

Practica 1

Arquitectura y organización de computadores 2



Javier mengod y álvaro fraidias

Índice

[Resolución de la práctica. 2](#_Toc65508528)

[Ejercicio 2. 2](#_Toc65508529)

[Ejercicio 3. 3](#_Toc65508530)

[Ejercicio 4. 5](#_Toc65508531)

[Ejercicio 5. 6](#_Toc65508532)

[Ejercicio 6. 7](#_Toc65508533)

[Ejercicio 7. 8](#_Toc65508534)

[Ejercicio 8. 9](#_Toc65508535)

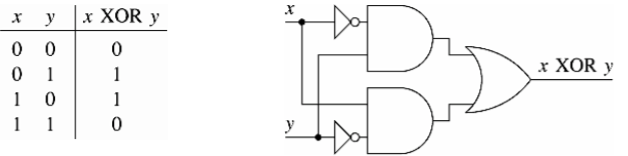
[Bibliografía. 13](#_Toc65508536)



# Resolución de la práctica.

## Ejercicio 2.

Implementar el ejemplo que aparece en el manual, es decir, implementar una puerta XOR y comprobar su funcionamiento.



En Logisim quedaría el siguiente circuito:

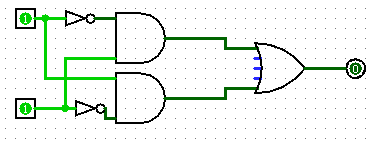


Ilustración : Puerta XOR en Logisim

En el que se puede observar que se cumple la tabla de la verdad XOR.

## Ejercicio 3.

3.- Implementar en Logisim la siguiente expresión Booleana:



Una vez implementada, simplificad el circuito.

Una vez dentro de logisim, al clicar sobre “proyecto” y después, sobre “analizar circuito” nos abrirá la pestaña de análisis combinacional. Primero tendremos que introducir las entradas y posteriormente la salida.

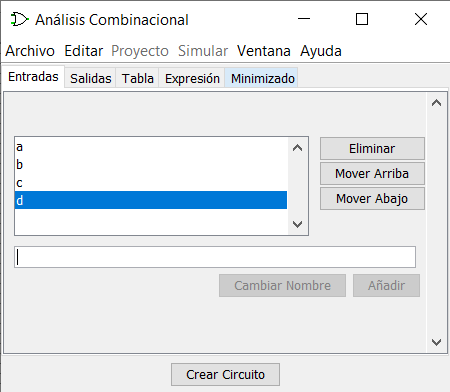


Ilustración : Ventana de Análisis Combinacional en Logisim

Posteriormente habrá que añadirle la expresión correspondiente a la función booleana que se nos pide. Quedaría de la siguiente manera:

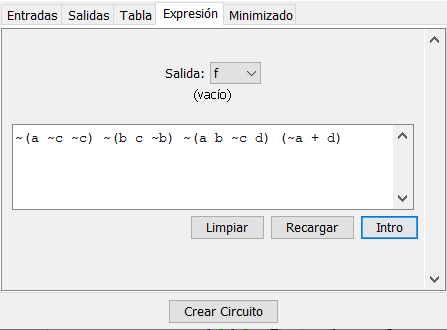


Ilustración : Ventana de Expresión booleana

Logisim nos generará el siguiente circuito:

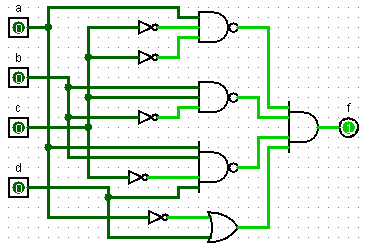


Ilustración : Circuito sin minimizar del ejercicio 3.

Para minimizarlo iremos a la pestaña de minimizado y fijaremos la expresión para minimizarlo. El siguiente circuito es el equivalente minimizado:

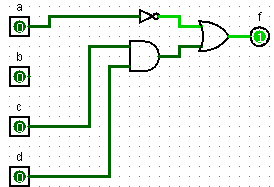


Ilustración : Circuito minimizado del ejercicio 3.

