Controlador de servo motor mediante PPM

Autor:
Doddy Castillo Caicedo

Profesor:
Ing. Nicolás Álvarez (FIUBA)

Esta trabajo fue realizado en Octubre 2025.

Índice

1 Introducción

El desarrollo de sistemas digitales orientados al control de actuadores requiere técnicas de modulación y temporización precisas que garanticen una comunicación confiable entre los módulos de transmisores y receptores. En este contexto, la Modulación por Posición de Pulsos (PPM) constituye una alternativa eficiente para la transmisión de señales de control en aplicaciones embebidas y de radiofrecuencia, al codificar la información en la posición temporal de los pulsos dentro de un tren periódico. El presente proyecto tiene como finalidad diseñar, implementar y verificar un sistema de generación y recepción de señales PPM utilizando el lenguaje de descripción de hardware VHDL en la plataforma FPGA. El sistema integra dos módulos principales:

1. PPM Generator, encargado de producir una secuencia de pulsos codificados según diferentes canales de control.
2. PPM Receiver, responsable de decodificar la señal recibida y reconstruir señales PWM (Pulse With Modulation) compatibles con servomotores tipo SG90, ampliamente utilizados en aplicaciones de robótica.

El trabajo comprende desde el diseño de los módulos digitales y de la asignación de pines mediante el archivo de restricciones (.xdc), hasta la simulación funcional con Vivado y la verificación práctica en la FPGA mediante la observación de las señales generadas en un analizador lógico y el movimiento del servomotor. Asimismo, se incluye un testbench VHDL para validar el comportamiento temporal de la señal PPM y comprobar su correcta decodificación. Esta implementación permite comprender de forma práctica los principios de sincronización, modulación temporal y control digital de actuadores, así como fortalecer las competencias en un diseño de sistemas embebidos basados en FPGA, el uso del entorno Vivado, y la metodología de verificación mediante simulación y prueba en hardware real.

2 Objetivos

Diseñar e implementar un sistema digital de generación y recepción de señales PPM en VHDL sobre la FPGA Arty Z7-10, que permita controlar un servomotor SG90 mediante la conversión de la señal PPM a PWM, garantizando una comunicación precisa, estable y verificable mediante simulación y pruebas prácticas.

3 Diseño e implementación

1. Arquitectura general del sistema: el sistema implementado está compuesto por tres bloques principales, diseñados y simulación en VHDL dentro del entorno Vivado
    - PPM Generator: módulo encargado de generar la señal PPM codificada con diferentes canales de control.
    - PPM Receiver: módulo que decodifica la señal PPM y generar señales PWM individuales.

- Servo interface (PPM Servo): módulo final que aplica la señal PWM reconstruida al microservo SG90

El objetivo de esta arquitectura es lograr una comunicación unidireccional y determinística, donde un solo tren de pulsos (PPM) transporte la información de varios canales PWM de control.

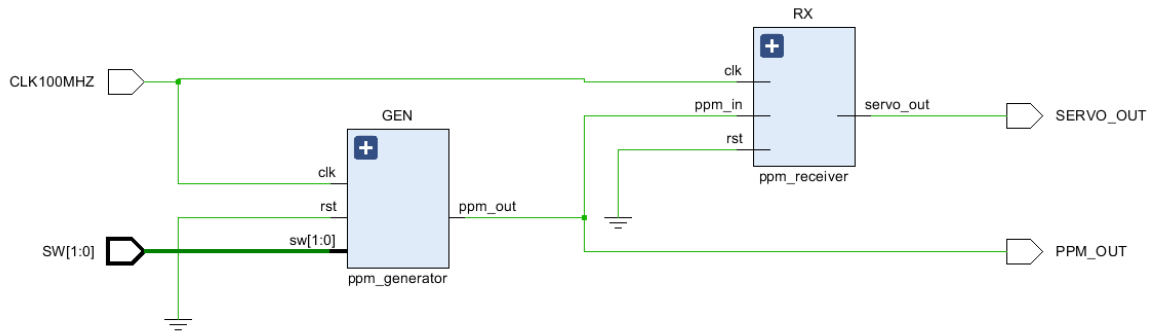


Figura 1: Diagrama general del sistema PPM implementado en FPGA.

