



Gowin USB to Multi Serial Protocol Bridge IP **User Guide**

IPUG1180-1.0E, 12/29/2023

Copyright © 2023 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN is a trademark of Guangdong Gowin Semiconductor Corporation and is registered in China, the U.S. Patent and Trademark Office, and other countries. 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 denotes, 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
12/29/2023	1.0E	Initial version published.

Contents

Contents	i
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 Overview	3
2.1 Introduction	3
2.2 Features	3
2.3 Operating Frequency	4
2.4 Resource Utilization	4
3 Functional Description	5
3.1 System Block Diagram	5
3.2 Introduction to USB to Multi Serial Protocol Bridge Module	5
3.2.1 USB High-speed Device (480Mbps)	5
3.2.2 Endpoint Count	5
3.2.3 Interface Endpoint Address	6
3.2.4 Interface Implementation Reference	6
3.3 Interface Parameters	6
3.4 Requests Related to UART Virtual Serial Port	6
3.5 I2C Complete Data Packet Format	7
3.6 I2C Register Configuration	7
3.6.1 Interrupt Enable Register(0x5)	8
3.6.2 Status Register (0x6)	8
3.6.3 Address Register (0x7)	10
3.6.4 Data Register (0x8)	10
3.6.5 Control Register (0x9)	11
3.6.6 Command Register (0xA)	11
3.6.7 Setup Register (0xB)	12

3.6.8 I2C Timing Parameter Multiple Register (0xC)	13
3.7 SPI Complete Data Packet Format.....	13
3.8 SPI Register Configuration.....	13
3.8.1 Transfer Format Register (0x04).....	14
3.8.2 Transfer Control Register (0x08).....	15
3.8.3 Command Register (0x09).....	16
3.8.4 Address Register (0x0A)	16
3.8.5 Data Register (0x0B).....	16
3.8.1 Control Register (0x0C)	17
3.8.2 Status Register (0x0D).....	17
3.8.3 Interrupt Enable Register(0x0E)	18
3.8.4 SPI Interrupt Status Register (0x0F).....	19
3.8.5 SPI Interface Timing Register (0x10)	19
3.8.6 Configuration Register (0x1F).....	20
3.9 JTAG Register Configuration	20
3.9.1 Configuration Register Command Frame Format.....	20
3.9.2 Register Information.....	21
4 Port Description.....	22
4.1 Port List	22
4.2 Parameter Configuration	24
5 Interface Configuration	25
6 Reference Design	28
7 File Delivery	29
7.1 Documentation	29
7.2 Design Source Code (Encryption)	29

List of Figures

Figure 3-1 System Diagram	5
Figure 4-1 Gowin USB to Multi Serial Protocol Bridge IP Port Diagram.....	22
Figure 5-1 Open IP Core Generator	25
Figure 5-2 Open USB to Multi Serial Protocol Bridge IP Configuration Interface.....	26
Figure 5-3 USB to Multi Serial Protocol Bridge Configuration Interface	27
Figure 6-1 Block Diagram of Reference Design	28

List of Tables

Table 1-1 Terminology and Abbreviations	2
Table 2-1 Gowin USB to Multi Serial Protocol Bridge IP	3
Table 2-2 Resource Utilization	4
Table 3-1 Device ID	5
Table 3-2 Endpoint Count	5
Table 3-3 Interface Endpoint Address	6
Table 3-4 GET LINE CODING Request Encoding Format	6
Table 3-5 GET LINE CODING Return Parameter.....	6
Table 3-6 SET LINE CODING Request Encoding Format.....	6
Table 3-7 I2C Data Packet Format.....	7
Table 3-8 I2C Command Set.....	7
Table 3-9 Gowin I2C Registers	7
Table 3-10 Interrupt Enable Register	8
Table 3-11 Status Register	8
Table 3-12 Transmit Register	10
Table 3-13 Transmit Register	10
Table 3-14 Status Register.....	11
Table 3-15 Command Register	11
Table 3-16 Setup Register	12
Table 3-17 I2C Timing Parameter Multiple Register	13
Table 3-18 SPI Data Packet Format	13
Table 3-19 SPI Command Set	13
Table 3-20 SPI Register	14
Table 3-21 Transfer Format Register	14
Table 3-22 Transfer Control Register.....	15
Table 3-23 Command Register	16
Table 3-24 Address Register.....	16
Table 3-25 Data Register	17
Table 3-26 Control Register	17
Table 3-27 Status Register.....	17
Table 3-28 Interrupt Enable Register	18
Table 3-29 SPI Interrupt Status Register	19

Table 3-30 SPI Interface Timing Register	19
Table 3-31 Configuration Register	20
Table 3-32 Set Registers.....	20
Table 3-33 Read Registers	20
Table 3-34 Response Frame.....	21
Table 3-35 Register Information.....	21
Table 4-1 Gowin USB to Multi Serial Protocol Bridge IP Port List	23
Table 4-2 Gowin USB to Multi Serial Protocol Bridge IP Parameter Configuration	24
Table 7-1 Documents List	29
Table 7-2 Design Source Code List of USB to Multi Serial Protocol Bridge	29

1 About This Guide

1.1 Purpose

The purpose of Gowin USB to Multi Serial Protocol Bridge IP is to help you learn the features and usage of Gowin USB to Multi Serial Protocol Bridge IP by providing the descriptions of features, functions, ports, GUI and reference design, etc. The software screenshots and the supported products listed in this manual are based on Gowin Software V1.9.9 (64-bit). 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](#)
- [DS841, GW1NZ series of FPGA Products Data Sheet](#)
- [DS891, GW1NSE series 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](#)
- [DS1103E, GW5A series of FPGA Products Data Sheet](#)
- [DS981E, GW5AST series of FPGA Products Data Sheet](#)
- [DS1108, GW5AR series of FPGA Products Data Sheet](#)

- [DS1105, GW5AS series of FPGA Products Data Sheet](#)
- [SUG100, Gowin Software User Guide](#)

1.3 Terminology and Abbreviations

Table 1-1 shows the abbreviations and terminology used in this manual.

Table 1-1 Terminology and Abbreviations

Terminology and Abbreviations	Meaning
CDC	Communication Device Class
JTAG	Joint Test Action Group
FPGA	Field Programmable Gate Array
IP	Intellectual Property
TAP	Test Access Port
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
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 Overview

2.1 Introduction

Gowin USB to Multi Serial Protocol Bridge IP achieves the capability to convert between a USB interface and multiple serial protocols, including JTAG, UART, SPI, and I2C. By employing the IP, users can achieve communication between the USB bus and multiple serial devices. This capability enhances the flexibility and convenience of serial communication within embedded systems.

