

PEC2 - Segunda prueba de evaluación continua

Presentación

Después de conocer los sistemas de representación de la información, esta PEC se focaliza en los circuitos lógicos combinacionales. Las funciones lógicas nos permiten describir la funcionalidad de un circuito y mediante las puertas lógicas y los bloques combinacionales podemos implementarlas en un circuito. Todo esto, no se podría hacer sin saber mecanismos de minimización, como mapas de Karnaugh, que nos permiten reducir las dimensiones del circuito que se tiene que implementar.

Competencias

- Conocer y saber aplicar el álgebra de Boole para la manipulación de funciones lógicas.
- Tener nociones tecnológicas de los circuitos digitales y entender la relación entre los circuitos digitales y las funciones lógicas.
- Conocer y saber utilizar las puertas lógicas y los módulos combinacionales en el diseño de circuitos lógicos.

Objetivos

- Saber aplicar las operaciones básicas y los axiomas del álgebra de Boole.
- Saber representar las funciones lógicas mediante tablas de verdad.
- Saber representar las funciones lógicas mediante expresiones algebraicas.
- Saber analizar un circuito combinacional.
- Saber realizar un cronograma a partir de un circuito digital combinacional.
- Saber sintetizar una función a dos niveles.
- Saber diseñar un circuito combinacional sencillo a partir de los bloques combinacionales de los materiales.

Recursos

Los recursos que se recomienda utilizar para esta PEC son los siguientes:

- **Básicos:** El módulo 3 de los materiales. En cada pregunta, se indica qué sección de los materiales se puede encontrar la información para resolverlos. Notad que también existe en los materiales una cantidad muy extensa de ejercicios para ver cómo se pueden resolver
- **PEC anteriores:** En el aula de CANVAS, en recursos adicionales podéis encontrar PECs de otros semestres resueltas.
- **Complementarios:** KeMap, VerilCIRC, VerilCHART y el Wiki de la asignatura.



Criterios de valoración

- Razonad la respuesta en todos los ejercicios. Las respuestas sin justificación no recibirán puntuación.
- Los ejercicios realizados con IA generativa no recibirán puntuación.
- Los ejercicios que se detecten como plagio mediante la herramienta de plagio de la universidad no recibirán puntuación.
- La valoración está indicada en cada uno de los subapartados

Uso de herramientas de IA

En esta actividad no está permitido el uso de herramientas de inteligencia artificial.

En el plan docente y en el [web sobre integridad académica y plagio](#) de la UOC encontraréis información sobre qué se considera conducta irregular en la evaluación y las consecuencias que puede tener.

Formato y fecha de entrega

- Para dudas y aclaraciones sobre el enunciado debéis dirigiros al consultor responsable de vuestra aula.
- Hay que entregar la solución en un fichero PDF utilizando una de las plantillas entregadas conjuntamente con este enunciado.
- Se debe entregar a través de la aplicación de **Entrega de la Actividad** correspondiente del apartado **Contenidos** de vuestra aula.
- La fecha límite de entrega es el 9 de abril (a las 24 horas).

Enunciado de la PAC

Ejercicio 1 [25 %]

Dada la siguiente expresión lógica de la función $f(a,b,c,d)$:

$$f(a,b,c,d) = (\bar{a} + b + c) + (a \cdot b \cdot c) + \bar{c} + (a \oplus b) \cdot d + \bar{d}$$

- [10 %] Simplificad la función f usando las propiedades y las leyes del álgebra de Boole. (Secciones 1.2. Álgebra de Boole y 1.3.1. Expresiones algebraicas)

