Gowin FOC Current Loop Control Light IP

User Guide

IPUG770-1.1E, 08/18/2023

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
03/16/2021	1.0E	Initial version published.
08/18/2023	1.1E	• IP ports updated. • Interface screenshots updated.

Contents

List of Figures

List of Tables

1 About This Guide

1.1 Purpose

The purpose of this Gowin FOC Current Loop Control Light IP User Guide is to help you to quickly understand the features and usage of Gowin FOC Current Loop Control Light IP by providing the descriptions of the functions, features, ports, timing, configuration, and reference design, etc. The software screenshots in this manual are based on 1.9.9 Beta-2. 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
- DS881, GW1NSER series of FPGA Products Data Sheet
- DS891, GW1NRF series FPGA Products Data Sheet
- DS102, GW2A series of FPGA Products Data Sheet
- DS226, GW2AR series of FPGA Products Data Sheet
- DS961, GW2ANR series of FPGA Products Data Sheet
- DS976, GW2A-55 Data SheetSUG100, 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
DSP	Digital Signal Processing
IP	Intellectual Property
RAM	Random Access Memory
LUT	Look-up Tables
FOC	Field Oriented Control
SVPWM	Space Vector Pulse Width Modulation
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 by the following ways.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

2 Overview

2.1 Introduction to FOC Current Loop Control Light IP

Gowin FOC Current Loop Control Light IP can realize FOC current loop and control the motor with less resources.

Table 2-1 Gowin FOC Current Loop Control Light IP

Gowin FOC Current Loop Control Light IP	
IP Core Application	
Logic Resource	Please refer to Table 3-1.
Delivered Doc.	
Design Files	Verilog (encrypted)
Reference Design	Verilog
TestBench	Verilog
Test and Design Flow	
Synthesis Software	GowinSythesis®
Application Software	Gowin Software (1.9.7.05 Beta and above)

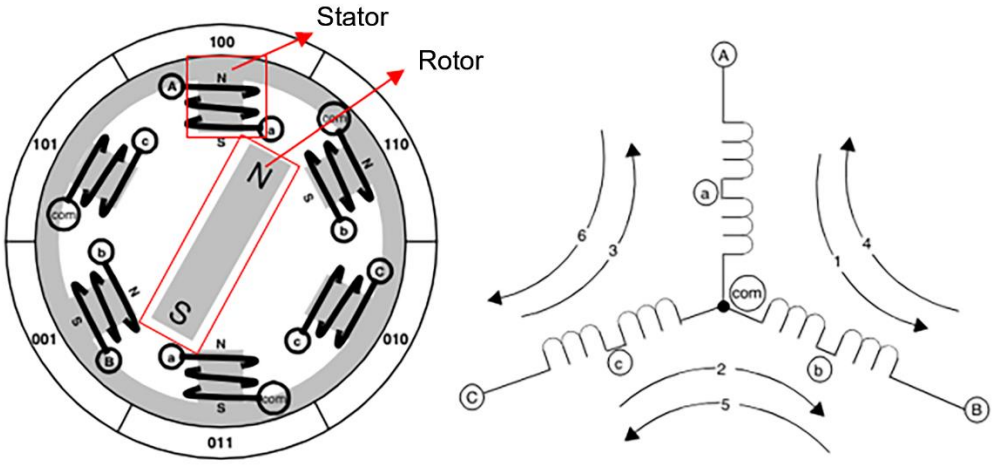
Note!

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

2.2 Introduction to FOC Current Loop Algorithm

2.2.1 Principle of Motor

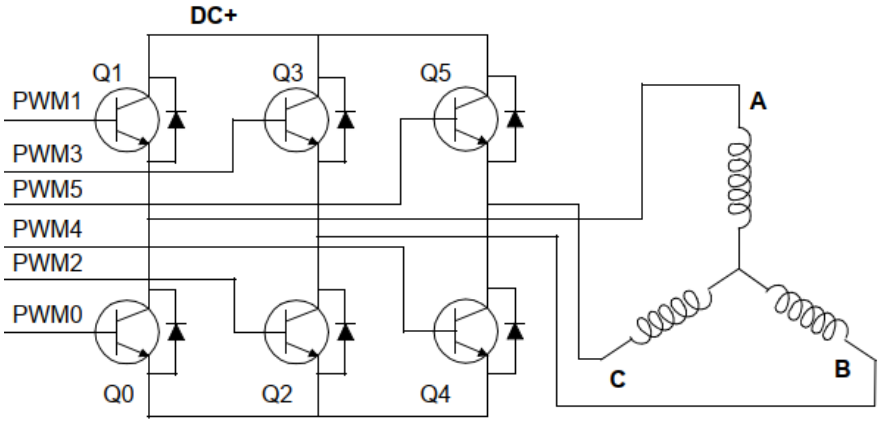
Figure 2-1 Motor Schematic



A motor is an electromagnetic device that converts or transfers electric energy according to the law of electromagnetic induction. It can produce drive torque to provide power for electrical appliances or machines.

The motor consists of stator and rotor. The motor rotates so that the current flowing through winding stator generates magnetic field. The rotor rotates due to the difference of polarities. The periodic current flows through three-phase bridge so that the rotor can rotate one round.