Table 2-1 Gowin USB to Multi Serial Protocol Bridge IP

Gowin USB to Multi Serial Protocol Bridge IP	
Logic Resource	See Table 2-2
Delivered Doc.	
Design Files	Verilog (encryption)
Reference Design	Verilog
TestBench	Verilog
Test and Design Flow	
Synthesis Software	GowinSynthesis
Application Software	Gowin Software (V1.9.9(64-bit) and above)

Note!

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

2.2 Features

- Provides the capability to connect to a USB 2.0 high-speed interface, enabling the bridging between USB 2.0 and JTAG, UART, SPI, I2C protocols.
- USB to JTAG is utilized for configuring Gowin FPGA devices SRAM and Flash, supporting a maximum TCK frequency of 30 MHz.
- UART interface is compatible with CDC (Communication Device Class) specifications for serial port drivers.
- USB to SPI supports various operating modes, including full-duplex and half-duplex.
- USB to I2C supports multiple speed grades, such as Standard Mode (100 kHz) and Fast Mode (400 kHz).

2.3 Operating Frequency

Gowin USB to Multi Serial Protocol Bridge IP operating frequency is limited by the maximum clock frequency supported by the selected FPGA.

2.4 Resource Utilization

Gowin USB to Multi Serial Protocol Bridge IP is implemented with Verilog HDL. Its performance and resource utilization may vary when the design is employed in different devices, or at different densities, speeds, or grades. Taking Gowin GW2AR-18 devices as an instance, the resource utilization is as shown in Table 2-2. For the resource utilization of other devices, please refer to later release information.

Table 2-2 Resource Utilization

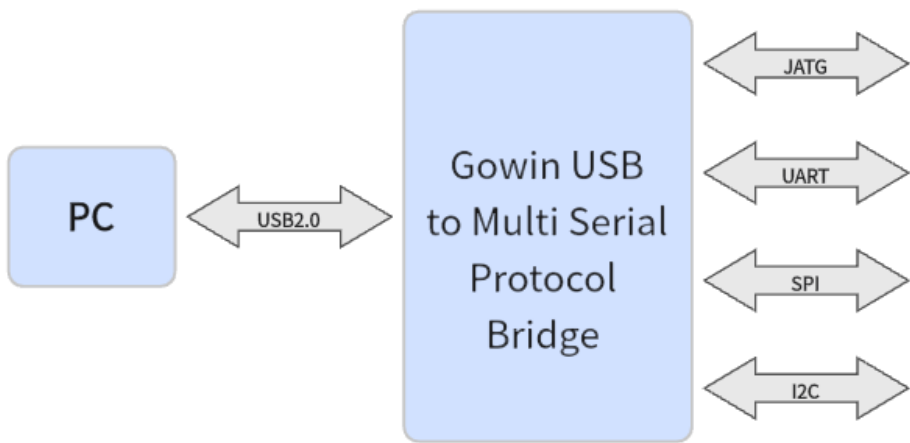
Device	Programming Language	LUTs	ALUs	REGs	BSRAM	SSRAM
GW2AR-18	Verilog HDL	3136	376	1787	6	16

3 Functional Description

3.1 System Block Diagram

Between Gowin USB to Multi Serial Protocol Bridge IP and the Config-Engine (CFG), four communication protocol interfaces are supported.

Figure 3-1 System Diagram



3.2 Introduction to USB to Multi Serial Protocol Bridge Module

3.2.1 USB High-speed Device (480Mbps)

Table 3-1 Device ID

VID	0x33AA
PID	0x0021

3.2.2 Endpoint Count

Table 3-2 Endpoint Count

Endpoint Type	Count	Note
CTRL	1	Endpoint 0, used for enumeration and register configuration
IN, BULK	1	Data input

Endpoint Type	Count	Note
OUT	1	Data output

The following is an example for enumeration endpoint.

07 05 83 02 00 02 00

07 05 03 02 00 02 00

3.2.3 Interface Endpoint Address

Table 3-3 Interface Endpoint Address

Name	Receive Endpoint Address	Transmit Endpoint Address
UART	0x82	0x02
JTAG	0x83	0x03
SPI	0x84	0x04
I2C	0x85	0x05

3.2.4 Interface Implementation Reference

The UART interface is compatible with CDC specifications for serial port drivers; for the SPI and I2C interface implementation, refer to the SPI and I2C interface standards; for the JTAG interface, refer to the related content within this manual.

3.3 Interface Parameters

3.4 Requests Related to UART Virtual Serial Port

The host initiates a request to obtain the current serial port attributes, including the baud rate, stop bits, parity bits, and the data bits.

Table 3-4 GET LINE CODING Request Encoding Format

Byte	7:6	5:4	3:2	1	0
Name	Length	Index	Value	Req	Type
Value	0x7	0x0	0x0	0x21	0xA1

Table 3-5 GET LINE CODING Return Parameter

Byte	6	5	4	3:0
Name	Data Bits	Parity Bits	Stop Bits	Baud Rate

When the host sets the current slave attributes, the baud rate, stop bits, parity bits, and data bits can be modified, and there is no data output for this request.

Table 3-6 SET LINE CODING Request Encoding Format

Byte	7:6	5:4	3:2	1	0
Name	Length	Index	Value	Req	Type

Value	0x7	0x0	0x0	0x20	0x21
-------	-----	-----	-----	------	------

3.5 I2C Complete Data Packet Format

A complete I2C data packet consists of a 4-byte command, a 4-byte data length, and N bytes of data, and the N bytes of data is determined by the data length. For example, a data packet 0x00030000 00000003 112233, where 00030000 is the command, 00000003 is the data length, and 112233 is the data.

Table 3-7 I2C Data Packet Format

Command	Data Length	Data
4 bytes	4 bytes	N bytes (determined by data length)

Table 3-8 I2C Command Set

Command	Command Length	Description
0x00030000	4 bytes	Set Register Command
0x00030001	4 bytes	Read Register Command
0x00030002	4 bytes	Read Data Command

3.6 I2C Register Configuration

Gowin I2C has a total of 8 registers, each with a width of 32 bits.

