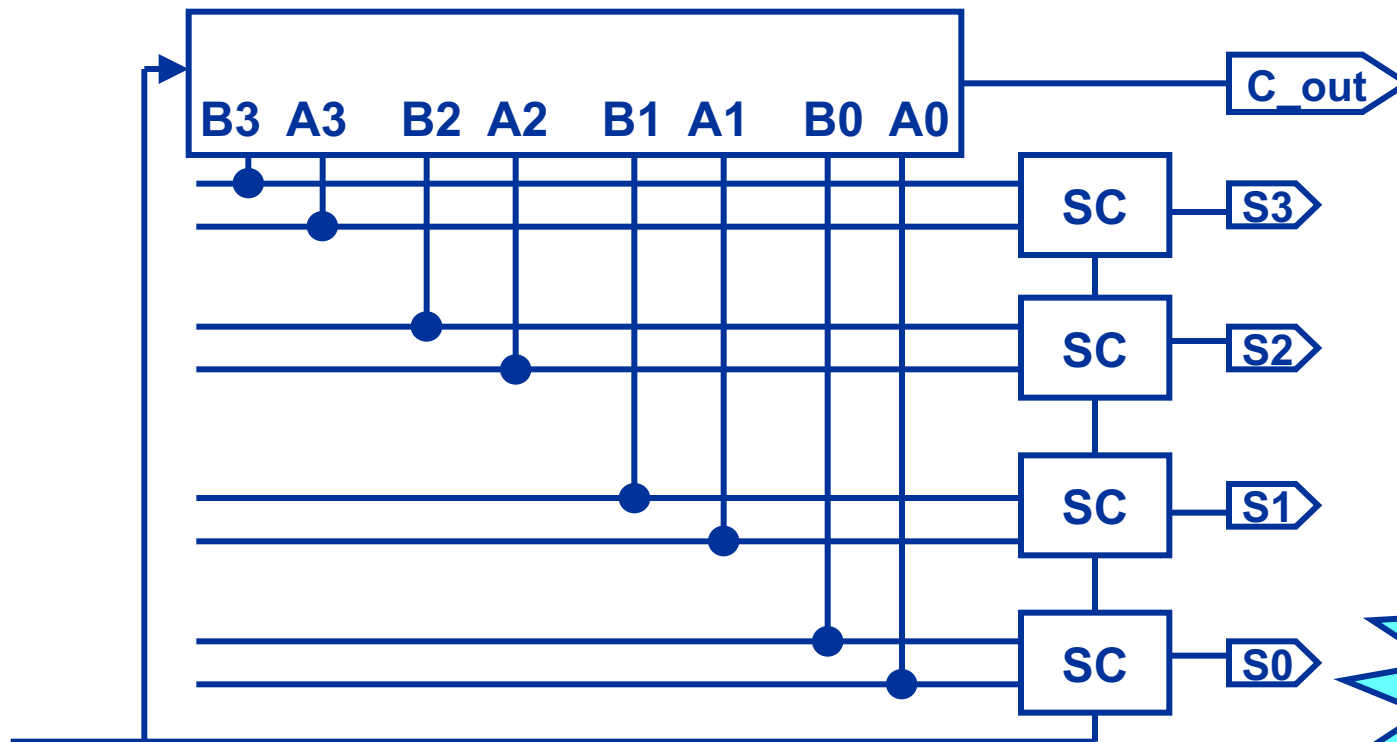


Sumador con propagación de acarreo serie.

