# Gowin I3C SDR QuickStart

**IPUG659-1.1E, 10/25/2018**

**Disclaimer**

**Revision History**

| Date | Version | Description |
|---|---|---|
| 01/29/2018 | 1.0E | Initial version published. |
| 10/25/2018 | 1.1E | The information on the IDE version, etc modified. |

# Contents

# List of Figures

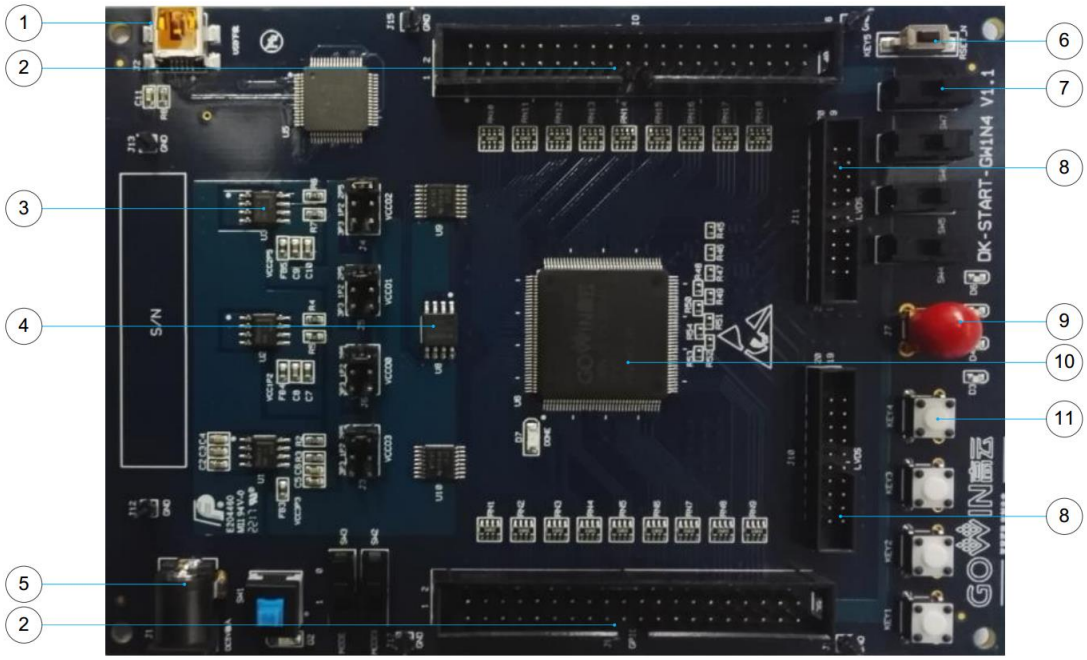
# List of Tables

# 1 Overview

## 1.1 I3C SDR Modules

The development board diagram is presented in Figure 1-1.

**Figure 1-1 Development Board**




① USB Download Port ② GPIO ③ LDO Circuit
④ Series Flash ⑤ DC5V Power Input ⑥ Reset
⑦ Slide Switch ⑧ LVDS ⑨ SMA Clock Input
⑩ FPGA ⑪ Key Switch

The development board provides a USB download interface. The datastream file can be downloaded to the internal SRAM, internal Flash, or external Flash as needed.

# 1.2 Development Board Interconnection

This manual uses two development boards as an example to introduce I3C SDR communication. Two development boards are connected: One is used as the Master, and the other one is used as the Slave. As shown in Figure 1-2 and Figure 1-3. The j10_1, j10_2, j10_5 and j10_6 of the development board 1 are respectively connected with the j10_1, j10_2, j10_5 and j10_6 of the development board 2. During the I3C SDR communication process, users can select from the SDA, SDA_PULL, SCL, and SCL_PULL development pins, as shown in Table 1-1.

