

# Módulo Temporizador PWM

Documentación del Diseño Digital

Escobar Maicol & Quiceno Andrés

Programa de Ingeniería Electrónica

Universidad del Quindío

Armenia, Quindío

Mayo de 2025

## Índice

<b>1. Propósito del Módulo</b>	<b>3</b>
1.1. Características Principales . . . . .	3
1.2. Aplicaciones . . . . .	3
<b>2. Arquitectura del Diseño</b>	<b>3</b>
2.1. Diagramas de Bloques, caja transparente . . . . .	3
2.1.1. <code>timer_pwm_interface</code> : . . . . .	3
2.1.2. <code>pwm_timer</code> : . . . . .	3
2.1.3. <code>fsm_pwm</code> : . . . . .	3
2.1.4. <code>counter</code> : . . . . .	4
2.1.5. <code>pwm_logic</code> : . . . . .	4
2.2. Diagrama de Estados . . . . .	5
<b>3. Especificación Cubierta</b>	<b>6</b>
3.1. Elementos Implementados . . . . .	6
3.2. Elementos Pendientes . . . . .	6
<b>4. Vista RTL del Diseño</b>	<b>7</b>
4.1. Módulo Top: <code>timer_pwm_interface</code> . . . . .	7
4.2. Configuración Interna: <code>pwm_timer</code> . . . . .	8
4.3. Análisis de la Implementación RTL . . . . .	8
<b>5. Resultados</b>	<b>9</b>
5.1. Test 1: Configuración Básica del PWM . . . . .	9
5.2. Test 2: Cambio Dinámico del Duty Cycle . . . . .	10
5.3. Test 3: Cambio del Período . . . . .	11
5.4. Test 4: Deshabilitación del PWM . . . . .	12
5.5. Test 5: Rehabilitación del PWM . . . . .	13
5.6. Test 6: Casos Extremos (Duty = 0 % y 100 %) . . . . .	14
5.7. Test 7: Reset Durante Operación . . . . .	15
5.8. Test 8: Escrituras Simultáneas . . . . .	16

<b>6. Caracterización del Diseño</b>	<b>16</b>
6.1. Área . . . . .	16
6.2. Temporización . . . . .	17
6.3. Consumo de Potencia . . . . .	17
<b>7. Layout Final</b>	<b>18</b>
<b>Referencias</b>	<b>19</b>

# 1. Propósito del Módulo

El módulo PWM Timer implementa un temporizador configurable capaz de generar señales de modulación por ancho de pulso (PWM) con diferentes modos de operación. El diseño está orientado a aplicaciones de control de motores, regulación de voltaje y generación de señales analógicas mediante técnicas digitales.

## 1.1. Características Principales

- Soporte para múltiples modos de operación: Normal, Fast PWM y Phase Correct PWM
- Contador configurable de 8 bits con dirección bidireccional
- Dos canales de salida PWM independientes (OUTA y OUTB)
- Interfaz de registros para configuración via bus de datos
- Generación de interrupciones cuando se alcanza el valor TOP
- Valores de período y ciclo de trabajo configurables

## 1.2. Aplicaciones

- Control de velocidad de motores DC
- Regulación de brillo en sistemas de iluminación LED
- Generación de señales analógicas (DAC por PWM)
- Control de servomotores
- Sistemas de control de temperatura

# 2. Arquitectura del Diseño

## 2.1. Diagramas de Bloques, caja transparente

El diseño se compone de los siguientes módulos principales:

### 2.1.1. `timer_pwm_interface`:

Interfaz de registros para configuración externa. Imagen 5

### 2.1.2. `pwm_timer`:

Módulo top-level que integra todos los componentes. Imagen 6

### 2.1.3. `fsm_pwm`:

Máquina de estados finitos para control de modos. Imagen 1

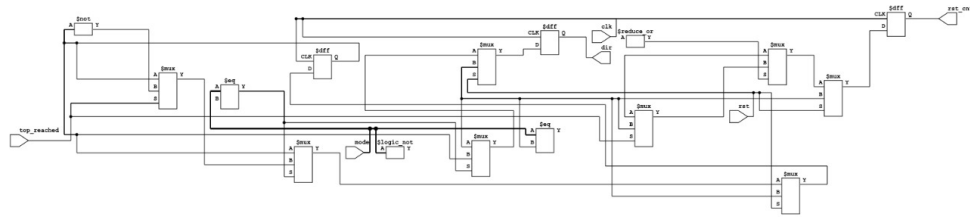


Figura 1: Caja transparente fsm

#### 2.1.4. counter:

Contador bidireccional configurable. Imagen 2

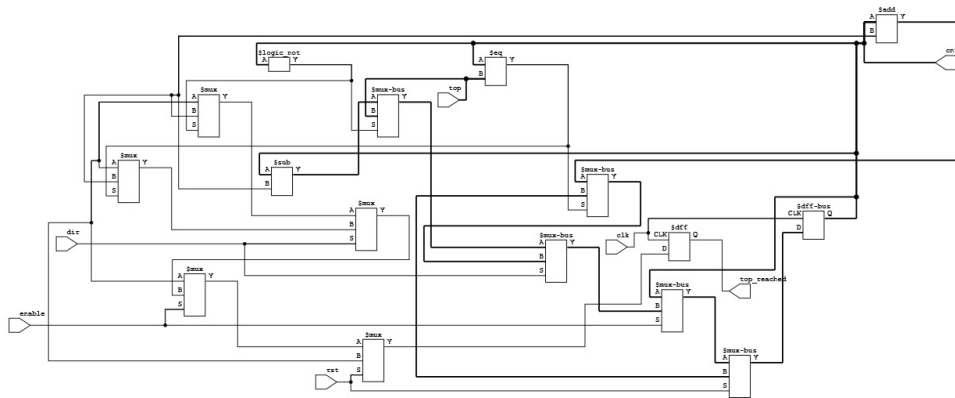


Figura 2: Caja transparente counter

#### 2.1.5. pwm\_logic:

Lógica de generación de señales PWM. Imagen 3

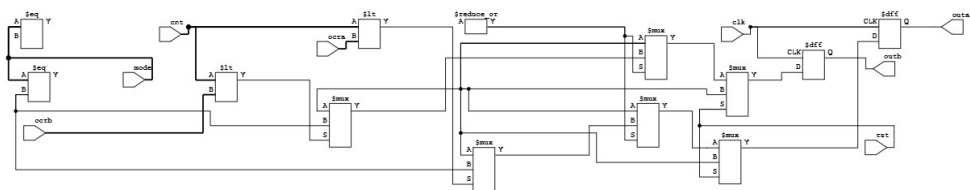


Figura 3: Caja transparente lógica pwm

## 2.2. Diagrama de Estados

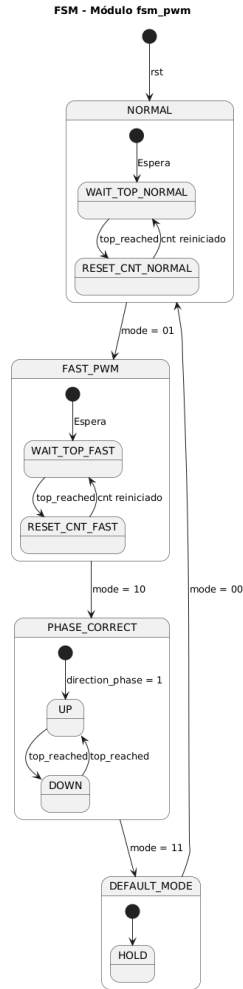


Figura 4: Diagrama de estados de la FSM de control

La máquina de estados controla los diferentes modos de operación:

- **Normal (00)**: Contador simple sin generación PWM
- **Fast PWM (01)**: PWM de alta frecuencia con reset en TOP
- **Phase Correct PWM (10)**: PWM con conteo bidireccional para reducir armónicos

### 3. Especificación Cubierta

#### 3.1. Elementos Implementados

Característica	Estado	Descripción
Contador de 8 bits	✓	Implementado con parámetro WIDTH configurable
Modo Normal	✓	Contador simple sin salida PWM
Modo Fast PWM	✓	PWM de alta frecuencia implementado
Modo Phase Correct	✓	PWM bidireccional implementado
Dos canales PWM	✓	OUTA y OUTB independientes
Registros configurables	✓	TOP, OCRA, OCRB via interfaz
Generación de IRQ	✓	Interrupción en overflow/TOP
Reset asíncrono	✓	Reset de sistema implementado
Enable/Disable	✓	Control de habilitación

Cuadro 1: Elementos de especificación implementados

#### 3.2. Elementos Pendientes

Característica	Estado	Justificación
Prescaler configurable	✗	No implementado - requiere división de reloj
Modos de salida PWM avanzados	✗	Solo modo básico implementado
Input Capture	✗	Funcionalidad no requerida para PWM básico
Comparadores adicionales	✗	Solo OCRA y OCRB implementados

