Nro. ord.	Apellido y nombre	L.U.	#hojas
	Manzetti Mauro		4

TN

SISTEMAS DIGITALES - **Primer Parcial**

Segundo Cuatrimestre 2024

Ej.1	Ej.2	Ej.3	Ej.4	Nota
3	1	2	4	10

Correctorx: Marcos

Felicidades! Excelente Parcial :)

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).

3/3

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. SS y C2

1/1

Ejercicio 2 (1 punto) Responder:

1. Sea $p|q = \overline{p.q}$ ¿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.

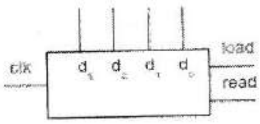
2/2

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

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

4/4

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 las 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 líneas[1]. Las señales read y load nunca valen *1* simultáneamente.



[1] *Ayuda:* utilice componentes de tres estados.

EJERCICIO 1

Manetti Mauro

i) En S+M, el primer bit determina el signo, y el resto el valor.

$\underbrace{X}_{S}\underbrace{XXX}_{magnitud}$ En 4 bits, el valor mínimo es 1111_2 y el máximo es 0111_2. $1111_{S+M} = -1 \cdot 111_2 = -1 \cdot 7 = -7$

$0111_{S+M} = 111_2 = 7$ por lo que el rango de representación son los valores entre -7 y 7 inclusive.

↘ Qué pasa con el cero? Faltaría aclarar eso

En C2, el primer bit también determina el signo. En 8 bits, el valor mínimo es 10000000 y el máximo es 01111111.

~~10000000 = 127~~

$0111\ 1111 = 1 \cdot 2^0 + 1 \cdot 2^1 + \ldots 1 \cdot 2^6 = 127$

y sé que el mínimo en C2 es el inverso del máximo -1 $= -128$ por lo que el rango de representación son los valores entre -128 y 127 inclusive.

ii) En C2, se calcula el inverso aditivo de un número de k bits invirtiendo cada uno de sus bits y sumándole 1_2.

Ej $inv(0101_2) + 1_2 = 1010_2 + 1_2 = 1011$

$0101_2 = 5 \quad 1011 = -5$

Esta fórmula vale si el inverso aditivo existe, por ej, no vale para -128 en C2!

Además, vale aclarar que viene de la fórmula de $inv(n) = 2^k - n$

iii) Para dos números a y b de k bits, la manera de determinar si hay carry en el bit más a la izquierda es: $C_k = (a_k \cdot b_k) + (c_{k-1} \cdot (a_k + b_k))$, es decir, si el bit de ambos números es 1 o si hay carry del bit derecho y uno de los dos es 1, ya que $1_2 + 1_2 = 10_2$ necesita un bit de más. Esto aplica a SS y C2.

Para determinar si hay overflow en la suma de SS, simplemente hay overflow si hay carry en el primer bit, y el resultado no es válido. Ej:

$$\begin{array}{rl} \overset{1}{0}111_2 & \rightarrow 7_{10} \\ + \ 1100_2 & \rightarrow + \ 12_{10} \\ \hline 1|0011 & \rightarrow 3_{10} \end{array}$$

Como solo puedo devolver la misma cantidad de bits, no puedo representar el número completo y obtengo un resultado incorrecto, ya que está fuera del rango de representación

En C2, uso el carry para operar con números negativos, por ejemplo:

$$\begin{array}{r} 0111_{C2} \\ + \; 1100_{C2} \\ \hline 1|0011_{C2} \end{array} \rightarrow \begin{array}{r} 7_{10} \\ + \; -4_{10} \\ \hline 3_{10} \end{array}$$

lo cual es correcto.

Solo obtengo overflow cuando mi resultado está fuera del rango de representación, y esto se determina si ambos números tienen el mismo signo y el resultado tiene el signo opuesto (ya que eso no tiene sentido)

POS + POS = NEG

V ⟷ NEG + NEG = POS

Ej:

$$\begin{array}{r} 1001_{C2} \\ + \; 1010_{C2} \\ \hline 1|0011_{C2} \end{array} \rightarrow \begin{array}{r} -7_{10} \\ + \; -6_{10} \\ \hline 3_{10} \end{array}$$

Error en OR?

$(a_k = b_k) + \overline{C_{k-1}}$

Más allá de ese detalle, muy bien!

(Habrás entendido mal mis fórmulas, disculpas)

Si querés después explicame cómo lo pensaste, pero que solo tengan el mismo signo a_k y b_k no es condición suficiente, por poner un ejemplo

EJERCICIO 2

Manzotti Mauro

Para demostrar, intento armar un AND, OR, y NOT usando únicamente el operador pedido.

