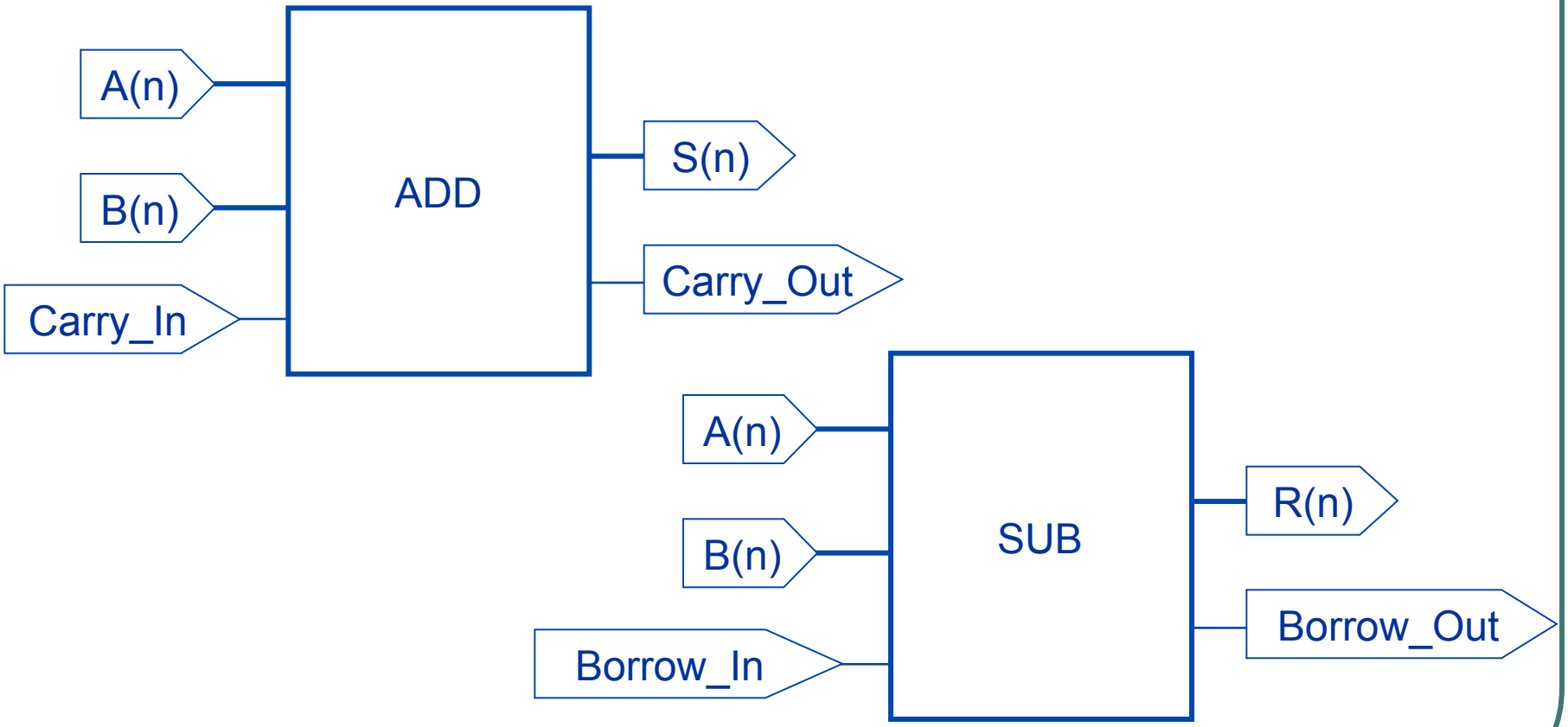
Arithmetic Combinational Circuits

© Luis Entrena, Celia López,
Mario García, Enrique San Millán

Universidad Carlos III de Madrid

Contents

Adders and Subtractors circuits

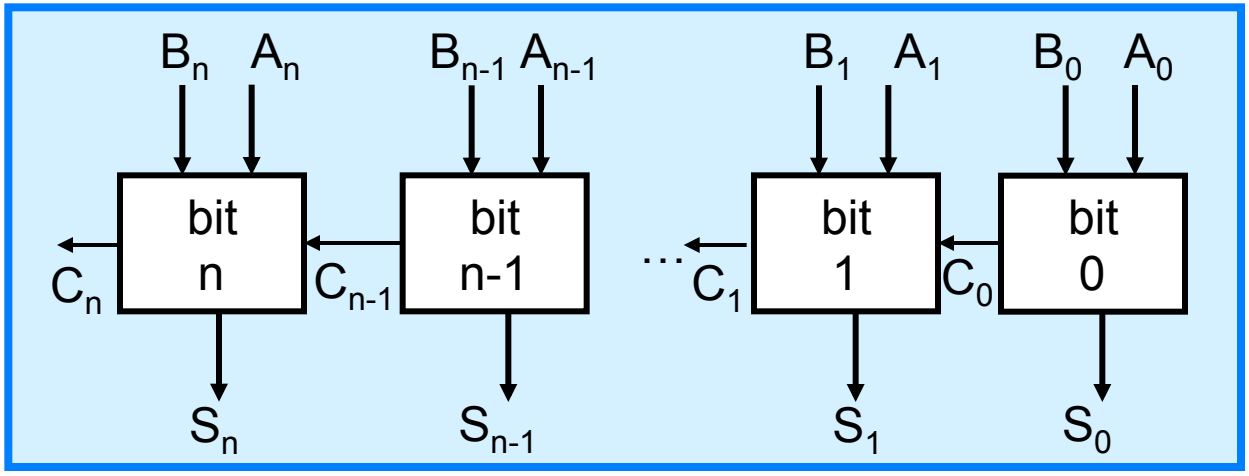


Adder with serial propagation of Carry.

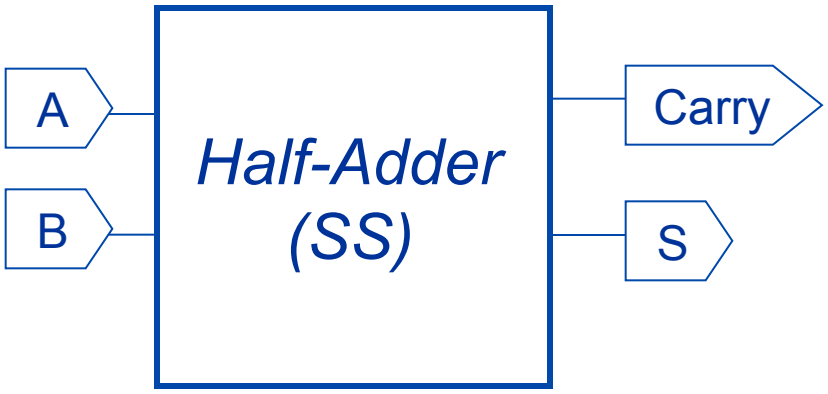
Decimal and binary Addition

1 1		
86d	→	1010110b
25d	→	0011001b
111d	→	11011111b

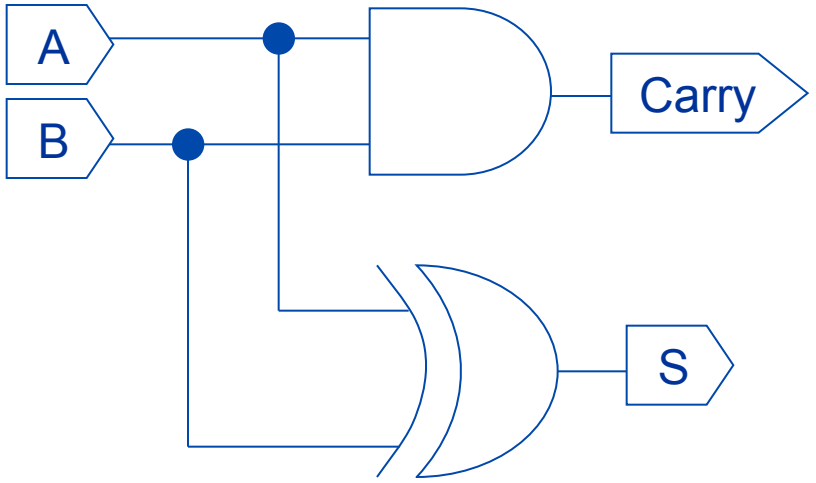
- Operands: **n bits**
- Result: **n+1 bits**



