

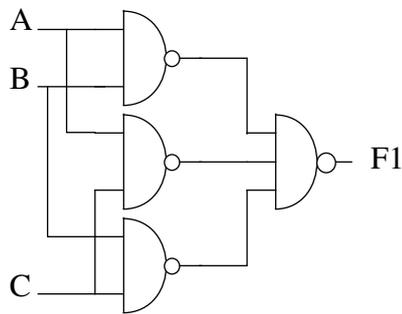
# Práctica I.- Introducción a los Sistemas Secuenciales

## 1.1.- Objetivos.

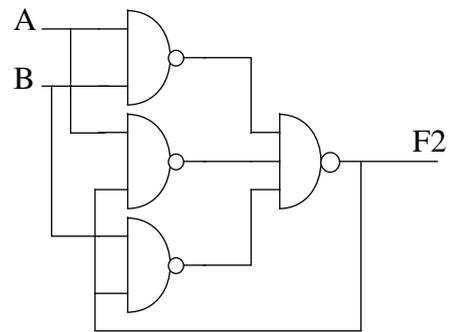
El objetivo de esta práctica consiste en consolidar las diferencias entre sistemas combinatoriales y secuenciales

## 1.2.- Problema.

Se disponen de los siguientes circuitos digitales:



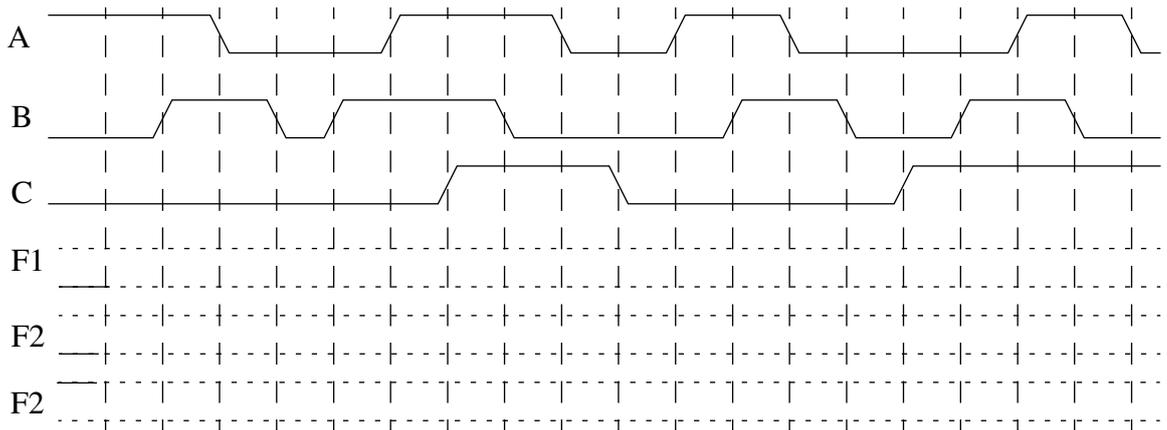
Circuito 1



Circuito 2

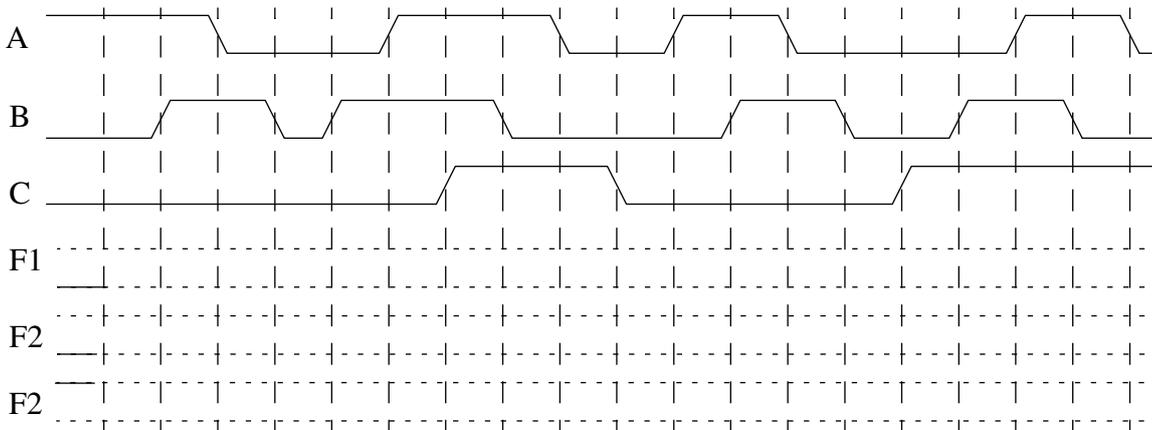
Se pide:

- Completar de forma teórica el siguiente cronograma, y a partir de él la tabla de combinaciones.

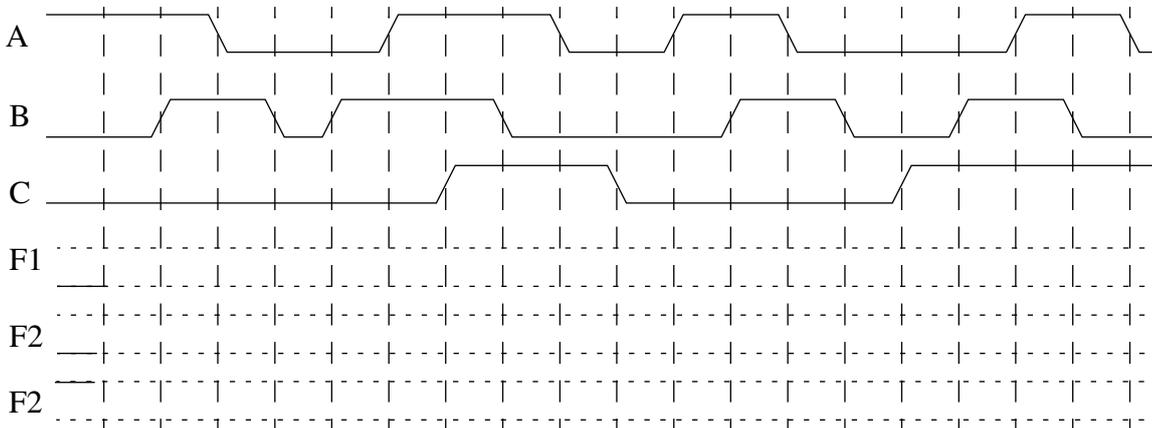


- Indicar razonadamente cuál de los dos circuitos tiene un comportamiento secuencial.

- Obtener el cronograma anterior y la tabla de combinaciones de ambos circuitos mediante simulación. Para lo cual, ambos circuitos deben alcanzar el estado estacionario para cada combinación de entradas.



- Obtener el cronograma y la tabla mediante emulación (a partir del montaje).



(Nota: En el caso de que los resultados de algún apartado no concuerden con los de otros, indicar razonadamente alguna causa de dicha incongruencia).

## Práctica II.- Modelado de Circuitos Secuenciales

### 2.1.- Objetivos.

El principal objetivo de esta práctica consiste en consolidar los conocimientos del alumno en la simplificación, modelado y test de las máquinas de estado.

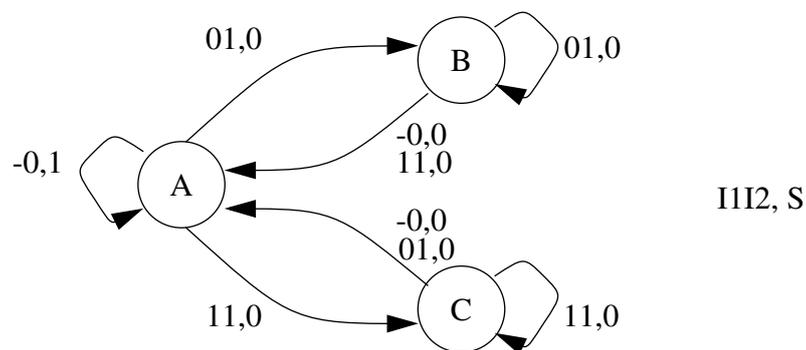
### 2.2.- Preliminares.

