



Gowin UART to Bus IP

User Guide

IPUG1022-1.1E, 08/18/2023

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 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
06/30/2023	1.0E	Initial version published.
08/18/2023	1.1E	8 bits and 32 bits options added in bus address width, and 64 bits option added in bus data width.

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.....	3
1.4 Support and Feedback	3
2 Function.....	4
2.1 Overview.....	4
2.2 Features.....	5
2.3 Resource Utilization.....	5
3 Functional Description	6
3.1 Structure	6
3.2 Bus Protocol	8
3.3 Bus Timing	9
4 Signal Definition	13
4.1 Signal Definition.....	13
4.2 Parameter Configuration Options	15
5 Interface Configuration	17
6 Reference Design	21
6.1 UART to Bus Reference Design	21
6.2 On-board Test.....	21
7 File Delivery.....	23
7.1 Documentation.....	23

7.2 Design Source Code (Encryption)	23
7.3 Reference Design	23

List of Figures

Figure 3-1 UART to Bus Block Diagram	7
Figure 3-2 APB Write Timing.....	10
Figure 3-3 Example Timing	10
Figure 3-4 APB Read Timing	10
Figure 3-5 Example Timing	11
Figure 3-6 Local Write Timing	11
Figure 3-7 Example Timing	11
Figure 3-8 Local Read Timing.....	12
Figure 3-9 Example Timing	12
Figure 4-1 Port Diagram in APB Mode.....	13
Figure 4-2 Port Diagram in Local Mode	14
Figure 5-1 IP Core Generator	17
Figure 5-2 UART to Bus IP Core.....	18
Figure 5-3 UART to Bus Configuration Interface (1).....	19
Figure 5-4 UART to Bus Configuration Interface (2).....	20
Figure 6-1 UART to Bus Reference Design Block Diagram	21
Figure 6-2 Test Process	22

List of Tables

Table 1-1 Terminology and Abbreviations	3
Table 2-1 Gowin UART to Bus IP Overview.....	4
Table 2-2 Gowin UART IP Resource Utilization.....	5
Table 3-1 Write Operation Instruction Format	8
Table 3-2 Read Request Instruction Format	9
Table 3-3 Read Response Instruction Format	9
Table 4-1 Signal Definition	14
Table 4-2 Configuration Options	15
Table 7-1 Documents List	23
Table 7-2 Gowin UART to Bus IP Design Source Code List.....	23
Table 7-3 Gowin UART to Bus IP RefDesign Folder Content List	24

1 About This Guide

1.1 Purpose

Gowin UART to Bus IP User Guide is intended to help you learn the features and usage of Gowin UART to Bus IP by providing descriptions of functions, signals, and interface configuration, etc. The software screenshots in this manual are based on 1.9.9 Beta-3. 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](#)
- [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](#)
- [DS1103, GW5A series of FPGA Products Data Sheet](#)
- [DS1104, GW5AST series of FPGA Products Data Sheet](#)

- [SUG100, Gowin Software User 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
APB	Advanced Peripheral Bus
IP	Intellectual Property
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

2.1 Overview

The core module of Gowin UART to Bus IP is an instruction parser that can be used to access the internal bus through the UART interface. This IP core can be used during initial board debugging or as a permanent solution when a high-speed interface is not needed.

The core implements a basic UART transmit and receive block that shares a common baud rate generator and an instruction parser. The parser supports the instructions input in text mode, which are designed to work with hyperterminal software and can access the internal bus.

Table 2-1 Gowin UART to Bus IP Overview

Gowin UART to Bus IP	
Logic Resource	See Table 2-2.
Delivered Doc.	
Design Files	Verilog (encrypted)
Reference Design	Verilog
TestBench	Verilog
Test and Design Flow	
Synthesis Software	GowinSynthesis
Application Software	Gowin Software (V1.9.9.Beta-2 and above)

Note!

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

2.2 Features

The features of Gowin UART to Bus IP are as follows:

- Up to 16 slave buses can be mounted
- APB bus or Local bus can be selected
- Bus address width and data width are configurable
- Instruction parser included
- Baud rate can be set
- Directly interfacing and debugging with Gowin JESD204B IP, Gowin CPRI IP and other IPs that use APB bus as internal bus interface

2.3 Resource Utilization

Gowin UART to Bus IP can be implemented by Verilog. Its performance and resource utilization may vary when the design is employed in different devices, or at different density, speed, or grade. Using Gowin GW5AT series of FPGA as an instance, the resource utilization of Gowin UART to Bus IP with two slaves, address bit width of 16 and data bit width of 32, is as shown in Table 2-2.