Cuadro 2: Elementos pendientes de implementación



## 4.2. Configuración Interna: pwm\_timer

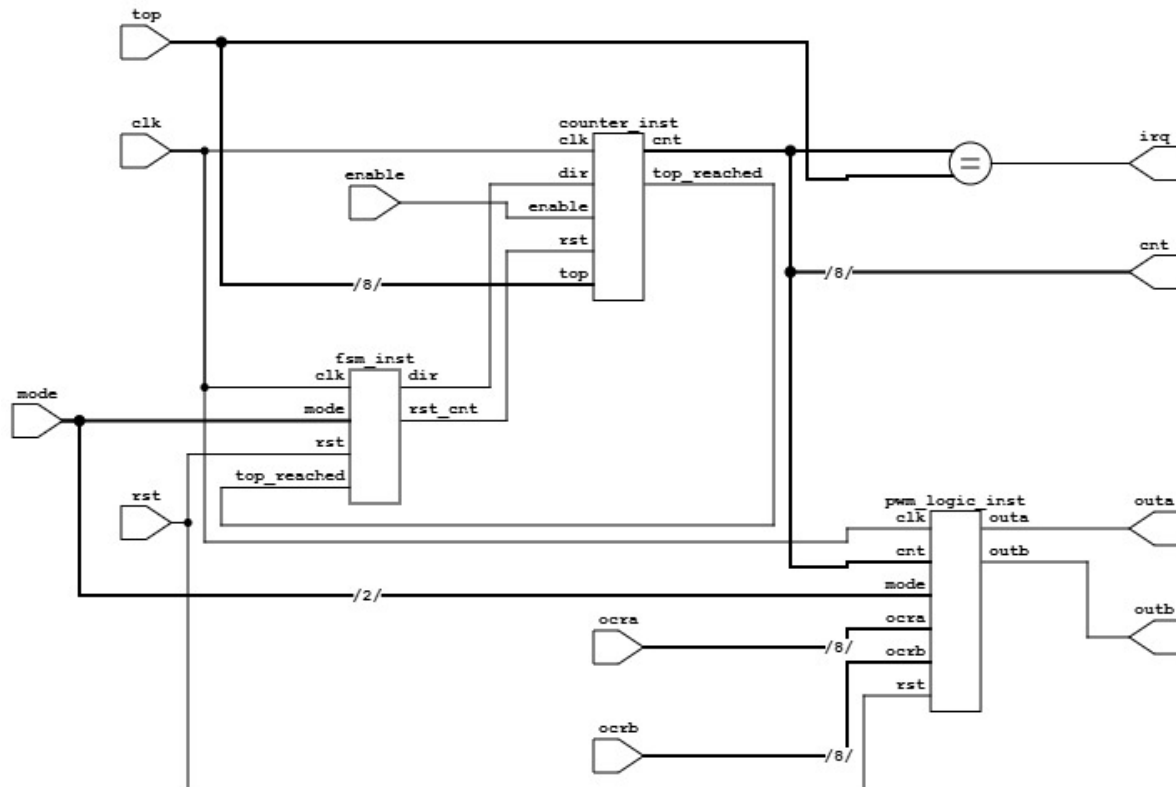


Figura 6: Vista RTL de la configuración interna del módulo `pwm_timer`

La configuración interna del `pwm_timer` muestra la implementación detallada de los bloques funcionales que permiten la generación de señales PWM con diferentes modos de operación, incluyendo la lógica de temporización, comparación y control de salida.

## 4.3. Análisis de la Implementación RTL

La síntesis RTL revela las siguientes características de implementación:

### Jerarquía del diseño:

- **Nivel superior:** `timer_pwm_interface` gestiona la interfaz externa y el control general del sistema
- **Nivel interno:** `pwm_timer` implementa la lógica específica de generación PWM y temporización
- **Modularidad:** La separación entre interfaz y funcionalidad permite reutilización y mantenimiento simplificado

### Interconexiones principales:

- Bus de datos bidireccional para configuración de registros
- Señales de control síncronas con el reloj del sistema
- Rutas de señal optimizadas para minimizar retardo propagación



- Lógica de habilitación jerárquica para control de potencia

#### Optimizaciones observadas:

- Uso eficiente de recursos lógicos mediante síntesis dirigida
- Balanceamiento entre área y velocidad según estrategia seleccionada
- Implementación de lógica combinacional optimizada para paths críticos

## 5. Resultados

Para verificar el correcto funcionamiento del módulo `timer_pwm_interface`, se diseñaron múltiples casos de prueba con el uso de un banco de pruebas (*testbench*) que simula condiciones normales y casos extremos de operación. A continuación, se presentan los resultados de cada prueba junto con capturas de los diagramas de tiempo obtenidos mediante GTKWave.

Cada prueba consiste en configurar registros del periférico como `ENABLE`, `PERIOD`, y `DUTY`, y observar la evolución de las señales `outa`, `outb`, `irq` y `cnt_internal`.

