

# Diseño de circuitos lógicos

# Compuertas lógicas

- Son los circuitos electrónicos que implementan una operación lógica.
- Toda compuerta tiene una o más entradas y una salida, que corresponden a los operandos y al resultado de la operación lógica.
- Gráficamente cada tipo de compuerta se identifica por un símbolo.
- Las conexiones entre compuertas se representan mediante líneas.

# Compuertas lógicas básicas (I)

NOT



A	$\sim A$
0	1
1	0

AND



A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

# Compuertas lógicas básicas (II)

OR



A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

XOR



A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

# Metodología de diseño

- 1) Si se parte de una tabla de verdad, obtener la expresión lógica utilizando la FND ó FNC.
- 2) Opcionalmente, simplificar la expresión lógica
- 3) Llevar la expresión lógica al circuito digital correspondiente

# Ejemplo:

## Diseño de un sumador completo (1)

Suma binaria

    Cin  
    A  
+   B  
-----  
Cout R

Tabla de verdad del sumador

Cin	A	B	R	Cout
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

# Ejemplo:

## Diseño de un sumador completo (2)

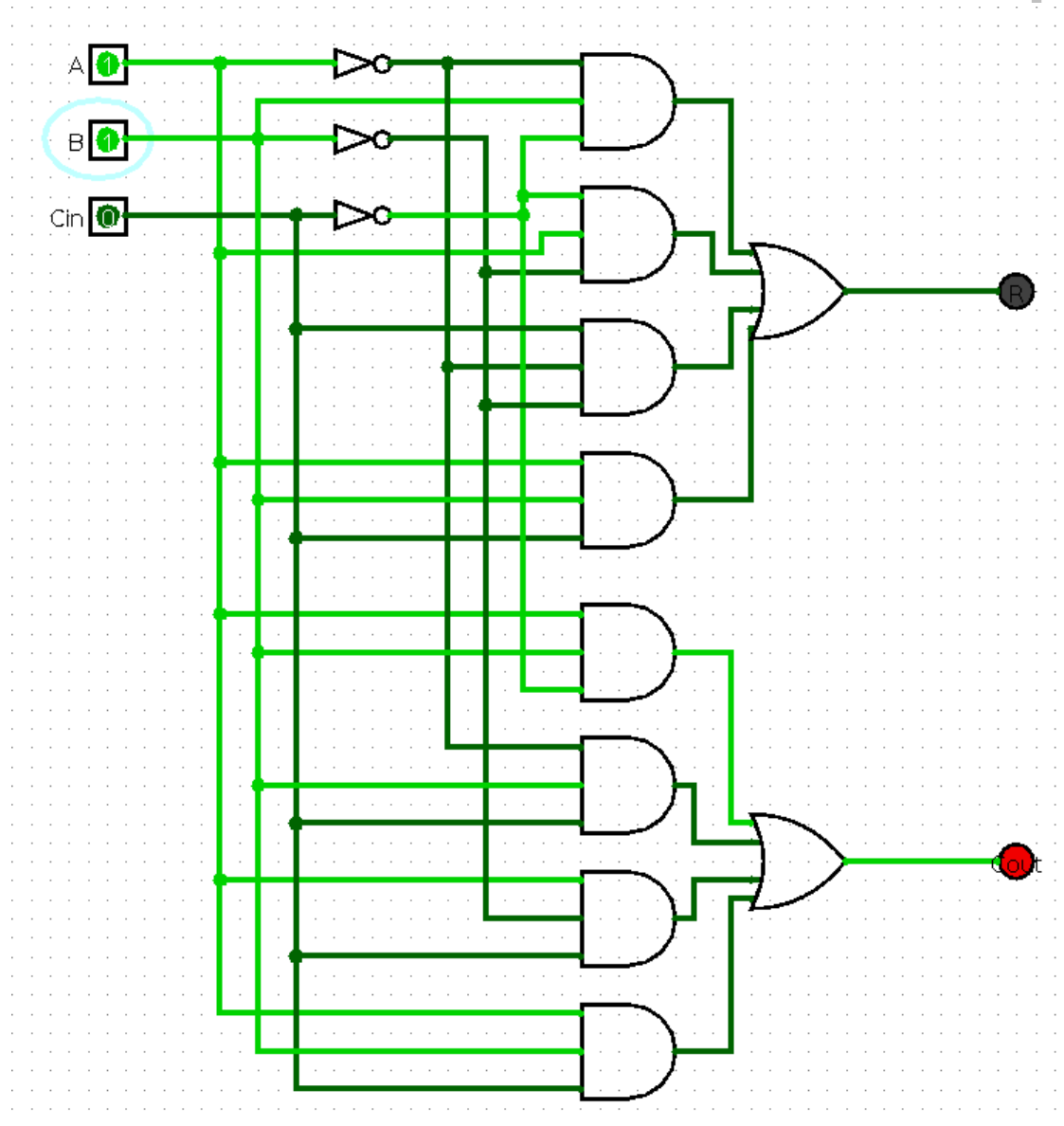
Expresiones lógicas (sin simplificar):