Table 2-2 Gowin UART IP Resource Utilization

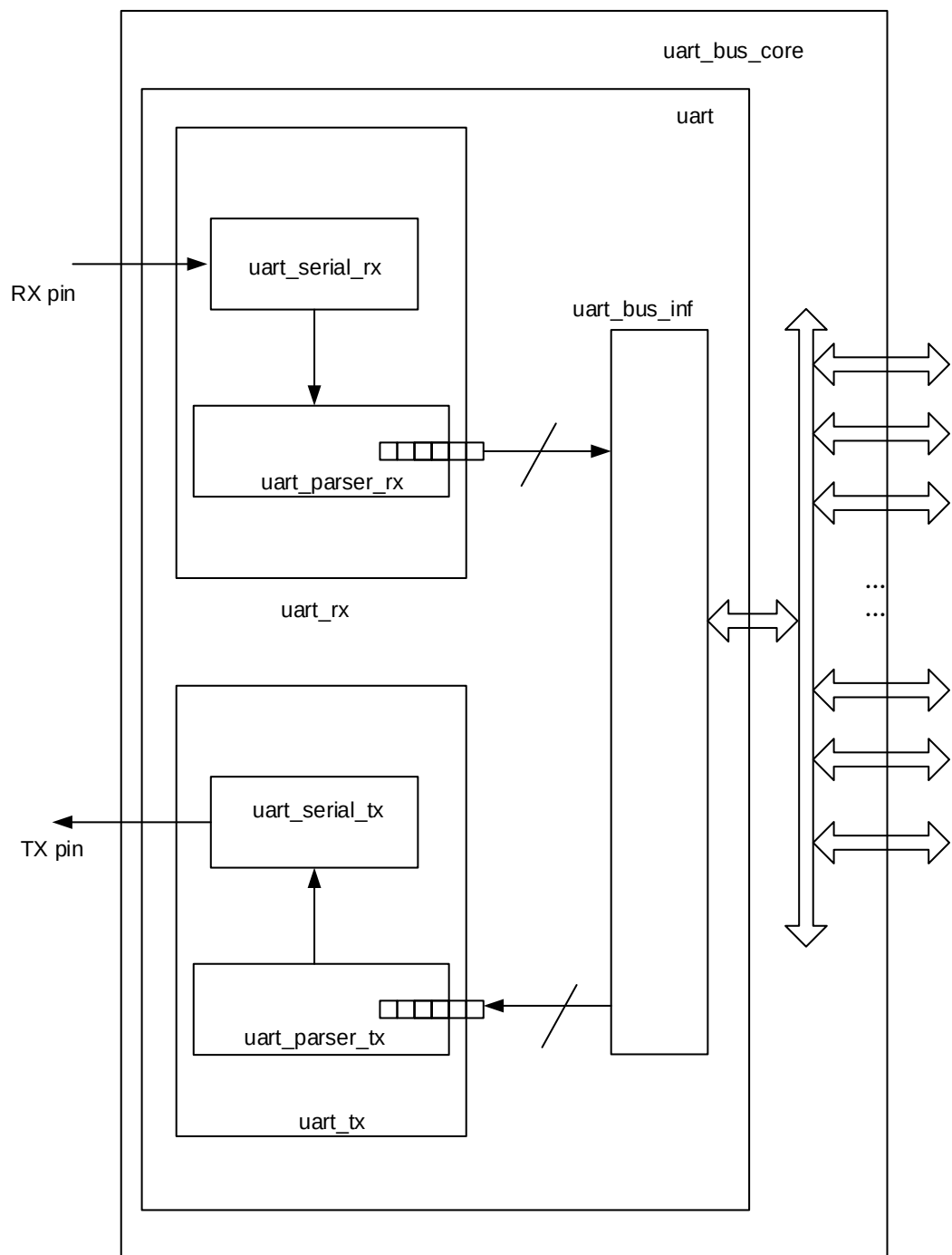
Register	LUT	ALU	DSP	BSRAM	SSRAM
1439	995	39	0	0	0

3 Functional Description

3.1 Structure

Gowin UART to Bus IP includes UART interface receive and transmit modules and an instruction parser, as shown in Figure 3-1.

Figure 3-1 UART to Bus Block Diagram



- **uart_serial_rx:** Serial receive module; convert the received bits into bytes
- **uart_parser_rx:** Frame recognition of the received bytes, and parse the frame instruction (identify r/w, address/data)
- **uart_parser_tx:** The data to be sent (address/data) is framed according to the custom protocol and transmit byte by byte

- uart_serial_tx: Serial transmit module; transmit bytes bit by bit
- uart_bus_inf: Bus timing simulation module

3.2 Bus Protocol

The protocol consists of three instructions: write operation instruction, read request instruction, and read response instruction. All values are in HEX format. The parser checks the upper and lower characters, and detects spaces (0x20) and tabs (0x09) as whitespace and LF (0x0A) and CR (0x0D) as end-of-line. Instructions that do not follow the required sequence or contain illegal characters will be aborted.