## Ejercicio 4.

Dada la siguiente tabla de verdad, implementad el circuito en Logisim:

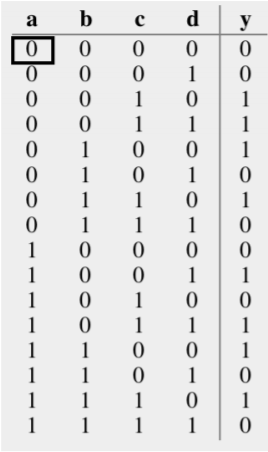


Ilustración 6: Tabla de verdad

De un modo parecido al ejercicio anterior, iremos a la pestaña de “análisis combinacional” y añadiremos las entradas y salidas, pero esta vez en vez de poner una expresión booleana iremos al apartado de “tabla” y colocaremos a mano los datos y crearemos finalmente el circuito.

El circuito resultante es el siguiente:

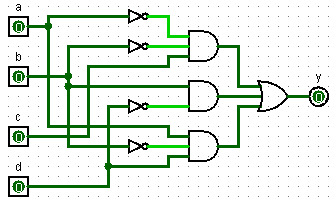


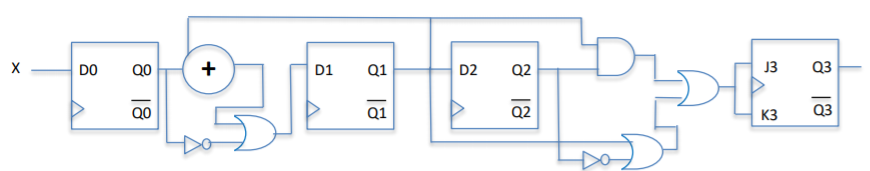
Ilustración : Circuito del ejercicio 4

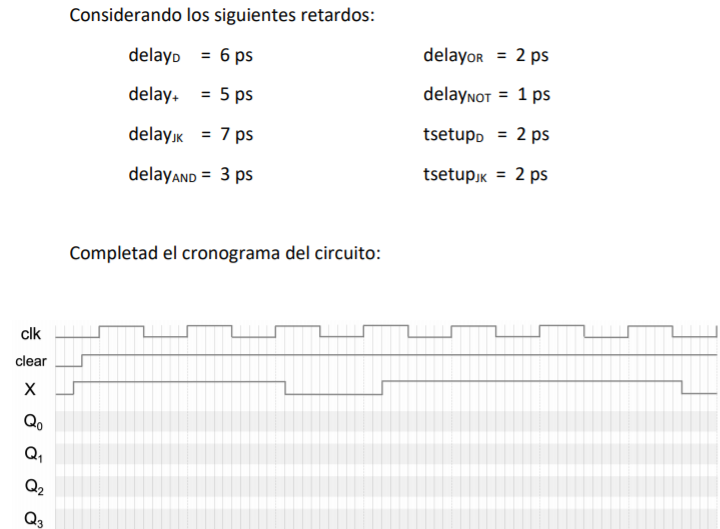
Para minimizarlo, haremos los mismos pasos que en el ejercicio anterior:

Iremos a la pestaña de minimizado, fijaremos la expresión y lo minimizaremos. Sin embargo, podemos fijarnos que la expresión booleana simplificada es idéntica que la que tenemos en el circuito, por tanto, ya está simplificado y no cambiará nada.

## Ejercicio 5.

5.- A partir de los flip-flops D y JK disponibles en la biblioteca proporcionada por Logisim, implementar el siguiente circuito y comprobar su correcto funcionamiento.