Sumador de varios bits

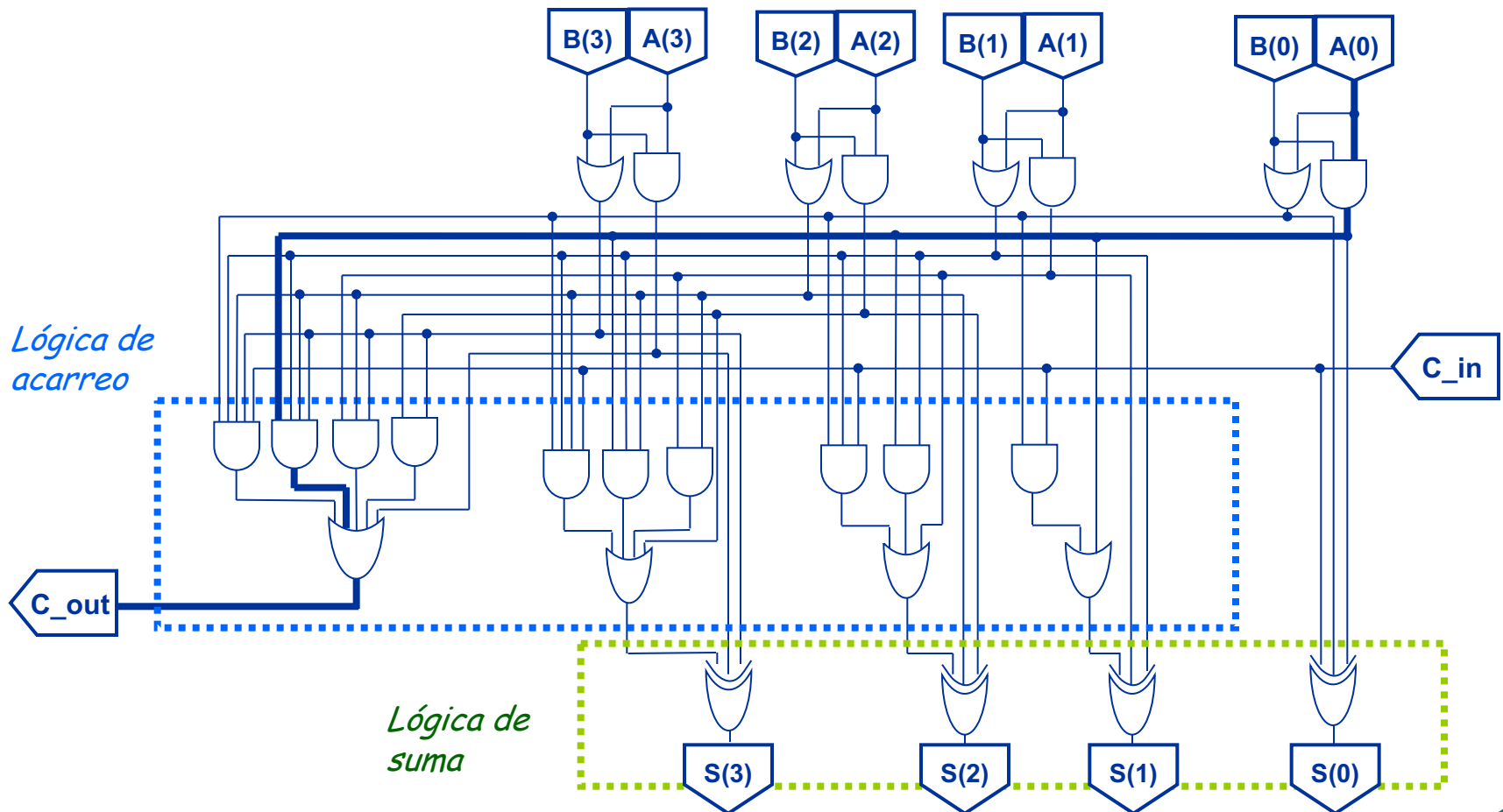


Sumador con acarreo anticipado.



