

Práctica 2 - Lógica Digital

Sistemas Digitales

Primer Cuatrimestre 2025

Todas las compuertas mencionadas en esta práctica son de 1 ó 2 entradas, a menos que se indique lo contrario. Usaremos los símbolos detallados a continuación para representar las distintas funciones lógicas: **XOR** $\rightarrow \oplus$, **NAND** $\rightarrow \downarrow$, **NOR** $\rightarrow \downarrow$.

Durante la presente práctica se recomienda fuertemente la utilización de un simulador para experimentar con los componentes y circuitos propuestos y verificar las soluciones. Una recomendación es el Logisim Evolution.

Circuitos Combinatorios

Ejercicio 1 Demostrar si las siguientes equivalencias de fórmulas booleanas son verdaderas o falsas:

a) $x.z = (x + y).(x + \bar{y}).(\bar{x} + z)$

b) $x \oplus (y.z) = (x \oplus y).(x \oplus z)$ donde se aplica la propiedad distributiva con respecto a \oplus ¹

Ejercicio 2 Una *fórmula* del álgebra de Boole es:

- p, q, r, \dots , una variable booleana que puede tener valor 1 o 0,
- 1, la constante *verdadero*,
- 0, la constante *falso*,
- Si p y q son fórmulas, entonces $p + q$ (p OR q), $p.q$ (p AND q) y \bar{p} (la negación de p) son fórmulas.

¿Se pueden expresar todas las funciones totales² $f : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$ usando fórmulas del álgebra de Boole? Justificar.

Ejercicio 3 Determinar la veracidad o falsedad de las siguientes afirmaciones:

- a) Sea $p|q = \bar{p}.\bar{q}$ ¿Alcanza este único operador (**NAND**) para representar todas las funciones booleanas?
- b) Sea $p \downarrow q = \overline{p + q}$ ¿Alcanza este único operador (**NOR**) para representar todas las funciones booleanas?

Ejercicio 4 Dibujar circuitos que implementen las siguientes funciones booleanas:

- a) $f(A, B, C) = A.B.C$ usando 2 compuertas NOR y varias compuertas NOT.
- b) $f(A, B) = \overline{(A.B) + (\bar{B}.A)}.\bar{B}$ ¿Para qué valores de A y B la función devuelve un 1?
- c) $f(A, B, C, D) = ((C \downarrow D) \oplus (B + A)).(((A|B)|C)) + (\bar{D}.B)$

Ejercicio 5 Dadas las funciones booleanas F y G definidas a partir de las siguientes tablas de verdad:

¹ $p \oplus q = (\bar{p}.q) + (p.\bar{q})$.

²Una función total es aquella para la que todo elemento del dominio tiene imagen.

A	B	C	$F(A, B, C)$	D	E	F	$G(D, E, F)$
1	1	0	0	0	0	0	1
1	0	0	1	0	1	0	0
1	0	1	1	0	0	1	1
0	1	0	0	0	1	1	1
0	0	1	0	1	0	0	1
0	1	1	1	1	1	0	1
1	1	1	1	1	0	1	0
0	0	0	0	1	1	1	1

- a) Escribir la *suma de productos* para ambas funciones. Calcular la cantidad de compuertas que la implementación literal requeriría en cada caso.
- b) ¿Se pueden simplificar las expresiones usando propiedades del álgebra booleana? Para cada función decidir si es posible y, en caso de que lo sea, dibujar el circuito utilizando la menor cantidad de compuertas que pueda.

Ejercicio 6 Armar un circuito que invierta o no tres entradas de acuerdo al valor de una entrada adicional que actúa como control. En otras palabras, un inversor de k -bits es un circuito de $k + 1$ entradas (e_k, \dots, e_0) y k salidas (s_{k-1}, \dots, s_0) que funciona del siguiente modo:

- Si $e_k = 1$, entonces $s_i = \overline{e_i} \quad \forall i < k$
- Si $e_k = 0$, entonces $s_i = e_i \quad \forall i < k$

Ejemplo:

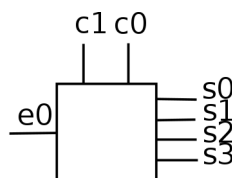
inversor(1,011)=100 inversor(0,011)=011
inversor(1,100)=011 inversor(1,101)=010

Ejercicio 7

- a) Diseñar un componente con 4 entradas e_0, \dots, e_3 y 4 salidas s_0, \dots, s_3 que calcule el inverso aditivo del número codificado en complemento a 2 por la entrada.
- b) Modificar el circuito anterior para que en una nueva salida indique si el número de la entrada no tiene un inverso aditivo representable con 4 *bits* en complemento a 2.