The serial bus protocol conventions are as follows (Take address width of 16 bits and data width of 32 bits as an example):

Write bus "W AAAA BBBBBBBB" + CR or LF

Read bus "R AAAA" + CR or LF

The read bus will return the following instruction "G AAAA BBBBBBBB" + CR+ LF

W or w means write, R or r means read, and G means read result return. They are separated by one or more spaces, and you need to enter a carriage return or a line feed at the end.

During write bus operation, AAAA means address in hexadecimal, 8/16/32 bits, case insensitive; BBBBBBBB means data in hexadecimal, 32/64 bits, case insensitive.

Table 3-1 Write Operation Instruction Format

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
'W' or 'w'	Space (at least one)	Address (hexadecimal), e.g. "1A00"	Space (at least one)	Data (hexadecimal), e.g. "1234ABCD"	End of line: CR or LF character

The address and data fields are case-insensitive, and the higher bits are 0, or they can be omitted, for example

W AB A1B1

w aB A1b1

W 00AB 0000A1B1

These three write instructions are equivalent.

During read bus operation, AAAA means address in hexadecimal, 8/16/32 bits, case insensitive.

Table 3-2 Read Request Instruction Format

Field 1	Field 2	Field 3	Field 4
'R' or 'r'	Space (at least one)	Address (hexadecimal)" e.g. "1A00"	End of line: CR or LF character

The address fields are case-insensitive; if the higher bits are 0, they can be omitted, for example

R AB

R aB

R 00AB

These three write instructions are equivalent.

After sending the read request instruction, the read response command will be returned, and the AAAA andBBBBBBBBBB in the read response command are all uppercase.

Table 3-3 Read Response Instruction Format

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
'G'	Space (one)	Address (hexadecimal), e.g. "1A00"	Space (one)	Data (hexadecimal), e.g. "1234ABCD"	End of line: CR and LF character

The read response instruction is an instruction that returns to the serial interface, not input by users, so the format is fixed. For example: G 00AB 0000A1B1

3.3 Bus Timing

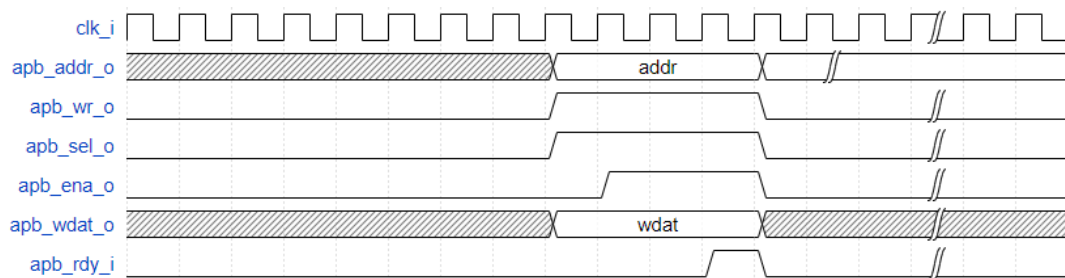
There are two types of bus for you to select: APB master interface or Local interface.

Note!

The address of the APB interface is in unit of 8 bits per byte, while the address of the serial protocol is in unit of 32 bits or 64 bits as set by the IP. The address of the Local interface is the same as the address of the serial protocol, in unit of 32 bits or 64 bits as set by the IP, which is different from the APB interface.

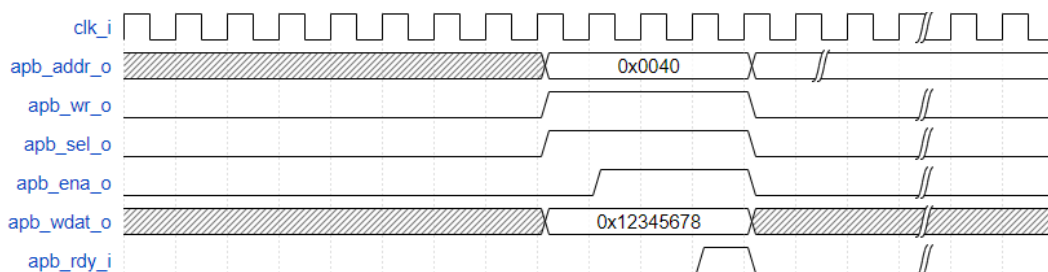
The APB interface that is in accordance with the standard APB interface timing can achieve read and write functions.