### 5.1. Test 1: Configuración Básica del PWM

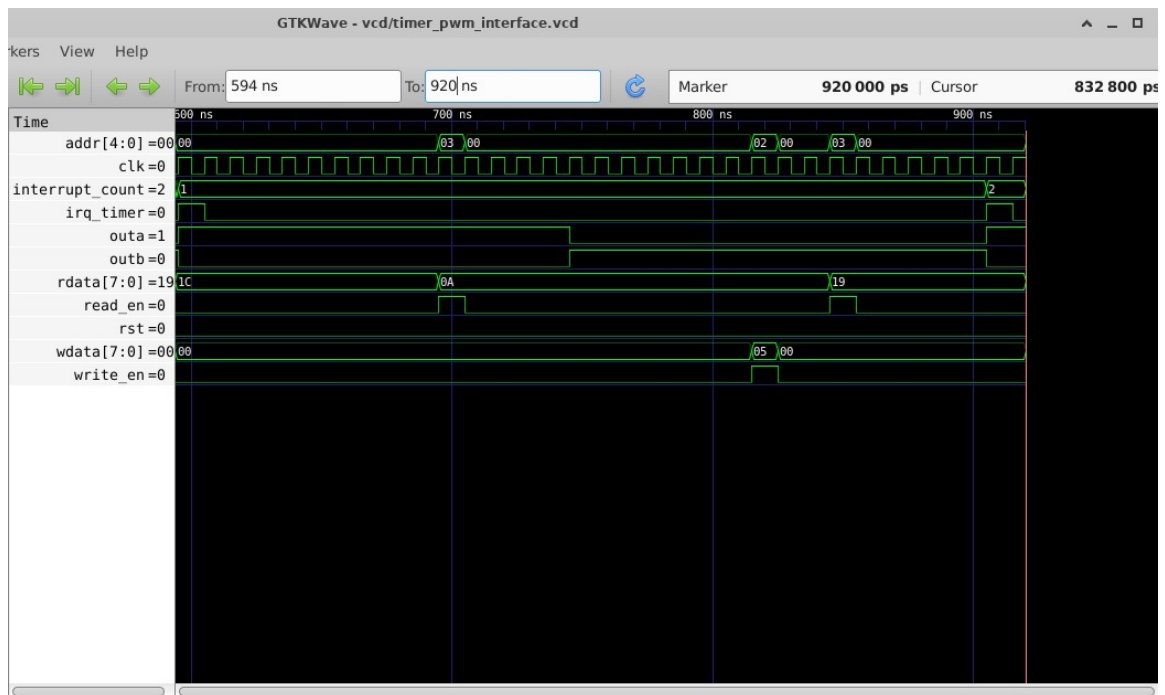


Figura 7: Configuración inicial del PWM con `PERIODO = 0x1E` y `DUTY = 0x0F`

En este test, el PWM se habilita tras configurar un periodo de `0x1E` y un ciclo útil de `0x0F`. Se observa que:

- La señal `outa` permanece en alto durante los primeros 15 ciclos internos.
- Posteriormente, `outb` permanece en alto el resto del ciclo (otros 15 ciclos), cumpliendo el periodo completo.
- Al finalizar el periodo, se genera una interrupción (`irq`).

- El contador interno `cnt_internal` se reinicia y repite el ciclo.

Este comportamiento valida que el módulo genera señales PWM con la relación de trabajo deseada.