Adder with serial propagation of carry.
Half Adder

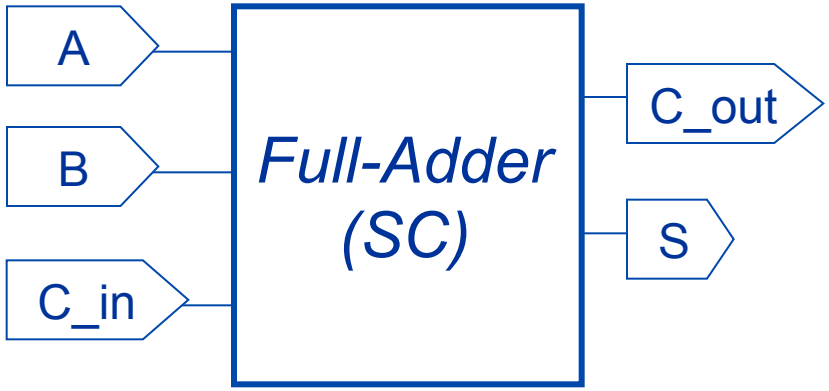


A	B	S	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

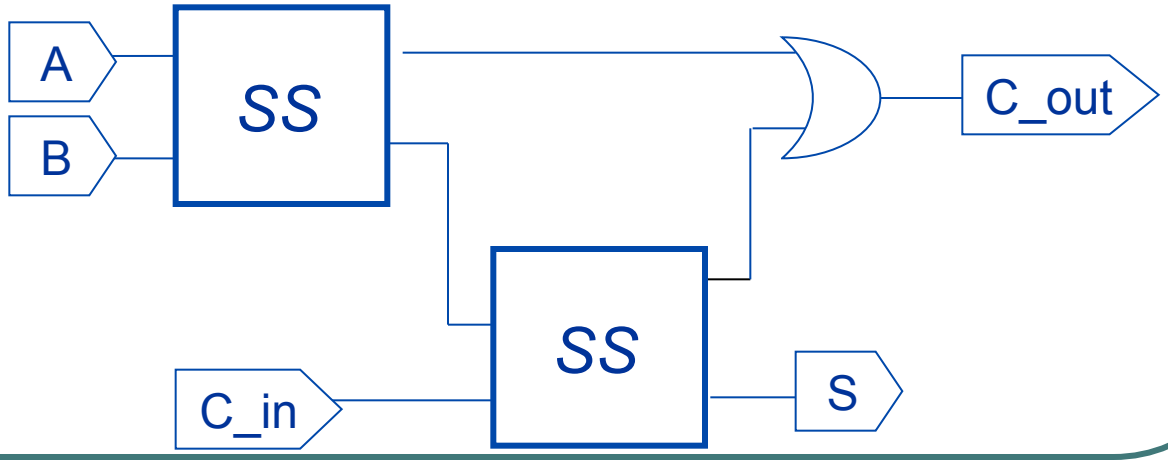


Adder with serial propagation of carry.

Full Adder

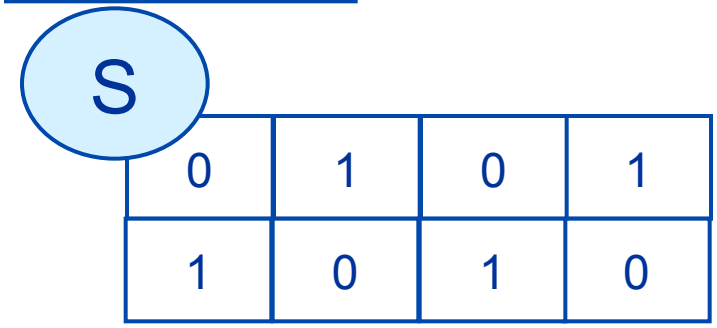


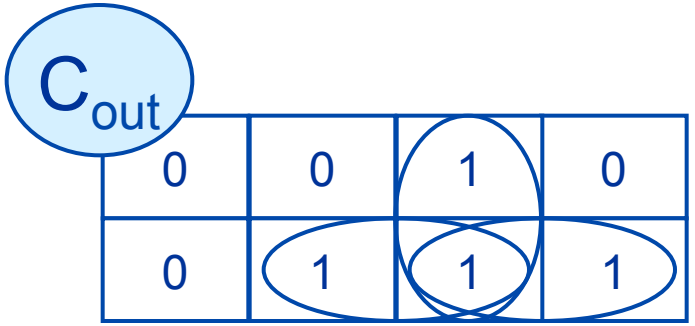
A	B	C_in	S	C_out
0	0	0	**0**	**0**
0	0	1	**1**	**0**
0	1	0	**1**	**0**
0	1	1	**0**	**1**
1	0	0	**1**	**0**
1	0	1	**0**	**1**
1	1	0	**0**	**1**
1	1	1	**1**	**1**



