Gowin I²C_UART IP

User Guide

IPUG925-1.0E, 04/15/2020

Disclaimer

Revision History

Date	Version	Description
04/15/2020	1.0E	Initial version published.

Contents

List of Figures

List of Tables

1 About This Guide

1.1 Purpose

Gowin I^2C_UART IP user guide provides the features and performance, functional description and reference design to help users learn the usage of Gowin I^2C_UART IP.

1.2 Related Documents

The latest user guides are available on the GOWINSEMI Website. You can find related documents at www.gowinsemi.com

1.3 Terminology and Abbreviations

The terminology and abbreviations used in this manual are as shown in Table 1-1.

Table 1-1 Abbreviations and Terminology

Terminology and Abbreviations	Meaning
FPGA	Field Programmable Gate Array
I^2C	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 by the following ways.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

2 Overview

I^2C is a two-wire serial bus that is used to connect the micro-controller with its peripherals.

Universal Asynchronous Receiver/Transmitter (UART) is part of computer hardware that transmits data between serial and parallel communications.

In order to reduce the difficulty of system development and improve the speed of product development, Gowin I^2C_UART IP is designed to realize the function of interface conversion between I^2C and UART.

2.1 Features

- Supports data communication of one I^2C and UART;
- I^2C supports slave mode;
- UART supports 4-wire mode with 1 stop bit, no paity and 8 bit width;
- UART baud rate can be set flexibly by configuring registers;
- Supports interrupt;
- The depth is 64 bytes for UART transmitting FIFO;
- The depth is 64 bytes for UART receiving FIFO;
- The input clock must not be less than 10 times the input I^2C data rate;
- Can be synthesized;
- The language is Verilog.

2.2 Performance

The frequency of Gowin I^2C_UART IP depends on the clock frequency provided by the I^2C Host and the maximum frequency supported by the IP on the selected chip.

2.3 Resource Utilization

Taking GW2A-LV18PG256C7/I6 as an example, the resource utilization is shown in Table 2-1. For application verification on other Gowin FPGAs, please pay attention to the post-release information.

Table 2-1 Resource Utilization

Part Number	Language	LUT4	REG
GW2A-LV18PG256C7/I6	Verilog	638	270

3 Signal Description

3.1 System Signal

The definition of system signal is as shown in Table 3-1.

Table 3-1 System Signal

No.	Name	I/O	Description	Remarks
1	I_rst_n	I	Reset	The I/O of all the signals takes IP as reference.
2	I_clk	I	Clock	

3.2 I²C Signal

The definition of I^2C signal is as shown in Table 3-2.

Table 3-2 I²C Signal

No.	Name	I/O	Description	Remarks
1	I_I2C_scl	I	I^2C clock	–
2	IO_I2C_sda	I/O	I^2C data	
3	O_I2C_irq_n	O	Interrupt request signal	

3.3 UART Signal

The definition of UART signal is as shown in Table 3-3.

Table 3-3 UART Signal

No.	Name	I/O	Description	Remarks
1	I_uart_rxd	I	Input serial data	The I/O of all the signals takes IP as reference.
2	O_uart_txd	O	Output serial data	
3	O_uart_rts_n	O	Allow to transmit signal	
4	I_uart_cts_n	I	Opposite port ready signal	

4 Parameter Configuration

The parameter configuration of Gowin I^2C_UART IP is as shown in Table 4-1.

Table 4-1 GUI Paramrter

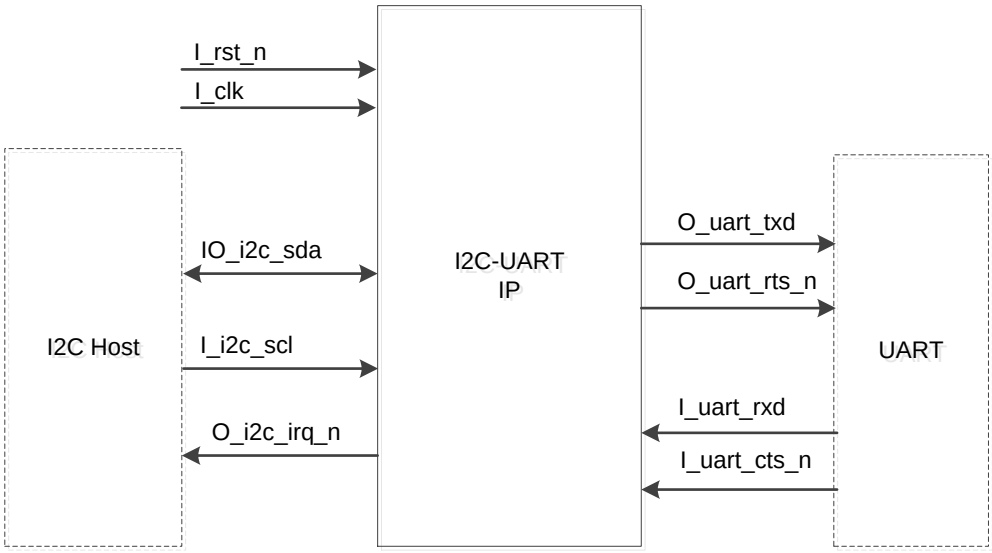
No.	Name	Range	Default Value	Description
1	I2C ADDR	7'h01~7'h7F	7'h01	I^2C slave address

