Gowin I2C Master and Slave

User Guide

IPUG504-1.5E, 09/29/2019

Disclaimer

GOWINSEMI®, LittleBee®, Arora™, and the GOWINSEMI logos are trademarks of GOWINSEMI and are 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, as described at www.gowinsemi.com.cn. 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. All information in this document should be treated as preliminary. 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
01/4/2018	1.1E	Initial version published.
03/28/2019	1.2E	Supported products updated.
05/08/2019	1.3E	Changed AXI interface to SRAM interface.
07/16/2019	1.4E	Interface configuration added.
09/30/2019	1.5E	I2C Master released as an IP; I2C Slave released as an open source reference design.

Contents

List of Figures

List of Tables

1 About This Guide

1.1 Purpose

This Gowin I2C Master and Slave user guide is designed to help users make the most of the features and usage of the Gowin I2C Master IP and Slave reference design. It contains an overview of the functions, signals definition, working principle, instantiation, etc. of the Gowin 12C Master and Slave operation.

1.2 Supported Products

The information in this guide applies to the following products:

1. GW1N series of FPGA products: GW1N-1, GW1N-1S , GW1N-2, GW1N-2B, GW1N-4, GW1N-4B, GW1N-6, and GW1N-9.
2. GW1NR series of FPGA products: GW1NR-4, GW1NR-4B and GW1NR-9.
3. GW1NS series FPGA products: GW1NS-2、GW1NS-2C
4. GW1NSR series FPGA products: GW1NSR-2、GW1NSR-2C
5. GW1NZ series FPGA products: GW1NZ-1
6. GW2A series of FPGA products: GW2A-18 and GW2A-55.
7. GW2AR series of FPGA products: GW2AR-18.

1.3 Related Documents

The latest user guides are available on the Gowin website. Refer to the related documents at www.gowinsemi.com:

1. DS100, GW1N series of FPGA Products Data Sheet
2. DS117, GW1NR series of FPGA Products Data Sheet
3. DS821, GW1NS series of FPGA Products Data Sheet
4. DS861, GW1NSR series of FPGA Products Data Sheet
5. DS841, GW1NZ series of FPGA Products Data Sheet
6. DS102, GW2A series of FPGA Products Data Sheet
7. DS226, GW2AR series of FPGA Products Data Sheet
8. SUG100, Gowin Software User Guide

1.4 Abbreviations and Terminology

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

Table 1-1 Abbreviations and Terminology

Abbreviations and Terminology	Full Name
FPGA	Field Programmable Gate Array
SRAM	Static Random Access Memory
I2C Bus	Inter-Integrated Circuit Bus

1.5 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
Tel: +86 755 8262 0391

2 Function

2.1 Overview

I2C is one kind of two-wire serial bus that is used to connect to the micro-controller to its peripheral equipment. I2C bus is easy to use and valid. I2C is also space-saving, which helps to reduce circuit space, the quantity of chips, and interconnection costs.

Gowin I2C Master IP is an I2C Master controller with a synchronous SRAM interface. It provides a low-speed, two-wire, serial bus interface that connects to the I2C bus via data pins (SDA) and clock pins (SCL) to complete data transmission and extend periphery components. Standard (up to 100 kHz) and fast (up to 400 kHz) I2C bus can be connected.

The Gowin I2C Slave reference design complies with the I2C bus protocol, and is mainly used to communicate with the I2C Master.

2.2 Features

Gowin I2C Master IP

- Compliant with industry standard I2C protocol.
- Bus arbitration and arbitration lost detection.
- Bus busy detection.
- Interrupt flag generation.
- Supports different I2C communication modes:
  - Standard mode (100 Kbps).
  - Fast mode (400 Kbps).
  - Fast (+) mode (1 Mbps).
  - High-speed mode (3.4 Mbps).
- Start/Stop/Repeated Start/Acknowledge generation.
- Start/Stop/Repeated Start detection.
- Supports 7-bit addressing mode.

Gowin I2C Slave

- Compliant with industry standard I2C protocol.
- Receive/Send data.
- Supports interrupt generation.
- Supports RAM and ROM.