Adder with serial propagation of carry.
Full Adder

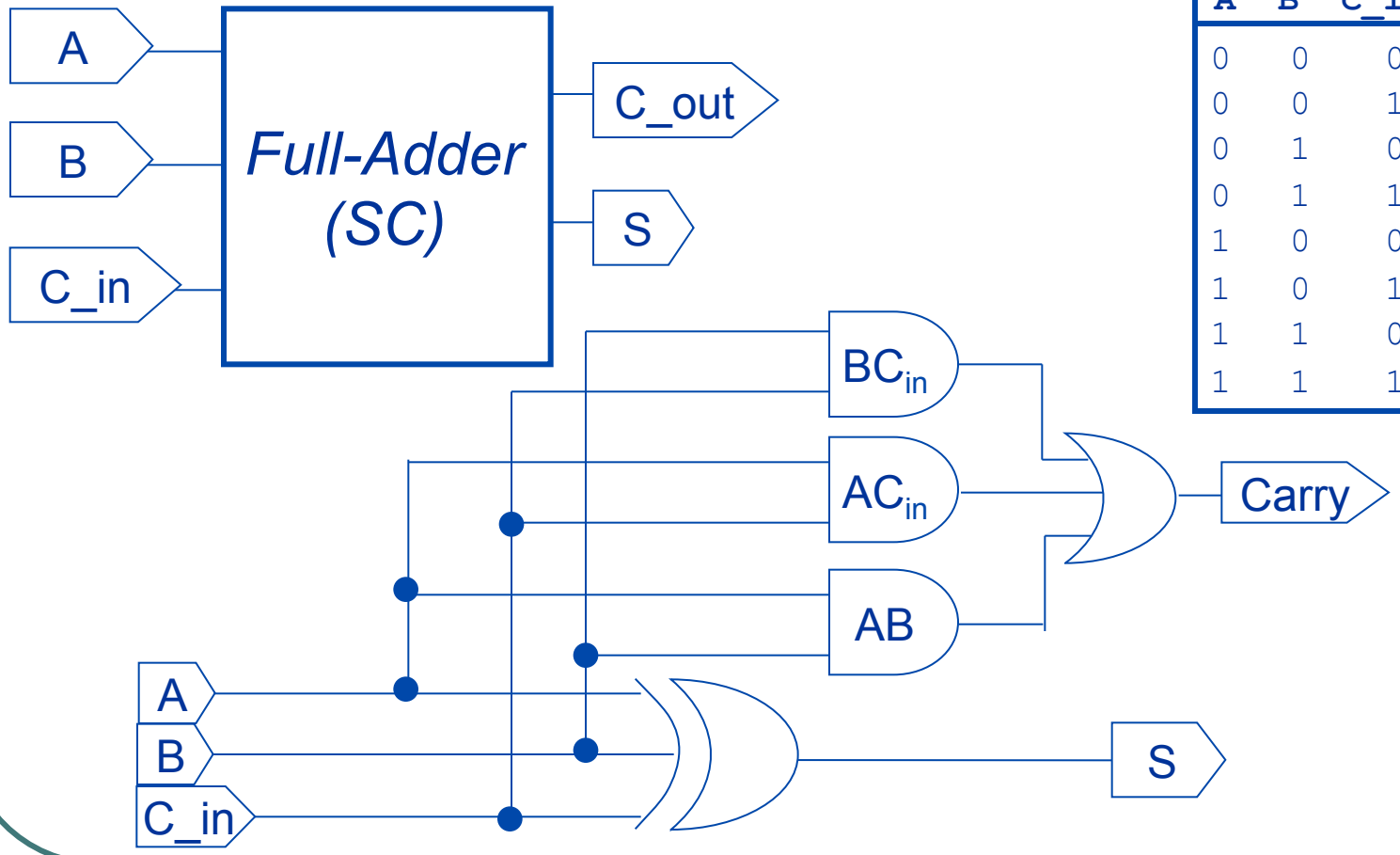
A	B	C_in	S	C_out
0	0	0	**0**	**0**
0	0	1	**1**	**0**
0	1	0	**1**	**0**
0	1	1	**0**	**1**
1	0	0	**1**	**0**
1	0	1	**0**	**1**
1	1	0	**0**	**1**
1	1	1	**1**	**1**





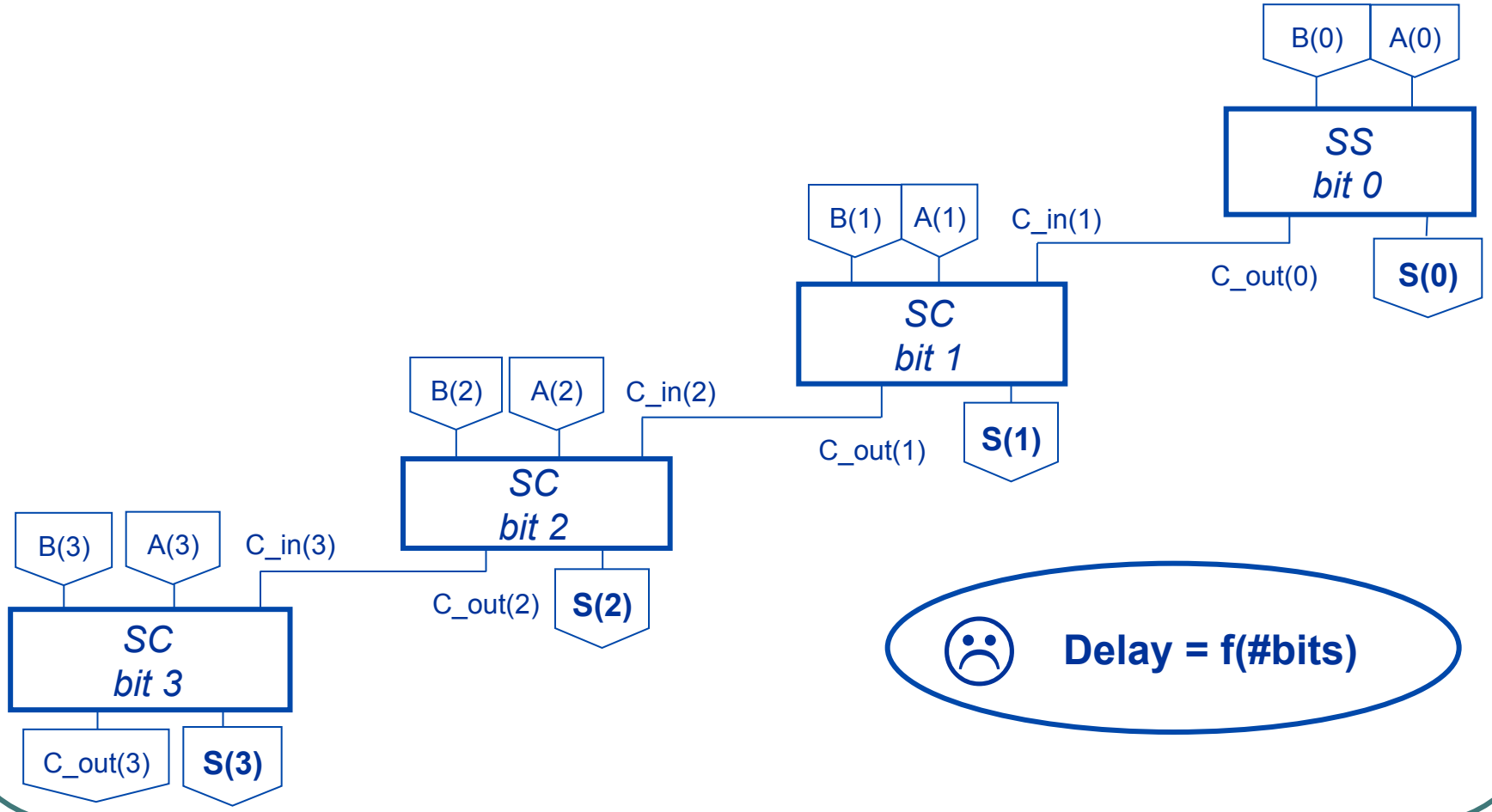
Adder with serial propagation of carry.