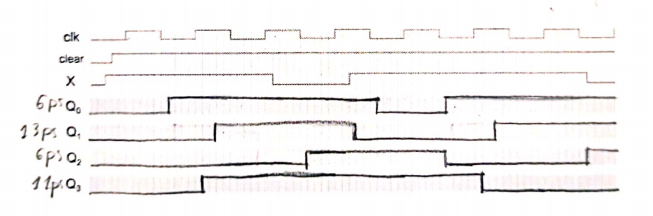


Ilustración 8: Cronograma ejercicio 5

* El retardo de Q0 es de 6ps ya que la señal solo debe atravesar el biestable, cuyo retardo es de 6ps.
* El retardo de Q1 es de 13ps ya que el camino crítico atraviesa el sumador (5ps), la puerta or (2ps) y el retardo del biestable (6ps).
* El retardo de Q2 es igual al retardo de Q0 (6ps) ya que solo atraviesa el biestable.
* El retardo de Q3 es de 11ps ya que la señal debe atravesar la puerta not (1ps), dos puertas or (4ps) y el retardo del biestable (6ps).

Este circuito en logisim quedaría de la siguiente forma:

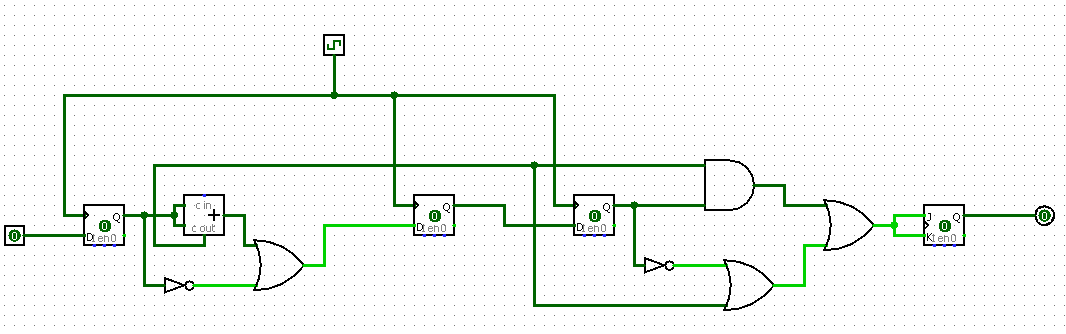


Ilustración 9: Circuito ejercicio 5 representado en logisim

Finalmente, calculad y mostrad el tiempo de todos los posibles caminos, y teniendo en cuenta esto, indicad cuál sería el tiempo de ciclo del circuito.

El camino crítico es entre el biestable Q0 y Q1 con un tiempo de 15 ps (delay biestable D + retardo sumador + retardo or + tiempo setup biestable D).

## Ejercicio 6.

Implementar el siguiente circuito y comprobar su funcionamiento. ¿Qué hace exactamente? ¿Os recuerda a algo conocido?

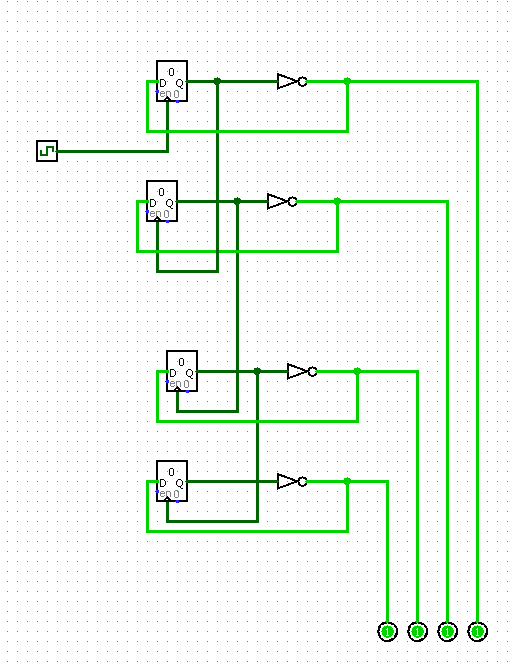
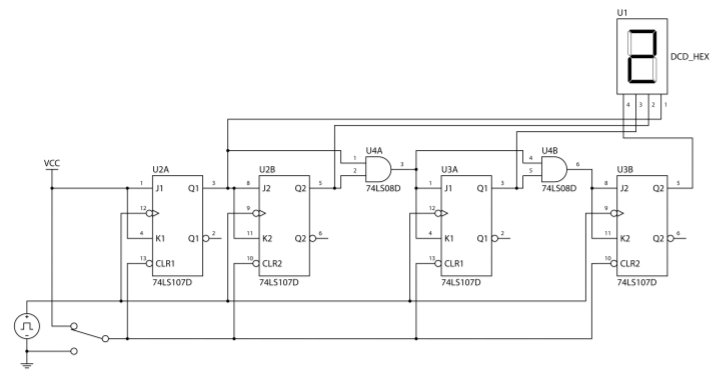


