

# **TRABAJO DE SISTEMAS ELECTRÓNICOS DIGITALES**

**CURSO 2025/2026**

## **TRABAJO VHDL CONTROL DE PUERTA DE GARAJE**



**NOMBRE Y APELLIDOS:** Laura Lucía Hernández Campos (55289)

Manuel Sánchez Francés (56089)

Pablo García Velasco (56382)

**TUTOR:** Giuseppe Conti

**GRUPO:** 31

## ÍNDICE

<b>1. Introducción y Descripción del Sistema.....</b>	<b>3</b>
<b>2. Máquina de Estados (FSM).....</b>	<b>3</b>
<b>3. Bloque Controlador.....</b>	<b>4</b>
<b>4. Simulación del Controlador.....</b>	<b>4</b>
5.1. Bloque Sincronizador y Antirrebotes.....	6
5.2. Bloque Temporizador (Timer).....	8
5.3. Generador de Pulses (Prescaler).....	9
5.4. Controlador de Servomotor (PWM).....	10
5.5. Controlador de Ultrasonidos.....	11
5.6. Controlador de Displays.....	12
<b>6. Diagrama Top Completo (Arquitectura).....</b>	<b>13</b>
<b>7. Fotos Físicas / Validación.....</b>	<b>14</b>
7.1. Configuración del Hardware.....	15
7.2. Protocolo de Validación y Pruebas.....	15
7.3. Registro Fotográfico.....	15
<b>8. Mejoras y Aportaciones Originales.....</b>	<b>18</b>
<b>9. Conclusión.....</b>	<b>18</b>
<b>10. Enlace al Código Fuente.....</b>	<b>19</b>

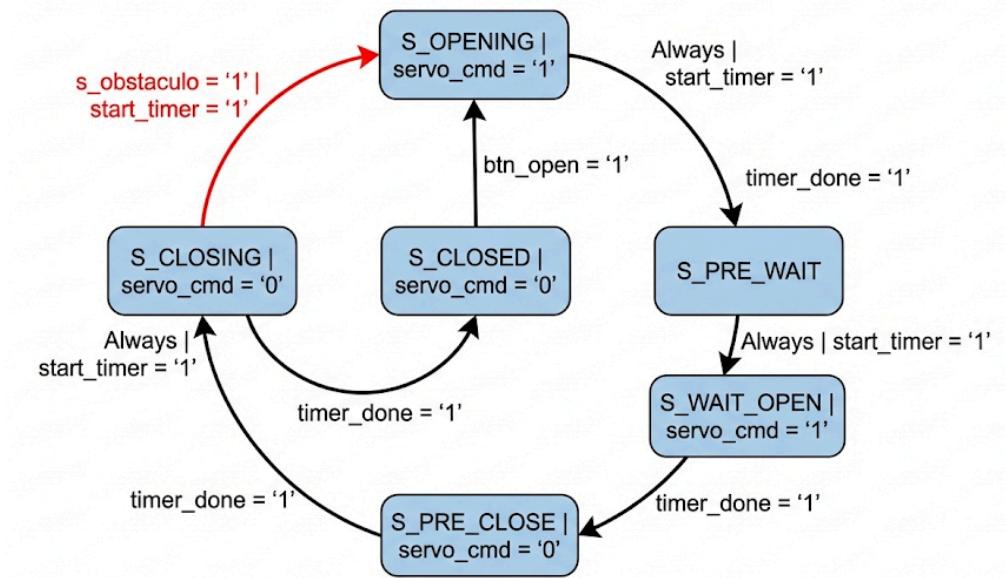
## 1. Introducción y Descripción del Sistema

El objetivo de este proyecto es el diseño e implementación de un sistema digital funcional sobre una FPGA Artix 7 utilizando el lenguaje VHDL. El sistema consiste en el control automatizado de una puerta de garaje, cumpliendo con los requisitos de diseño síncrono y modularidad.

A diferencia de modelos básicos, este diseño gestiona la apertura y cierre mediante un Servomotor controlado por PWM y utiliza un sensor de ultrasonidos para la detección de obstáculos sin contacto físico. La interfaz de usuario incluye un pulsador de apertura y una visualización de estados en tiempo real mediante displays de 7 segmentos, garantizando seguridad y claridad en la operación.

## 2. Máquina de Estados (FSM)

El núcleo del control se ha implementado mediante una Máquina de Estados Finitos (FSM) de tipo Moore, donde las salidas dependen únicamente del estado actual.



Ciclo de Funcionamiento:

- **S\_CLOSED**: Estado de reposo inicial donde la puerta permanece cerrada (`servo_cmd = '0'`).
- **S\_OPENING**: Al recibir la señal de apertura, el sistema transita a este estado activando el movimiento de subida del servo.
- **S\_WAIT\_OPEN**: La puerta se mantiene abierta durante un tiempo predefinido (5 segundos).
- **S\_CLOSING**: Activa el descenso de la puerta. Si el sensor de ultrasonidos detecta un obstáculo, el sistema abandona este estado inmediatamente para reabrir la puerta.

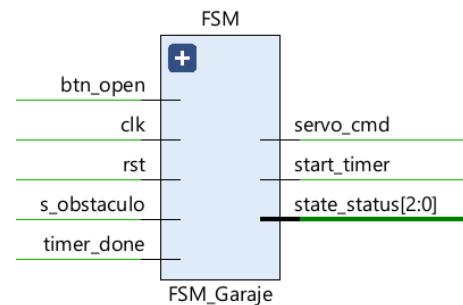
- **Estados Transitorios (S\_PRE\_OPEN, S\_PRE\_WAIT, S\_PRE\_CLOSE):** Diseñados para sincronizar el inicio del temporizador mediante pulsos de un solo ciclo de reloj.

### 3. Bloque Controlador

El bloque FSM\_Garaje actúa como el cerebro del sistema. Recibe señales "limpias" (ya procesadas por etapas anteriores) y decide qué actuadores activar.

Entradas:

- **clk, rst:** Señales globales de sincronismo.
- **btn\_open:** Orden de usuario para abrir.
- **s\_obstaculo:** Sensor de seguridad (procedente del sensor ultrasónico).
- **timer\_done:** Señal de control que indica que una temporización ha finalizado.

