75.562 **·** Fundamentos de Computadores **·** 2024-25

PEC3 - Tercera prueba de evaluación continua

Apellidos:

Nombre:

**Formato y fecha de entrega**

* Para dudas y aclaraciones sobre el enunciado debéis dirigiros al consultor responsable de vuestra aula.
* Hay que entregar la solución en un fichero PDF utilizando una de las plantillas entregadas conjuntamente con este enunciado.
* Se debe entregar a través de la aplicación de **Entrega de la Actividad** correspondiente del apartado **Contenidos** de vuestra aula.
* La fecha límite de entrega es el  **30 de abril** (a las 24 horas).
* **Razonad la respuesta en todos los ejercicios. Las respuestas sin justificación no recibirán puntuación.**

**Respuestas**

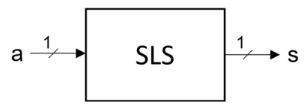
**Ejercicio 1 [25 %]**

(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 ai 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, ai ai-1 ai-2, es un número múltiplo de tres. Cuando se detecta que el valor es múltiplo 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 múltiplo de cualquier número y, por lo tanto, es múltiplo 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 |

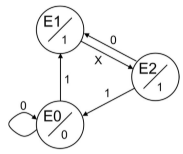
Se pide que diseñéis el grafo de estados del circuito SLS, especificando claramente la

leyenda del mismo.

**Ejercicio 2 [25 %]**

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

1. **[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 (22=4), lo cual es suficiente para nuestros 3 estados.*

1. ***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).*

1. ***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****.*

1. ***Número de palabras en la ROM***

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

* *Hay* ***3 estados posibles*** *(E0, E1, E2).*
* *Hay* ***2 posibles valores de X (0 o 1)****.*

*Entonces, el número total de combinaciones es:*

*3×2=6*

*Por lo tanto, la ROM es de* ***6 palabras****.*

1. ***Contenido de la ROM***

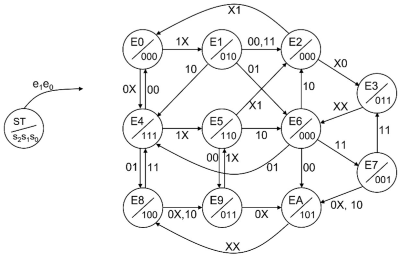
*Cada fila de la tabla de transición se traduce en una dirección de la ROM:*

| **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* | *010001* |
| *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*

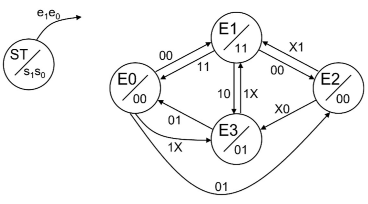
Razonad la corrección de la solución y, en caso de error, indicad qué errores aparecen en la solución propuesta.

1. **[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:

1. **[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, poned en primer lugar las variables que codifican el estado y a continuación las variables de entrada.

**NOTA**: Tenéis disponible el ejercicio a VerilChart (20242\_PAC3\_2c\_vch). Para poder probar este ejercicio en VerilChart tenéis que sustituir, si fuera el caso, los bits *don’t care* por valores 0.

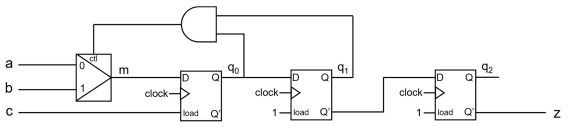
1. **[5 %]** Diseñad el circuito que implementa el sistema representado por el grafo de estados utilizando una memoria ROM. Mostrad el contenido completo de la memoria ROM en hexadecimal.

**NOTA:** Tenéis disponible el ejercicio en VerilCirc (20242\_PAC3\_2d\_circ). 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:

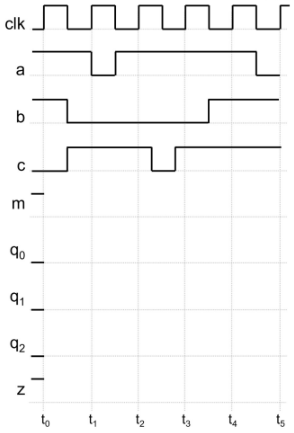


1. **[10 %]** Analizad el circuito y rellenad la tabla siguiente por los valores de entrada y estados especificados:

| **q2** | **q1** | **q0** | **a** | **b** | **c** | **d2** | **d1** | **d0** | **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 |  |  |  |  |

1. **[15 %]** Completad el cronograma siguiente:

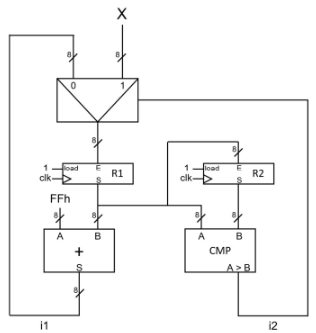
**NOTA**: Tenéis disponible este ejercicio en VerilChart (20242\_PAC3\_3b\_vch).



**Ejercicio 3 [25 %]**

(Sección 3.1. Registro)

Dado el circuito siguiente:



Completad el cronograma siguiente, poniendo los valores de los registros en hexadecimal, e incluyendo los valores intermedios i1y i2.

**NOTA**: Tenéis disponible este ejercicio en VerilChart (20242\_PAC3\_4\_vch).