- Interrupt Enable Register
- Status Register
- Address Register
- Data Register
- Control Register
- Command Register
- Setup Register
- I2C Timing Parameter Multiple Register

Table 3-9 Gowin I2C Registers

Name	Address	Width	Type	Description
Interrupt Enable Register	0x5	32	Write/Read	–
Status Register	0x6	32	RW1C	
Address Register	0x7	32	Write/Read	
Data Register	0x8	32	Write/Read	
Control Register	0x9	32	Write/Read	
Command Register	0xA	32	Write/Read	
Setup Register	0xB	32	Write/Read	
I2C Timing Parameter Multiple Register	0xC	32	Write/Read	

3.6.1 Interrupt Enable Register(0x5)

Table 3-10 Interrupt Enable Register

Bit	Name	Type	Description	Reset
31:10	Reserved	–	–	–
9	Cmpl	Write/ Read	Enable interrupt Master: Triggered when a transaction from this master device completed without losing bus arbitration. Slave: Triggered when a transaction targeting this controller completed.	0x0
8	ByteRecv	Write/ Read	Enable byte receive interrupt Triggered when a byte of data is received.	0x0
7	ByteTrans	Write/ Read	Enable byte transmit interrupt Triggered when a byte of data is transmitted.	0x0
6	Start	Write/ Read	Enable start condition interrupt Triggered when a start/repeated start condition is detected.	0x0
5	Stop	Write/ Read	Enable stop condition interrupt Triggered when a stop condition is detected.	0x0
4	ArbLose	Write/ Read	Enable arbitration lost interrupt Master: Triggered when the controller loses bus arbitration. Slave: Not available in this mode.	0x0
3	AddrHit	Write/ Read	Enable address hit interrupt Master: Triggered when the addressed slave device returns ACK. Slave: Triggered when the controller is addressed.	0x0
2	FIFOHalf	Write/ Read	Enable FIFO half interrupt This interrupt is dependent on the transfer direction. Enable this interrupt only when the transfer direction is known to avoid triggering unexpected interrupts.	0x0
1	FIFOFull	Write/ Read	Enable FIFO full interrupt Triggered when the FIFO is full.	0x0
0	FIFOEmpty	Write/ Read	Enable FIFO empty interrupt Triggered when the FIFO is empty.	0x0

3.6.2 Status Register (0x6)

Table 3-11 Status Register

Bit	Name	Type	Description	Reset
31:15	Reserved	-	Reserved bits	–
14	LineSDA	Read-	Indicates the current status of the	SDA line

Bit	Name	Type	Description	Reset
		only	SDA line on the bus. 1: High level 0: Low level	status
13	LineSCL	Read-only	Indicates the current status of the SCL line on the bus. 1: High level 0: Low level	SCL line status
12	GenCall	Read-only	Indicates whether the address of the current transaction is a general call address. 1: general call address 0: Non-general call address	0x0
11	BusBusy	Read-only	Indicates whether the bus is busy. 1: Busy 0: Not busy	0x0
10	ACK	Read-only	Indicates the type of the last received/transmitted acknowledge bit. 1: ACK 0: NACK	0x0
9	Cmpl	R/W1C	Transaction completion flag In master mode, it indicates that a transaction has been initiated and successfully completed. In slave mode, it indicates completion of a transaction targeting this controller. This status bit needs to be cleared to receive the next transaction; otherwise, the next incoming transaction will be blocked.	0x0
8	ByteRecv	R/W1C	Indicates that a byte of data has been received.	0x0
7	ByteTrans	R/W1C	Indicates that a byte of data has been transmitted.	0x0
6	Start	R/W1C	Indicates the transmission/reception of the START signal or a repeated START signal.	0x0
5	Stop	R/W1C	Indicates the transmission/reception of the STOP signal.	0x0
4	ArbLose	R/W1C	Indicates that the controller lost bus arbitration (Master mode only)	0x0
3	AddrHit	R/W1C	Master mode: Indicates that a slave device responded to the transaction. Slave mode: Indicates that the transaction target is the controller	0x0

Bit	Name	Type	Description	Reset
			(general call address included).	
2	FIFOHalf	Read-only	Transmitter: Indicates that the FIFO buffer is at least half empty. Receiver: Indicates that the FIFO buffer is at least half full.	0x0
1	FIFOFull	Read-only	Indicates that the FIFO buffer is full.	0x0
0	FIFOEmpty	Read-only	Indicates that the FIFO buffer is empty.	0x1

Note!

R/W1C indicates read and write, and writing 1 will automatically clear it to 0.

3.6.3 Address Register (0x7)

The address register stores the address of the slave device. When operating as the master program, it represents the target slave address for the next transaction. When operating as the slave program, it serves as the address of the controller on the bus.

Table 3-12 Transmit Register

Bit	Name	Type	Description	Reset
31:10	Reserved	-	Reserved bits	—
9:0	Addr	Write/Read	Slave address In 7-bit addressing mode, the most significant 3 bits are ignored, and only the least significant 7 bits of Addr are valid.	—

3.6.4 Data Register (0x8)

The address register stores the address of the slave device. When operating as the master program, it represents the target slave address for the next transaction. When operating as the slave program, it serves as the address of the controller on the bus.

Table 3-13 Transmit Register

Bit	Name	Type	Description	Reset
31:10	Reserved	—	Reserved bits	—
9:0	Addr	Write/Read	Slave address In 7-bit addressing mode, the most significant 3 bits are ignored, and only the least significant 7 bits of Addr are valid.	—

3.6.5 Control Register (0x9)

The control register is used to control the selection of transaction phase and record the progress of the data phase.

Table 3-14 Status Register

Bit	Name	Type	Description	Reset
31:13	Reserved	–	Reserved bits	–
12	Phase_start	Write/ Read	Enable this bit to send a START condition at the beginning of a transaction, only valid in master mode.	0x1
11	Phase_addr	Write/ Read	Enable this bit to send the address after the START condition, only valid in master mode.	0x1
10	Phase_data	Write/ Read	Enable this bit to send data after the address phase, only valid in master mode.	0x1
9	Phase_stop	Write/ Read	Enable this bit to send a STOP condition at the end of a transaction, only valid in master mode.	0x1
8	Dir	Write/ Read	Transaction direction Master mode: Set this bit to determine the direction of the next transaction. 0: Transmitter 1: Receiver Slave mode: Direction of the last received transaction. 0: Receiver 1: Transmitter.	0x0
7:0	DataCnt	Write/ Read	Data count, in bytes Master mode: Number of bytes to be transmitted/received. 0 represents 256 bytes. DataCnt decrements once for each byte transmitted/received. Slave mode: The meaning of DataCnt depends on DMA.	0x0

3.6.6 Command Register (0xA)

Table 3-15 Command Register

Bit	Name	Type	Description	Reset
31:3	Reserved	–	Reserved bits	–
2:0	CMD	Write/ Read	0x0: No operation 0x1: Initiate data transaction (Master mode only) 0x2: Response ACK to the received byte 0x3: Response NACK to the received byte 0x4: Clear FIFO	–

Bit	Name	Type	Description	Reset
			0x5: Reset the I2C controller, reset the status register and interrupt enable register, and empty the FIFO.	

3.6.7 Setup Register (0xB)

Table 3-16 Setup Register

Bit	Name	Type	Description	Reset
31:29	Reserved	–	Reserved bits	–
28:24	T_SUDAT	Write/ Read	T_SUDAT defines the data setup time before releasing SCL	0x5
23:21	T_SP	Write/ Read	T_SP defines the pulse width that must be suppressed by the input filter	0x1
20:16	T_HDDAT	Write/ Read	T_HDDAT defines the data hold time after SCL becomes low	0x5
15:14	Reserved	–	Reserved bits	–
13	T_SCLRatio	Write/ Read	The low period of the generated SCL clock is defined by the combination of T_SCLRatio and T_SCLHi values. When T_SCLRatio = 0, the low period is equal to the high period. When T_SCLRatio = 1, the low period is approximately twice the high period; for 1: ratio = 2 and 0: ratio = 1, this field is only valid when the controller is in master mode.	0x1
12:4	T_SCLHi	Write/ Read	The high period of the generated SCL clock is defined by T_SCLHi, and this field is only valid when the controller is in master mode.	0x10
3	DMAEn	Write/ Read	Enable the direct memory access mode data transfer 1: Enable 0: Disable	0x0
2	Master	Write/ Read	Configure the device as a master a slave 1: Master mode 0: Slave mode	0x0
1	Addressing	Write/ Read	I2C address mode 1: 10-bit address mode 0: 7-bit address mode	0x0
0	IICEn	Write/ Read	Enable ATCIIC100 I2C controller 1: Enable 0: Disable	0x0

3.6.8 I2C Timing Parameter Multiple Register (0xC)

This register stores a multiplier value used to enlarge the I2C bus timing parameters defined in the Setup Register. This multiplier value assists the controller in meeting the timing requirements of the I2C bus interface when operating at higher APB frequency.

Table 3-17 I2C Timing Parameter Multiple Register

Bit	Name	Type	Description	Reset
31:5	Reserved	–	Reserved bits	–
4:0	TPM	Write/ Read	The multiplication value for I2C timing parameters. And All timing parameters in the Setup Register will be multiplied by (TPM+1).	0x0

3.7 SPI Complete Data Packet Format

A complete SPI data packet consists of a 4-byte command, a 4-byte data length, and N bytes of data, and the N bytes of data is determined by the data length. For example, a data packet 0x00020000 00000003 112233, where 00020000 is the command, 00000003 is the data length, and 112233 is the data.

Table 3-18 SPI Data Packet Format

Command	Data Length	Data
4 bytes	4 bytes	N bytes (determined by length)

Table 3-19 SPI Command Set

Command	Command Length	Description
0x00020000	4 bytes	Set register command
0x00020001	4 bytes	Read register command
0x00020002	4 bytes	Read data command

3.8 SPI Register Configuration

Gowin SPI has a total of 11 registers, each with a width of 32 bits.

- Transfer Format Register
- Transfer Control Register
- Command Register
- Address Register
- Data Register
- Control Register
- Status Register
- Interrupt Enable Register
- Interrupt Status Register

- Interface Timing Register
- Configuration Register

Table 3-20 SPI Register

Name	Address	Width	Type	Description
Transfer Format Register	0x04	32	Write/Read	–
Transfer Control Register	0x08	32	Write/Read	
Command Register	0x09	32	Write/Read	
Address Register	0x0A	32	Write/Read	
Data Register	0x0B	32	Write/Read	
Control Register	0x0C	32	Write/Read	
Status Register	0x0D	32	Read-only	
Interrupt Enable Register	0x0E	32	Write/Read	
Interrupt Status Register	0x0F	32	W1C	
Interface Timing Register	0x10	32	Write/Read	
Configuration Register	0x1F	32	Read-only	

3.8.1 Transfer Format Register (0x04)

This register defines the SPI transfer format.

Table 3-21 Transfer Format Register

Bit	Name	Type	Description	Reset
31:18	Reserved	–	Reserved bits	–
17:16	AddrLen	Write/ Read	Address length in bytes	0x2
15:13	Reserved	–	Reserved bits	–
12:8	DataLen	Write/ Read	Length of each data unit (in bits), and the actual number of bits per data unit is (DataLen + 1).	0x07
7	DataMerge	Write/ Read	Enable data merge mode, which automatically splits data on write and merges data on read. This bit is valid only when DataLen = 0x7. In data merge mode, each write to the data register will transmit all four bytes of write data; each read from the data register will retrieve four bytes of received data as a single-word data.	0x1
6:5	Reserved	–	Reserved bits	–
4	MOSIBiDir	Write/ Read	Bi-directional MOSI in normal (single) mode: 0x0: In normal mode, MOSI is a	0x0

Bit	Name	Type	Description	Reset
			unidirectional signal. 0x1: In normal mode, MOSI is a bidirectional signal. This bidirectional signal replaces two signal lines.	
3	LSB	Write/ Read	Transfer data with the least significant bit 0x0: Most significant bit first 0x1: Least significant bit first	0x0
2	SlvMode	Write/ Read	SPI master/slave mode selection 0x0: Master mode 0x1: Slave mode	—
1	CPOL	Write/ Read	SPI clock polarity 0x0: SCLK is low in the idle state 0x1: SCLK is high in the idle state	—
0	CPHA	Write/ Read	SPI clock phase 0x0: Sample data at odd SCLK edges 0x1: Sample data at even SCLK edges	—

3.8.2 Transfer Control Register (0x08)

Table 3-22 Transfer Control Register

Bit	Name	Type	Description	Reset
31	SlvDataOnly	Write/ Read	Data-only mode (slave mode only) 0x0: Disable 0x1: Enable	0x0
30	CmdEn	Write/ Read	SPI command phase enable (master mode only) 0x0: Disable 0x1: Enable	0x0
29	AddrEn	Write/ Read	SPI address phase enable (master mode only) 0x0: Disable 0x1: Enable	0x0
28	AddrFmt	Write/ Read	SPI address phase format (master mode only) 0x0: Address phase is the regular mode 0x1: The format of the address phase is the same as the data phase	0x0
27:24	TransMode	Write/ Read	Transfer Mode 0x0: Write and read simultaneously 0x1: Write only 0x2: Read only 0x3: Write and read 0x4: Read and write 0x5: Write, dummy, and read 0x6: Read, dummy, and write 0x7: No data	0x0

Bit	Name	Type	Description	Reset
			0x8: Dummy and write 0x9: Dummy and read 0xa~0xf: Reserved	
23:22	DualQuad	Write/ Read	SPI data phase format 0X0: Regular (Single) mode 0X1: Dual I/O mode 0X2: Quad I/O mode 0x3: Reserved	0x0
21	TokenEn	Write/ Read	Token transfer enable Append an one-byte special token following the address phase for SPI read transfer. 0X0: Disable the one-byte special token 0X1: Enable the one-byte special token	0x0
20:12	WrTranCnt	Write/ Read	Transfer count for write data	0x0
11	TokenValue	Write/ Read	The value of the one-byte special token following the address phase for SPI read transfer. 0x0: Token value is 0x00. 0x1: Token value is 0x69.	0x0
10:9	DummyCnt	Write/ Read	Dummy data count The actual dummy count is (DummyCnt + 1).	0x0

3.8.3 Command Register (0x09)

Writing operations on this register will trigger SPI transfers. Even if the command phase is not enabled, a dummy value must be written to this register to start a SPI transfer.

