



Gowin goConfig UART IP User Guide

IPUG1197-1.0E, 2025-02-14

Copyright © 2025 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN高云, **GOWIN**, , GOWINSEMI, GOWIN, Gowin, 高云, 晨熙, 小蜜蜂, LittleBee, -V, GowinPnR, GoBridge are trademarks of Guangdong Gowin Semiconductor Corporation and are registered in China and/or other countries or regions. All other words and logos identified as trademarks or service marks are the property of their respective holders. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

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
02/14/2025	1.0E	Initial version published.

Contents

List of Figures.....	iii
List of Tables.....	iv
1 About This Guide.....	1
1.1 Purpose.....	1
1.2 Related Documents.....	1
1.3 Terminology and Abbreviations.....	2
1.4 Support and Feedback.....	2
2 Function Introduction.....	3
2.1 Background Programming Overview.....	3
2.2 UART Interface.....	4
2.2.1 UART Features.....	4
2.2.2 Flash Interface Features.....	5
2.3 IP Features.....	5
2.4 Resource Utilization.....	6
3 Functional Description.....	7
3.1 Gowin goConfig UART IP Structure Diagram.....	7
3.2 Gowin goConfig UART IP Control Commands.....	8
3.3 Flash Interface Operations.....	12
3.3.1 SPI Interface Commands.....	12
3.4 UART Command and SPI Interface Mapping.....	15
4 Signal Definition.....	18
5 Interface Configuration.....	20
6 Reference Design.....	23
6.1 Project Setup Procedure.....	23
6.1.1 Configure and Generate IP.....	23
6.1.2 Call the IP at the Top Level.....	23
6.1.3 Software Configuration.....	26
6.2 Reference Design.....	28
6.2.1 FPGA Project Description.....	28
6.2.2 Reference Design Files.....	29

6.2.3 Reference Design Considerations.....	29
--	----

List of Figures

Figure 2-1 Example of UART Frame Composition.....	4
Figure 2-2 SPI Interface Timing.....	5
Figure 3-1 Block Diagram.....	7
Figure 3-2 Sector Erase Command.....	9
Figure 3-3 Page Program Command.....	9
Figure 3-4 Page Read Command.....	10
Figure 3-5 Status Command.....	10
Figure 3-6 Read Flash ID Command Example.....	11
Figure 3-7 Reset IP Command Example.....	11
Figure 3-8 Reconfig Command Example.....	11
Figure 3-9 Write Enable (06H).....	13
Figure 3-10 Write Disable (04H).....	13
Figure 3-11 Read Status Register-1 (05H).....	14
Figure 3-12 Sector Erase (20H).....	14
Figure 3-13 Page Program (02H).....	14
Figure 3-14 Read JEDEC ID (9FH).....	15
Figure 3-15 Read Data (03H).....	15
Figure 3-16 Erase Command Mapping.....	16
Figure 3-17 Write Data Command Mapping.....	17
Figure 3-18 Read Data Command Mapping.....	17
Figure 3-19 Read Flash ID Command Mapping.....	17
Figure 5-1 IP Core Generator.....	20
Figure 5-2 Open Gowin goConfig UART IP Core.....	21
Figure 5-3 Gowin goConfig UART IP Core Configuration Interface.....	22
Figure 6-1 Example of Invoking IP IO Names.....	24
Figure 6-2 Example of Top-level IO Definitions.....	25
Figure 6-3 Example One of Starting Configuration.....	26
Figure 6-4 Example Two of Starting Configuration.....	27
Figure 6-5 Check "Use MSPI as regular IO".....	28
Figure 6-6 Error Example.....	28

Figure 6-7 Reference Design Block Diagram..... 29

List of Tables

Table 1-1 Terminology and Abbreviations.....	2
Table 2-1 Gowin goConfig UART IP Overview.....	3
Table 2-2 SPI Interface Timing Description.....	5
Table 2-3 Resource Utilization.....	6
Table 3-1 Command Definition.....	8
Table 3-2 Status Bit Table.....	10
Table 3-3 Typical Value of ST_data.....	10
Table 3-4 SPI Interface Commands.....	12
Table 4-1 IO Definitions.....	18
Table 5-1 Parameter Configuration.....	22
Table 6-1 src Folder List in Reference Design.....	29

1 About This Guide

1.1 Purpose

Gowin goConfig UART IP User Guide is to help you learn the features and usage of this IP by providing the descriptions of functions, signals, and interface configuration, etc. The software screenshots and the supported products listed in this manual are based on Gowin Software 1.9.11. 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*](#)
- [*DS102, GW2A series of FPGA Products Data Sheet*](#)
- [*DS981, GW5AT series of FPGA Products Data Sheet*](#)
- [*DS1103, GW5A series of FPGA Products Data Sheet*](#)
- [*SUG100, Gowin Software User Guide*](#)
- [*SUG283, Gowin Primitive User Guide.*](#)
- [*UG290, Gowin FPGA Products Programming and Configuration 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
FPGA	Field Programmable Gate Array
IP	Intellectual Property
JTAG	Joint Test Action Group
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver/Transmitter

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 Function Introduction

Background programming is a function that has been in high demand by users in recent years. In the use of Gowin FPGA products, there is "Function A", if you want to upgrade or re-program "Function A" to "Function B", and in this process, "Function A" will not stop working; once the upgrade is completed, the upgraded "Function B" can be immediately loaded to the chip, which requires the background programming.

Gowin goConfig UART IP can realize the on-line update function of UART port. Users can program the FPGA through UART port to replace the data stored internally, realizing the background programming.

Table 2-1 Gowin goConfig UART IP Overview

Gowin goConfig UART IP Overview	
Logic Resource	Please refer to Table 2-3 .
Delivered Doc.	
Design Files	Verilog (encrypted)
Reference Design	Verilog
TestBench	Verilog
Test and Design Flow	
Synthesis Software	GowinSynthesis
Application Software	Gowin Software (V1.9.11 and above)

Note!

For the devices supported, you can click [here](#) to get the information.

2.1 Background Programming Overview

The programming includes two processes: Program Flash and Configure SRAM.

- Program Flash: The process of writing the bitstream file generated by Gowin Software to Flash (hereinafter referred to as Write to Flash).

- **Configure SRAM:** The process of loading the bitstream data from the Flash into the SRAM in the FPGA (hereinafter referred to as Load to SRAM).

Note!

- **Normal programming:** When programming, SRAM is erased (i.e., the running function stops) and then write to Flash.
- **Background programming:** When programming, the SRAM is not erased (i.e., the running function is not affected), and write to Flash directly.
- When configuring, the process of normal programming and background programming is the same.

Here is an example of background programming.

Users generally write Function B to be updated to the Flash through the running Function A; then users can load the Function B in the Flash through a series of operations when the function needs to be updated.

Note!

- When Function A is operated according to the background programming, it must have a background programming module. In this manual, i.e. Function A must have Gowin goConfig UART IP.
- Function B can be operated without background programming module, but it can not perform background programming subsequently.
- Function A will still run normally during Write to Flash, but it will stop running during loading to SRAM.

2.2 UART Interface

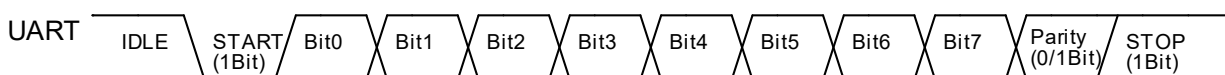
2.2.1 UART Features

The UART interface of Gowin goConfig UART IP consists of TX and RX. TX is used for data transmission by the IP, and RX is used for data reception by the IP.

The UART frame typically consists of: Start Bit, Data, Parity Bit, and Stop Bit. [Figure 2-1](#) shows the data frame utilized by the IP.

- **Start Bit (START):** 1 bit.
- **Data (Data):** 8 bits. For example, when transmitting 0x01, the bit transmission sequence is 1 > 0 > 0 > 0 > 0 > 0 > 0 > 0.
- **Parity Bit:** 0/1 bit. Configurable options include Odd Parity (1 bit), Even Parity (1 bit), and No Parity (0 bit).
- **Stop bit:** 1 bit.