Una forma de probar los circuitos combinacionales consistía en generar la tabla de combinaciones, ya que la salida siempre era la misma para una combinación de entradas dada. En el caso de los sistemas secuenciales, esta afirmación no es cierta, luego el test de estos circuitos cambia.

El test de los circuitos secuenciales debe comprobar dos situaciones: test funcional y test de temporización; ya que para que un circuito secuencial funcione correctamente, se debe cumplir una determinada temporización. Otro punto importante es que debemos empezar por un estado inicial válido. La elección incorrecta de este estado suele ser motivo de que el resultado del test no sea el correcto, no por el modelo del sistema, sino por la elección de inicio del test. Por lo tanto, el test debe ser tal que recorra todas las transiciones que pueda tener la máquina de estados equivalente.

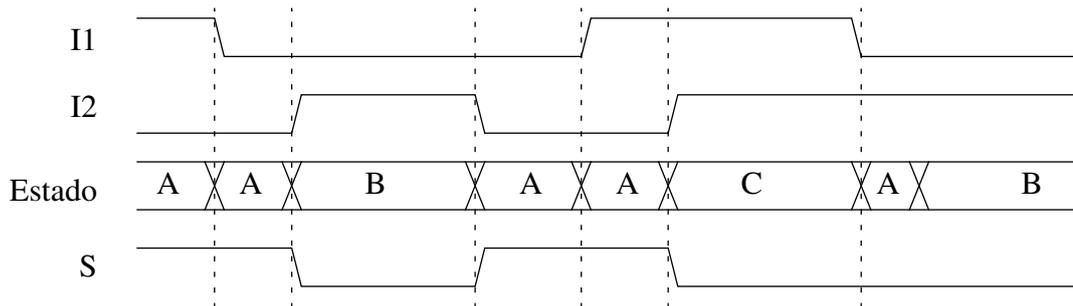
En el caso de que no tengamos una máquina secuencial equivalente, debemos generar todas las combinaciones, incluyendo las variables de estado presente. Estas últimas variables no deben ser introducidas como señales externas, ya que en este caso no comprobaríamos la temporización del circuito.

Supongamos que tenemos el siguiente diagrama de estados:



A la hora de realizar el test del sistema, y por lo tanto, su simulación, debemos partir de un estado bien conocido. La correcta inicialización del test, implica que deberíamos partir de un estado estable, y que no cambie cuando se inicie la simulación, a no ser que se cambien los patrones de entradas. De una revisión del diagrama, podemos apreciar que el estado A se

alcanza y se mantiene cuando  $I2='0'$ ; entonces tomaremos este estado como estado inicial. Así un posible cronograma que verificaría dicha máquina de estado es el siguiente:



Una vez que tenemos un comportamiento modelado como diagrama de estado, podemos comprobarlo directamente gracias a la posibilidad de simular algoritmos en VHDL. Para poder modelar un diagrama de estados directamente, debemos crear un nuevo tipo que se denominará, por lo general, “estado”. Para pasar de un estado a otro se aplica un algoritmo secuencial, por lo que necesitamos declarar un proceso. Este proceso modelará la operación del diagrama, a través de la sentencia “case”. Una vez que se haya modelado la operación del diagrama, pasamos a actualizar el estado. Es útil que en la actualización exista algún retraso para que en el caso de que se produzca un ciclo de estados, el simulador no se quede “colgado” al intentar dar dos valores diferentes a una misma señal en el mismo instante.

El código VHDL de dicho diagrama será el mostrado a continuación:

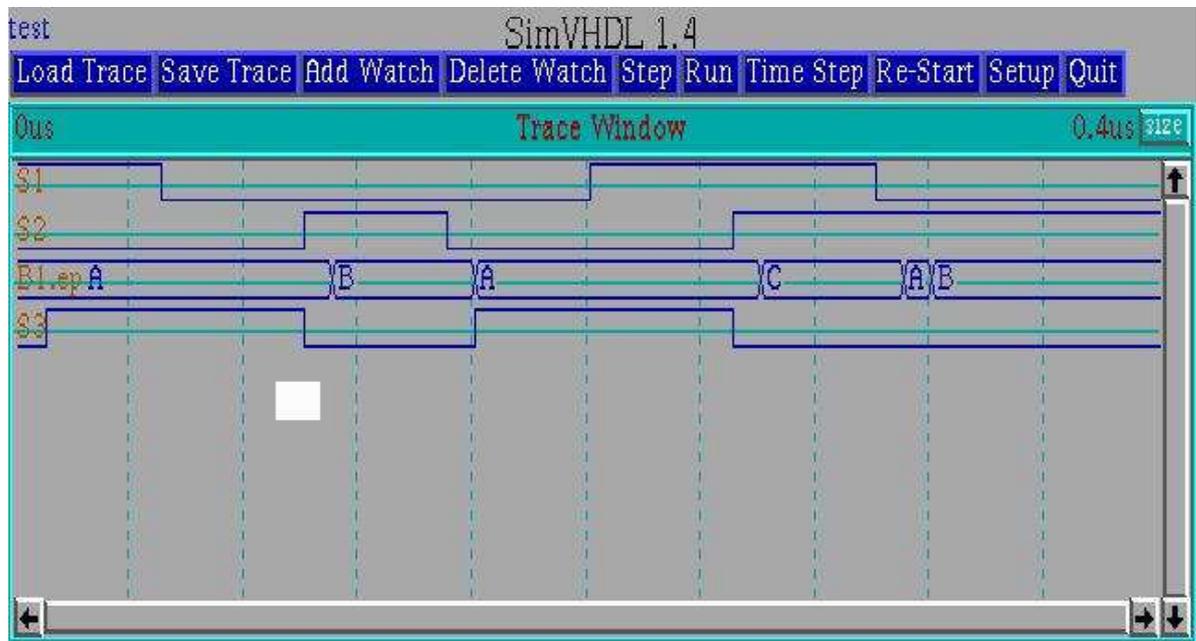
```
entity pract_2a is
port( I1, I2 : in bit;
      F1 : out bit);
end pract_2a;

architecture op of pract_2a is
-- Definición del tipo “estado”
type estado is (A, B, C);
-- Asignación de señales internas del tipo estado
signal presente1, proximo1 : estado;
begin
-- Inicio del proceso, que se ejecutará cuando cambien alguna
-- de las señales que tiene como argumento: señales de entrada
-- y el estado presente
P1:process(presente1, I1, I2)
begin
case presente1 is
when A =>
if (I2='0') then
proximo1<=A; F1<='1';
elsif (I1='0') then
proximo1<=B; F1<='0';
else proximo1<=C; F1<='0';
end if;
when B =>
if (I1='0' and I2='1') then
proximo1<=B; F1<='0';
else proximo1<=A;F1<='0';
end if;
when C =>
if (I1='1' and I2='1') then
proximo1<=C; F1<='0';
else proximo1<=A;F1<='0';
end if;
end case;
end process;
-- Actualización del estado presente
presente1 <= proximo1 after 10ns;
end;
```

Una vez que hemos modelado el diagrama de estado, debemos modelar el test para poder obtener el comportamiento a partir de los patrones de señales de entrada.

