

INGENIERÍA DE COMPUTADORES 3

Trabajo Práctico - Convocatoria extraordinaria 2026

INSTRUCCIONES

- El trabajo práctico debe realizarse de manera individual. No puede realizarse en grupo. Por ello, se penalizará cualquier uso compartido de las soluciones propuestas y de los códigos programados.
- El trabajo debe entregarse a través del curso virtual de la asignatura.
- La fecha límite de entrega es el día 10 de septiembre.
- El alumno debe entregar un fichero comprimido, en formato zip o tar, que contenga:
 - Una memoria en la cual explique la solución a los ejercicios, incluyendo los listados documentados del código VHDL desarrollado. Este documento deberá estar en formato pdf.
 - Los ficheros del código VHDL solución a los ejercicios.

El nombre del fichero comprimido debe ser la concatenación de los apellidos y nombre del alumno. Por ejemplo, GomezMartinLuisa.zip

CRITERIOS DE EVALUACIÓN DEL TRABAJO

- Para que el trabajo pueda ser corregido, es imprescindible que el alumno entregue dentro del plazo establecido un fichero comprimido que contenga tanto la memoria en formato pdf, como los ficheros fuente del código VHDL de los ejercicios que haya realizado.
- La memoria ha de incluir el código VHDL de los ejercicios que haya realizado así como capturas de pantalla de las simulaciones realizadas.
- El trabajo se compone de 2 ejercicios con varios apartados. En el enunciado se indica la puntuación de cada apartado.
- Para aprobar el trabajo es necesario que la suma de la nota obtenida en los dos ejercicios sea igual o mayor que 5.
- Si el código VHDL solución de un apartado tiene uno o varios errores de compilación, o no tiene la funcionalidad pedida, dicho apartado se valorará con cero puntos.
- Si el código solución de un apartado compila sin errores y tiene la funcionalidad pedida, la puntuación en dicho apartado será al menos el 80 % de la nota del apartado.
- Se valorará positivamente la adecuada documentación del código, así como la presentación y calidad de las explicaciones proporcionadas en la memoria.

PROCEDIMIENTO DE EVALUACIÓN EN ESTA ASIGNATURA

Para aprobar la asignatura es necesario aprobar el trabajo y aprobar el examen.

Plantearémos un trabajo para la convocatoria ordinaria (junio) y otro diferente para la convocatoria extraordinaria (septiembre). Este trabajo que está leyendo corresponde a la convocatoria extraordinaria de 2026.

La nota obtenida en la convocatoria ordinaria en el trabajo y en el examen se guarda para la convocatoria extraordinaria. Es decir:

- Si un alumno aprueba el trabajo de la convocatoria ordinaria y no aprueba el examen, se le guarda la nota del trabajo para la convocatoria extraordinaria. Es decir, no debe hacer el trabajo de la convocatoria extraordinaria.
- Si un alumno no entrega o suspende el trabajo en convocatoria ordinaria, pero sí aprueba el examen en convocatoria ordinaria, entonces se le guarda la nota del examen para la convocatoria extraordinaria, debiendo aprobar el trabajo de la convocatoria extraordinaria para superar la asignatura.

La nota del trabajo y del examen no se guarda de un curso para otro.

EJERCICIO 1

Se desea diseñar un circuito digital que implemente las funciones X e Y mostradas a continuación, que dependen de las cuatro variables A , B , C y D :

$$X = A'B'C' + A'B'C + A'BC' + A'BC + AB'D + ABD$$

$$Y = A'B'D + A'BD + AB'D + ABD + ABCD'$$

- 1.a) (0.5 puntos) Escriba en VHDL la **entity** del circuito que implemente las dos funciones lógicas. Es decir, que tenga cuatro entradas A , B , C y D , y dos salidas X e Y .
- 1.b) (1 punto) Escriba en VHDL la **architecture** que describa el *comportamiento* del circuito utilizando las expresiones lógicas sin simplificar.
- 1.c) (0.5 puntos) Simplifique las funciones lógicas X e Y . Dibuje el diagrama de un circuito que implemente las funciones simplificadas al nivel de puertas lógicas.

No es necesario obtener una implementación mínima en número de puertas, pero sí debe corresponderse con las expresiones lógicas obtenidas en el apartado anterior.

