

Arquitectura de Sistemas Audiovisuales I

Tema 3. Ejercicios Resueltos de Bloques Combinacionales

Ejercicio 1. Conversor BCD

Realizar un circuito que a partir de un número binario puro de 4 bits obtenga el número BCD equivalente. El circuito se debe realizar mediante bloques combinacionales y no mediante mapas de Karnaugh o sólo puertas lógicas. El resultado será un número BCD que representa dos cifras decimales (8 bits en total).

Para realizar el circuito se pueden usar los siguientes bloques combinacionales: decodificadores, codificadores, comparadores, multiplexores, demultiplexores y sumadores de cualquier número de bits. Así como puertas lógicas. Cualquier otro bloque se deberá realizar en función de los anteriores o en puertas lógicas. Para cada bloque utilizado se debe especificar claramente qué bloque es, las señales de entrada y salida, y el ancho de bus.

Solución:

Primero estudiamos la relación entre los dos sistemas de numeración, el binario y el BCD, para números binarios de 4 bits (rango de 0 a 15). Esta relación se muestra a continuación:

decimal	binario	BCD	
		decenas	unidades
0	0000	0000	0000
1	0001	0000	0001
2	0010	0000	0010
3	0011	0000	0011
4	0100	0000	0100
5	0101	0000	0101
6	0110	0000	0110
7	0111	0000	0111
8	1000	0000	1000
9	1001	0000	1001
10	1010	0001	0000
11	1011	0001	0001
12	1100	0001	0010
13	1101	0001	0011
14	1110	0001	0100
15	1111	0001	0101

Número binario menor o igual a 9:
- unidades igual que en binario,
- decenas igual a 0000

Número binario mayor que 9:
- unidades: número binario
menos 10
- decenas igual a 0001

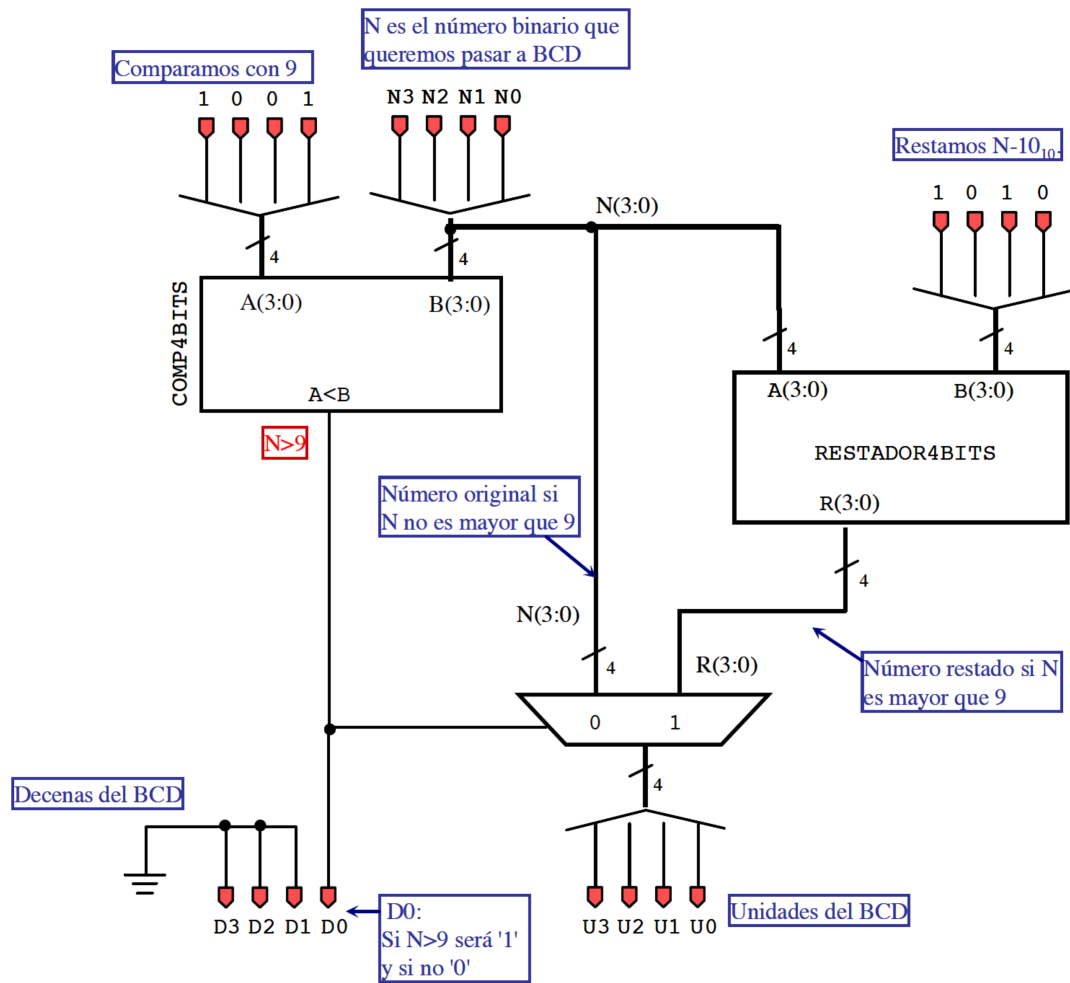
Por ejemplo, si tenemos el número 14,
le restamos 10, y obtenemos las unidades

