

Nro. ord.	Apellido y nombre	L.U.	#hojas
83	Gurbamov Manuel		2

SISTEMAS DIGITALES - Primer Parcial
Segundo Cuatrimestre 2024

Ej.1	Ej.2	Ej.3	Ej.4	Nota
2	0	1	4	7

Correctorx:

Aclaraciones

- Anote apellido, nombre, LU y numere *todas* las hojas entregadas, entregando los distintos ejercicios en hojas separadas.
- El parcial no es a libro abierto, justifique sus respuestas.
- El parcial se aprueba con 6 y se deben tener ambos parciales aprobados para aprobar la materia (promoción directa).

Ejercicio 1 (3 pts.) Responder y justificar la respuesta:

- ¿Cuál es el rango de representación de un número de cuatro bits en signo y magnitud? ¿Cuál es el rango de representación de un número de ocho bits en complemento a dos?
- ¿Cómo se calcula el inverso aditivo de un número n en complemento a dos de k bits?
- Para dos números a y b de k bits, para una operación de suma cuyo resultado nombramos c , ¿cómo se determina el **carry**, sobre qué tipo de datos lo observamos? ¿Cómo se determina el **overflow**, sobre qué tipo de datos lo observamos? Explique por qué se definen de esta manera.

Ejercicio 2 (1 punto) Responder:

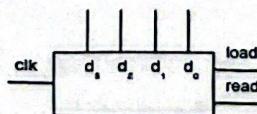
1. Sea $p \downarrow q = \overline{p \cdot q}$ ¿Alcanza este único operador (NAND) para representar todas las funciones booleanas?
2. Sea $p \downarrow q = \overline{p + q}$ ¿Alcanza este único operador (NOR) para representar todas las funciones booleanas?

Consejo: recuerde que las operaciones de conjunción (AND), disyunción (OR) y negación son suficientes para representar todas las funciones booleanas.

Ejercicio 3 (2 pts.) Dibujar circuitos que implementen las siguientes funciones booleanas:

1. $f(A, B, C) = A \cdot B \cdot C$ usando 2 compuertas NOR y varias compuertas NOT.
2. $f(A, B) = \overline{((A \cdot B) + (\overline{B} \cdot A))} \cdot \overline{B}$ ¿Para qué valores de A y B la función devuelve un 1?

Ejercicio 4 (4 pts.) *Registro bidireccional* Diseñar un registro *bidireccional* de cuatro bits. Este tipo de registros es un circuito con dos señales de control de entrada (load, read, el clk) y cuatro señales de entrada y salida de datos (d_0 a d_3). Su funcionamiento es el siguiente: si la señal load vale 1 cuando clk alcanza su flanco ascendente, almacena los valores recibidos en d_0 a d_3 ; en cambio, si read está alta, se emite el valor almacenado en el registro por esas mismas líneas¹. Las señales read y load nunca valen 1 simultáneamente.



¹ Ayuda: utilice componentes de tres estados.

\checkmark = ① En signo y magnitud, usamos el bit más significativo para representar si el número es positivo (0) o negativo (1).

En un número de 4 bits, usando signo y magnitud podemos representar entre -15 y 7 .

$$1111 = -15 \text{ (número más bajo que podemos representar)}$$

$$0111 = 7 \text{ (número más alto que podemos representar.)}$$

Si tenemos un número de K bits en complemento a dos, podemos calcular su inverso aditivo reemplazando ceros por unos y unos por ceros para luego a este número sumarle 1.

Por ejemplo, veamos el caso del 5:

$$5 = 0101 = 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Para calcular su inverso aditivo, invertimos los bits $\Rightarrow 1010$ y sumamos uno.

$$\begin{array}{r} 1010 \\ + \quad 1 \\ \hline 1011 \end{array}$$

Entonces, el inverso aditivo es -5.

Sean a y b números de K bits, analicemos la suma:

$$+ \quad a_K + a_{K-1} + \dots + a_0$$

$$+ \quad b_K + b_{K-1} + \dots + b_0$$

$$+ \quad C_K + C_{K-1} + \dots + C_0$$

Tanto en complemento a dos como en signo, definimos el carry en la suma como un bit extra que se genera en el resultado al sumar los bits más significativos de a y b y el carry C_K

En este ejemplo, siendo K el número de bits, debemos ver si el resultado C es de la cantidad $K+1$ de bits, más específicamente si este último es un 1.

Ej 2) No hay X

Por otro lado, definimos al overflow como la imposibilidad de representar el resultado en el rango que estamos trabajando.

Por ejemplo, en ~~suma~~ complemento a dos siempre la suma de dos números negativos debe dar negativo (en decir, si a y b tienen a 1 en el valor del bit más significativo, c debe tenerlo también) y análogamente la suma de dos positivos debe ser un número positivo.

En cuanto a signo, debemos asegurarnos de que el resultado de la suma sea representable en la cantidad de bits que estamos trabajando para no tener overflow.

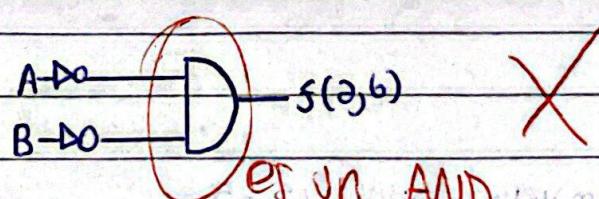
③ Dibujar circuitos que representen:

A) $f(a,b,c) = a \cdot b \cdot c$ Compuertas disponibles: \rightarrow $D_0 \times 2$
 \rightarrow Variables.