Figure 3-2 APB Write Timing



For example, send "W 0010 12345678", and the timing is as follows:

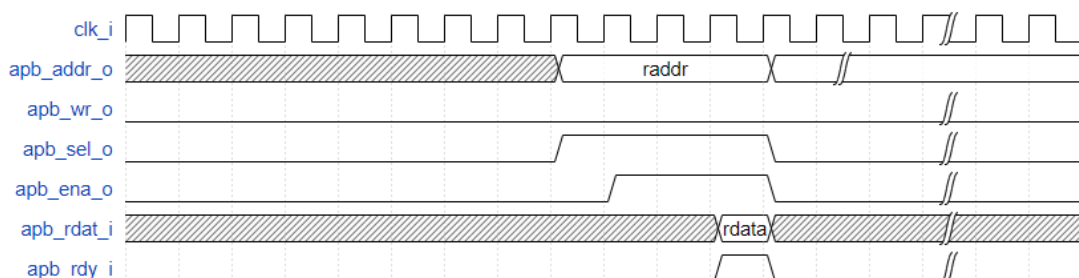
Figure 3-3 Example Timing



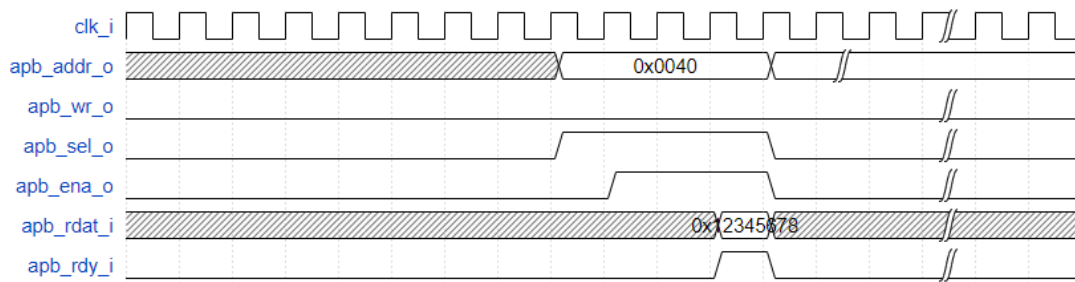
Note!

The address of **apb_addr_o** is 0x0040 (0x0010*4).

Figure 3-4 APB Read Timing



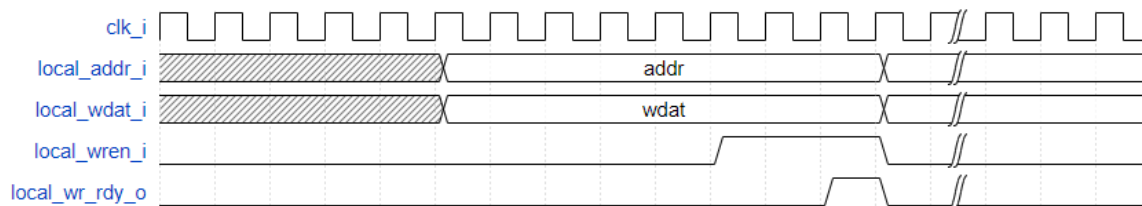
For example, send "R 0010", assuming that the address is 0x12345678, the timing is as follows:

Figure 3-5 Example Timing**Note!**

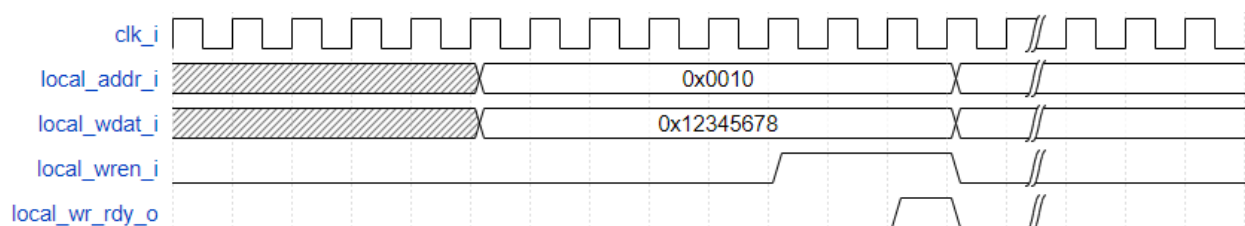
The address of `apb_addr_o` is 0x0040 (0x0010*4).

The timing and operation of Local interface is simple, and the interface can be used for the connection of some modules without specific protocol definition.