**Carry
look-ahead**

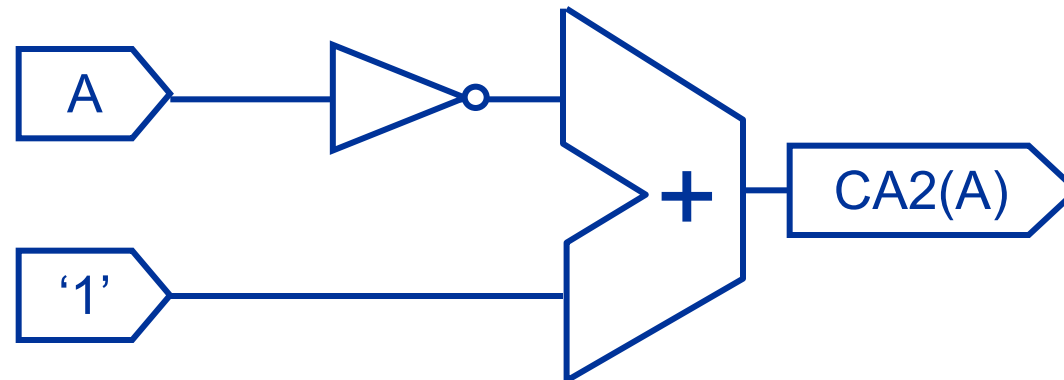
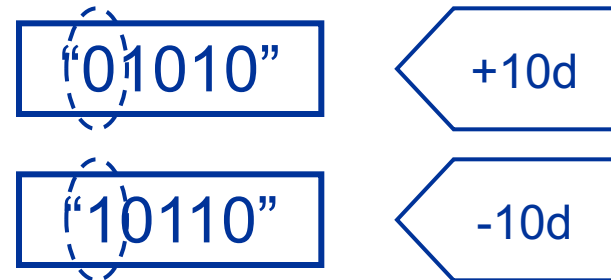
Sumador con acarreo anticipado.



Sumador/restador en CA2.