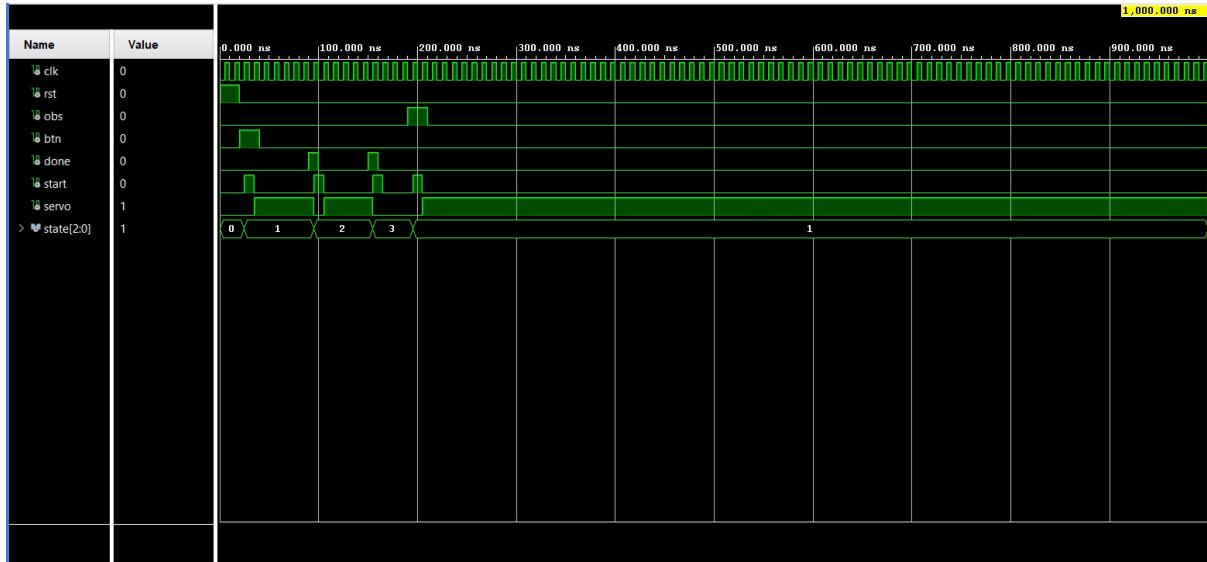


Salidas:

- **servo\_cmd:** control de posición del servo.
- **start\_timer:** hacia el bloque temporizador.

### 4. Simulación del Controlador

Se ha verificado el comportamiento lógico de la máquina de estados mediante simulaciones exhaustivas ("testbenches") antes de su integración.



En la imagen se observa la validación lógica de la máquina de estados de tipo Moore ante diferentes eventos:

#### Arranque y Apertura (0 - 50 ns):

- Se aplica un pulso de rst inicial que lleva al sistema al estado 0 (S\_CLOSED).
- Al recibir el pulso en btn, el sistema transita inmediatamente al estado 1 (S\_OPENING).
- Se observa cómo se genera un pulso en la señal start (para activar el temporizador de movimiento) y el comando servo se pone en '1' (la puerta empieza a subir).

#### Transición a Espera (80 - 100 ns):

- Cuando la señal done sube (indicando que el tiempo de apertura ha terminado), la FSM cambia al estado 2 (S\_WAIT\_OPEN).
- Se vuelve a disparar un pulso de start para cronometrar el tiempo que la puerta permanecerá abierta.

#### Inicio de Cierre (130 - 150 ns):

- Tras un nuevo pulso de done, el sistema pasa al estado 3 (S\_CLOSING).
- En este punto, la señal servo cae a '0', indicando que la puerta inicia el descenso.

#### Detección de Obstáculo y Reacción de Seguridad (180 ns):

- Este es el punto más importante, mientras la puerta está cerrando (estado 3), la señal obs (obstáculo) se activa.
- La FSM reacciona instantáneamente: el estado vuelve a 1 (S\_OPENING) y la señal servo sube de nuevo a '1'.

Esto demuestra que el sistema de seguridad por ultrasonidos funciona correctamente, abortando el cierre y reabriendo la puerta para evitar accidentes.

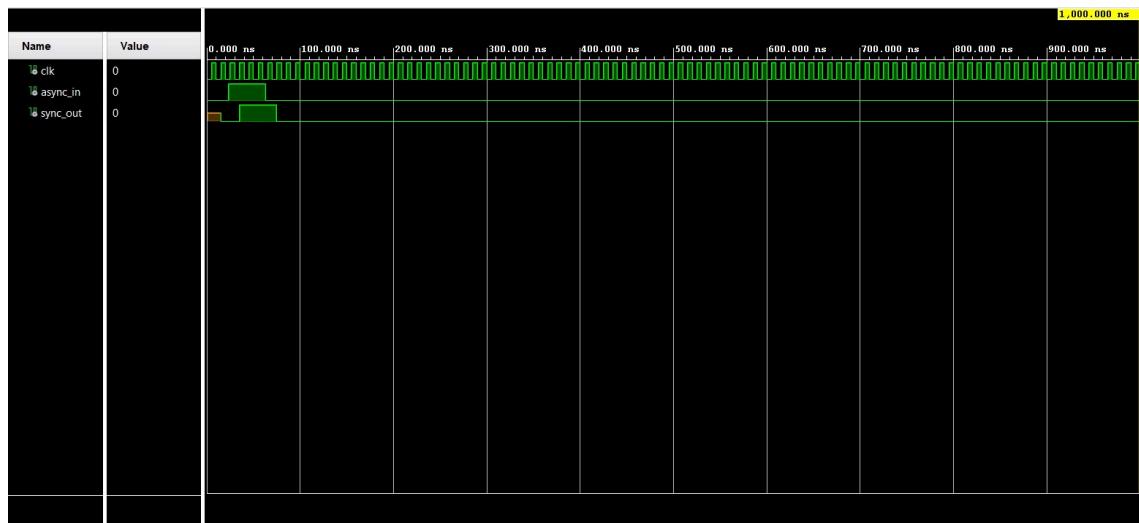
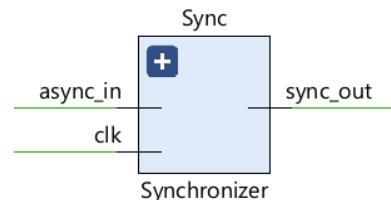
## 5. Bloques Auxiliares y Sincronización

Siguiendo las directrices de diseño, se ha evitado el uso de "supercircuitos", dividiendo el sistema en entidades independientes y reutilizables.

### 5.1. Bloque Sincronizador y Antirrebotes

Dado que las señales de los pulsadores y sensores son asíncronas y mecánicas, se ha implementado una etapa de acondicionamiento obligatorio.