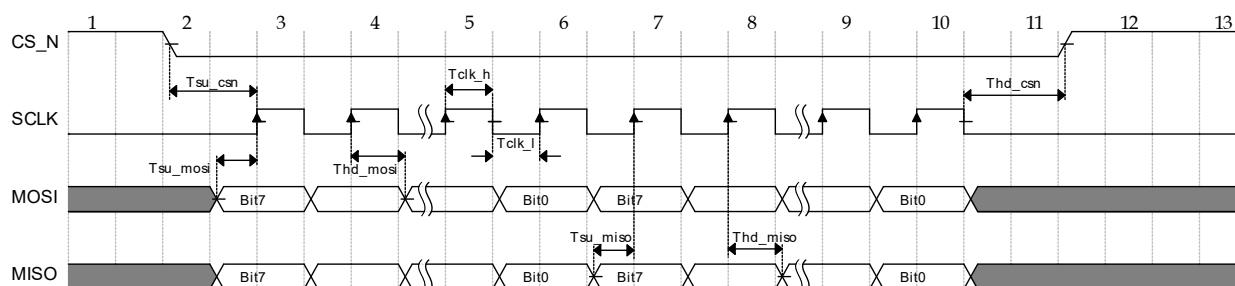
Figure 2-1 Example of UART Frame Composition



2.2.2 Flash Interface Features

The Flash interface of Gowin goConfig UART IP is SPI, using a four-wire mode: CS_N, SCLK, MOSI, MISO. SPI Data Transmission/Reception Method: The most significant bit (MSB) is transmitted first, and the transmission is generally done in byte (8-bit) unit. Taking the example of sending 0x01, the order is 0 > 0 > 0 > 0 > 0 > 0 > 0 > 1.

Figure 2-2 SPI Interface Timing



The SPI clock is 25 MHz (verified), and other details are as shown in [Table 2-2](#). The data in the table are theoretical values.

Table 2-2 SPI Interface Timing Description

Name	Description	Min	Max	Unit
T _{clk_l}	LOW period of the SCLK	20	-	ns
T _{clk_h}	HIGH period of the SCLK	20	-	ns
T _{hd_csn}	CS_N Setup Time	100	-	ns
T _{su_csn}	CS_N Setup Time	240	-	ns
T _{hd_mosi}	MOSI Hold Time	20	-	ns
T _{su_mosi}	MOSI Setup Time	20	-	ns
T _{hd_miso}	MISO Hold Time	20	-	ns
T _{su_miso}	MISO Setup Time	20	-	ns

2.3 IP Features

Gowin goConfig UART IP includes the following features:

- Supports 8 data bits and 1 stop bit.
- Supports odd parity, even parity, or no parity.
- Supports 110~1000000 baud rate.

2.4 Resource Utilization

Using Gowin GW2A-18 device as an instance, the resource utilization of Gowin goConfig UART IP with default configuration is as shown in [Table 2-3](#).

Table 2-3 Resource Utilization

Device	Programming Language	LUT	REG
GW2A-18	Verilog	774	513

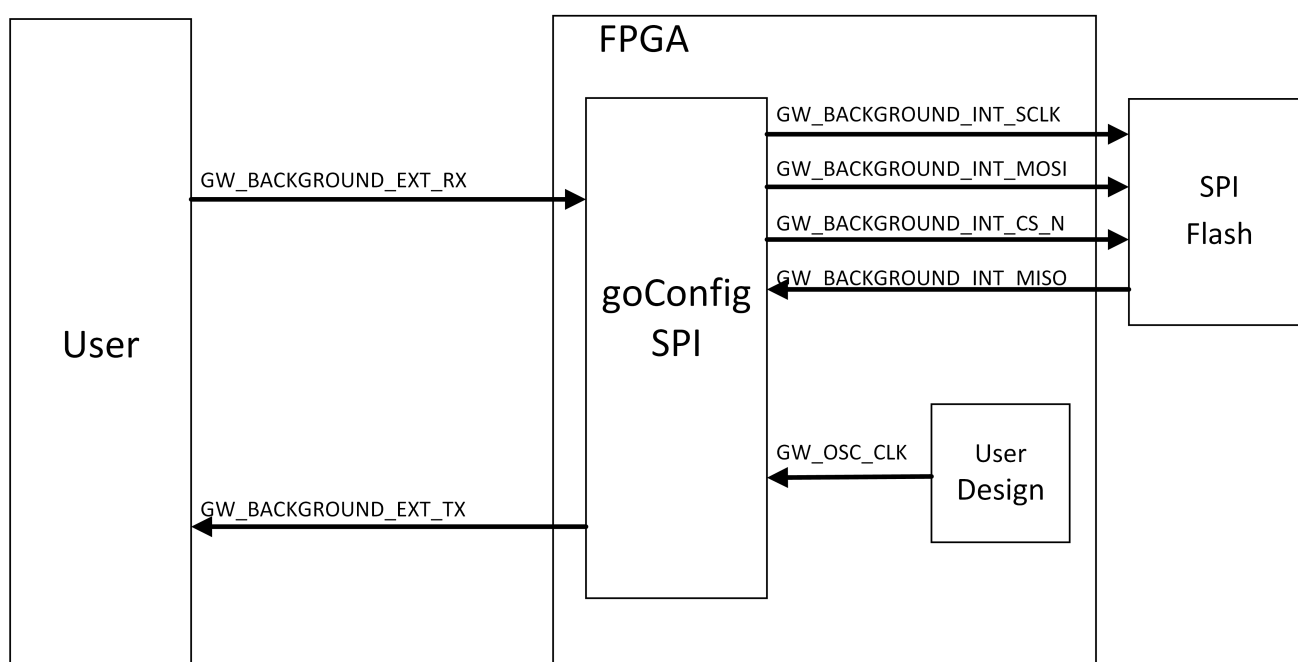
3 Functional Description

3.1 Gowin goConfig UART IP Structure Diagram

Gowin goConfig UART IP diagram includes User end and FPGA end. The User end is used to control the UART interface as well as other user functions. The FPGA end is connected to SPI Flash and is used to implement the functions that users required on FPGA. If only IP is considered, the User end controls the IP in the FPGA through the UART interface, which in turn controls the Flash to complete background programming.

The block diagram of Gowin goConfig UART IP is shown in [Figure 3-1](#). The UART interface is on the User side, and other signals are selected for control or use based on actual functional requirements.

Figure 3-1 Block Diagram



3.2 Gowin goConfig UART IP Control Commands

The commands supported by Gowin goConfig UART IP and their usage are described as follows.

Command Definition

The commands supported by Gowin goConfig UART IP are as shown in [Table 3-1](#).

Table 3-1 Command Definition

Name	Command Address	Description	Command Format
Write Data (Page)	0xABCDEF02	Write data (Page)	CMD + ADDR + Data
Read Data (Page)	0xABCDEF03	Read data (Page)	CMD + ADDR + Data
Status	0xABCDEF04	Read status	CMD + Data
Read Falsh ID	0xABCDEF05	Read Flash ID	CMD + Data
Erase(Sector)	0xABCDEF01	Erase (Sector)	CMD + ADDR
Reset	0xABCDEF55	Soft reset	CMD + Length + Data
Reconfign	0xABCDEF99	Reconfign command	CMD

The command is defined and described as follows:

- CMD: See [Table 3-1](#) for the command support.
 - Definition: Command address
 - Bit width: 32 bits
 - Transmission order: CMD[31:24] > CMD[23:16] > CMD[15:8] > CMD[7:0]
- ADDR:
 - Definition: Flash address
 - Bit width: 24 bits
 - Transmission order: ADDR[23:16] > ADDR[15:8] > ADDR[7:0]
- Data:
 - Definition: Valid data
 - Width: 8 bits per data unit
 - Length: The length of valid data varies depending on the command. Generally, Data_0 refers to the first valid data, Data_1 refers to the second valid data, and so on, with Data_N representing the last valid data.

Note!

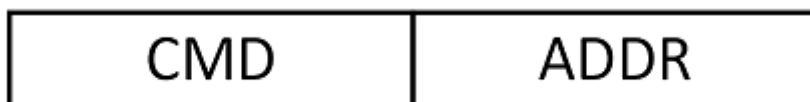
A sector consists of 4K bytes, and a page consists of 256 bytes. Therefore, 1 Sector = 16 Pages.

Erase(0xABCDEF01)

Erase (Sector) Command: 0xABCDEF01; Format: CMD + ADDR; The command diagram is as shown in [Figure 3-2](#).