Table 3-23 Command Register

Bit	Name	Type	Description	Reset
31:8	Reserved	–	Reserved bits	–
7:0	CMD	Write/Read	SPI Command	0x0

3.8.4 Address Register (0x0A)

Table 3-24 Address Register

Bit	Name	Type	Description	Reset
31:10	Reserved	–	Reserved bits	–
9:0	ADDR	Write/Read	SPI address (Master mode only)	0x0

3.8.5 Data Register (0x0B)

When the controller is in data merge mode, the byte order of the SPI data register is little-endian. For write operations, data will be stored into the TX FIFO. The least significant byte is always transmitted first.

If the TX FIFO is full and the SPIActive bit in the status register is set to 1, the ready signal (hready/pready) will be de-asserted to insert a wait state into the transfer. For read operations, data will be read and dequeued from the RX FIFO. The least significant byte is the first byte received.

If the RX FIFO is empty and the SPIActive bit in the status register is set to 1, the ready signal (hready/pready) will be de-asserted to insert a wait state into the transfer.

Table 3-25 Data Register

Bit	Name	Type	Description	Reset
31:0	DATA	Write/Read	SPI data	0x0

3.8.1 Control Register (0x0C)

Table 3-26 Control Register

Bit	Name	Type	Description	Reset
31:24	Reserved	–	Reserved bits	–
23:16	TXTHRES	Write/Read	TX FIFO threshold	0x0
15:08	RXTHRES	Write/Read	RX FIFO threshold	0x0
7:5	Reserved	–	Reserved bits	–
4	TXDMAEN	Write/Read	TX DMA enable	0x0
3	RXDMAEN	Write/Read	RX DMA enable	0x0
2	TXFIFORST	Write/Read	TX FIFO reset. Write 1 to reset. It is automatically clears to 0 after the reset operation is completed.	0x0
1	RXFIFORST	Write/Read	RX FIFO reset. Write 1 to reset. It is automatically clears to 0 after the reset operation is completed.	0x0
0	SPIRST	Write/Read	SPI reset. Write 1 to reset. It is automatically clears to 0 after the reset operation is completed.	0x0
23:16	TXTHRES	Write/Read	TX FIFO threshold	0x0

3.8.2 Status Register (0x0D)

Table 3-27 Status Register

Bit	Name	Type	Description	Reset
31:30	Reserved	–	Reserved bits	–
29:28	TXNUM[7:6]	Read-only	Number of valid entries in the TX FIFO	0x0
27:26	Reserved	–	Reserved bits	–
25:24	RXNUM[7:6]	Read-only	Number of valid entries in RX FIFO	0x0
23	TXFULL	Read-only	The flag of TX FIFO full	0x0
22	TXEMPTY	Read-only	The flag of TX FIFO empty	0x1

Bit	Name	Type	Description	Reset
21:16	TXNUM[5:0]	Read-only	Number of valid entries in the TX FIFO	0x0
15	RXFULL	Read-only	The flag of RX FIFO full	0x0
14	RXEMPTY	Read-only	The flag of RX FIFO empty	0x1
13:8	RXNUM[5:0]	Read-only	Number of valid entries in RX FIFO	0x0
7:1	Reserved	–	Reserved bits	–
0	SPIActive	Read-only	In master mode, SPIActive becomes 1 after the SPI command register is written and becomes to 0 after the transfer is completed. In slave mode, SPIActive become 1 after CS signal is asserted and becomes to 0 after the CS signal is de-asserted.	–

3.8.3 Interrupt Enable Register(0x0E)

Table 3-28 Interrupt Enable Register

Bit	Name	Type	Description	Reset
31:6	Reserved	–	Reserved bits	–
5	SlvCmdEn	Write/ Read	Enable slave command Interrupt Control whether an interrupt is triggered upon receiving a slave mode command (Slave mode only).	0x0
4	EndIntEn	Write/ Read	Enable SPI transfer end interrupt Control whether an interrupt is triggered when SPI transfers end (In slave mode, end of read status transaction end does not trigger this interrupt).	0x0
3	TXFIFOIntEn	Write/ Read	Enable SPI TX FIFO threshold interrupt Control whether an interrupt is triggered when the number of valid entries is less than or equal to the TX FIFO threshold.	0x0
2	RXFIFOIntEn	Write/ Read	Enable SPI RX FIFO threshold interrupt Control whether an interrupt is triggered when the number of valid entries is greater than or equal to the RX FIFO threshold.	0x0
1	TXFIFOURIntEn	Write/ Read	Enable SPI TX FIFO underrun interrupt Control whether an interrupt is triggered when the TX FIFO runs out of data (Slave mode only).	0x0

Bit	Name	Type	Description	Reset
0	RXFIFOORInt En	Write/ Read	Enable SPI RX FIFO overrun interrupt Control whether an interrupt is triggered when the RX FIFO overrun (Slave mode only).	0x0

3.8.4 SPI Interrupt Status Register (0x0F)

Table 3-29 SPI Interrupt Status Register

Bit	Name	Type	Description	Reset
31:6	Reserved	-	Reserved bits	–
5	SlvCmdEn	W1C	Enable slave command Interrupt Control whether an interrupt is triggered upon receiving a slave mode command (Slave mode only).	0x0
4	EndIntEn	W1C	Enable SPI transfer end interrupt Controls whether an interrupt is triggered when SPI transfer completes. (In slave mode, end of read status transaction end does not trigger this interrupt.)	0x0
3	TXFIFOIntEn	W1C	Enable SPI TX FIFO threshold interrupt Control whether an interrupt is triggered when the number of valid entries is less than or equal to the TX FIFO threshold.	0x0
2	RXFIFOIntEn	W1C	Enable SPI RX FIFO threshold interrupt Control whether an interrupt is triggered when the number of valid entries is greater than or equal to the TX FIFO threshold.	0x0
1	TXFIFOURInt En	W1C	Enable SPI TX FIFO underrun interrupt Control whether an interrupt is triggered when the TX FIFO runs out of data (Slave mode only).	0x0
0	RXFIFOORInt En	W1C	Enable SPI RX FIFO overrun interrupt Control whether an interrupt is triggered when RX FIFO overrun (Slave mode only).	0x0