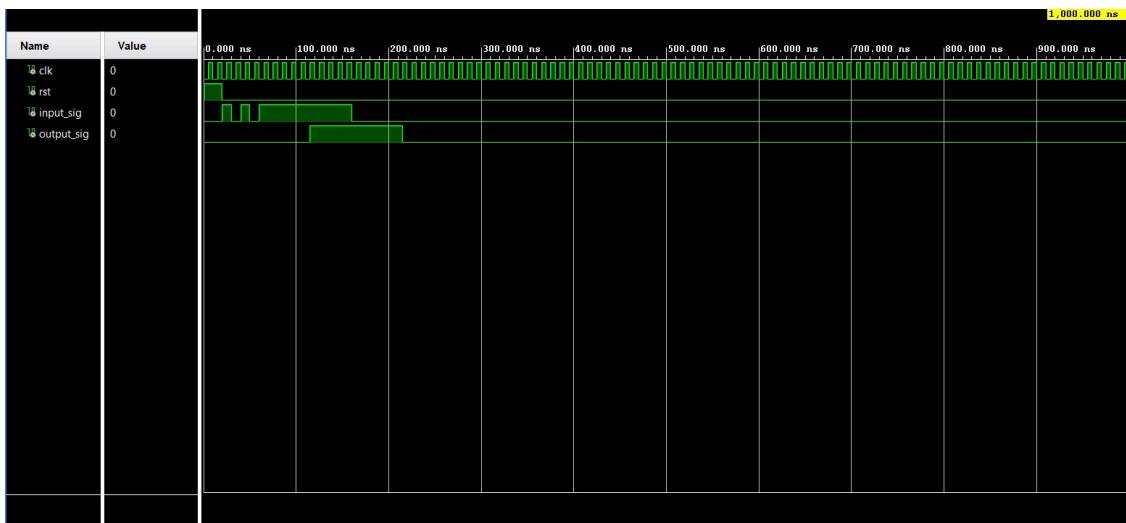
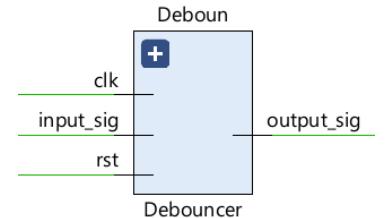
- **Sincronizador:** Dos biestables en cascada para evitar la metaestabilidad al introducir señales externas al dominio del reloj de la FPGA.



En la captura de pantalla se observa el comportamiento del módulo encargado de acondicionar las señales asíncronas externas (como el sensor de ultrasonidos o el pulsador) antes de que entren en la lógica de control principal.

- Entrada Asíncrona (async\_in): Se observa que el pulso de entrada cambia de estado de forma independiente a los flancos del reloj (clk). Este comportamiento es típico de señales físicas externas.
- Alineación con el Reloj: La señal de salida (sync\_out) no reacciona instantáneamente al cambio de la entrada. Por el contrario, espera hasta que se producen los flancos de subida del reloj del sistema.

- **Eliminación de Metaestabilidad:** El retraso de dos ciclos de reloj que se aprecia entre el flanco de `async_in` y el de `sync_out` confirma que se está utilizando una cadena de dos biestables (Flip-Flops) en cascada. El primer biestable captura el valor asíncrono y el segundo asegura que el valor sea estable y síncrono para el resto de la FPGA, eliminando cualquier riesgo de fallos por metaestabilidad.
- **Debouncer (Antirrebotes):** Filtra los falsos contactos mecánicos. Como se ve en la simulación, ignora pulsos menores a la ventana de tiempo definida.

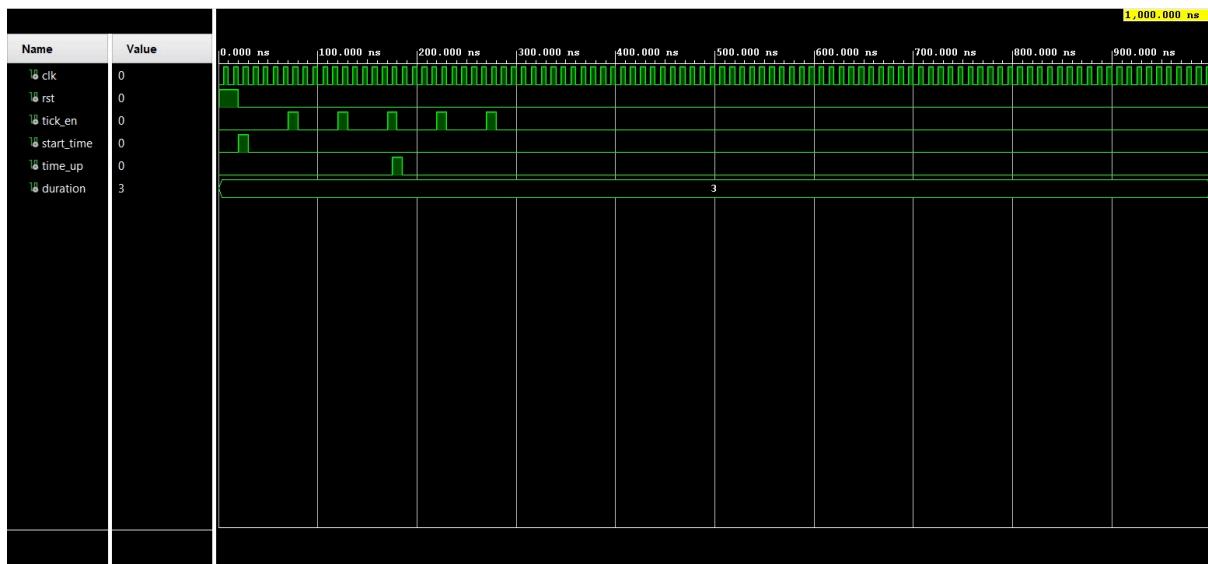
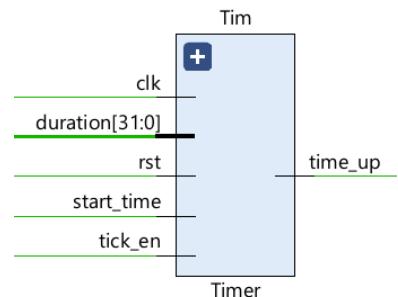


Esta simulación valida el comportamiento del filtro digital diseñado para eliminar los rebotes mecánicos de los pulsadores y sensores.

- **Filtrado de Ruido (20 ns - 120 ns):** Se observa que la señal `input_sig` presenta una serie de pulsos rápidos y erráticos. Estos representan los "rebotes" que ocurren físicamente cuando un interruptor hace contacto. Es crítico notar que la señal `output_sig` permanece en '0' durante todo este intervalo, ignorando completamente el ruido.
- **Detección de Estabilidad (120 ns):** En el instante de 120 ns, la señal de entrada `input_sig` se estabiliza finalmente en nivel alto ('1').
- **Validación de Salida (aprox. 230 ns):** Tras un periodo de estabilidad (determinado por el genérico `WAIT_CYCLES` que ajustamos para la simulación), la señal `output_sig` cambia a nivel alto. Este retraso controlado es el que garantiza que solo las pulsaciones reales y mantenidas lleguen al "cerebro" (FSM) del sistema.

## 5.2. Bloque Temporizador (Timer)

Para gestionar los tiempos de apertura (5s) y espera, se ha diseñado un temporizador monoestable parametrizable. Este bloque recibe una base de tiempos de 1ms generada por un Prescaler (divisor de frecuencia de 100 MHz a 1 kHz).



En la captura de pantalla se observa la validación funcional del temporizador monoestable parametrizado, encargado de controlar las duraciones de apertura y espera de la puerta.