5 Functional Description

5.1 System Block Diagram

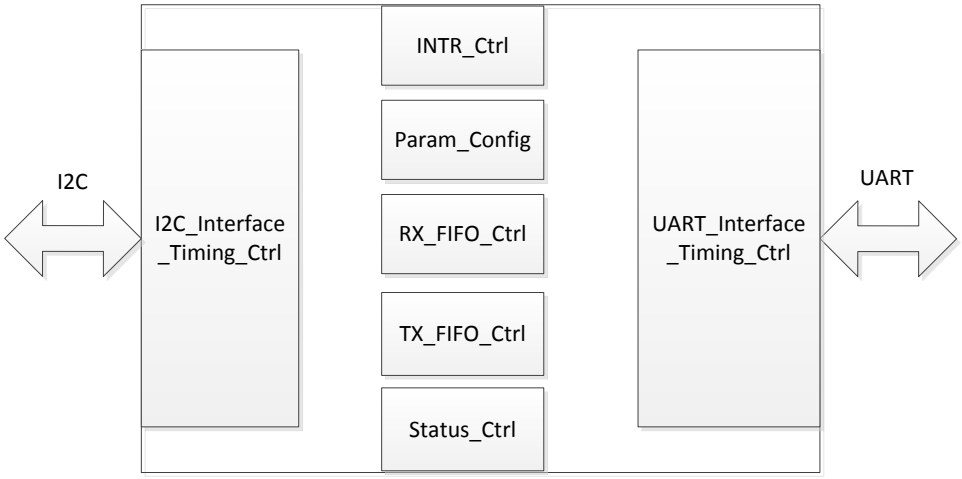
I^{2}C_UART IP is between I^{2}C host and UART. It realizes data communication from I^{2}C to UART. I^{2}C host includes data, clock, interrupt signal. UART includes receiving data, transmitting data and flow control signal. I^{2}C_UART includes reset and clock signals. The system block diagram is shown in Figure 5-1.

Figure 5-1 System Block Diagram



5.2 Architecture

Figure 5-2 Architecture



The IP design includes I^{2}C_interface_Timing_Ctrl, INTR_Ctrl, Param_Config, RX_FIFO_Ctrl, TX_FIFO_Ctrl, Status_Ctrl, UART_Interface_Timing_Ctrl modules.

- I^{2}C_interface_Timing_Ctrl: Realizes data communication with I^{2}C host and completes I^{2}C interface timing control and protocol parse;
- vINTR_Ctrl: Implement interrupt control function according to IP internal status;
- Param_Config: Receives the configuration parameters from the I^{2}C host and delivers them to the corresponding module.
- RX_FIFO_Ctrl: Implements data buffer received from UART;
- TX_FIFO_Ctrl: Implements data buffer from I^{2}C host;
- Status_Ctrl: Detect and count the internal interface status and working status of IP;
- UART_Interface_Timing_Ctrl: Implements data communication with UART and completes UART interface timing control and protocol parse.

5.3 Registers

The definition of Gowin I^{2}C_UART IP registers is as shown in Table 5-1.