Full Adder

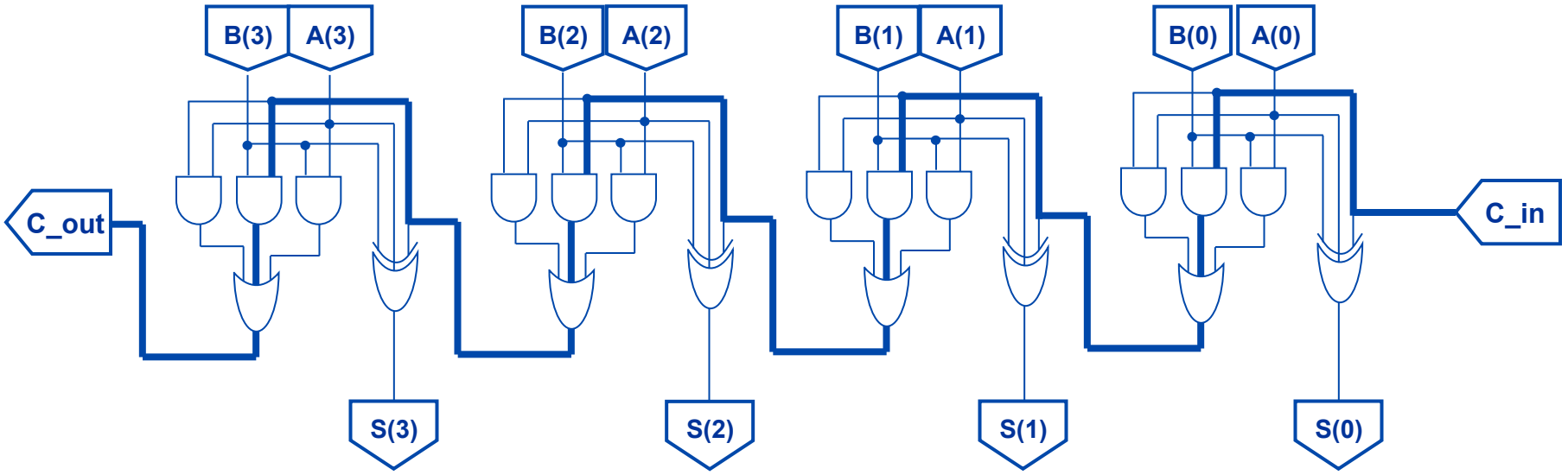


A	B	C_in	S	C_out
0	0	0	**0**	**0**
0	0	1	**1**	**0**
0	1	0	**1**	**0**
0	1	1	**0**	**1**
1	0	0	**1**	**0**
1	0	1	**0**	**1**
1	1	0	**0**	**1**
1	1	1	**1**	**1**

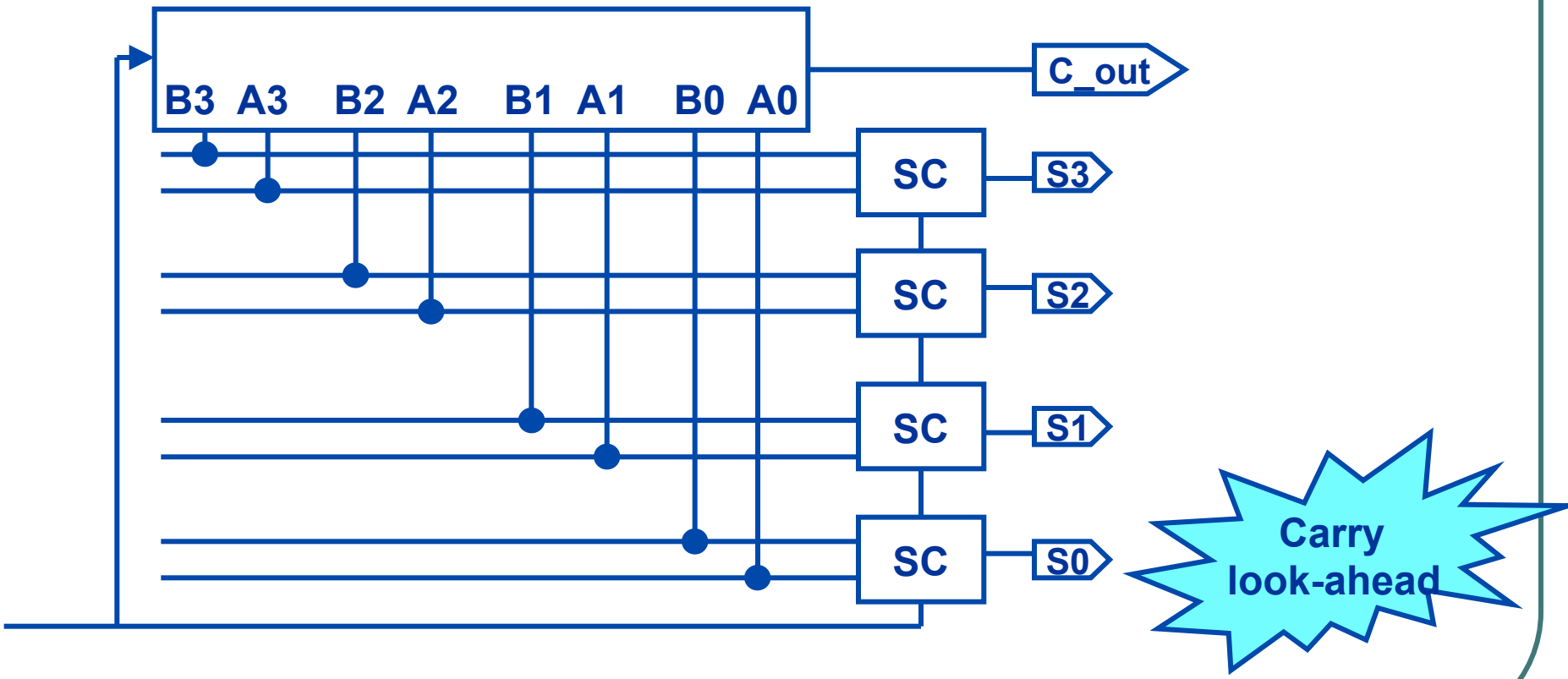
Adder with serial propagation of carry.
N-bits adder



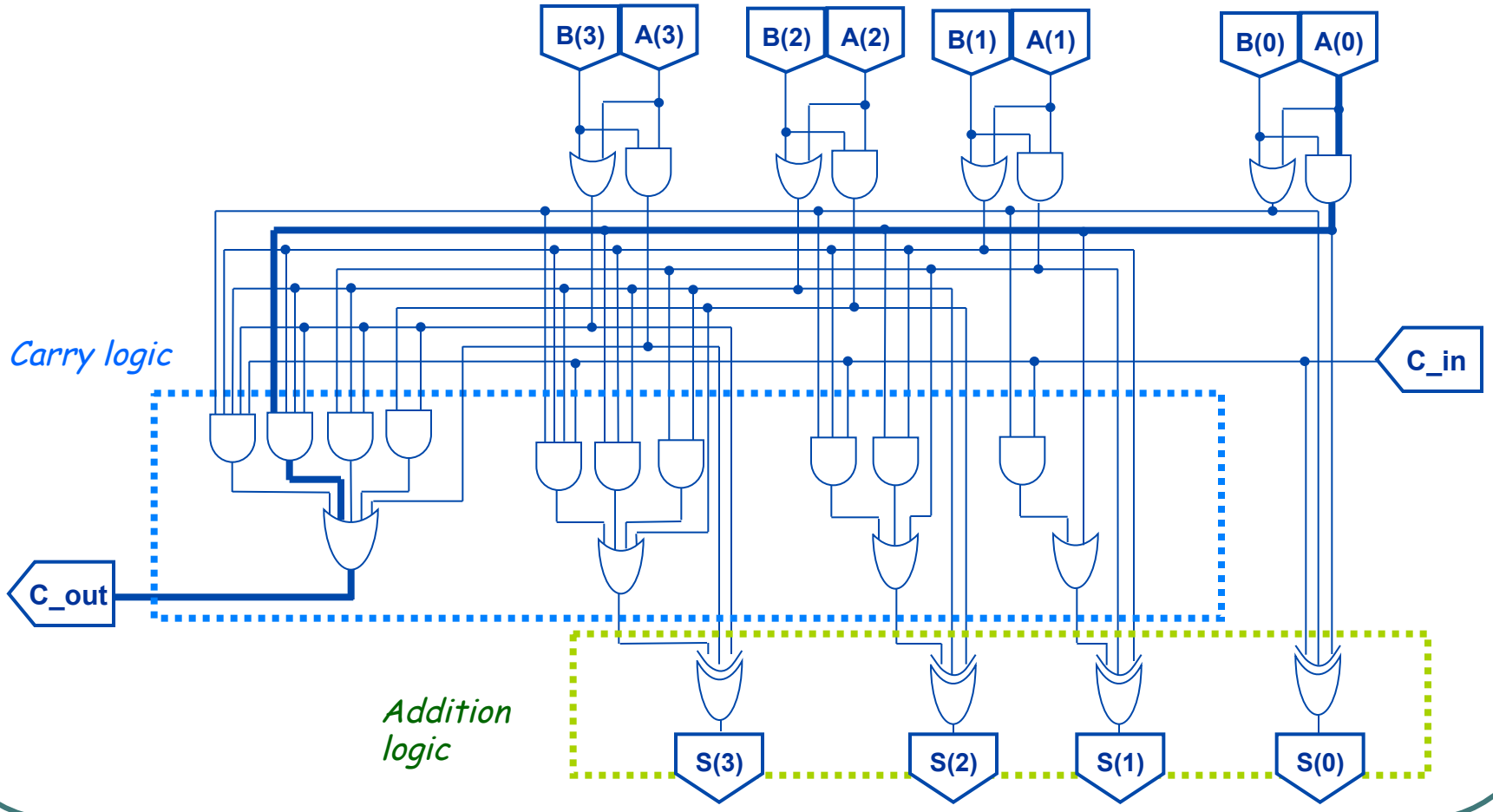
Adder with serial propagation of carry.
N-bits adder



Adder with carry look ahead.