2. Módulo PPM Generator: produce un tren de pulsos periódicos con frecuencia de 50 Hz (20 ms por frame). Dentro de cada frame se genera un pulso corto de 300us, cuya posición depende de la entrada SW(1 downto 0):

SW	Posición del pulso	Equivalente
01	1000 us	Mínimo
00/11	1500 us	Centro
10	2000 us	Máximo

Cuadro 1: Resumen de módulos implementados en el sistema.

La modulación por posición de pulso (PPM) se logra manteniendo el ancho del pulso constante y desplazando su posición temporal dentro del período total.

2.1. Diseño digital: el bloque se implementa como un sistema secuencial síncrono que usa:

- Un contador de reloj (`clk_cnt`) para generar ticks de 1 µs a partir de los 100 MHz del reloj base.
- Un contador de microsegundos (`us_cnt`) para medir los 20 ms de cada frame.
- Un comparador que activa la salida `ppm_out` cuando `us_cnt` se encuentra dentro del rango [`pos_us`, `pos_us + PULSE_WIDTH`].

2.2. Implementación VHDL

- clk: reloj de 100 MHz
- rst: reset síncrono

- sw: selección de posición
- `ppm_out`: salida modulada

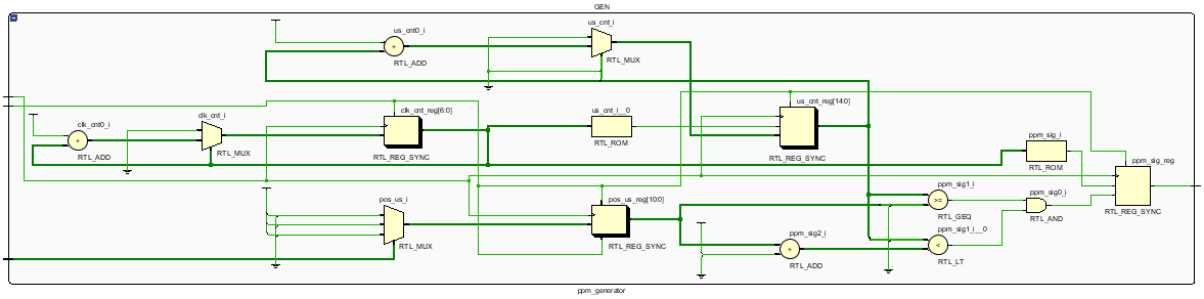


Figura 2: Diagrama del generador de PPM.

3. Módulo PPM Receiver: mide la posición del pulso PPM recibido (`ppm_in`) y genera una señal PWM proporcional, destinada a un actuador (servo). Su función es decodificar la información temporal contenida en la PPM, reproduciendo la misma duración pero en forma de PWM (donde el ancho del pulso representa el valor).

   3.1. Diseño digital

   - Un contador de tiempo que mide la duración entre el flanco de subida del pulso PPM y el flanco de bajada.
   - rst: reset síncrono
   - Un comparador para mantener `ppm_out` activo mientras el tiempo actual esté dentro del valor medido.
   - Un registro de captura para almacenar la duración detectada y usarla en el siguiente PWM.

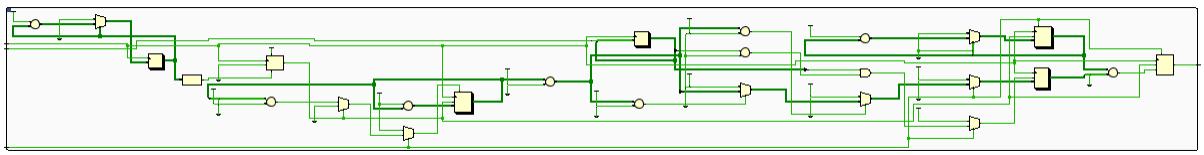


Figura 3: Diagrama del recevidor de PPM.

4. Módulo Servo Driver: el Servo Driver recibe la señal PWM generada por el receptor y la aplica al pin de control del servo SG90, el cual convierte la anchura del pulso (1–2 ms) en una posición angular (0°–180°).

   4.1. Diseño digital

   - Un comparador que limita el rango de pulsos a los `1000_2000 us`.
   - Un registro de estado para evitar jitter en el movimiento del servo.

4 Simulación del sistema

El archivo testbench inicializa los módulos `ppm_generator`, `ppm_receiver` y `ppm_servo`, y permite observar las señales principales:

- CLK100MHZ: reloj del sistema.
- SW[1:0]: entrada de seleccion de posición.
- `PPM_OUT`: señal PPM generada por el módulo transmisor.
- `SERVO_OUT`: señal PWM decodificada por el receptor (salida hacia el servo).