Figure 2-2 Schematic of Three-phase Bridge Driven by Motor

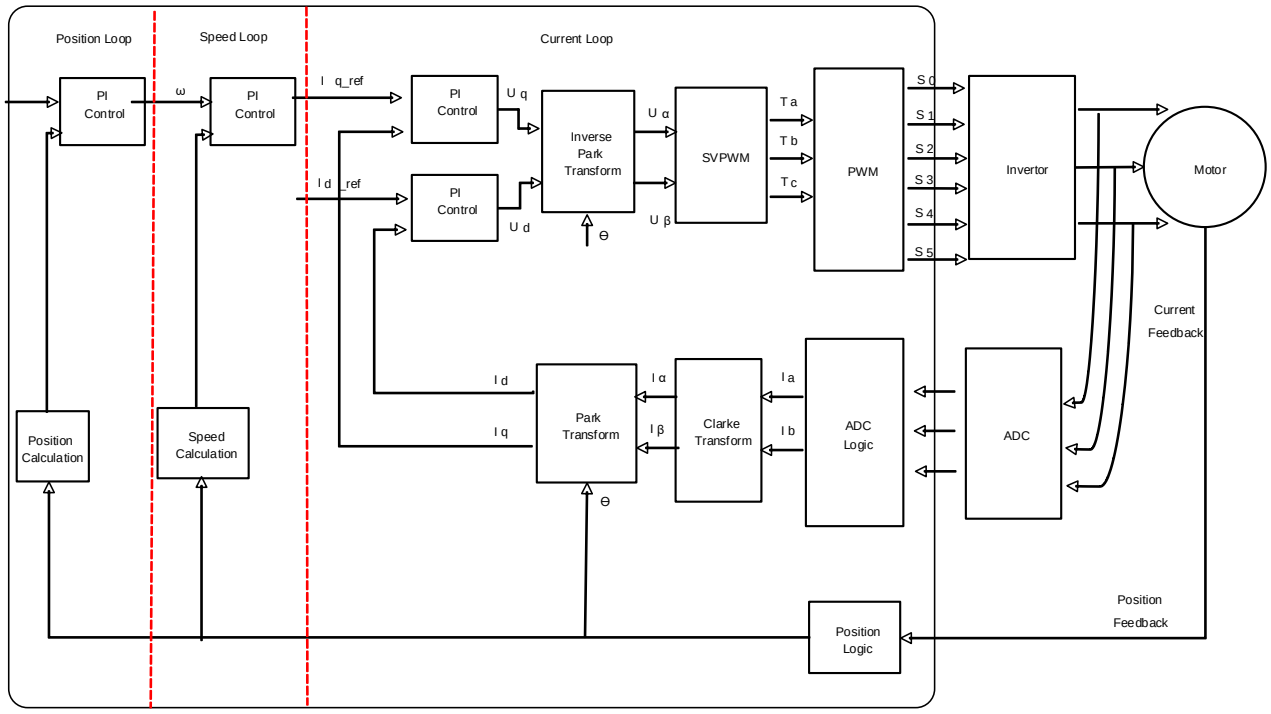


The three-phase bridge driven by motor-driven is shown in Figure 2-2. By turning on/off the switches, energize Phase A, Phase B, and Phase C to generate the magnetic field, making the rotor rotates. If Q1 and Q4 on and others off, the positive current flows from Q1 to the winding stator Phase A,

then the winding stator Phase C and flows back from Q4 to the negative pole. The current flowing through the Phase A and Phase C generates a magnetic field. According to electromagnetic induction and right hand law, it can be seen that the direction is parallel to Phase B. Therefore, the rotor turns to the position parallel to B under the magnetic field force. If to make the motor rotates continuously in a specified direction, the stator must be energized in a certain order, i.e. phase change.

2.2.2 Principle of FOC

Figure 2-3 FOC Schematic



FOC, also called vector control, controls the excitation current and torque current of the motor by measuring and controlling the stator current vector.

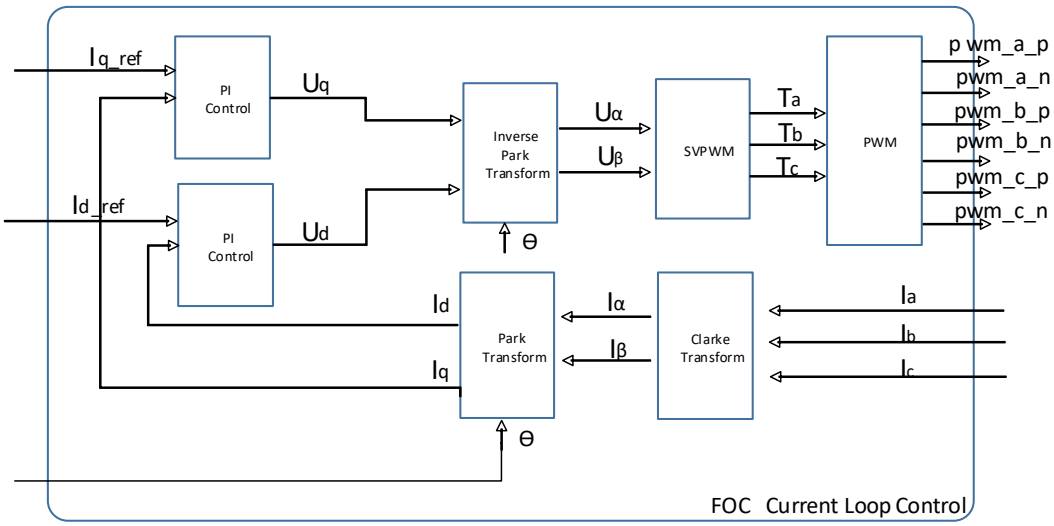
The typical FOC control is shown in Figure 2-3. The position of the rotor, the speed of the motor and the current are used as the feedback information. After a series of mathematical transformation of the three-phase current, the decoupling feedback value which is user-friendly and controllable is obtained. Then, dynamic adjustment is performed according to the error between the feedback value and the target value, and finally the three-phase sinusoidal wave is output to drive the motor to rotate.

FOC is generally controlled by three loops, namely current loop, speed loop and position loop. The current loop controls the torque of the motor, which is the basis of FOC control. While the speed loop and position loop

are adjusted by detecting the signal of the motor encoder. The PI output in the loops is the value given by the current loop, in other words, any mode must use the current loop, and the system is actually controlling the current (torque) to achieve the corresponding control of the speed and position.

2.2.3 Principle of FOC Current Loop

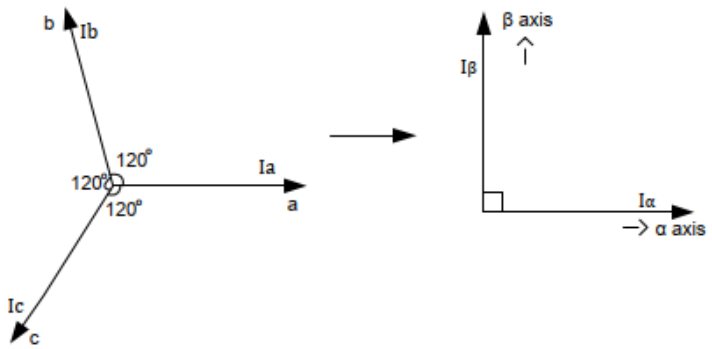
Figure 2-4 FOC Current Loop Schematic



FOC inner loop control is the core control of FOC. FOC current loop control is a process of converting three-phase current to mutually quadrature current, as shown in Figure 2-4. Firstly, the three-phase current of the motor is input through ADC sampling. The three-phase stationary coordinate system is converted to two-phase rotating coordinate system by Clarke transform and Park transform. The speed is adjusted by PI control; Then, the coordinate system is restored by Park inverse transform; SVPWM controls the voltage output to generate PWM; Finally, three-phase bridge completes the motor drive control.