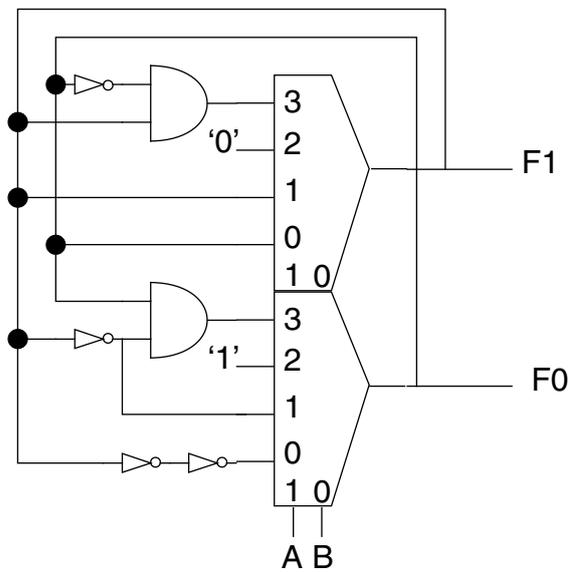
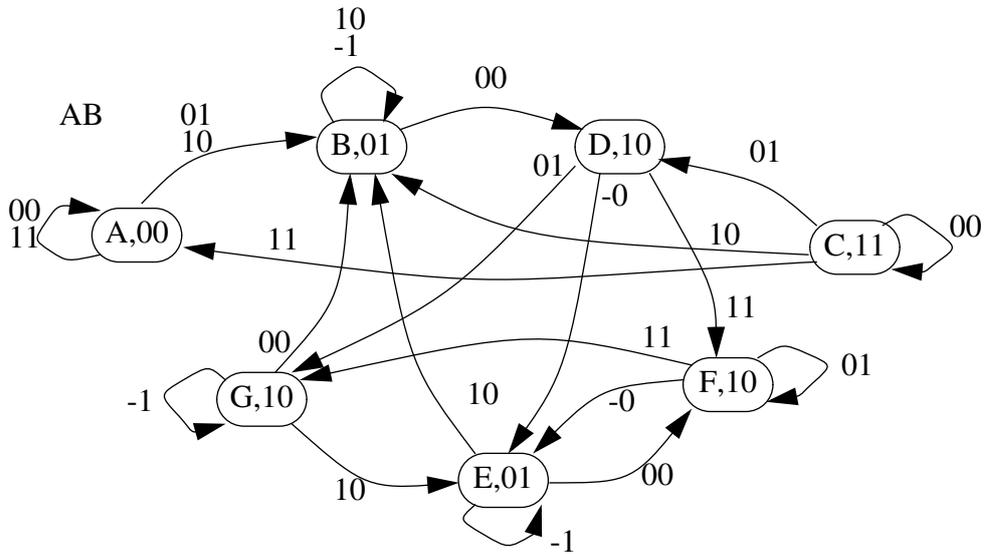
```
entity test is end test;
architecture op of test is
    signal S1, S2, S3 : bit;
    component diag
        port( I1, I2 : in bit;
              S : out bit);
    end component;
    for all:diag use entity work.diag;
begin
    B1: diag port map (S1, S2, S3);
    S1 <= '1', '0' after 50ns, '1' after 200ns, '0' after 300ns;
    S2 <= '0', '1' after 100ns, '0' after 150ns, '1' after 250ns;
end;
```

Luego, el comportamiento obtenido mediante simulación es el mismo que se obtuvo de forma teórica.



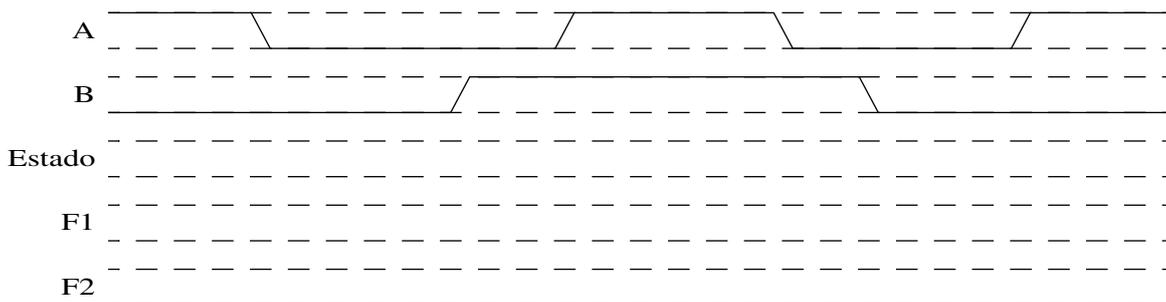
### 2.3.- Problema.

Se dispone de la siguiente máquina de estado y del circuito que la lleva a cabo.

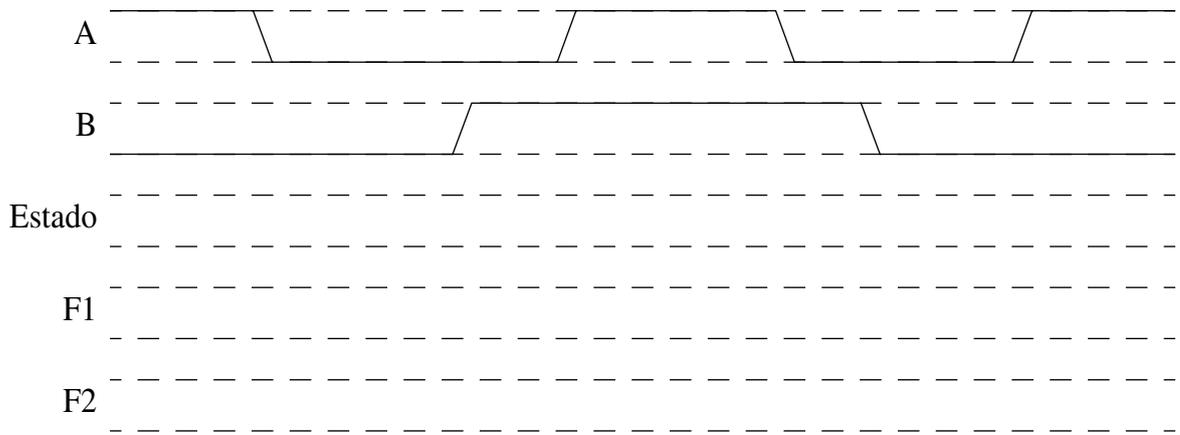


Se pide:

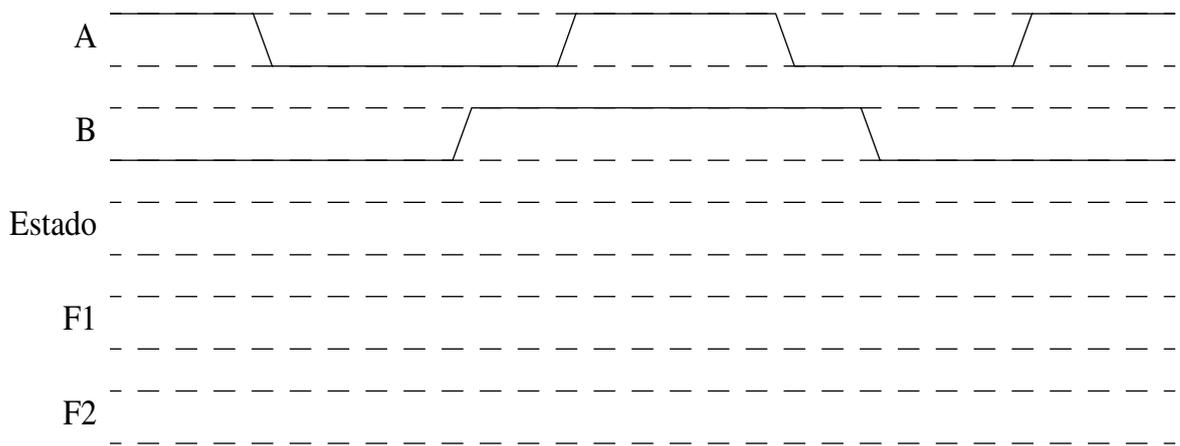
- Identificar el tipo de máquina de estado al que pertenece.
- Obtener el comportamiento de la máquina de estado, completando el siguiente cronograma.