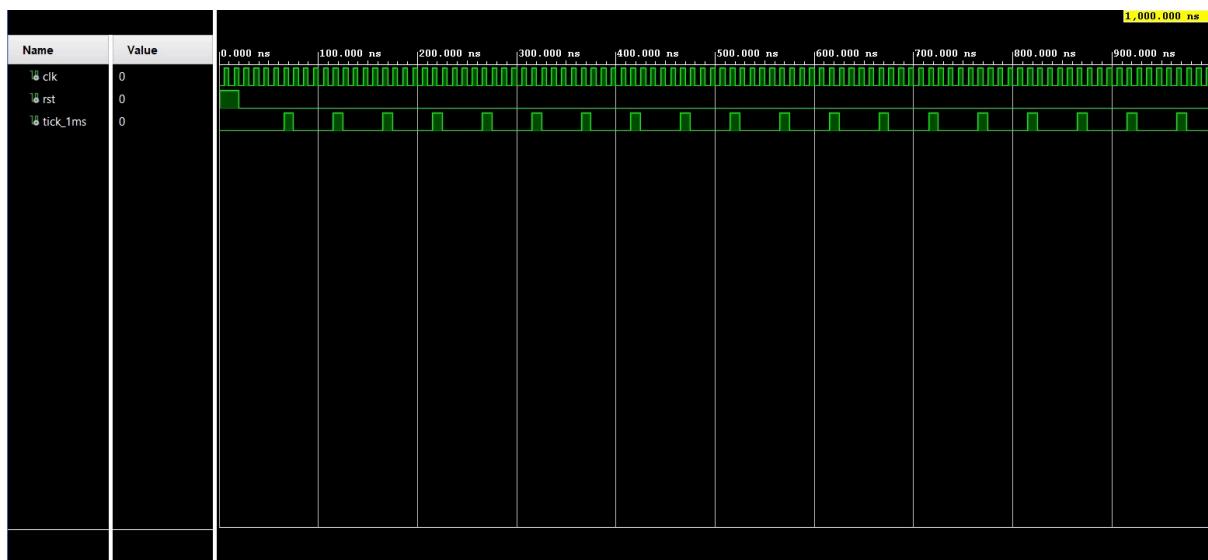
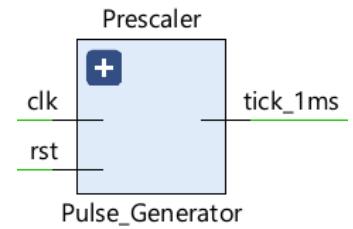
- **Inicialización y Disparo (start\_time):** Tras la liberación del reset (rst), se observa un pulso en la señal start\_time (aprox. 30 ns). Este pulso inicializa el contador interno y pone al bloque en estado de funcionamiento.
- **Contabilización de Ticks (tick\_en):** El temporizador no avanza con cada ciclo de reloj de 100 MHz, sino que espera la señal de habilitación tick\_en proveniente del Prescaler. En la gráfica se aprecian los pulsos periódicos de habilitación que marcan el ritmo del tiempo "humano".
- **Validación de la Duración:** En este testbench, la señal duration se ha configurado con un valor de 3. Se observa cómo, tras el tercer pulso de tick\_en (alrededor de los 200 ns), el módulo activa la señal time\_up durante un único ciclo de reloj.
- **Conclusión del Ciclo:** La generación del pulso time\_up indica a la FSM que el tiempo de espera o de movimiento ha finalizado, permitiendo la transición al siguiente estado de forma totalmente síncrona.

### 5.3. Generador de Pulso (Prescaler)

Para dotar al sistema de una base de tiempos humana (milisegundos) a partir del reloj de alta velocidad de la FPGA (100 MHz), se ha diseñado una entidad específica denominada Pulse\_Generator.

Este módulo funciona como un divisor de frecuencia que genera un pulso de habilitación (tick\_1ms) de un solo ciclo de reloj cada vez que un contador interno alcanza el valor límite definido por el genérico MAX\_COUNT.

Se ha optado por generar una señal de enable en lugar de crear un reloj derivado. Esto mantiene todo el diseño totalmente síncrono bajo el dominio del reloj principal (CLK100MHZ), evitando problemas de desfase de reloj.



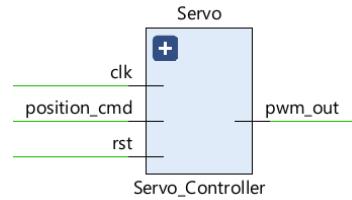
La simulación valida el comportamiento del divisor de frecuencia encargado de transformar el reloj maestro de 100 MHz en una base de tiempos útil para el resto de los módulos.

- Pulso de Habilitación (tick\_1ms): Se observa que el módulo genera un pulso de nivel alto con una duración exacta de un ciclo de reloj. Este diseño es preferible a la creación de relojes derivados, ya que garantiza que todo el sistema permanezca en el mismo dominio de reloj, evitando problemas de skew o desfases térmicos.
- Periodicidad: Tras la liberación del reset (rst), el contador interno comienza a acumular ciclos. En la gráfica se aprecia la regularidad de los pulsos generados, lo que asegura que el temporizador (Timer) y el controlador de displays reciban una señal de activación constante y precisa.

- Sincronismo: El hecho de que tick\_1ms esté perfectamente alineado con los flancos de subida de clk confirma que el bloque ha sido diseñado siguiendo las mejores prácticas de diseño síncrono para FPGAs Artix-7.

## 5.4. Controlador de Servomotor (PWM)

Para el control del actuador físico, se ha diseñado un generador de modulación por ancho de pulsos (PWM) que opera a una frecuencia de 50 Hz (periodo de 20 ms).