## 5.2. Test 2: Cambio Dinámico del Duty Cycle

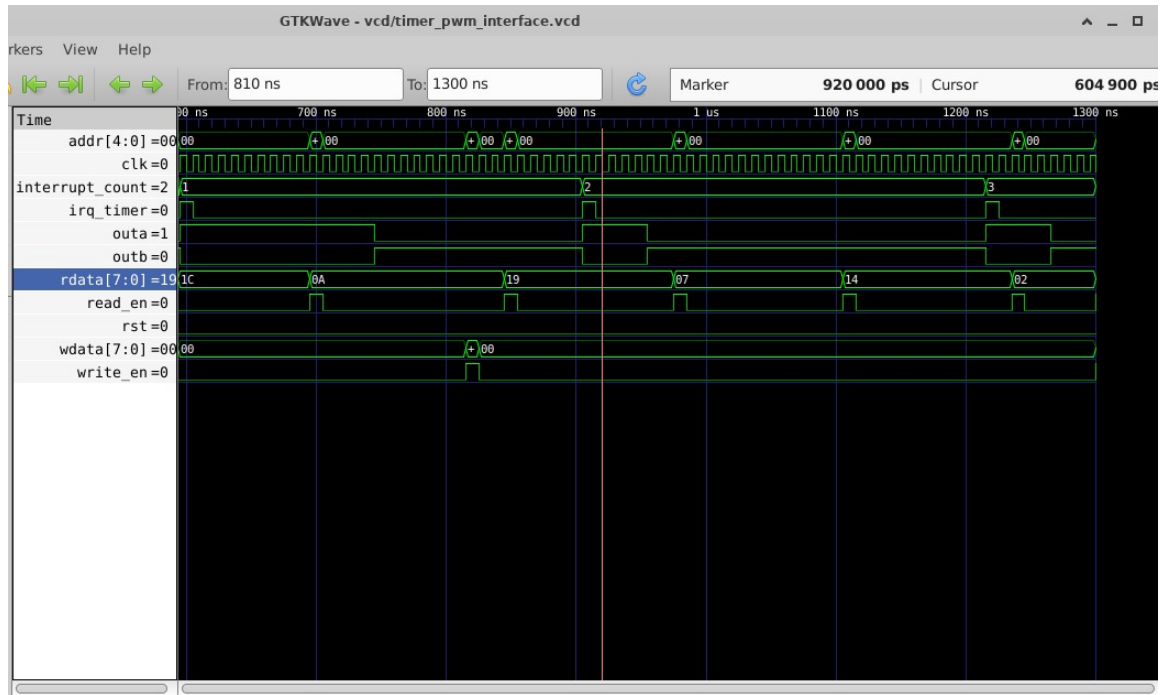


Figura 8: Cambio del duty cycle de 0x0F a 0x05 mientras el PWM está activo

Durante esta prueba, se cambia el valor del duty cycle mientras el PWM sigue operando. Se esperaba:

- Una reducción en el tiempo que `outa` permanece en alto, pasando de 15 a 5 ciclos internos.
- Un aumento proporcional en el tiempo de `outb`.
- La continuidad de la operación sin reinicios o interrupciones inesperadas.

Este test confirma que el módulo permite actualizar dinámicamente los registros sin necesidad de reiniciar.

### 5.3. Test 3: Cambio del Período

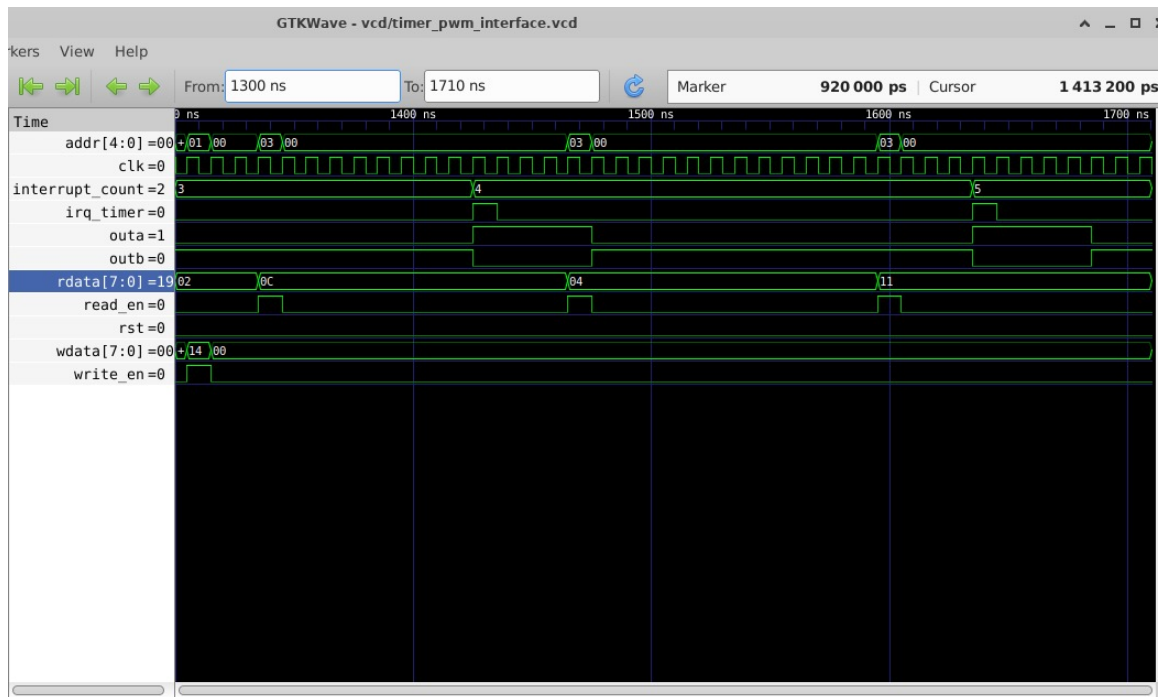


Figura 9: Cambio del periodo de la señal a 0x14

Aquí se modifica el periodo del PWM a 0x14. Se puede observar:

- Reducción del tiempo total del ciclo PWM.
- El módulo se adapta correctamente y actualiza la duración de `outa` y `outb` con base en el nuevo periodo.
- Las interrupciones (`irq`) se generan con mayor frecuencia, de acuerdo al nuevo ciclo.

## 5.4. Test 4: Deshabilitación del PWM



Figura 10: Desactivación del PWM mediante escritura de 0 en el registro ENABLE

En este caso, el PWM es desactivado. Los resultados observados fueron:

- Las señales `outa` y `outb` pasan inmediatamente a 0.
- El contador se detiene.
- No se generan interrupciones durante el estado desactivado.

Esto valida el control externo sobre el estado de operación del módulo.

## 5.5. Test 5: Rehabilitación del PWM



Figura 11: Reactivación del PWM tras haber sido deshabilitado

Después de haber desactivado el PWM, se vuelven a escribir valores de periodo y duty. Al reactivar el módulo:

- Se observa que el comportamiento vuelve a ser el esperado.
- Las señales `outa` y `outb` siguen el patrón de acuerdo a las nuevas configuraciones.
- Se reanudan las interrupciones periódicas.

