



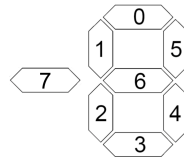
IIC2343 - Arquitectura de Computadores (II/2025)

Guía de Ejercicios: Compuertas Lógicas

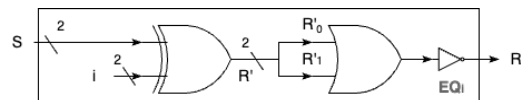
Ayudantes: Daniela Ríos (danielaarp@uc.cl), Alberto Maturana (alberto.maturana@uc.cl), Fernanda Escobar (ferjez2002@uc.cl)

Pregunta 1: I1-2023-2 (Pregunta 2)

Se busca implementar el *display* de un ascensor cuyos pisos van del -2 al 1 (siendo 0 el piso base). Construya, a partir de compuertas lógicas, el controlador de este *display*. Este recibe como *input* una señal de 2 bits que representa un número entero con signo equivalente al número del piso del ascensor. Puede basarse en la figura para señalar los segmentos del *display* que deben encenderse (1) o apagarse (0) para cada *input*.



Solución: Para construir este circuito, en primer lugar definimos el componente EQ_i cuya salida es 1 si S es igual al valor i :



Por ejemplo, en el caso de EQ_{-2} , su salida será 1 si S es -2.

Usaremos este componente con los valores $i = -2, -1, 0, 1$ y veremos las condiciones para que

se encienda cada segmento del *display*:

$$\text{Seg}_0 = \text{EQ}_{-2} \vee \text{EQ}_0$$

$$\text{Seg}_1 = \text{EQ}_0$$

$$\text{Seg}_2 = \text{EQ}_{-2} \vee \text{EQ}_0$$

$$\text{Seg}_3 = \text{EQ}_{-2} \vee \text{EQ}_0$$

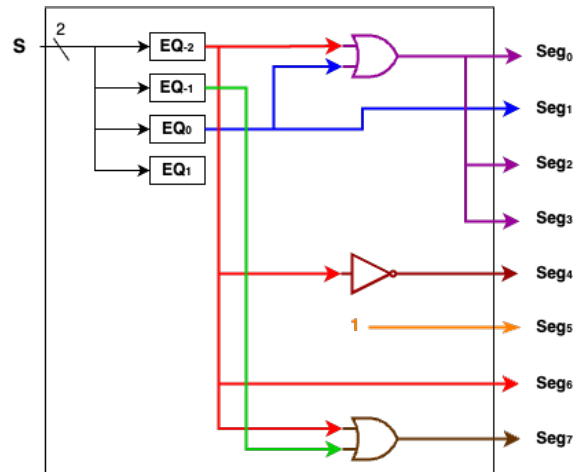
$$\text{Seg}_4 = \text{EQ}_{-1} \vee \text{EQ}_0 \vee \text{EQ}_1 \equiv \neg \text{EQ}_{-2}$$

$$\text{Seg}_5 = \text{EQ}_{-2} \vee \text{EQ}_{-1} \vee \text{EQ}_0 \vee \text{EQ}_1 \equiv 1$$

$$\text{Seg}_6 = \text{EQ}_{-2}$$

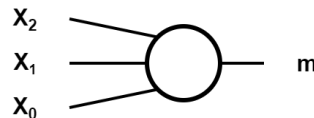
$$\text{Seg}_7 = \text{EQ}_{-2} \vee \text{EQ}_{-1}$$

Por lo tanto, el diagrama interno del controlador se construye de la siguiente forma:



Pregunta 2: I1-2024-1 (Pregunta 2)

Definiremos una nueva compuerta lógica llamada **Mayoría**. Esta recibirá tres señales de entrada de 1 bit y su salida, de 1 bit, corresponderá al valor que más se repite. Por ejemplo, si tenemos $X_0 = 0$, $X_1 = 0$, $X_2 = 1$, la salida de la compuerta **Mayoría** es 0.