3 Signal Definition

3.1 Definition of the Gowin I2C Master Signals

3.1.1 SRAM Interface Bus Signals

Table 3-1 SRAM Interface Signals

No.	Signal Name	I/O	Description	Remarks
1	I_CLK	I	Work clock, sampling at rising edge	–
2	I_RESETN	I	Reset signal	–
3	O_IIC_INT	O	Interrupt signal output	–
4	I_TX_EN	I	Write enable signal	SRAM write address channel signal
5	I_WADDR	I	Write address signal	
6	I_WDATA	I	Write data signal	
7	I_RX_EN	I	Read enable signal	SRAM read address channel signal
8	I_RADDR	I	Read address signal	
9	O_RDATA	O	Read data signal	

3.1.2 Definition of I2C Signals

Table 3-2 Definition of I2C Signals

No.	Signal Name	I/O	Description	Remarks
1	SCL	I/O	Serial clock line	-
2	SDA	I/O	Serial data line	-

3.2 Definition of Gowin I2C Slave Signals

Table 3-3 Definition of Gowin I2C Slave Signals

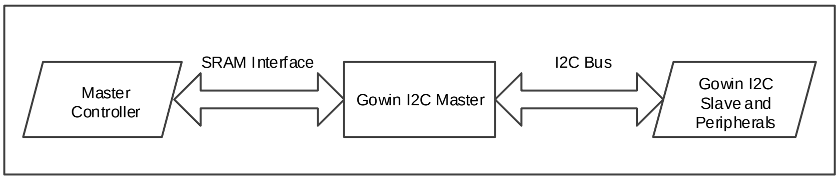
No.	Signal Name	I/O	Description	Remarks
1	clk_50m	I	Clock signal	-
2	rst_n	I	Reset signal	-
3	scl	Bidir	Serial clock line	-
4	scl_pull	O	Serial clock line pull up	-
5	sda	Bidir	Serial data line	-
6	sda_pull	O	Serial data line pull up	
7	int_o	O	Interrupt signal	

4 Principle

4.1 System Architecture

As shown in Figure 4-1, the main controller issues commands and data to the Gowin I2C Master through synchronous SRAM interface. The Gowin I2C Master then issues the information to the Gowin I2C Slave and peripheral through the I2C bus, or feeds the Gowin I2C Slave and peripheral data to the main controller through synchronous SRAM interface.

Figure 4-1 System Architecture



4.2 I2C Register

The Gowin I2C Master has six 8-bit registers in total:

- Prescale Register
- Control Register
- Transmit Register
- Receive Register
- Command Register
- Status Register

Notes!

- The Transmit Register and the Receive Register share the same address (address = 0x30).
- The Status Register and the Command Register share the same address (address = 0x04). For the detailed information, please refer to Table 4-1.

Table 4-1 Gowin I2C Master Register

Register Name	Address	Width	Type	Description
Prescale_reg0	0x00	8	Read/Write	Clock prescale register low-byte.
Prescale_reg1	0x01	8	Read/Write	Clock prescale register high-byte.
Control_reg	0x02	8	Read/Write	Control register.

Register Name	Address	Width	Type	Description
Transmit_reg	0x03	8	Write	Transmit register.
Receive_reg	0x03	8	Read	Receive register.
Command_reg	0x04	8	Write	Command register.
Status_reg	0x04	8	Read	Status register.

4.2.1 Prescale Register

The prescale register is a16-bit register that contains two 8-bit registers. The value in the prescale register is used to prescale the main clock of the I2C Master. The main clock of the I2C Master is 5*SCL, and the value in the prescale register is determined by the following equation:

[Master clock frequency / (5*sclk frequency) -1].

The prescale register is illustrated in Table 4-2 and described in Table 4-3.

Table 4-2 Prescale Register

15	0
Prescale data	

Table 4-3 Prescale Register

Bit(s)	Name	Default Value	Access Type	Description
15:0	Prescale data	0	Read/Write	16-bit prescale data

4.2.2 Control Register

The Control Register is an 8-bit register. Only two bits are used; six bits are reserved.

The MSB bit enables and disables the I2C Master:

- If MSB is 0, the I2C Master does not work.
- If MSB is 1, the I2C Master enters into working state.

The sixth bit is the interrupt bit. If it is "1," the interrupt is enabled.

The Control Register is illustrated in Table 4-4 and described in Table 4-5.