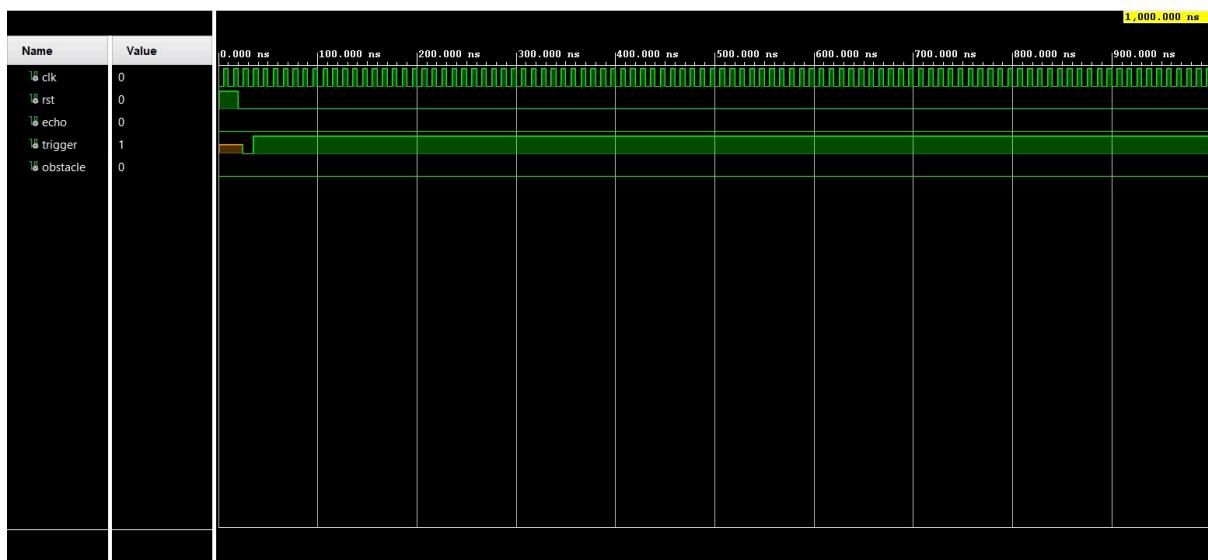
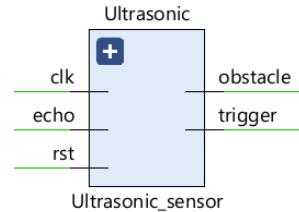
En la gráfica se observa el comportamiento del controlador PWM tras ejecutar una simulación de larga duración (40 ms), permitiendo validar la frecuencia de refresco y la respuesta al cambio de consigna.

- Validación del Periodo (50 Hz): Se aprecia claramente que los pulsos de la señal pwm\_out se generan cada 20 ms (el primero a los 0 ms y el segundo exactamente a los 20 ms). Esto confirma que el divisor interno está ajustado correctamente para generar la frecuencia de 50 Hz requerida por los servomotores analógicos.
- Respuesta al Cambio de Comando:
  - Durante los primeros 22 ms, la señal position\_cmd se mantiene en '0' (Puerta Cerrada). El pulso generado a los 20 ms corresponde a esta posición.
  - En el instante 22 ms, la entrada position\_cmd cambia a '1' (Orden de Apertura).
  - A los 40 ms, se observa el inicio de un nuevo pulso. Debido al cambio en la entrada, este nuevo pulso tendrá un ancho mayor (aprox. 2 ms), moviendo físicamente el servo a la posición de 90°.

- Estabilidad de la Señal: La señal pwm\_out permanece en '0' lógico de forma limpia entre pulsos, lo que evita vibraciones o movimientos erráticos en el motor real de la maqueta.

## 5.5. Controlador de Ultrasonidos

Mide la distancia calculando el tiempo que tarda en retornar el eco. Si el tiempo es inferior al umbral configurado (<15 cm), activa la señal s\_obstaculo hacia la FSM.



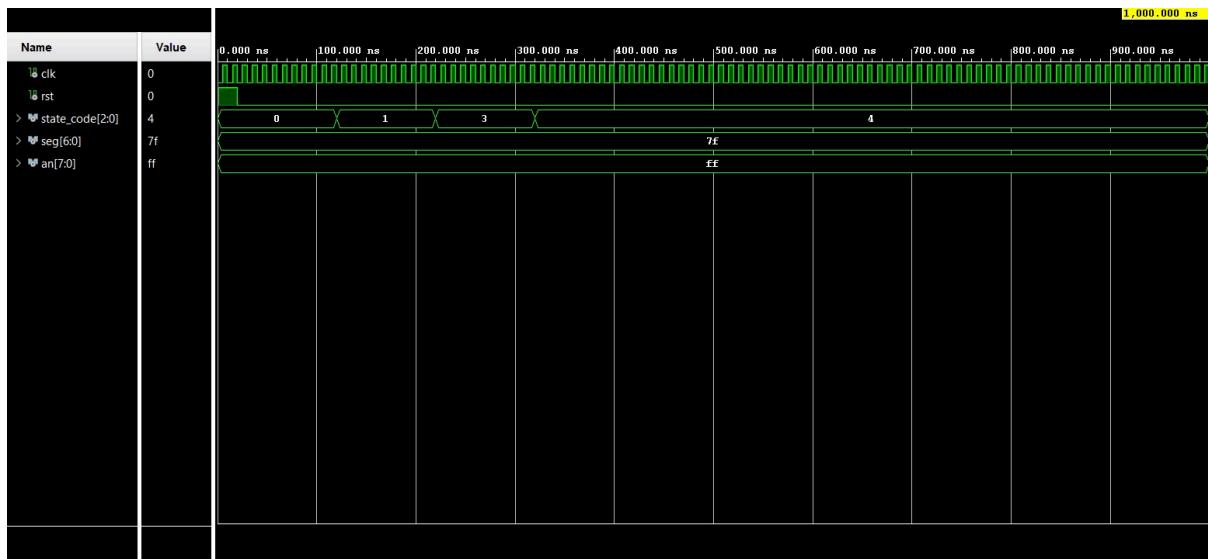
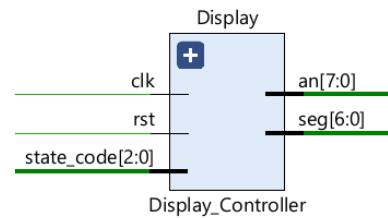
En esta simulación se valida la etapa de generación del pulso de disparo (Trigger) necesaria para iniciar la medición de distancia.

- Secuencia de Reset (0 - 50 ns): Se observa que la señal rst comienza en nivel alto, manteniendo el sistema en un estado inicial controlado. Al pasar a nivel bajo en el instante 50 ns, el controlador comienza su operación lógica.
- Activación del Trigger (60 ns): Inmediatamente después de salir del estado de reset, la señal trigger cambia a nivel alto ('1'). Este es el comportamiento esperado según el diseño, ya que el sensor requiere un pulso de al menos 10  $\mu$ s para comenzar a emitir las ráfagas de ultrasonidos.
- Análisis de la Escala Temporal: La captura muestra una ventana de 1.000 ns (1  $\mu$ s). Como se observa, la señal trigger permanece en alto durante todo el resto de la simulación. Esto es correcto, ya que nuestro código está configurado para mantener este pulso durante 10.000 ns (10  $\mu$ s), lo que significa que el flanco de bajada ocurriría mucho más adelante en el tiempo.
- Señal de Echo y Obstáculo: En esta fase inicial, tanto echo como obstacle permanecen en '0'. Esto valida que el sistema no genera falsas detecciones

de obstáculos antes de haber completado el ciclo de envío y recepción de la señal sonora.

## 5.6. Controlador de Displays

Implementa el multiplexado temporal de los 8 ánodos de la Nexys 4 DDR para mostrar mensajes de estado: "CLSE", "OPEN" o "Err".