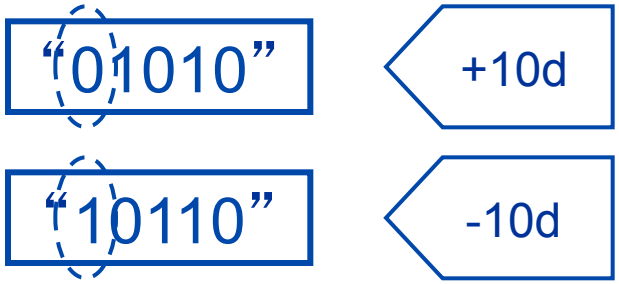
Adder with carry look ahead.

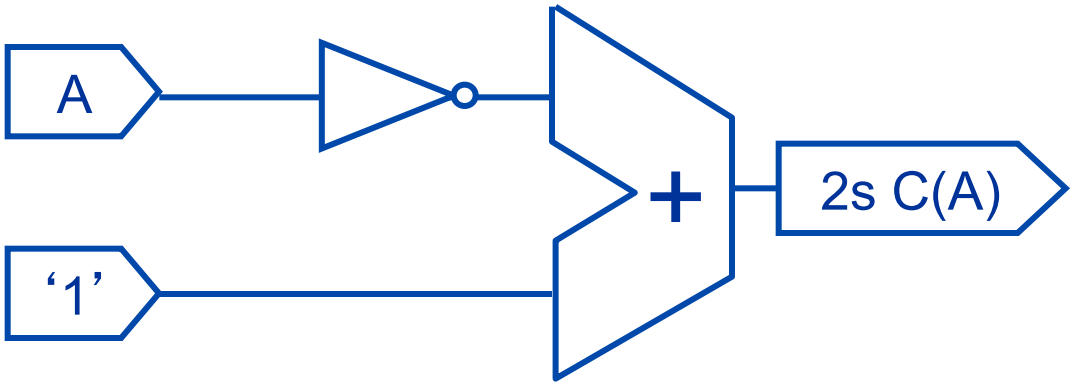


Adder/Subtractor with 2s complement.