Table 4-4 Control Register

7	6	5	0
EN	IEN	Reserved	

Table 4-5 Control Register

Bit(s)	Name	Default Value	Access Type	Description
7	EN	0	Read/Write	I2C Master enable bit: ● “0”= disable Master. ● “1”= enable Master.
6	IEN	0	Read/Write	I2C Master interrupt bit: ● “0”= disable Master interrupt. ● “1”= enable Master interrupt.
5:0	Reserved	N/A	Read/Write	Reserved

4.2.3 Transmit Register

The Transmit Register stores the data to be transmitted via the I2C bus. When the address is transmitted, the bit 0 is the write/read signal. The Transmit Register is illustrated in Table 4-6 and described in Table 4-7.

Table 4-6 Transmit Register

7 1	0
TX DATA	LSB/RW

Table 4-7 Transmit Register

Bit(s)	Name	Default Value	Access Type	Description
7:1	TX DATA	Indeterminate	Write	Data to be transmitted via I2C.
0	LSB/RW	Indeterminate	Write	● This bit represents the data’s LSB. ● This bit represents the Read/Write bit during Slave address transfer: ‘1’ = reading from Slave. ‘0’ = writing to Slave.

4.2.4 Receive Register

The receive register stores the data received via the I2C bus. The receive register is illustrated in Table 4-8 and described in Table 4-9.

Table 4-8 Receive Register

7	0
RX DATA	

Table 4-9 Receive Register

Bit(s)	Name	Default Value	Access Type	Description
7:0	RX DATA	Indeterminate	Read	Byte received via I2C.

4.2.5 Status Register

This register contains the status of the I2C bus interface. The status register is cleared up on reset. The status register is illustrated in Table 4-10 and described inTable 4-11.

Table 4-10 Status Register

7	6	5	4:2	1	0
RX ACK	Busy	AL	Reserved	TIP	IF

Table 4-11 Status Register

Bit(s)	Name	Default Value	Access Type	Description
7	RX ACK	0	Read	Received acknowledge from the Slave. ● “1” = No acknowledge received. ● “0” = Acknowledge received.
6	Busy	0	Read	Indicates the I2C bus is busy. ● “1”= busy. ● “0”= idle.
5	AL	0	Read	Arbitration lost. The bit is "1" when a STOP signal is detected but not requested, or the Master drives SDA high, but SDA is low.
4:2	Reserved	N/A	Read	Reserved
1	TIP	0	Read	Transfer in progress: ● “1”= transferring data. ● “0”= transfer is completed.
0	IF	0	Read	Interrupt flag. This bit is set when an interrupt is pending, which will cause a processor interrupt request if the IEN bit is set. The bit is "1" when one-byte transfer has been completed, or arbitration is lost.

4.2.6 Command Register

The Gowin I2C Master configures I2C operation modes via write commands. The command register stores the next command for the next I2C operation, and the bits in the Command Register are cleared automatically after each operation. Therefore, the synchronous SRAM interface is rewritten every time the I2C operation is started, written, read, or stopped. The command register is illustrated in Table 4-12 and described in Table 4-13.

Table 4-12 Command Register

7	6	5	4	3	2 1	0
STA	STO	RD	WR	ACK	Reserved	IACK

Table 4-13 Command Register

Bit(s)	Name	Default Value	Access Type	Description
7	STA	0	Write	(repeated) start
6	STO	0	Write	Stop
5	RD	0	Write	Read
4	WR	0	Write	Write
3	ACK	0	Write	Acknowledge
2:1	Reserved	0	Write	Reserved
0	IACK	0	Write	Interrupt acknowledge. When set, clears a pending interrupt.

4.3 Common Operation Sequence

The Gowin I2C Master supports common I2C operations. This section mainly describes the I2C write and read operations.

4.3.1 I2C Master Bus Initialization

Program the prescale register with the desired value. This value is determined by the clock frequency and the speed of the I2C bus.

Write 8’h80 to the control register to enable the I2C Master.

4.3.2 Write to a Slave Device

1. Set the value of the transmit register: Slave address + Write bit. For example, {7’b1000110,1’b0}, of which 7’b1000110 is the Slave address, and the write bit is 1’b0.
2. Set the value of the command register as 8’h90 to enable the START and WRITE commands. This starts the transmission on the IIC bus.
3. Check the Transfer In Progress (TIP) bit of the status register to ensure the command is complete.
4. Set the transmit register with a Slave memory address for the data to be written to the corresponding address memory.
5. Set the command register with 8’h10 to enable a WRITE command to send the Slave memory address.
6. Check the Transfer In Progress (TIP) bit of the status register to ensure that the command is complete.
7. Set the transmit register with 8-bit data for the Slave device to be written to.
8. Set the command register with 8’h10 to enable a WRITE command to send data.
9. Check the Transfer In Progress (TIP) bit of the status register to ensure that the command is complete.
10. Repeat Steps 7 to 9 to continue to send data to the Slave device.
11. Set the transmit register with the last byte of data.
12. Set the command register with 8’h50 to enable a WRITE command to send the last byte of data. Then issue a STOP command.

