

PAC3 - Tercera prueba de evaluación continuada

Presentación

Esta PEC se focaliza en los circuitos secuenciales. Los circuitos combinacionales nos permiten describir funcionalidades de un circuito, pero no nos permite guardar información. Mediante biestables y registros podemos guardar información en memoria y hacer circuitos más complejos. En este PEC practicaremos con este tipo de circuitos.

Competencias

- Entender el funcionamiento de los circuitos lógicos secuenciales y conocer y saber aplicar técnicas de diseño de sistemas secuenciales.

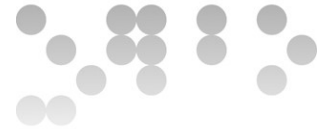
Objetivos

- Saber discernir, a partir de la funcionalidad que se quiere que tenga un circuito lógico, si el circuito tiene que ser de tipo secuencial o combinacional.
- Conocer el funcionamiento del biestable D y de todas las entradas de control que puede tener.
- Saber analizar un circuito secuencial.
- Saber realizar un cronograma a partir d'un circuito digital secuencial.
- Saber analizar un grafo de estados.
- Saber diseñar un circuito cualquiera a partir de la descripción de su funcionalidad mediante el modelo de Moore.

Recursos

Los recursos que se recomienda usar por esta PEC son los siguientes:

- **Básicos:** El módulo 4 de los materiales. En cada pregunta, se indica en qué sección de los materiales se puede encontrar la información para resolverlos. Notáis que también existe en los materiales una cantidad muy extensa de ejercicios para ver como se pueden resolver.
- **PEC anteriores:** En el aula de CANVAS, en recursos adicionales podéis encontrar PACS resueltas otros semestres.
- **Complementarios:** VerilCIRC, VerilCHART y el Wiki de la asignatura.



Criterios de valoración

- Razonáis la respuesta en todos los ejercicios. Las respuestas sin justificación no recibirán puntuación.
- Los ejercicios realizados con IA generativa no recibirán puntuación.
- La valoración esta indicada en cada uno de los apartados y subapartados.

Uso de herramientas de IA

En esta actividad no está permitido el uso de herramientas de inteligencia artificial. Al plan docente y al [web sobre integridad académica y plagio](#) de la UOC encontraréis información sobre que se considera conducta irregular en la evaluación y las consecuencias que puede tener.

Formato y fecha de entrega

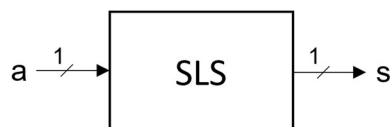
- Para dudas y aclaraciones sobre el enunciado, dirigíos al consultor responsable de vuestra aula.
- Hay que librar la solución en un fichero PDF usando una de las plantillas libradas conjuntamente con este enunciado.
- Se tiene que librar a través de la aplicación de **Entrega de la Actividad** correspondiendo del apartado **Contenidos** de vuestra aula.
- La fecha tope de entrega es lo **30 de abril** (a las 24 horas).

Descripción de la PEC a realizar

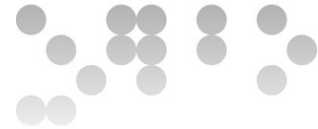
Ejercicio 1 [25 %]

(Sección 4.2: Representación gráfica: grafos de estados)

Se quiere diseñar el grafo de estados de un circuito lógico secuencial llamado SLS. El circuito tiene una entrada de un bit, denominada a , y una salida de un bit, denominada s , según la estructura siguiente:



El circuito SLS va leyendo el valor de la entrada en cada ciclo. Denominamos a_i al valor en la entrada en el ciclo i . El circuito tiene que detectar en cada ciclo si el valor formado por los tres dígitos leídos en los últimos tres ciclos, $a_i a_{i-1} a_{i-2}$, es un número múltiple de



tres. Cuando se detecta que el valor es múltiple de tres en el siguiente ciclo se pone el valor 1 en la salida s durante un ciclo. En cualquier otro caso la salida s vale 0.

Tened en cuenta que el número cero es múltiple de cualquier número y , por lo tanto, es múltiple de tres. Inicialmente podéis considerar que el circuito se encuentra en el estado 000 (como si las últimas tres entradas fueran cero) y, por lo tanto, la salida es 1.

Ejemplo de funcionamiento:

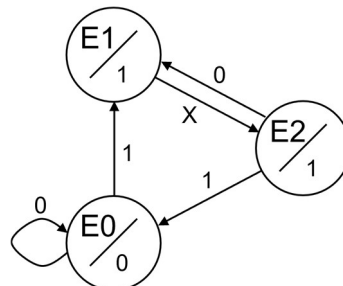
Entrada a	0	1	0	0	0	1	1	1	0	1	0	1	...
Salida s	1	1	0	0	0	1	0	1	0	1	0	0	0

Se pide que diseñéis el grafo de estados del circuito SLS, especificando claramente la leyenda del mismo.