1. Forma de onda obtenida: en la figura siguiente se presenta la forma de onda obtenida en la simulación del sistema completo:

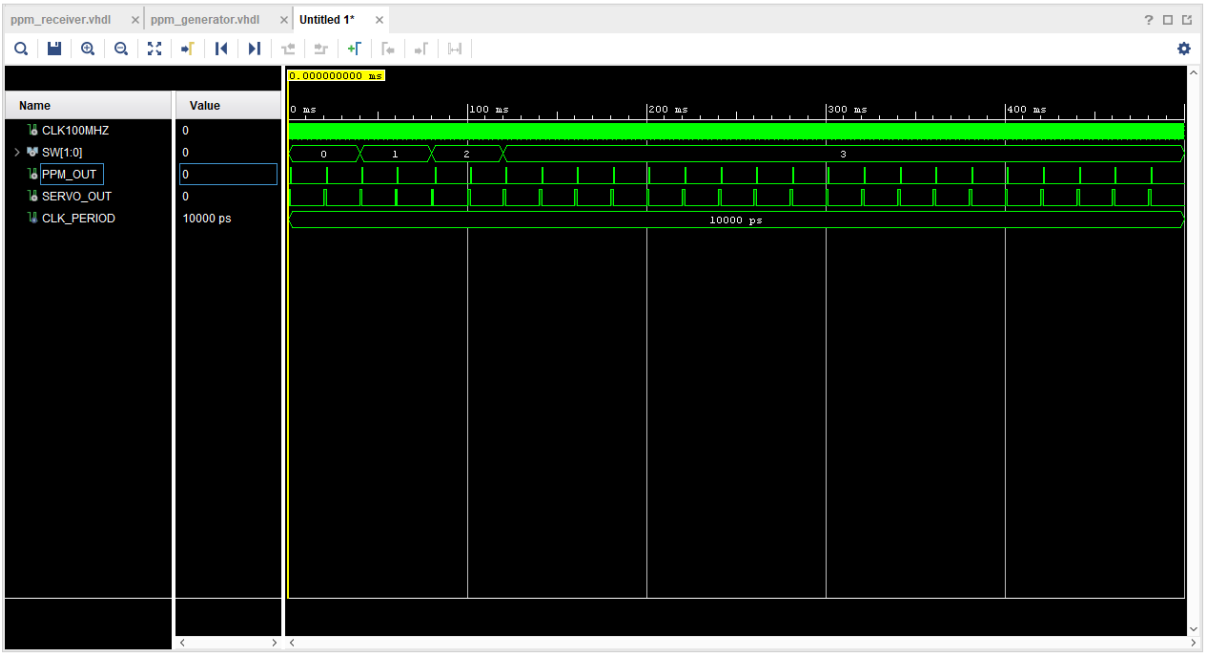


Figura 4: Simulación de la señal PPM y salida del servo

2. Análisis de la simulación

   2.1. Señal del reloj (CLK100MHZ): la señal de reloj se mantiene estable con un periodo de 10ns, lo que equivale a un frecuencia de 100 MHz. A partir de este reloj se genera el tick de 1 µs utilizado por los contadores internos del módulo generador.

   2.2. Señal PPM (`PPM_OUT`): la simulación se observa un pulso de corta duración (300 µs) generado al inicio del frame. El pulso cambia de posición dependiendo del valor de los interruptores SW[1:0].

   - Un solo pulso activo al inicio del frame 1.5ms.

- El ancho de pulso constante (300 µs).