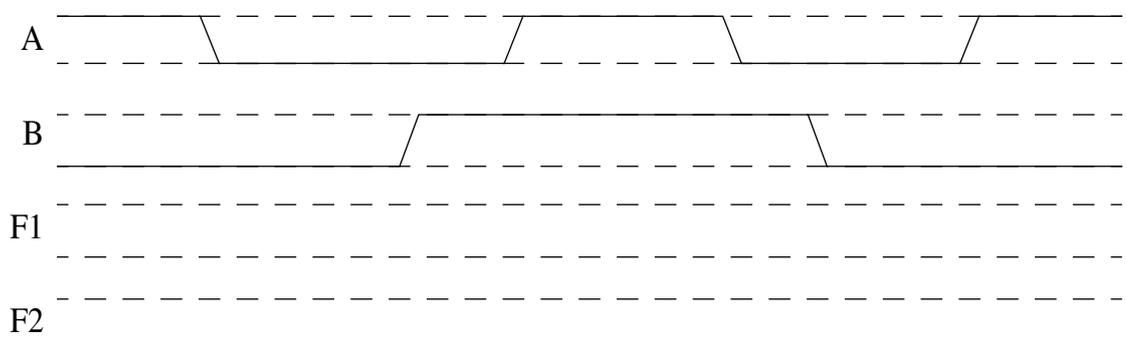
- Minimizar la máquina de estado, denominando a los nuevos estados como S1, ..., S99.
- Comprobar el comportamiento de la máquina minimizada, utilizando el siguiente cronograma. ¿Son equivalentes la máquina original y la minimizada?, ¿por qué?



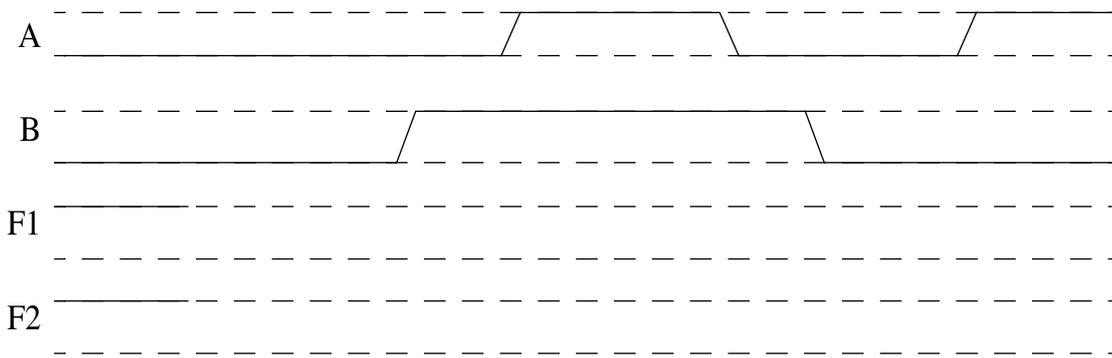
- Realizar dicha comprobación a través de simulación, utilizando las formas de onda de la figura inferior.



- Comprobar que la máquina minimizada tiene el mismo comportamiento que el circuito adjunto, mediante emulación, utilizando las formas de onda de la figura inferior.



- Comprobar que la máquina minimizada tiene el mismo comportamiento que el circuito adjunto, mediante emulación, utilizando las formas de onda de la figura inferior.



(Nota: En el caso de que los resultados de algún apartado no concuerden con los de otros, indicar razonadamente alguna causa de dicha incongruencia).

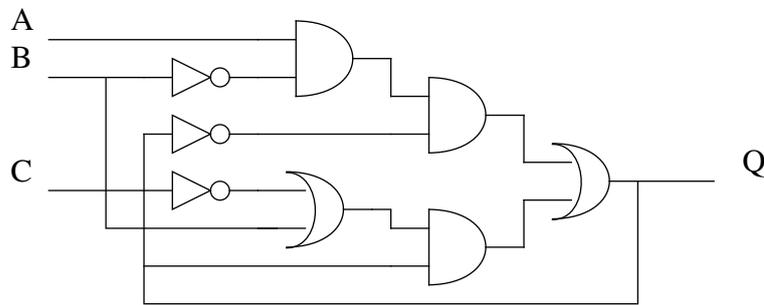
## Práctica III.- Elementos de Memoria

### 3.1.- Objetivos.

El principal objetivo de esta práctica consiste en la familiarización del alumno con los elementos de memoria, así como con sus principales características.

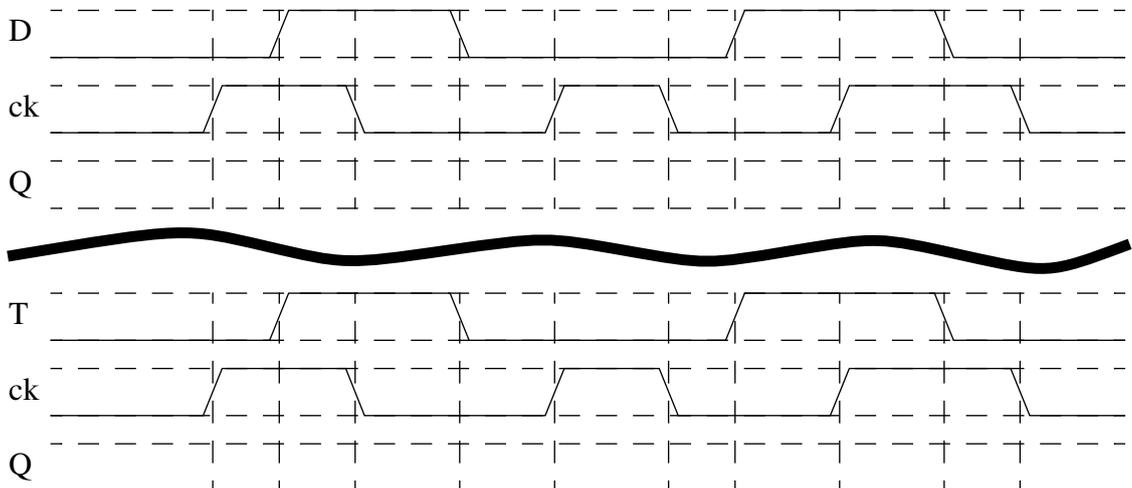
### 3.2.- Problema 1.

