# Gowin PWM IP

# User Guide

**IPUG1207-1.0E, 2025-06-13**

**Disclaimer**

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

## Revision History

| Date | Version | Description |
|---|---|---|
| 06/13/2025 | 1.0E | Initial version published. |

# Contents

# List of Figures

# List of Tables

# 1 About This Guide

## 1.1 Purpose

Gowin PWM IP user guide aims to help users quickly learn the functions, features, and usage of Gowin PWM IP. It mainly helps users quickly understand the features and usage of Gowin PWM IP. The software screenshots and the supported products listed in this manual are based on Gowin Software 1.9.11.02. As the software is subject to change without notice, some information may not remain relevant and may need to be adjusted according to the software that is in use.

## 1.2 Related Documents

The latest user guides are available on the GOWINSEMI website. You can find the related documents at *www.gowinsemi.com*:

- *DS100, GW1N series of FPGA Products Data Sheet*
- *DS117, GW1NR series of FPGA Products Data Sheet*
- *DS821, GW1NS series of FPGA Products Data Sheet*
- *DS861, GW1NSR series of FPGA Products Data Sheet*
- *DS961, GW2ANR series of FPGA Products Data Sheet*
- *DS102, GW2A series of FPGA Products Data Sheet*
- *DS226, GW2AR series of FPGA Products Data Sheet*
- *DS971, GW2AN-18X & 9X Data Sheet*
- *DS976, GW2AN-55 Data Sheet*
- *DS981, GW5AT series of FPGA Products Data Sheet*
- *DS1103, GW5A series of FPGA Products Data Sheet*
- *DS1239, GW5AST series of FPGA Products Data Sheet*
- *DS1105, GW5AS series of FPGA Products Data Sheet*

- *DS1108, GW5AR series of FPGA Products Data Sheet*
- *DS1118, GW5ART series of FPGA Products Data Sheet*
- *SUG100, Gowin Software User Guide*

## 1.3 Terminology and Abbreviations

The terminology and abbreviations used in this manual are as shown in Table 1-1.

**Table 1-1 Terminology and Abbreviations**

| Terminology and Abbreviations | Meaning |
|---|---|
| IP | Intellectual Property |
| LUT | Look-up Table |
| PWM | Pulse width modulation |

## 1.4 Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly using the information provided below.

Website: *www.gowinsemi.com*

E-mail: *support@gowinsemi.com*

# 2 Overview

Gowin PWM IP can implement the basic function of pulse-width modulation.

**Table 2-1 Gowin PWM IP Overview**

| Gowin PWM IP | |
|---|---|
| Logic Resource | See Table 3-1. |
| Delivered Doc. | |
| Design Files | Verilog (encryption) |
| Reference Design | Verilog |
| TestBench | Verilog |
| Test and Design Flow | |
| Synthesis Software | GowinSynthesis |
| Application Software | Gowin Software (V1.9.11.02 and above) |

**Note!**

For the devices supported, you can click *here* to get the information.

# 3 Features and Performance

## 3.1 Features

- Low resource utilization
- Precision up to nanosecond level
- Supports two types of pulse width adjustment

## 3.2 Operating Frequency

The operating frequency of Gowin PWM IP depends on the precision required by the IP, as detailed in *4.1 Clock*.

## 3.3 Resource Utilization

Table 3-1 shows the resource utilization under a certain configuration. The resource utilization varies with different configurations, and an assessment needs to be made based on the actual situation.

**Table 3-1 Resource Utilization**

| LUTs | REGs | Device Series | Speed Level |
|---|---|---|---|
| 84 | 57 | GW1N-9 | C6/I5 |