Here, CMD is 32 bits, and the transmission sequence is 0xAB > 0xCD > 0xEF > 0x01. ADDR is 24 bits, taking ADDR=0x010203 as an example, the transmission sequence is 0x01 > 0x02 > 0x03.

Figure 3-2 Sector Erase Command



Note!

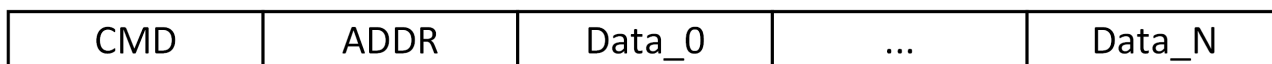
- In the figure, the valid data with "white shading" is sent from "User" to the "IP-UART interface".
- In the following text, the valid data with "gray shading" is sent from the "IP-UART interface" to "User".
- In the following text, unless otherwise specified, the names in similar figures have the same meanings as in this example.
- The names that may be used in the figure are defined as follows.
 - CMD: See [Table 3-1](#) for the command support.
 - Data_0: First valid data
 - Data_N: Last valid data
 - ADDR: Flash address

Write Data (0xABCDEF02)

Write Data (Page) Command: 0xABCDEF02; Format: CMD + ADDR + Data; The command diagram is as shown in [Figure 3-3](#).

Here, CMD is 32 bits, and the transmission sequence is 0xAB > 0xCD > 0xEF > 0x02. ADDR is 24 bits, taking ADDR=0x010203 as an example, the transmission sequence is 0x01 > 0x02 > 0x03. The valid length of Data is 256.

Figure 3-3 Page Program Command



Read Data(0xABCDEF03)

Read Data (Page) Command: 0xABCDEF03; Format: CMD + ADDR + Data; The command figure is as shown in [Figure 3-4](#).

Here, CMD is 32 bits, and the transmission sequence is 0xAB > 0xCD > 0xEF > 0x03. ADDR is 24 bits, taking ADDR=0x010203 as an example, the transmission sequence is 0x01 > 0x02 > 0x03. The valid length of Data is 256.

Figure 3-4 Page Read Command

CMD	ADDR	Data_0	...	Data_N
-----	------	--------	-----	--------

Status(0xABCDEF04)

Status Command: 0xABCDEF04; Format: CMD + Data; The command figure is as shown in [Figure 3-5](#).

Here, CMD is 32 bits, and the transmission sequence is 0xAB > 0xCD > 0xEF > 0x04. The valid length of Data is 3, with two bytes being fixed values. Data = 0xAA + 0x55 + ST_data, where ST_data reflects the status of the IP.

Figure 3-5 Status Command

CMD	0xAA	0x55	ST_data
-----	------	------	---------

The bit mapping table for ST_data is as shown in [Table 3-2](#).

Table 3-2 Status Bit Table

Bit(s)	Default	Description
Bit[6]	0	Reserved
Bit[5]	0	Page Program timeout
Bit[4]	0	Page Program operation completed
Bit[7]	0	Flash busy
Bit [3:2]	00	Reserved
Bit[1]	0	Erase timeout
Bit[0]	0	Erase operation completed

The typical values of ST_data are as shown [Table 3-3](#).

Table 3-3 Typical Value of ST_data

Typ.	Description
0x01	Erase operation completed
0x03	Erase timeout
0x10	Page Program completed
0x00	Normal
0x30	Page Program timeout
0x80	Flash operating

Note!

The status value only reflects the status of the previous operation. For example, after an erase operation is completed, ST_data = 0x01. If a Page Program operation is executed immediately after an erase operation, ST_data = 0x10, meaning the erase status is not reflected, and only the Page Program status is reflected.

Read Flash ID(0xABCDEF05)

Read Flash ID Command: 0xABCDEF05; Format: CMD + Data; The command figure is as shown in [Figure 3-6](#).

Here, CMD is 32 bits, and the transmission sequence is 0xAB > 0xCD > 0xEF > 0x05. The valid data length is 3. For example, if Data = 0x010203, the received data order is 0x01 > 0x02 > 0x03.

Figure 3-6 Read Flash ID Command Example

**Reset(0xABCDEF55)**

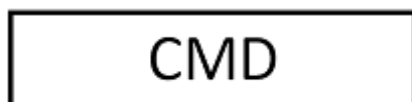
Reset IP Command: 0xABCDEF55. It can reset internal status of the IP.

Here, CMD is 32 bits, and the transmission sequence is 0xAB > 0xCD > 0xEF > 0x55.

Note!

This process only involves the IP, not the background programming.

Figure 3-7 Reset IP Command Example

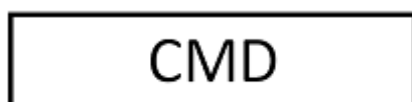
**Reconfig(0xABCDEF99)**

Reconfig Command: 0xABCDEF99

Here, CMD is 32 bits, and the transmission sequence is 0xAB > 0xCD > 0xEF > 0x99.

It can be used to pull down the "GW_BACKGROUND_RECONFIG_N" pin, which can be mapped to the device's "RECONFIG_N" pin to achieve the function of pulling down "RECONFIG_N".

Figure 3-8 Reconfig Command Example



3.3 Flash Interface Operations

The Flash interface of the IP is an SPI interface. This chapter will describe the operations performed by the Flash interface after the UART receives and executes relevant commands. For the basic features of the SPI interface, see [2.2.2 Flash Interface Features](#). Based on the commands supported by the IP and the selected Flash, it can be determined whether the IP can control the Flash.

3.3.1 SPI Interface Commands

This section describes the commands used in the SPI interface. Read Status Register-1 is abbreviated as RD_ST1.

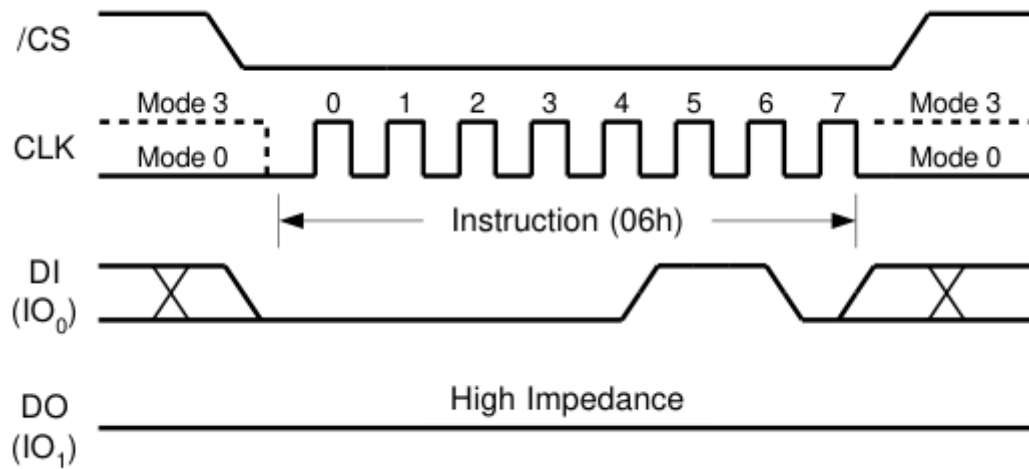
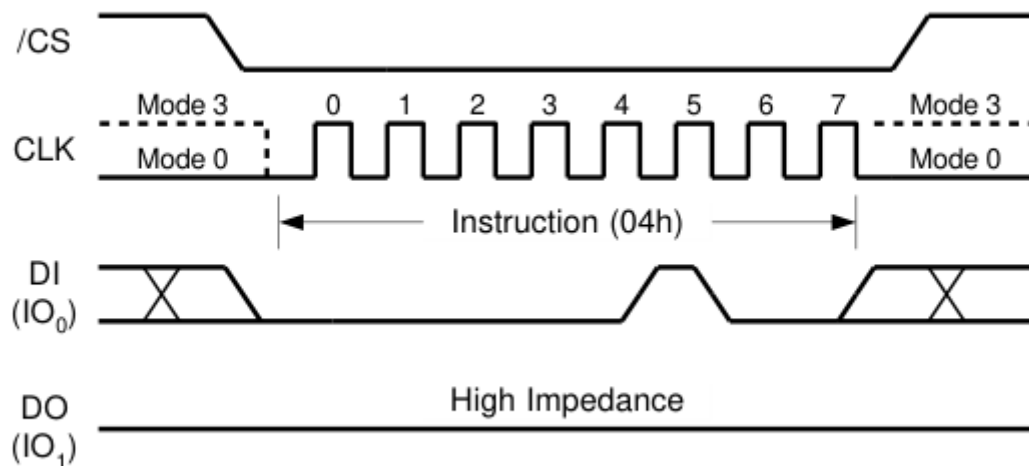
Table 3-4 SPI Interface Commands