A continuación, se muestra una posible minimización a partir de las leyes del álgebra de Boole. Tened en cuenta que pueden existir varias soluciones según el orden de aplicación de las leyes:



$$f(a,b,c,d) = (a' + b + c)' + abc + c' + (a \oplus b)d + d'$$

$$(a' + b + c)' = ab'c' \quad \text{ley de De Morgan}$$

$$ab'c' + c' = c' \quad \text{ley de absorción}$$

$$(a \oplus b) = a'b + ab' \quad \text{desarrollo del xor}$$

$$(a'b + ab')d = a'bd + ab'd \quad \text{prop. distributiva}$$

$$f(a,b,c,d) = abc + c' + a'bd + ab'd + d'$$

$$abc + c' = (a + c')(b + c')(c + c') \quad \text{prop. distributiva respecto la disyunción (+)}$$

$$a'bd + d' = (a' + d')(b + d')(d + d') \quad \text{prop. distributiva respecto la disyunción (+)}$$

$$ab'd + d' = (a + d')(b' + d')(d + d') \quad \text{prop. distributiva respecto la disyunción (+)}$$

$$f(a,b,c,d) = (a + c')(b + c')(c + c') + (a' + d')(b + d')(d + d') + (a' + d')(b + d')(d + d')$$

$$c + c' = 1 \text{ y } d + d' = 1 \quad \text{complementación}$$

$$(a + c')(b + c')(c + c') = (a + c')b + (a + c')c' \quad \text{prop. distributiva}$$

$$(a' + d')(b + d')(d + d') = (a' + d')b + (a' + d')d' \quad \text{prop. distributiva}$$

$$(a + d')(b' + d')(d + d') = (a + d')b' + (a + d')d' \quad \text{prop. distributiva}$$

$$f(a,b,c,d) = (a + c')b + (a + c')c' + (a' + d')b + (a' + d')d' + (a + d')b' + (a + d')d'$$

$$= ab + bc' + ac' + c' + a'b + bd' + ad' + d' + ab' + b'd' + ad' + d' \quad \text{prop. distributiva}$$

$$bc' + ac' + c' = c' \quad \text{ley de absorción}$$

$$bd' + ad' + d' = d' \quad \text{ley de absorción}$$

$$b'd' + ad' + d' = d' \quad \text{ley de absorción}$$

$$f(a,b,c,d) = ab + a'b + ab' + c' + d'$$

$$ab + a'b = (a + a')b = b \quad \text{prop. distributiva y complementación}$$

$$ab + ab' = (b + b')a = a \quad \text{prop. distributiva y complementación}$$

Por lo tanto, el resultado de la simplificación es el siguiente:

$$f(a,b,c,d) = a + b + c' + d'$$



b) [5%] Obtened la tabla de verdad de la función. (Sección 1.3.2. Tablas de verdad)

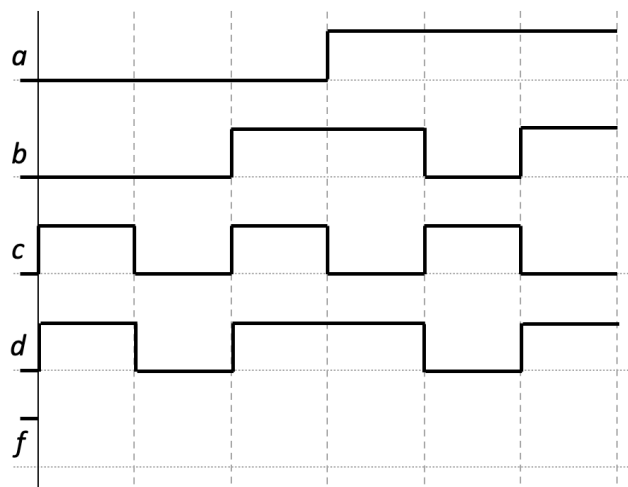
NOTA: Tenéis disponible el ejercicio a VerilCHART para verificarlo. No hay limitación en el número de intentos.