3.8.5 SPI Interface Timing Register (0x10)

Table 3-30 SPI Interface Timing Register

Bit	Name	Type	Description	Reset
31:14	Reserved	–	Reserved bits	0x0
13:12	CS2SCLK	Write/Re ad	Minimum time between SPI CS edge and SCLK edge	0x0

Bit	Name	Type	Description	Reset
11:8	CST	Write/Read	Minimum time that SPI CS should be held HIGH	0x0
7:0	SCLK_DIV	Write/Read	Clock frequency ratio between the clock source and SPI interface SCLK	0x0
7:0	Reserved	–	Reserved bits	0x0

3.8.6 Configuration Register (0x1F)

Table 3-31 Configuration Register

Bit	Name	Type	Description	Reset
31:15	Reserved	–	Reserved bits	–
14	Slave	Read-only	Support SPI slave mode	Configuration dependent
13	EILMMem	Read-only	Supports memory-mapped access through EILM bus (read-only)	Configuration dependent
12	AHBMem	Read-only	Supports memory-mapped access through AHB bus (read-only)	Configuration dependent
11	DirectIO	Read-only	Support direct SPI IO	Configuration dependent
10	Reserved	–	Reserved bits	–
9	QuadSPI	Read-only	Support quad I/O SPI	Configuration dependent
8	DualSPI	Read-only	Support dual I/O SPI	Configuration dependent
7:4	TxFIFOSize	Read-only	Depth of TX FIFO	Configuration dependent
3:0	RxFIFOSize	Read-only	Depth of RX FIFO	Configuration dependent

3.9 JTAG Register Configuration

3.9.1 Configuration Register Command Frame Format

Table 3-32 Set Registers

Command	Base Address	Offset Address	Length	Data
0xA0	1 byte	1 byte	1 byte	N bytes

Table 3-33 Read Registers

Command	Base Address	Offset Address	Length
0xA1	1 byte	1 byte	1 byte

Table 3-34 Response Frame

Identifier		Data
0xAA	0x55	N bytes

3.9.2 Register Information

Table 3-35 Register Information

Type	Base Address	Offset Address	Name	Description
Global	0x00	0x00	Mode Selection	0x00: JTAG1
JTAG1	0x01	0x00	TCK divider factor n	0x00: 30MHZ Formula: $TCK \text{ Frequency} = 30 / (n * 2)$, $n > 0$; For example: 0x01: 15MHZ 0x02: 7.5MHZ 0x03: 5MHZ 0x04: 3.75MHZ

4 Port Description

4.1 Port List

The ports of Gowin USB to Multi Serial Protocol Bridge IP are as shown in Figure 4-1.

Figure 4-1 Gowin USB to Multi Serial Protocol Bridge IP Port Diagram

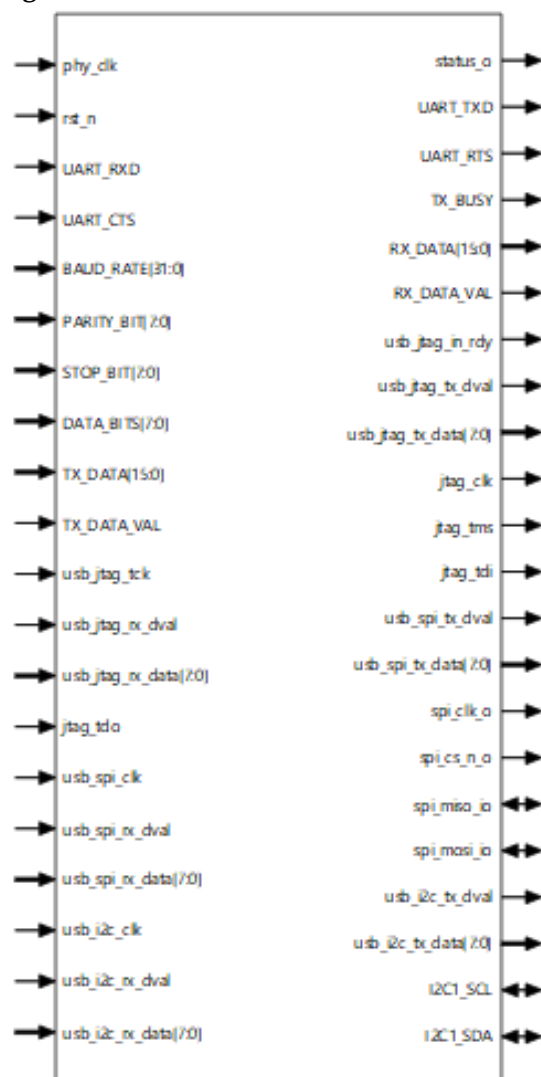


Table 4-1 Gowin USB to Multi Serial Protocol Bridge IP Port List

Signal Name	I/O	Data Width	Description
phy_clk	input	1	IP clock input
rst_n	output	1	Reset signal, active-low
UART_RXD	input	1	UART RX data
UART_CTS	input	1	UART RX data control signal
BAUD_RATE	input	[31:0]	Baud rate for UART communication
DATA_BITS	input	[7:0]	Data bits for UART communication
TX_DATA	input	[15:0]	UART TX data
TX_DATA_VAL	input	1	UART TX data valid signal
usb_jtag_tck	input	1	JTAG clock input
usb_jtag_rx_dval	input	1	JTAG RX data valid signal
usb_jtag_rx_data	input	[7:0]	JTAG RX data
jtag_tdo	input	1	JTAG interface output data
usb_spi_clk	input	1	SPI clock input
usb_spi_rx_dval	input	1	SPI RX data valid signal
usb_spi_rx_data	input	[7:0]	SPI RX data
usb_i2c_clk	input	1	I2C clock input
usb_i2c_rx_dval	input	1	I2C RX data valid signal
usb_i2c_rx_data	input	[7:0]	I2C RX data
status_o	output	1	IP status output
UART_TXD	output	1	UART TX data
UART_RTS	output	1	UART TX data control signal
TX_BUSY	output	1	UART interface is busy transmitting data
RX_DATA	output	[15:0]	UART RX data
RX_DATA_VAL	output	1	UART RX data control signal
usb_jtag_in_rdy	output	1	JTAG data input ready signal
usb_jtag_tx_dval	output	1	JTAG TX data valid signal
usb_jtag_tx_data	output	[7:0]	JTAG TX data
jtag_clk	output	1	JTAG interface clock
jtag_tms	output	1	JTAG mode selection signal
jtag_tdi	output	1	JTAG interface data input
usb_spi_tx_dval	output	1	SPI TX data valid signal
usb_spi_tx_data	output	[7:0]	SPI TX data
spi_clk_o	output	1	SPI interface clock
spi_cs_n_o	output	1	SPI chip selection signal output
spi_miso_o	inout	1	SPI master input/slave output
spi_mosi_o	inout	1	SPI master output/slave input
usb_i2c_tx_dval	output	1	I2C TX data valid signal
usb_i2c_tx_data	output	[7:0]	I2C TX data
I2C1_SCL	output	1	I2C interface clock line