2-s complement

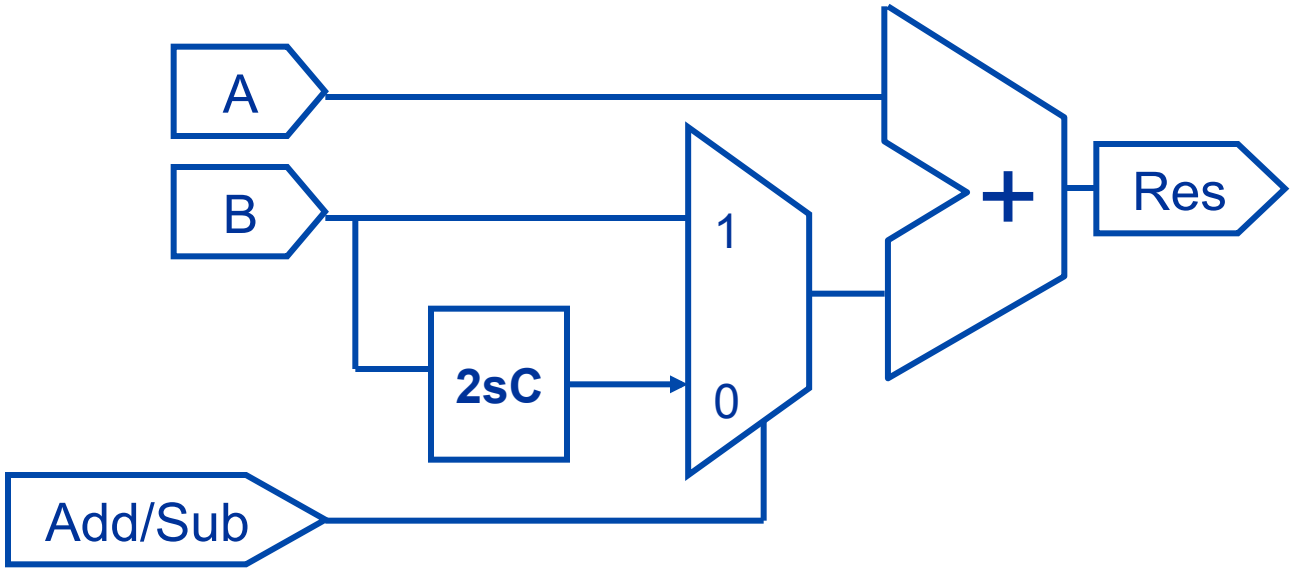
- Positive numbers
- Negative numbers



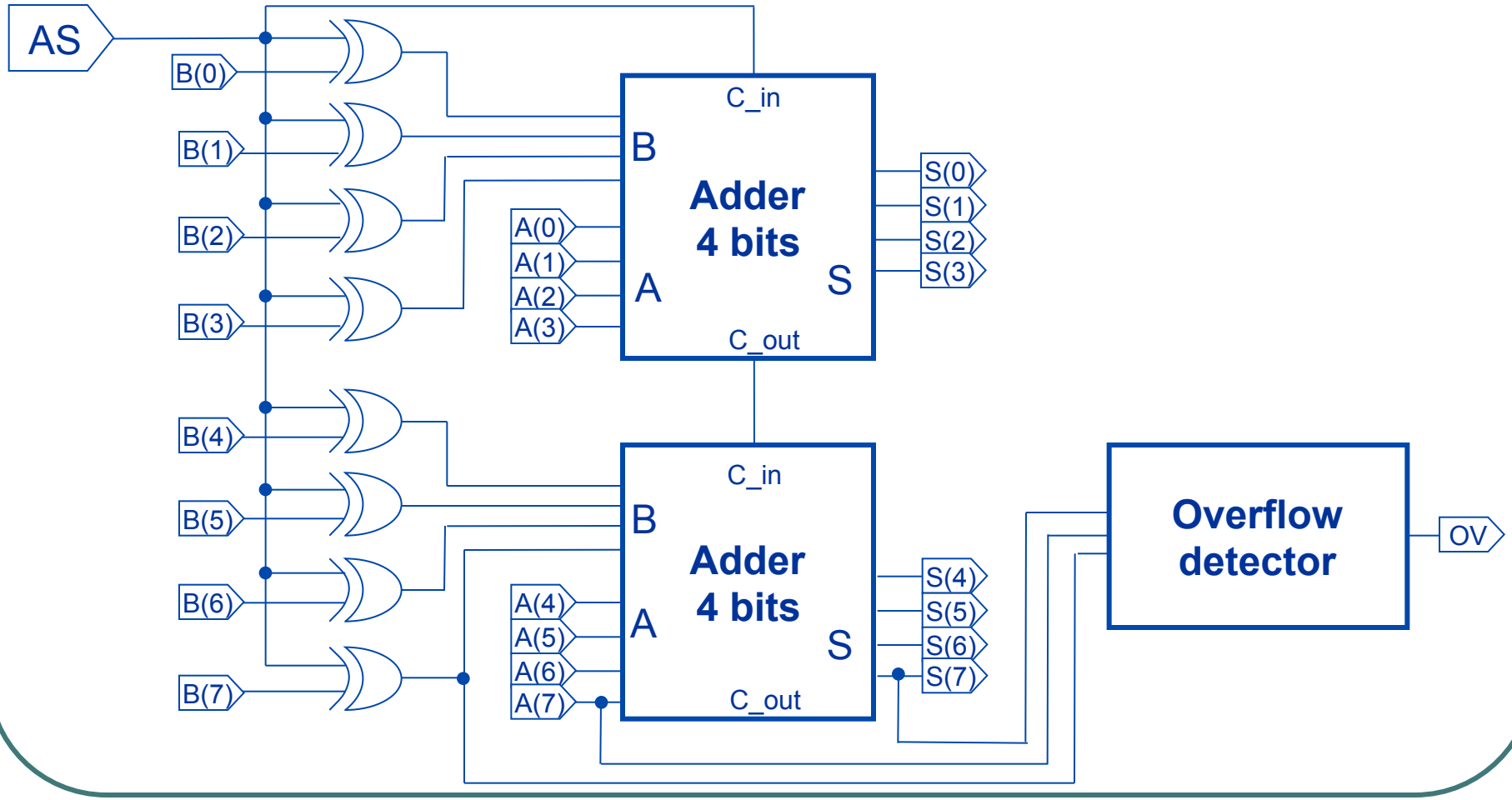


Adder/Subtractor with 2s complement.

★ A – B = A + (-B)



Adder/Subtractor with 2s complement.



Adder/Subtractor with 2s complement.

Exercise

Multiplier.

Decimal and binary multiplier

86d	→	1010110b
15d	→	0001111b
1290d	→	10100001010b

Decimal

86d	
15d	
30	5x6
40	5x8 shifted left 1 position.
6	6x1 shifted left 1 position.
8	8x1 shifted left 2 positions.
1290	

- Operands: **n bits**
- Result: **2*n bits**

Multiplier.

Binary

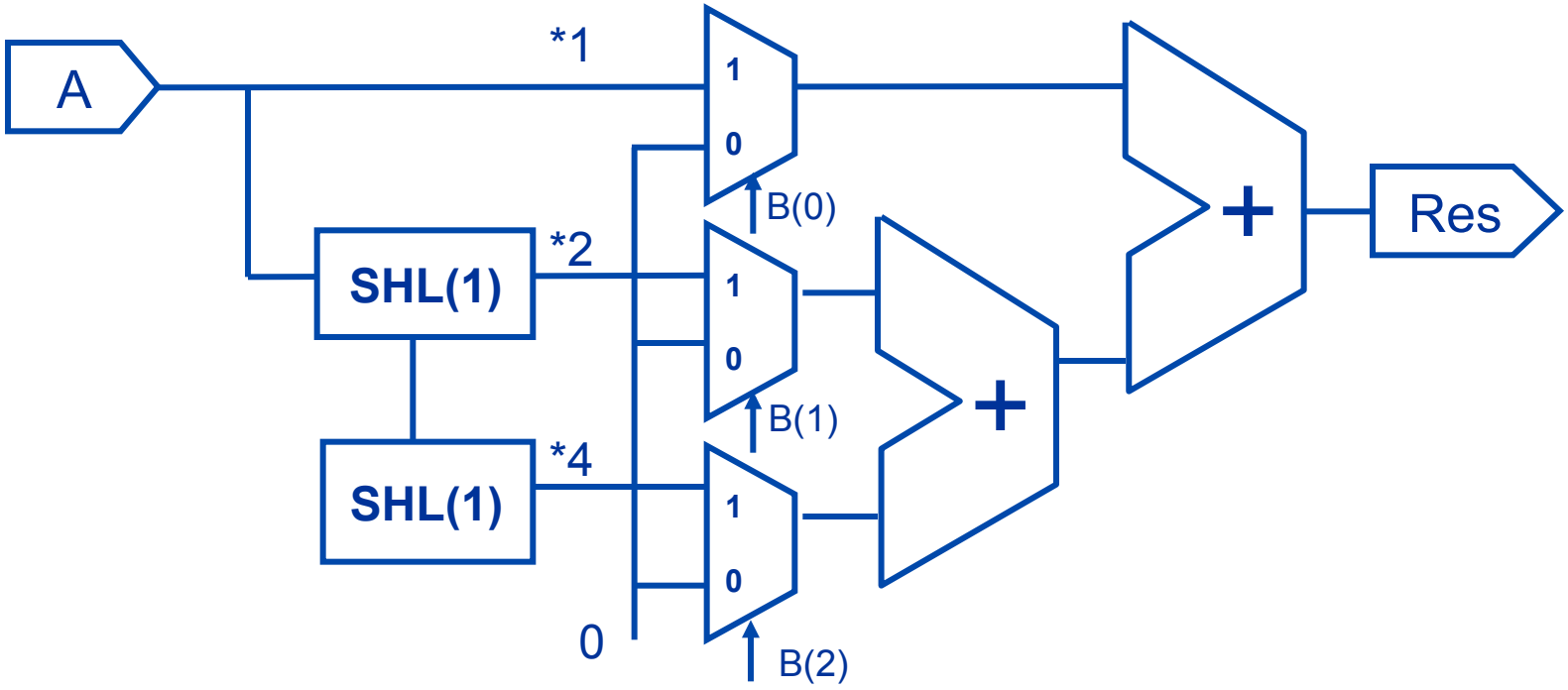
$$A * B = A * (b_{n-1}*2^{n-1} + b_{n-2}*2^{n-2} + \ldots + b_1*2^1 + b_0*2^0$$

¡¡ ’1s’ o ‘0s’ !!

The binary multiplication of two numbers A (m bits) y B (n bits) consists in an adding of as many elements as bits in B (n). Every element i is number A shifted left i-times when the corresponding weight of B is ‘1’.
In other case the element is ‘0’.