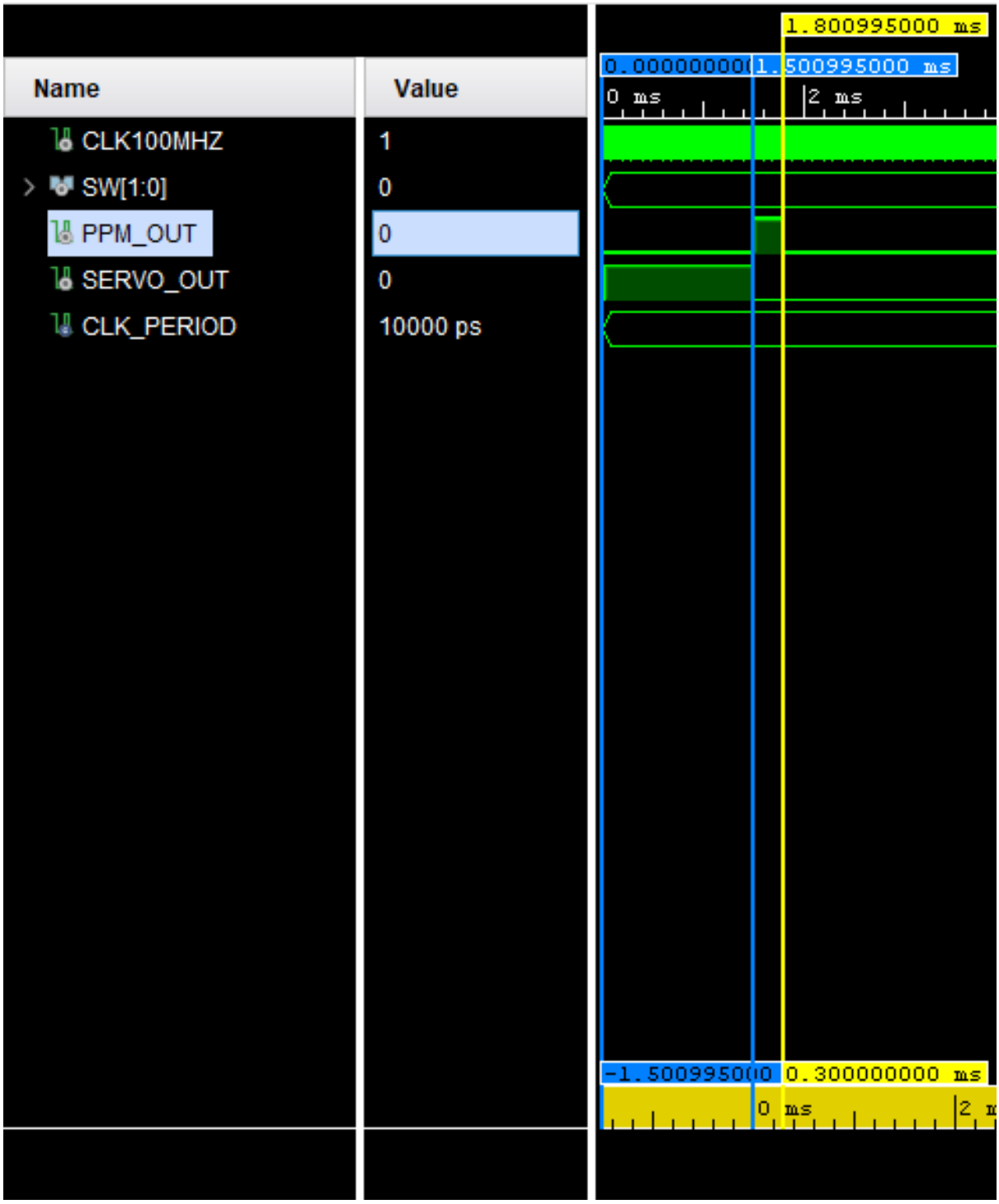


Figura 5: Inicio del pulso generado

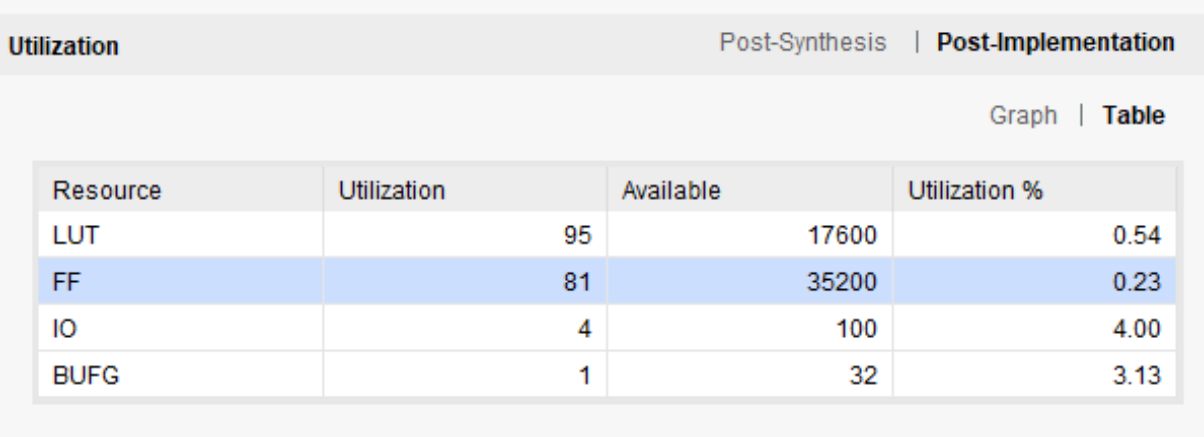
- Un periodo de repetición de 20 ms, correspondiente a una frecuencia de 50 Hz, cumpliendo la temporización estándar para control de servos.