$$\begin{array}{r} 1110 \\ -1010 \\ \hline 0100 \end{array} \quad \begin{array}{r} 14 \\ -10 \\ \hline 4 \end{array}$$

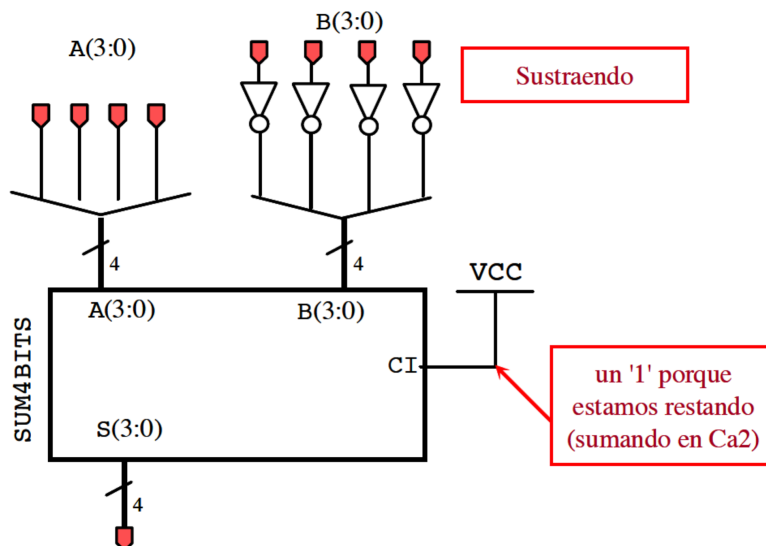
Nos fijamos que hay dos grupos, los números menores o iguales a 9, y los mayores que 9.

Por lo tanto, si el número binario es menor o igual a nueve, dejamos la unidades igual que el número binario y las decenas a cero.