Command	Byte1	Byte2	Byte3	Byte4	Byte5
Write Enable	06H	-			
Write Disable	04H				
Read Status Register-1	05H	(S7-S0)	-		
Sector Erase	20H	A23-A16	A15-A8	A7-A0	-
Page Program	02H	A23-A16	A15-A8	A7-A0	D7-D0
Read JEDEC ID	9FH	(MF7-MF0)	(ID15-ID8)	(ID7-ID0)	-
Read Data	03H	A23-A16	A15-A8	A7-A0	D7-D0

Write Enable (06H)

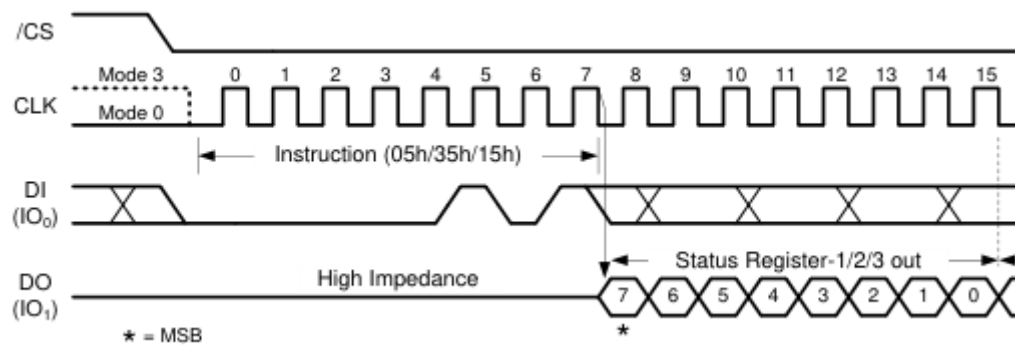
Note!

Unless otherwise specified, the timing diagram for the SPI used by the IP follows Mode 0.

Figure 3-9 Write Enable (06H)**Write Disable (04H)****Figure 3-10 Write Disable (04H)****Read Status Register-1 (05H)**

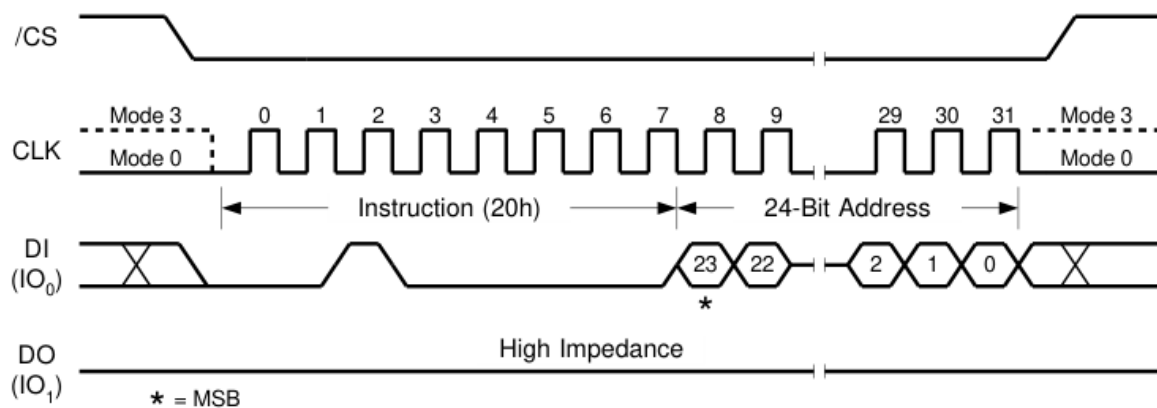
This register is abbreviated as RD_ST1.

Figure 3-11 Read Status Register-1 (05H)



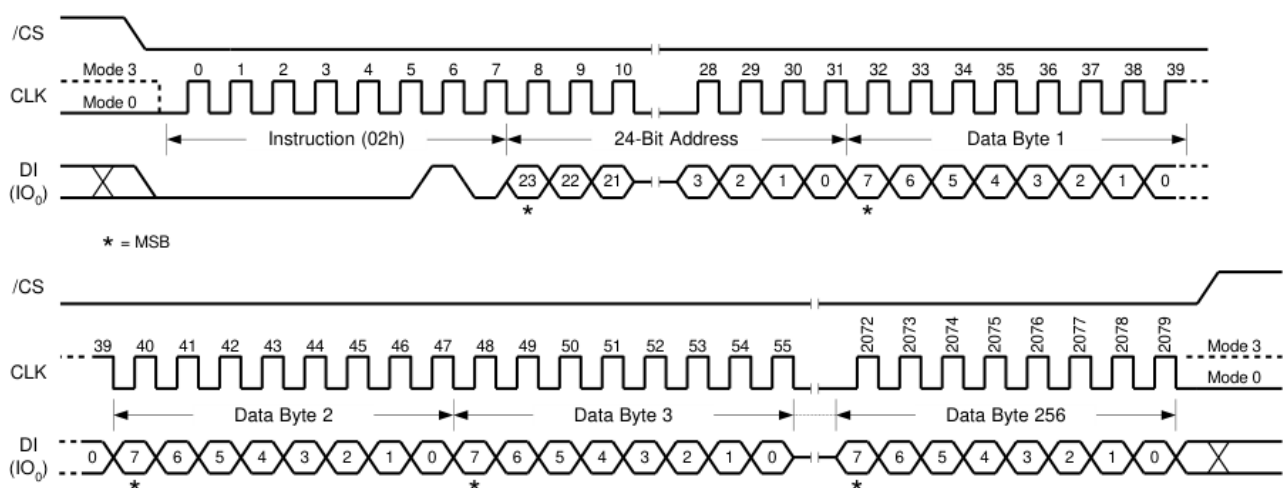
Sector Erase (20H)

Figure 3-12 Sector Erase (20H)



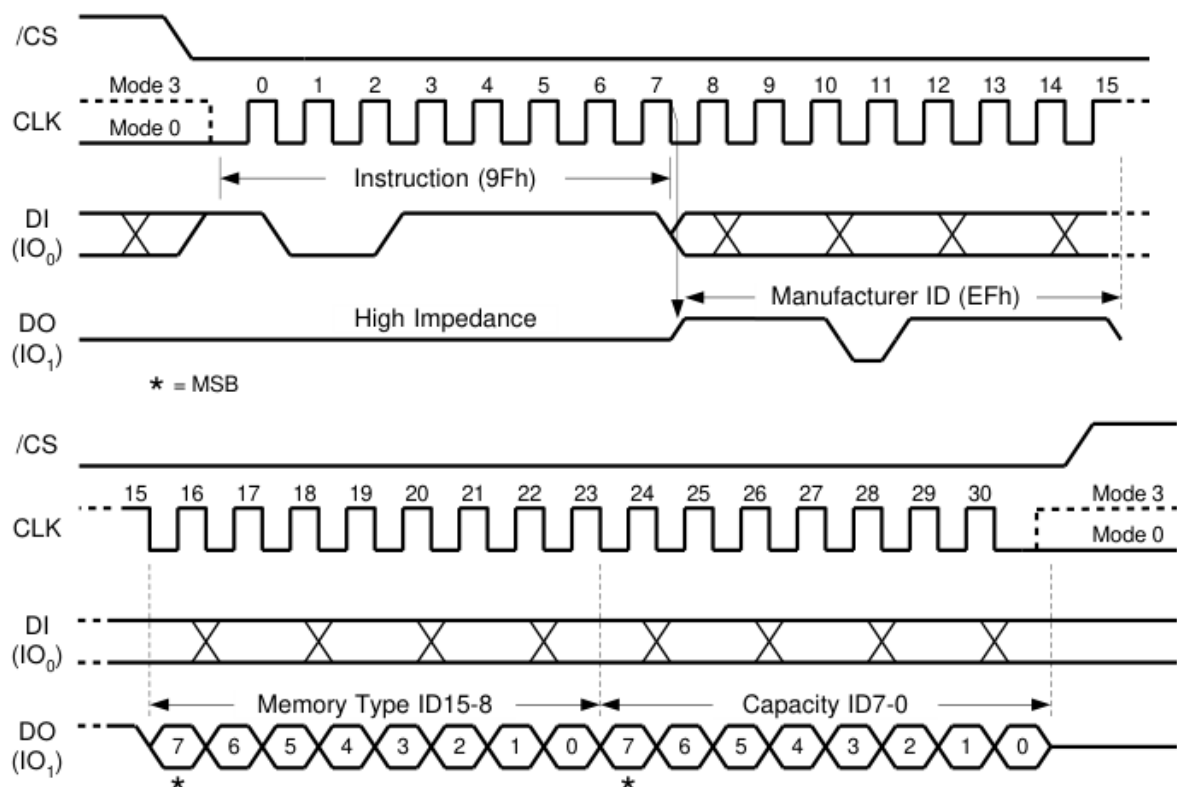
Page Program (02H)

