



## IIC1001 — Algoritmos y Sistemas Computacionales — 2024-1

### Actividad Evaluada 1 - Codificación y Compuertas Lógicas

Miércoles 03-Marzo-2024

#### Instrucciones

Esta es una actividad evaluada y contará con una nota. Deberán realizarla de acuerdo a los grupos designados AE1 según Canvas, y entregarla **al finalizar el módulo de clases (16:00)**. La entrega se realiza mediante un buzón en Canvas, **Solo una persona debe subir una foto de su desarrollo con los nombres de todos los integrantes presentes a Canvas**. No se aceptarán entregas posteriores. No está permitido el uso de celular o computador para resolver los ejercicios.

#### 1. Mensaje Codificado

Un byte (8 bit) puede almacenar valores numéricos (en base 2) desde 0 a 255 si solo consideramos enteros positivos, o bien desde 127 a  $-128$  si utilizamos la representación de números negativos en “complemento de 2”. Alternativamente un byte también puede representar caracteres definidos de acuerdo a un mapa (*encoding*); para esto tomamos el valor numérico que representa el byte y buscamos su correspondencia en un mapa como ASCII.

Por ejemplo, para representar el valor  $(740)_{10} = (10\ 1110\ 0100)_2$  necesitamos 10 bit, por lo tanto hay que ocupar 2 bytes y rellenar con ceros:  $(0000\ 0010\ 1110\ 0100)$ . Si queremos representar estos dos byte como hexadecimal, podemos escribir  $(02\ E4)_{16}$ .

Ahora, si queremos representar el texto LW, podemos usar 2 bytes. El primero almacena el valor numérico  $(76)_{10} = (100\ 1100)_2 = (4C)_{16}$  que corresponde a la letra L, y el segundo byte almacena el valor numérico  $(87)_{10} = (101\ 0111)_2 = (57)_{16}$  que corresponda la letra W. Escribimos 4C 57.

**[6pts]** Escriba un mensaje codificado en bytes. Cada byte debe estar escrito usando dos caracteres hexadecimales. Debe seguir exactamente el siguiente formato:

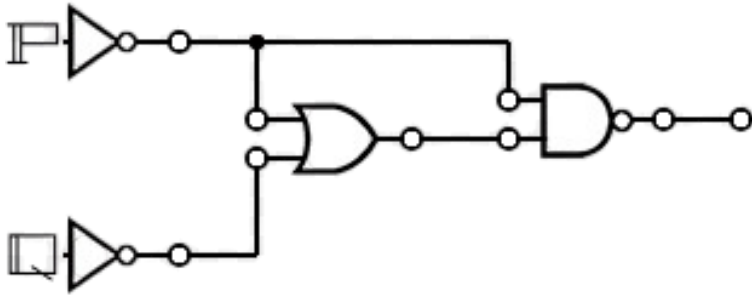
- **[1pt]** 2 bytes donde se almacene el valor numérico  $(1001)_{10}$ . == 03 E9
- **[1pt]** 1 byte que almacene el número de su grupo
- **[1pt]** 2 byte que almacenen la cantidad de caracteres que tendrá el mensaje que va a escribir
- **[3pts]** Un mensaje codificado en ASCII. Cada caracter utiliza un byte. El mensaje debe contener **al menos** 15 caracteres. El último caracter del mensaje **debe** ser el valor 0.

**BONUS post-clase [2pts]**. Al finalizar la clase, y después de haber hecho su entrega en Canvas, entregue su hoja al grupo siguiente (si es el último grupo, se la entrega al grupo 1). Debe decodificar el mensaje que recibió y subir la traducción correspondiente a un segundo buzón de Canvas. Esto tiene plazo hasta las 20:00, y es opcional.

## 2. Tablas de verdad desde compuertas lógicas [3pts]

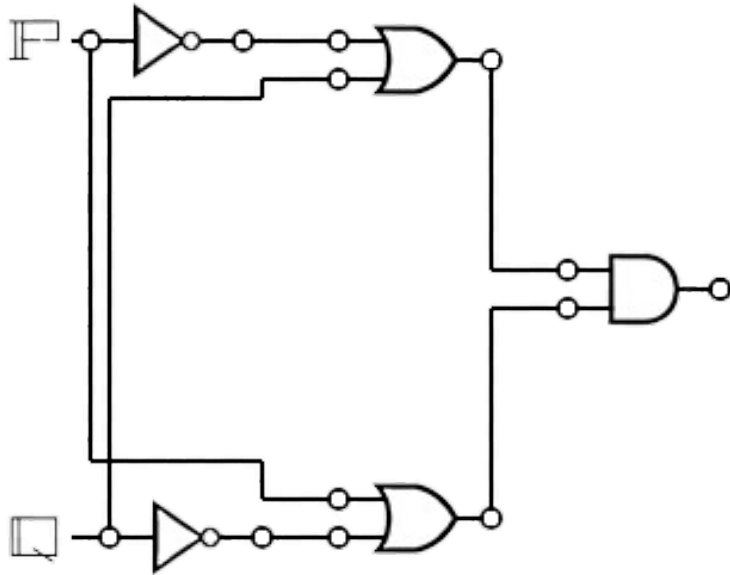
Complete las tablas de verdad para las siguientes compuertas lógicas. La entrada superior es P y la inferior es Q. Puede usar las columnas para almacenar valores intermedios, pero lo que importa es indicar claramente la salida final.

1) [1pt]



<b>p</b>	<b>q</b>	$\neg p \vee \neg q$	$\neg p \wedge (\neg p \vee \neg q)$
0	0	1	1
0	1	1	1
1	0	1	0
1	1	0	0

2) [2pts]



<b>p</b>	<b>q</b>	$\neg p \vee q$	$p \vee \neg q$	$(\neg p \vee q) \wedge (p \vee \neg q)$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	1	1	1

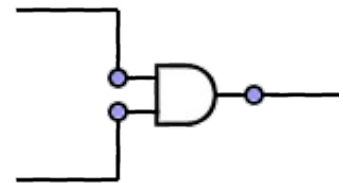
### 3. Compuertas lógicas desde tablas de verdad [3pts]

Complete la tabla de verdad y dibuje el circuito lógico que permite representar las siguientes proposiciones.

(1 puntos c/u):

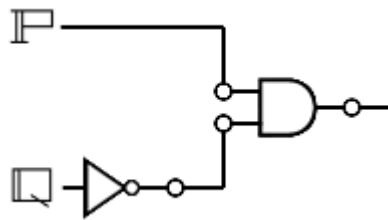
Ejemplo:

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1



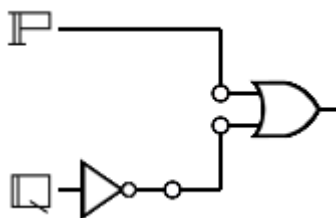
1)

p	q	$\neg q$	$p \wedge \neg q$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0



2)

p	q	$\neg q$	$q \rightarrow p$
0	0	1	1
0	1	0	0
1	0	1	1
1	1	0	1



3)

p	q	r	$(p \wedge q)$	$(q \vee \neg r)$	$(p \wedge q) \wedge (q \vee \neg r)$
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	0	1	0
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	1	1