Complemento a 2

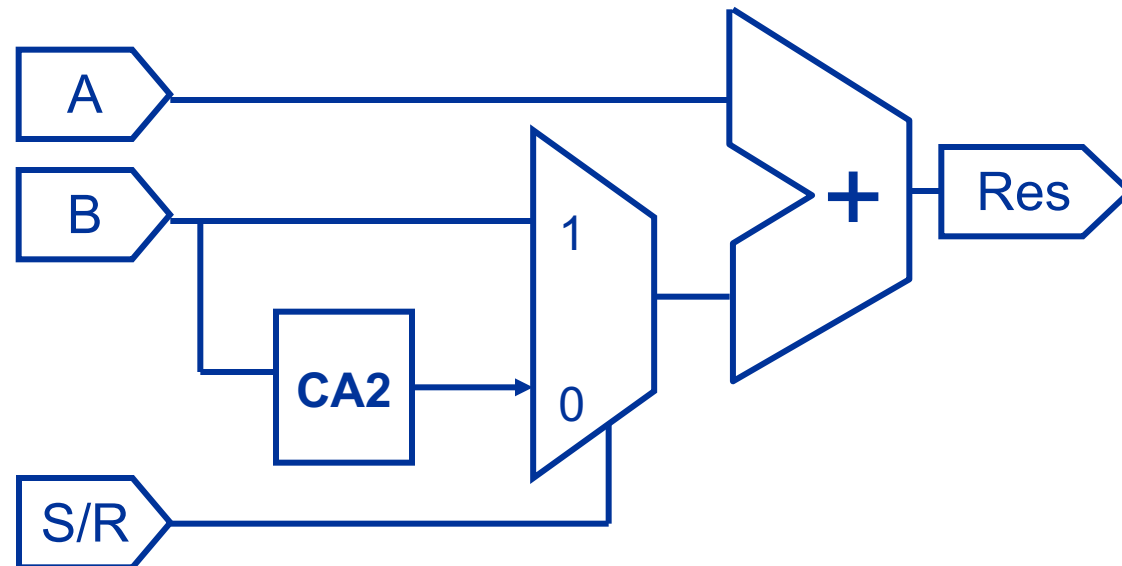
- Números positivos
- Números negativos



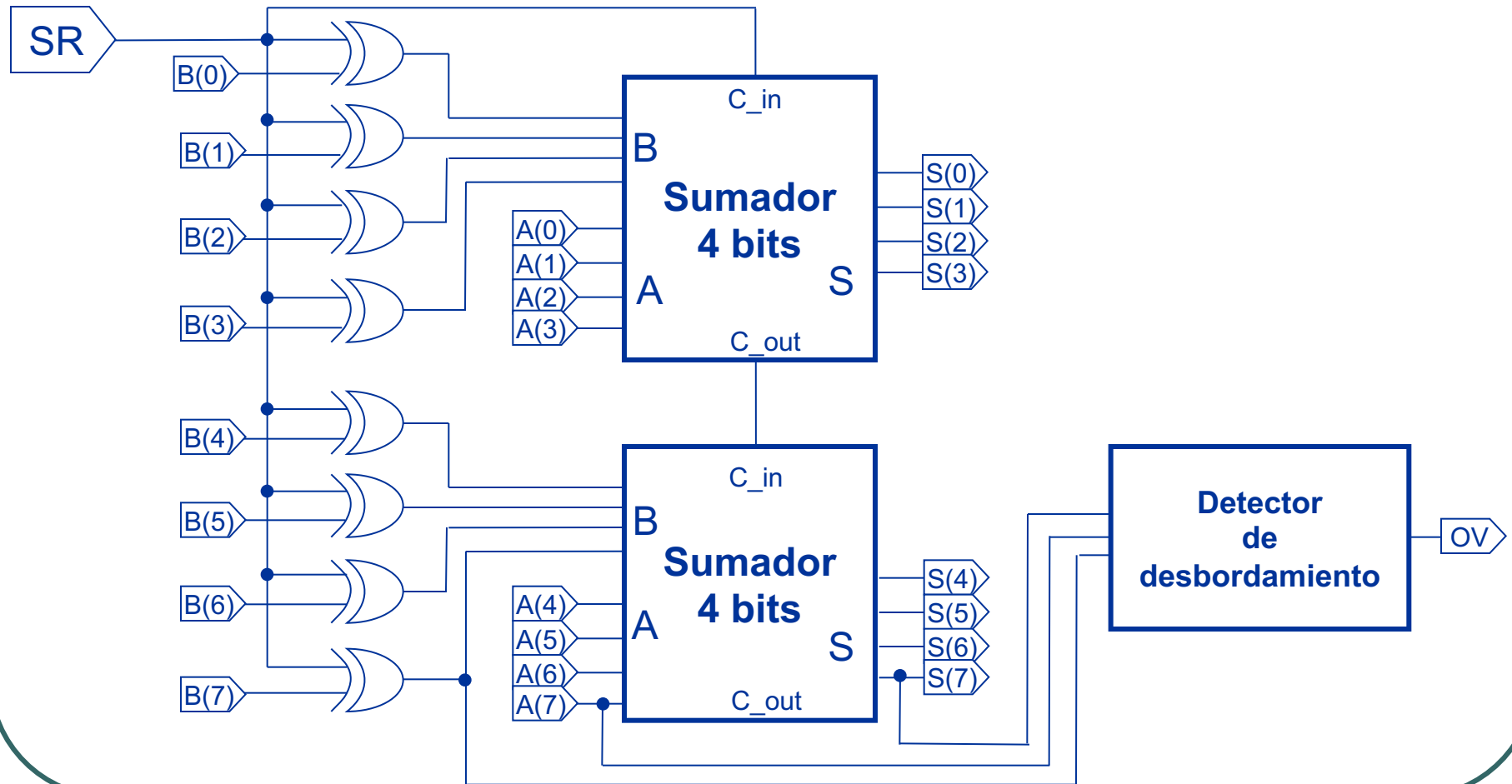
Sumador/restador en CA2.



$$A - B = A + (-B)$$



Sumador/restador en CA2.