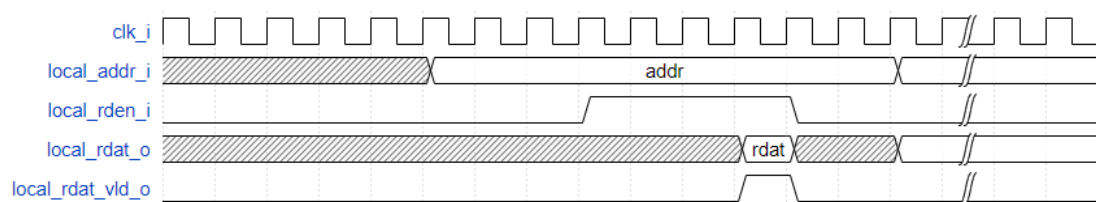
During write register operations, it is necessary for the slave to give the `local_wdat_rdy` flag to the master considering whether the slave is ready to receive the data or not. The write is active when both `local_wren` and `local_wdat_rdy` are high at the same time, and the timing diagram is shown in the following figure.

Figure 3-6 Local Write Timing

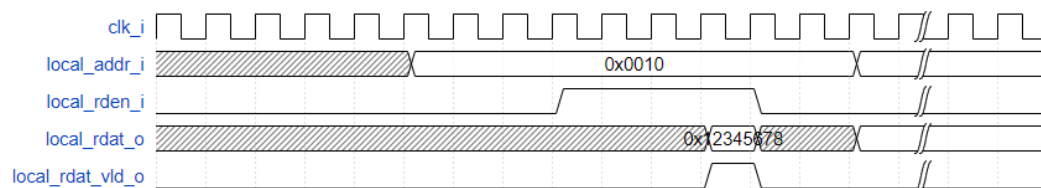
For example, send "W 0010 12345678", and the timing is as follows:

Figure 3-7 Example Timing

During read register operations, it is necessary for the slave to send the `local_wdat_rdy` flag to the master considering whether the slave is ready or not. The read is active when both `local_rden` and `local_rdat_vld` are high at the same time, and the timing diagram is shown in the following figure.

Figure 3-8 Local Read Timing

For example, send "R 0010", assuming that the address is 0x12345678, timing diagram is shown in the following figure.