Table 5-1 Gowin I²C_UART IP Registers

No.	Name	Address（4 bits）	Default Value（8 bits）	Type	Description
1	RHR	0x0	-	RO	Receive holding register
2	THR	0x0	-	WO	Transmit holding register
3	IER	0x1	0x00	R/W	Interrupt Enable Register
4	IIR	0x2	0x01	RO	Interrupt Identification Register

No.	Name	Address（4 bits）	Default Value（8 bits）	Type	Description
5	LSR	0x3	0x60	RO	Line Status Register
6	MSR	0x4	0x10	RO	Modem Status Register
7	MCR	0x5	0x00	R/W	Modem Control Register
8	FCR	0x6	0x00	R/W	FIFO Control Register
9	TXLVL	0x7	0x00	RO	Number of valid data of transmitting FIFO
10	RXLVL	0x8	0x00	RO	Number of valid data of receiving FIFO
11	DLL	0x9	0x0F	R/W	UART frequency division register (lower 8 bits)
12	DLH	0xA	0x00	R/W	UART frequency division register (upper 8 bits)

5.3.1 RHR Register

The definition of RHR register is shown in Table 5-2.

Table 5-2 RHR Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x0	Bit[7:0]	-	RO	Buffer data receiving from UART, 64 bytes depth.

5.3.2 THR Register

The definition of THR register is shown in Table 5-3.

Table 5-3 THR Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x0	Bit[7:0]	-	WO	Buffer data transmitting to UART, 64 bytes depth.

5.3.3 IER Register

The definition of IER register is shown in Table 5-4.

Table 5-4 IER Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x1	Bit[7:4]	-	-	Reserved
	Bit[3]	0	WO	Modem status interrupt 0: Turning off; 1: enable
	Bit[2]	0	WO	Receive line status interrupt. 0: Turning off; 1: enable
	Bit[1]	0	WO	Transmit holding register interrupt. 0: Turning off; 1: enable
	Bit[0]	0	WO	Receive holding register interrupt. 0:Turning off; 1: enable

5.3.4 IIR Register

The definition of IIR register is shown in Table 5-5.

Table 5-5 IIR Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x2	Bit[7:3]	-	-	Reserved

Address（4 bits）	Bit(s)	Default Value	Type	Description
	Bit[2:1]	00	RO	Interrupt Type: 11: Receive line status error interrupt (highest priority) 10: Receive holding register interrupt (second priority) 01: Transmit holding register interrupt (third priority) 00: Modem status interrupt (lowest priority)
	Bit[0]	1	RO	Interrupt status 0: Interrupt waiting; 1: No interruption waiting

5.3.5 LSR Register

The definition of LSR register is shown in Table 5-6.

Table 5-6 LSR Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x3	Bit[7]	-	-	Reserved
	Bit[6]	1	RO	THR/TSR data status 0: THR or TSR has data to transmit; 1: Both THR and TSR are null
	Bit[5]	1	RO	THR data status 0: THR has data to transmit 1: THR is null
	Bit[4]	0	RO	Communication interrupt 0: No communication interrupt 1: Communication interrupt detected (data transmission at low, including start bit, data bit and stop bit), clear after read
	Bit[3]	0	RO	Frame error; 0: No frame error; 1: frame error detected (stop bit missed), clear after read
	Bit[2]	-	-	Reserved
	Bit[1]	0	RO	Overrun error; 0: No overrun error; 1: RHR register data overrun, clear after read
	Bit[0]	0	RO	RHR data status 0: No data 1: At least one data is stored in RHR

5.3.6 MSR Register

The definition of MSR register is shown in Table 5-7.

Table 5-7 MSR Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x4	Bit[7:5]	-	-	Reserved
	Bit[4]	X	RO	Allow to transmit（active-high）
	Bit[3:1]	-	-	Reserved
	Bit[0]	0	RO	CTS signal status indicator (active-high), clear after read

Note!

X means the signal value is determined by the status of the input signal.

5.3.7 MCR Register

The definition of MCR register is shown in Table 5-8.

Table 5-8 MCR Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x5	Bit[7:3]	-	-	Reserved
	Bit[2]	0	R/W	Receiving and transmitting loopback 0: Normal 1: Loopback on
	Bit[1]	0	R/W	Requset to transmit 0: Drive RTS high; 1: Drive RTS Low
	Bit[0]	-	-	Reserved

5.3.8 FCR Register

The definition of FCR register is shown in Table 5-9.