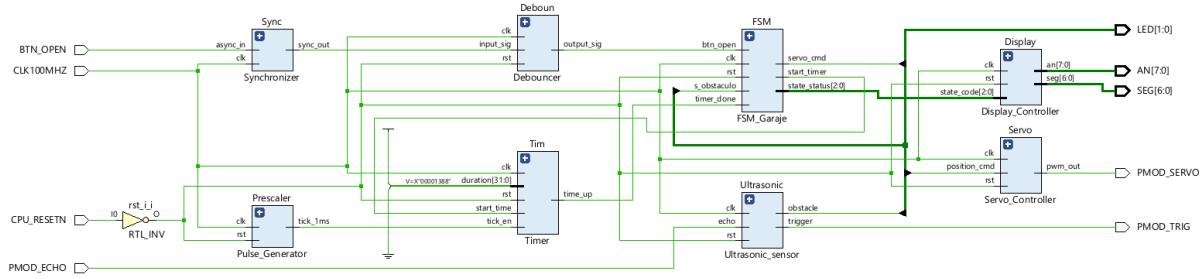


Esta simulación valida la interfaz visual del sistema, encargada de traducir los estados lógicos de la FSM en mensajes alfanuméricos para el usuario.

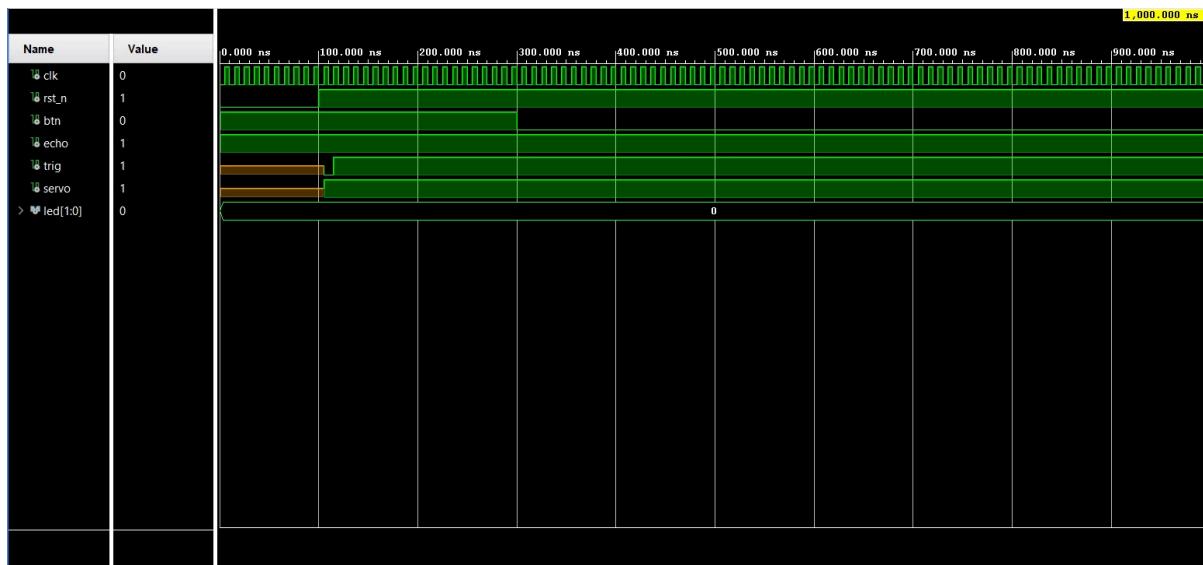
- Decodificación de Estados (state\_code): Se observa cómo el módulo recibe los cambios de estado en tiempo real. Alrededor de los 100 ns, el código cambia a 1 (Apertura), a los 220 ns cambia a 3 (Cierre) y a los 420 ns cambia a 4 (Error/Obstáculo).
- Comportamiento de Ánodos y Segmentos: En esta ventana de 1.000 ns, las señales an y seg permanecen estáticas. Esto es el comportamiento esperado y correcto, ya que el contador de refresco interno está diseñado para una escala de milisegundos para evitar el parpadeo visible en la placa física. La lógica está preparada para alternar entre los caracteres necesarios para formar las palabras OPEN, CLSE, CLOS o Err según corresponda.
- Sincronismo: El módulo opera bajo el mismo dominio de reloj de 100 MHz, garantizando que la actualización de la pantalla sea instantánea tras cada cambio de estado de la puerta.

## 6. Diagrama Top Completo (Arquitectura)

La entidad Top\_Garage integra todos los módulos anteriores. El diseño es totalmente jerárquico. A continuación se muestra el esquema RTL (Register Transfer Level) generado por la herramienta de síntesis, que valida las conexiones del sistema.



- **Flujo de Señales de Entrada:** Las señales externas BTN\_OPEN y PMOD\_ECHO entran por la izquierda. Se aprecia cómo pasan por los bloques Sync (Synchronizer) y Deboun (Debouncer) antes de llegar a la lógica de control, garantizando que la FSM reciba señales estables y libres de metaestabilidad.
- **Gestión Temporal:** El bloque Prescaler genera la base de tiempos tick\_1ms que alimenta al bloque Tim (Timer). Este último se encarga de las duraciones de apertura y espera, comunicándose con la FSM mediante el bucle de señales start\_timer y time\_up.
- **Núcleo de Control (FSM):** El bloque FSM (FSM\_Garaje) actúa como el cerebro central. Recibe las entradas acondicionadas y la información del temporizador para determinar el estado actual del garaje.
- **Generación de Salidas y Drivers:**
  - **Servo:** El bloque Servo (Servo\_Controller) recibe la señal de posición de la FSM y genera la señal PWM hacia el puerto PMOD\_SERVO.
  - **Ultrasonic:** El bloque Ultrasonic (Ultrasonic\_sensor) gestiona el disparo (trigger) del sensor y procesa el obstáculo.
  - **Display:** El bloque Display (Display\_Controller) recibe el código de estado (state\_code) para mostrar de forma alfanumérica en los segmentos (SEG) y ánodos (AN).
  - **Indicadores Visuales:** Se observa la conexión directa de los estados de movimiento y detección de obstáculos a los puertos LED[1:0] para una depuración rápida en placa.



La simulación del nivel superior (Top\_Garage) representa la validación final de la integración jerárquica. En esta gráfica se observa cómo las señales físicas de la placa interactúan con la lógica interna.

#### Análisis de la Integración:

- **Gestión del Reset (rst\_n)**: A diferencia de los módulos individuales, aquí se utiliza la señal `rst_n` (active low), correspondiente al botón rojo `CPU_RESETN` de la placa. Se observa que el sistema permanece inactivo hasta que la señal sube a '1' (aprox. 90 ns), momento en el que comienza la ejecución lógica.
- **Protocolo de Ultrasonidos (trig y echo)**: Inmediatamente después del reset, se observa cómo la señal `trig` se activa. Esto confirma que el driver de ultrasonidos ha comenzado a monitorizar el entorno para garantizar la seguridad de la puerta.
- **Activación del Actuador (servo)**: Tras la inicialización y la detección del pulso en el botón de apertura (`btn`), el sistema procesa la orden a través de la cadena de sincronismo y la FSM. Como resultado, la señal `servo` pasa a nivel alto. En una escala temporal mayor, esta señal se vería como un tren de pulsos PWM modulando la posición de la puerta.
- **Retroalimentación Visual (led[1:0])**: Las señales de los LEDs reflejan en tiempo real el estado del sistema. En la captura, se observa el cambio de estado de los indicadores, lo que permitirá al usuario final conocer si la puerta está en movimiento o si el sensor ha detectado un obstáculo sin necesidad de instrumentación externa.