2.2.4 Principle of Clarke Transform

Figure 2-5 Clarke Transform Schematic



Clarke transform is a process of transforming a three-axis two-dimensional coordinate system to a two-axis two-dimensional coordinate system, as shown in Figure 2-5.

$$I_\alpha = \frac{2}{3}I_a - \frac{1}{3}I_b - \frac{1}{3}I_c$$

$$I_\beta = \frac{1}{\sqrt{3}}(I_b - I_c)$$

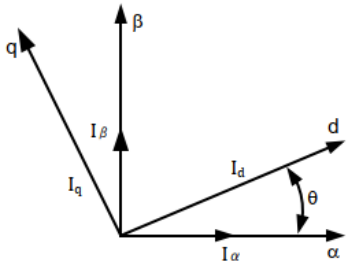
If $I_a + I_b + I_c = 0$, then

$$I_\alpha = I_a$$

$$I_\beta = \frac{1}{\sqrt{3}}(I_a + 2I_b)$$

2.2.5 Principle of Park Transform

Figure 2-6 Park Transform Schematic



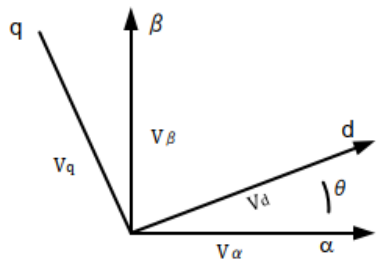
Park transform is to convert the stationary coordinate system to a rotating coordinate system, as shown in Figure 2-6.

$$I_d = I_\alpha \cos(\theta) + I_\beta \sin(\theta)$$

$$I_q = I_\beta \cos(\theta) - I_\alpha \sin(\theta)$$

2.2.6 Principle of Park Inverse Transform

Figure 2-7 Park Inverse Transform Schematic



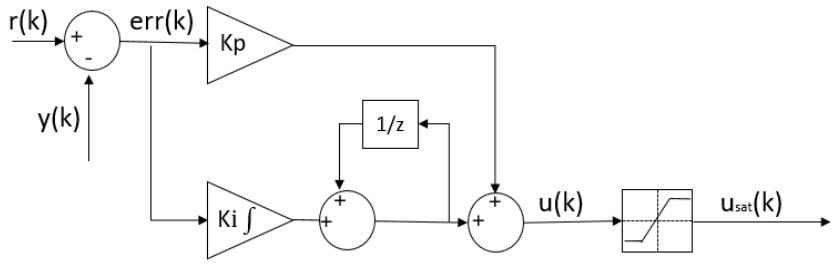
Park inverse transform is to convert the rotating coordinate system to a stationary coordinate system, as shown in Figure 2-7.

$$V_\alpha = V_d \cos(\theta) - V_q \sin(\theta)$$

$$V_\beta = V_q \cos(\theta) + V_d \sin(\theta)$$

2.2.7 Principle of PI Control

Figure 2-8 PI Control Schematic

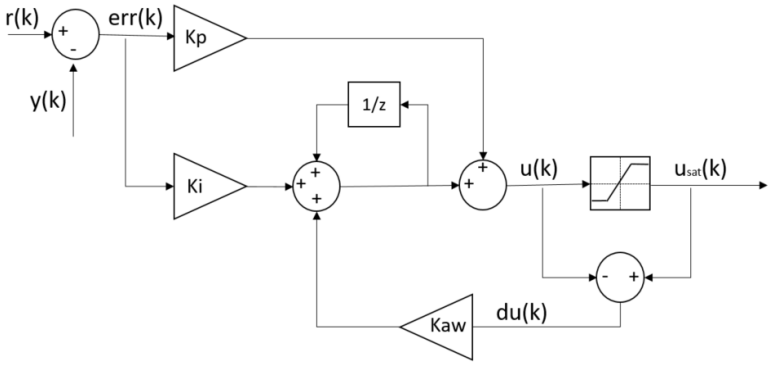


$$u(k) = Kp * err(k) + Ki * err(k) * (1+\frac{1}{z})$$

PI controller controls the deviation by proportionality factor and integrating factor. As shown in Figure 2-8, the actual value r(k) and the reference value y(k) generate err (k). PI controller controls the deviation by proportionality factor Kp and integrating factor Ki.

PI control is usually designed in a linear region, without considering the input limit of the control object, resulting in excessive deviation output, and the closed loop system will correspondingly unstable. This phenomenon is called Wind-up. In order to solve this problem, the PI controller of Anti-Windup is designed as shown in Figure 2-9.

Figure 2-9 Anti-Windup PI Schematic



$$u(k) = Kp * err(k) + \left(Ki + Kaw * du(k)\frac{z}{z-1}\right) err(k)$$

The PI controller of Anti-Windup reduces the influence of integrator by feeding back the deviation of input and output to integrator to suppress Wind-up.

2.2.8 Principle of SVPWM

SVPWM space vector pulse width modulation is a reference coordinate system with the stator of the three-phase motor as the ideal magnetic chain. The system switches on/off by controlling three-phase bridge tube, thus generating the PMW waveform approximating the ideal magnetic chain.

Figure 2-10 SVPWM Sectors

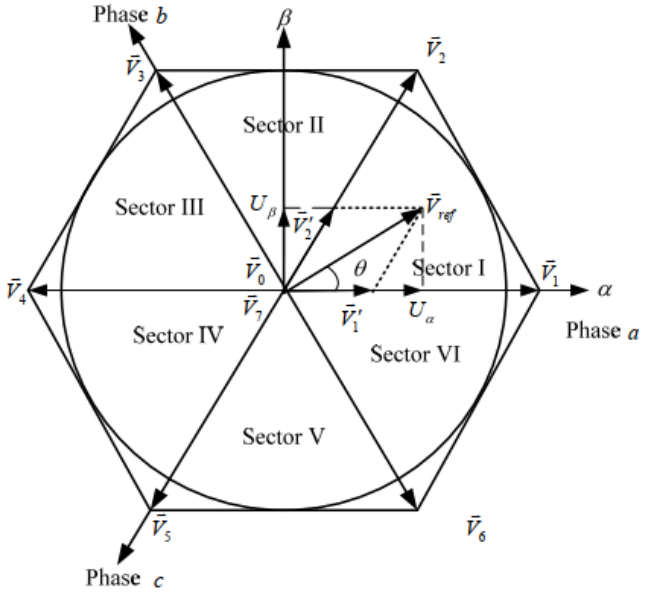
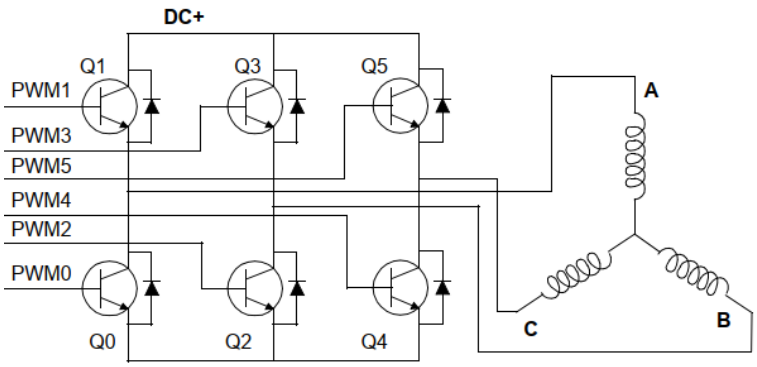


Figure 2-11 Schematic of Three-phase Bridge Driven by Motor



SVPWM divides the space into six sectors, and each sector corresponds to the three-phase bridge switching tube by encoding (A, B, C), wherein 0 indicates that the upper tube is off and the lower tube is on, and 1 indicates that the upper tube is on and the lower tube is off. For example, sector Ⅰ (1, 0, 0) indicates that Q1, Q2, Q4 are on and other tubes are off. Then control the duration of switching current to make rotor rotates to the angular position of the sector Ⅰ.