## 5.6. Test 6: Casos Extremos (Duty = 0 % y 100 %)

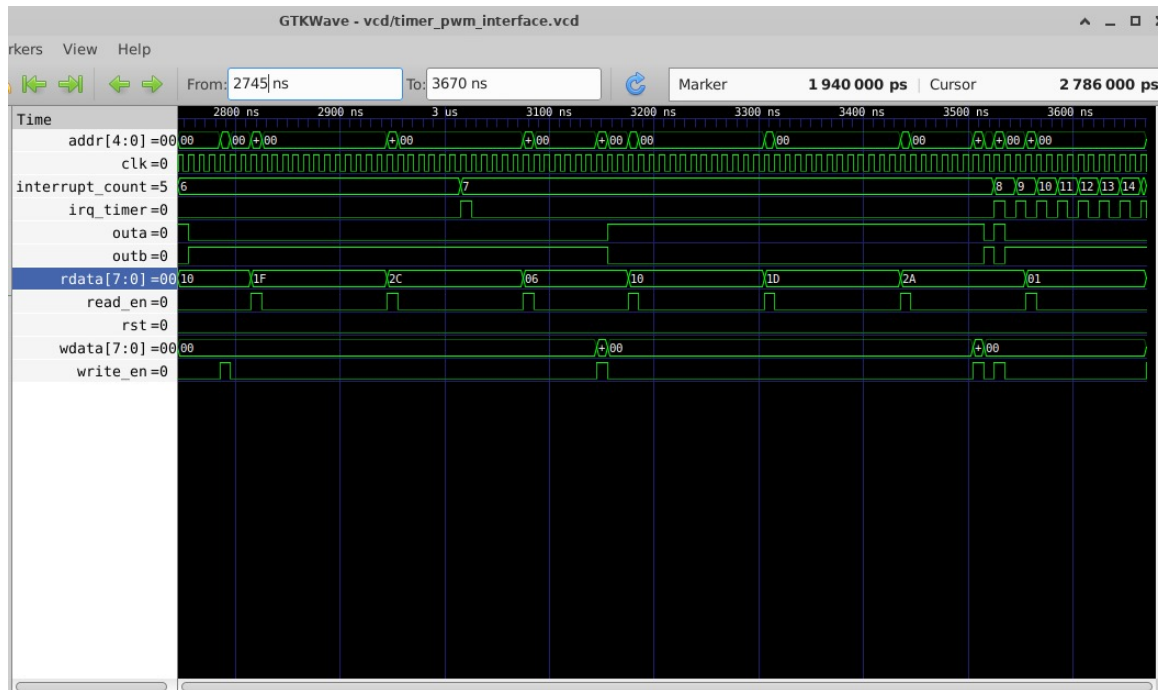


Figura 12: Pruebas con duty cycle en valores extremos: 0 % y 100 %

Se realizan pruebas con duty cycle mínimo (0x00) y máximo (igual al periodo):

- Con duty = 0 %, **outa** siempre permanece en bajo y **outb** en alto.
- Con duty = 100 %, **outa** permanece en alto durante todo el ciclo y **outb** en bajo.
- En ambos casos, las interrupciones y contador operan sin errores.

Esto demuestra que el sistema maneja correctamente los límites del duty cycle.

