

# Guía de ejercicios # 3 - Lógica Digital

Organización de Computadoras

UNQ

## De la lógica proposicional a la lógica digital

Los ejercicios de esta sección te permitirán repasar lo que estudiaste de lógica proposicional y entender el paralelismo con las compuertas digitales.

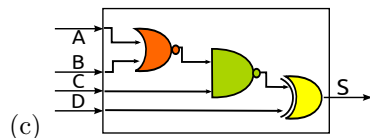
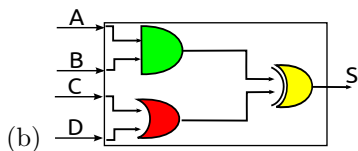
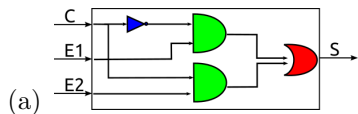
1. Escribir la tabla de verdad de las siguientes expresiones:

- (a)  $p \bullet q + \overline{p \bullet q}$
- (b)  $(\overline{p} + q) \bullet (p + \overline{q})$
- (c)  $(p \bullet q \bullet w) + p \bullet \overline{q} \bullet \overline{w} + \overline{p} \bullet q \bullet \overline{w}$
- (d)  $p \bullet (q \bullet \overline{w} + \overline{q} \bullet w)$

2. Derivar la fórmula de verdad de la tabla y escribir dicha fórmula. Luego, para **verificar** el resultado, construir la **tabla de verdad** en base a la expresión obtenida.

| A | B | C | $F(A, B, C)$ |
|---|---|---|--------------|
| 0 | 0 | 0 | 0            |
| 0 | 0 | 1 | 1            |
| 0 | 1 | 0 | 1            |
| 0 | 1 | 1 | 0            |
| 1 | 0 | 0 | 1            |
| 1 | 0 | 1 | 0            |
| 1 | 1 | 0 | 0            |
| 1 | 1 | 1 | 0            |

3. Derivar la fórmula de verdad para los siguientes circuitos:



## Circuitos combinatorios

Los ejercicios de esta sección te permitirán comprender los conceptos básicos de los circuitos combinatorios

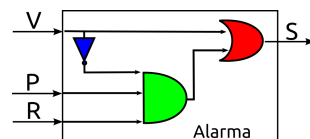
e incorporar mecanismos para construir circuitos: a partir de un enunciado (*from scratch*) o componiendo otros circuitos mas simples.

4. Diseñar un circuito **And3** de 3 entradas y una salida que compute la conjunción entre sus entradas.
5. Diseñe un circuito de 3 entradas y una salida, cuya salida sea tal que la cantidad de unos entre las entradas y las salidas sea par. Plantee la solución como una tabla de verdad y derive el circuito correspondiente. Pruebe el circuito obtenido con al menos 4 casos, es decir, simule que se ingresan valores (0 o 1) en las entradas y verifique si la salida obtenida es la esperada.
6. Haga el circuito de un comparador de 1 bit. Debe tener 2 entradas (a y b) y 5 salidas ( $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_4$ ,  $S_5$ ), tales que:

- $S_1 = 1 \leftrightarrow a > b$
- $S_2 = 1 \leftrightarrow a \geq b$
- $S_3 = 1 \leftrightarrow a = b$
- $S_4 = 1 \leftrightarrow a \leq b$
- $S_5 = 1 \leftrightarrow a < b$

7. Construir un circuito que a partir de un número de 3 bits  $x_0, x_1, x_2$  tenga 4 salidas que computen el resultado de hacer  $3 * x$ . Si el resultado no entra en 4 bits, debe devolver 1 en sus cuatro salidas. Escribir ejemplos con distintas cadenas de entradas y su respectiva salida para comprender el problema.

8. Se tiene un circuito y se sabe que su función es la de disparar una alarma en una habitación con dos sensores: uno en la única ventana y otro en la única puerta. Si se abre la ventana, la alarma debe sonar de inmediato, pero si se abre la puerta entonces comienza a correr un reloj (Externo a nuestro circuito) que activa la entrada R luego de 5 segundos. El circuito es el siguiente:



¿Cómo se comprueba que el circuito cumple con su función?

## Circuitos específicos

Los ejercicios de esta sección te permitirán entender la motivación para los circuitos aritméticos y otros circuitos recurrentes, así como comprender sus usos y cómo usarlos para componer otros circuitos.

9. Se tiene el siguiente **demultiplexor**:

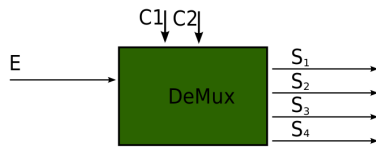


Figure 1: Caja negra de Demultiplexor

Diseñar el circuito teniendo en cuenta la siguiente información:

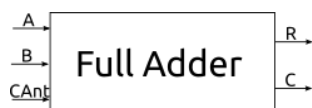
El demultiplexor tiene 2 líneas de control, 1 línea de entrada y 4 líneas de salida.

- Si  $c_1$  y  $c_0$  valen 0 entonces se proyecta en  $s_1$  y las restantes salidas valen 0.
- Si  $c_1 = 0$  y  $c_0 = 1$  entonces se proyecta en  $s_2$  y las restantes salidas valen 0.
- Si  $c_1 = 1$  y  $c_0 = 0$  entonces se proyecta en  $s_3$  y las restantes salidas valen 0.
- Si  $c_1$  y  $c_0$  valen 1 entonces se proyecta en  $s_4$  y las restantes salidas valen 0.

10. Dibujar el circuito de un **decodificador** de 2 líneas de entrada ( $e_i$ ) y 4 líneas de salida ( $s_i$ ), cuya tabla de verdad es la siguiente:

| $e_1$ | $e_0$ | $s_3$ | $s_2$ | $s_1$ | $s_0$ |
|-------|-------|-------|-------|-------|-------|
| 0     | 0     | 0     | 0     | 0     | 1     |
| 0     | 1     | 0     | 0     | 1     | 0     |
| 1     | 0     | 0     | 1     | 0     | 0     |
| 1     | 1     | 1     | 0     | 0     | 0     |

11. Usando el **circuito anterior**, dibujar un demultiplexor de 1 línea de entrada, 2 líneas de control y 4 líneas salidas.
12. Diseñar un *full adder* de 1 bit, es decir, un circuito capaz de sumar dos cadenas de 1 bit cada una y un carry de entrada, y cuyo resultado es una cadena de un bit y un carry de salida.



13. Diseñar un restador de 1 bit:



14. Diseñar un restador con carry de 1 bit, que tiene 3 entradas: los 2 operandos y el borrow de entrada (el anterior pidió 1) y tiene 2 salidas, el resultado de la resta y el nuevo borrow.
15. Diseñar un *full adder* de 4 bits reusando circuitos que conozca. Un *full adder* de 4 bits tiene como entrada un carry anterior y dos cadenas de 4 bits cada una (es decir, 9 entradas en total), y su salida es una cadena de 4 bits y un carry de salida (es decir, 5 salidas en total).
16. Diseñar un *restador* de 4 bits reusando circuitos que conozca.