**Figure 1-2 Electrical Schematic of Two Board Interconnection**

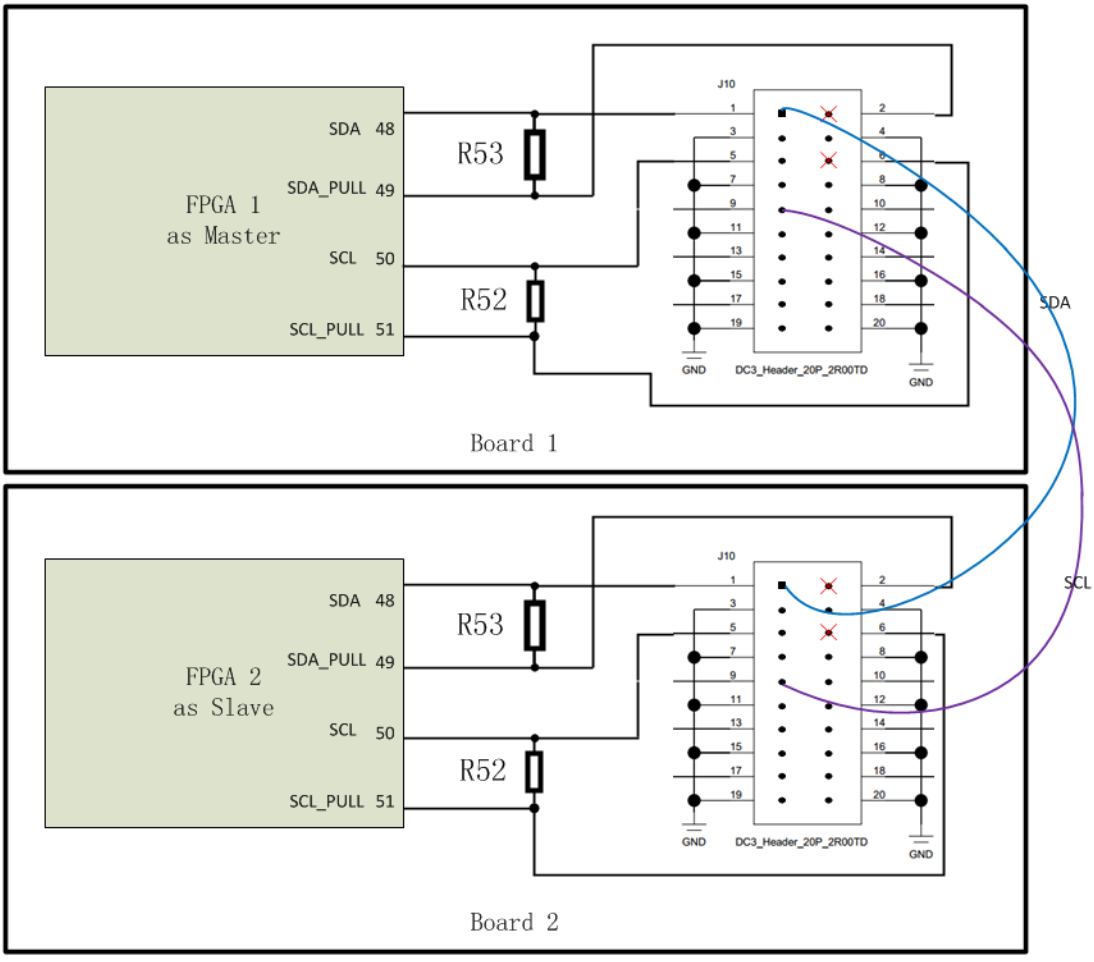


**Figure 1-3 Development Board Physical Interconnection**

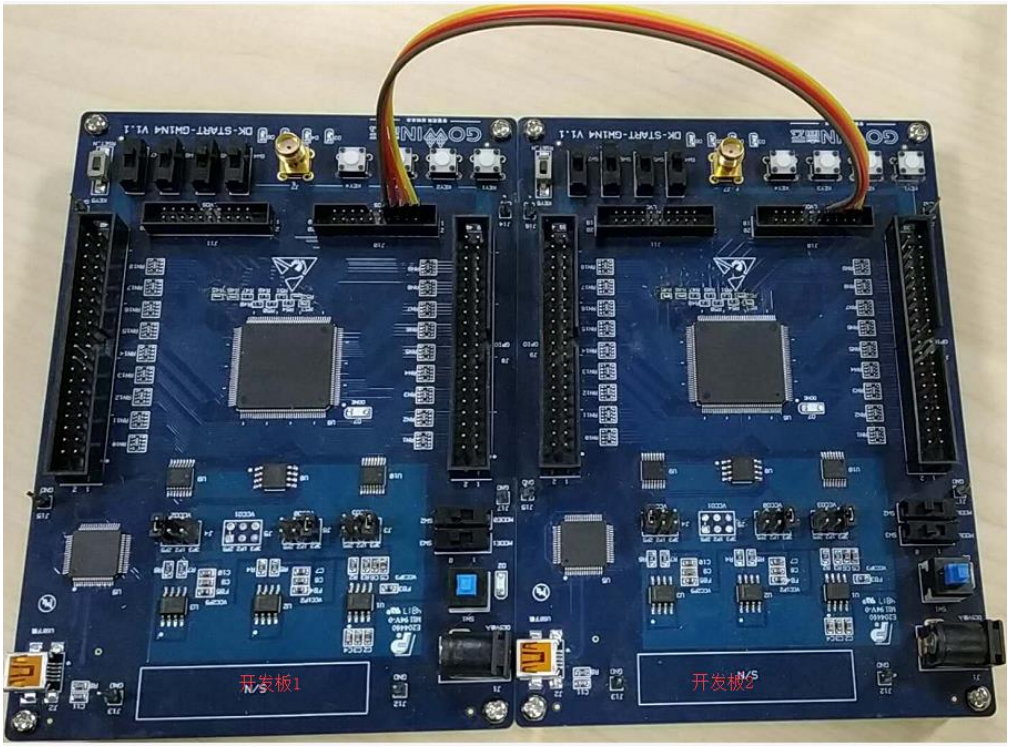



**Table 1-1 GPIO Used for I3C SDR BUS**

| Pin | FPGA IO | Resistance Configuration |
|---|---|---|
| J10_1, J10_2 | 48, 49 | R53 |
| J10_5, J10_6 | 50, 51 | R52 |
| J11_13, J11_14 | 72, 75 | R46 |
| J11_17, J11_18 | 76, 78 | R45 |

# 2Example of I3C SDR Communication Test

## 2.1 Introduction

Take the I3C SDR write-read operation as an example to introduce the I3C SDR communication. Place the sdata, sdata_pull, scl and scl_pull of the I3C SDR system on the j10_1, j10_2, j10_5 and j10_6 pins respectively of the development board, and control the communication between the I3C SDR master and the slave using the key. In the master module, key1, key2, and key3 control the reset signal, CMS (master set signal) and STAS signal. In the slave module, the key1 controls the reset signal.