Figure 2-12 Sector Distribution Diagram

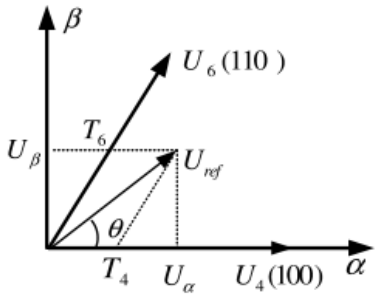
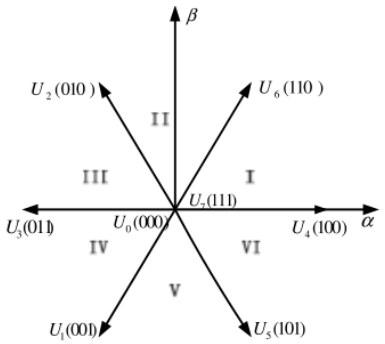


Figure 2-13 Synthesized Vector Diagram of Sector Ⅰ



The expected output voltage is obtained U_{ref} by synthesizing vector U4T4 and U6T6, as shown in Figure 2-12 and Figure 2-13.

$$\int_0^T U_{ref} = \int_0^{T4} U_4 + \int_{T4}^{T4+T6} U_6 + \int_{T4+T6}^{T} U_0$$

$$U_{out}\ T = U_4T_4 + U_6T_6 + U_0T_{T-T4-T6} = U_xT_x + U_yU_y + \ U_0T_0$$

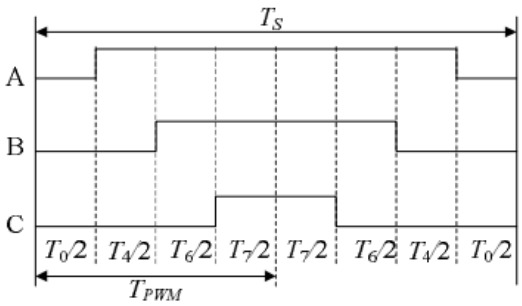
U_{ref} is the expected output voltage, T is a cycle. T_x, T_y, T_0 is the time of corresponding coordinate system and zero vector U_0 taken in one cycle.

The actual pulse width modulation waveform is obtained by calculating the vector switching duration. Selecting appropriate zero vector can reduce the number of switching. Avoiding the switching at the time of high load current can minimize the switch loss. Therefore, in order to reduce the number of switching, the allocation of the basic vector needs to set as follows: In every switching, only change the switching, and equally distribute the zero vector in time, so that the generated PWM is symmetrical and the harmonic component of PWM is effectively reduced.

Taking the sector Ⅰ as an example, the voltage vector is U0, U4, U6,

U7, U7, U6, U4, U0, and PWM output is shown in Figure 2-14.

Figure 2-14 PWM Output of Sector Ⅰ



The switching order of other sectors is shown in Table 2-2.

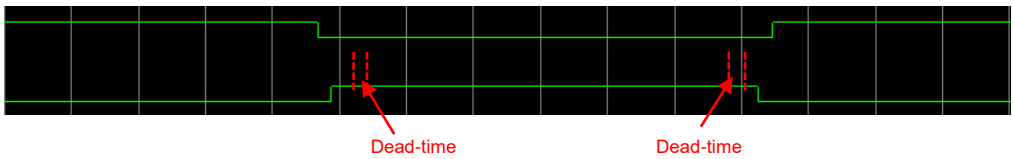
Table 2-2 Switching Order of Sectors

Sector	U_{ref} Location	Switching Order
Sector Ⅰ	0≤θ≤60	…0-4-6-7-7-6-4-0…
Sector Ⅱ	60≤θ≤120	…0-2-6-7-7-6-2-0…
Sector Ⅲ	120≤θ≤180	…0-2-3-7-7-3-2-0…
Sector Ⅳ	180≤θ≤240	…0-1-3-7-7-3-1-0…
Sector Ⅴ	240≤θ≤300	…0-1-5-7-7-5-1-0…
Sector Ⅵ	300≤θ≤360	…0-4-5-7-7-5-4-0…

2.2.9 Principle of PWM Dead-time Compensation

Due to the self-property of switching tube, switching on and off both need a certain time. And the time of switching off is longer than that of switching on. Therefore, if switching tube is switched on/off by the control of ideal trigger signal, it will lead to the shoot-through of two switching tubes on the same bridge arm, damaging the switch. To prevent shoot-through, a certain delay time must be inserted between switching on and off, as shown in Figure 2-15. And the delay time is called dead-time compensation.

Figure 2-15 Dead-time Compensation Diagram



3 Features and Performance

3.1 Features

- Simplified resource structure
- PWM output with dead-time compensation

3.2 Max. Frequency

The maximum frequency of Gowin FOC Current Loop Control Light IP is mainly determined by the speed grade of the devices, and it can reach up to 110M.

3.3 Latency

The latency of FOC Current Loop Control Light IP is about 86 cycles.

3.4 Resource Utilization

FOC Current Loop Control Light is implemented by Verilog. Its performance and resource utilization may vary when the design is employed in different devices, or at different density, speed, grade or the different modes of IP configuration.

Take the GW2A-18 series of FPGA as an example. The resource utilization of FOC Current Loop Control Light IP is shown in Table 3-1. For the applications on the other GOWINSEMI devices, see the later release.