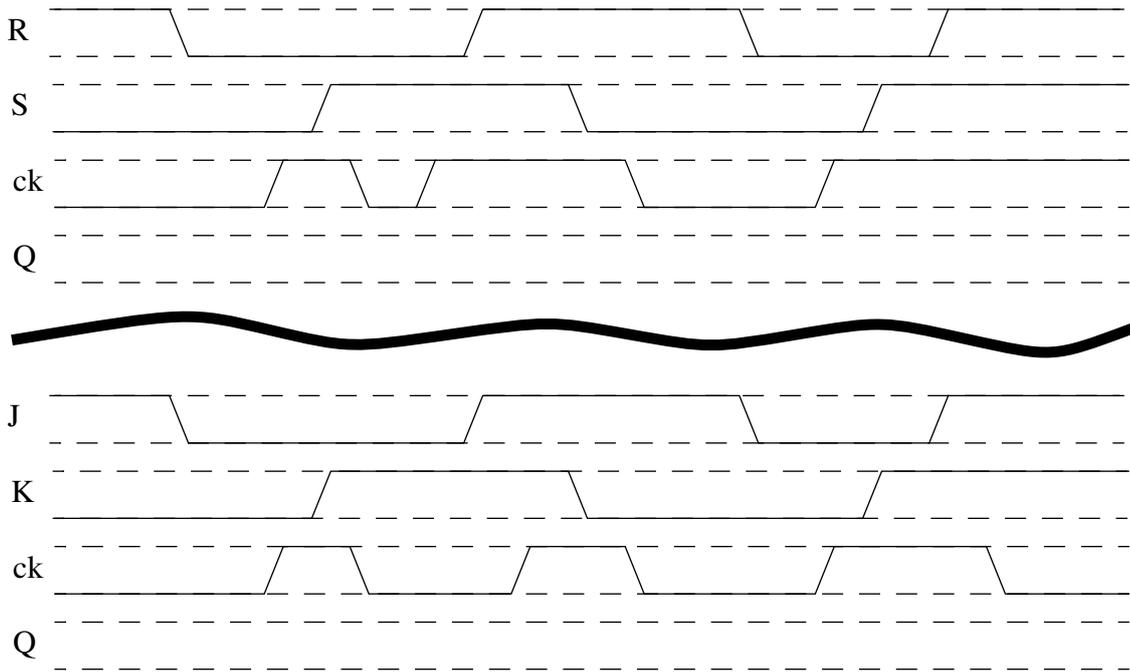
Disponemos del circuito de la siguiente figura:



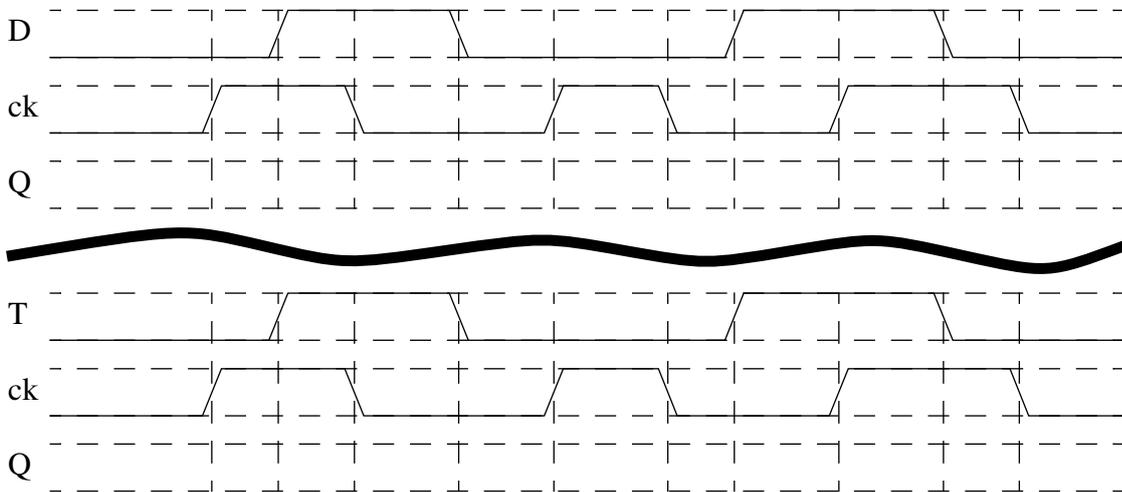
Se pide:

- Razonar si dicho circuito tiene el mismo comportamiento de algunos de los biestables estudiados (D, T, RS, o JK). En caso afirmativo, indicar qué tipo de biestable es, así como la categoría temporal en la que se encontraría (biestable transparente, latch o flip-flop).
- Realizar de forma teórica el diseño de todos los tipos de biestables estudiados (D, T, RS y JK), a partir del biestable anterior.
- Modelar los circuitos obtenidos en el punto anterior en VHDL a partir de los operadores lógicos. Los patrones para el test serán los siguientes.





- Montar en el laboratorio dichos circuitos y comprobar su comportamiento completo, utilizando las formas de onda anteriores.
- En el caso de los biestables tipo D y T, observar su comportamiento en el osciloscopio en los siguientes patrones.



(Nota: En el caso de que los resultados de algún apartado no concuerden con los de otros, indicar razonadamente alguna causa de dicha incongruencia).

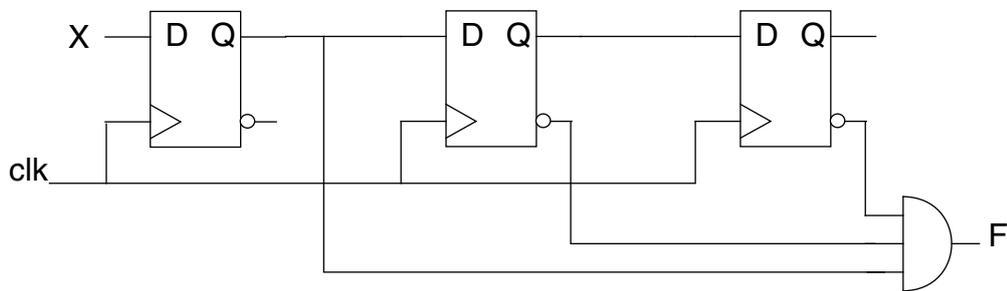
## Práctica IV.- Análisis de Circuitos Secuenciales

### 4.1.- Objetivos.

El principal objetivo de esta práctica consiste en familiarizar al alumno con el problema de análisis de circuitos secuenciales asíncronos y síncronos.

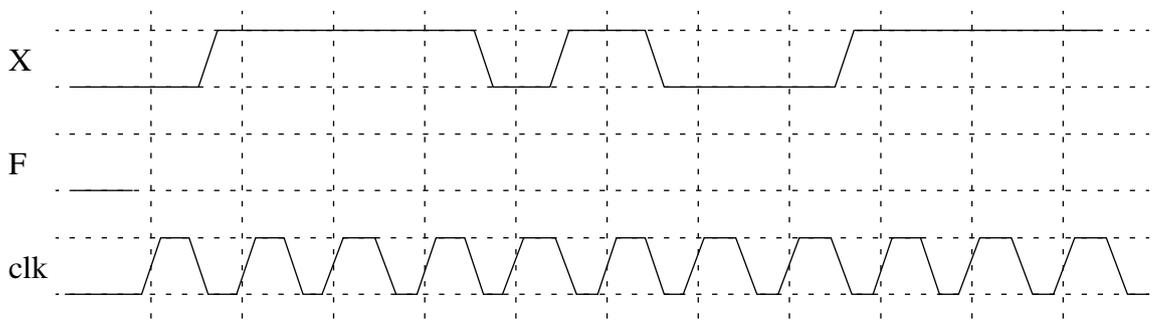
### 4.2.- Problema 1.