Figure 3-13 Page Program (02H)



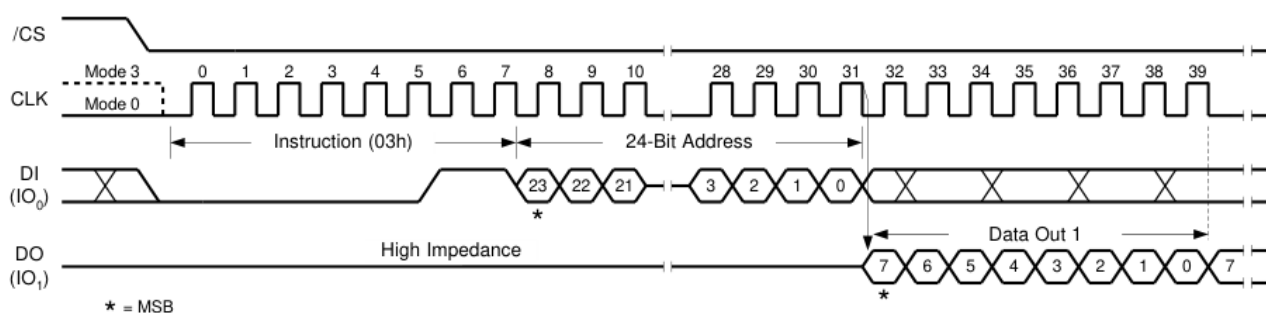
Read JEDEC ID (9FH)

Figure 3-14 Read JEDEC ID (9FH)



Read Data (03H)

Figure 3-15 Read Data (03H)



3.4 UART Command and SPI Interface Mapping

This section describes the mapping relationship between UART commands and the SPI interface, specifically focusing on UART commands that can be mapped to SPI commands. For convenience, command names will be used to replace their respective detailed timings.

Erase(0xABCDEF01)

When the UART interface detects the "Erase" command, the SPI interface will execute the following sequence: Write Enable > Sector Erase > RD_ST1 > Write Disable.

The RD_ST1 operation is as follows:

1. Check for timeout (greater than 200ms)
 - a. If timeout, proceed to 5.
 - b. If no timeout, proceed to 2.
2. Wait for 4ms.
3. Execute the RD_ST1 command.
4. Check the return value (whether it is 0x00).
 - a. If it equals 0x00, proceed to 5.
 - b. If not 0x00, proceed to 1.
5. The next operation is "Write Disable".

Figure 3-16 Erase Command Mapping

UART:

Erase(Sector)

SPI:

Write Enable	Sector Erase	RD_ST1	Write Disable
--------------	--------------	--------	---------------

Write Data (0xABCDEF02)

When the UART interface detects the "Write Data" command, the SPI interface will execute the following sequence: Write Enable > Page Program > RD_ST1 > Write Disable.

The RD_ST1 operation is as follows:

1. Check for timeout (greater than 16ms)
 - a. If timeout, proceed to 5.
 - b. If no timeout, proceed to 2.
2. Wait for 4ms.
3. Execute the RD_ST1 command.
4. Check the return value (whether it is 0x00).
 - a. If it equals 0x00, proceed to 5.
 - b. If not 0x00, proceed to 1.
5. The next operation is "Write Disable".

Figure 3-17 Write Data Command Mapping

UART:

Write Data(Page)

SPI:

Write Enable

Page Program

RD_ST1

Write Disable

Read Data(0xABCDEF03)

When the UART interface detects the "Read Data" command, the SPI interface will continuously read 256 bytes (1 page) of data.

Figure 3-18 Read Data Command Mapping

UART:

Read Data(Page)

SPI:

Read Data

Read Flash ID(0xABCDEF05)

When the UART interface detects the "Read Flash ID" command, the SPI interface will execute the "Read JEDEC ID" command.

Figure 3-19 Read Flash ID Command Mapping

UART:

Read Flash ID

SPI:

Read JEDEC ID

4 Signal Definition

The IO definitions of Gowin goConfig UART IP are shown in [Table 4-1](#).

Table 4-1 IO Definitions

Signal Name	I/O	Description	I/O Con- straints	Note
goConfig general interface				The I/O of all the signals takes IP as reference
GW_OSC_CLK	Input	Clock input (25 MHz)	-	
GW_RSTN	Input	Reset, active-low.		
goConfig Specific Interface				
GW_BACKGROUND_RE- CONFIG_N	Output	It can be combined with Reconfig command to pull down a certain signal.		
goConfig External Interface				
GW_BACK- GROUND_EXT_RX	Input	UART-RX	-	
GW_BACK- GROUND_EXT_TX	Output	UART-TX	-	
goConfig Flash Interface				
GW_BACK- GROUND_INT_SCLK	Output	SPI-SCLK	MSPI	
GW_BACK- GROUND_INT_CS_N	Output	SPI-CS_N	MSPI	
GW_BACK- GROUND_INT_MOSI	Output	SPI-MOSI	MSPI	
GW_BACK- GROUND_INT_MISO	Input	SPI-MISO	MSPI	

Note!

- The I/O constraints are defined as follows:

- "-": Unlimited, usage for general IP interface Names can be freely assigned, and locations can be freely constrained.
- "MSPI": Constrains IOs to the corresponding MSPI pins on the device.
- Note that the MSPI pins on some FPGAs may not be exactly 4. You can see the FPGA Pinout Manual and the Flash datasheet for proper mapping.
- "GW_OSC_CLK" usage description: Input stable 25MHz clock.
- "GW_BACKGROUND_RECONFIG_N" usage description:
 - Typically used in conjunction with "Reconfig" command, and this pin is constrained to the device-specific "RECONFIG_N" pin to trigger the "RECONFIG_N" function.
- The IO interfaces are classified and defined as follows:
 - "goConfig General Interface": The IOs that are general in other IPs as well, for example, Clock, Reset etc.
 - "goConfig Specific Interface": IOs possibly included in the background programming IP. The functions of these IOs are often related to the overall process of background programming or some specific functions.
 - "goConfig External Interface": The IOs primarily for communications in background programming IP. These IO pin definitions and functions adhere to established standards such as I2C, SPI, JTAG, etc.
 - "goConfig Flash Interface": The IO connected with Flash in the background programming IP.