$$R = (\neg C_{in} \wedge \neg A \wedge B) \vee (\neg C_{in} \wedge A \wedge \neg B) \vee \\ (C_{in} \wedge \neg A \wedge \neg B) \vee (C_{in} \wedge A \wedge B)$$

$$C_{out} = (\neg C_{in} \wedge A \wedge B) \vee (C_{in} \wedge \neg A \wedge B) \vee \\ (C_{in} \wedge A \wedge \neg B) \vee (C_{in} \wedge A \wedge B)$$

# Ejemplo:

## Diseño de un sumador completo (3)





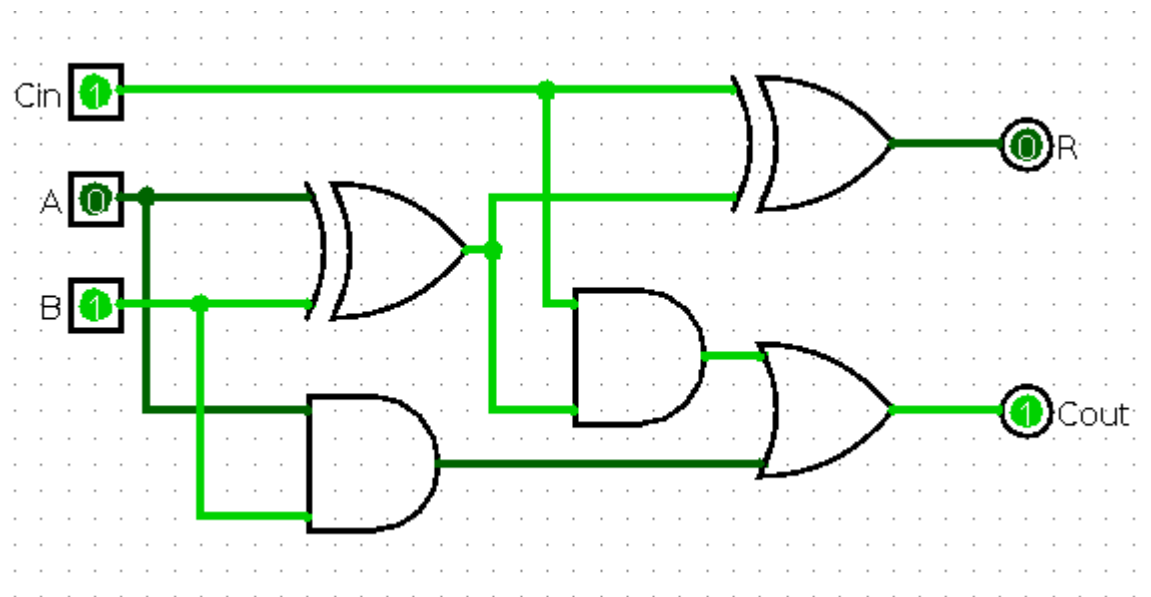
# Simplificación

- Simplificando las expresiones lógicas se obtienen:

$$\begin{aligned} R &= \neg C_{in} \wedge [ (\neg A \wedge B) \vee (A \wedge \neg B) ] \vee C_{in} \wedge [ (\neg A \wedge \neg B) \vee (A \wedge B) ] \\ &= \neg C_{in} \wedge [ A \oplus B ] \vee C_{in} \wedge [ \neg(A \oplus B) ] \\ &= C_{in} \oplus (A \oplus B) \end{aligned}$$

$$\begin{aligned} C_{out} &= [(\neg C_{in} \vee C_{in}) \wedge (A \wedge B)] \vee [C_{in} \wedge [(\neg A \wedge B) \vee (A \wedge \neg B)]] \\ &= (A \wedge B) \vee [C_{in} \wedge (A \oplus B)] \end{aligned}$$

# Circuito simplificado



# Otros circuitos lógicos comunes

- Operaciones aritméticas, la ALU de un procesador:
  - Suma, resta
  - Multiplicación, división
- Multiplexores, demultiplexores.
-

# Simulación y diseño de circuitos lógicos

- **Logisim (Java)**

<http://www.cburch.com/logisim/>

- **Logisim Evolution (Java)**

<https://github.com/logisim-evolution/logisim-evolution>

- **Logic Gate Simulator**

<http://www.kolls.net/gatesim/>

- **CEDAR Logic Simulator**

<http://sourceforge.net/projects/cedarlogic/>

- **Logic.ly** – Basado en el browser

<https://logic.ly/demo/>

# Bibliografía

- [Digital Circuits](#)  
Wikibooks.
- Digital Logic Design Principles, [Chapter 4](#).