Table 5-9 FCR Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x6	Bit[7:6]	00	R/W	Receive holding register interrupt trigger (triggered when the number of data in FIFO is greater than or equal to the following values) 00: 8 bytes; 01: 16 bytes; 10: 32 bytes; 11: 48 bytes;
	Bit[5:4]	00	R/W	Transmit holding register interrupt trigger (triggered when the free space in the FIFO is greater than or equal to the following values): 00: 8 bytes; 01: 16 bytes; 10: 32 bytes; 11: 48 bytes;
	Bit[3]	-	-	Reserved
	Bit[3]	0	R/W	Reset and transmit FIFO. 0: No reset and transmit FIFO; 1: Reset and transmit FIFO
	Bit[1]	0	R/W	Reset and receive FIFO. 0: No reset and receive FIFO; 1: Reset and receive FIFO
	Bit[0]	0	R/W	FIFO enable 0: Disable and receive FIFO;

Address（4 bits）	Bit(s)	Default Value	Type	Description
				1: Enable and receive FIFO;

5.3.9 TXLVL Register

The definition of TXLVL register is shown in Table 5-10.

Table 5-10 TXLVL Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x7	Bit[7]	-	-	Reserved
	Bit[6:0]	0x00	RO	Number of valid data of TX FIFO Range: 0 ~ 64

5.3.10 RXLVL Register

The definition of RXLVL register is shown in Table 5-11.

Table 5-11 RXLVL Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x8	Bit[7]	-	-	Reserved
	Bit[6:0]	0x00	RO	Number of valid data of RX FIFO Range: 0 ~ 64

5.3.11 DLL Register

The definition of DLL register is shown in Table 5-12.

Table 5-12 DLL Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0x9	Bit[7:0]	0x0F	R/W	UART frequency division register (lower 8 bits)

5.3.12 DLH Register

The definition of DLH register is shown in Table 5-13.

Table 5-13 DLH Register

Address（4 bits）	Bit(s)	Default Value	Type	Description
0xA	Bit[7:0]	0x00	R/W	UART frequency division register (upper 8 bits)

Note!

UART baud rate = input clock frequency/frequency division register value (DLH+DLL), and frequency division register value is not less than 10.

5.4 Operation Flow

5.4.1 Initialization Flow

1. After power on, the I^2C host needs to configure the IP parameters, the recommended sequence is as follows:
   a). MCR
   b). FCR
   c). DLL
   d). DLH
   e). IER

2. I^2C host can communicate with UART according to read/write operation flow.

5.4.2 Write Operation Flow

Disable Interrupt Write Operation

When transmit holding register interrupt is disabled, the write operation flow from I^2C host is as follows:

1. The I^2C host reads the current FIFO buffer status by the TXLVL register.
2. When the transmit FIFO has enough space, the I^2C host can write data through the THR register.

Enable Interrupt Write Operation

When transmit holding register interrupt is enabled, the write operation flow from I^2C host is as follows:

1. After receiving the interrupt signal from the I^2C slave, the I^2C host reads the current interrupt from the IIR register.
2. If the interrupt is from transmit holding register, the I^2C host can write data to the THR register.

5.4.3 Read Operation Flow

Disable Interrupt Read Operation

When receive holding register interrupt is disabled, the read operation flow from I^2C host is as follows:

1. The I^2C host reads the current FIFO buffer status the RXLVL register.
2. When the receive FIFO has enough space, the I^2C host can write data through RHR register.

Enable Interrupt Read Operation

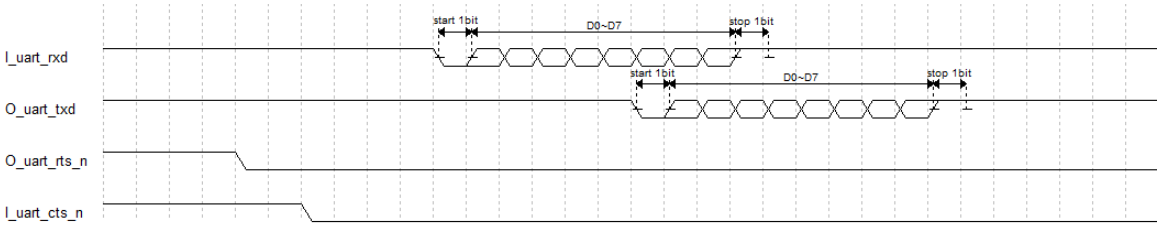
When receive holding register interrupt is enabled, the read operation flow from I^2C host is as follows:

1. After receiving the interrupt signal from the I^2C slave, the I^2C host reads the current interrupt from the IIR register.
2. If the interrupt is from receive holding register, the I^2C host can read data from RHR register.