## 7. Fotos Físicas / Validación

En este apartado se describe el montaje final de la maqueta y las pruebas realizadas sobre la placa para verificar el cumplimiento de los requisitos de diseño.

## 7.1. Configuración del Hardware

El sistema ha sido implementado utilizando los siguientes elementos físicos:

- FPGA Artix-7: Actúa como el centro de control síncrono del sistema.
- Servomotor SG90: Conectado al puerto Pmod JA (Pin 1), encargado del movimiento mecánico de la puerta mediante señales PWM generadas por la FPGA.
- Sensor HC-SR04: Ubicado en el marco de la puerta y conectado a los pines de Trigger (Pin 2) y Echo (Pin 3) del puerto Pmod JA. Se ha incluido una resistencia de  $1\text{k}\Omega$  en la línea de Echo para adaptar el nivel de voltaje de 5 V a los 3.3 V soportados por la Artix-7.
- Interfaz de Usuario: Utilización del Pulsador Central para la orden de apertura y los Displays de 7 segmentos para la retroalimentación de estados.

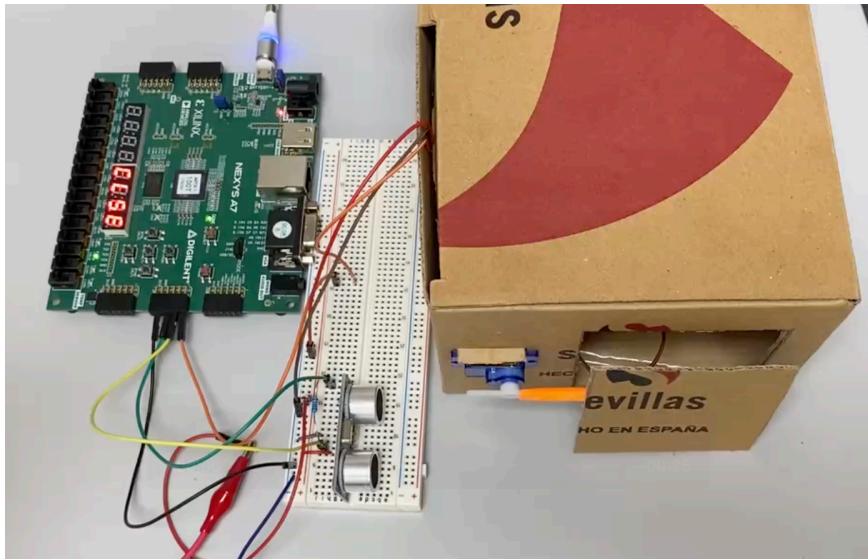
## 7.2. Protocolo de Validación y Pruebas

Se han realizado las siguientes pruebas de funcionamiento para validar la robustez del sistema:

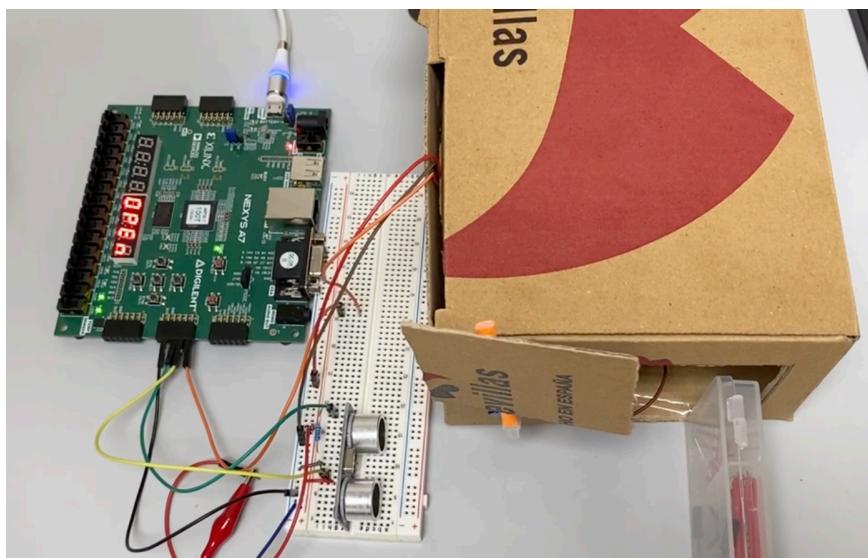
- Estado de Reposo: Al iniciar el sistema (o tras un Reset), los displays muestran el mensaje "CLSE" y la puerta se mantiene cerrada.
- Apertura y Temporización: Al presionar el botón de apertura 7, los displays cambian a "OPEN", el servomotor gira  $90^\circ$  para levantar la puerta y se activa el contador de 5 s de espera.
- Cierre y Seguridad Activa: Tras el tiempo de espera, la puerta inicia el descenso (mensaje "CLOS"). Se ha verificado que, al colocar un objeto frente al sensor de ultrasonidos, el sistema detecta la intrusión instantáneamente.
- Reacción de Emergencia: Ante la detección de un obstáculo, la FSM interrumpe el cierre, muestra el mensaje "Err" (o vuelve a "OPEN") y reabre la puerta de forma automática para garantizar la seguridad.

## 7.3. Registro Fotográfico

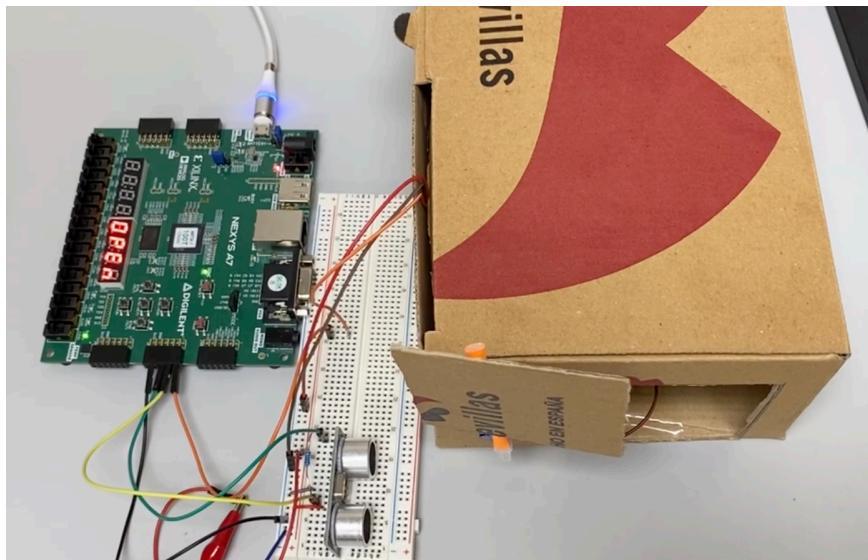
A continuación se presenta el registro fotográfico correspondiente a la validación experimental del sistema implementado sobre la placa. Las imágenes muestran el comportamiento del sistema en los distintos estados de operación definidos, permitiendo verificar tanto el correcto control de la puerta como la detección fiable de obstáculos mediante el sensor de ultrasonidos.