Table 3-1 The Resource Utilization of FOC Current Loop Control Light IP

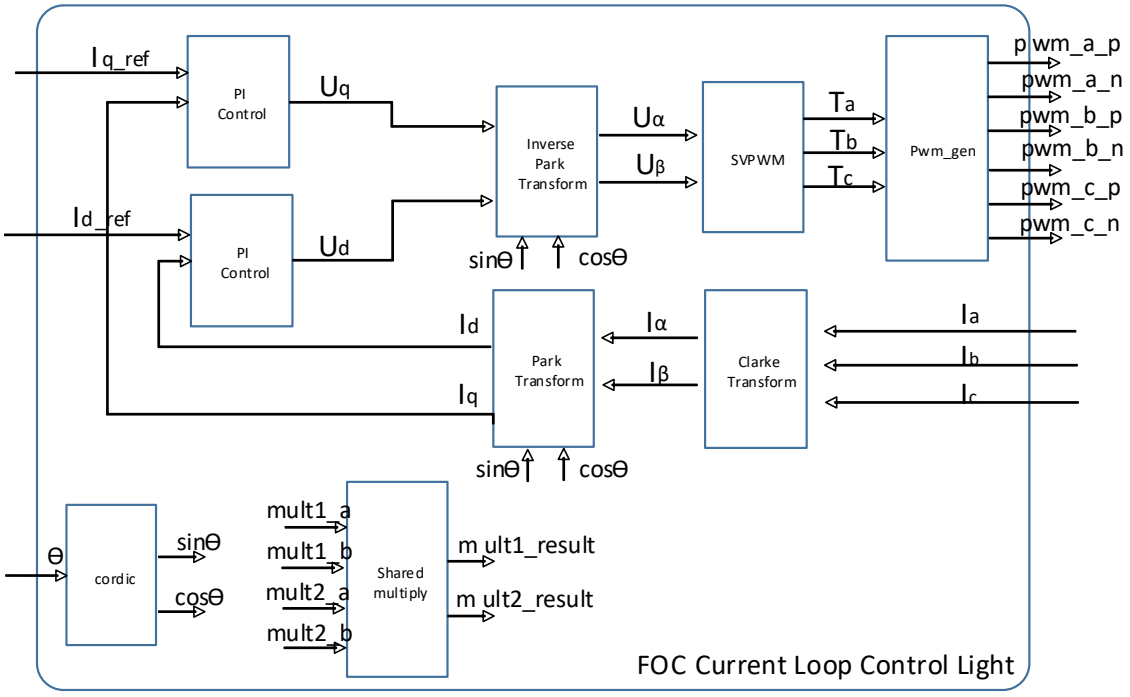
Device Series	Speed Grade	Name	Resource Utilization	Remarks
GW2A-18	-7	Registers	1696	–
		LUT	1765	
		ALU	615	
		DSP	3	
		BSRAM	0	

4 Functional Description

The Structure and Function of FOC Current Loop Control Light IP

FOC Current Loop Control Light IP can be implemented in FPGA, as shown in Figure 4-1.

Figure 4-1 Implementation Block Diagram of FOC Current Loop Control Light IP



FOC current loop control light IP is a process of decoupling three-phase current to mutually quadrature current, as shown in Figure 4-1. Firstly, collect the three-phase current and electrical angle of the motor, and Clarke transform and Park transform coordinate to convert the three-phase stationary coordinate system into two-phase rotating coordinate system and decouple into easily controllable torque component and magnetic flux component. Then the current and electrical angle are regulated by PI control, and the coordinate system is restored by Park

inverse, and finally the motor drive board is driven by SVPWM modulation + PWM.

FOC current loop control Light IP consists of the following modules.

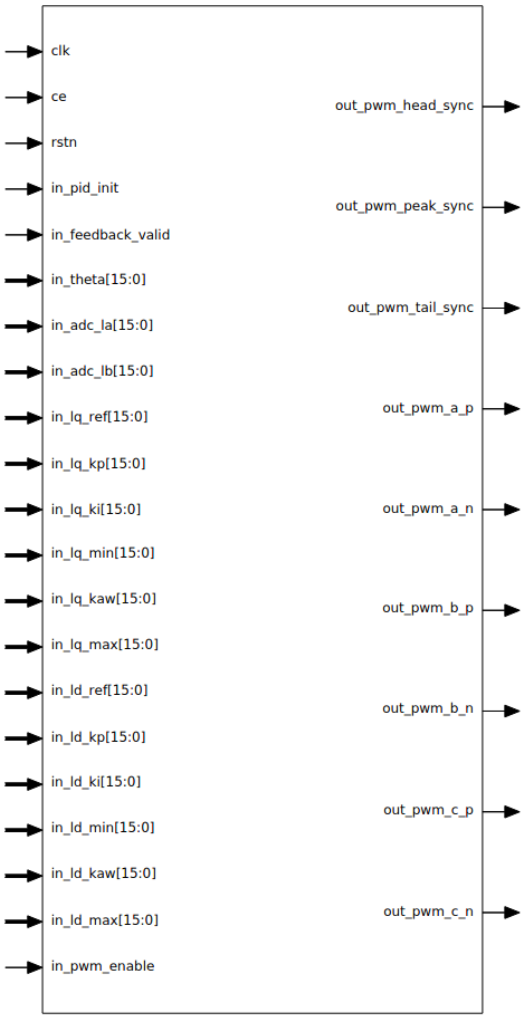
- Clarke Transform: The ADC-sampled phase currents I_a, I_b, I_c (if $I_a + I_b + I_c = 0$ is satisfied, then only need I_a and I_b.) are transformed by the coordinate system, converting the original three-phase stationary coordinate system into a two-phase stationary coordinate system, and outputting I_α and I_β.
- Park Transform: Converts the two-phase stationary coordinate system, I_α and I_β, to the rotating coordinate system. And Outputs I_d and I_q.
- PI Control: Controls proportionality factor and integrating factor of the deviation generated by I_q, I_{q_ref}, I_d and I_{d_ref}. Calculates and outputs U_q and U_d.
- Inverse Park Transform: Converts the rotating coordinate, U_q and U_d, to a stationary coordinate system, U_α and U_β.
- SVPWM: Performs SVPWM modulation according to U_α and U_β, and it outputs phase vector time T_a, T_b and T_c.
- PWM: Outputs PWM wave based on the phase vector time.

5 Port Description

5.1 Introduction to the Port of FOC Current Loop Control Light IP

The I/O ports of the Gowin FOC Current Loop Control Light IP is shown in Figure 5-1.

Figure 5-1 I/O Port Diagram of FOC Current Loop Control Light IP



The I/O ports details of FOC Current Loop Control Light IP are shown in Table 5-1.