2.3. Señal del receptor / servo (`SERVO_OUT`): la señal (`SERVO_OUT`) corresponde a la salida generada por el PPM Receiver, que convierte la información temporal del pulso PPM en una señal PWM proporcional. Durante la simulación, (`SERVO_OUT`) mantiene un nivel alto durante un tiempo equivalente al valor medido en (`PPM_OUT`). En este caso, el tiempo medido corresponde a aproximadamente 1 ms, lo cual representa la posición mínima del servo.

5 Tabla de uso de recursos de la FPGA

Después de la etapa de implementación en Vivado, se obtuvo el reporte de utilización de recursos de la FPGA Arty-Z7-10 (dispositivo XC7Z010).

1. Resultados generales de síntesis e implementación: los resultados muestran el uso extremadamente bajo de recursos lógicos, evidenciando la simplicidad y eficiencia del diseño.



Resource	Utilization	Available	Utilization %
LUT	95	17600	0.54
FF	81	35200	0.23
IO	4	100	4.00
BUFG	1	32	3.13

Figura 6: Utilización de recursos lógicos (post-implementación en Vivado)

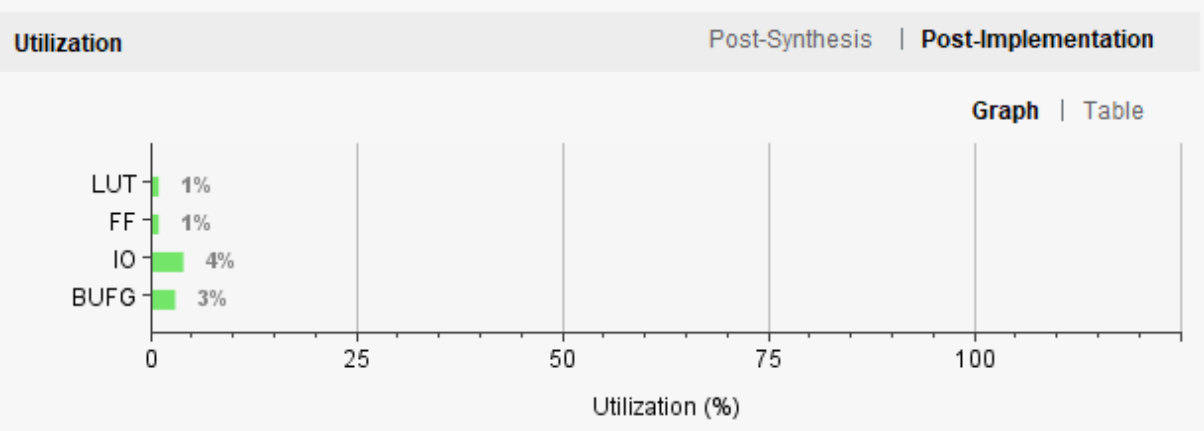


Figura 7: Utilización de recursos lógicos (post-implementación en Vivado)

2. Análisis de potencia: el reporte de consumo energético obtenido de Vivado muestra un consumo total de 0.094 W, donde el `99 %` corresponde a potencia estática del dispositivo y solo el `1 %` es potencia dinámica generada por el diseño.

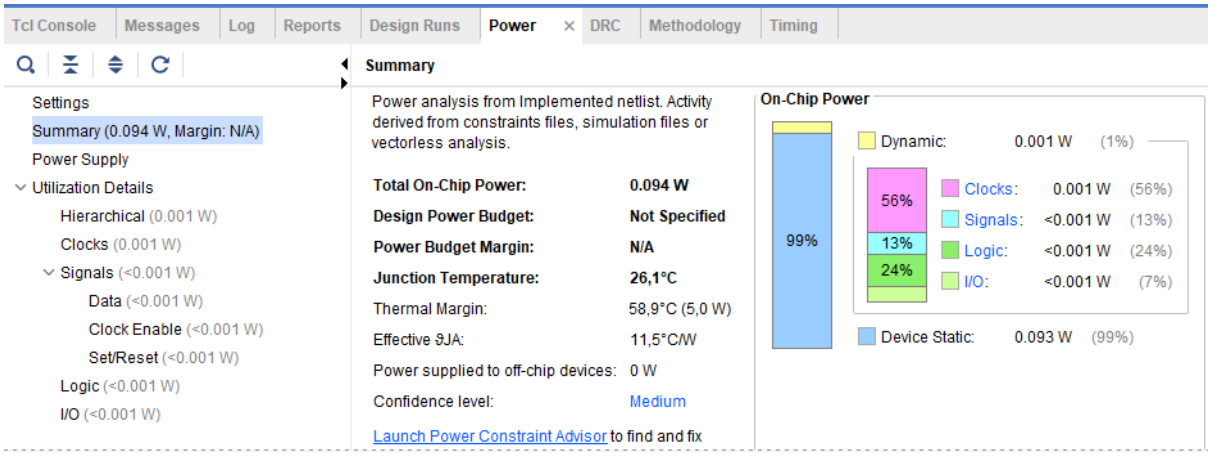


Figura 8: Utilización de recursos lógicos (post-implementación en Vivado)

3. Distribución física en el chip: La distribución del diseño dentro del tejido programable muestra una ocupación localizada en una pequeña región del plano lógico, principalmente en el cuadrante superior izquierdo del dispositivo (`SLICE_X0Y1`).

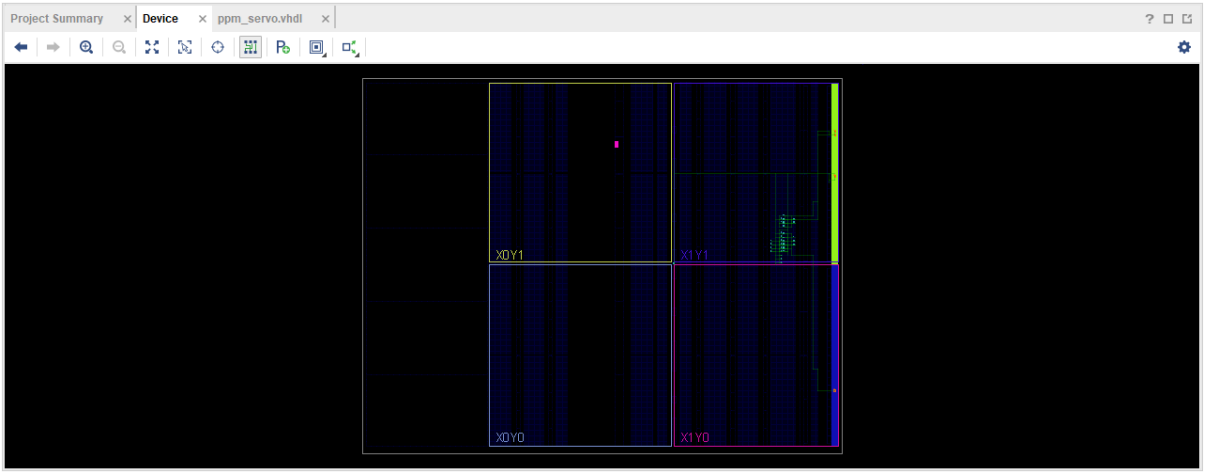


Figura 9: Distribución del diseño

6 Pruebas en hardware

Se implementó el diseño en hardware y se programó la FPGA con el bitstream generado. Se conectó el servo motor a 3.3V y a la pin Y19. Cuando se accionan los switches de la placa pone en marcha el servo motor.