A partir de esta compuerta, responda los incisos (a) y (b).

- (a) Diseñe la compuerta lógica **Mayoria**, de 3 entradas, a partir de las siguientes compuertas: AND, OR, NOT, XOR, NAND o NOR.

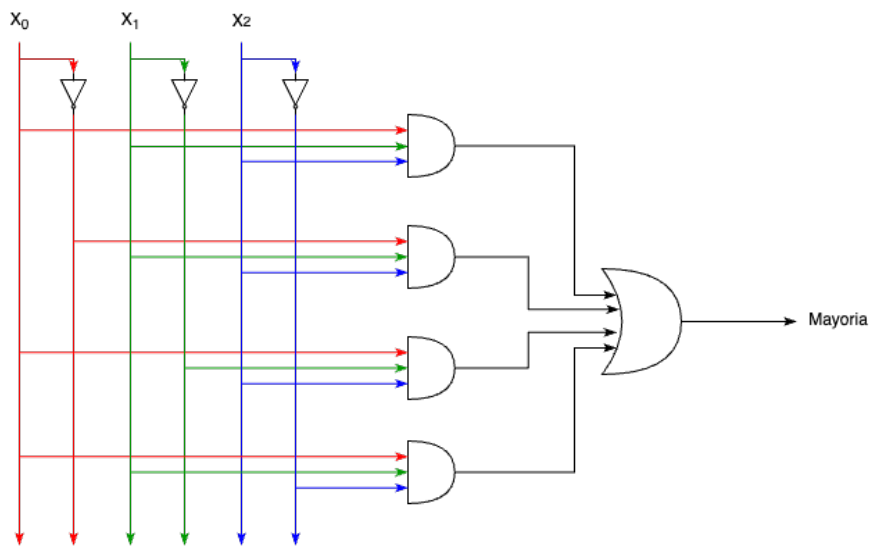
Solución: Si tenemos tres entradas X_0 , X_1 y X_2 solo pueden darse dos casos:

- Que el valor que más se repite esté en las tres entradas (3 a 0).
- Que el valor que más se repite esté en dos de las tres entradas (2 a 1).

De esta manera, usamos *minterms* o *maxterms* (*minterms* en este caso) para obtener la expresión que representa la salida.

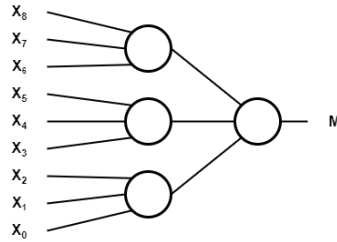
$$m = X_0X_1X_2 + \overline{X_0}X_1X_2 + X_0\overline{X_1}X_2 + X_0X_1\overline{X_2}$$

Finalmente, diseñamos el circuito a partir de esta expresión.



Es importante aclarar que se evaluará la correctitud del circuito diseñado, no que sea exactamente igual al incluido en la pauta.

- (b) Haciendo uso de la compuerta **Mayoria**, se propone el siguiente circuito para determinar el valor que más se repite entre 9 *inputs*:



Indique si el circuito propuesto resuelve correctamente el problema planteado. Si lo resuelve, justifique por qué; si no lo hace, indique en qué casos no funciona.

Solución: Hay solo dos casos que analizar:

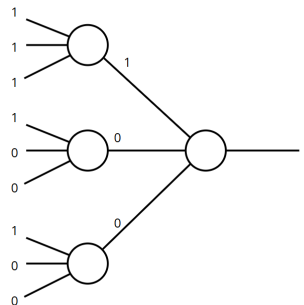
- Si la mayoría fue de 6 a 3.
- Si la mayoría fue de 5 a 4.