4.3.3 Read from a Slave Device

1. Set the value of transmit register: Slave address + Write bit.
2. Set the value of the command register as 8’h90 to enable the START and WRITE command. This starts the transmission on the IIC bus.
3. Check the Transfer In Progress (TIP) bit of the Status Register to ensure the command is complete.
4. Set the transmit register with a Slave memory address for the data to be read from.
5. Set the command register with 8’h10 to enable a WRITE command to send the Slave memory address.
6. Check the Transfer In Progress (TIP) bit of the status register to ensure the command is complete.
7. Set the value of transmit register: Slave address + Read bit. For example, {7’b1000110,1’b1}, of which 7’b1000110 is the slave address, and the Read bit is 1’b1.
8. Set the command register with 8’h90 to enable START (repeated START in this case) and WRITE to write the value in the transmit register to the Slave device.
9. Check the Transfer In Progress (TIP) bit of the status register to ensure that the command is complete.
10. Set the command register with 8’h20 to issue a READ command and an acknowledge command to the read data from the Slave device.
11. Check the Transfer In Progress (TIP) bit of the status register to ensure the command is complete.
12. Repeat steps 10 and 11 to continue to read data from the Slave device.
13. When the Master stops reading from the Slave, set the command register with 8’h68. This will allow the Slave device to read the last byte of data and then issue NACK.

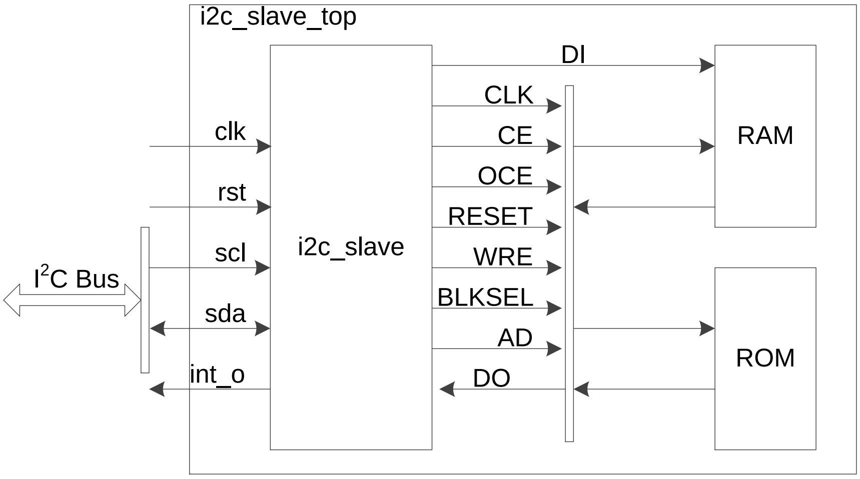
4.4 Gowin I2C Slave Reference Design

The Gowin I2C Slave reference design contains two working modes: RAM and ROM. If you select RAM, the Gowin I2C Slave can store the data sent from the I2C Master or transmit the data that is stored in the RAM to the I2C Master. if you select ROM, the I2C Master can only read the data that is prestored in the ROM. At present, the ROM can store 0~255, 256 data in total.

The Gowin I2C Slave supports interrupt function. If you select RAM, and when RAM is written full or read through, the Gowin I2C Slave will generate the int_o interrupt signal. The Gowin I2C Slave supports two interrupt modes. "0" means write stop, and the written data is read out. "1" means write stop and read stop, and back to the initial state. If you select ROM, the Master reads only, so no writing full or reading through will occur.

The Gowin I2C Slave contains the i2c_Slave_top module and i2c_Slave module. The i2c_Slave_top module is mainly used for the interconnection between i2c_Slave module, RAM module, and ROM module. The i2c_Slave module is mainly used to drive the RAM module or ROM module for data storage or data transmission. For the architecture view of Gowin I2C Slave IP, please refer to Figure 4-8.