A continuación, escriba en VHDL la **entity** y la **architecture** de cada una de las puertas lógicas que componen el circuito que acaba de dibujar.

- 1.d) (1 punto) Escriba en VHDL una **architecture** que describa la *estructura* del circuito que ha dibujado, instanciando y conectando las puertas lógicas que ha diseñado anteriormente.
- 1.e) (1 punto) Escriba en VHDL un banco de pruebas que permita visualizar, para todos los posibles valores de las entradas, las salidas de los circuitos diseñados en los Apartados 1.b y 1.d.

Compruebe mediante inspección visual que ambos diseños producen los mismos resultados para todas las combinaciones posibles de entrada. Incluya en la memoria la captura de pantalla de los cronogramas obtenidos al realizar con el simulador de VHDL la simulación del banco de pruebas con los circuitos diseñados en los Apartados 1.b y 1.d.

EJERCICIO 2

Se desea diseñar en VHDL una máquina expendedora capaz de vender tres productos distintos, devolver el cambio sobrante cuando exista sobrepago y permitir la cancelación de una compra en curso.

Los productos disponibles son:

Producto	Precio (euro)
A	1
B	1,50
C	2

La máquina acepta monedas de 50 céntimos y de 1 euro.

La entity del circuito es la siguiente:

```
entity VendingMachine is
  Port (
    DISP_A    : out  STD_LOGIC;
    DISP_B    : out  STD_LOGIC;
    DISP_C    : out  STD_LOGIC;
    CHANGE50  : out  STD_LOGIC;
    ERROR     : out  STD_LOGIC;
    CLK       : in   STD_LOGIC;
    RST       : in   STD_LOGIC;
    M50       : in   STD_LOGIC;
    M100      : in   STD_LOGIC;
    SEL_A     : in   STD_LOGIC;
    SEL_B     : in   STD_LOGIC;
    SEL_C     : in   STD_LOGIC;
    CANCEL    : in   STD_LOGIC);
end VendingMachine;
```

Las señales de entrada y salida tienen la siguiente función:

- DISP_A: indica la dispensación del producto A.
- DISP_B: indica la dispensación del producto B.

- DISP_C: indica la dispensación del producto C.
- CHANGE50: salida utilizada para devolver dinero al usuario. Cada pulso equivale a la devolución de una moneda de 50 céntimos.
- ERROR: indica una condición de error.
- CLK: señal de reloj.
- RST: señal de reinicio síncrono.
- M50: inserción de una moneda de 50 céntimos.
- M100: inserción de una moneda de 1 euro.
- SEL_A: selección del producto A.
- SEL_B: selección del producto B.
- SEL_C: selección del producto C.
- CANCEL: cancelación de la compra y solicitud de devolución del dinero introducido.

El funcionamiento de la máquina se describe a continuación:

1. El usuario seleccionará un único producto mediante una de las señales SEL_A, SEL_B o SEL_C.
2. Si se activan simultáneamente dos o más señales de selección, la máquina deberá activar la señal ERROR.
3. Una vez seleccionado un producto, la máquina entrará en un estado de espera de monedas.
4. El saldo se almacenará internamente en múltiplos de 50 céntimos.
5. Los precios de los productos serán:
 - Producto A: 1 euro.
 - Producto B: 1,50 euros.
 - Producto C: 2 euros.
6. La máquina aceptará monedas de 50 céntimos y de 1 euro.

7. La activación simultánea de las entradas M50 y M100 se considerará una condición de error y provocará la transición al estado de error.
8. Cuando el saldo acumulado alcance o supere el precio del producto seleccionado, la máquina dispensará automáticamente el producto correspondiente.
9. Si el saldo acumulado supera el precio del producto, la diferencia deberá devolverse automáticamente al usuario utilizando la salida CHANGE50.
10. Si durante la compra se activa la entrada CANCEL, la máquina cancelará la operación y devolverá todo el saldo acumulado.
11. La devolución de dinero se realizará mediante pulsos sobre la salida CHANGE50. Cada pulso representará la devolución de una moneda de 50 céntimos.
12. Tras completar una dispensación, una devolución de cambio, una cancelación o una condición de error, la máquina volverá al estado inicial.