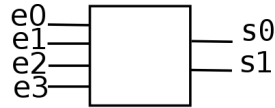
Ejercicio 8 Dibujar el diagrama lógico de un *demultiplexor* de 2 líneas de control, 1 línea de entrada y 4 líneas de salida. Este circuito dirige la única línea de entrada a una de cuatro líneas de salida, dependiendo del estado de las dos líneas de control.

c_1	c_0	s_i
0	0	$s_0 = e_0, s_i = 0$ si $i \neq 0$
0	1	$s_1 = e_0, s_i = 0$ si $i \neq 1$
1	0	$s_2 = e_0, s_i = 0$ si $i \neq 2$
1	1	$s_3 = e_0, s_i = 0$ si $i \neq 3$



Ejercicio 9

- a) Dibujar el diagrama lógico de un *codificador* de 4 líneas de entrada (e_i) y 2 líneas de salida (s_i). Si únicamente e_i está alta, las salidas deben representar el número i en notación sin signo. No está definido cuál es el resultado si no se cumple que sólo una de las líneas de entrada tiene valor 1.



- b) Dotar al circuito anterior de una salida adicional que indique si el estado de la entrada es válido o inválido.

Ejercicio 10

- a) Dibujar con compuertas lógicas el circuito de un *decodificador* de 2 líneas de entrada (e_i) y 4 líneas de salida (s_i), cuya tabla de verdad es la siguiente:

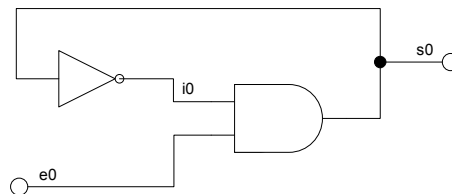
e_1	e_0	s_3	s_2	s_1	s_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

- b) Usando el circuito anterior, reescribir el *demultiplexor* de 1 línea de entrada, 2 líneas de control y 4 líneas de salida.

Circuitos Secuenciales

Ejercicio 11

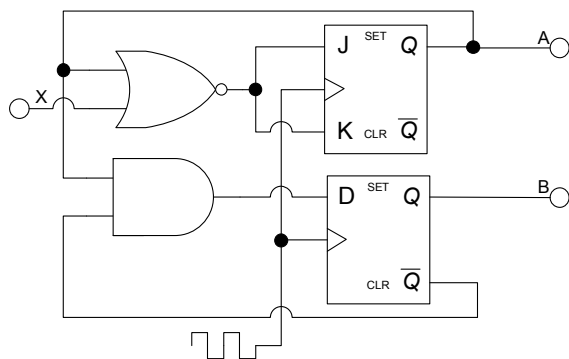
- a) Escribir el diagrama temporal para el siguiente circuito secuencial desde 0 ns hasta 65 ns, suponiendo
- un retardo de 15 ns para la compuerta AND,
 - un retardo de 5 ns para la compuerta NOT,
 - en el tiempo 0 ns la señal e_0 cambia a 1, inicialmente en 0.
 - las señales i_0 y s_0 tienen valor 1 y 0 respectivamente en el tiempo 0 ns.
 - suponer que los componentes empiezan a estabilizarse cuando sus señales de entrada están estables.



- b) ¿Podría alcanzar s_0 un valor estable en el punto anterior? ¿Y en el caso en que e_0 fuera 0 en lugar de 1, se estabilizaría?

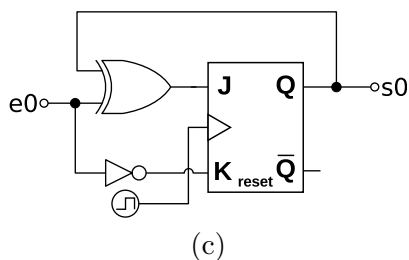
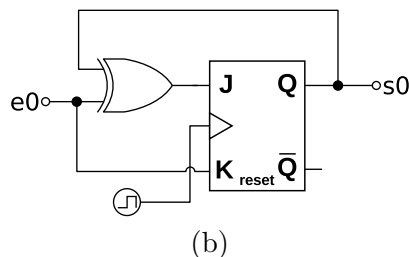
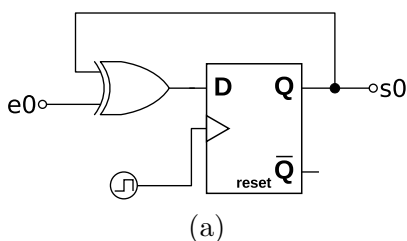
Ejercicio 12 ³ Completar la tabla característica para el siguiente circuito secuencial con los valores *estables* de la salida. Indicar si para alguna configuración el circuito no es estable:

³Ej 41. Capítulo 3 del L. Null & J. Lobur - Essentials Of Computer Organization And Architecture

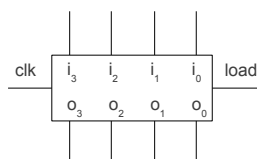


A	B	X	A'	B'
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Ejercicio 13 Escriba tablas características que especifiquen el comportamiento de cada uno de los siguientes circuitos secuenciales:

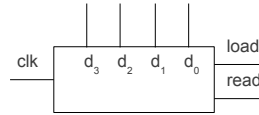


Ejercicio 14 Registro simple Diseñar un registro *simple* de cuatro *bits*. Este tipo de registros es un circuito de seis entradas (i_0 a i_3 , *load*, *clk*) y cuatro salidas (o_0 a o_3), cuyo funcionamiento es el siguiente: cuando la señal *clk* alcanza su flanco ascendente, si *load* está alta, almacena las señales recibidas en i_0 a i_3 , si no, no cambia su contenido. Por las líneas de salida, se emite el valor almacenado en el registro.

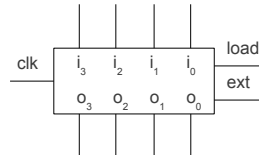


Ejercicio 15 Registro bidireccional Diseñar un registro *bidireccional* de cuatro bits. Este tipo de registros es un circuito con tres entradas (*load*, *read*, *clk*) y cuatro señales de entrada y salida (d_0 a d_3). Su funcionamiento es el siguiente: si la señal *load* vale 1 cuando *clk* alcanza su flanco ascendente, almacena los 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 líneas⁴. Las señales *read* y *load* nunca valen 1 simultáneamente.

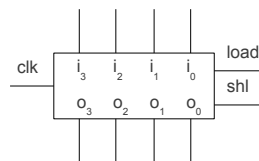
⁴Ayuda: utilice componentes de tres estados.



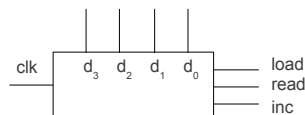
Ejercicio 16 Extensor de signo Diseñar un registro *extensor de signo* de dos a cuatro bits. Del mismo modo que un registro simple (ver ejercicio ??) este circuito toma el valor de sus cuatro entradas i_0 a i_3 cuando su señal clk atraviesa su flanco ascendente si la entrada $load$ vale 1. Por sus líneas de salida (o_0 a o_3) se emite el valor almacenado si la señal ext está baja; por el contrario, si vale 1, se emite una representación de cuatro *bits* del número almacenado en los dos *bits* menos significativos del registro, interpretados como un entero codificado en complemento a 2.



Ejercicio 17 Desplazador Diseñar un registro *desplazador a izquierda* de cuatro bits. Del mismo modo que un registro simple (ver ejercicio ??) este circuito toma el valor de sus cuatro entradas i_0 a i_3 cuando su señal clk atraviesa su flanco ascendente, si la entrada $load$ vale 1. Por sus líneas de salida (o_0 a o_3) se emite el valor almacenado si la señal shl está baja; por el contrario, si vale 1, se emite ese valor desplazado en uno hacia la izquierda.



Ejercicio 18 Registro Auto-incrementador Diseñar un registro *bi-direccional auto-incrementador* de cuatro bits. Este circuito funciona de la misma manera que un registro bi-direccional (ver ejercicio ??), pero tiene una línea de entrada extra inc . Si, cuando clk alcanza su flanco ascendente, inc vale 1, el valor almacenado por el registro se incrementa en uno. Nuevamente, sólo una de las tres líneas de control ($load$, $read$ e inc) puede valer 1 cuando clk alcanza su flanco ascendente.



Ejercicio 19

Dado el siguiente circuito, indique mediante un diagrama de tiempos la secuencia de activaciones y desactivaciones de señales de control necesarias para que el valor almacenado en el registro bidireccional (ejercicio ??) R0 se sume al valor del registro bidireccional R1 y el resultado se almacene en el registro R0.