5.5 Interface Timing

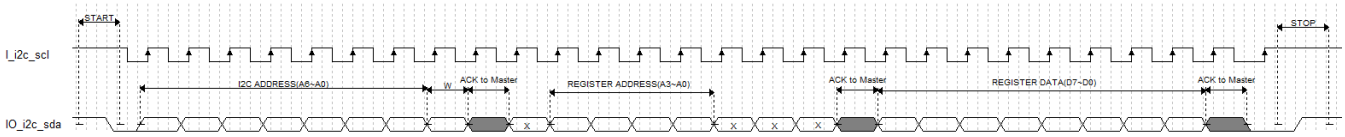
5.5.1 UART Interface Timing

Figure 5-3 UART Interface Timing



5.5.2 I²C Interface Register Write Timing

Figure 5-4 I²C Interface Register Write Timing

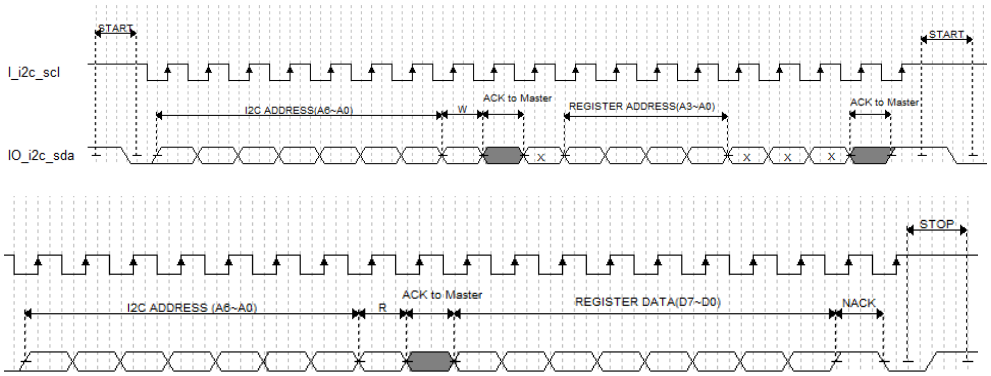


Note!

- W: Write operation; 0: Valid;
- X: Don not care about the bit;
- A3~A0: Register address, 4 bits.
- The input clock must not be less than 10 times the input I2C data rate.

5.5.3 I²C Interface Register Read Timing

Figure 5-5 I²C Interface Register Read Timing

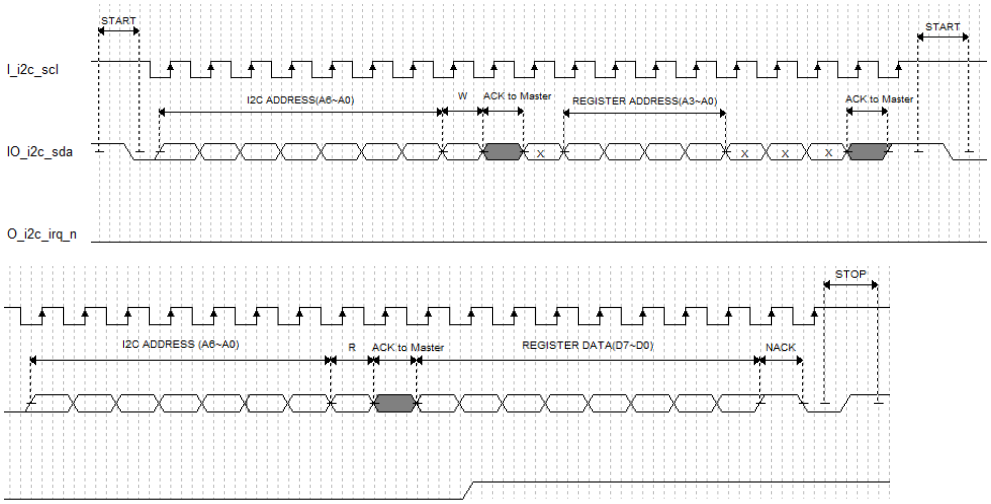


Note!

- W: Write operation; 0: Valid;
- R: Read operation; 1: Valid
- X: Don not care about the bit;
- A3~A0: Register address, 4 bits.