Table 5-1 I/O Port List of FOC Current Loop Control Light IP

Signal	I/O	Data Width	Description
clk	Input	1	Working clock
rstn	Input	1	Reset signal, asynchronous reset, active-low. Synchronous reset and asynchronous release recommended.
ce	Input	1	Clock Enable Signal
in_pid_init	Input	1	Initialize the internal signal of the Pi Controller to 0
Feedback input (phase current & electrical angle)			
in_feedback_valid	Input	1	Feedback signal valid indicator, actively high
in_adc_Ia	Input	16	U-phase current, signed

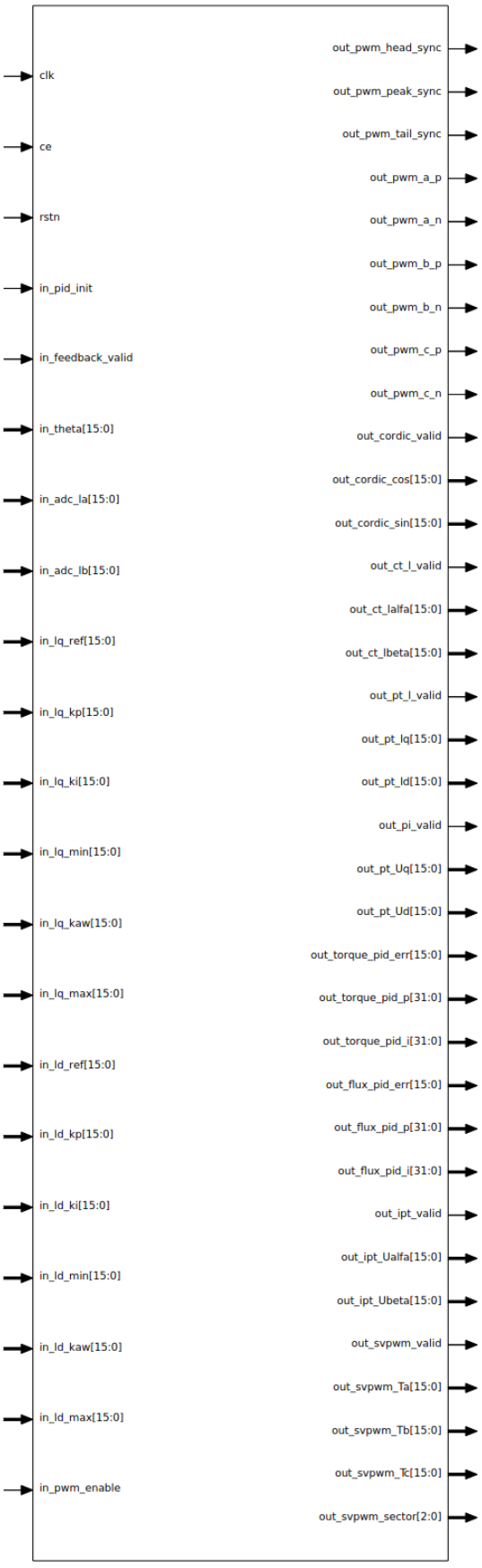
Signal	I/O	Data Width	Description
in_adc_Ib	Input	16	V-phase current, signed
in_adc_Ic	Input	16	W-phase current, signed (When 3-input Clark Transform is selected in IP GUI, the signal exists.)
in_theta	Input	16	Electrical angle, unsigned, 0~65536 corresponding to 0~360 degrees
Torque PI parameter			
in_Iq_ref	Input	16	The reference value of torque PI parameter, signed
in_Iq_kp	Input	16	The proportionality parameter kp of torque PI , signed If kp=4000, scaling factor=12 (scaling factor is set in IP GUI), then the actual proportionality parameter of the PI controller is 4000/2^12.
in_Iq_ki	Input	16	The integrating parameter ki of torque PI, signed If ki=4000, scaling factor=12 (scaling factor is set in IP GUI), then the actual proportionality parameter of the PI controller is 4000/2^12.
in_Iq_min	Input	16	The min. saturation limit value output by torque PI, signed
in_Iq_kaw	Input	16	anti-windup coefficient in PI controller
in_Iq_max	Input	16	The Max. saturation limit value output by torque PI, signed
Magnetic flux PI parameter			
in_Id_ref	Input	16	The reference value of Magnetic flux PI parameter, signed
in_Id_kp	Input	16	The proportionality parameter kp of Magnetic flux PI, signed. Usage is as above.
in_Id_ki	Input	16	The integrating parameter ki of Magnetic flux PI, signed. Usage is as above.
in_Id_min	Input	16	The min. saturation limit value output by Magnetic flux PI, signed
in_Id_kaw	Input	16	anti-windup coefficient in PI controller
in_Id_max	Input	16	The Max. saturation limit value output

Signal	I/O	Data Width	Description
			by Magnetic flux PI, signed
PWM			
In_pwm_enable	Input	1	PWN signal enable, active-high
out_pwm_head_sync	Output	1	The initial value indicator of internal triangular-wave counter, active-high
out_pwm_peak_sync	Output	1	Intermediate value indicator of internal triangular-wave counter, active-high
out_pwm_tail_sync	Output	1	Tail value indicator of internal triangular-wave counter, active-high
out_pwm_a_p	Output	1	PWM Output
out_pwm_a_n	Output	1	
out_pwm_b_p	Output	1	
out_pwm_b_n	Output	1	
out_pwm_c_p	Output	1	
out_pwm_c_n	Output	1	

5.2 Debug Mode Port of FOC Current Loop Control Light IP

The I/O ports of the debug mode of FOC Current Loop Control Light IP are shown in Figure 5-2.

Figure 5-2 Debug Mode Port Diagram of FOC Current Loop Control Light IP



The detail I/O ports of the debug mode of FOC Current Loop Control Light IP are shown in Figure 5-2.