Figure 4-2 Gowin I2C Slave Architecture View



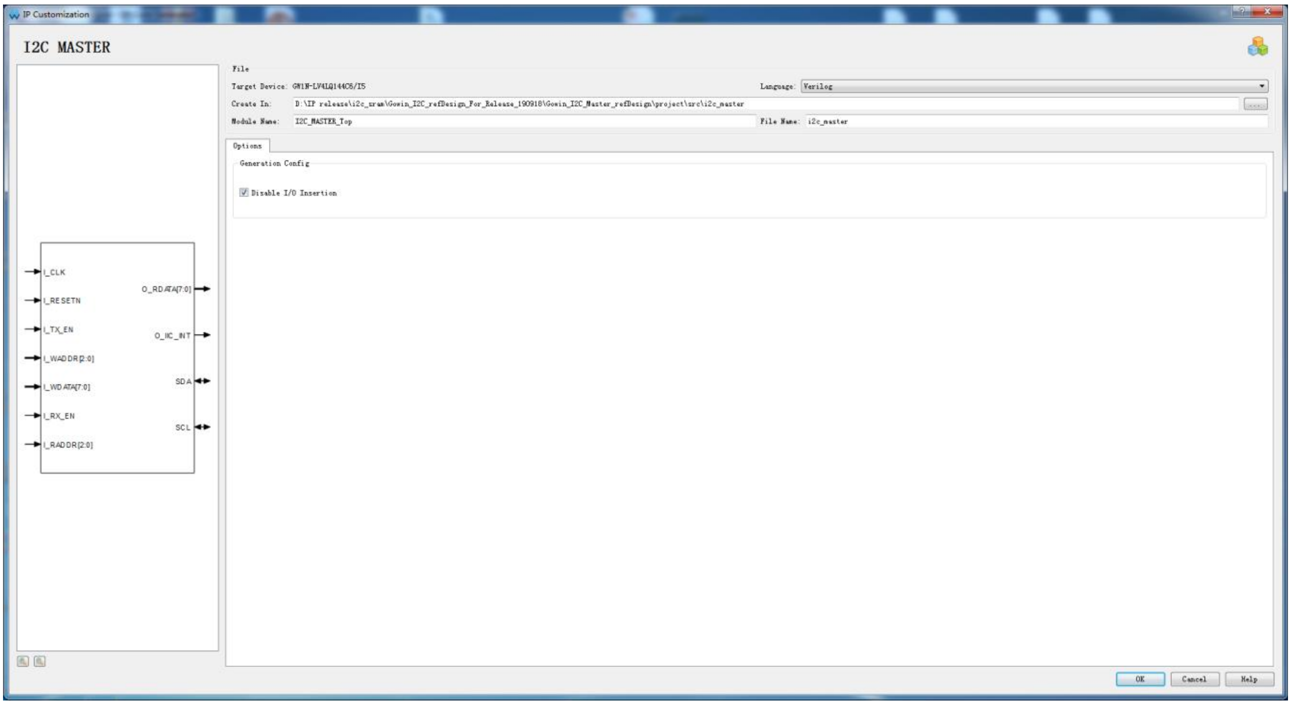
5 Interface Configuration

Users can use the IP core generator tool in the IDE to call and configure Gowin I2C MASTER IP.

5.1 I2C MASTER IP Core Interface

I2C MASTER configuration interface is shown in Figure 5-1.

Figure 5-1 I2C MASTER Congiguration Interface



I2C MASTER

File

Target Device: GW1N-LV4LQ144C6/I5

Language: Verilog

Create In: D:\IP release\i2c_sram\Gowin_I2C_refDesign_For_Release_190918\Gowin_I2C_Master_refDesign\project\src\i2c_master