B) $f(a,b) = ((A \cdot B) + (\bar{B} \cdot A)) \cdot \bar{B}$

$\Rightarrow B$ $((A \cdot B) + (\bar{B} \cdot A)) \cdot \bar{B} = \bar{A} \cdot \bar{B}$ \Rightarrow Armo el circuito

observo que esto es equivalente
a A por tabla de verdad



A	b	$(A \cdot B) + (A \cdot \bar{B})$
0	0	0
0	1	0
1	0	1

es un AND

\Rightarrow ¿Para qué valores el circuito da 1?

A	B	$\bar{A} \cdot \bar{B}$
0	0	1
0	1	0
1	0	0
1	1	0

El circuito vale 1 cuando $a, b = 0$.

OK

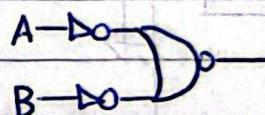
3) A $f(a, b, c) = a \cdot b \cdot c$ pero debo usar dos compuertas NOR y varias compuertas NOT.

Entonces uso álgebra de boole para buscar fórmulas equivalentes

$$a \cdot b \cdot c = (a \cdot b) \cdot c = (\overline{a} \cdot \overline{b}) \cdot c = \overline{\overline{a} + \overline{b}} \cdot c$$

de modo que

Aquí tengo dos NOT y un NOR.



⇒ Debo ver cómo trato a la entrada C tal que solo de 1 la salida cuando C sea 1

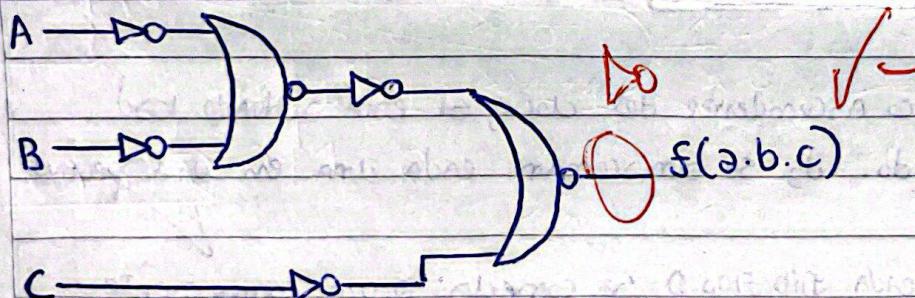
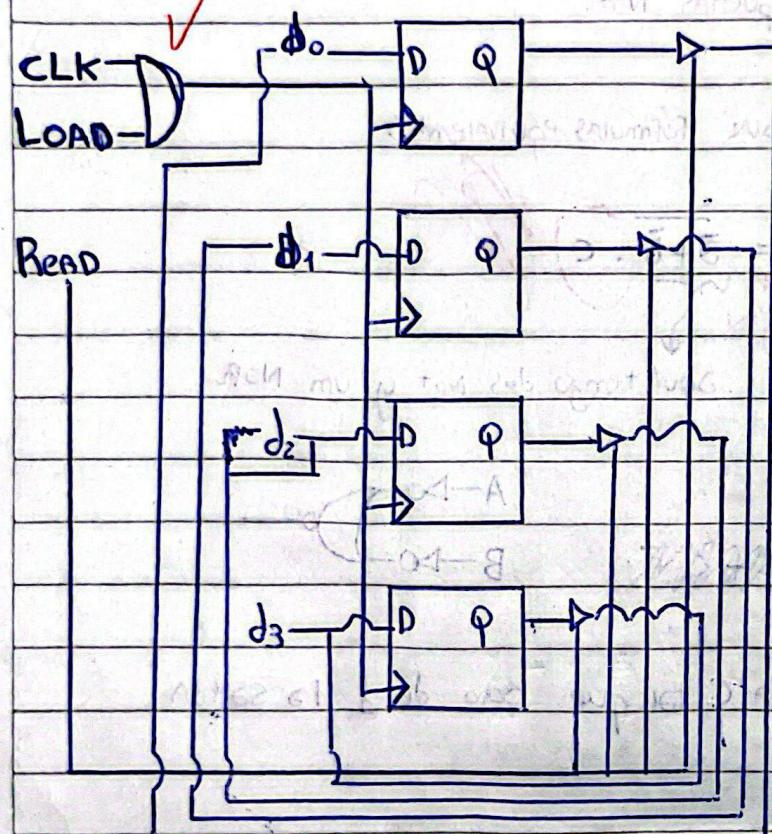


TABLA de Verdad

	A	B	$\overline{A} + \overline{B}$	C	\overline{C}	$(\overline{A} + \overline{B}) + \overline{C}$
Pruebo que la salida es 1 cuando (solo cuando) $A, B, C = 1, 1, 1$.	0	0	1	0	1	0
	0	0	1	1	0	0
	0	1	1	0	1	0
	1	0	1	0	1	0
	0	1	1	1	0	0
	1	0	1	1	0	0
	1	0	1	0	1	0
	1	1	0	1	0	1

④ Armar registro bidireccional

Comportamiento: cload y clk? Se almacenan las entradas en el registro
c.read? Se emite el valor del registro por las entradas.



funcionamiento: En el FIANO ASCENDENTE del clock, si está activado load los valores de las entradas $d_0 \dots d_2$ se almacenan cada una en su respectivo flip flop D.

Por otro lado, la salida de cada flip flop D se conecta a un componente de tres estados que se controla con la señal read.

Como las señales load y read nunca valen 1 simultáneamente, no necesito añadir ninguna otra restricción.