Se dispone del circuito mostrado en la siguiente figura:



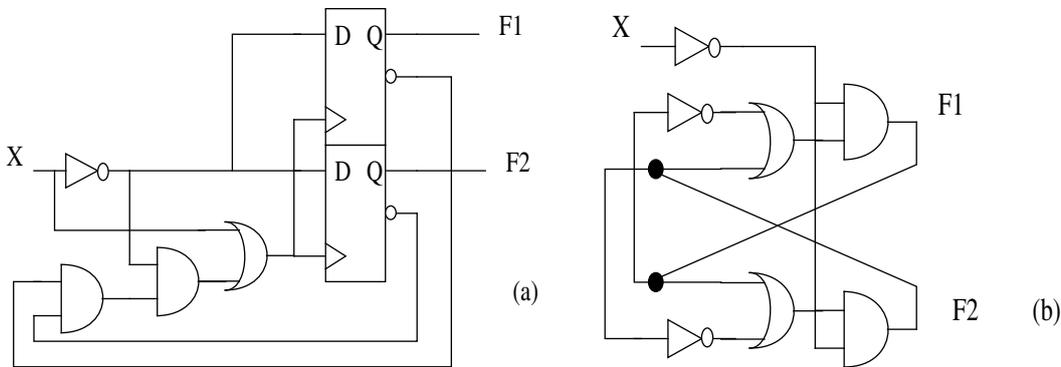
Se pide:

- Analizar de forma teórica dicho circuito, indicando razonadamente a qué categoría de circuitos secuenciales pertenece.
- Indicar los posibles problemas de temporización que pueda tener dicho circuito. En particular, estudiar la presencia de carreras y de ciclos de estados.
- Comprobar mediante simulación el comportamiento de dicho circuito, así como el de la máquina secuencial equivalente obtenida en el análisis anterior.
- Comprobar mediante emulación el comportamiento de dicho circuito. Para ello completar el siguiente cronograma. ¿Se han puesto de manifiesto los problemas indicados en el punto 2?; en caso negativo, indicar una respuesta razonada de esta incongruencia.



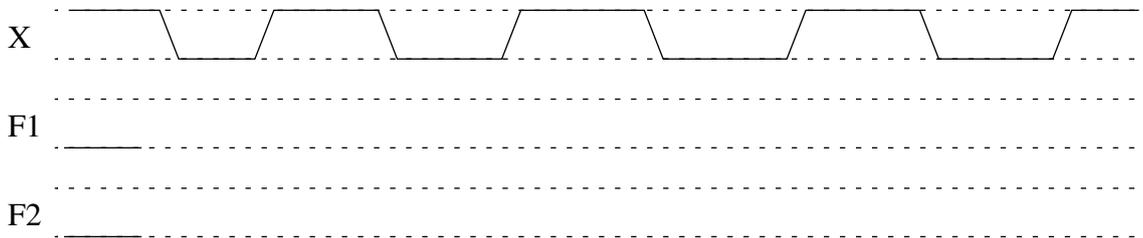
### 4.3.- Problema 2.

Se dispone de los circuitos mostrados en la siguiente figura:

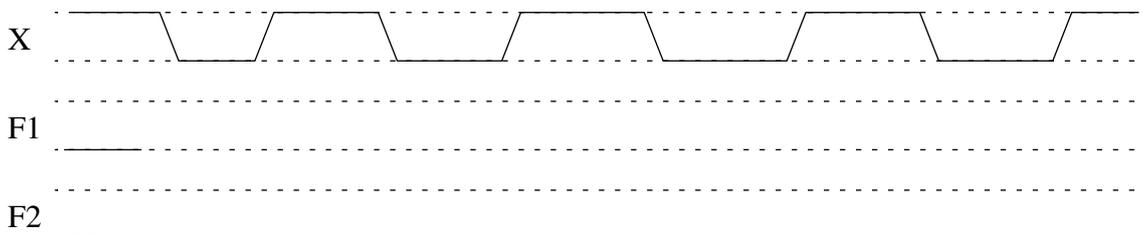


Se pide:

- Analizar de forma teórica los circuitos anteriores, indicando razonadamente a qué categoría de circuito secuencial pertenece.
- Indicar los posibles problemas de temporización que puedan tener dichos circuitos. En particular, estudiar la presencia de carreras y de ciclos de estados.
- Comprobar mediante simulación el comportamiento de dichos circuitos, así como el de las máquinas secuenciales equivalentes obtenida en el análisis anterior, completando el siguiente cronograma para cada circuito y cada máquina.



- Comprobar mediante emulación el comportamiento de dichos circuitos. Para ello completar el siguiente cronograma. ¿Se han puesto de manifiesto los problemas indicados en el punto 2?; en caso negativo, indicar una respuesta razonada de esta incongruencia.



(Nota: En el caso de que los resultados de algún apartado no concuerden con los de otros, indicar razonadamente alguna causa de dicha incongruencia).

## Práctica V.- Diseño de Sistemas Secuenciales

### 5.1.- Objetivos.

El principal objetivo de esta práctica consiste en la familiarización del alumno con los diferentes pasos en el proceso de diseño de un circuito secuencial a nivel de puertas lógicas.

### 5.2.- Problema.

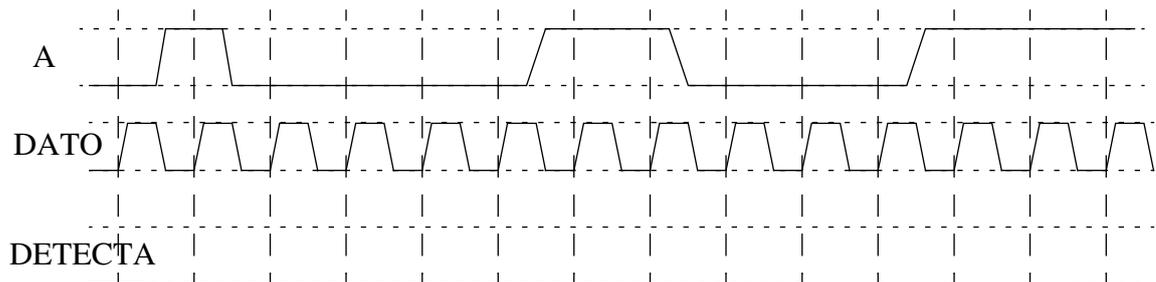
Una empresa de telecomunicaciones quiere enviar palabras de código en una comunicación serie, es decir, bit a bit. El código que utiliza es el mostrado en la siguiente tabla:

Palabras	B1	B2	B3
0	0	0	1
1	1	0	0
2	0	1	1

Cada bit de la palabra viene por una línea denominada A, y la llegada de un nuevo bit de la palabra es determinada por una transición de subida en una línea denominada DATO. La empresa necesita instalar en las estaciones receptoras un circuito que detecte la llegada de una palabra correcta o incorrecta. Esta detección será determinada por una señal, denominada DETECTA (valdrá '1' cuando la palabra sea correcta y llegue el tercer bit de la palabra).

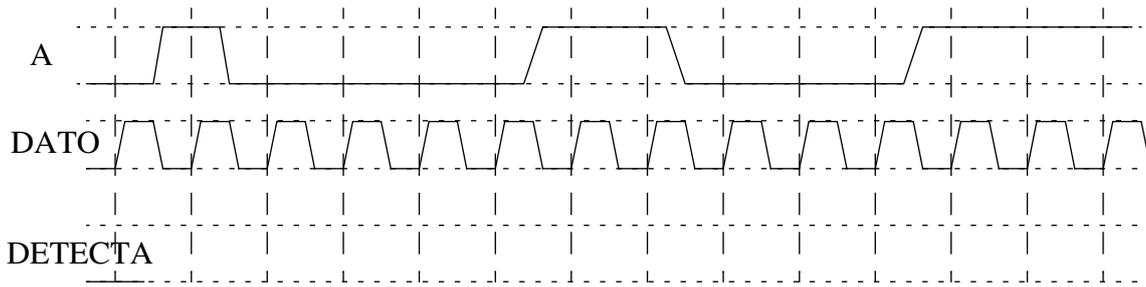
Se nos ha encargado el diseño del circuito de detección, para lo cual se pide:

- Completar, según el comportamiento indicado en el problema, el siguiente cronograma.