1) NAND: $P|Q$ $(\overline{P\cdot Q})$

AND: ~~AND~~ $P\cdot Q \equiv (P|Q)|(P|Q)$

OR: $P+Q \equiv (P|P)|(Q|Q)$

NOT: $\overline{P} \equiv (P|P)$

A	B	A|B	(A|B)|(A|B)	A.B
0	0	1	0	0
0	1	1	0	0
1	0	1	0	0
1	1	0	1	1

A	B	A|B	(A|A)|(B|B)	A+B
"	"	"	0	0
			1	1
			1	1
			1	1

A	(A|A)	$\bar{A}$
0	1	1
1	0	0

Por lo que alcanza NAND para representar todas las funciones booleanas.

2) NOR: $P\downarrow Q$ $(\overline{P+Q})$

AND: $P\cdot Q \equiv (P\downarrow P)\downarrow(Q\downarrow Q)$

OR: $P+Q \equiv (P\downarrow Q)\downarrow(P\downarrow Q)$

NOT: $\overline{P} \equiv P\downarrow P$

A	A↓A	$\bar{A}$
0	1	1
1	0	0

A	B	A↓B	(A↓B)↓(A↓B) ~~(A↓A)↓(B↓B)~~	A+B
0	0	1	0	0
0	1	0	1	1
1	0	0	1	1
1	1	0	1	1

Por lo que también alcanza solo NOR para representar todas las funciones booleanas.

A	B	A↓A	B↓B	(A↓A)↓(B↓B)	A.B
0	0	1	1	0	0
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	1	1

EJERcicio 3

Manzotti Mauro

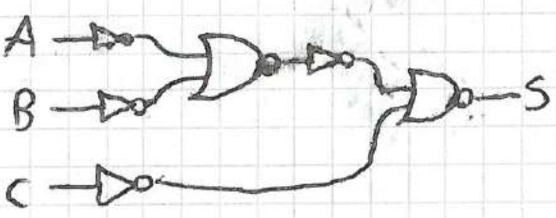
1) $f(A,B,C) = A \cdot B \cdot C$

A	B	C	S
0	0	0	0
0	0	1	0
...			...
1	1	1	1

$A \cdot B = \overline{A} \downarrow \overline{B}$

$A \cdot B \cdot C = \overline{(\overline{A} \downarrow \overline{B})} \downarrow \overline{C}$

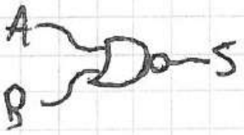
A	B	C	$(\overline{A} \downarrow \overline{B})$	S
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1



2) $F(A,B) = \overline{((A.B)+(\overline{B}.A)).\overline{B}}$

A	B	(A.B)	$(\overline{B}.A)$	$\overline{((A.B)+(\overline{B}.A))}$	S
0	0	0	0	1	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$S \equiv \overline{A}.\overline{B} \equiv \overline{(A+B)} \equiv A \downarrow B$

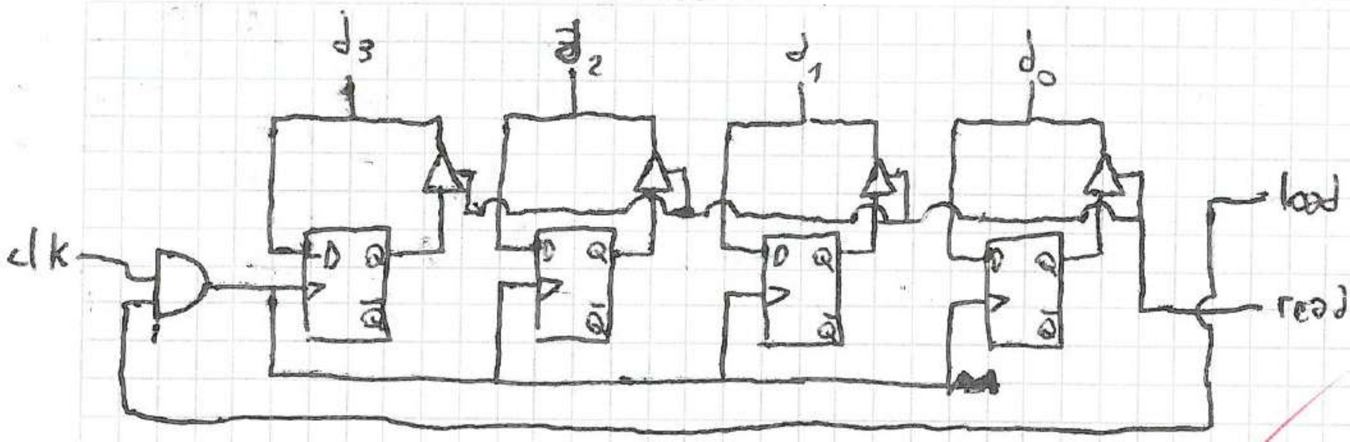


La función solo devuelve 1 cuando ambos A y B son 0.

Genial!

EJERcicio 4

Manzotti Mauro



Excelente!