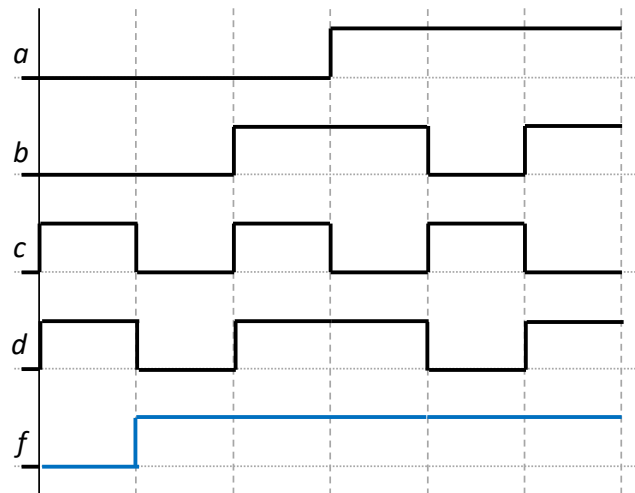
En base a la función original o la simplificación hecha en el apartado a), calculamos el valor de la función para cada combinación posible de las entradas.

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

c) [10%] Completad el siguiente cronograma (Sección 2.2.1. Retardos. Cronogramas. Niveles de puertas):

En base a la tabla de verdad obtenida en el apartado anterior, indicamos el valor de la función para cada combinación posible de las entradas





NOTA: Tenéis disponible el ejercicio a VerilCHART para verificarlo. No hay limitación en el número de intentos.

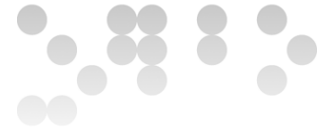
Ejercicio 2 [20 %]

Dada la tabla de verdad siguiente:

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>	<i>g</i>
0	0	0	0	X	1
0	0	0	1	0	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	X	0
0	1	0	1	0	0
0	1	1	0	1	1
0	1	1	1	0	0
1	0	0	0	0	1
1	0	0	1	X	0
1	0	1	0	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	1	X	0
1	1	1	0	1	1
1	1	1	1	1	0

- a) [7,5%] Utilizando el método de Karnaugh, indicad las dos funciones mínimas posibles de la función *f*. ¿Cuál de las dos expresiones mínimas utiliza una cantidad menor de puertas lógicas? Implementad esta última opción con puertas lógicas en dos niveles. (Sección 2.3.2. Síntesis mínima a dos niveles. Método de Karnaugh)

El mapa de Karnaugh de la función *f* y las dos posibles agrupaciones que minimizan la función son las siguientes:



ab \ cd	00	01	11	10
00	X	X	0	0
01	0	0	X	X
11	0	0	1	1
10	1	1	1	0

Donde obtenemos la función mínima:

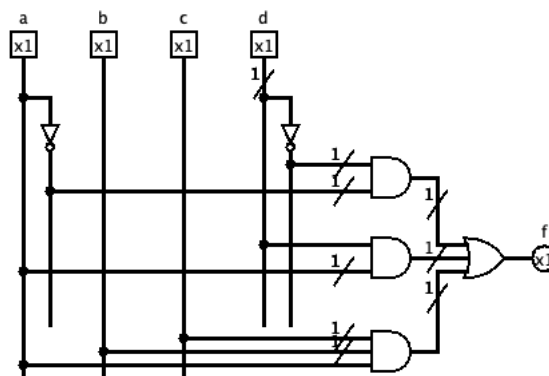
$$f = a'd' + ad + abc$$

ab \ cd	00	01	11	10
00	X	X	0	0
01	0	0	X	X
11	0	0	1	1
10	1	1	1	0

Donde obtenemos la función mínima:

$$f = a'd' + ad + bcd'$$

Estas dos funciones usan la misma cantidad puertas lógicas. La implementación de la primera función es la siguiente:





- b) [5%] Hemos preguntado a una herramienta de IA generativa la minimización mediante el método de Karnaugh de la función g , y nos ha dado el siguiente resultado (Sección 2.3.2. Síntesis mínima a dos niveles. Método de Karnaugh):