La implementación deberá utilizar obligatoriamente los siguientes estados:

- **IDLE**: estado inicial. La máquina espera una selección válida.
- **SELECTED**: espera de pago tras la selección de un producto.
- **DISPENSE**: activación de la salida de dispensación correspondiente.
- **DISPENSE_WAIT**: estado auxiliar para mantener visible la dispensación durante un ciclo adicional.
- **RETURN_MONEY**: estado encargado de devolver dinero mediante pulsos sobre CHANGE50.
- **RETURN_WAIT**: estado auxiliar para mantener visible cada devolución durante un ciclo adicional.
- **ERROR_STATE**: estado que activa la señal ERROR.
- **ERROR_WAIT**: estado auxiliar que mantiene visible la señal de error durante un ciclo adicional.

La codificación binaria de los estados queda a elección del alumno.

La máquina deberá seguir la siguiente lógica de funcionamiento:

1. El estado inicial es `IDLE`.
2. Desde `IDLE`, una selección válida provoca la transición a `SELECTED`.
3. Desde `IDLE`, una selección múltiple provoca la transición a `ERROR_STATE`.
4. En `SELECTED` se acumula el saldo introducido por el usuario.
5. Si se detecta la inserción simultánea de monedas mediante `M50` y `M100`, la máquina pasa a `ERROR_STATE`.
6. Si durante `SELECTED` se activa `CANCEL`, la máquina pasa a `RETURN_MONEY`.
7. Cuando el saldo acumulado alcanza o supera el precio del producto seleccionado, la máquina pasa a `DISPENSE`.
8. Tras la dispensación, la máquina pasa a `DISPENSE_WAIT`.
9. Si existe cambio pendiente de devolución, la máquina pasa a `RETURN_MONEY`. En caso contrario regresa a `IDLE`.
10. La devolución de dinero se realizará alternando los estados `RETURN_MONEY` y `RETURN_WAIT` hasta devolver completamente el saldo pendiente.
11. Cuando no quede dinero por devolver, la máquina regresará al estado `IDLE`.
12. Desde `ERROR_STATE` se transitará a `ERROR_WAIT`.
13. Finalizada la gestión del error, la máquina volverá a `IDLE`.

Tenga en cuenta los siguientes requisitos de diseño:

1. La implementación deberá realizarse mediante una FSM síncrona gobernada por la señal `CLK` activa en el flanco de subida.
2. Todas las señales internas de almacenamiento deberán inicializarse exclusivamente mediante la señal `RST`.
3. Las salidas únicamente podrán activarse en los estados correspondientes.
4. Los estados `DISPENSE` y `DISPENSE_WAIT` deberán mantener visible la dispensación durante al menos un ciclo completo de reloj.
5. Los estados `RETURN_MONEY` y `RETURN_WAIT` deberán mantener visible la devolución de cada moneda durante al menos un ciclo completo de reloj.

6. Los estados `ERROR_STATE` y `ERROR_WAIT` deberán mantener visible la señal de error durante al menos un ciclo completo de reloj.
7. El cambio correspondiente a un sobrepago deberá devolverse automáticamente antes de regresar al estado `IDLE`.

2.a) (3 puntos)

Diseñe en VHDL la `architecture` que describa el comportamiento de la máquina expendedora. El alumno podrá realizar las consideraciones de diseño que estime oportunas para resolver situaciones no especificadas explícitamente en el enunciado, siempre que dichas decisiones no entren en contradicción con los requisitos funcionales establecidos ni alteren el comportamiento exigido de la máquina.

2.b) (3 puntos)

Programe en VHDL el banco de pruebas del circuito diseñado en el apartado anterior.

El banco de pruebas deberá comprobar, como mínimo, los siguientes escenarios:

- Compra exacta del producto A.
- Compra exacta del producto B.
- Compra exacta del producto C.
- Compra del producto B con sobrepago y devolución automática del cambio.
- Cancelación de una compra tras introducir una moneda de 50 céntimos.
- Selección simultánea de varios productos.
- Inserción simultánea de monedas de 50 céntimos y 1 euro.

Las comprobaciones deberán realizarse de forma automática mediante sentencias `assert` y deberán generar mensajes descriptivos cuando el comportamiento observado no coincida con el esperado.

Al finalizar la simulación, el banco de pruebas deberá mostrar un mensaje indicando que todas las pruebas han sido ejecutadas correctamente.

Incluya en la memoria el cronograma obtenido al realizar la simulación.