Ilustración 10: Implementación el logisim del ejercicio 6

Al implementarlo y ejecutarlo nos damos cuenta que se comporta como una tabla de verdad de 4 bits.

## Ejercicio 7.

Usando la biblioteca de componentes (es decir, los biestables y el display de 7 segmentos que ya os proporciona Logisim), implementad el siguiente circuito:



La implementación en logisim quedaría de la siguiente manera:

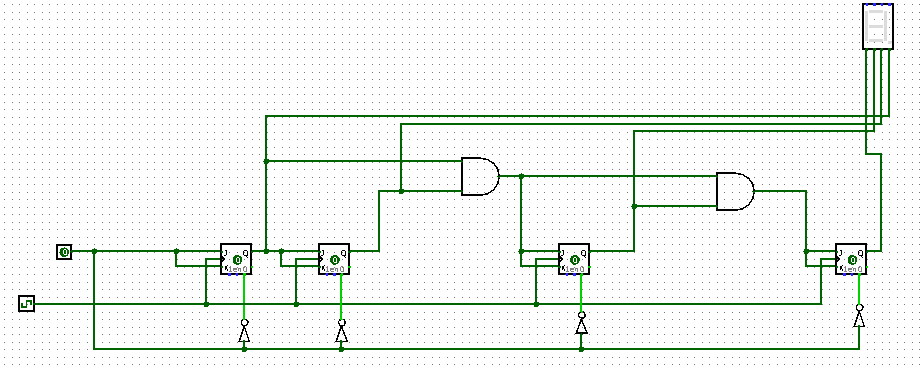


Ilustración 11: Implementación del ejercicio 7 en logisim

Nuestro problema a la hora de implementarlo es el display de 4 segmentos ya que en logisim solo hay un display de 7 segmentos.

## Ejercicio 8.

8.- Implementar una ALU que realice las siguientes operaciones: and, or, xor y suma, con valores de 4 bits.

Para implementar esta ALU debemos implementar un mecanismo que ejecute las 4 operaciones que nos piden y nos permita seleccionar una de las 4 operaciones. El circuito es el siguiente:

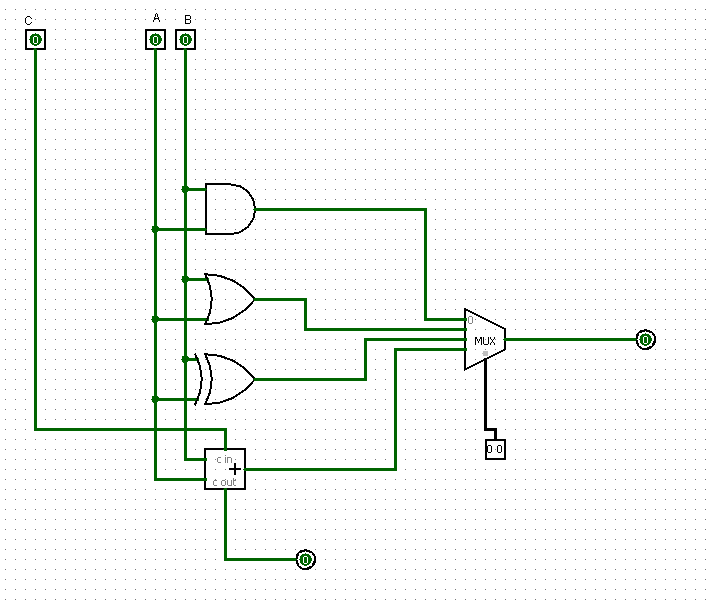


Ilustración 12: ALU

Una vez creado esto debemos encapsularlo e implementar una por cada bit del número que queremos sumar. Una ALU de 4 bits quedaría de la siguiente manera:

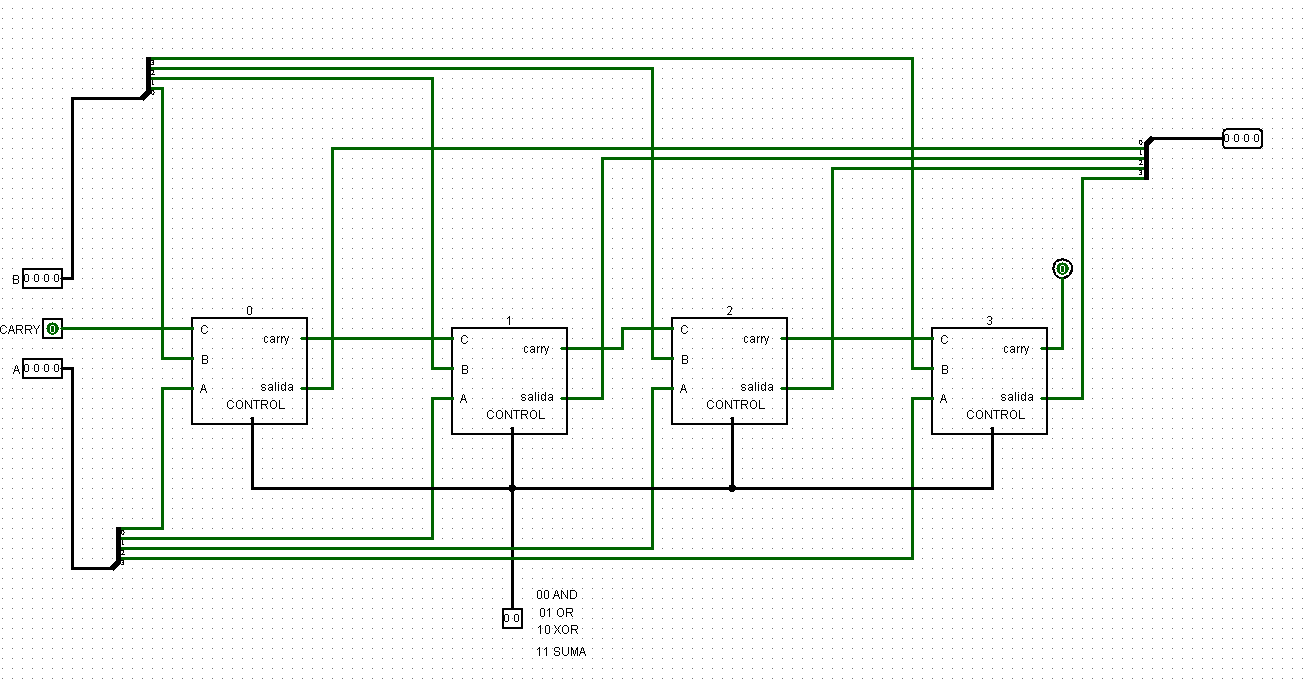


Ilustración : ALU de 4 bits

Finalmente, encapsulamos este circuito en un chip y usamos 8 de ellos interconectados para hacer una ALU de 32 bits.:

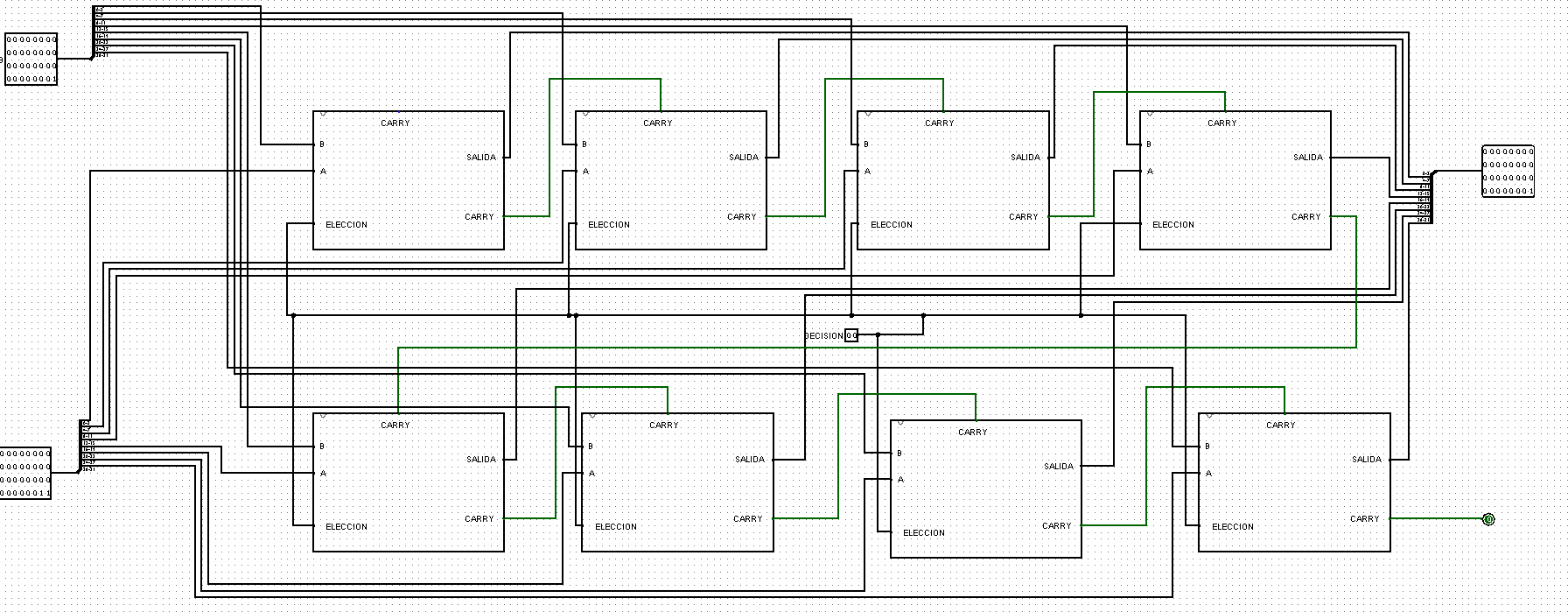
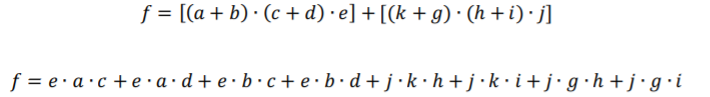


Ilustración : ALU 32 bits

Ejercicio 9.

En este ejercicio, queremos que reflexionéis sobre el coste de la implementación y la temporización de los circuitos. Para ello, será necesario que implementéis, usando Logisim y en dos ficheros distintos, los circuitos que cumplen las siguientes ecuaciones:



A continuación, debéis simplificar ambos circuitos. ¿Son equivalentes? En caso afirmativo, ¿cuál de los dos usaríais suponiendo que todas las puertas tienen un retardo de 8ps? Justificad vuestra respuesta.