Ejercicio 2 [25 %]

(Sección 4.4: Implementación (del modelo de Moore))

- a) [5 %] El grafo siguiente describe el comportamiento de un circuito secuencial (no dibujamos su leyenda porque no es necesario, para este ejercicio). Suponiendo que este circuito se implementa de la forma habitual, con una memoria ROM y biestables, indicad cuántas palabras tiene que tener la memoria ROM, de cuántos bits tiene que ser cada uno, y el contenido de la memoria ROM.



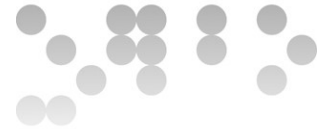
Le hemos preguntado a una herramienta generativa que resuelva el ejercicio y ha dado el siguiente resultado:

Para determinar las características de la memoria ROM en la implementación del circuito secuencial, seguimos estos pasos:

1. Determinar la cantidad de estados

Observamos que el circuito tiene tres estados:

- $E0$
- $E1$
- $E2$



Para codificar estos estados en la memoria, necesitamos **al menos 2 bits** (n) porque con 2 bits podemos representar hasta 4 estados ($2^2=4$), lo cual es suficiente para nuestros 3 estados.

2. Determinar las entradas y salidas

La tabla de transición nos dice que:

- Hay una única entrada X (valores 0 o 1).
- Hay una única salida S (valores 0 o 1).

3. Formato de la memoria ROM

Cada palabra de la ROM tiene que contener:

1. El estado siguiente E^+ , que está codificado en 2 bits.
2. La salida S , que ocupa 1 bit.

Por lo tanto, cada palabra de la ROM tendrá $2 + 1 = 3$ bits.

4. Número de palabras en la ROM

Cada combinación de (E, X) representa una dirección en la ROM. Como:

- Hay 3 estados posibles (E_0, E_1, E_2).
- Hay 2 posibles valores de X (0 o 1).

Entonces, el número total de combinaciones es:

$$3 \times 2 = 6$$

Por lo tanto, la ROM es de 6 palabras.

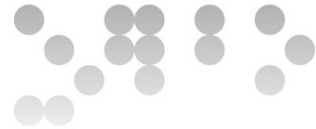
5. Contenido de la ROM

Cada fila de la tabla de transición se traduce en una dirección de la ROM:

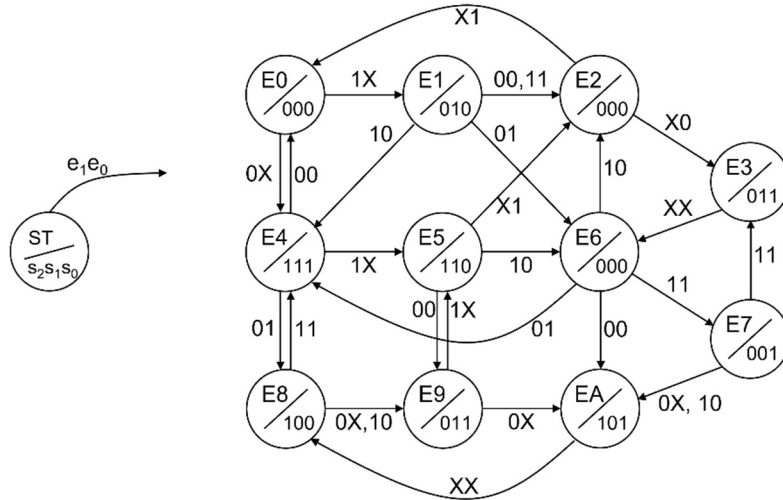
Estado Actual (E)	Entrada (X)	Estado Siguiente (E+)	Salida (S)	Codificación ROM
E0 (00)	0	E0 (00)	0	000000
E0 (00)	1	E1 (01)	0	001001
E1 (01)	X	E2 (10)	1	010101
E2 (10)	0	E1 (01)	1	100011
E2 (10)	1	E0 (00)	1	101001

(Nota: La tercera fila se aplica a cualquier X , por lo cual se escribe una sola vez).

Razonad la corrección de la solución y, en caso de error, indicad qué errores aparecen en la solución propuesta.

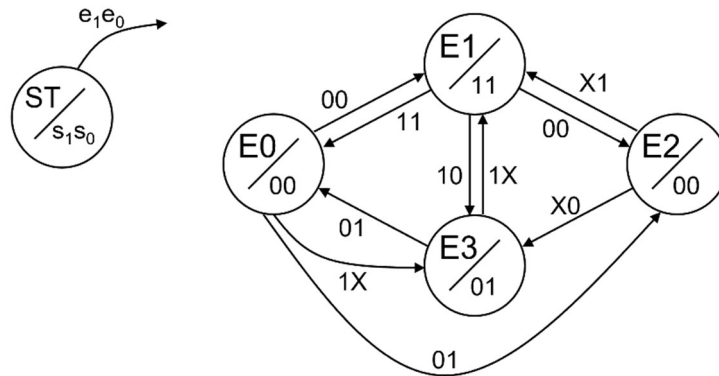


b) [5 %] Dado el grafo de estados siguiente:



Cuántos bits de entrada tiene el circuito que implementa este grafo? Cuántos bits de salida? Cuál será el número mínimo de biestables para implementarlo? Si lo implementamos utilizando una memoria ROM, cuántos bits de direcciones y cuántos bits de datos necesitará esta memoria?