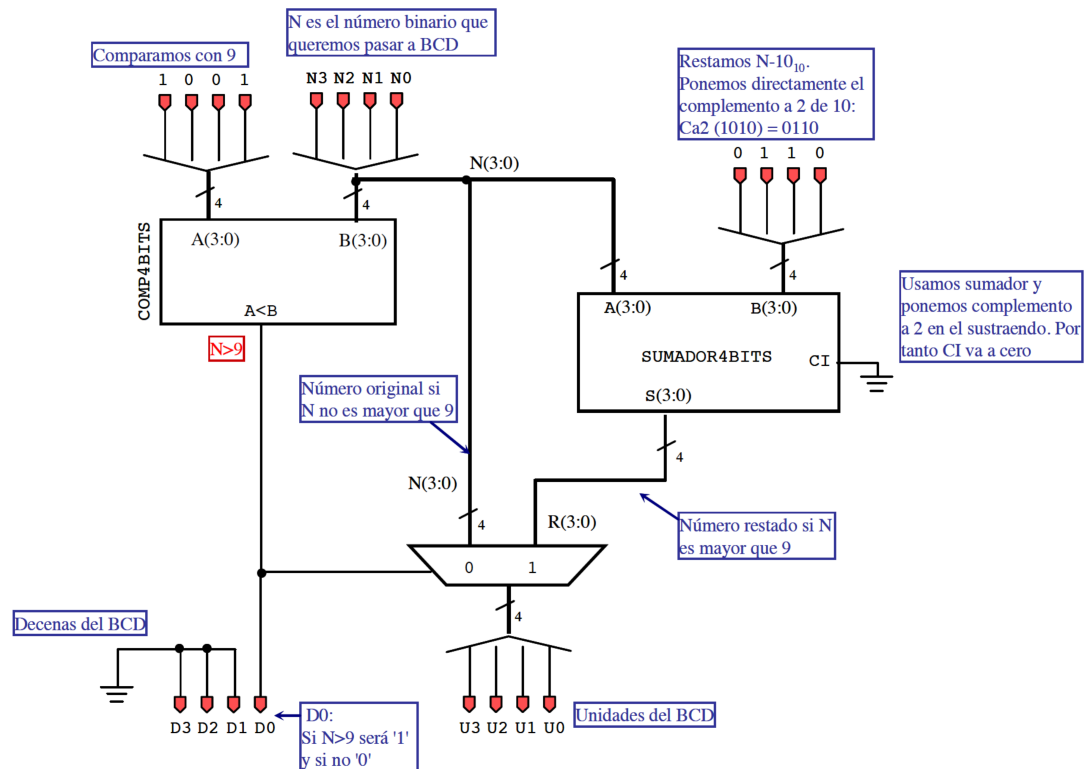
Y si el número binario es mayor a nueve, dejamos la unidades igual que el número binario y las decenas a cero.



Sólo nos quedaría especificar cómo se hace el restador de 4 bits:

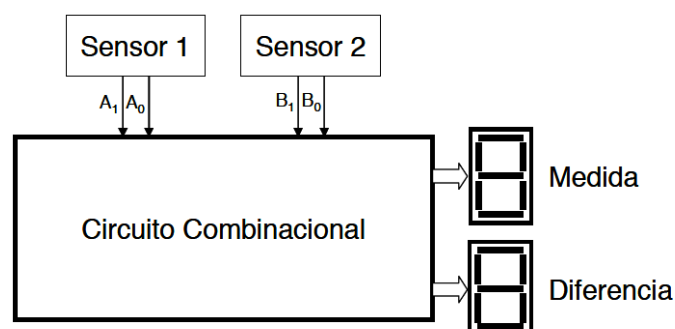


Por último, como curiosidad, fíjate que como el sustraendo es un número fijo (10102), el circuito se podría optimizar un poco, usando directamente un sumador habiendo hecho previamente el complemento a 2 de 10102, que es 0110. Por tanto es como sumar 6 y desprejar el acarreo. Lo que es otra manera de hacerlo, ya que por ejemplo, $13+6=19$ que en binario es 100112. Tomando los 4 primeros bits nos da el 0011 que es el tres de las unidades del 13. El circuito quedará:

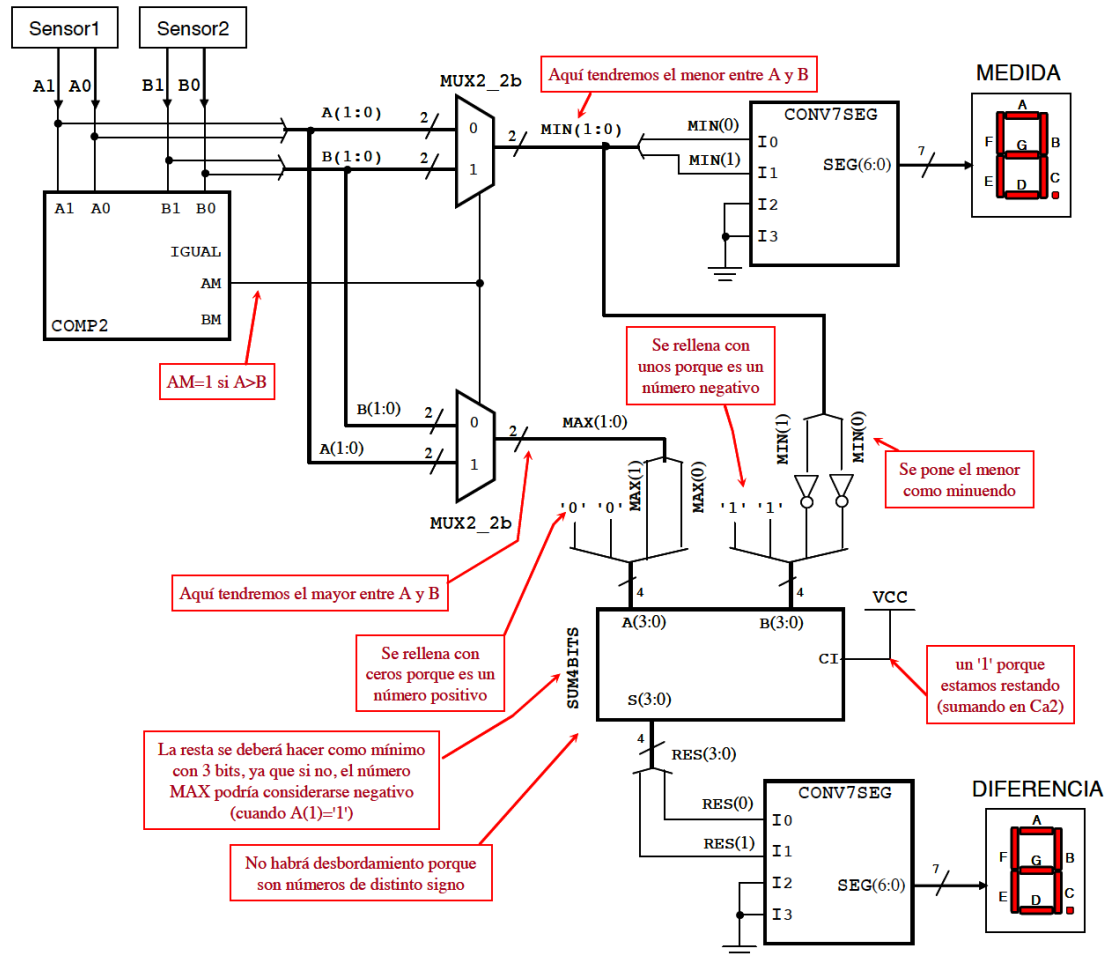


Ejercicio 2. Báscula de precisión

Una báscula utiliza dos sensores. Estos sensores son idénticos y están midiendo la misma pieza, aunque su medida puede diferir. Las dos medidas se entregan a un sistema digital codificadas cada una con 2 bits ($A_1 A_0$ para el sensor 1 y $B_1 B_0$ para el sensor 2). Para visualizar la medida se dispone de dos displays de 7 segmentos. En uno de ellos se visualizará la diferencia entre los sensores y en el otro la menor de las medidas en el caso de que no coincidan las lecturas de los dos sensores. Diseñar mediante bloques combinacionales estándares (multiplexores, decodificadores, sumadores, comparadores...) el esquema del circuito combinacional.