Signal Name	I/O	Data Width	Description
I2C1_SDA	output	1	I2C interface data line

4.2 Parameter Configuration

Table 4-2 Gowin USB to Multi Serial Protocol Bridge IP Parameter Configuration

No.	Name	Option	Default Value	Description
1	Interface Selection	UART SPI JTAG I2C	UART SPI JTAG I2C	You can check UART, SPI, JTAG, I2C

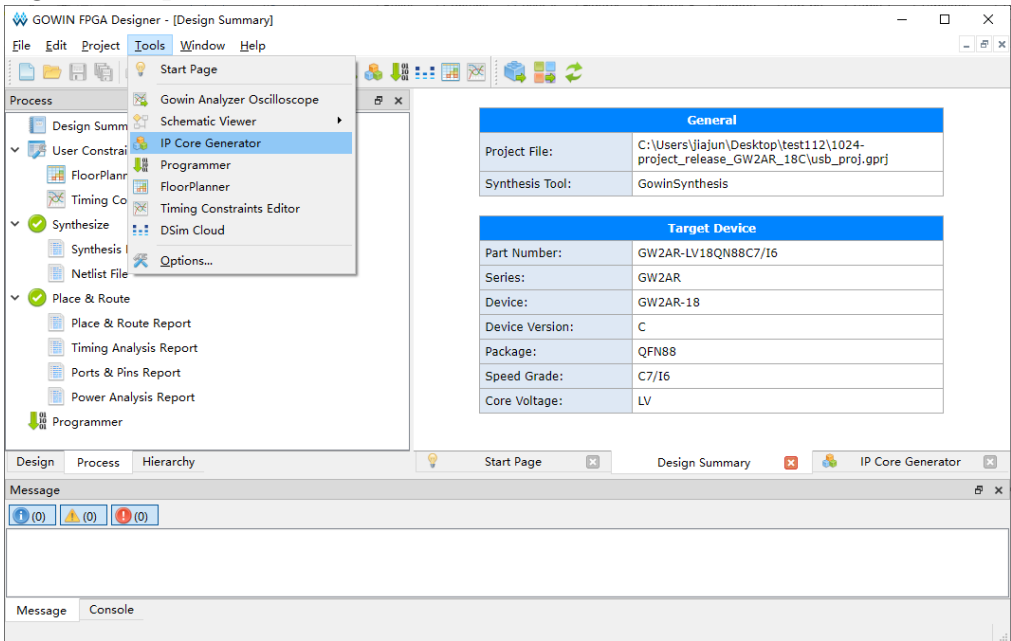
5 Interface Configuration

You can use IP core generator tool in Gowin Software to call and configure Gowin USB to Multi Serial Protocol Bridge IP.

- 1. Open IP Core Generator.

After creating the project, click the "Tools" tab in the upper left, click "IP Core Generator" from the drop-down list to open Gowin IP Core Generator, as shown in Figure 5-1.

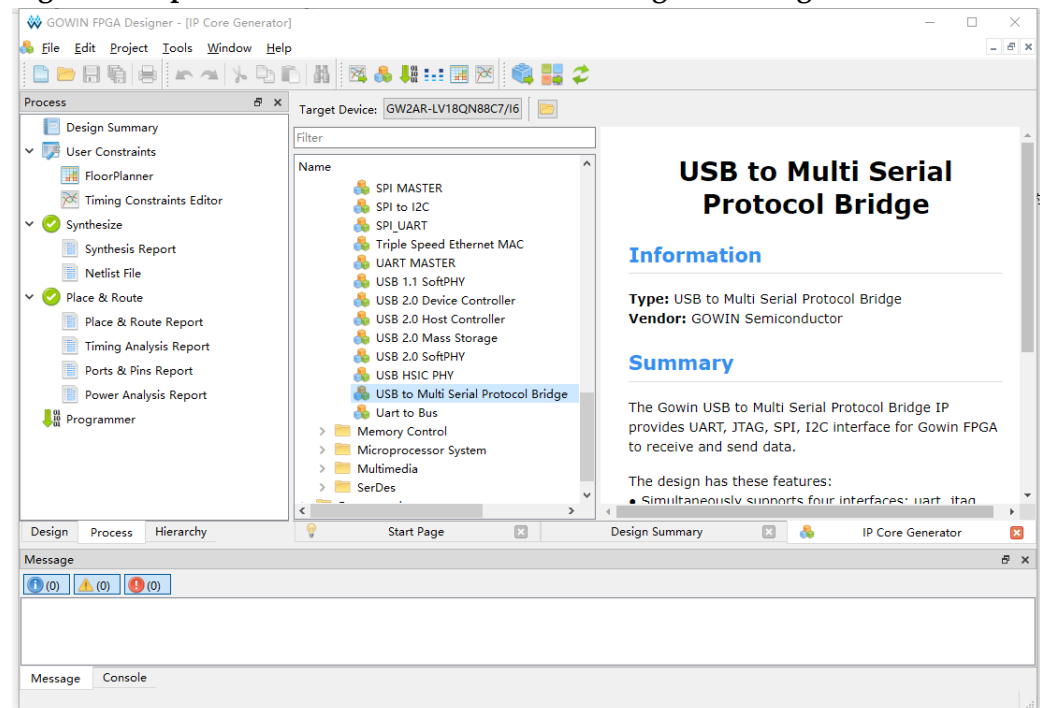
Figure 5-1 Open IP Core Generator



2. Open USB to Multi Serial Protocol Bridge IP configuration interface

Double click "USB To Multi Serial Protocol Bridge" to open the configuration interface, as shown in Figure 5-2.