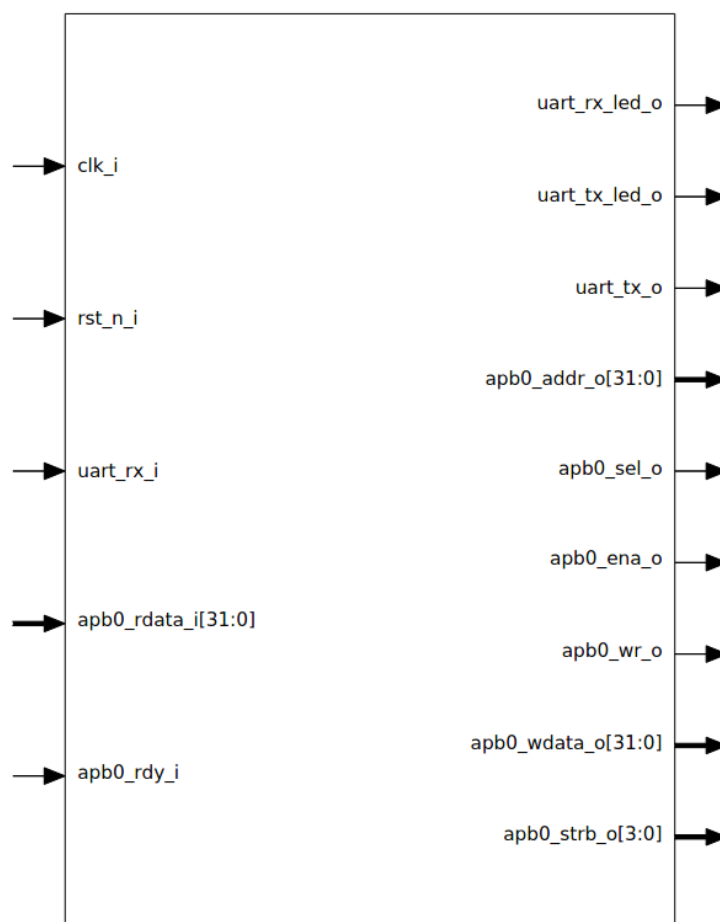
Figure 3-9 Example Timing

4 Signal Definition

4.1 Signal Definition

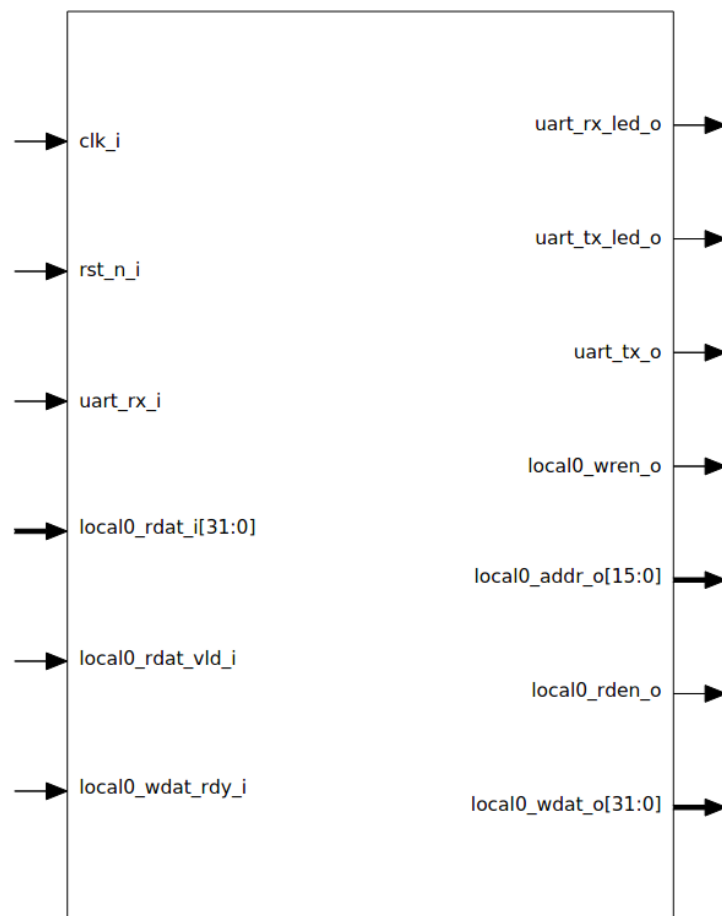
When APB mode is selected, the port diagram is as shown in Figure 4-1.

Figure 4-1 Port Diagram in APB Mode



When Local mode is selected, the port diagram is as shown in Figure 4-2.

Figure 4-2 Port Diagram in Local Mode



The signal descriptions of Gowin UART to Bus IP are as shown in Table 4-1.

Table 4-1 Signal Definition

No.	Signal Name	I/O	Data Width	Description	Note
1	clk_i	I	1	Clock	—
2	rst_n_i	I	1	Asynchronous reset signal	
3	uart_rx_led_o	O	1	The RX indicator	
4	uart_tx_led_o	O	1	The TX indicator	
5	uart_rx_i	I	1	RX cable	
6	uart_tx_o	O	1	TX cable	
7	apbX_addr_o	O	A+log2(B/8)	APB bus address	These interfaces exist

No.	Signal Name	I/O	Data Width	Description	Note
8	apbX_sel_o	O	1	APB bus selection	when APB mode is selected. When slave0 is checked, X=0 interface exists. When slave0, slave1 are checked, X= 0, 1 interfaces exist.
9	apbX_ena_o	O	1	APB bus enable	
10	apbX_wr_o	O	1	APB bus I/O signal 1: Write 0: Read	
11	apbX_rdata_i	I	B	APB read data	
12	apbX_wdata_o	O	B	APB write data	
13	apb0_rdy_i	I	1	APB ready	
14	apbX_strb_o	O	B/8	APB write strobe signal, fixed to all strobes	These interfaces exist when Local mode is selected. When slave0 is checked, X=0 interface exists. When slave0, slave1 are checked, X= 0, 1 interfaces exist.
15	localX_wren_o	O	8	Write enable signal	
16	localX_addr_o	O	A	Write address signal	
17	localX_rden_o	O	1	Read enable signal	
18	localX_wdat_o	O	B	Write data signal	
19	localX_rdat_i	I	B	Read data	
20	localX_rdat_vld_i	I	1	Read data valid indicator	
21	localX_wdat_rdy_i	I	1	Write data ready	

Note!

- Port apb* and local* ports belong to two different timing interfaces, and the two timing interfaces will not exist at the same time through GUI configuration.
- X can represent 0-15, and up to 16 slaves are supported.
- A is the content of the Address Width text box in the IP GUI, which can be configured as 8/16/32.
- B represents the content of the Data Width text box in the IP GUI, which can be configured as 32/64.

4.2 Parameter Configuration Options

Gowin UART to Bus IP parameter configuration is shown in Table 4-2.