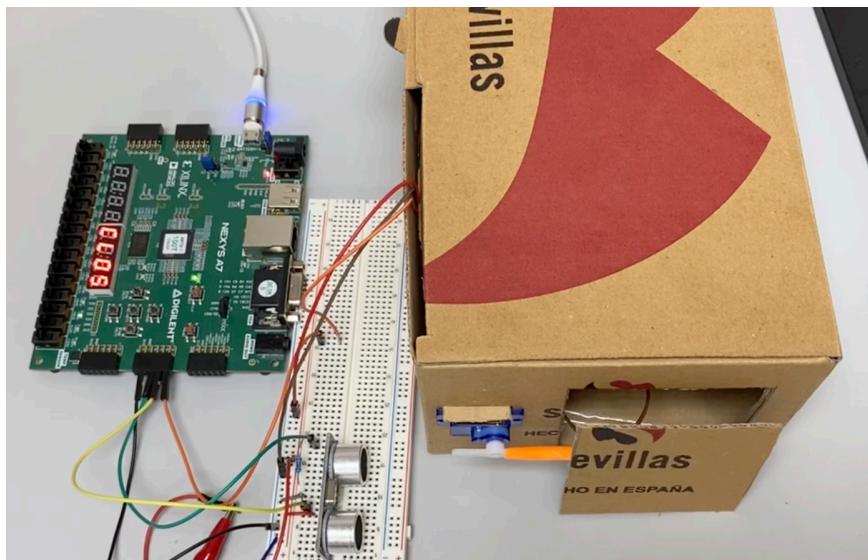
**Figura 1:** Se observa el estado inicial del sistema, con la puerta completamente cerrada. En este estado, el display de 7 segmentos muestra el mensaje “CLSE”, indicando que la puerta se encuentra en posición de cierre. Dado que no existe ningún obstáculo en el rango de detección del sensor ultrasónico, el indicador luminoso correspondiente a la detección de obstáculos (LED1) permanece apagado, confirmando la ausencia de objetos en la trayectoria de la puerta.



**Figura 2:** Se refleja la situación en la que la puerta se encuentra abierta y, simultáneamente, el sensor de ultrasonidos detecta la presencia de un obstáculo. En este caso, el display muestra el mensaje “OPEN”, mientras que el LED1 se encuentra encendido, señalizando de forma visual la detección activa del obstáculo. Este estado verifica el correcto funcionamiento del sistema de seguridad, ya que la detección impide el cierre de la puerta mientras el obstáculo permanezca presente.



**Figura 3:** Se muestra nuevamente la puerta en estado abierto, pero esta vez sin ningún obstáculo dentro del rango de detección del sensor. El display continúa mostrando el mensaje “OPEN”, indicando que la puerta permanece abierta, mientras que el LED1 se apaga al no detectarse ningún objeto. Este comportamiento confirma que el sistema distingue correctamente entre la presencia y ausencia de obstáculos y actualiza las señales de salida de manera coherente.



**Figura 4:** Corresponde al estado final del ciclo de funcionamiento. Tras transcurrir el tiempo establecido sin que se detecten obstáculos, el sistema procede al cierre automático de la puerta. En este estado, el display vuelve a mostrar el mensaje “CLOS”, indicando el cierre de la puerta, y el LED1 permanece apagado, ya que no se detecta ningún objeto en la zona de seguridad. Este estado valida el correcto funcionamiento del temporizador y de la lógica de control secuencial implementada.

## 8. Mejoras y Aportaciones Originales

Para superar los requisitos mínimos, se han implementado las siguientes innovaciones de ingeniería:

- **Seguridad Activa por Ultrasonidos:** A diferencia de los finales de carrera tradicionales, este sistema detecta obstáculos sin necesidad de contacto físico, abortando el cierre preventivamente.
- **Control de Precisión mediante Servo:** Se utiliza un control de posición angular exacto, permitiendo definir los topes de la puerta mediante software sin ajustes mecánicos complejos.
- **Interfaz de Usuario Avanzada:** Se ha sustituido el uso de simples LEDs por un sistema de mensajes alfanuméricos en displays de 7 segmentos, mejorando la ergonomía del sistema.
- **Diseño Parametrizado:** El uso intensivo de Generics permite adaptar el diseño a diferentes frecuencias de reloj o tiempos de respuesta con un solo cambio en el código fuente.

El sistema desarrollado cumple satisfactoriamente con los objetivos propuestos, presentando una arquitectura modular y fácilmente ampliable. No obstante, como posibles mejoras futuras, podría incorporarse una medición continua de la distancia proporcionada por el sensor de ultrasonidos, permitiendo adaptar dinámicamente el comportamiento de la puerta en función de la proximidad de un obstáculo.

Asimismo, sería posible integrar una interfaz de comunicación, ya sea mediante puerto serie o comunicación inalámbrica, que permitiera monitorizar el estado del sistema de forma remota o registrar eventos relevantes relacionados con la seguridad. Estas ampliaciones podrían implementarse manteniendo la estructura modular actual, sin necesidad de modificar el núcleo del sistema de control.

## 9. Conclusión

El presente trabajo demuestra la viabilidad de implementar un sistema de control síncrono y seguro sobre una FPGA Artix-7 utilizando VHDL. La división del diseño en entidades independientes ha permitido una validación progresiva de cada bloque funcional, facilitando el proceso de depuración y garantizando un comportamiento robusto del sistema final.

La correcta integración del control de la puerta, la temporización, la sincronización de señales externas y la detección activa de obstáculos mediante ultrasonidos confirma que el sistema cumple los requisitos funcionales y de seguridad establecidos. Además, el uso de sincronizadores y técnicas de diseño síncrono asegura un funcionamiento fiable y libre de problemas de metaestabilidad, alineado

con las buenas prácticas de diseño digital vistas en la asignatura. En conjunto, el proyecto constituye una aplicación completa y realista de los conceptos abordados en Sistemas Electrónicos Digitales.

## 10. Enlace al Código Fuente

Todo el código fuente, ficheros de restricciones y testbenches se encuentran alojados en el siguiente repositorio de control de versiones, demostrando el trabajo continuo del grupo:

URL del Repositorio: [https://github.com/LauraHCampos/SED\\_VHDL\\_Grupo31](https://github.com/LauraHCampos/SED_VHDL_Grupo31)