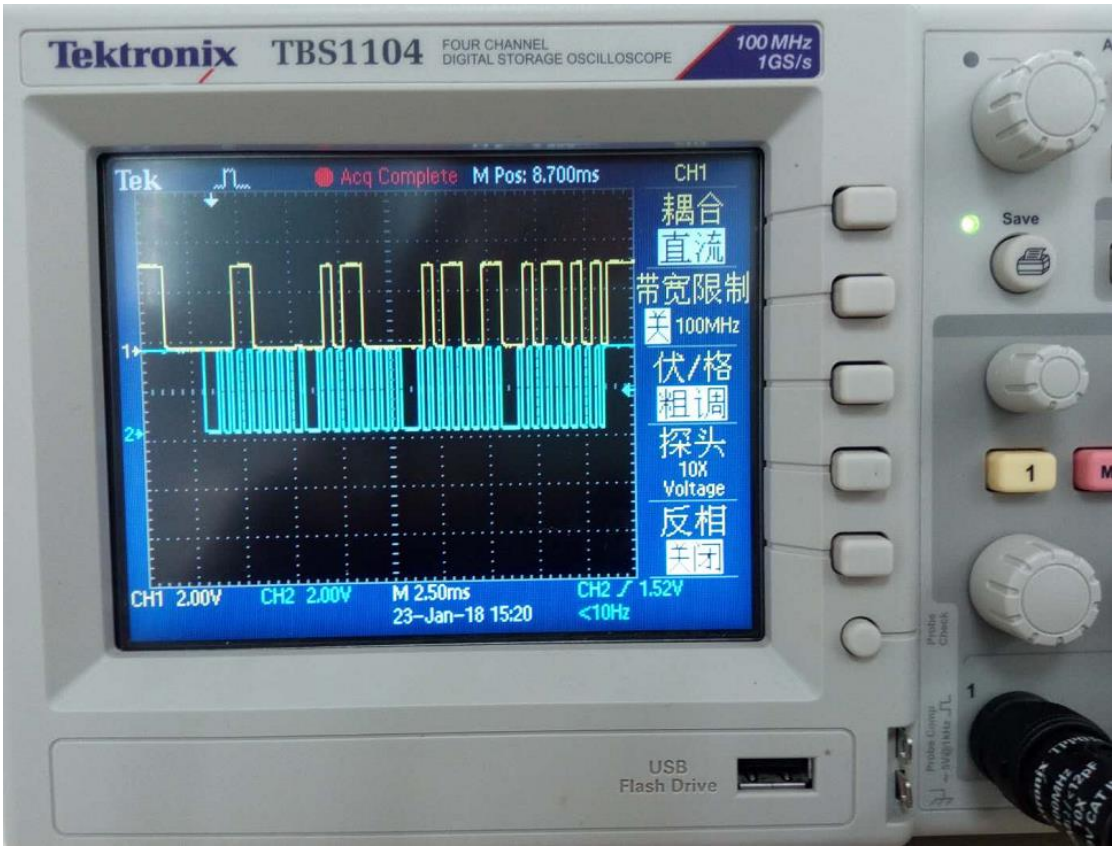
## 2.2 Communication Test

The bitstream file is provided for your reference. Download master.fs to Board 1 as Master and download slave.fs to Board 2 as Slave.

First, download the bit files of the master and slave to the development board, then reset the slave (i.e., press key1 of the slave development board); reset the master board, and press key1 of the master board; set CMS, dynamically configure it as Master (i.e., press key2 of the master development board); set the STAS signal of the master, and press key2 of the master development board. Communication will start automatically. The results obtained through the oscilloscope are shown in Figure 2-1. Channel 1 is an SDA signal and channel 2 is an SCL signal.

To observe the whole waveform on a portable oscilloscope more conveniently, the system clock frequency that is used for testing is 10 kHz, and the frequency of the SCL signal is adjusted to around 2 kHz.