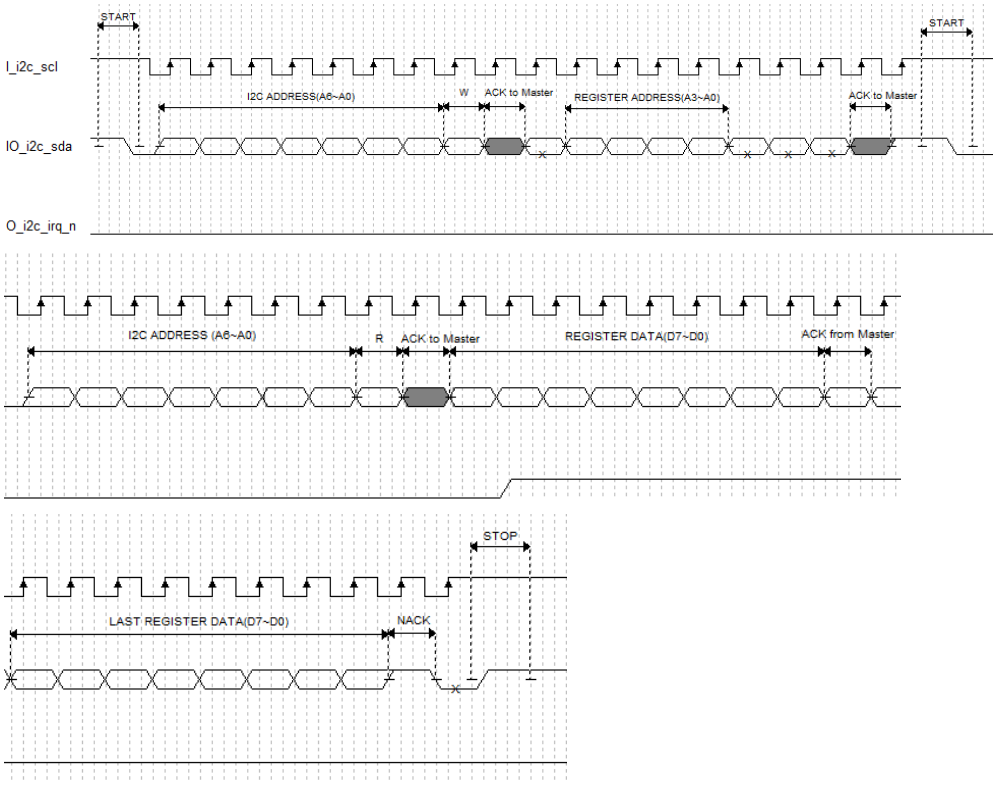
5.5.4 I²C Interface Read LSR/MSR Interrupt Cleared Timing

Figure 5-6 I²C Interface Read LSR/MSR Interrupt Cleared Timing



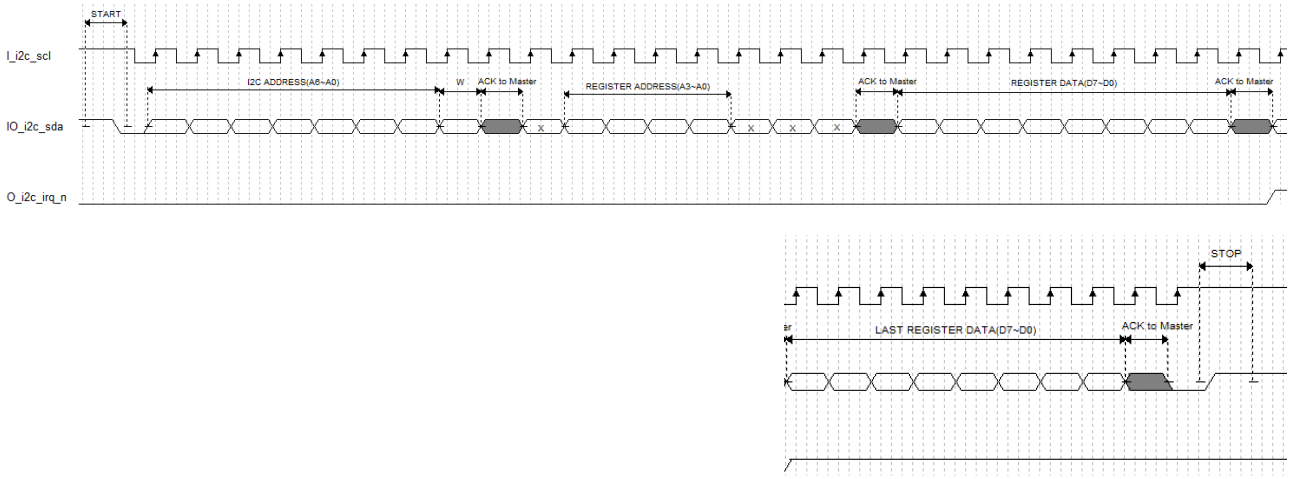
5.5.5 I²C Interface Read RHR Interrupt Cleared Timing