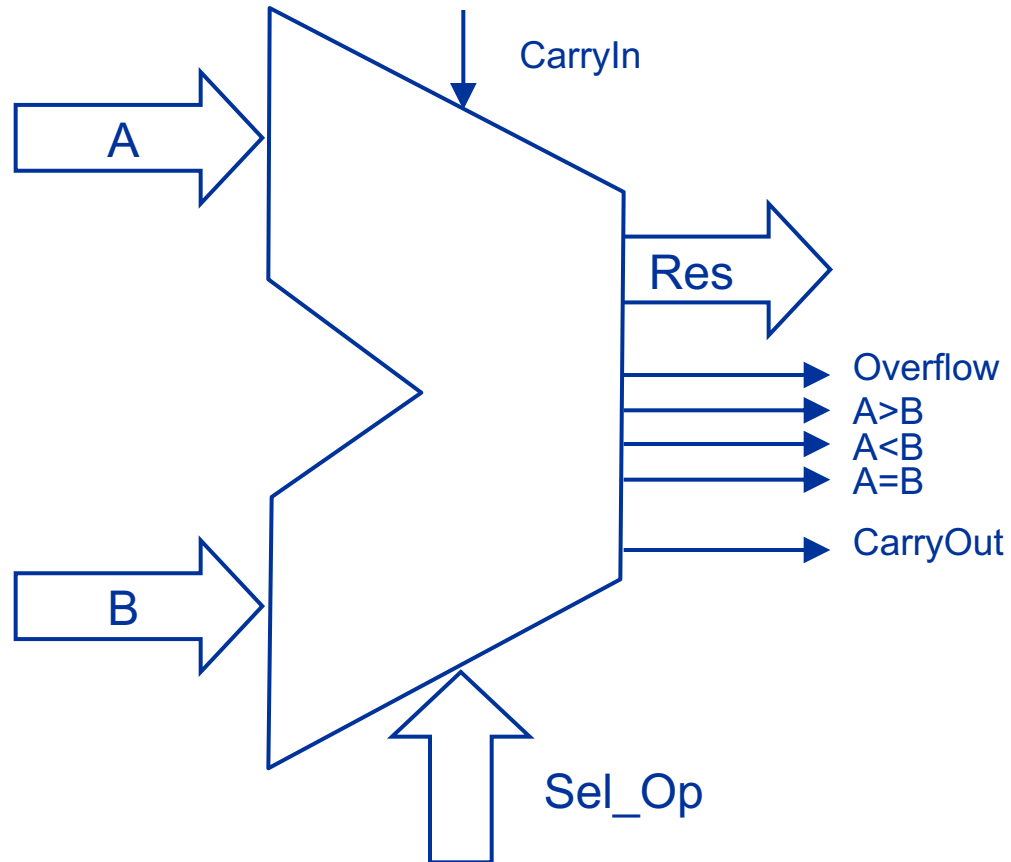


Sumador/restador en CA2.

Ejercicio



Unidad Aritmético-Lógica



Unidad Aritmético-Lógica

Combinacional

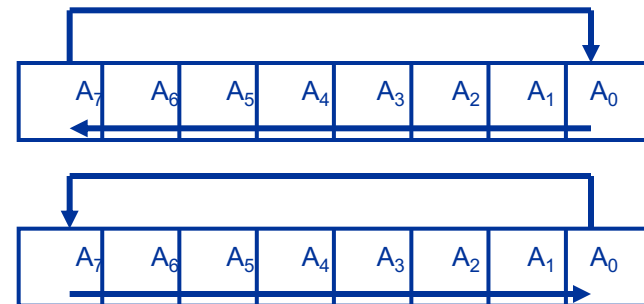
Bloque para la realización de operaciones aritmético-lógicas:

- Suma
 $A + B$
- Resta
 $A - B$
- Complemento a 2
 $-B$
- Comparación
 $A > B$
 $A < B$
 $A = B$
- Desplazamiento a la izquierda
 $SHL(A) \leftarrow$
- Desplazamiento a la derecha
 $SHR(A) \rightarrow$

RESTA

Operaciones lógicas (bit a bit)

- AND
- OR
- XOR
- XNOR
- NOT



Referencias

- “Circuitos y Sistemas Digitales”. J. E. García Sánchez, D. G. Tomás, M. Martínez Iniesta. Ed. Tebar-Flores
- “Electrónica Digital”, L. Cuesta, E. Gil, F. Remiro, McGraw-Hill
- “Fundamentos de Sistemas Digitales”, T.L Floyd, Prentice-Hall

Extra

Multiplicador.

Multiplicador decimal y binario



| | | |
|-------|---|--------------|
| 86d | → | 1010110b |
| 15d | → | 0001111b |
| <hr/> | | |
| 1290d | → | 10100001010b |

Decimal

86d

15d

30 5x6

40 5x8 desplazado a izqda 1 p.

6 6x1 desplazado a izqda 1 p.

8 8x1 desplazado a izqda 2 p.

1290

- Operandos: **n bits**
- Resultado: **2*n bits**

Multiplicador.