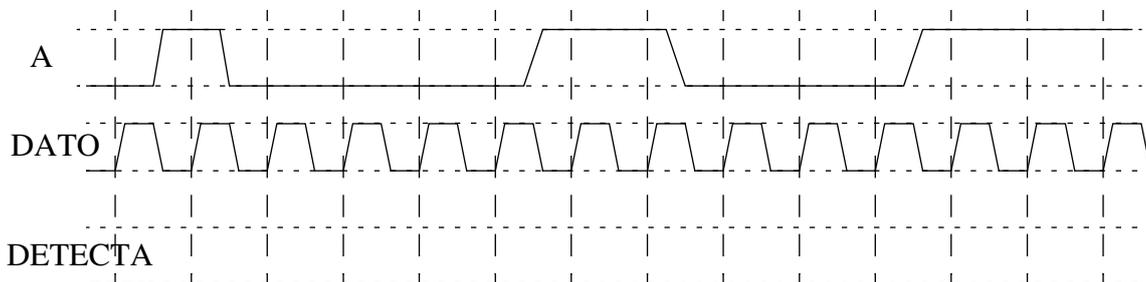


- Indicar razonadamente si el diseño solicitado se trata de un diseño combinacional o secuencial.
- Indicar razonadamente cuál sería la categoría de circuito secuencial que más se acomode a las especificaciones.
- Obtener la máquina secuencial que modela dicho circuito. Comprobar que dicha máquina es mínima, y en caso negativo, minimizar la máquina obtenida.

- Comprobar mediante simulación el comportamiento correcto de dicha máquina secuencial. Para ello, utilizar el siguiente patrón de señales de entradas.



- Obtener un asignamiento de estados, y a partir de él, la tabla de transición.
- Estudiar en dicha tabla de transición, y en la tabla de estados, los diferentes problemas de temporización.
- Indicar razonadamente si los problemas vistos, tales como carreras, transiciones consecutivas, ..., son considerados como problemas en este diseño.
- Obtener las tablas de excitaciones para una posible implementación con biestables tipo D, tipo T, tipo JK y tipo RS. Tras lo cual, indicar cuál de las implementaciones sería la mínima (considera todos los elementos de memoria con el mismo coste).
- Comprobar mediante emulación la implementación para biestables tipo D. Para ello, completar el siguiente cronograma.



(Nota: En el caso de que los resultados de algún apartado no concuerden con los de otros, indicar razonadamente alguna causa de dicha incongruencia).

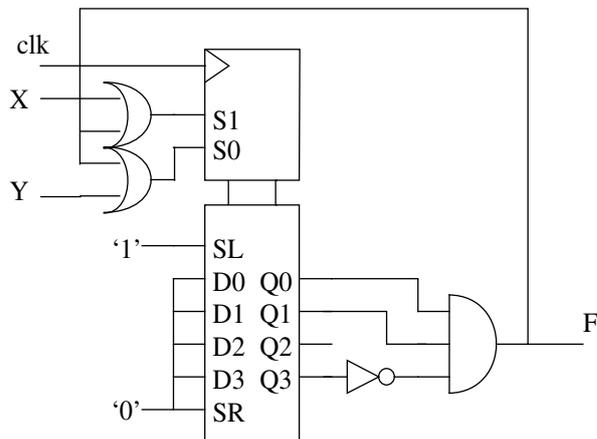
# Práctica VI.- Diseño Secuencial con Subcircuitos Secuenciales I

## 6.1.- Objetivos.

El objetivo principal de esta práctica consiste en familiarizar al alumno con el diseño y análisis de sistemas secuenciales utilizando registros de desplazamientos y contadores como elementos secuenciales.

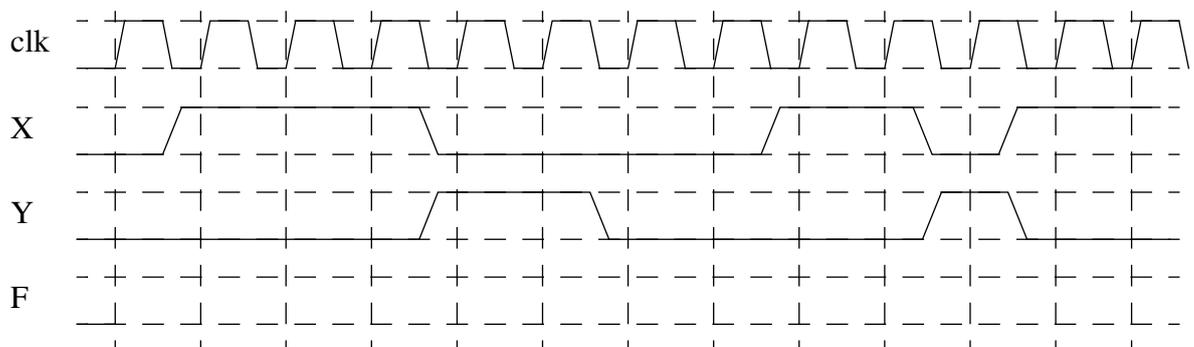
## 6.2.- Problema 1.

Una empresa de juguetería ha adquirido el siguiente circuito:



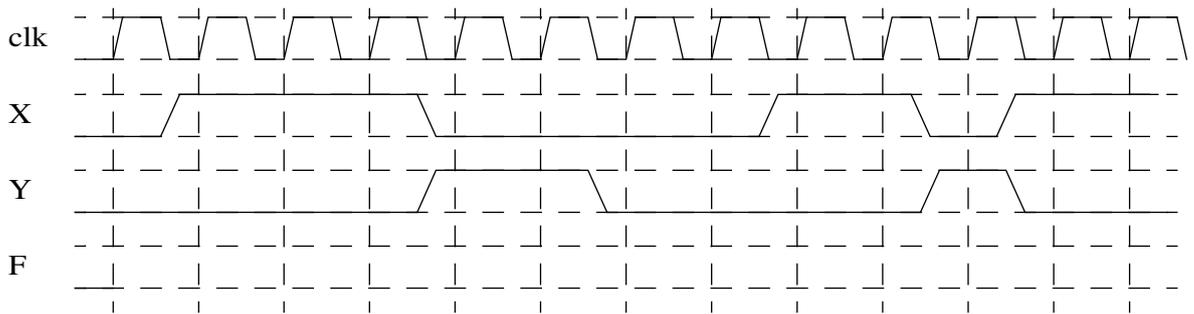
La empresa nos ha solicitado un análisis completo de dicho sistema, para lo cual se pide:

- Obtener las ecuaciones de excitación de dicho circuito.
- Obtener la tabla de transición de dicho circuito.
- Realizar la simulación del circuito utilizando el siguiente cronograma.



- Obtener la máquina secuencial correspondiente a dicho circuito.

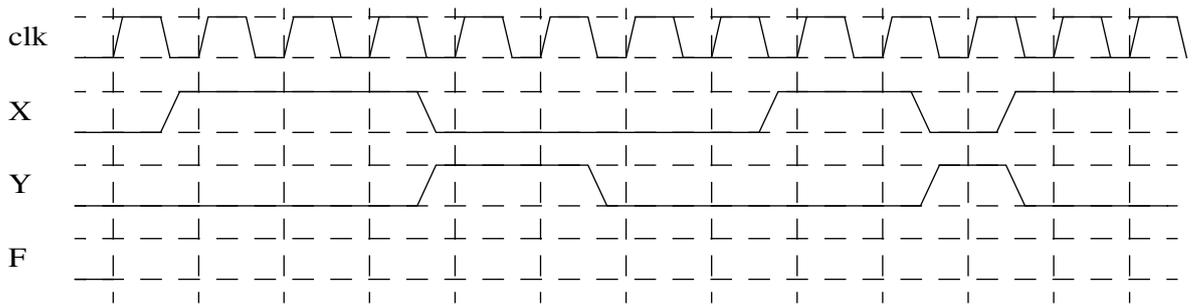
- Realizar el montaje del circuito anterior, completando el siguiente cronograma.