Paso 1: **Organización en un mapa de Karnaugh 4x4**

ab\cd	00	01	11	10
00	1	0	1	0
01	0	0	0	1
11	1	0	1	1
10	1	0	1	1

Paso 2: **Agrupaciones y obtención de la función mínima**

- **Grupo 1:** $g = a$ (seleccionando la columna $a=1$ donde $g=1$)
- **Grupo 2:** $g = c \cdot d$ (para agrupar las celdas donde solo depende de c y d)

Resultado final: Función mínima simplificada:

$$g(a, b, c, d) = a + (c \cdot d)$$

Razonad la corrección de la solución y, en caso de ser incorrecta, indicad en cada paso los errores detectados, pero no hace falta obtener la solución correcta.

La solución aportada es incorrecta por varias razones:

El propio mapa de Karnaugh es incorrecto, ya que la función g no vale 1 para el mintermino $abcd$ (valor 1111).

Además, los dos grupos dados por la IA generativa son incorrectos:

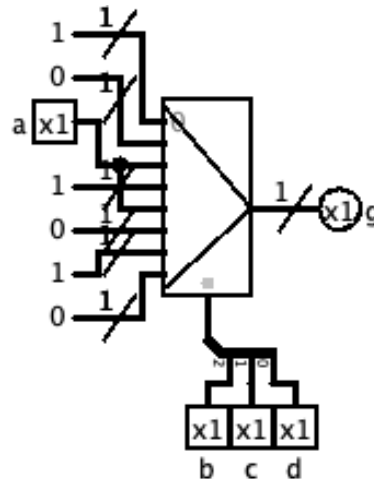
- $g = a$: para poder eliminar 3 variables hace falta un grupo de 8 1's que no existe. g es 0 para los minterminos $ab'c'd$ (1001) y $abc'd$ (1101) donde tendría que ser 1 para completar el grupo de 8 1s necesario.
- $g = cd$: en este caso haría falta que en la columna donde c y d son 1 la función valiera 1 en todos los casos y g es 0 para el mintermino $a'bcd$ (0111), y recordemos que para el mintermino $abcd$ (1111) la salida es 0.



- c) [7,5%] Sintetizad la función g con un multiplexor de 3 variables de control y el menor número de puertas lógicas. (Sección 3.1. Multiplexor. Multiplexor de buses. Demultiplexor)

Seleccionando como entradas de control las variables b , c y d , las entradas del multiplexor son constantes (0 o 1) o tiene el mismo valor que la variable a . Así que en este caso no hace falta ninguna puerta lógica extra.

El circuito queda de la siguiente forma:



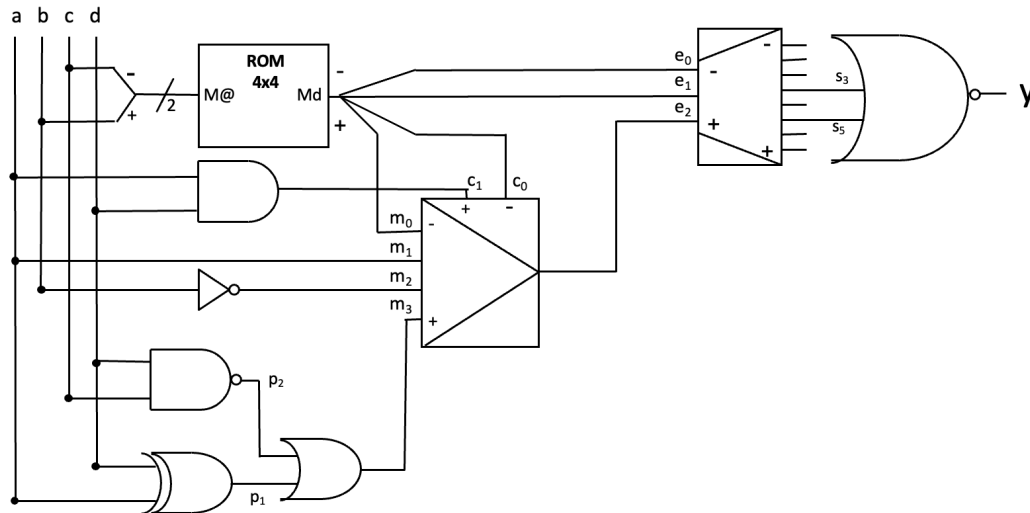
Tened en cuenta que existen diferentes soluciones dependiendo de las entradas que se seleccionan como entradas de control del multiplexor.