Binario

$$A * B = A * (b_{n-1} * 2^{n-1} + b_{n-2} * 2^{n-2} + \dots + b_1 * 2^1 + b_0 * 2^0)$$

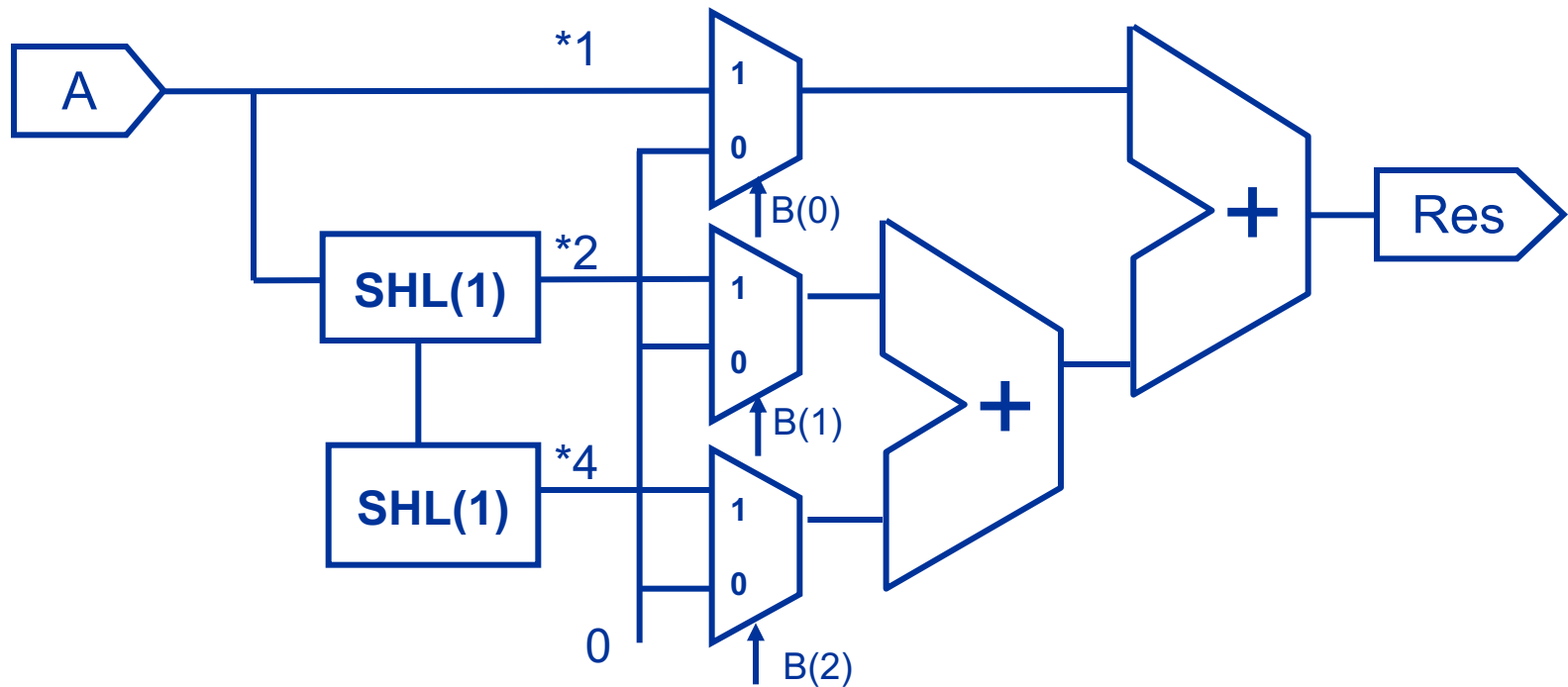
¡¡ '1s' o '0s' !!

La multiplicación binaria de dos números A (m bits) y B (n bits) consiste en una suma de tantos elementos como bits tenga B (n). Cada elemento i es el número A desplazado a la izquierda i veces si el peso correspondiente de B vale '1'. En caso contrario el elemento i es '0'.

Multiplicador.