Table 4-2 Configuration Options

Options	Description
Bus Mode	APB bus and Local bus can be selected

Options	Description
Address Width	Address width, can be configured as 8/16/32
Data Width	Data width, can be configured as 32/64
Timeout Value	The bus response timeout; if the wait time exceeds this setting, read/write operations are forced. If it is set to 0, you must wait until there is a response before the next read/write operation.
Clock Frequency	Module clock frequency
Baudrate Value	Baud rate setting; you can select one value or enter manually.
SlaveX enable	Slave X enable
SlaveX Base Address	Slave X base address
SlaveX High Address	Slave X high address

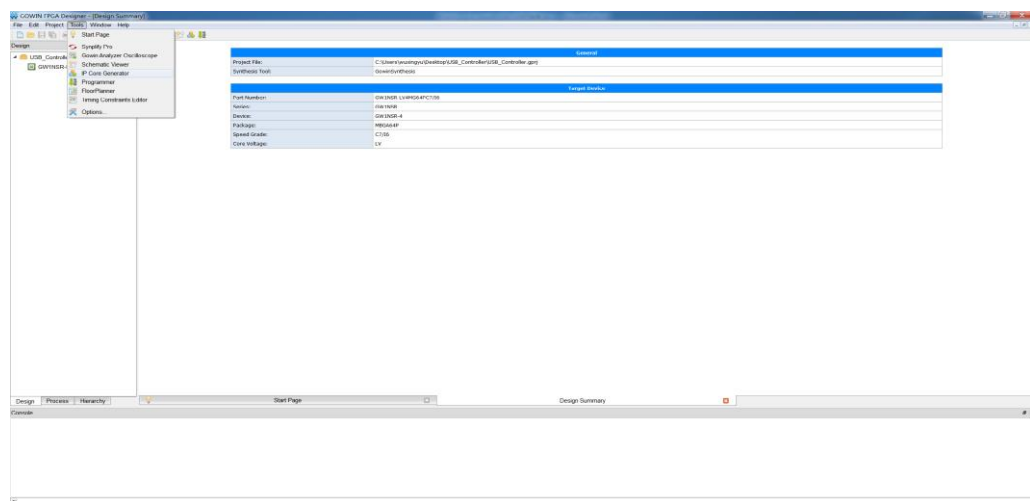
5 Interface Configuration

Select "Tools > IP Core Generator" in Gowin Software, then you can call and configure the UART to Bus IP.

1. Open IP Core Generator

After creating the project, you can click the "Tools" tab in the upper left, then click the IP Core Generator via the drop-down list, as shown in Figure 5-1.

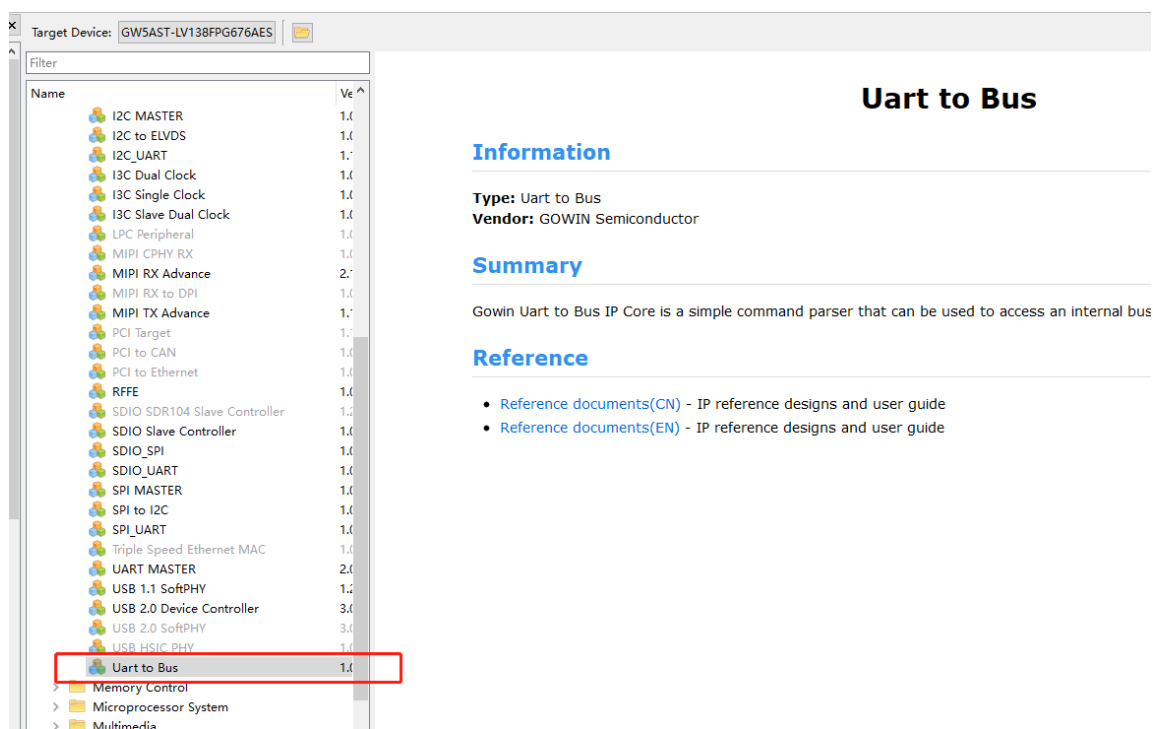
Figure 5-1 IP Core Generator



2. Open UART to Bus IP core

Select "Soft IP Core > Interface and Interconnect > UART to Bus", as shown in Figure 5-2. Double click to open the configuration interface.