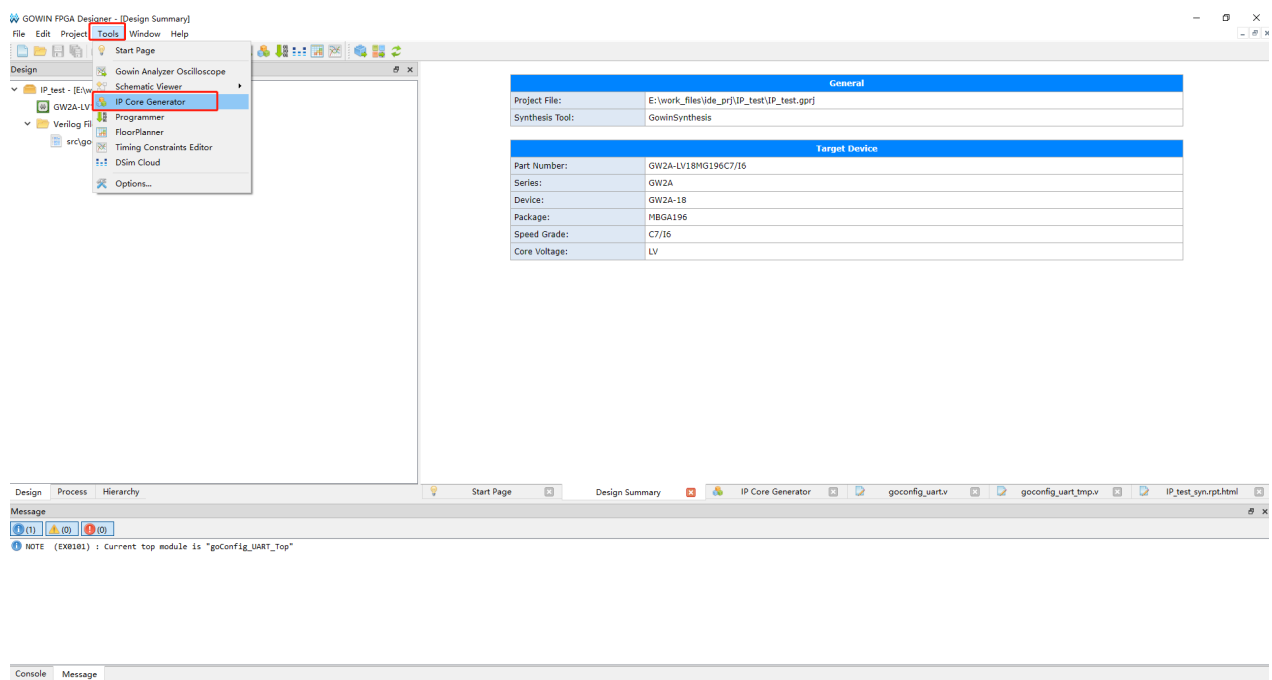
5 Interface Configuration

You can click “Tools > IP Core Generator > goConfig UART” to call and configure Gowin goConfig UART IP in Gowin Software.

1. Open IP Core Generator.

After creating the project, you can click the “Tools” tab in the upper-left corner, then click the “IP Core Generator” via the drop-down list, as shown in [Figure 5-1](#).

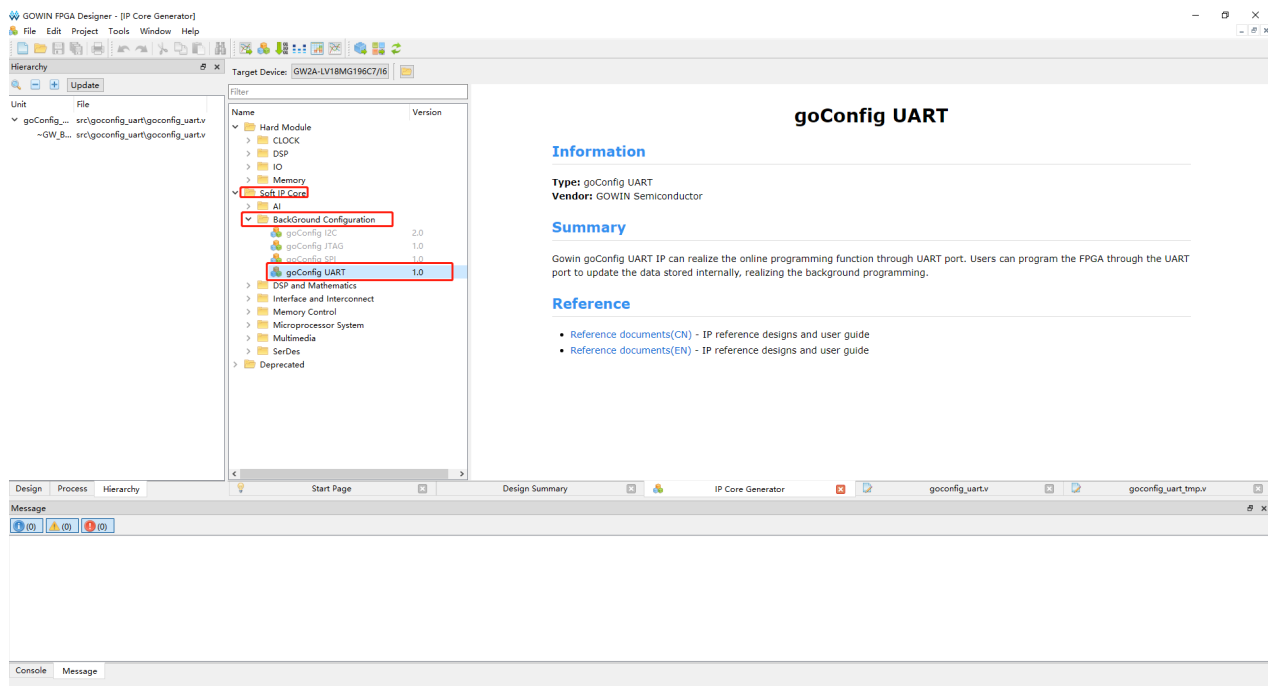
Figure 5-1 IP Core Generator



2. Open Gowin goConfig UART IP core.

Select “Soft IP Core> BackGround Configuration > goConfig UART”, as shown in [Figure 5-2](#). Double click to open the configuration interface.

Figure 5-2 Open Gowin goConfig UART IP Core



3. Gowin goConfig UART IP core configuration interface.
 The configuration interface of Gowin goConfig UART IP is as shown in [Figure 5-3](#). The port diagram is displayed on the left, and options are displayed on the right.
 - You can configure the generated file name in “File Name” text box.
 - You can configure the generated top module name in “Module Name” text box.
 - You can configure slave address by modifying SPI Slave Addr.

Figure 5-3 Gowin goConfig UART IP Core Configuration Interface

goConfig UART

General

Device: GW5AT-60 Device Version: B

Part Number: GW5AT-LV60GW369ES Language: Verilog

File Name: goconfig_uart Module Name: goConfig_UART_Top

Create In: E:\work_files\ide_prj\IP_test\src\goconfig_uart

Options

System Clock: 25.000MHz

Idle Cycle: 100

UART Polarity: Odd

UART Baud Rate

☒ Standard 115200

Custom: 115200

OK Cancel

Table 5-1 Parameter Configuration

Option	Description	Note
System clock	System clock	25 MHz is recommended; the clock setting should match the actual application.
Idle Cycle	UART idle cycle Range: 2~65535	Refers to system clock cycles; for example, at 25 MHz, one cycle is 40 ns, and the default configuration is 100 * 40 ns.
UART Polarity	Parity setting: None, Odd, Even	-
UART Baud Rate	Baud rate setting Range: 110~1000000	

6 Reference Design

This chapter is intended to introduce the usage and structure of Gowin goConfig UART IP [reference design](#). Please see the reference design for details at Gowinsemi website.

6.1 Project Setup Procedure

When users build an IP project with background programming function, or insert the IP with background programming into an existing project, the following steps can be followed:

- Configure and generate IP: Gowin goConfig UART IP
- Call the IP at the Top Level
- Software configuration: Check "Use MSPI as regular IO" in "Dual-Purpose Pin".

The preliminary setting up of the project, as well as other operations after completing the above steps, are consistent with the operations of normal project, so it will not be detailed in this document.

If issues arise during the IP project building, you can also follow this process to troubleshoot and determine if it is an IP using issue.

6.1.1 Configure and Generate IP

Select Gowin goConfig UART IP for the background programming, and configure it for generation. For more detailed information, see [5 Interface Configuration](#).

6.1.2 Call the IP at the Top Level

In the top level of the project, when invoking the IP, there are generally no special requirements. However, when constraining the IO positions, the Flash interface of the IP needs to correspond to the MSPI ports. For an example of calling the IP, you can see the "Top.v" file in [6.2 Reference Design](#). For more detailed information, see [4 Signal Definition](#).

IO Invocation Example

Figure 6-1 shows an example of invoking Gowin goConfig UART IP using the "Top.v" file in the reference design.

Figure 6-1 Example of Invoking IP IO Names

```
//*****
// IP
//*****
goConfig_UART_Top inst_uart(
// SPI Flash
    .GW_BACKGROUND_INT_SCLK(spi_sclk),
    .GW_BACKGROUND_INT_CS_N(spi_cs_n),
    .GW_BACKGROUND_INT_MOSI(spi_mosi),
    .GW_BACKGROUND_INT_MISO(spi_miso),
// UART
    .GW_BACKGROUND_EXT_RX(uart_rx),
    .GW_BACKGROUND_EXT_TX(uart_tx),
//
    .GW_BACKGROUND_RECONFIG_N(reconfig_n),
// General IO
    .GW_RSTN(rstn),
    .GW_OSC_CLK(clk_25M)
);
```

Figure 6-2 shows the IO definitions when Gowin goConfig UART IP is used correctly in the "Top.v".