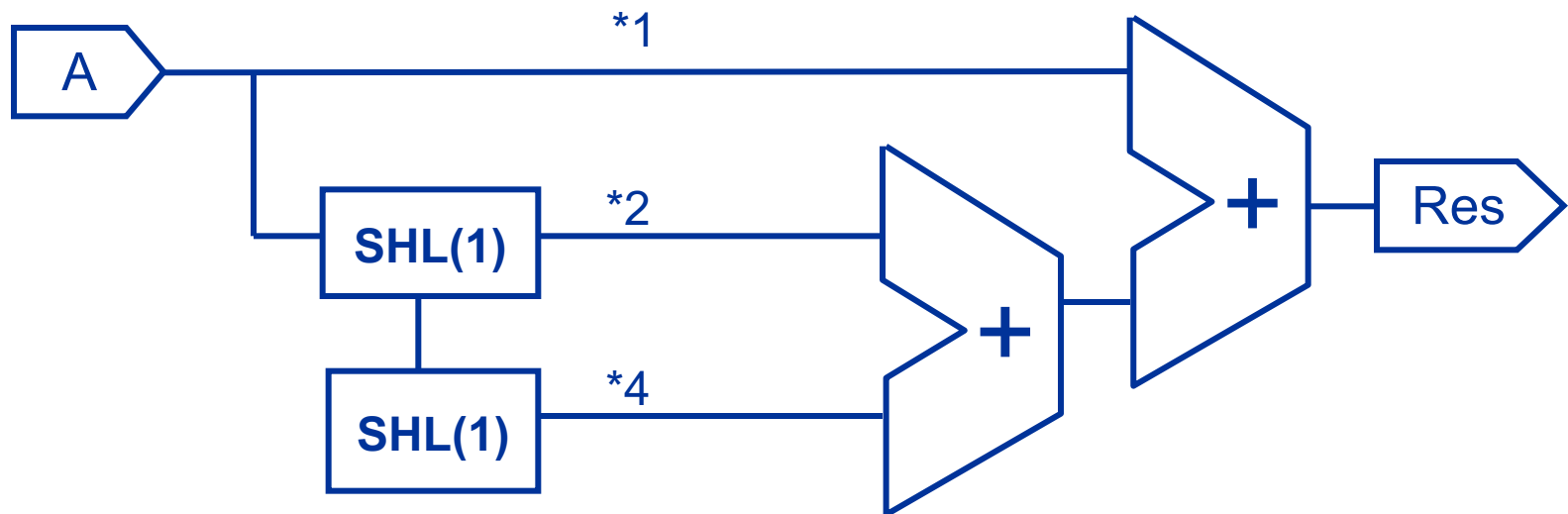
$$A * B = A * (4*B(2)+2*B(1)+1*B(0))$$



Multiplicador.