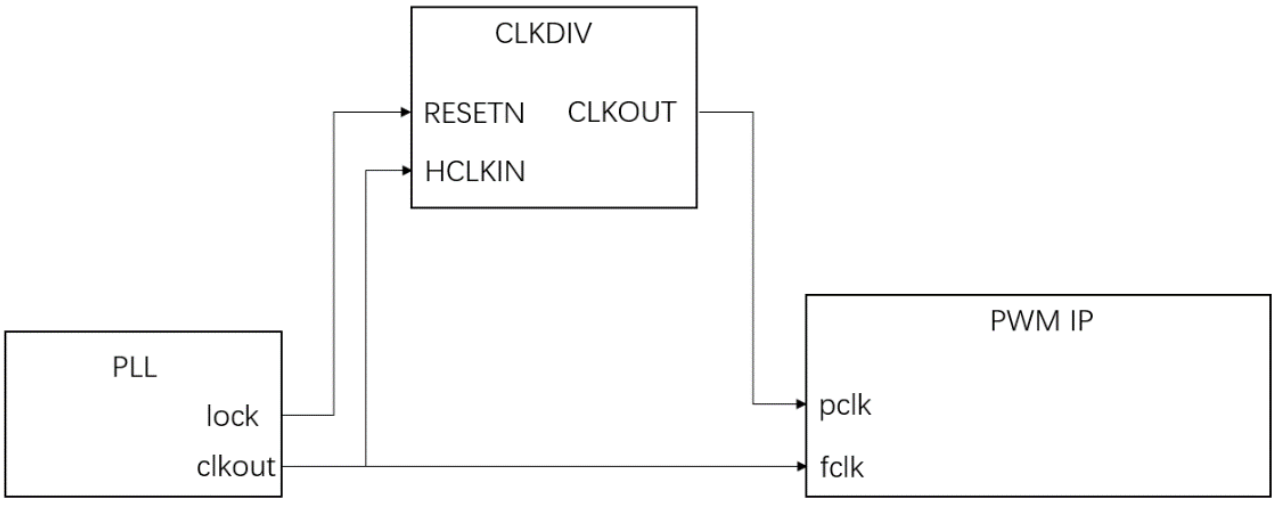
# 4 Functional Description

## 4.1 Clock

When calling Gowin PWM IP, the user must provide the correct clock topology as shown in the figure below to ensure proper operation of the IP. The division factor for CLKDIV must be configured as "4".

If multiple Gowin PWM IPs are instantiated with the same precision, the pclk, fclk, and fclk_lock signals can be shared among them. The frequency of pclk is equal to the PLL output clock frequency divided by 4.

**Figure 4-1 Clock Block Diagram**




## 4.2 Precision

The PWM precision refers to the minimum step size for increasing or decreasing the pulse width. It is calculated using the following formula:

Precision = 1/(fclk × 2), and the fclk is the frequency of the PLL output clock.

# 4.3 Period

The PWM period refers to the time it takes for a PWM signal to go from high level to low level and back to high level in one complete cycle. It is calculated using the following formula:

Period = initial_cycle/(fclk*2), and fclk represents the frequency of the PLL clkout clock, and initial_cycle is the input signal of the IP core.

# 4.4 Initialization

The user can initialize the PWM period and initial pulse width through the input signals provided by IP. When initial_update transitions from 0 to 1, the values of initial_cycle and initial_duty_cycle are loaded into the IP. initial_cycle is used to initialize the PWM period. The period is given by initial_cycle/(fclk*2), and initial_cycle must be an integer multiple of 8. initial_duty_cycle is used to initialize the initial PWM pulse width. The pulse width is given by initial_duty_cycle/(fclk*2).

# 4.5 Pulse Width Adjustment

Gowin PWM IP provides two methods for modifying the pulse width, including continuous adjustment and direct assignment.

## 4.5.1 Continuous Adjustment

Users can increase or decrease the pulse width by one precision unit using the IP input signals up and down. When the up signal transitions from 0 to 1, the pulse width increases by one precision step. When the down signal transitions from 0 to 1, the pulse width decreases by one precision step.

## 4.5.2 Direct Assignment

Users can set the pulse width to an arbitrary value using the IP input signals duty_cycle and duty_cycle_update. When duty_cycle_update transitions from 0 to 1, the PWM output pulse width is updated to the input value of duty_cycle, with a width of duty_cycle/(fclk*2).

# 5 Port List

The IO port of Gowin PWM IP is shown in Table 5-1.