Figure 6-2 Example of Top-level IO Definitions

```

1 module top
2 #(
3   // uart
4   parameter    UART_SLAVE_SYSCLK    = 25_000_000    //采样频率
5 )
6 (
7   // uart
8   input    uart_rx,
9   output   uart_tx,
10  // SPI
11  output    spi_sclk,
12  output    spi_cs_n,
13  input     spi_miso,
14  output    spi_mosi,
15
16  output    reconfig_n,
17  //test_IO
18  //    output    spi_cs_n_test,
19  //    output    spi_sclk_test,
20  //    output    spi_mosi_test,
21  //    output    spi_miso_test,
22  output    uart_rx_test,
23  output    uart_tx_test,
24
25  output    test,
26  //others
27  input     clk_in,
28  input     rstn
29 );
30

```

6.1.3 Software Configuration

You can select “Project > Configuration > Place & Route > Dual-Purpose Pin”, and then check “Use MSPI as regular IO”.

There are two methods to start "Configuration" as shown in [Figure 6-3](#) and [Figure 6-4](#).

Figure 6-3 Example One of Starting Configuration

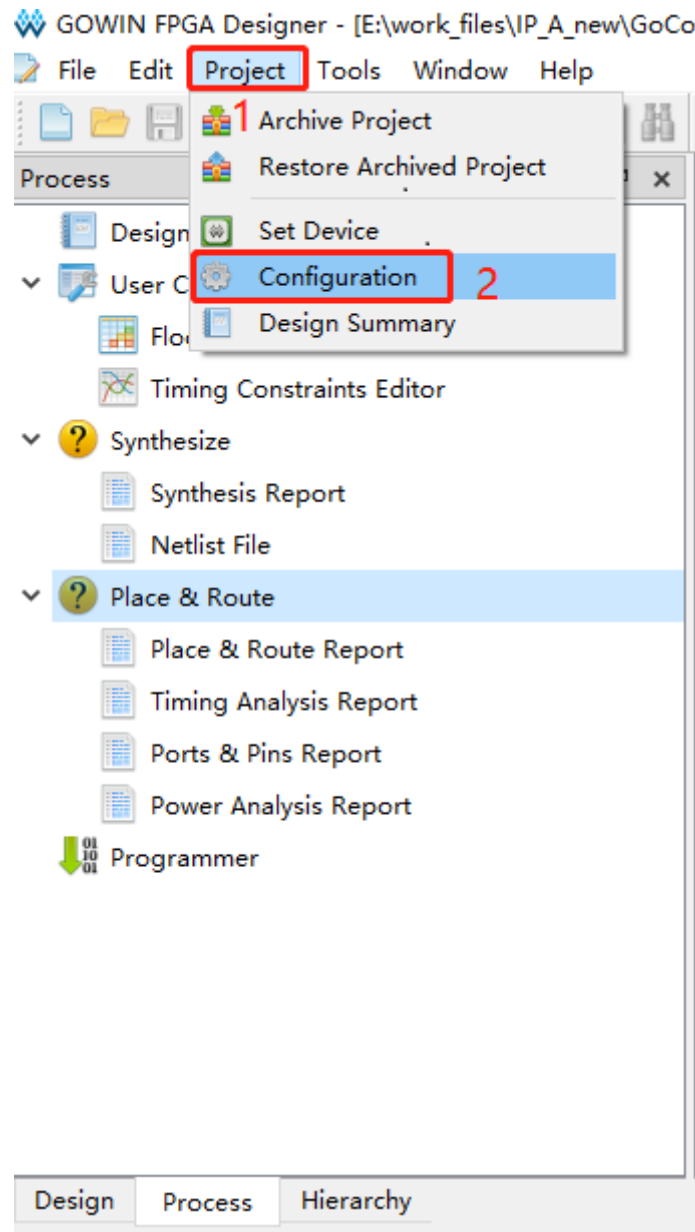


Figure 6-4 Example Two of Starting Configuration

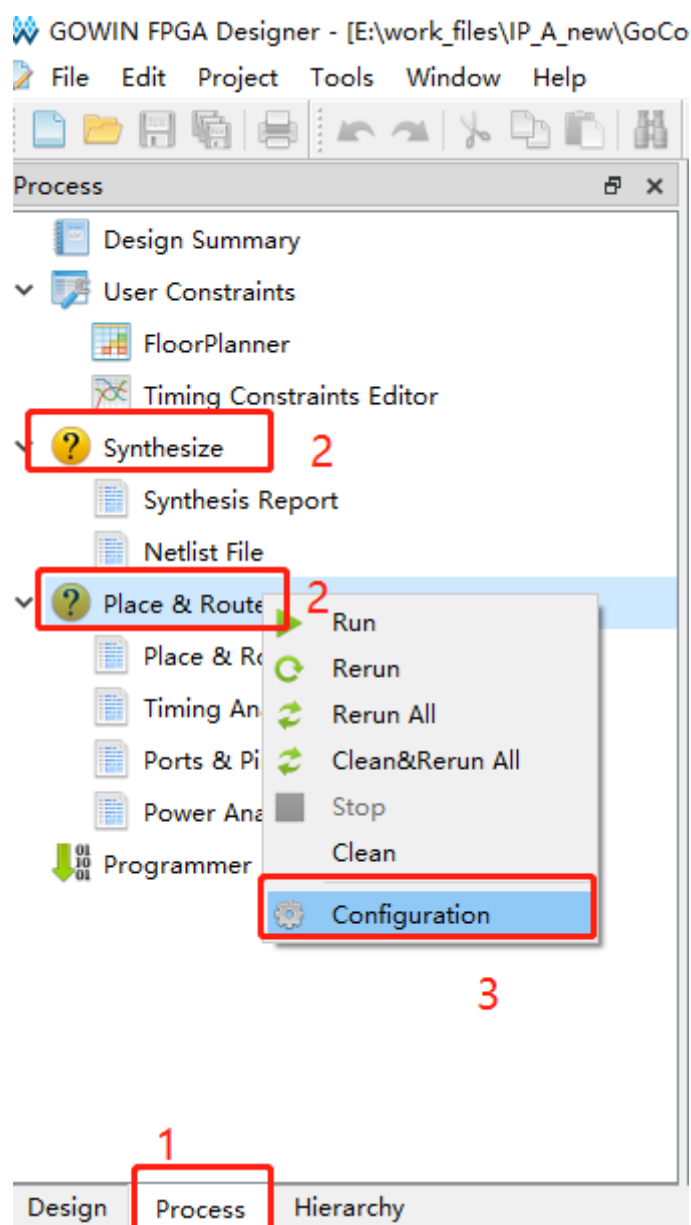
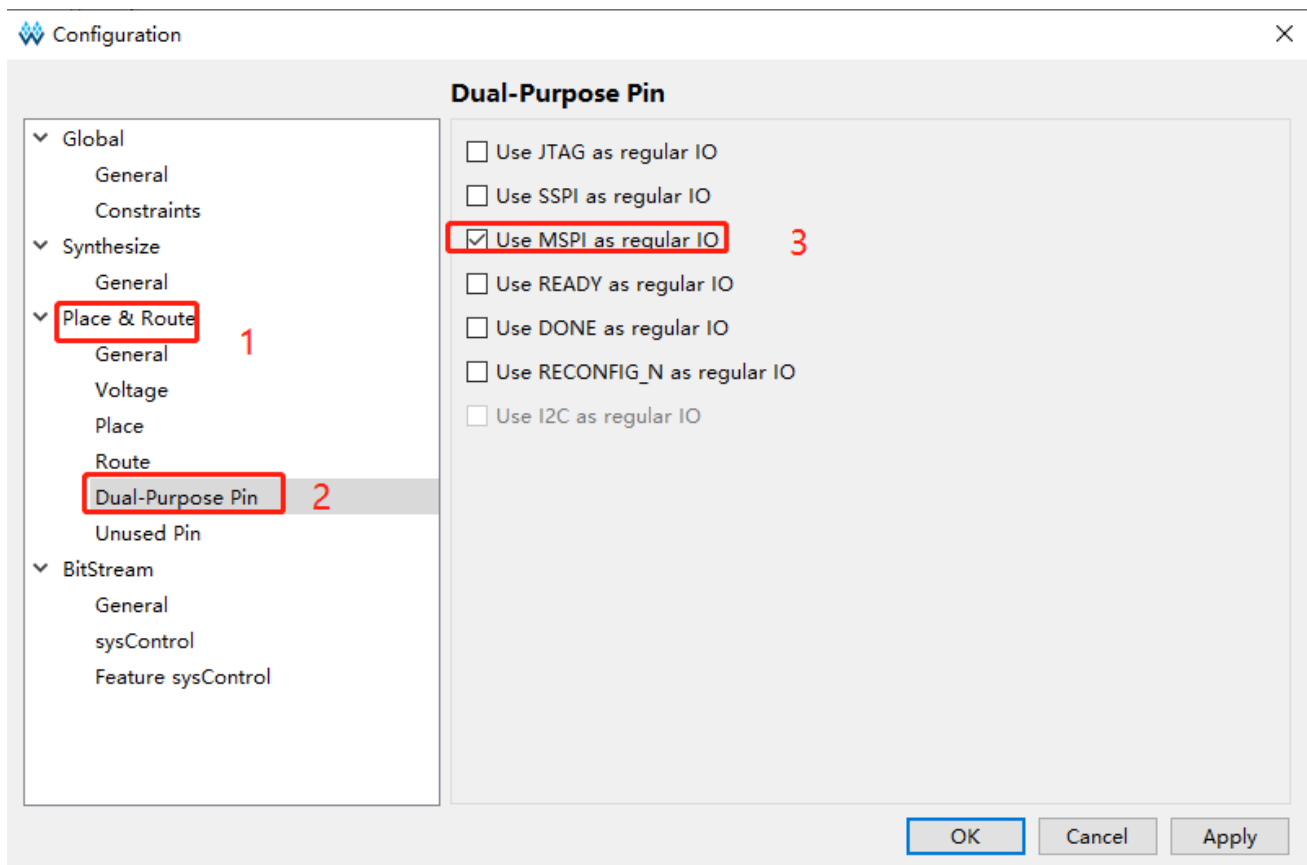