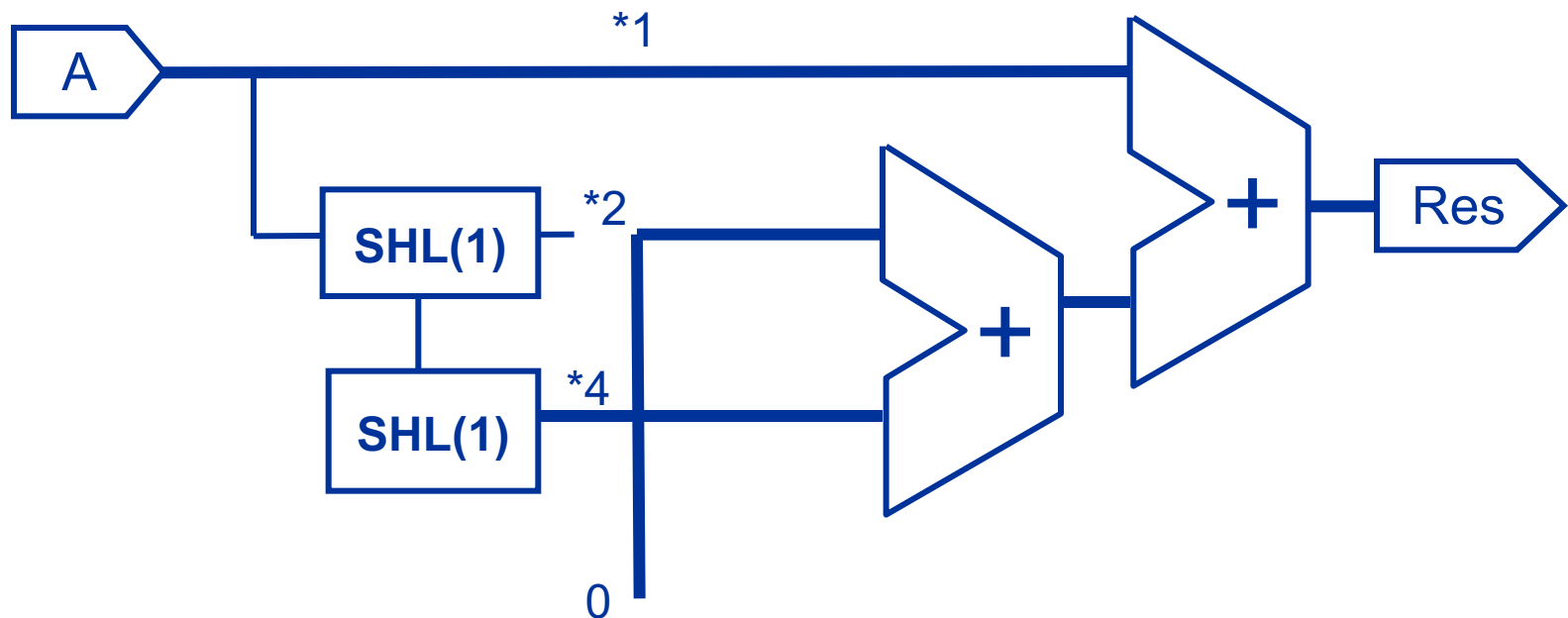
$$A * 7 = A * (4 + 2 + 1)$$



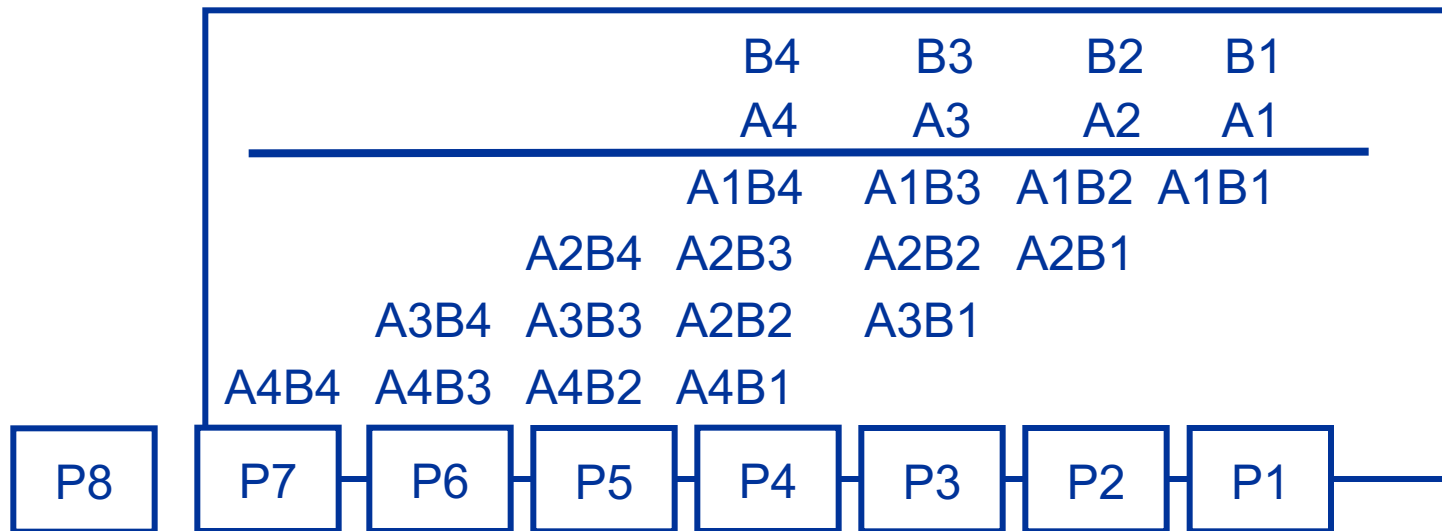
Multiplicador.



$$A * 5 = A * (4 + 0 + 1)$$



Multiplicador.



Multiplicador.