**Table 5-1 Gowin PWM IP IO Port List**

| Signal | I/O | Data Width | Description |
|---|---|---|---|
| Clock Input Signals | | | |
| fclk | input | 1 | High-speed clock input, typically generated by the PLL. The clock frequency determines the PWM precision. |
| pclk | input | 1 | Low-speed clock input, typically generated by CLKDIV. The frequency is 1/4 of fclk. |
| Control Signals | | | |
| pwm_en | input | 1 | PWM enable signal:<br>1: PWM outputs a pulse signal<br>0: PWM outputs a low level |
| up | input | 1 | Increases PWM pulse width; When this signal transitions from 0 to 1, the pulse width increases by one precision step. |
| down | input | 1 | Decreases PWM pulse width; When this signal transitions from 0 to 1, the pulse width decreases by one precision step. |
| duty_cycle | input | Cycle Width | PWM pulse width configuration inpu; When duty_cycle_update transitions from 0 to 1, this value is loaded into the IP and used for PWM output. |
| duty_cycle_update | input | 1 | PWM pulse width update; When this signal transitions from 0 to 1, the value of duty_cycle is loaded into the IP and used for PWM output. |
| Initialization Signals | | | |
| initial_cycle | input | Cycle Width | Initial PWM period input |
| initial_duty_cycle | input | Cycle Width | Initial PWM pulse width input |
| initial_update | input | 1 | Load initialization configuration; When this signal transitions from 0 to 1, the initial settings are loaded into the IP. |

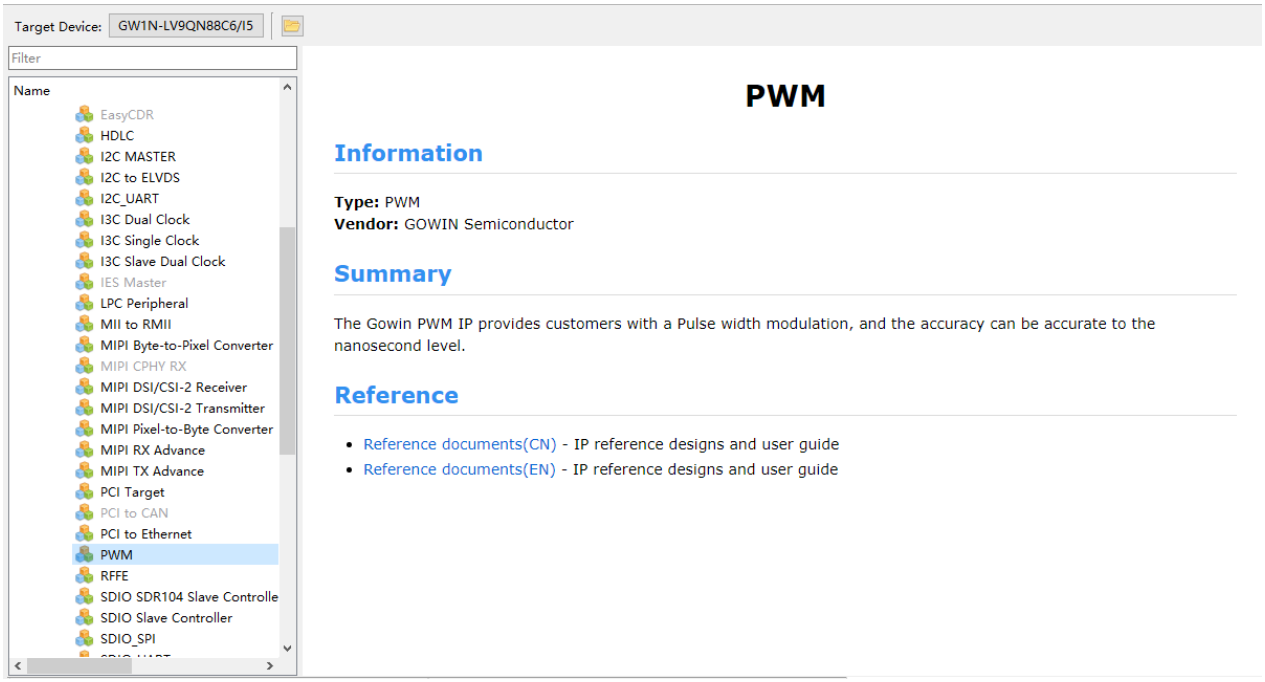
| Signal | I/O | Data Width | Description |
|---|---|---|---|
| PWM output | | | |
| pwm | output | 1 | PWM output signal |

# 6 Interface Configuration

You can invoke and configure Gowin PWM IP using the IP Core Generator tool in the IDE. This section will introduce the configuration interface, the configuration process, and the configuration options.