Solución:



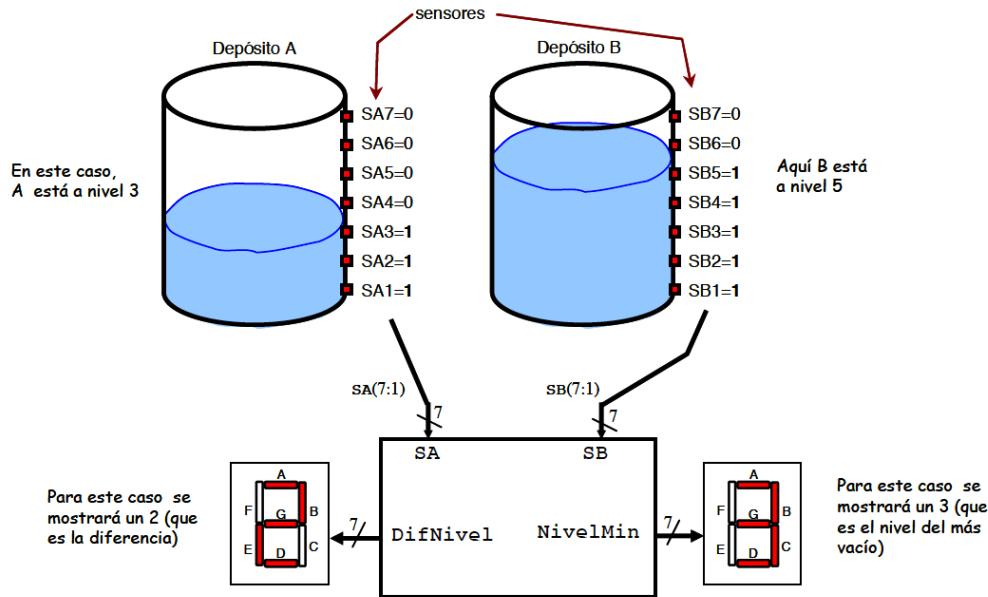
Ejercicio 3. Nivel de depósitos de agua

Se tienen dos depósitos de agua de los que se quiere conocer en cada momento el nivel de agua del que esté más vacío, y la diferencia de nivel respecto al mayor (en valor absoluto).

Para conocer el nivel, cada depósito tiene 7 sensores distribuidos a lo alto del depósito. Cada sensor dará un '1' lógico si está cubierto de agua, y un '0' lógico si está al aire.

La salida se dará mediante dos displays de 7 segmentos, uno para indicar el nivel del más vacío, y otro para indicar la diferencia.

Como se tienen 7 sensores para conocer el nivel, el rango de valores va desde 0 a 7. Se supone que ningún sensor va a fallar, por lo tanto, si un sensor indica un '1' lógico, todos los sensores que estén debajo de él darán un '1' lógico (pues el agua los cubrirá también).



Se pide realizar el diagrama de bloques del circuito:

Para el diseño se podrá emplear cualquiera de los siguientes bloques sin necesidad de describirlos en puertas (no todos son necesarios): multiplexores, sumadores, codificadores, decodificadores, demultiplexores, comparadores y decodificadores de 7 segmentos. Todos ellos de uno o varios bits. Además se podrá emplear cualquier tipo de puertas lógicas. Cualquier otro bloque deberá ser descrito en función de los bloques citados o en puertas.

Es muy importante indicar todos los nombres de cada señal y su índice (o peso lógico).

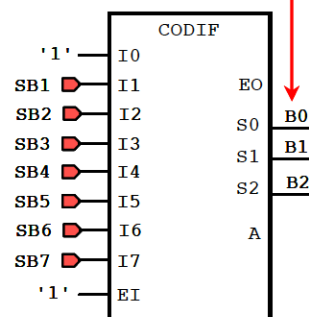
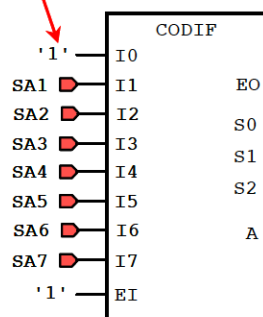
Se valorará la reducción de componentes del circuito y la sencillez.

Solución:

Para empezar, una buena opción será codificar las señales de entrada (las que provienen de los sensores). Ya que tenemos 7 señales, y como el rango va de 0 a 7, las podemos codificar en 3 bits. Necesitamos por tanto un codificador con prioridad de 8 a 3 para cada depósito.