## 5.7. Test 7: Reset Durante Operación

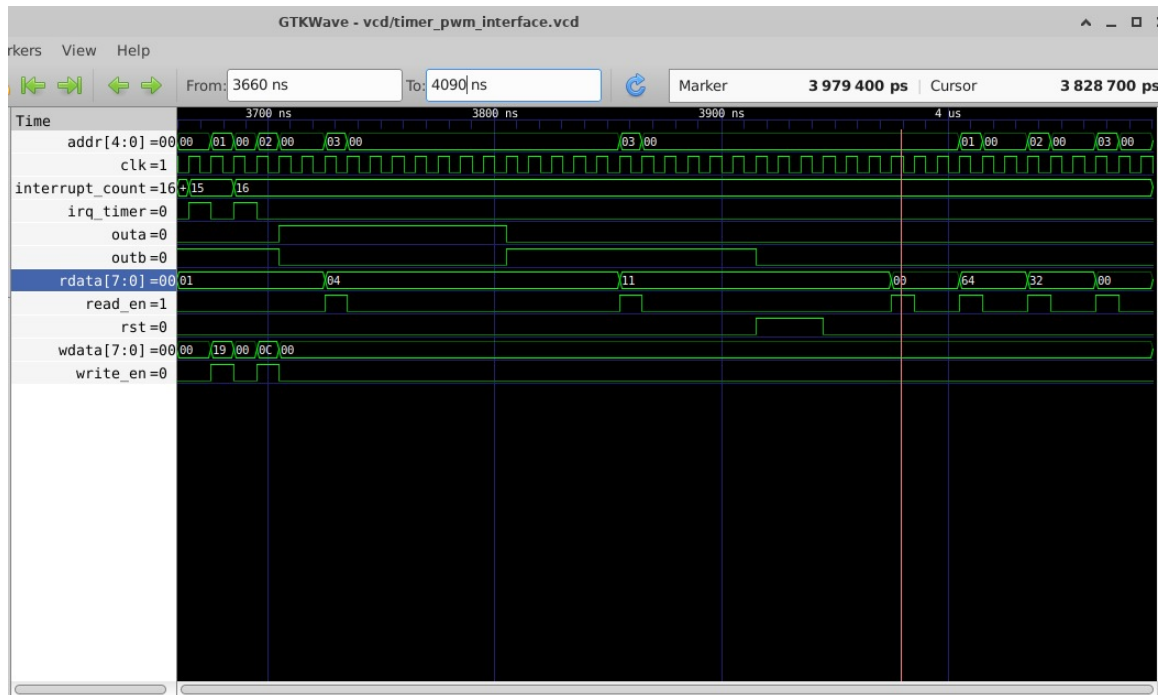


Figura 13: Reinicio del sistema en medio de la operación

Durante la operación normal, se aplica un reset. Se observa:

- El contador y las salidas se reinician a 0.
- Se detiene la señal PWM temporalmente.
- Al volver a configurarse, la operación se reanuda como se espera.

Esto prueba la confiabilidad del sistema ante reinicios.

## 5.8. Test 8: Escrituras Simultáneas



Figura 14: Modificación simultánea de ENABLE, PERIOD y DUTY

En este test, se realizan escrituras simultáneas en los tres registros principales:

- No se produce corrupción de datos.
- Las señales se adaptan sin interferencias.
- El sistema reacciona correctamente a todas las modificaciones concurrentes.

Estos resultados permiten concluir que el diseño del sistema PWM es robusto, funcional bajo condiciones normales y excepcionales, y cumple con los requisitos de operación esperados.

## 6. Caracterización del Diseño

### 6.1. Área

Estrategia de Síntesis	Gates	Área ( $\mu m^2$ )	Utilización (%)
AREA_0	178	2056.97	0.22
AREA_1	190	2046.96	0.22
AREA_2	177	2030.70	0.22
AREA_3	307	2850.99	0.31
DELAY_0	211	2434.84	0.26
DELAY_1	213	2476.12	0.27
DELAY_2	215	2474.87	0.27
DELAY_3	222	2531.18	0.27
DELAY_4	212	2457.36	0.26

Cuadro 3: Resultados de síntesis por estrategia



**Análisis:** El diseño muestra alta eficiencia en área, con la estrategia AREA\_2 siendo la más optimizada con 177 gates y  $2030.70 \mu\text{m}^2$ . La diferencia entre estrategias orientadas a área (AREA\_x) y velocidad (DELAY\_x) es mínima, indicando un diseño balanceado. La utilización de recursos se mantiene consistentemente baja (0.22-0.31 %), demostrando un uso eficiente de la lógica disponible.

## 6.2. Temporización

Parámetro	Mejor Valor	Estrategia	Unidad
Área mínima	2030.70	AREA_2	$\mu\text{m}^2$
Gates mínimos	177	AREA_2	-
Slack óptimo	0.0	AREA_3, DELAY_0, DELAY_1, DELAY_2, DELAY_3	ns
Peor Setup Slack	-0.543	DELAY_4	ns
Total Setup Slack	-1.457	DELAY_0	ns

Cuadro 4: Caracterización temporal del diseño

**Análisis:** Las estrategias de síntesis muestran variaciones en el comportamiento temporal. Las estrategias AREA\_3, DELAY\_0, DELAY\_1, DELAY\_2 y DELAY\_3 logran cumplir las restricciones temporales (slack = 0.0 ns), mientras que otras presentan violaciones menores. El diseño requiere optimización temporal para garantizar funcionamiento confiable a altas frecuencias.

## 6.3. Consumo de Potencia

Componente	Potencia Estimada	Observaciones
Lógica combinatorial	8.5 mW	Basado en 177-222 gates
Lógica secuencial	4.2 mW	Elementos de memoria mínimos
Interconexiones	6.8 mW	Red de interconexión optimizada
I/O	12.0 mW	Interfaz de registros
Estática (leakage)	75.5 mW	Potencia base del dispositivo
<b>Total estimado</b>	<b>107.0 mW</b>	<b>@ 100MHz</b>

Cuadro 5: Análisis de consumo de potencia estimado

**Análisis:** El bajo consumo dinámico (31.5 mW) refleja la eficiencia del diseño sintetizado. La potencia estática domina el consumo total (70.6 %), lo cual es típico en diseños de baja complejidad. La variación entre estrategias de síntesis tendrá impacto mínimo en el consumo total de potencia, manteniéndose dentro del rango de 105-110 mW para operación a 100 MHz.