**Figure 2-1 I3C SDR Write-Read Operation**



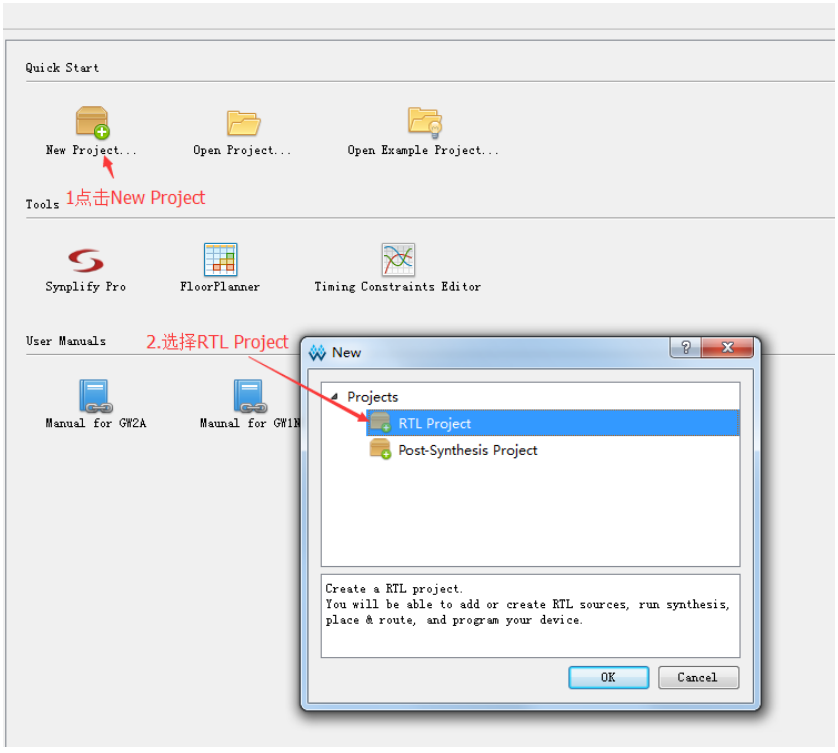
# 3 I3C SDR Application

## 3.1 IP Instantiation, Configuration, and Connection

The I3C SDR module used in this manual is directly generated by the I3C SDR IP module in 1.8.0 IDE. The instantiation and data communication for the I3C SDR module are then conducted in the excitation file of the Master and Slave modules. The specific steps involved in generating the I3C SDR module by IP are as follows:

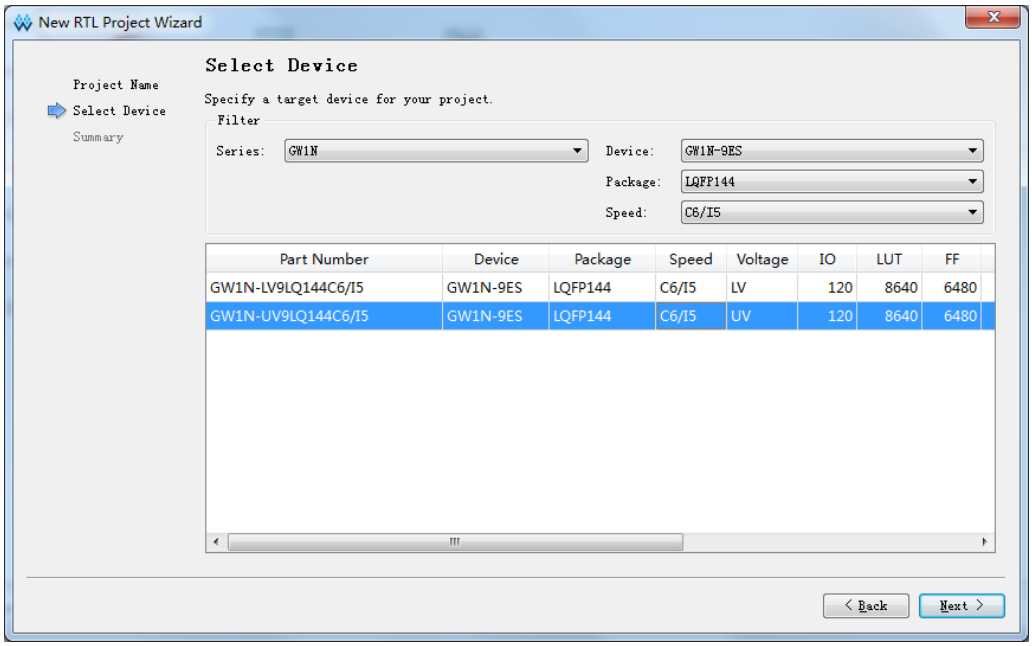
1. Open the IDE software and click the "New Project" option. Select "RTL Project", as shown in Figure 3-1.