Module Name: I2C_MASTER_Top

File Name: i2c_master

Options

Generation Config

☑ Disable I/O Insertion

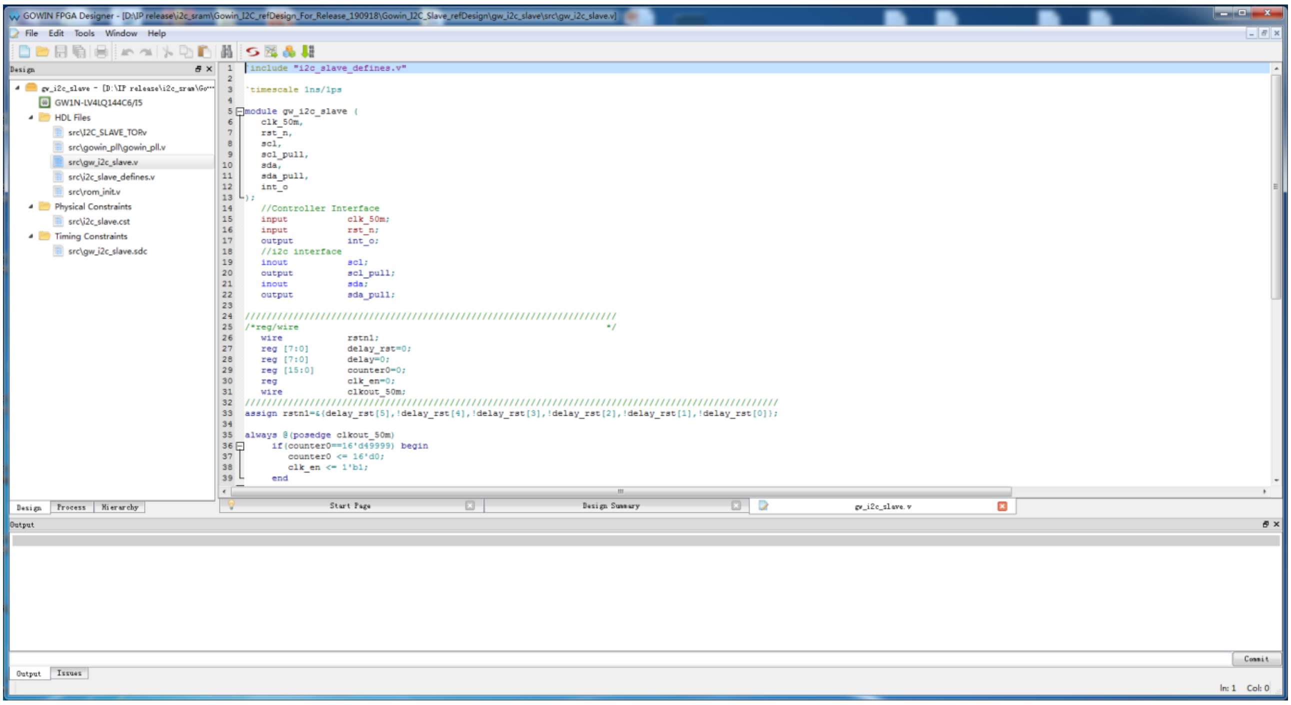
5.2 I2C SLAVE IP Core Interface

Open Gowin YunYuan software, click “File > Open” to open the “Open File” dialog box, select the project file (*.gprj), and then open the project, as shown in Figure 5-2.

Note!

There are three menthods to open the project. Please refer to *SUG100,Gowin YunYuan Software User Guide > 5 Operation > Open an Existing Project*.

Figure 5-2 I2C SLAVE Configuration Interface



```
`include "i2c_slave_defines.v"

`timescale 1ns/1ps

module gw_i2c_slave (
   clk_50m,
   rst_n,
   scl,
   scl_pull,
   sda,
   sda_pull,
   int_o
);
   //Controller Interface
   input           clk_50m;
   input           rst_n;
   output          int_o;
   //i2c interface
   inout           scl;
   output          scl_pull;
   inout           sda;
   output          sda_pull;

//////////////////////////////////////////////////////////////////////
/*reg/wire                                                          */
   wire            rstn1;
   reg [7:0]       delay_rst=0;
   reg [7:0]       delay=0;
   reg [15:0]      counter0=0;
   reg             clk_en=0;
   wire            clkout_50m;
/////////////////////////////////////////////////////////////////////////////////////////////////
assign rstn1=&{delay_rst[5],!delay_rst[4],!delay_rst[3],!delay_rst[2],!delay_rst[1],!delay_rst[0]};

always @(posedge clkout_50m)
     if(counter0==16'd49999) begin
        counter0 <= 16'd0;
        clk_en <= 1'b1;
     end
```

5.3 Bitstream File Generation

Under necessary constraints, bitstream file can be generated after synthesis, plancement and routing. Download the bitstream file to the development board or test board via Gowin USB cable. Observe the communication via the test interface.

6 Reference Design

For more details, please refer to the I2C reference design at Gowin official website.

PROGRAMMING FOR FUTURE