Table 5-2 Debug Mode Port List of FOC Current Loop Control Light IP

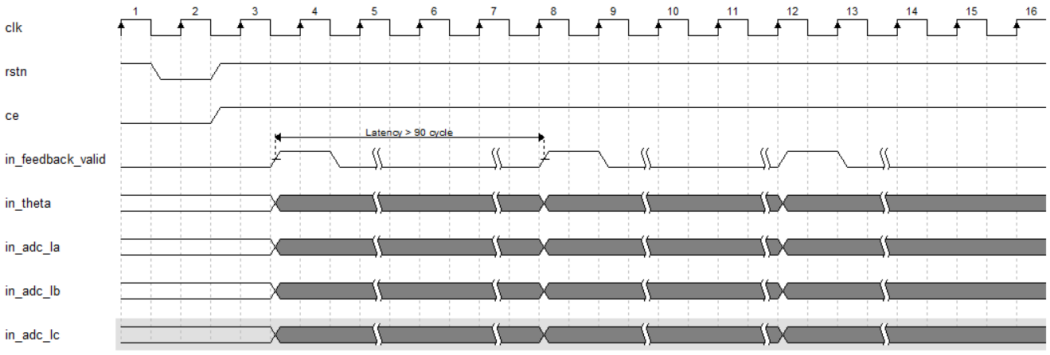
Signal	I/O	Data Width	Description
Cordic module output			
out_cordic_valid	Output	1	Cordic module output valid signal, active-high
out_cordic_cos	output	18	The sin value output of Cordic module, signed
out_cordic_sin	output	18	The cos value output of Cordic module, signed
Clarke Transform module output			
Out_ct_I_valid	Output	1	Clarke Transform module output valid signal, actively high
Out_ct_Ialfa	Output	18	The I_{α} value output of Clarke Transform module, signed
Out_ct_Ibeta	Output	18	Clarke Transform module output I_{β} value, signed
Park Transform module output			
Out_pt_I_valid	Output	1	Park Transform module output valid signal, actively high
Out_pt_Iq	Output	18	The I_{q} value output of Park Transform module, signed
Out_pt_Id	Output	18	Park Transform module output I_{d} value, signed
PI Control module			
Out_pi_valid	Output	1	PI Control module output valid signal, actively high
Out_pi_Uq	Output	18	The U_{q} value output of torque PI Control module, signed
Out_torque_pid_err	Output	18	The error calculation value output of torque PI Control module, signed
Out_torque_pid_p	Output	31	The proportionality calculation value output of torque PI Control module, signed
Out_torque_pid_i	Output	31	The integrating calculation value output of torque PI Control module, signed
Out_pi_Ud	Output	18	The U_{d} value output of magnetic flux PI Control module, signed

Signal	I/O	Data Width	Description
Out_flux_pid_err	Output	18	The error calculation value output of magnetic flux PI Control module, signed
Out_flux_pid_p	Output	31	The proportionality calculation value output of magnetic flux PI Control module, signed
Out_flux_pid_i	Output	31	The integrating calculation value output of magnetic flux PI Control module, signed
Inverse Park Transform module			
Out_ipt_valid	Output	1	Inverse Park Transform module output valid signal, actively high
Out_ipt_Ualfa	Output	18	The I_q value output of Inverse Park Transform module, signed
Out_ipt_Ubeta	Output	18	Inverse Park Transform module output I_d value, signed
SVPWM module			
Out_svpwm_valid	Output	1	SVPWM module output valid signal, active-high
Out_svpwm_Ta	Output	18	The A-phase vector time output of SVPWM module, unsigned
Out_svpwm_Tb	Output	18	The B-phase vector time output of SVPWM module, unsigned
Out_svpwm_Tc	Output	18	The C-phase vector time output of SVPWM module, unsigned
Out_svpwm_sector	Output	3	The sector location output of SVPWM module, unsigned
Open Loop Debug mode			
In_svpwm_valid	Input	1	SVPWM module input valid signal, actively high
In_svpwm_Ualfa	Input	18	The U_α value input of SVPWM module, signed
In_svpwm_Beta	Input	18	SVPWM module output U_β value, signed

6 Timing Description

This chapter describes the timing of Gowin FOC Current Loop Control Light IP. The timing of FOC Current Loop Control Light IP in input status is shown in Figure 6-1.

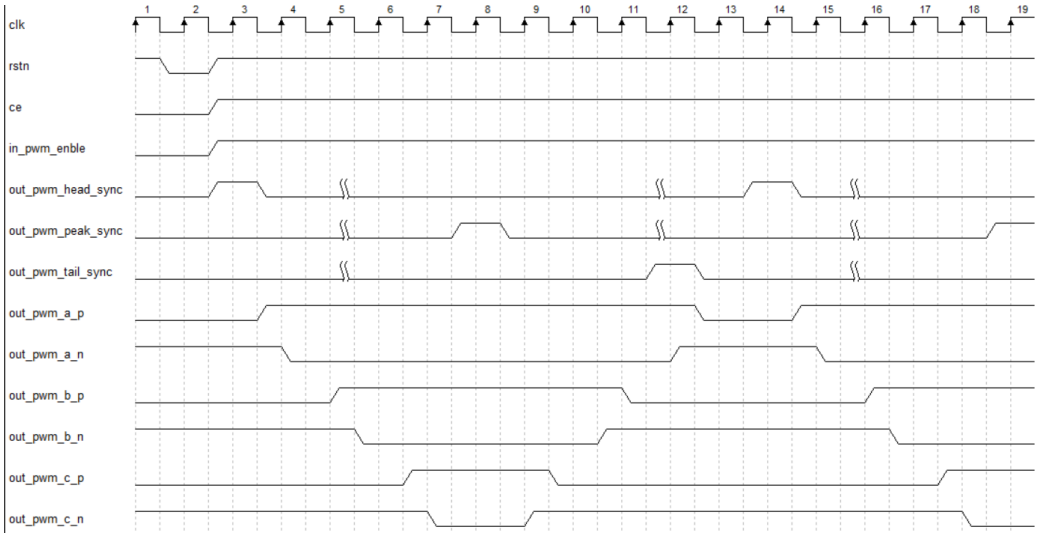
Figure 6-1 Timing of FOC Current Loop Control Light IP in Input Status



1. Each time a valid signal (in_feadback_valid) is input high, the feedback angle signal and the feedback current signal are input validly.
2. A more than 90-cycle delay between valid signals is required to ensure the operation of internal modules.

The timing of FOC Current Loop Control Light IP in output status is shown in Figure 6-2.

Figure 6-2 Timing of FOC Current Loop Control Light IP in Output Status



1. Set the signal (in_pwm_enable) to high to make the PWM module work.
2. The PWM uses an internal triangular-wave counter, which outputs a initial indicator signal, an intermediate indicator signal and a tail indicator signal that are valid at high level each time.
3. Finally, output three-phase PWM.

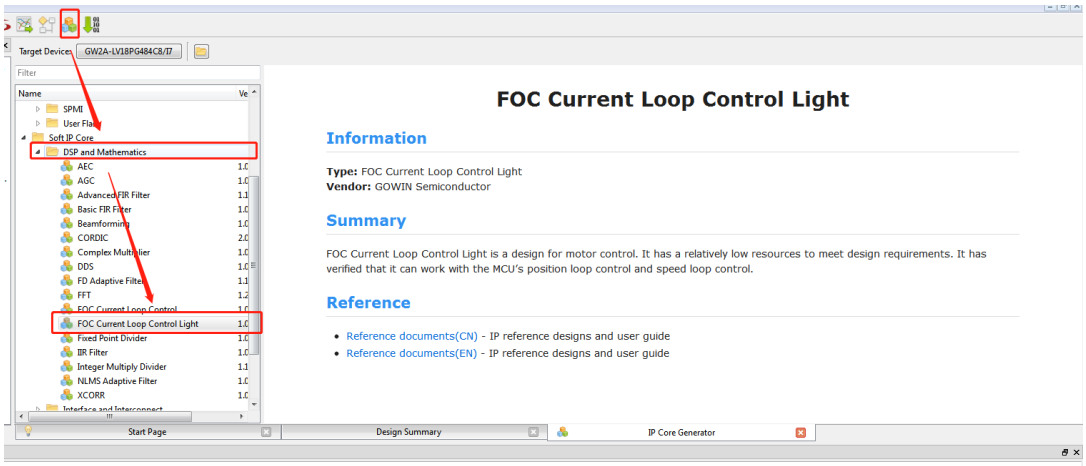
7 Interface Configuration

Select Tools in the Gowin software GUI, and it can start the IP Core Generator to invoke and configure FOC Current Loop Control Light IP.