1. Open IP Core Generator.
   After creating the project, click the “Tools” tab in the upper left, click “IP Core Generator” to open Gowin IP Core Generator via the drop-down list, and select “PWM”, as shown in Figure 6-1.

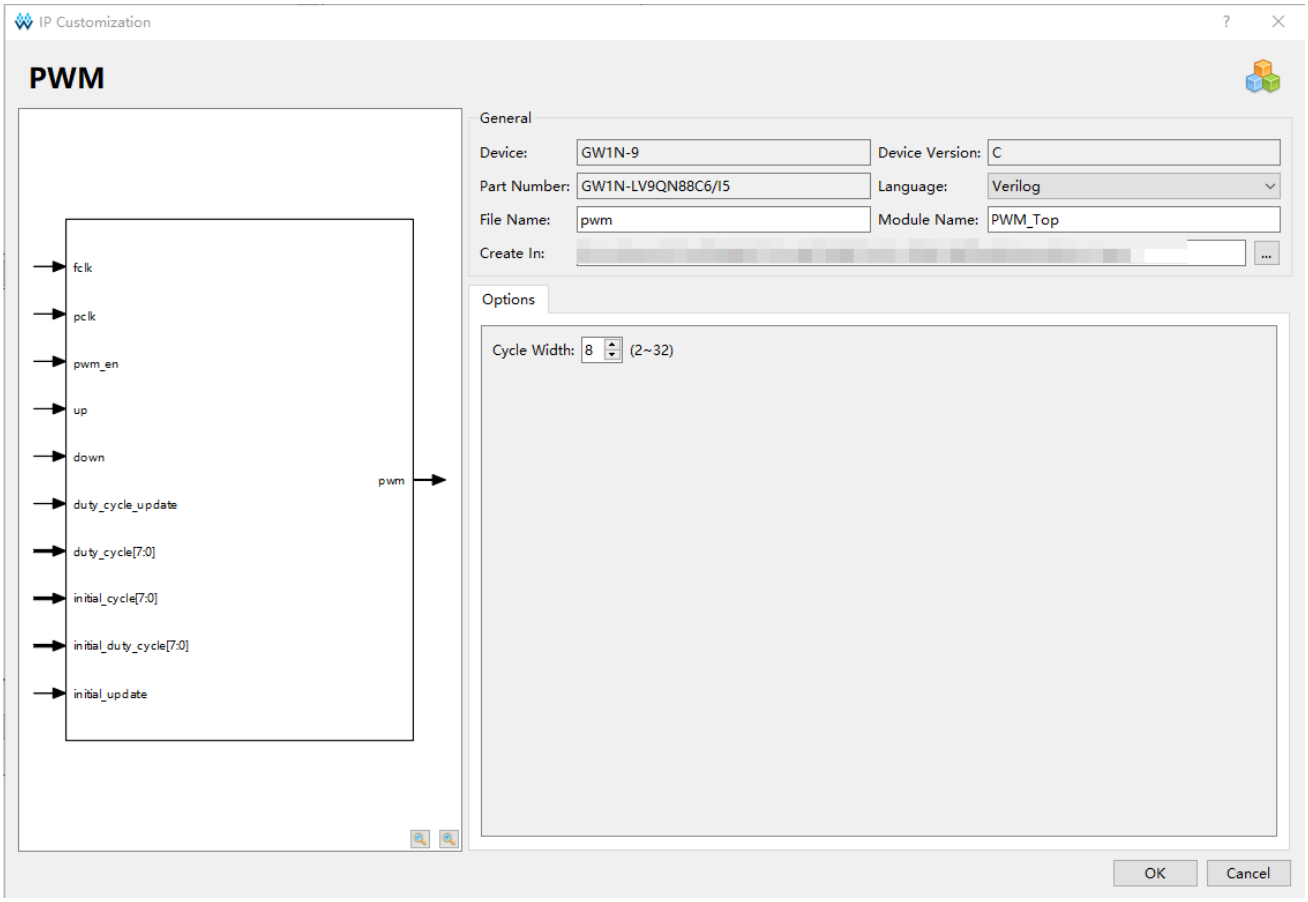
**Figure 6-1 Select PWM**




2. PWM Ports Interface

The left of the configuration interface is the Gowin PWM IP ports diagram, and the right is the IP configuration options, as shown in Figure 6-2.