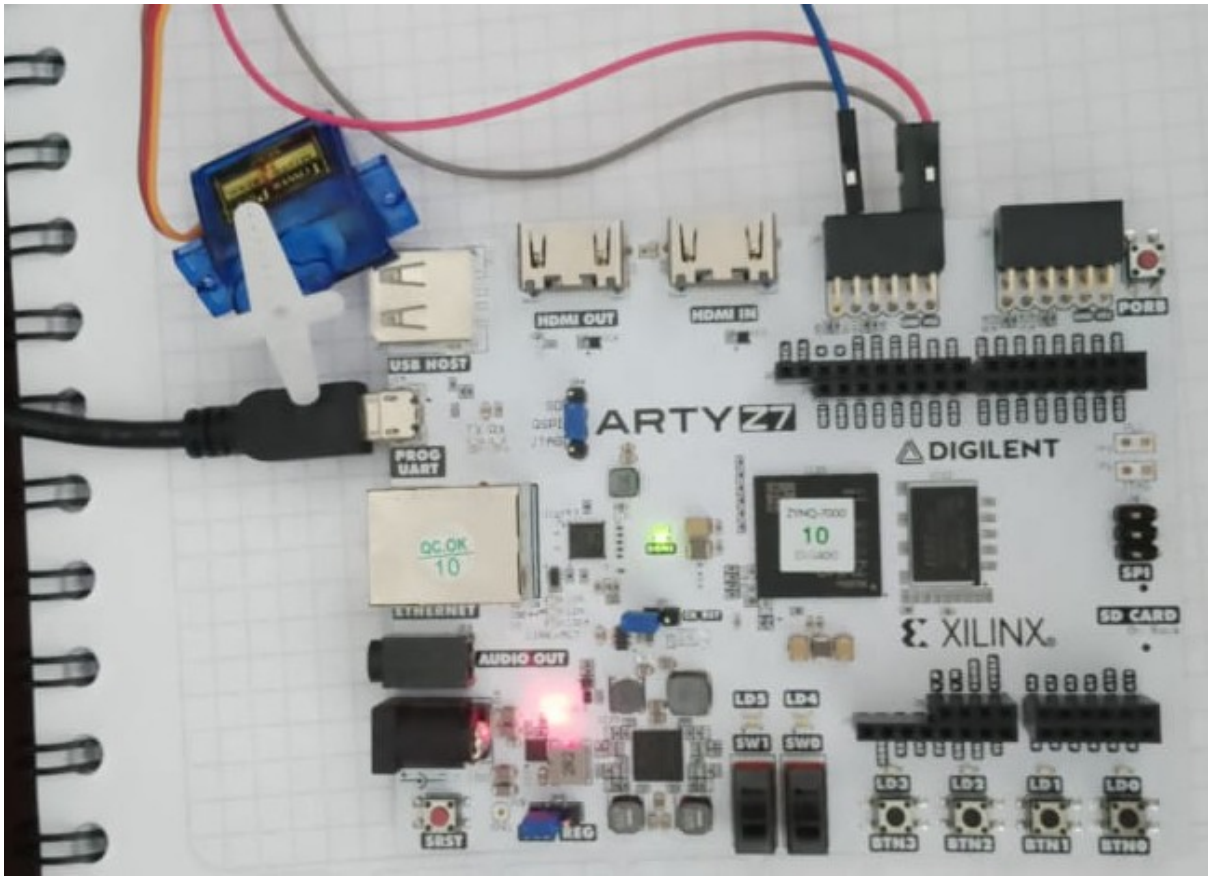


Figura 10: Implementación del proyecto

7 Conclusión

1. Se desarrolló una arquitectura modular compuesta por tres bloques principales.
2. El diseño se basó en principios de sistemas secuenciales síncronos, utilizando contadores y comparadores controlados por un reloj de 100 MHz, lo cual garantizó exactitud temporal y estabilidad en la modulación.
3. Los resultados simulados coincidieron plenamente con los valores teóricos definidos en el diseño, lo que valida la funcionalidad completa del sistema antes de su implementación física.
4. Tras la síntesis e implementación, se comprobó que el diseño ocupa menos del 2 % de los recursos lógicos del FPGA (LUTs y FFs), lo cual evidencia una alta eficiencia y escalabilidad.
5. El análisis del floorplanning mostró una distribución compacta dentro del tejido programable, dejando disponible la mayor parte del dispositivo para posibles expansiones.

6. El proyecto demuestra la capacidad de una FPGA para:
    - Implementar control temporal preciso a nivel de microsegundos.
    - Procesar señales de modulación y demodulación digital en tiempo real sin procesadores adicionales.
    - Simular y verificar hardware determinista de forma reproducible.

Este desarrollo confirma que las FPGA son plataformas óptimas para el procesamiento de señales modulares en tiempo real, combinando precisión, paralelismo y flexibilidad de configuración. Asimismo, sienta las bases para futuros proyectos orientados al control de múltiples ejes, automatización robótica o comunicaciones digitales basadas en PPM.