Realizando el mismo procedimiento que en el ejercicio 3, obtenemos el siguiente circuito para la función 1:

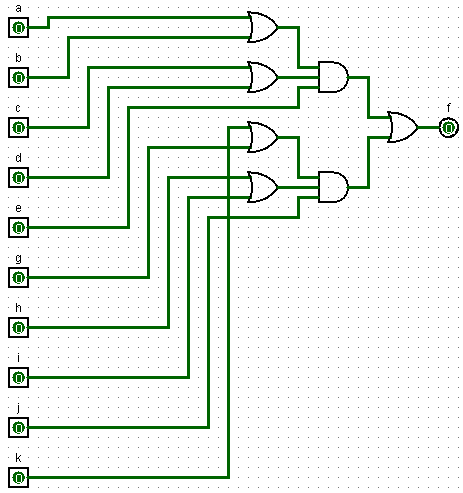


Ilustración : Circuito sin simplificar de la función 1.

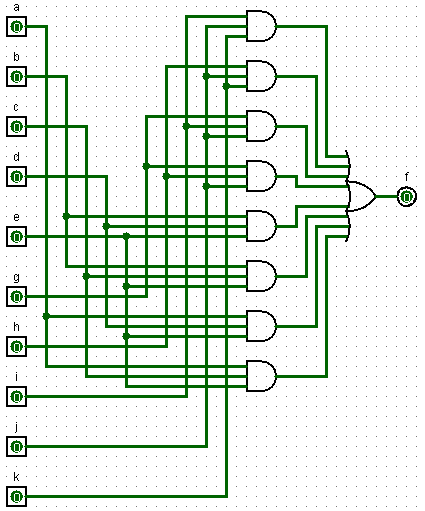
Simplificando obtenemos el siguiente circuito:  


Ilustración : Circuito simplificado de la función 1.

Para la función dos realizamos los mismos pasos. El primer circuito obtenido es el siguiente:

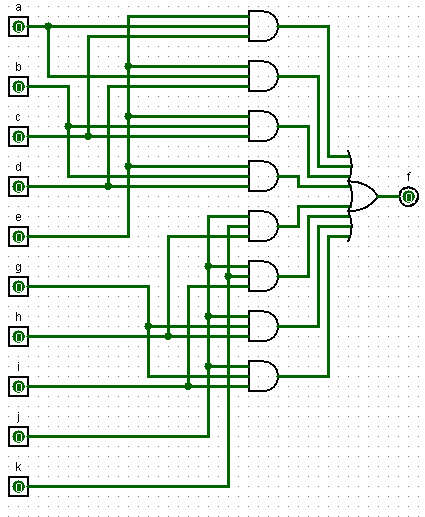


Ilustración : Circuito sin simplificar de la función 2.

Simplificando obtenemos el siguiente:

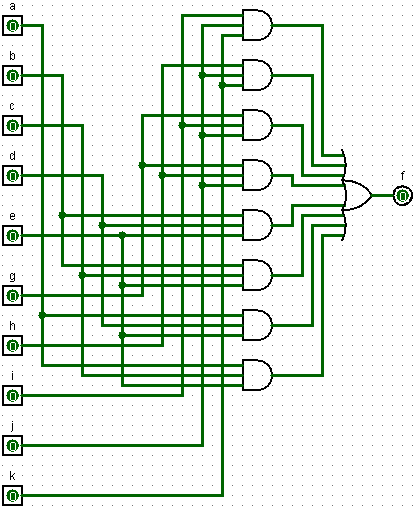


Ilustración : Circuito simplificado de la función 2.

El cual podemos observar que es el mismo. La expresión booleana simplificada de ambos por tanto también es la misma.

Si los circuitos están sin simplificar y se tiene que escoger uno de los dos y el tiempo de retardo de todas las puertas es de 8 ps la opción más eficiente sería la segunda, ya que tiene un nivel de puertas menor y por tanto cualquier entrada pasará por una puerta menos y tardará 8 ps menos que el otro circuito. Si están simplificados la decisión de escoger uno u otro evidentemente es irrelevante ya que son el mismo circuito.

# Bibliografía.

Apuntes de la asignatura en Moodle.

Estructura y diseño de computadores. D.A. Patterson y J.I. Hennessy editorial Reverté