**Figure 3-1 Create New Project**



2. Select the device. Select the GW1N-9ES-LQFP144 package in this project, as shown in Figure 3-2.

**Figure 3-2 Select Device**




3. After creating the project, double-click on the IP Core Generator ( ) option (as shown in Figure 3-3), configure the I3C IP module (as shown in Figure 3-4), and click "OK" to generate the I3C module.

**Figure 3-3 IP Core Generator**

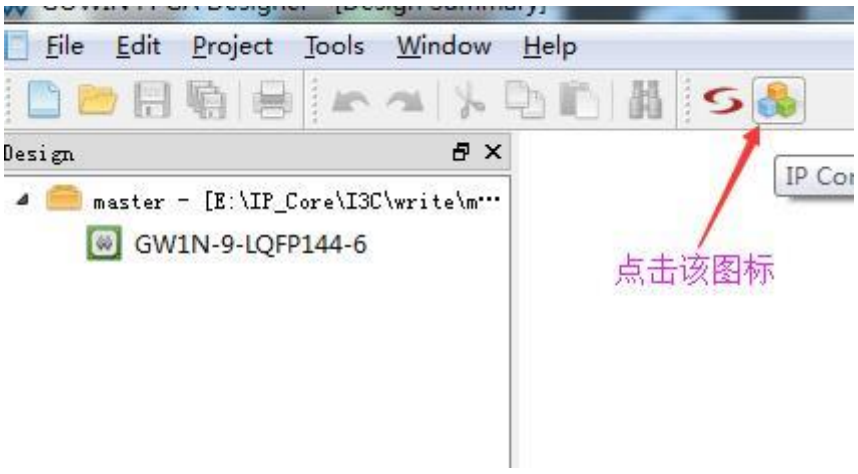
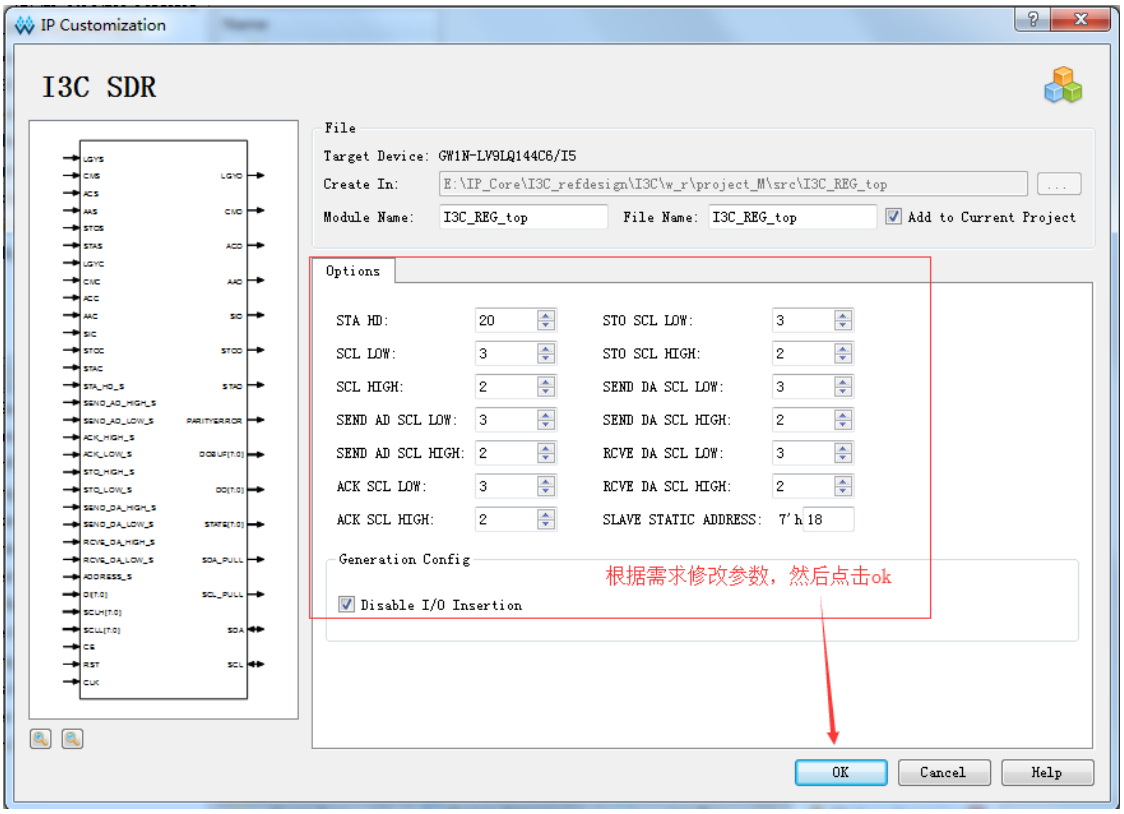
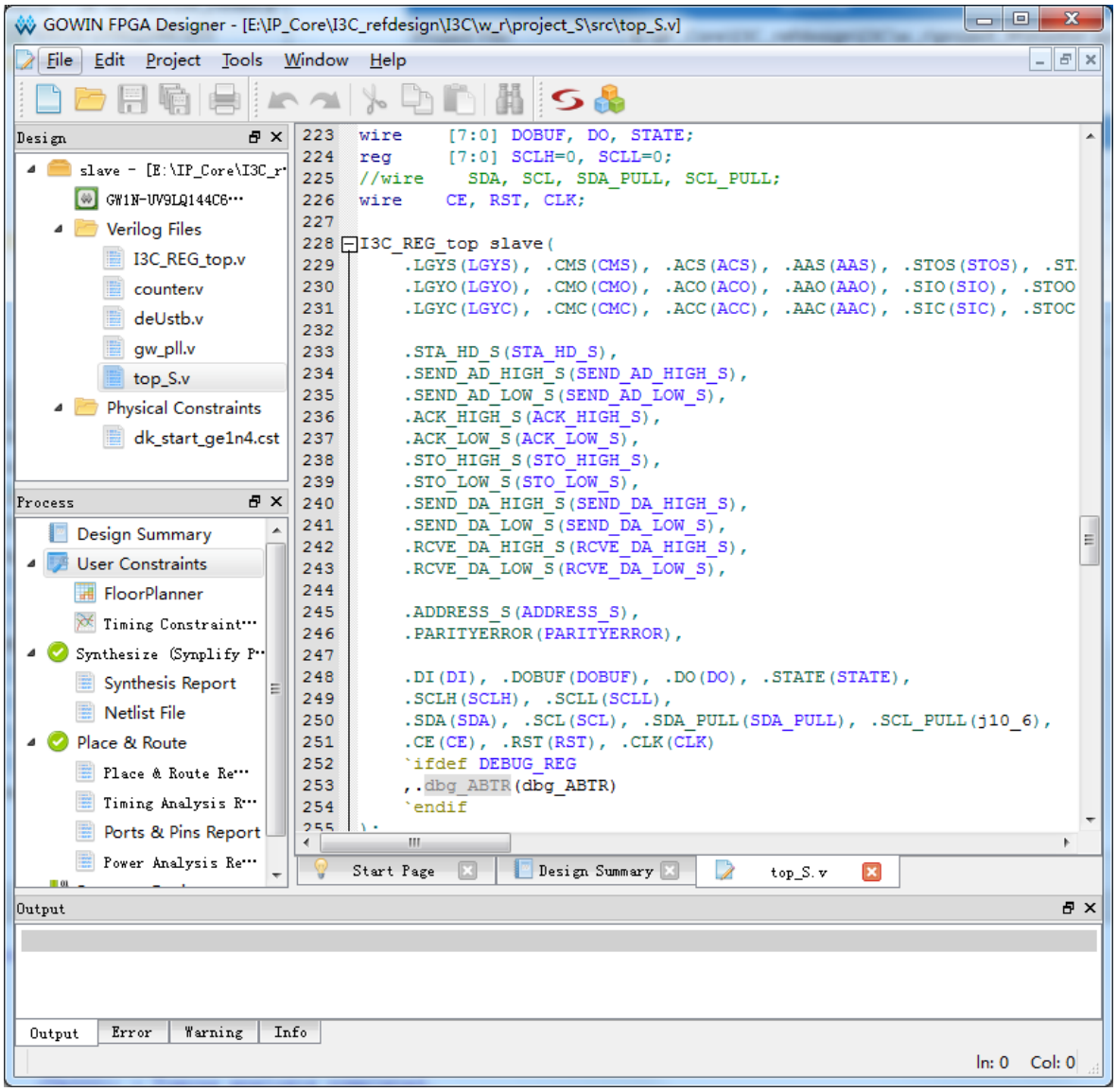