**Figure 6-2 PWM IP Configuration Interface**



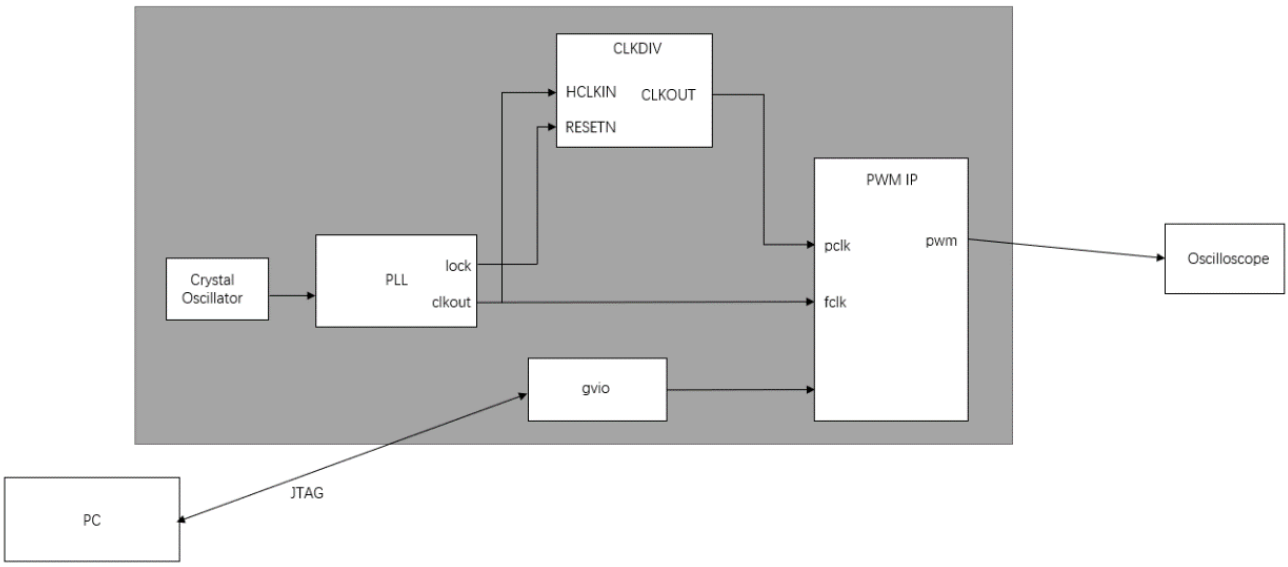

The parameter configuration descriptions are as follows:

Cycle Width: The input bit width for the PWM period and pulse width, ranging from 2 to 32. This parameter determines the input bit width of the following IP signals: duty_cycle, initial_cycle, and initial_duty_cycle. The value is calculated based on the target PWM period (Tcycle) and the frequency of fclk (F), using the following formula: Cycle Width=Integer part of [log2(2*F*Tcycle)+1].

# 7 Reference Design

For more information, see Gowin PWM IP *reference design* at Gowin website The reference design block diagram is as shown in Figure 7-1.

**Figure 7-1 Reference Design Block Diagram**




As shown in the figure above, Gowin PWM IP is instantiated in the reference design and the PLL and CLKDIV modules are invoked to provide the necessary clocks and signals for the IP. In this design, the GVIO tool is used to drive the control signals of Gowin PWM IP. Users can dynamically configure Gowin PWM IP by modifying the GVIO signal values through Gowin IDE via JTAG. Users can observe the PWM output using an oscilloscope.

The operation steps are as follows:

1. Download the fs to the development board.
2. Open Gowin Analyzer Oscilloscope tool in Gowin IDE.
3. Click Gvio Core “  ” button to enable the GVIO tool.

4. Configure initial_cycle = 0x200, initial_duty_cycle = 0xA, initial_update = 1, and pwm_en = 1 through gvio. After the above configuration, the PWM output generates the expected pulse.
5. Set up =1 via GVIO to increase the PWM pulse width by one precision step.
6. Set down =1 via GVIO to decrease the PWM pulse width by one precision step.
7. Set duty_cycle =0x20 and duty_cycle_update =1 via GVIO to update the PWM pulse width to the latest configuration.


GOWIN
PROGRAMMING FOR THE FUTURE