Figure 6-5 shows an example of checking "Use MSPI as regular IO". You can click "Place & Route > Dual-Purpose Pin", and then check "Use MSPI as regular IO".

Figure 6-5 Check "Use MSPI as regular IO"



When the following error occurs, check whether "Use MSPI as regular IO" is selected.

Figure 6-6 Error Example

```

reading constraint file: E:\work_files\IP_A_new\GoConfig\00_2024\uart_spi\pro\12-11\ref_design\fpga_project\src\fpga_project.cst
WARN (CT1122) : 'reconfig_n' will be placed to RECONFIG pin according to constraint, and then the signal may lead to gwCONFIG programming d
Physical Constraint parsed completed
Running placement.....
ERROR (PR2028) : The constrained location is useless in current package
ERROR (PR2017) : 'spi_cs_n' cannot be placed according to constraint, for the location is a dedicated pin (MSPI)
ERROR (PR2028) : The constrained location is useless in current package
ERROR (PR2017) : 'spi_sclk' cannot be placed according to constraint, for the location is a dedicated pin (MSPI)
ERROR (PR2028) : The constrained location is useless in current package
ERROR (PR2017) : 'spi_mosi' cannot be placed according to constraint, for the location is a dedicated pin (MSPI)
ERROR (PR2028) : The constrained location is useless in current package
ERROR (PR2017) : 'spi_miso' cannot be placed according to constraint, for the location is a dedicated pin (MSPI)
Generate file "E:\work_files\IP_A_new\GoConfig\00_2024\uart_spi\pro\12-11\ref_design\fpga_project\impl\pnr\fpga_project.pin.html" completed

```

6.2 Reference Design

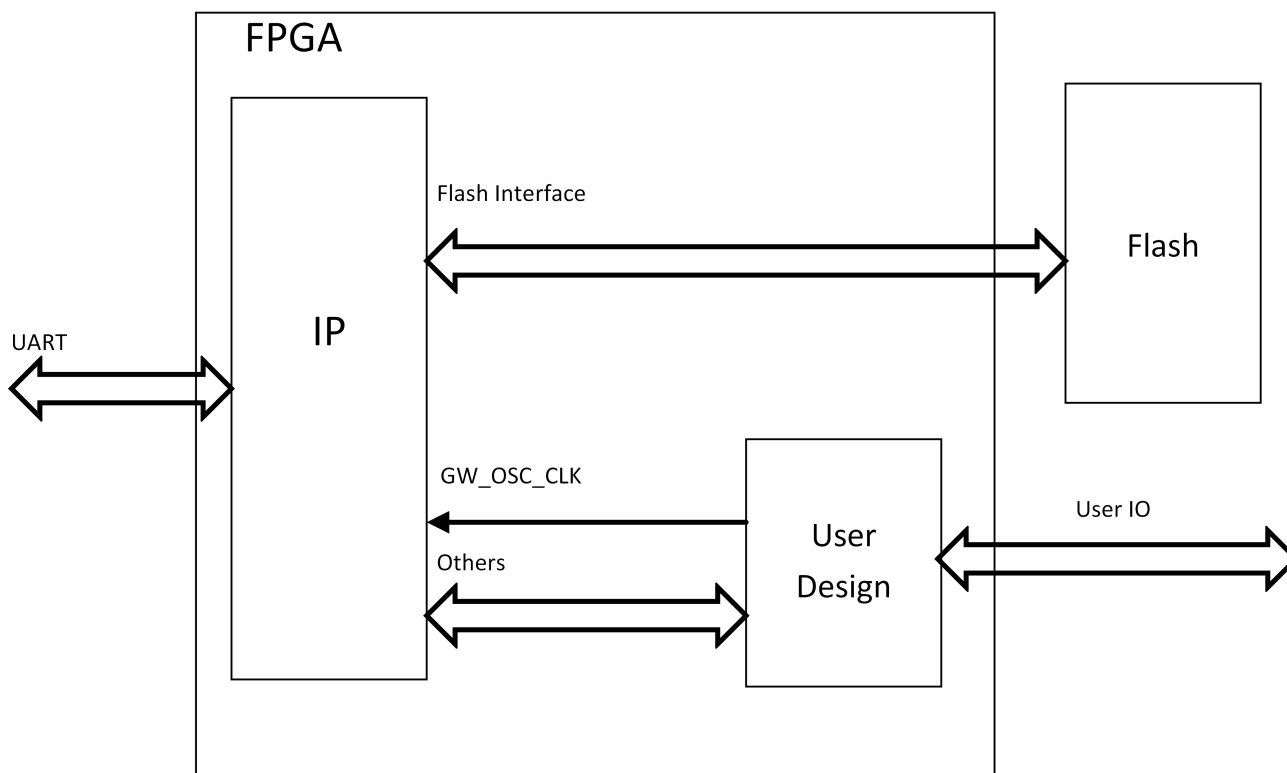
6.2.1 FPGA Project Description

The FPGA reference design for Gowin goConfig UART IP consists of two components: the IP module and the user module.

- IP module: Used to call IP and provide an instance. Module descriptions are as follows:
 - You need modify Gowin Software configuration to use this IP, and there is an example in the reference design; see [6.1 Project Setup Procedure](#) for detailed descriptions.
 - There are some I/O constraints for the IP, you can follow the reference design for implementation; for detailed explanations, see [4 Signal Definition](#).

- User module: A simple LED test is used to represent the user design.

Figure 6-7 Reference Design Block Diagram



6.2.2 Reference Design Files

The descriptions of the files for the reference design of Gowin goConfig UART IP are shown in [Table 6-1](#).

Table 6-1 src Folder List in Reference Design

Name	Description
fpga_project.cst	Project physical constraints file
goconfig_uart	IP folder
----goconfig_uart.v	Generated IP file, encrypted
----goconfig_uart.vo	Generated IP netlist file for simulation
gowin_rpll	gowin_rpll folder
---- gowin_rpll.v	gowin_rpll module, used to generate a 25 MHz clock.
Top.v	Top-level module, including LED test.

6.2.3 Reference Design Considerations

This reference design of Gowin goConfig UART IP contains only FPGA project and some documentation; the simulation environment and the UART driver example are not provided. Please contact Gowin FAE for UART bus driver example.