Figure 5-7 I²C Interface Read RHR Interrupt Cleared Timing



5.5.6 I²C Interface Write THR Interrupt Cleared Timing

Figure 5-8 I²C Interface Write THR Interrupt Cleared Timing

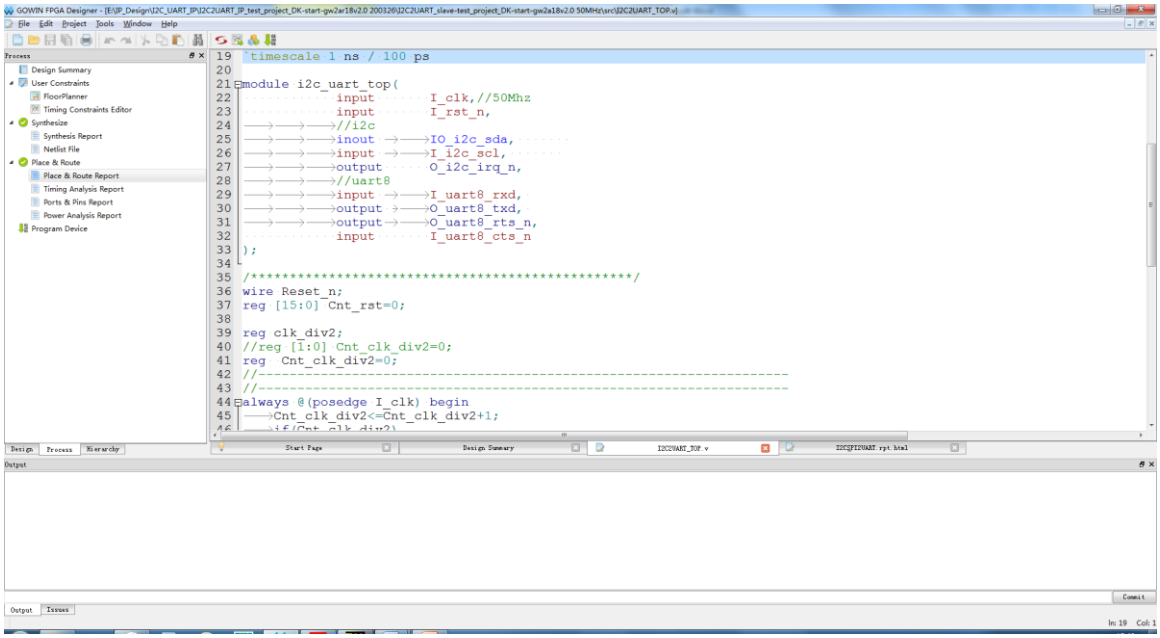


6 Interface Configuration

In "Tools" menu bar of the Gowin software interface, it can start the IP Core Generator tool and complete IP call and configuration.