Si fue de **6 a 3**, el circuito propuesto determina correctamente el valor que más se repite. Para comprobarlo, hay que ver cómo pueden dividirse los 6 valores de mayoría entre las tres compuertas de la primera “fase” del circuito; solo hay tres posibilidades:

- 3, 3, 0
- 2, 2, 2
- 3, 2, 1

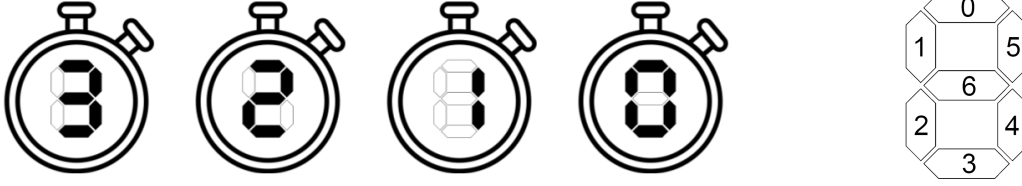
En los tres casos, la cuarta compuerta (la segunda “fase”) siempre produce el resultado correcto.

Si fue de **5 a 4**, el circuito propuesto puede equivocarse. Por ejemplo: si los *inputs* se dividen en 3-0, 1-2 y 1-2, el circuito dice que la opción minoritaria, de 4 valores, es la mayoritaria.



Pregunta 3: I1-2024-2 (Pregunta 2)

Pronto se llevarán a cabo las DCCarreras de sacos dieciocheras y, para ahorrar recursos, le piden a usted que diseñe un *timer* para marcar el inicio de cada carrera. A continuación, una imagen de referencia del *timer* esperado y del *display* de 7 segmentos que tendrá integrado para desplegar el número:



Para construir el *timer* diseñe, para cada segmento S_i del *display*, un circuito con una señal de entrada de 2 bits $I_1 I_0$ y una señal de salida de 1 bit. La entrada corresponderá al número a desplegar en el *timer* (3, 2, 1 o 0), mientras que la salida indica si el segmento S_i se prende (1) o no (0) según el número que se busca desplegar.

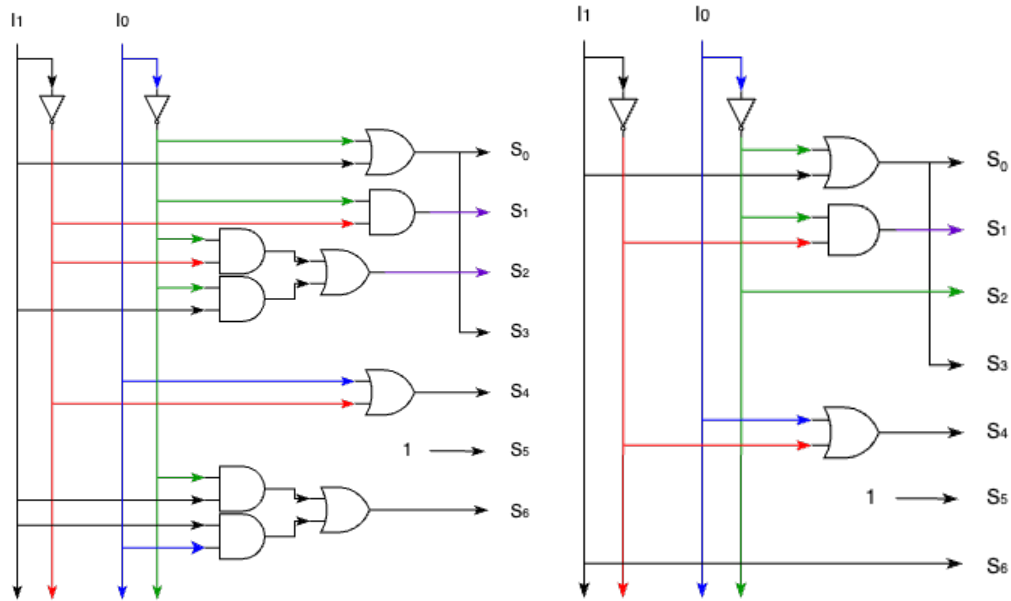
Solución: A continuación, la tabla de verdad para los segmentos S_i a partir de $I_1 I_0$:

I_1	I_0	S_0	S_1	S_2	S_3	S_4	S_5	S_6
0	0	1	1	1	1	1	1	0
0	1	0	0	0	0	1	1	0
1	0	1	0	1	1	0	1	1
1	1	1	0	0	1	1	1	1

A partir de ella, se puede hacer uso de *minterms* y *maxterms* para obtener la expresión que representa cada salida. A continuación, se listan expresiones válidas para cada segmento, junto con su reducción a través de equivalencias lógicas.

- $S_0 = S_3 = I_1 \vee \overline{I_0}$
- $S_1 = \overline{I_1} \wedge \overline{I_0}$
- $S_2 = (\overline{I_1} \wedge \overline{I_0}) \vee (I_1 \wedge \overline{I_0}) = \overline{I_0} \wedge (I_1 \vee \overline{I_1}) = \overline{I_0} \wedge 1 = \overline{I_0}$
- $S_4 = \overline{I_1} \vee I_0$
- $S_5 = 1$
- $S_6 = (I_1 \wedge \overline{I_0}) \vee (I_1 \wedge I_0) = I_1 \wedge (\overline{I_0} \vee I_0) = I_1 \wedge 1 = I_1$

Finalmente, se puede diseñar un circuito a partir de estas expresiones. Se incluye un circuito para las primeras expresiones obtenidas con *minterms* y *maxterms* (izquierda), y otro reducido a partir de equivalencias lógicas (derecha).



Es importante aclarar que se evaluará la correctitud del circuito diseñado, no que sea exactamente igual a los incluidos la pauta.

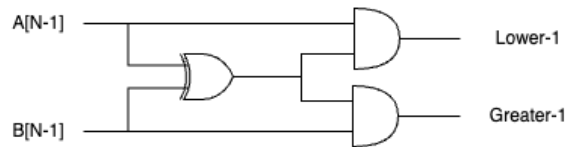
Pregunta 4: T1-2023-1 (Pregunta 2)

Un comparador de números es un circuito que, dados dos números A y B en representación posicional, indica cuál es el mayor, o si estos son iguales. El circuito posee tres salidas, donde la primera entrega un 1 solo si A es el mayor, la segunda un 1 solo si ambos son iguales, y la tercera un 1 solo si B es mayor. Haciendo uso de las compuertas lógicas vistas en clases, diseñe un comparador de números **enteros** de N bits, explicando la funcionalidad de cada uno de los circuitos que elabore.

Solución: Las tres salidas de nuestro circuito se llamarán *Greater*, *Equal* y *Lower* según lo descrito en el enunciado. Luego, para facilitar la construcción de nuestro circuito, vemos que existen cuatro casos a evaluar:

- **Caso 1:** $A \geq 0, B \geq 0$
- **Caso 2:** $A < 0, B < 0$
- **Caso 3:** $A \geq 0, B < 0$
- **Caso 4:** $A < 0, B \geq 0$

Los casos 3 y 4 se pueden resolver rápidamente haciendo uso del **bit de signo**: Verificamos que A y B posean signo distinto a través de una compuerta **XOR** (cuya salida es 1 solo si ambas señales son distintas) y este resultado lo conectamos con compuertas **AND** que nos ayudarán a determinar si se cumple que la condición *Greater-1* o *Lower-1* (primer caso para identificar si el resultado es mayor o menor):



Se observa en el circuito que si $A_{N-1} = 0, B_{N-1} = 1$, entonces la señal *Greater-1* estará activa y la señal *Lower-1* no, mientras que se dará el caso contrario para $A_{N-1} = 1, B_{N-1} = 0$. Si $A_{N-1} = B_{N-1}$ ninguna de las señales se activará, independiente del signo de estos números.