Lo ponemos a '1', ya que si los sensores están a '0', la salida será cero: S(2:0)=0

Niveles de A y B codificados en binario puro 3 bits



Esto se puede hacer directamente, aunque si tienes dudas, lo puedes comprobar con la tabla de verdad.

EI	I7	I6	I5	I4	I3	I2	I1	I0	S2	S1	S0	A	EO	Nivel
0	X	X	X	X	X	X	X	X	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	0	0	0	1	1	0	0	1	1	0	1
1	0	0	0	0	0	1	1	1	0	1	0	1	0	2
1	0	0	0	0	1	1	1	1	0	1	1	1	0	3
1	0	0	0	1	1	1	1	1	1	0	0	1	0	4
1	0	0	1	1	1	1	1	1	1	0	1	1	0	5
1	0	1	1	1	1	1	1	1	1	1	0	1	0	6
1	1	1	1	1	1	1	1	1	1	1	0	1	0	7
1	el resto de combinaciones son imposibles								X	X	X	X	X	X

Estos '1' en realidad en el codificador pueden ser X, pero en nuestro caso no es posible.

De la tabla de verdad se puede comprobar que para el nivel 0, la entrada I0 vale '0'. Debido a que cuando un sensor está en contacto con el agua todos los sensores que están debajo también lo están, se rellenan con '1' y no con X (se muestra en un triángulo). El resto de combinaciones no son posibles. Una vez codificadas las señales de los depósitos, el resto del circuito se puede hacer como sigue (similar al problema de la báscula):

Ejercicio 4. Unidad Aritmético-Lógica

Realizar una unidad aritmético lógica con dos operandos de 4 bits (OPA y OPB) y 4 operaciones.

Las operaciones que se realizarán son: suma, resta, complemento a 2 (Ca2) y complemento a 1 (Ca1). Para seleccionar una operación se dispone de 4 pulsadores:

- BTN0: realiza la suma $S=OPA+OPB$
- BTN1: realiza la resta. $S=OPA-OPB$
- BTN2: devuelve el complemento a 2 de OPB
- BTN3: devuelve el complemento a 1 de OPB.

Siendo BTN0 la operación de mayor prioridad, y BTN3 la de menor (para el caso en que se pulsen más de un botón a la vez). Por ejemplo, si se pulsan a la vez BTN1 y BTN2 el circuito devolverá la resta y no el Ca2.

Como salida se tendrá la señal RESULT de 4 bits.

a) Realizar el diagrama de bloques del circuito:

Para el diseño se podrá emplear cualquiera de los siguientes bloques sin necesidad de describirlos en puertas (no todos son necesarios): multiplexores, sumadores, codificadores, decodificadores, demultiplexores y comparadores. Todos ellos de uno o varios bits. Además se podrá emplear cualquier tipo de puertas lógicas.

Cualquier otro bloque deberá ser descrito en función de los bloques citados o en puertas.

b) Incluir la señal de desbordamiento, sabiendo que:

Para el caso de la suma se considera que los operandos están en binario puro.

Para la resta y el Ca2, los operandos de entrada están codificados en Ca2

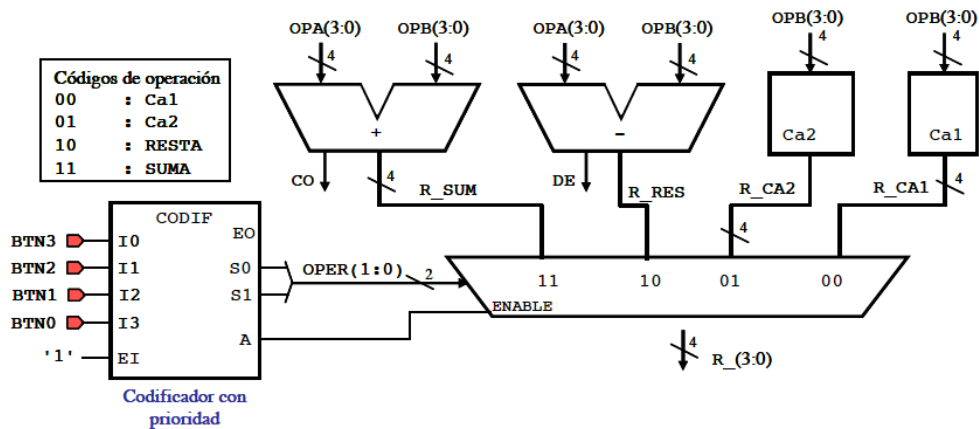
Para el Ca1 el operando de entrada está codificado en Ca1.