1. Open IP Core Generator

After creating a new project, you can click the “Tools” button to open the IP Core Generator, as shown in Figure 7-1.

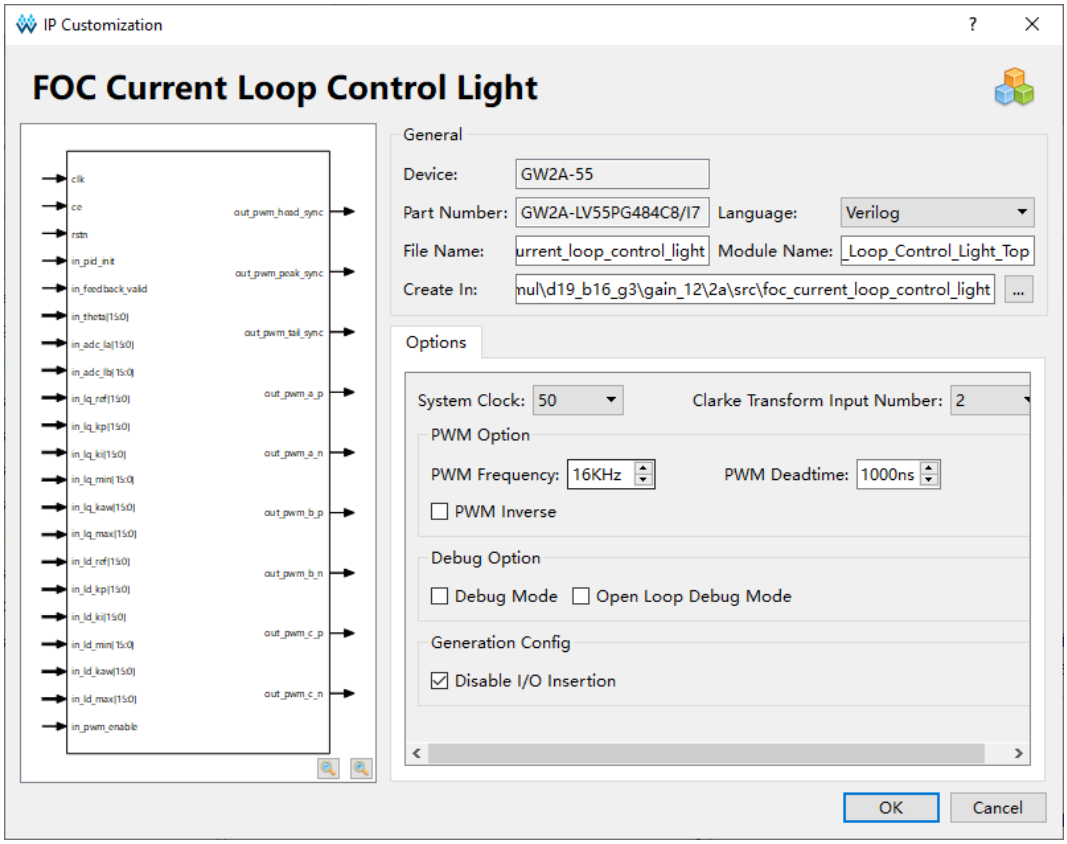
Figure 7-1 IP Core Generator GUI



2. Open IP core FOC Current Loop Control Light.

Click “DSP and Mathematics”, then double-click “FOC Current Loop Control Light” to open GUI, as shown in Figure 7-2.

Figure 7-2 IP FOC Current Loop Control Light GUI



3. Configure IP core FOC Current Loop Control Light

IP configuration mainly includes: System Clock, Clarke Transform Option, PWM Option, and Debug Option.

- System Clock configures input IP clock frequency
- Clarke Transform Option configures the input mode of Clarke Transform module (including 2 and 3 inputs).
- PWM Option configures PWM frequency, PWM dead-time, PWM signal whether need to reverse.
- If Debug Option is selected, test signal is output. If Open Loop Debug Mode is selected, Open Loop test is conducted.

8 Reference Design

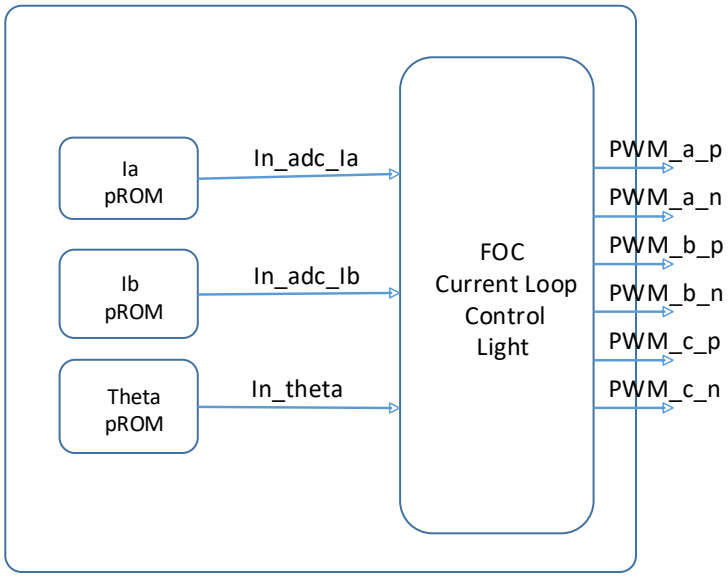
This chapter introduces the creation and usage of FOC Current Loop Control Light reference design. FOC Current Loop Control Light design instance has only one module. See the Current Loop Control reference design for details.

Application

The block diagram of reference design is as shown in Figure 8-1. In the design instance, the steps are as follows:

1. The input data (Theta, Ia, Ib, Ic) is stored in pROM, then input to FOC Current Loop Control module.
2. After the operation in IP, PWM output is obtained.

Figure 8-1 Block Diagram of Reference Design



FOC Current Loop Control Light functions can be verified quickly using this design. When the reference design is applied to board level test, you need to provide appropriate incentive to the reference design and use an

on-line logic analyzer or oscilloscope to observe the signals.


GOWIN
PROGRAMMING FOR THE FUTURE