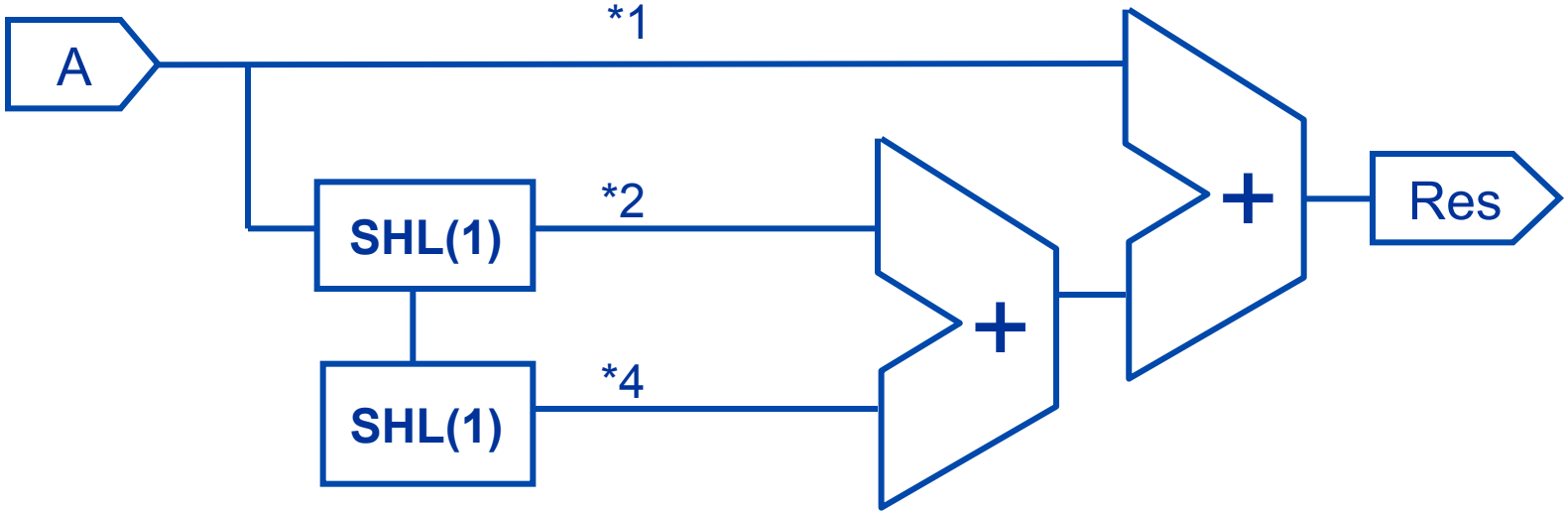
Multiplier.

★ A * B = A * (4*B(2)+2*B(1)+1*B(0))



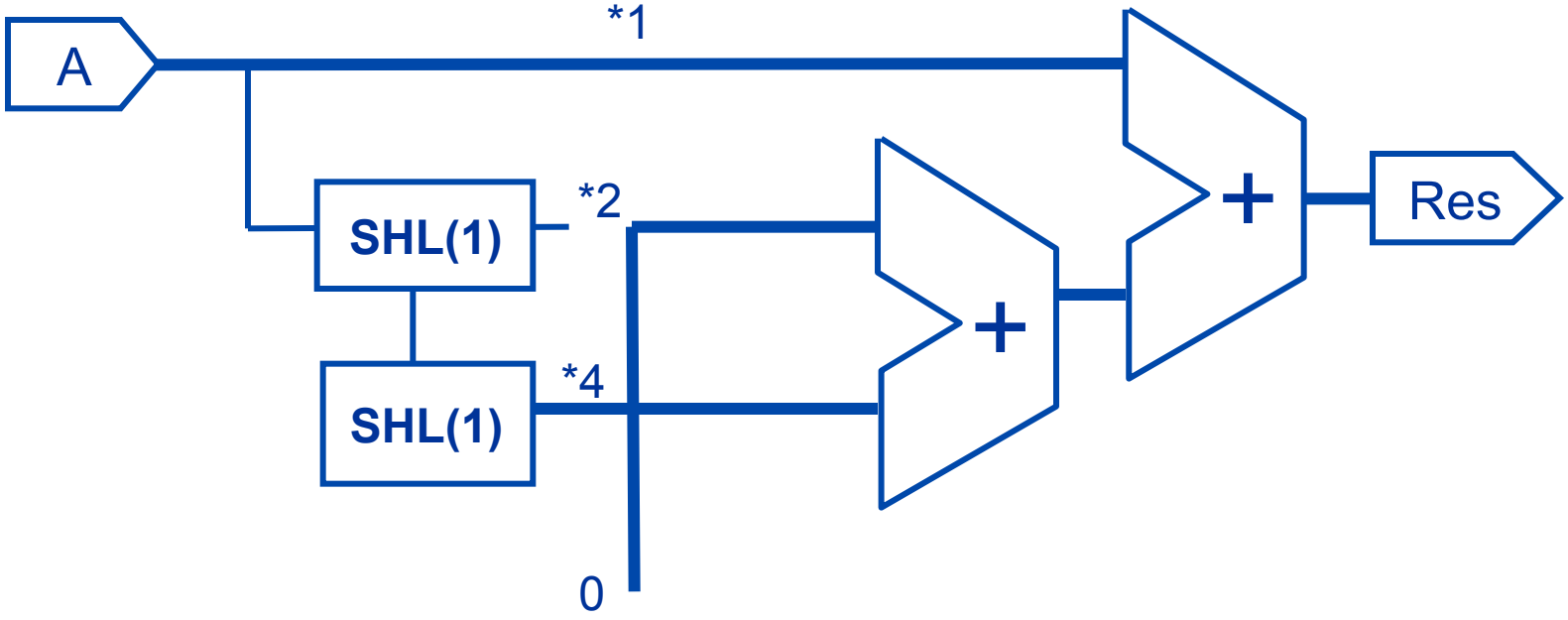
Multiplier.

★ A * 7 = A * (4+2+1)

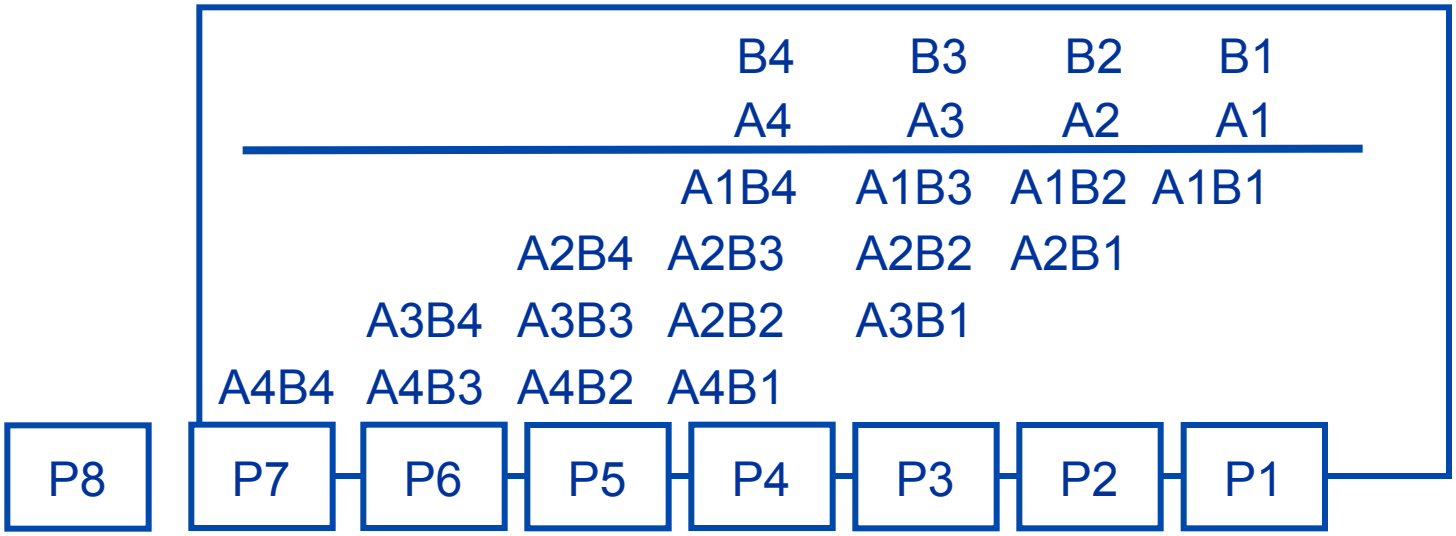


Multiplier.