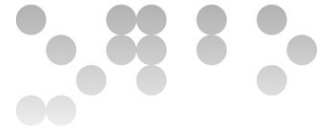
Dado el grafo de estados siguiente:



Se pide:

c) [10 %] Escribid la tabla de transiciones y la tabla de salidas del sistema representado por el grafo, codificando los estados según su índice asociado. Al escribir la tabla de transiciones, ponded en primer lugar las variables que codifican el estado y a continuación las variables de entrada.

Nota: Tenéis disponible el ejercicio a VerilCHART. Para poder probar este ejercicio en VerilCHART tenéis que sustituir, si fuera el caso, los bits *don't care* por valores 0.



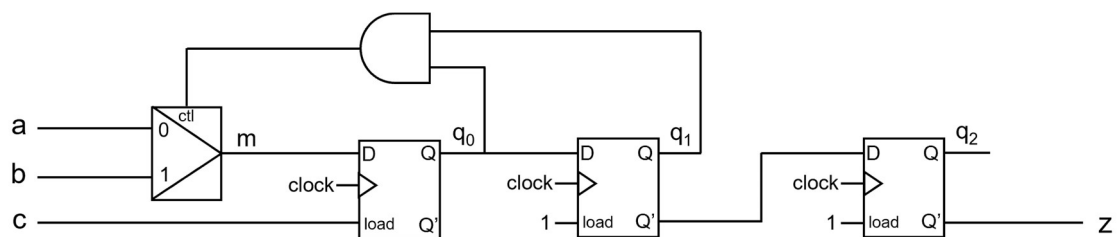
- d) **[5 %]** Diseñad el circuito que implementa el sistema representado por el grafo de estados utilizando una memoria ROM. Mostrad el contenido completo de la memoria ROM en hexadecimal.

Nota: Tenéis disponible el ejercicio en VerilCIRC. Para poder probar este ejercicio en VerilCIRC tenéis que sustituir, si fuera el caso, los bits *don't care* por valores 0.

Ejercicio 3 [25 %]

(Sección 2.3: Entradas asíncronas)

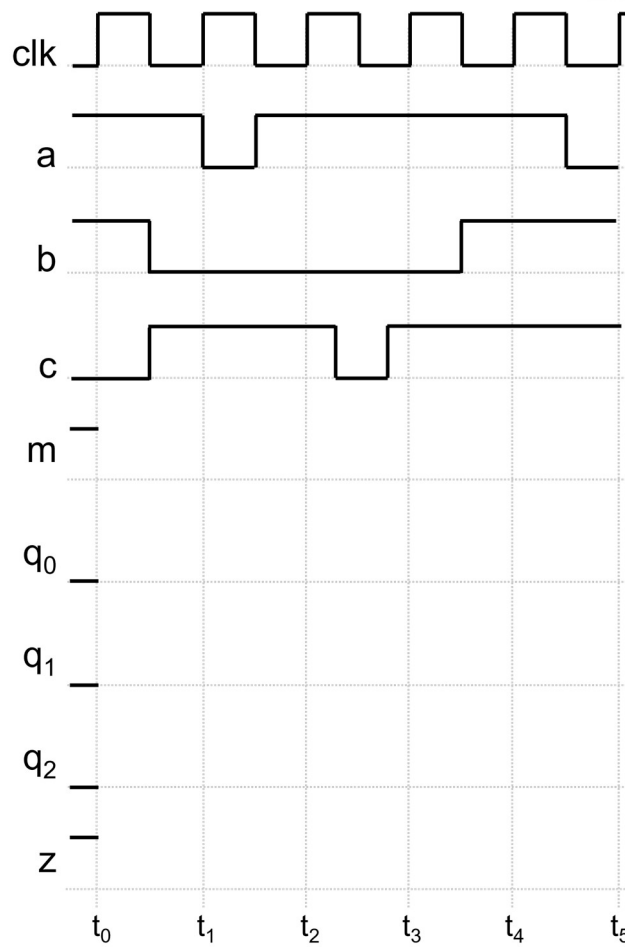
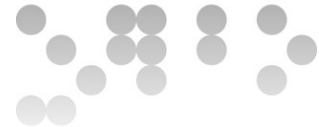
Dado el circuito siguiente:



- a) **[10 %]** Analizad el circuito y rellenad la tabla siguiente por los valores de entrada y estados especificados:

q ₂	q ₁	q ₀	a	b	c	d ₂	d ₁	d ₀	z
0	0	1	0	1	1				
0	1	0	1	0	1				
0	1	1	0	1	1				
1	0	0	0	1	1				
1	1	0	1	1	0				
1	1	1	1	0	1				

- b) **[15 %]** Completad el cronograma siguiente:



Nota: Tenéis disponible este ejercicio en VerilCHART.

Ejercicio 4 [25 %]

(Sección 3.1. Registro)

Dado el circuito siguiente:

