



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Diseño e Implementación de un Módulo PWM para Bus Paralelo

Daniel Felipe Acosta Castro

Paola Andrea Ladino Valbuena

Profesor

Juan bernardo Gómez Mendoza

Ingeniería Electrónica, Universidad Nacional de Colombia - Sede Manizales

Julio de 2025



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Resumen

Este proyecto presenta el diseño e implementación de un módulo PWM (Pulse Width Modulation) en lenguaje Verilog, concebido como un periférico configurable para sistemas digitales basados en la arquitectura RISC-V. El módulo permite la configuración dinámica de tres parámetros fundamentales: el divisor de frecuencia, el ciclo de trabajo (duty cycle) y el modo de alineación de la señal (izquierda, centrada o derecha), todo ello a través de un bus paralelo de 32 bits.

El sistema incorpora una máquina de estados finita (FSM) encargada de gestionar de manera eficiente las operaciones de lectura y escritura sobre el bus, asegurando integridad en la comunicación con el procesador. Adicionalmente, se integran registros sombra y mecanismos de protección que garantizan el correcto manejo de valores límite, restringiendo el duty cycle al rango válido entre 0% y 100%.

La señal PWM se genera mediante un contador interno que compara umbrales de activación y desactivación, definidos según el modo de alineación seleccionado. La arquitectura propuesta permite una modulación precisa, versátil y adaptable a distintos escenarios de control.

Para la validación funcional, se desarrolló un testbench exhaustivo que incluye casos nominales y situaciones límite, tales como duty cycles extremos, cambios dinámicos de configuración y variaciones del divisor de frecuencia. Las simulaciones realizadas en GTKWave corroboraron el comportamiento esperado, demostrando la estabilidad y confiabilidad del diseño.

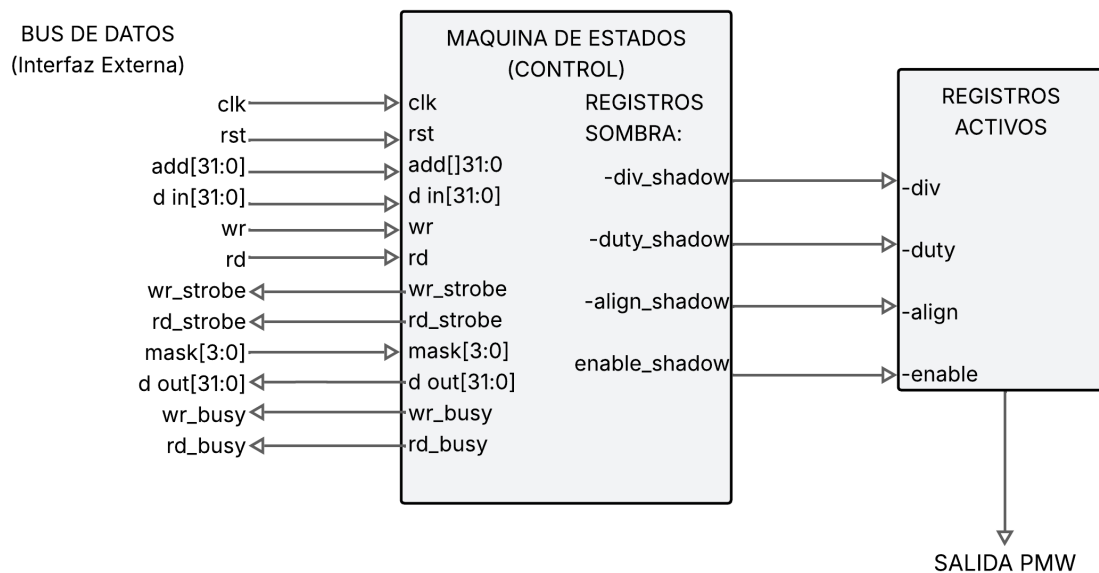
Este desarrollo constituye una solución eficiente y escalable para la generación de señales PWM en entornos de sistemas embebidos, con potencial de integración en plataformas SoC orientadas a control y procesamiento en tiempo real.



Descripción general

El presente proyecto consiste en el desarrollo de un módulo de modulación por ancho de pulso (PWM) programable, implementado en Verilog y diseñado para integrarse como periférico en un sistema RISC-V. El módulo permite configurar parámetros como el divisor de frecuencia, el ciclo de trabajo y el tipo de alineación de la señal, mediante un bus paralelo. Su funcionamiento fue validado a través de simulaciones que demostraron su correcto desempeño en distintos escenarios operativos.

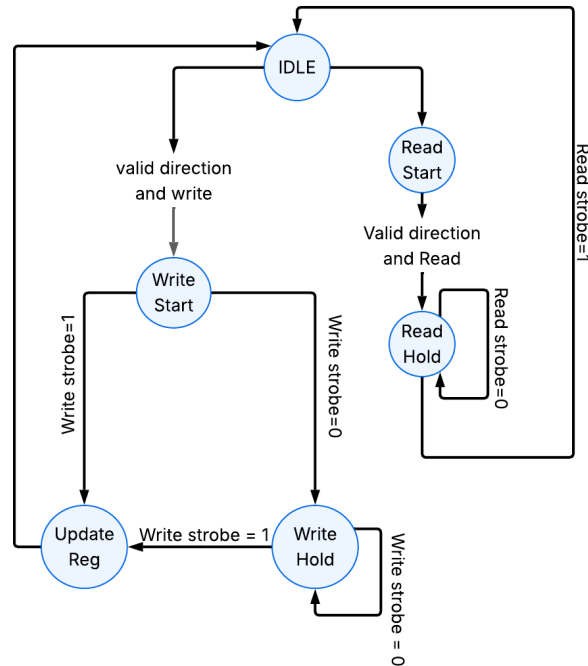
Componentes del módulo (Diagrama de bloques)



El diagrama ilustra la arquitectura del módulo PWM, compuesto por tres bloques principales: el bus de datos, la máquina de estados de control y los registros activos. La máquina de estados coordina las operaciones de lectura y escritura desde el bus, utilizando registros sombra para almacenar temporalmente los valores configurados. Estos valores se transfieren luego a los registros activos, que determinan el comportamiento de la señal PWM generada. Este esquema garantiza una comunicación sincronizada y una actualización segura de los parámetros del **sistema**.



Diagramas de estados



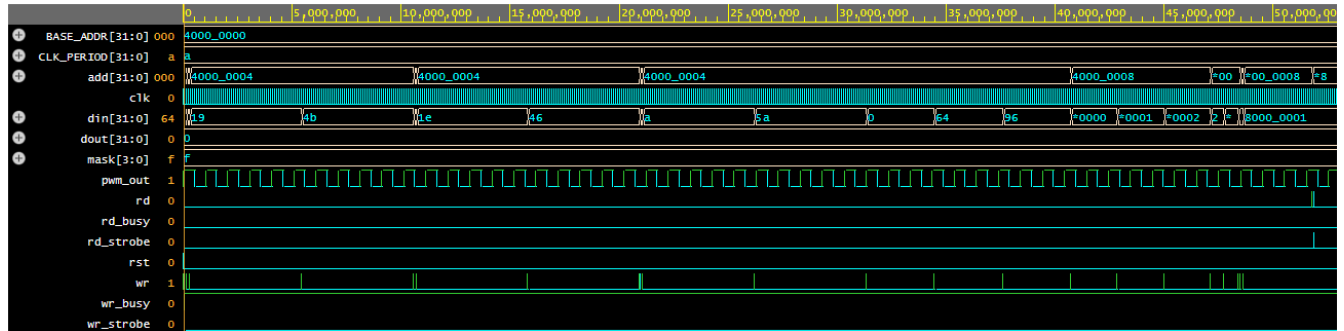
Explicación del diagrama de estados:

El diagrama representa el comportamiento secuencial de la máquina de estados encargada del control de acceso a los registros mediante operaciones de lectura y escritura. El sistema inicia en el estado **IDLE**, y dependiendo de la señal de control (lectura o escritura válida), transiciona hacia **Write Start** o **Read Start**.

En una escritura, se genera una señal `write_strobe = 1`, se actualizan los registros en **Update Reg**, y luego se pasa a **Write Hold** para asegurar la sincronización antes de regresar a **IDLE**. En lectura, la transición ocurre hacia **Read Start** y luego **Read Hold**, manteniéndose en este último mientras `read_strobe = 0`. Esto garantiza una gestión segura y ordenada del flujo de datos, evitando condiciones de carrera o errores de sincronización.



Diagramas de tiempo:



La imagen corresponde a la simulación temporal del sistema PWM con control de escritura y lectura mediante señales de strobe en un entorno EDA Playground.

Se observa cómo la señal **pwm_out** genera una onda cuadrada modulada en ancho de pulso, con cambios cuando se actualizan los valores en los registros a través de las señales de escritura (**wr**, **wr_strobe**, **wr_busy**). Por ejemplo, cuando **add** toma el valor 0x40000004, se cargan datos (**din**) como 0x64 y 0x96, que corresponden a la configuración del ciclo de trabajo y el divisor de frecuencia, respectivamente.

Además, se puede ver cómo el sistema gestiona correctamente el protocolo de escritura: la señal **wr** activa el proceso, **wr_busy** indica el estado de ocupación y **wr_strobe** marca el momento exacto de validación. No se evidencian accesos de lectura (**rd_strobe** y **rd_busy** permanecen bajos), lo que indica que el testbench está enfocado en probar configuraciones de salida PWM mediante escritura.

Figura 1:



Funcionamiento normal donde con el **rst** pone el **pwm** en estado idle y se generan unos valores

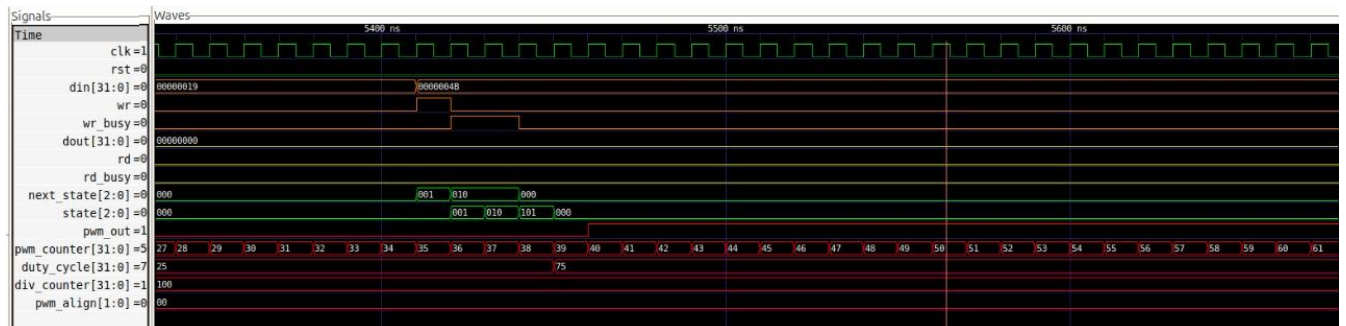


Figura 2:



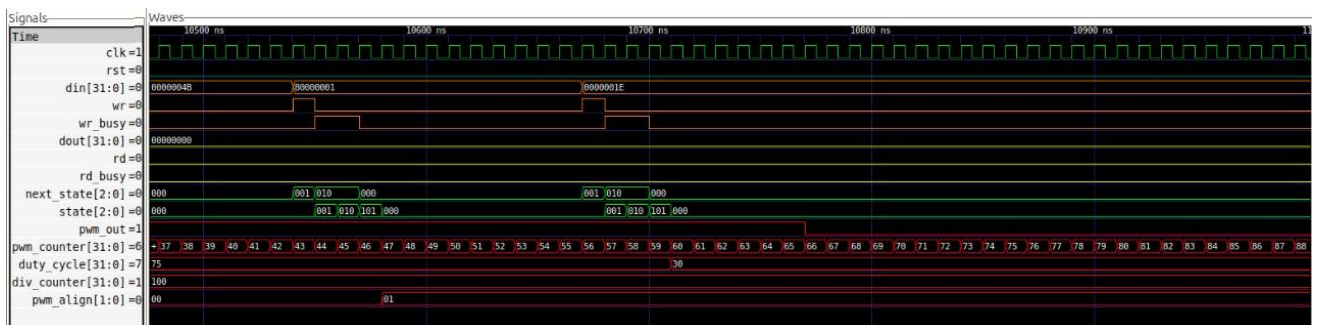
Comportamiento del PWM, después del rst, con configuración duty de 25% y alineación a la izquierda con periodo de 100 ciclos de reloj.

Figura 3:



Escritura de datos, cambio de duty al 75%, se observa el comportamiento de la máquina de estados

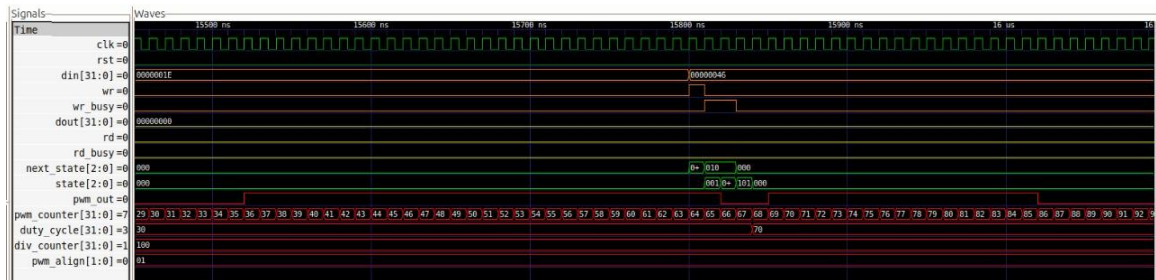
Figura 4:



Cambio de alineación del PWM a posición central y duty de 30%. Se observa que el pwm comienza a ejecutar el comportamiento desde el momento de la escritura.

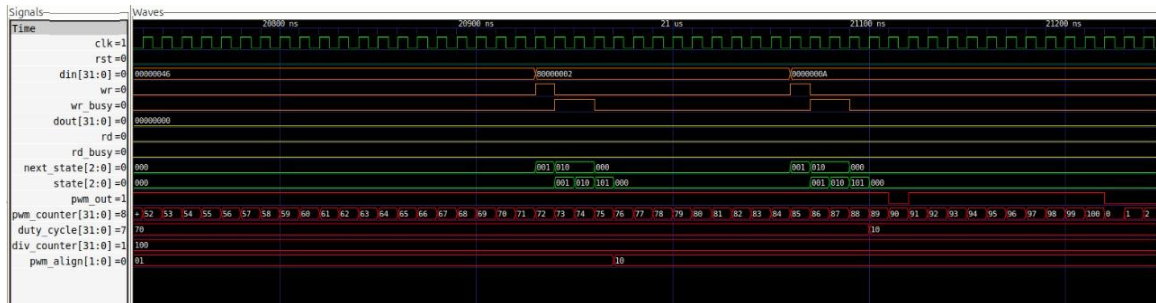


Figura 5:



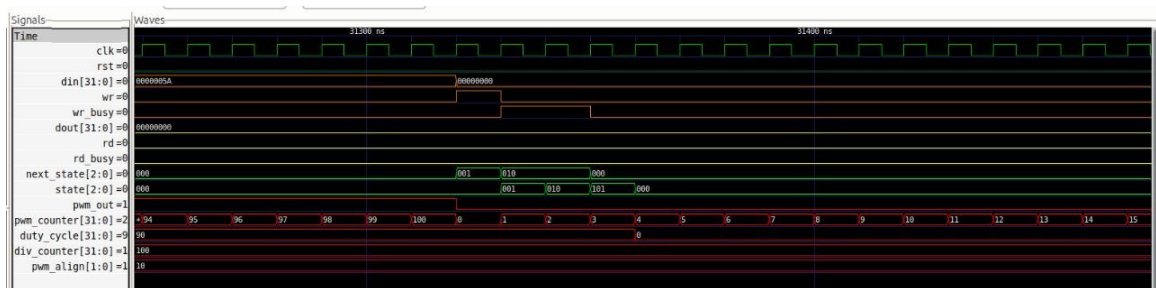
Cambiamos el duty a 70%

Figura 6:



Cambio de alineación de pwm a la derecha y porcentaje de duty en 10%.

Figura 7:

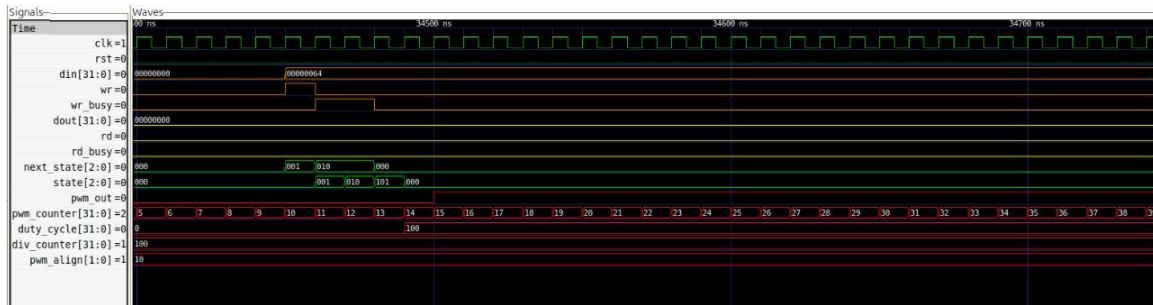


Implementación en un caso límite con duty al 0%.



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Figura 8:



Implementación de un caso limite con duty al 100%.

Figura 9:



Caso limite duty al 150%

Figura 10:



Implementación caso limite, cambio de frecuencia, periodo de 2 ciclos de reloj



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Figura 11:



Lectura de registros del pwm

Documentación de módulos

Módulo principal: pwm_peripheral

Entradas:

- clk, rst: reloj y reinicio
- add, din: dirección y datos de entrada del bus
- wr, rd: señales de escritura/lectura
- wr_strobe, rd_strobe: flancos de validación
- mask: máscara de escritura

Salidas:

- pwm_out: señal de salida PWM
- dout: datos de lectura
- wr_busy, rd_busy: indicadores de ocupación

Testbench: propósito y justificación

El propósito del testbench desarrollado es validar el correcto funcionamiento del módulo PWM parametrizable diseñado en Verilog, el cual permite configurar el ciclo de trabajo, el divisor de frecuencia y el modo de alineación mediante un sistema de escritura con bus paralelo. Esta validación es esencial para garantizar que el diseño cumple con los



requerimientos funcionales y responde adecuadamente a las configuraciones escritas desde una unidad de control o procesador embebido.

- El testbench fue construido para simular múltiples casos de prueba que ejercitan todas las funcionalidades principales del sistema:
- Inicialización del sistema: Se aplica un reinicio mediante la señal rst, asegurando que las salidas y registros internos comienzan en un estado conocido y estable.
- Escritura del ciclo de trabajo: Utilizando la dirección base 0x40000004, se envía un valor como 0x64 (100 en decimal) a través de din, lo cual representa el ciclo de trabajo del PWM.
- Escritura del divisor de frecuencia: A la misma dirección se envían valores como 0x96, que configuran la frecuencia de la señal de salida.
- Configuración del modo de alineación: En la dirección 0x40000008, se escriben valores que activan distintos modos de alineación (izquierda, centro, derecha), evidenciado por los datos 0x00000001, 0x00000002, y 0x80000001, entre otros.