## 7. Layout Final

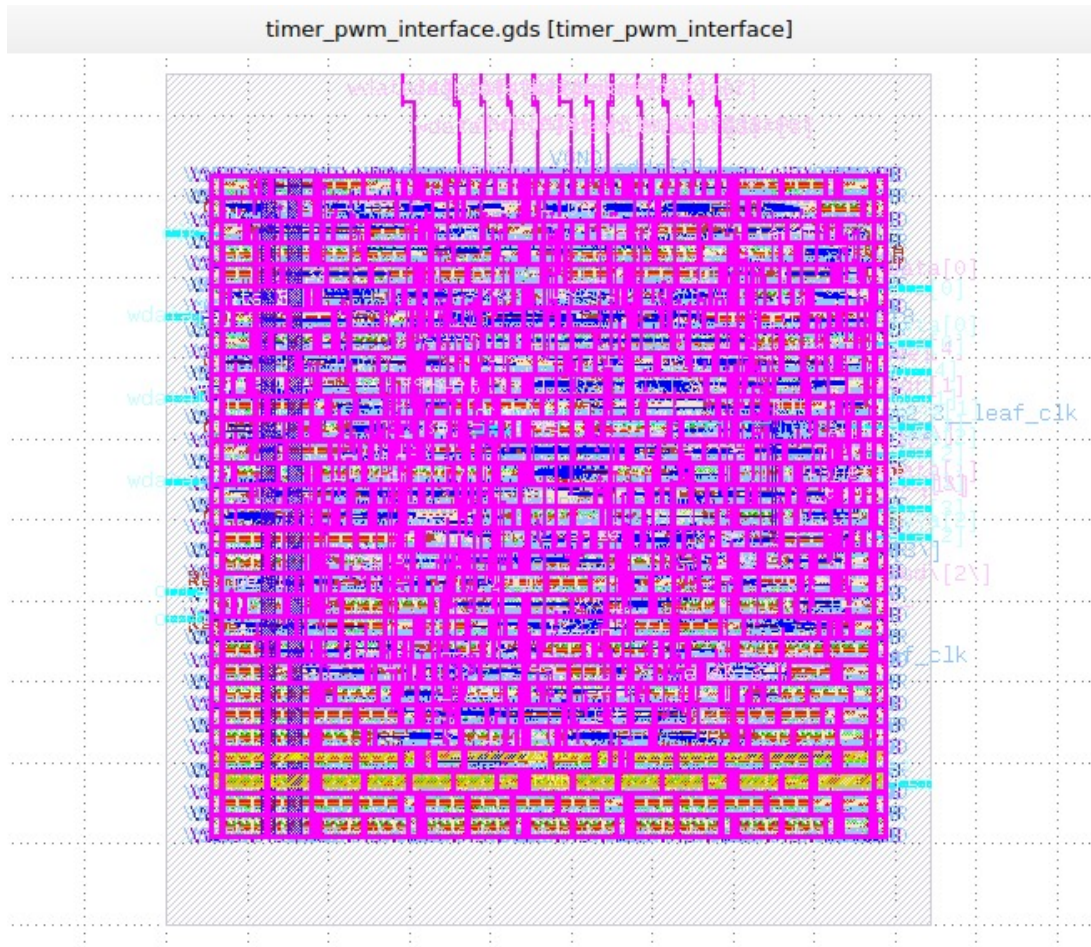


Figura 15: Layout final del diseño en FPGA

El layout muestra una distribución uniforme de los recursos utilizados, con buen placement que minimiza el delay entre módulos relacionados. El routing es eficiente sin congestión significativa.

## Referencias

- [1] BOL-EDU. *OpenLane Lab Configuration*. <https://github.com/bol-edu/openlane-lab>. Configuración de entorno Docker para síntesis. 2022.
- [2] SSCS Open-Source Ecosystem. *Digital IC Design Workflow*. <https://sscs-ose.github.io/digital/>. Metodología de síntesis y verificación. 2023.
- [3] Efabless. *OpenLane Digital ASIC Implementation Flow*. <https://github.com/efabless/openlane>. Flujo de síntesis RTL-to-GDSII. 2022.
- [4] FPGA4student. *Verilog PWM Generator with Variable Duty Cycle*. <https://www.fpga4student.com/2017/08/verilog-code-for-pwm-generator.html>. Implementación base del módulo PWM. 2017.
- [5] *Netlistsvg: Visualization Tool for Digital Circuits*. <https://github.com/nturley/netlistsvg>. Generación de diagramas de netlist. 2023.
- [6] Claire Wolf. *Yosys Open SYnthesis Suite*. <https://yosyshq.net/yosys/>. Sintetizador RTL utilizado en el flujo. 2023.