NOTA: Tenéis disponible el ejercicio en VerilCIRC y en KeMAP para verificarlo. Os recomendamos hacer pruebas con KeMAP con otros ejercicios antes de hacer el de la PEC.



Ejercicio 3 [25 %]

Dado el circuito lógico combinacional siguiente:



Donde el contenido de la memoria ROM es el siguiente:

M@	Md
0	0000
1	0111
2	1100
3	1001

Completad la tabla de verdad que especifica la salida y en función de las entradas a , b , c y d . Calculad previamente los valores intermedios (p_2 , p_1 , m_3 , m_2 , m_1 , m_0 , c_1 , c_0 , e_2 , e_1 , e_0 , s_5 , s_3) indicados en el circuito y añadidos en la tabla de verdad siguiente (Sección 3. Bloques combinacionales):

Para completar la tabla, necesitamos saber cómo se calcula cada una de las señales intermedias:

- p_2 se calcula como $(cd)'$ y p_1 como $a \oplus d$
- La dirección de la ROM viene dada por las señales b y c como bits más y menos significativo, respectivamente.
- Las señales de control del multiplexor son $c_1 = ad$ y $c_0 = Md[2]$, mientras que sus entradas son $m_0 = Md[3]$, $m_1 = a$, $m_2 = b'$ y $m_3 = p_1 + p_2$.
- Las entradas del decodificador son $e_0 = Md[0]$, $e_1 = Md[1]$ y e_2 que es la salida del multiplexor.
- Finalmente, la salida $y = (s_3 + s_5)'$.

Teniendo en cuenta esta información, podemos completar la tabla de verdad.

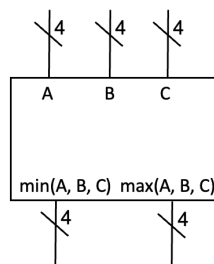


a	b	c	d	p ₂	p ₁	m ₃	m ₂	m ₁	m ₀	c ₁	c ₀	e ₂	e ₁	e ₀	s ₅	s ₃	y
0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1
0	0	1	0	1	0	1	1	0	0	0	1	0	1	1	0	1	0
0	0	1	1	0	1	1	1	0	0	0	1	0	1	1	0	1	0
0	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	0	0	1	0	1	0	0	0	0	0	1
0	1	1	0	1	0	1	0	0	1	0	0	1	0	1	1	0	0
0	1	1	1	0	1	1	0	0	1	0	0	1	0	1	1	0	0
1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1
1	0	0	1	1	0	1	1	1	0	1	0	1	0	0	0	0	1
1	0	1	0	1	1	1	1	1	0	0	1	1	1	1	0	0	1
1	0	1	1	0	0	0	1	1	0	1	1	0	1	1	0	1	0
1	1	0	0	1	1	1	0	1	1	0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	0	1	1	1	1	1	0	0	0	0	1
1	1	1	0	1	1	1	0	1	1	0	0	1	0	1	1	0	0
1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	0	0	1

NOTA: Tenéis disponible el ejercicio en VerilChart para verificarlo.