La señal wr se activa para indicar la intención de escribir, mientras que wr_strobe y wr_busy gestionan el protocolo de control para asegurar que los datos son tomados correctamente por el sistema. A lo largo de la simulación, la señal pwm_out refleja los cambios esperados en la salida PWM en respuesta a las configuraciones realizadas, lo que demuestra que el módulo reacciona de forma coherente y precisa.

Casos de prueba testbench:

Para validar el correcto funcionamiento del módulo PWM, se implementaron dos grupos de pruebas:

- **Prueba 1:** Funcionamiento en los tres modos de alineación

Se verificó que la señal PWM se generara correctamente en los modos de alineación izquierda, centro y derecha. En cada caso, se observó el efecto de cambiar dinámicamente el ciclo de trabajo, confirmando que la salida se ajustaba de forma esperada.

- **Prueba 2:** Condiciones extremas y transiciones críticas

Esta prueba incluyó:

- Duty cycles en los límites (0%, 100%) y fuera de rango (>100%).
- Cambios de alineación en tiempo de ejecución, sin reiniciar el sistema.



UNIVERSIDAD
NACIONAL
DE COLOMBIA

- Uso de divisores mínimos y máximos para evaluar frecuencia y resolución.
- Escritura simultánea de varios registros de configuración.

Estas pruebas garantizaron que el módulo respondiera de forma robusta ante configuraciones típicas y condiciones límite, validando su confiabilidad para aplicaciones embebidas.

Resultados de simulación:

La simulación del módulo PWM evidenció un comportamiento correcto en todos los modos de operación y configuraciones evaluadas. En el waveform generado, se observó una señal PWM coherente con los valores de ciclo de trabajo y frecuencia definidos, cambiando dinámicamente según la escritura en los registros internos.

Durante la simulación se verificó:

- La salida pwm_out respondió correctamente ante cambios en el duty cycle y en el modo de alineación (izquierda, centro, derecha).
- La escritura en los registros fue validada mediante el control de las señales wr_strobe, wr_busy y wr, respetando los tiempos de espera y evitando conflictos.
- Se simularon condiciones límite como duty cycle del 0 %, 100 % y valores fuera de rango, con respuestas estables del sistema.
- Cambios de configuración (modo de alineación y divisor de frecuencia) se aplicaron sin necesidad de resetear el módulo, demostrando flexibilidad en tiempo real.
- El monitoreo de la señal pwm_out permitió confirmar el correcto ajuste del ancho de pulso conforme a los valores cargados en din.

En resumen, los resultados de simulación confirmaron la funcionalidad completa y robustez del diseño, validando su implementación para aplicaciones digitales programables.



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Conclusiones

- El desarrollo del módulo PWM parametrizable en Verilog demostró ser una solución eficaz y flexible para la generación de señales moduladas por ancho de pulso en sistemas digitales. A través de su integración con un bus paralelo, se logró una configuración dinámica del ciclo de trabajo, frecuencia y modo de alineación, permitiendo adaptabilidad en tiempo real sin necesidad de reiniciar el sistema.
- La validación mediante simulaciones en EDA Playground y Ubuntu evidenció un comportamiento correcto ante múltiples escenarios, incluyendo casos límite y escritura simultánea de registros. Esto confirmó la robustez y confiabilidad del diseño, así como su idoneidad para aplicaciones embebidas, sistemas de control, o módulos de salida digital.
- Además, el testbench permitió cubrir exhaustivamente los posibles modos operativos, garantizando que el diseño cumple con los objetivos funcionales y de rendimiento planteados desde su concepción. Este proyecto aporta una base sólida para futuras ampliaciones, como la implementación de interrupciones, control desde software embebido, o integración con sistemas SoC más complejos.