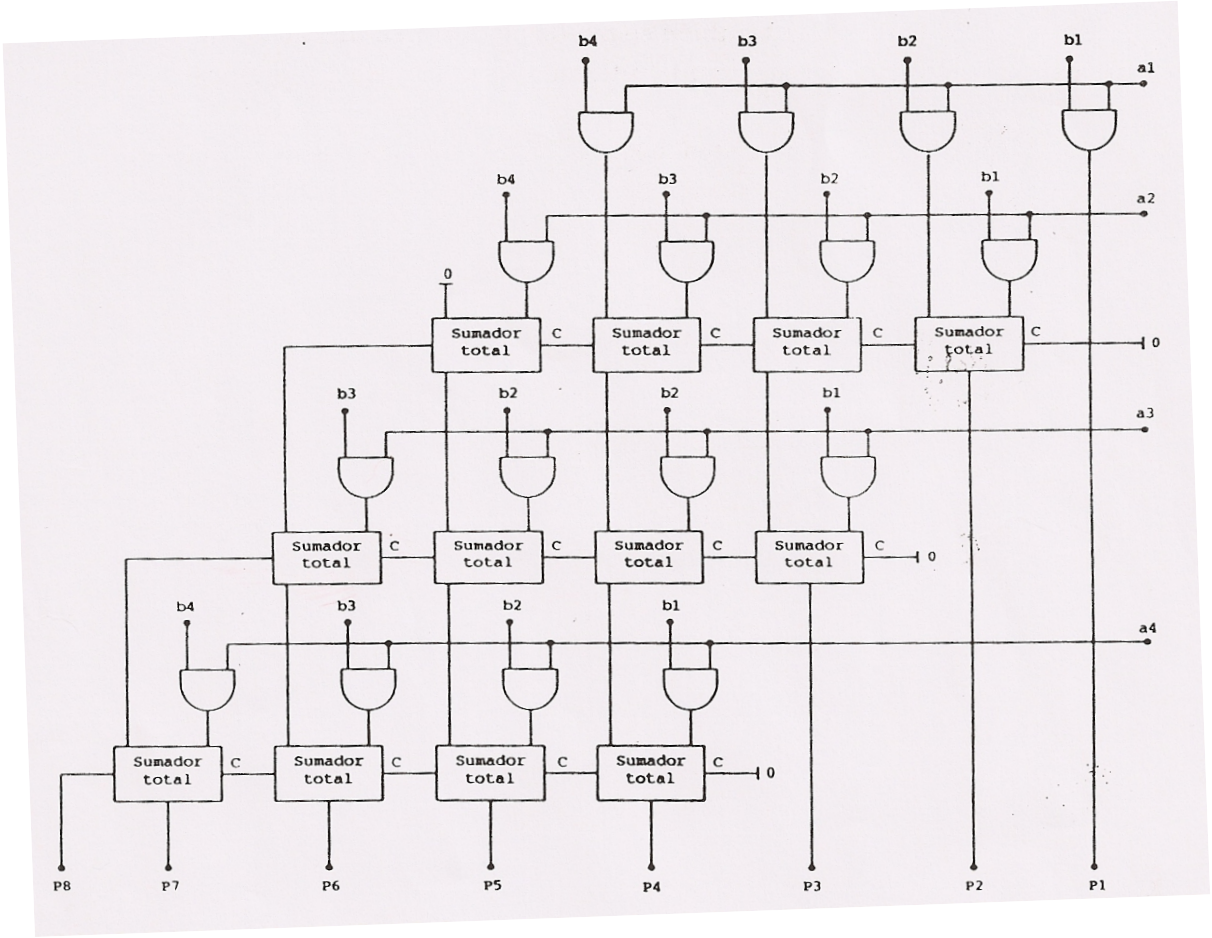
★ A * 5 = A * (4+0+1)



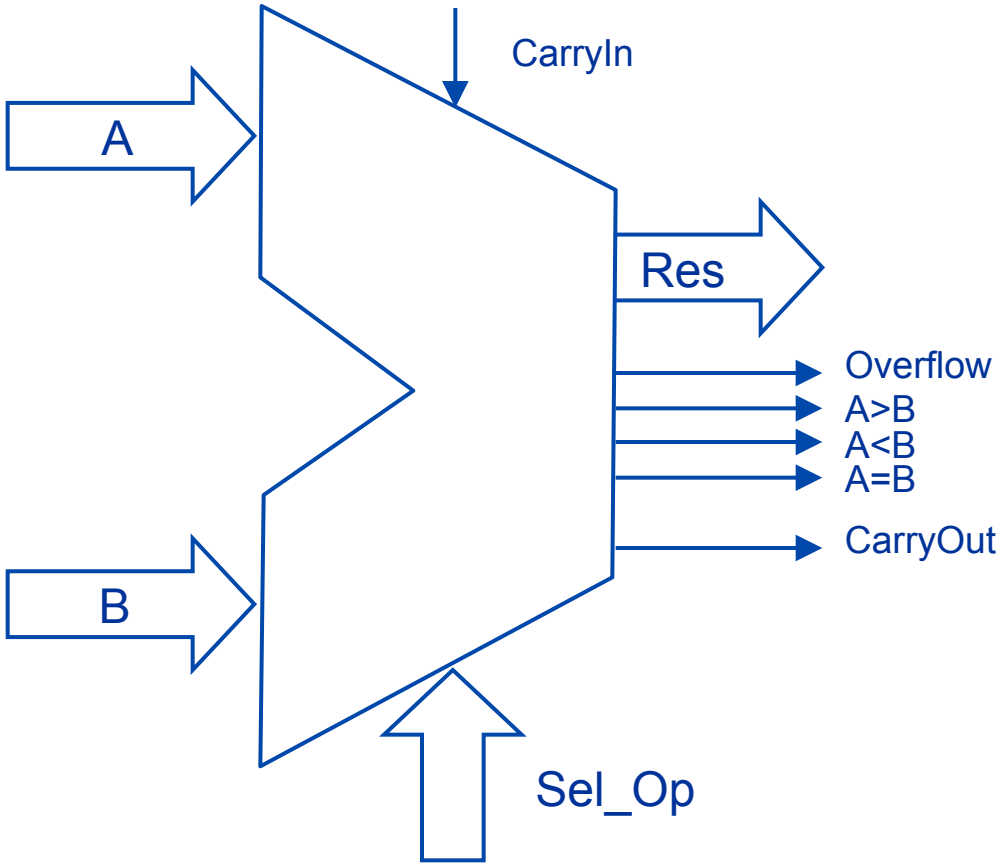
Multiplier.



Multiplier.



Arithmetic-Logic Unit



Arithmetic-Logic Unit.

Combinational

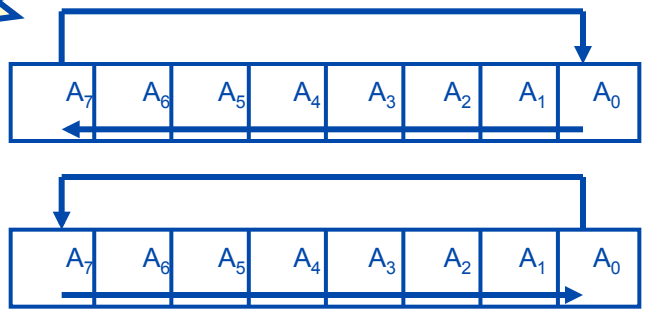
Block for execution of arithmetic-logic operations:

- Adding
  - A + B
- Substraction
  - A - B
- 2-s Complement
  - - B
- Comparison
  - A > B
  - A < B
  - A = B
- Shift Left
  - SHL(A) ←
- Shift Right
  - SHR (A) →

SUBSTRACTION

Logic operations (bit to bit)

- AND
- OR
- XOR
- XNOR
- NOT



References

- “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