Ejercicio 4 [30 %]

Se desea diseñar un circuito que, dados 3 números de 4 bits en complemento a 2, muestre el menor de los tres valores en la salida $\min(A,B,C)$ y el mayor de los tres valores en la salida $\max(A,B,C)$ siguiendo la siguiente estructura (Sección 3. Bloques combinacionales):



- a) [20%] Diseñad un circuito que, dados dos números A y B de 4 bits en complemento a 2, obtenga el menor y el mayor de los dos operandos a partir de bloques combinacionales (excepto ROMs) y las puertas que consideréis necesarias. Llamad a las salidas del circuito como $\min2$ para $\min(A,B)$ y $\max2$ para $\max(A,B)$.

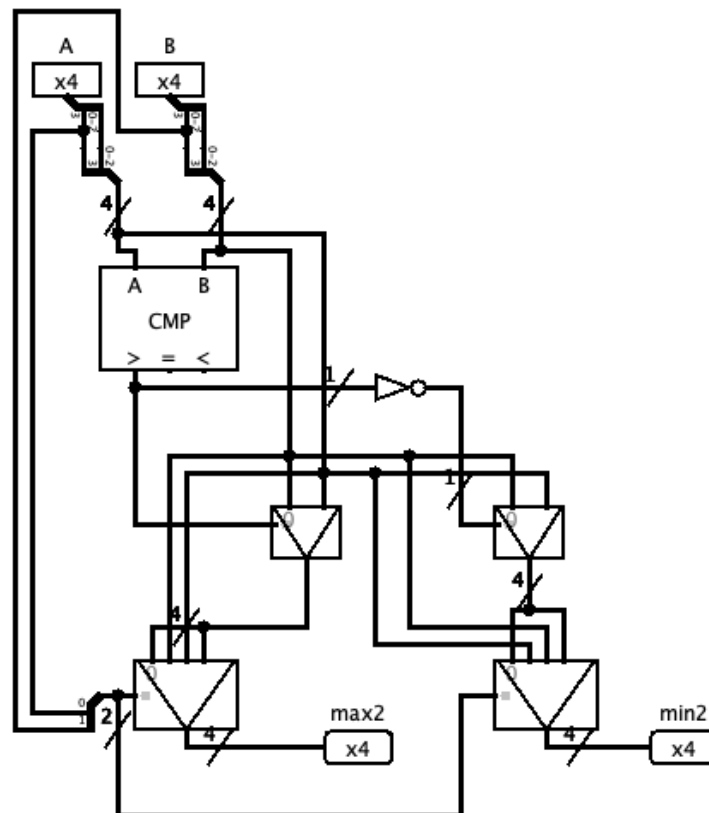
Una posible solución para este ejercicio es dividir el problema del máximo y mínimo según el valor del bit de signo de los operandos A y B en CA2.



- Si los números tienen signo contrario, el máximo será el positivo y el mínimo el negativo.
- Si los números tienen el mismo signo, hay que compararlos usando un comparador, la salida del comparador nos permite seleccionar el más grande y el más pequeño de los dos. Independientemente de que sean positivos o negativos, la salida $A > B$ siempre indica el valor mayor.

Los bits más significativos de los dos números se usan para seleccionar el resultado en función del caso en que estamos (signos iguales o diferentes) y de si buscamos el máximo o el mínimo.

El circuito queda de la siguiente forma:

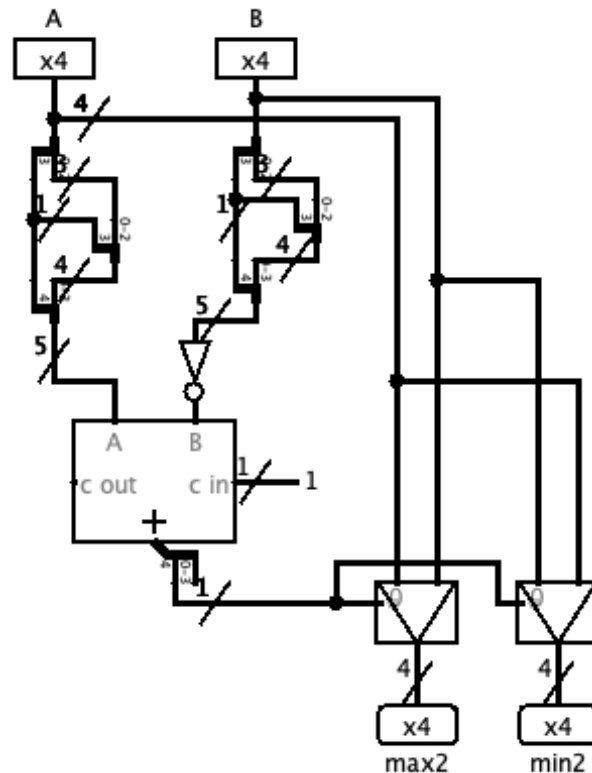


También se puede implementar el circuito haciendo una operación de resta de los dos números. Recordemos que la resta de dos números en CA2, se realiza sumando A al resultado de invertir los bits de B y sumando 1 al resultado final, es decir, $A - B = A + B' + 1$.

Necesitaremos extender el signo para evitar que se produzca desbordamiento. Entonces usamos el bit más significativo del resultado para seleccionar el máximo y el mínimo. Si el bit es 0, el resultado es positivo y $A \geq B$ (max2 es A y min2 es B). Si el bit es 1 el resultado es negativo y $A < B$ (max2 es B y min2 es A).



En este caso el circuito queda de la siguiente forma:

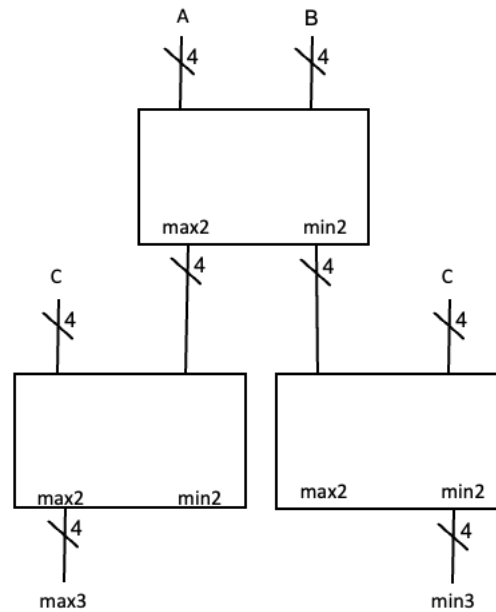


- b) [10%] Utilizando el circuito del apartado a), bloques combinacionales (excepto ROMs) y las puertas que consideréis necesarias, diseñad el circuito $\min(A,B,C)$ y $\max(A,B,C)$. Llamad a las salidas del circuito como $\min3$ y $\max3$ respectivamente.

Con un circuito como el diseñado en el apartado a) calculamos el máximo y el mínimo entre A y B . La salida $\min2$ la conectamos a un segundo circuito del mismo tipo y la otra entrada la conectamos al número C . La salida $\min2$ de este segundo circuito nos dice el mínimo de los 3 números, así que será $\min3$.

Actuamos de forma análoga para la salida $\max2$, conectándola a un tercer circuito del mismo tipo y, de nuevo, la otra entrada de estos circuitos la conectamos al número C . La salida $\max2$ de este circuito nos dice el máximo de los tres números, así que será $\max3$.

Esquemáticamente, el circuito será como el siguiente:



NOTA: Tenéis disponible el ejercicio en VerilCIRC. No hay límite en el número de intentos que uséis para comprobar vuestra solución.

NOTA2: En el caso de diseñar el circuito en VerilCIRC, la herramienta no admite utilizar bloques diseñados por el usuario. Por lo tanto, en caso de reutilizar un circuito, como por ejemplo el circuito *del apartado a)*, tendréis que copiar el circuito tantas veces como necesitéis utilizarlo.