**Figure 3-4 I3C SDR Module Configuration Interface**




4. After generating the I3C SDR module, add the top. V (incentive file), counter. V, deustb.v, and configure the PLL module to the project. Take the Slave, for example, the Slave module configuration of the I3C SDR is introduced, as shown in Figure 3-5.

**Figure 3-5 Slave Project View**




# 3.2 I3C SDR IP Control and Usage

## 3.2.1 Module Overview

module counter (cnto,clko,over,clki,ce): Frequency-dividing counter

module deUstb(out,in,clk): Key jitter elimination module

module GW_PLL (clkout, clkin): PLL instance module, which can be generated by the IP Core Generator.

### 3.2.2 Top Module in the Master Project

All the available IO ports are listed from the top 115 lines on the development board. They can be modified as input output inout as needed.

- Clk_ext signal is constrained to the PCB crystal oscillator;
- The GW_PLL myPll in line 119 is PLL, which is used to change the clock value from 50M to 10M (midClk);
- The counter cnt1 (lowOut lowClk, lowOver, midClk, 1 'b1) on line 127 is used to divide the clock frequency from 10 M to 10 kHz, which provides a convenient method of viewing the signal waveform via the portable oscilloscope;
- Two counters are instantiated in line 143 and line 139 to generate a lower frequency clock;
- 143-191 is a set of localparam definitions that is used to determine the IP status.
- I3C IP and I3C_REG_top master are constrained in line 224.
- A set of elimination modules is constrained in lines 250-259, which is used to eliminate the key jitter;
- A set of variables and a control block are defined in lines 260-404, which controls the I3C IP and core code.
- A set of observed signals are listed in lines 408-440. Users can observe these on a logical analyzer. These signals can be used in projects.

See lines 270-401 for controlling code, lines 278-295 for resetting logic, line 297 for SIO interruption, and line 298 for determining the current state. For line 299, send the address 8'h30 during writing in S_CM_WAIT_AD state, send the address 8'h31 during reading in S_CM_WAIT_AD state.

### 3.2.3 Top Module in Slave Project

Most codes are consistent with the Master code. Only the core control code is different; however, the structure of the two modules is similar. Because this IP adopts a unified design style, the operation is consistent.

The core control block of the Slave is in 270-352 lines. Line 294 is used to judge the state and perform the corresponding operations when the SIO interruption arrives.

### 3.2.4 Constraint File

The GPIO, KEY, LED, and CLK in lines 9-114 are provided on the constraint development board

Note the following lines:

- IO_PORT "j10_1" PULL_MODE=NONE;
- IO_PORT "j10_5" PULL_MODE=NONE;
- IO_PORT "j10_2" PULL_MODE=NONE;
- IO_PORT "j10_6" PULL_MODE=NONE;

If I/O is used as SDA, SDA_PULL, SCL, and SCL_PULL, PULL_MODE needs to set as NONE.

### 3.2.5 Synthesis & Download

After configuring the I3C SDR Master and Slave modules, click "Synthesize" and then "Place&Route" to synthesize, place & route, generate the bitstream file, and then download the Master and Slave bitstream files to the corresponding development board.


GOWIN
PROGRAMMING FOR FUTURE