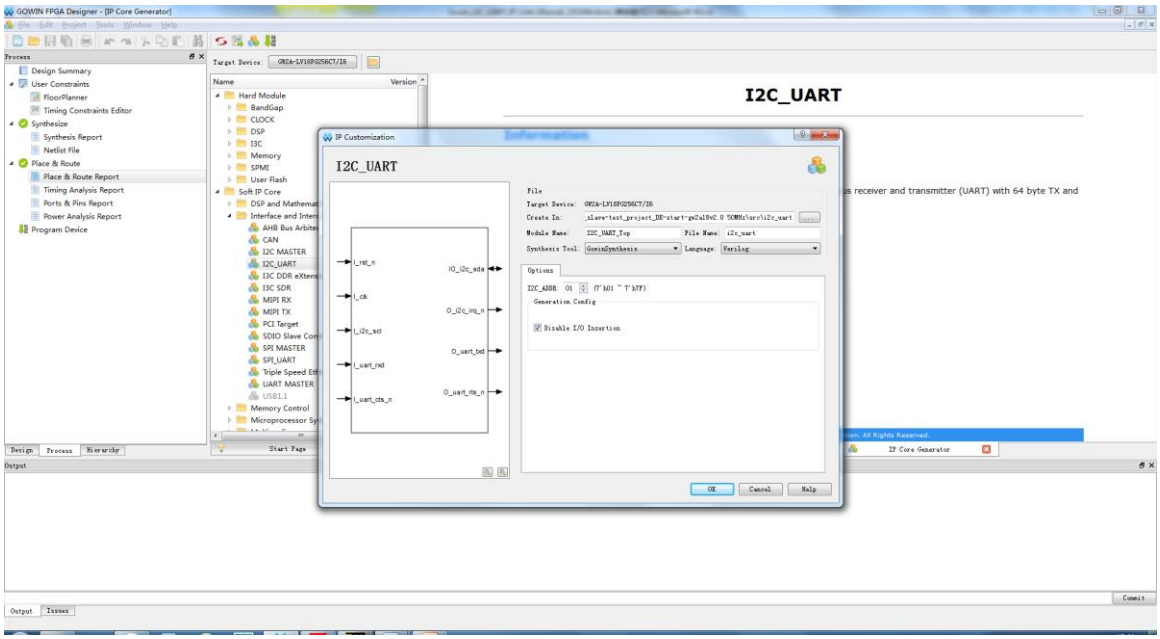
1. Start Gowin Software and open a project, as shown in Figure 6-1;

Figure 6-1 Open an Existing Project



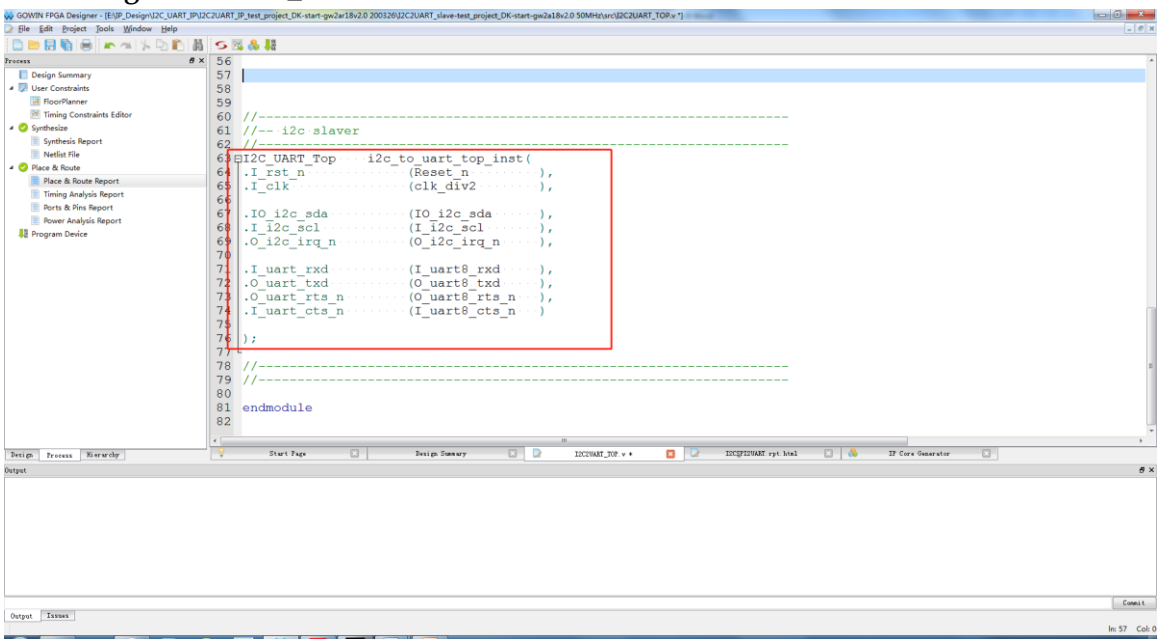
2. Select "Tools > IP Core Generator > I2C_UART", then click "OK" to generate I2C_UART Module, as shown in Figure 6-2.

Figure 6-2 I2C_UART Configuration Interface



3. Instantiate the "I2C_UART" module as shown in Figure 6--3.

Figure 6--3 I2C_UART Instantiation



Synthesize, P&R, download bitstream file to FPGA chip to implement Gowin I^2C_UART IP function.

7 Reference Design

See Gowin I^2C_UART reference design for details at Gowinsemi website.


GOWIN
PROGRAMMING FOR THE FUTURE