Se valorará la reducción de componentes del circuito y la sencillez.

SOLUCIÓN:

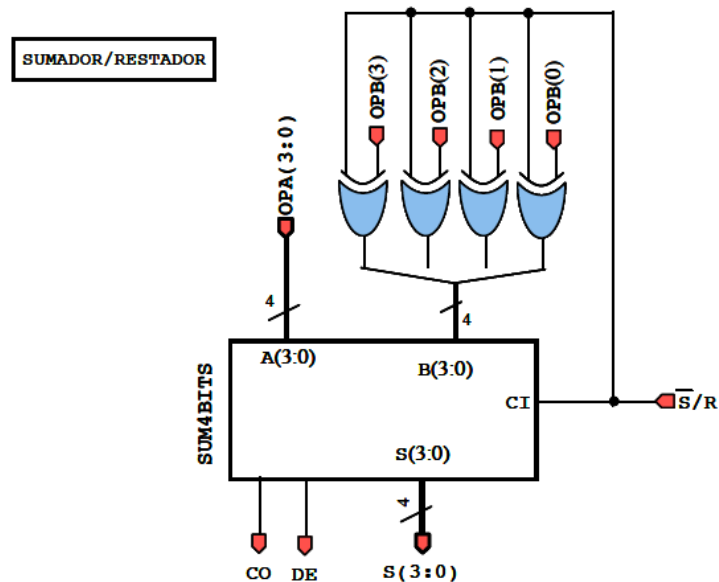
Hay muchas posibles soluciones para este diseño.

Inicialmente el circuito podría diseñarse de la siguiente manera.

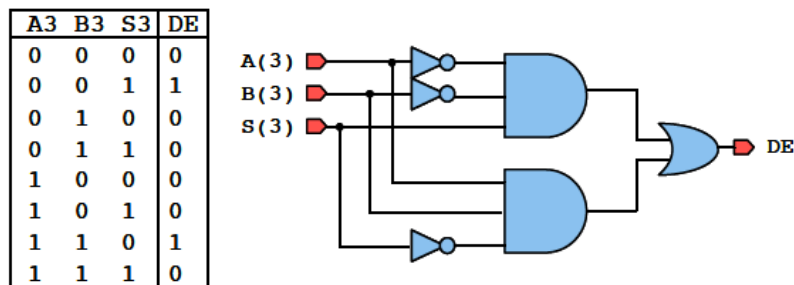


Según el enunciado, no disponemos del restador y de los bloques que realizan el Ca2 y Ca1.

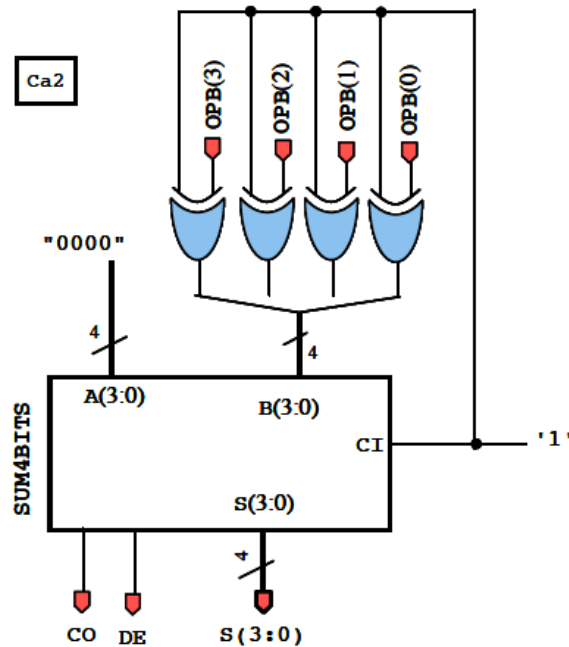
El restador se puede hacer con un sumador, haciendo el complemento a dos del minuendo. Y que como nunca tendremos que hacer una suma y resta a la vez, podemos realizar un bloque sumador/restador:



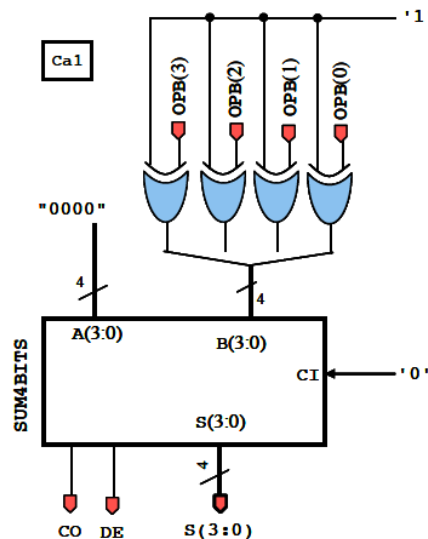
La señal DE corresponde al desbordamiento de la resta (hecha como suma) en complemento a dos, y se puede calcular de distintas maneras, una de ellas es (condición de desbordamiento en la suma en Ca2):



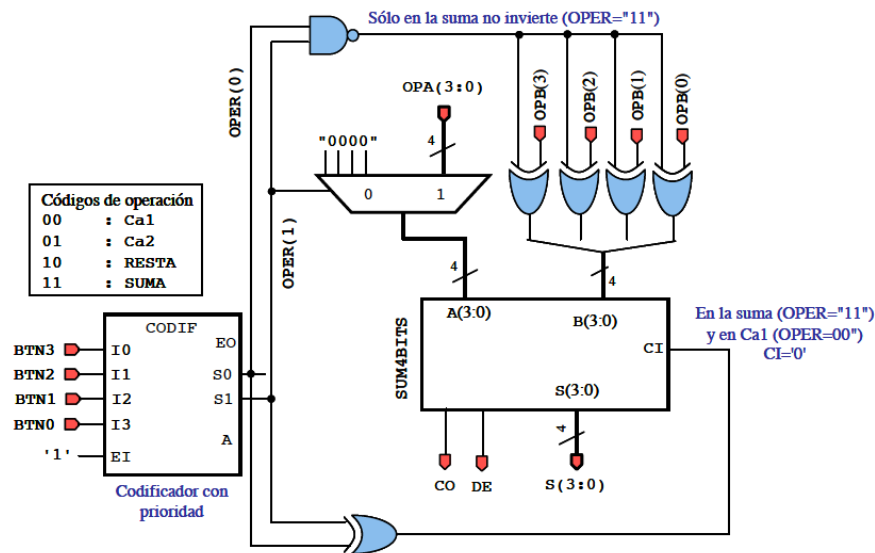
Hacer el bloque que calcula el complemento a dos de B, va a ser sencillo, simplemente tenemos que invertir B y añadirle uno. Como necesitamos añadir uno, podemos utilizar el mismo bloque sumador. De hecho podemos hacer igual que si fuésemos a realizar una resta, pero haciendo que OPA sea igual a cero.



Y por último, podemos hacer el complemento a uno simplemente invirtiendo B. Para ello podemos usar un bloque a parte, o utilizar lo mismo que hemos usado, sólo que no añadimos el acarreo de entrada.



Evidentemente esta una forma ineficiente de hacer el complemento a uno si sólo necesitásemos realizar esta operación, pero al tener que realizar sumas y restas, nos quedará un diseño muy compacto.



b) Desbordamiento:

Y por último queda la señal de desbordamiento. En la suma se selecciona el acarreo de salida (CO), ya que están en binario puro; en la resta y Ca2 se selecciona DE, y en Ca1 cualquiera de ellas ya que siempre será cero (porque se le suma 0). Esta señal se quedaría así.