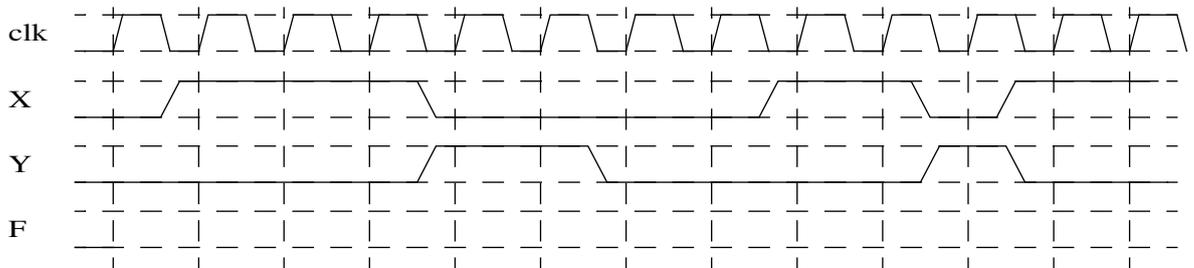
### 6.3.- Problema 2.

La misma empresa desea implementar el mismo comportamiento anterior, pero utilizando contadores universales, en lugar de utilizar registros de desplazamiento. Para ello, se pide:

- Obtener la máquina mínima correspondiente al circuito del problema 1.
- Simular la máquina secuencial anterior, con los siguientes patrones de entrada.



- Obtener un asignamiento de estados, considerando que el elemento secuencial será un contador universal de cuatro bits.
- Obtener la tabla de transición correspondiente a la máquina y asignamiento anterior.
- Obtener la tabla de excitación.
- Obtener el circuito correspondiente a la anterior tabla de excitación.
- Realizar el montaje del circuito anterior, y completar el siguiente cronograma.



(Nota: En el caso de que los resultados de algún apartado no concuerden con los de otros, indicar razonadamente alguna causa de dicha incongruencia).

# Práctica VII.- Diseño Secuencial con Subsistemas Secuenciales II

## 7.1.- Objetivos.

El objetivo principal de esta práctica consiste en familiarizar al alumno con el empleo de los diagramas ASM, así como su síntesis en un bloque procesador-controlador.

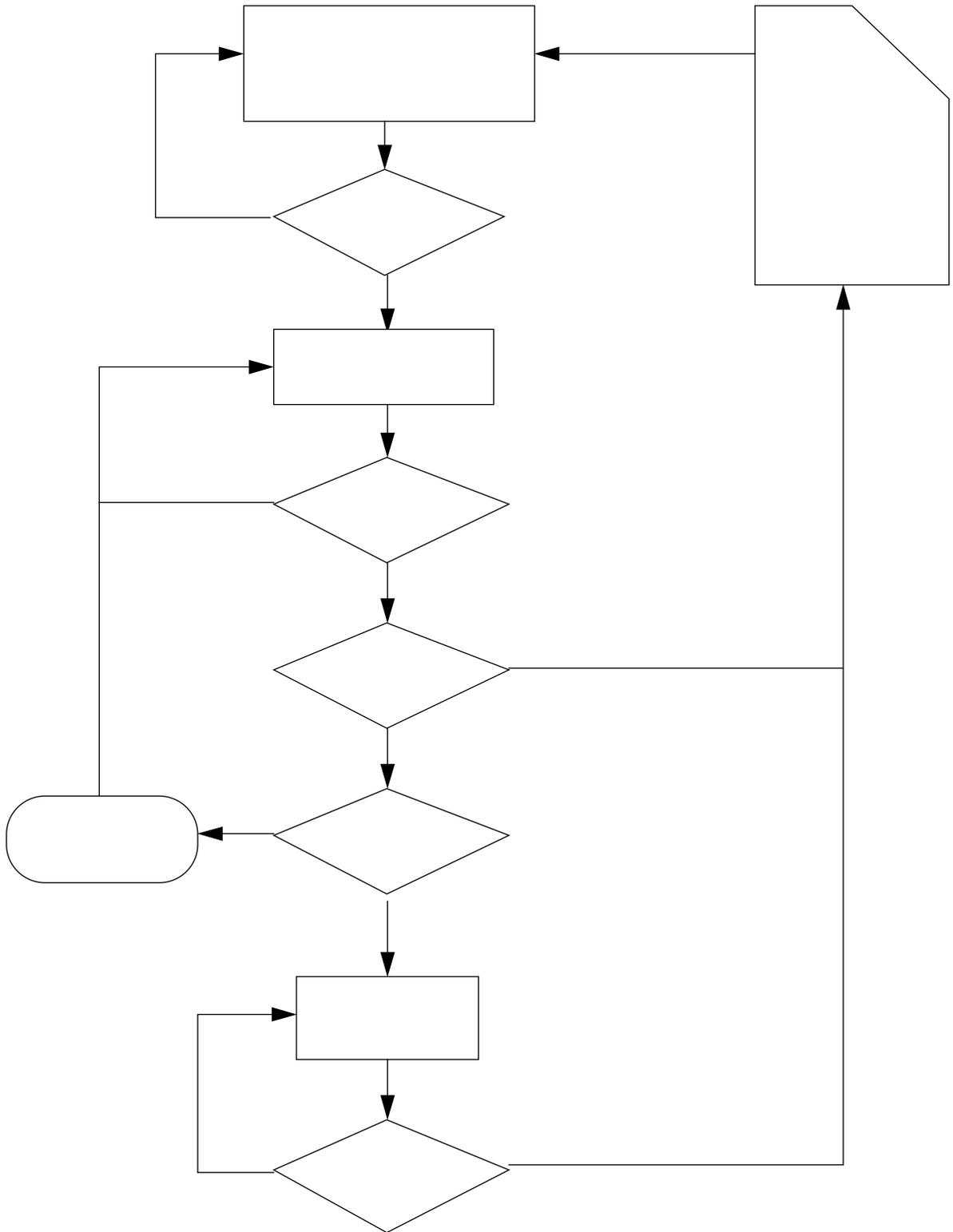
## 7.2.- Problema.

Una empresa de refinería dispone de un tanque de combustible, al cual le quiere adjuntar un sistema de seguridad debido a la peligrosidad de la mercancía almacenada en el tanque. Las especificaciones que deben cumplir el sistema de seguridad son las siguientes:

- El sistema trabaja con un sensor de nivel (indicando la capacidad utilizada del tanque). dicho sensor devuelve su valor con una precisión de cuatro bits.
- Se dispone de un botón de reset síncrono, que devuelve el sistema a su estado inicial.
- Se dispone de un botón de inicio, cuya activación implica el inicio de la operación del sistema.
- Si el nivel del tanque es superior a 8 unidades durante cinco ciclos consecutivos de reloj, se debe activar una alarma indentificada con un led azul.
- Una vez que se encienda el led azul, el sistema entrará en un estado del cual solamente se podrá salir con el botón de reset.

Para este diseño se pide lo siguiente:

- Completar el diagrama ASM adjunto para que modele las especificaciones dadas.
- Indicar los bloques necesarios para realizar el procesador de dicho sistema.
- Indicar el diagrama de estados que debe implementar el controlador de dicho sistema, indicando las señales de entrada y salida así como de su significado (el de las señales internas al sistema que no esté en las especificaciones).
- Obtener el circuito secuencial que implemente dicho diagrama de estados.
- Realizar la simulación de dicho circuito secuencial.
- Unir el procesador y el controlador para obtener la versión final del sistema de seguridad.
- Montar el diseño completo obtenido a partir del diagrama ASM.



(Nota: En el caso de que los resultados de algún apartado no concuerden con los de otros, indicar razonadamente alguna causa de dicha incongruencia).