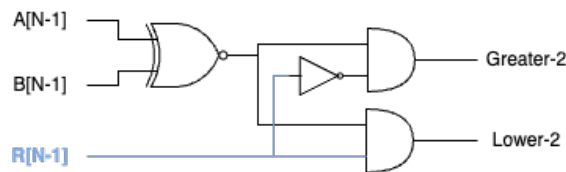
Para resolver los casos 1 y 2, haremos uso de un restador de N bits para obtener un nuevo bus de N bits R :



R nos sirve por las siguientes observaciones:

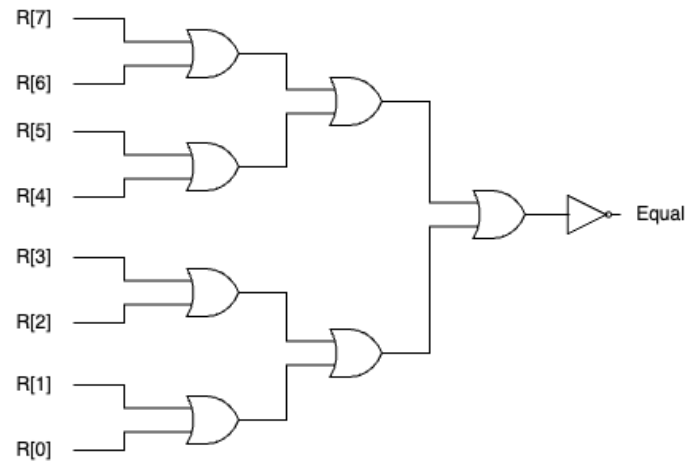
- $A - B > 0 \rightarrow A > B \rightarrow (A - B)_{N-1} = R_{N-1} = 0$
- $A - B < 0 \rightarrow A < B \rightarrow (A - B)_{N-1} = R_{N-1} = 1$

Notamos entonces que el bit más significativo de R nos indica si $A > B$ o $A < B$, lo que podemos verificar con el siguiente circuito:



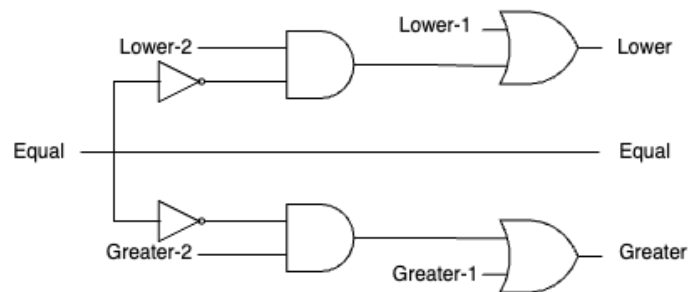
En contraste con el circuito anterior, aquí hacemos uso de una compuerta **XNOR** para verificar que los signos de A y B sean **iguales**. Observamos entonces que si $R_{N-1} = 0$, entonces se activará la señal *Greater-2* que indica que $A > B$ y, en contraste, si $R_{N-1} = 1$ entonces se activará la señal *Lower-2* que indica que $A < B$ (siempre que se cumpla $A \text{ XNOR } B = 1$ para ambos casos).

El caso anterior no está del todo completo, dado que no estamos considerando la posibilidad donde $A = B$. Para ello, definimos un nuevo circuito que verifique que **todos los bits de R sean iguales a cero**:



En este caso se muestra para $N = 8$ pero su extensión a cualquier N es trivial. El circuito activa la señal *Equal* si, y solo si **ninguna** compuerta **OR** entrega como resultado 1 (que se da solo si alguno de los bits de R es igual a 1).

Usamos este último resultado para entregar la señal faltante *Equal* y para que los casos *Greater-2* y *Lower-2* sean válidos solo si $Equal = 0$:



De esta forma, *Greater*, *Lower* y *Equal* se activan correctamente según los casos señalados al comienzo.