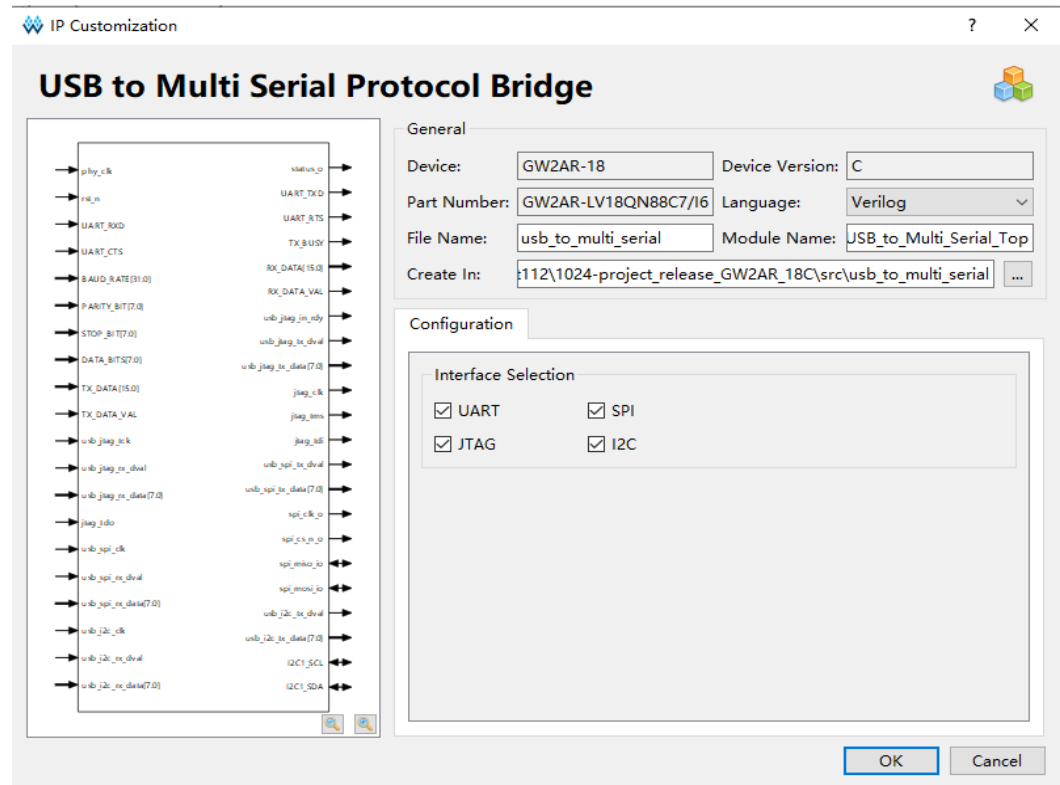
Figure 5-2 Open USB to Multi Serial Protocol Bridge IP Configuration Interface



3. Configure USB to Multi Serial Protocol Bridge IP

Select different communication interfaces such as UART, SPI, JTAG, I2C by checking the parameters in "Interface Selection" as shown in Figure 5-3.

Figure 5-3 USB to Multi Serial Protocol Bridge Configuration Interface



4. IP Generation

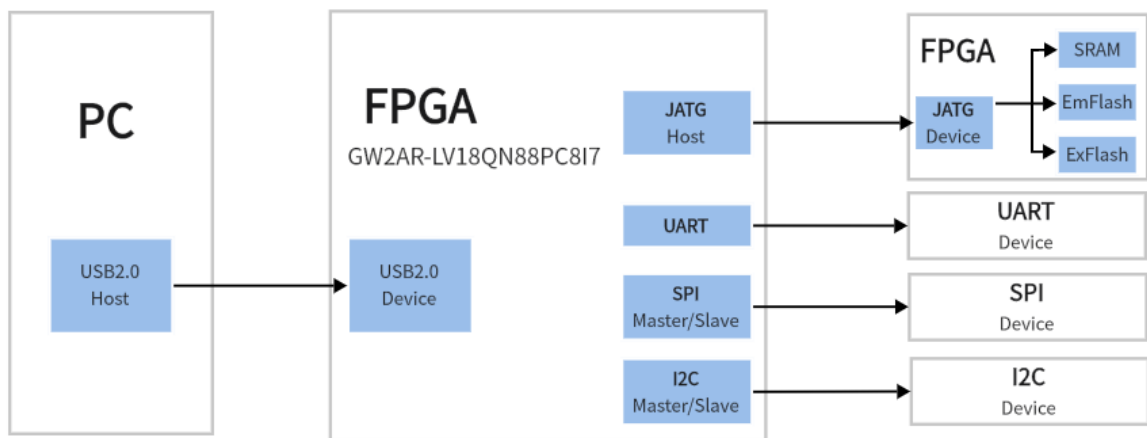
After completing the configuration, click the "OK" button at the bottom right corner of the interface to generate the USB to Multi Serial Protocol Bridge IP files.

6 Reference Design

This chapter is intended to introduce the usage and structure of the reference design of USB to Multi Serial Protocol Bridge IP. For the details, you can see USB to Multi Serial Protocol Bridge IP Reference Design at Gowinsemi website.

This reference design uses the following development board: DK_USB2.0_GW2AR-LV18QN88PC8I7_GW1NSR-LV4CMG64PC7I6_V3.0. You can see more information about the development board at Gowinsemi website. The block diagram of reference design is shown in Figure 6-1.

Figure 6-1 Block Diagram of Reference Design



In the reference design, the JTAG host can obtain the FPGA device's ID, which can be used for erasing and configuring the FPGA device's SRAM, as well as erasing, writing, and reading the internal and external Flash. The UART interface can be utilized for data transfer between standard UART devices. The SPI interface can operate as a Master or Slave to communicate with standard SPI devices. Similarly, the I2C interface can function as a Master or Slave for communication with standard I2C devices.

7 File Delivery

The delivery file for Gowin USB to Multi Serial Protocol Bridge IP includes documentation and design source code.

7.1 Documentation

The folder mainly contains the user guides in PDF version.

Table 7-1 Documents List

Name	Description
IPUG1180, Gowin USB to Multi Serial Protocol Bridge IP	Gowin USB to Multi Serial Protocol Bridge, i.e., this manual.
UG1251, Gowin USB to Multi Serial Protocol Bridge Software API User Guide.	Software driver API function description

7.2 Design Source Code (Encryption)

The encrypted code file contains Gowin USB to Multi Serial Protocol Bridge IP RTL encrypted code which is used for GUI in order to cooperate with Gowin software to generate the IP core required by users.

Table 7-2 Design Source Code List of USB to Multi Serial Protocol Bridge

Name	Description
usb_to_multi_Serial.v	The top-level file of the IP core, which provides users with interface information, encrypted.