Figure 5-2 UART to Bus IP Core



3. UART to Bus IP Core Configuration Interface

UART to Bus IP configuration interface is as shown in Figure 5-3 and Figure 5-4. The port diagram is on the left and the configuration options are on the right.

- You can configure the file name in "File Name" text box.
- You can configure the top module name in "Module Name" text box.
- Basic tab configuration is as follows:
 - Bus Mode: APB bus and Local bus can be select
 - Address Width: 8/16/32 bits
 - Data Width: 32/64 bits
 - Timeout Value: The timeout value of the response signal after the request is transmitted. If there is no response from the slave within the time set through Timeout Value after transmitting a read/write request, the read/write operation will be automatically ended. If the Timeout Value is set to 0, then the next read/write request can be made only when the slave responds.
- Uart Config serial port parameter configuration

- Clock Frequency: Module clock frequency, used to calculate the baud rate counter
- Baudrate Vaule: Set Baudrate
- Slave Config is used to set the number of slaves and address range

Figure 5-3 UART to Bus Configuration Interface (1)

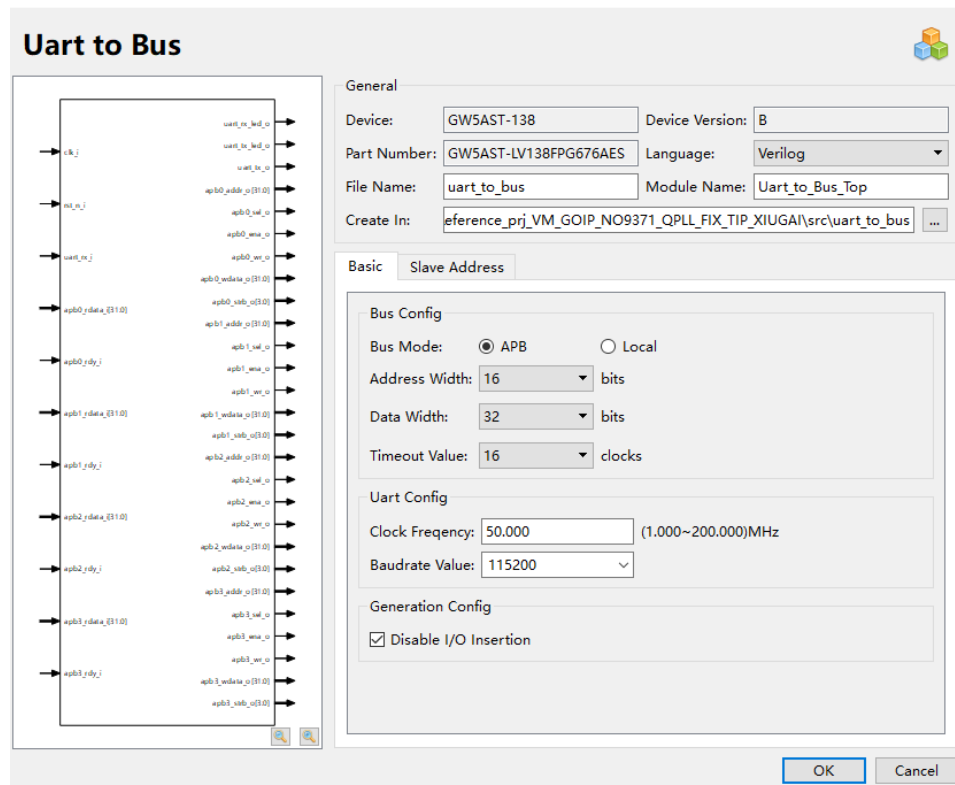


Figure 5-4 UART to Bus Configuration Interface (2)

IP Customization

Uart to Bus

General

Device: Device Version:

Part Number: Language:

File Name: Module Name:

Create In:

Basic **Slave Address**

Slave Config

Slave	Base Address	High Address
☒ slave0	0x0000	0x00FF
☒ slave1	0x0100	0x01FF
☒ slave2	0x0200	0x02FF
☒ slave3	0x0300	0x03FF
☐ slave4	0x0400	0x04FF
☐ slave5	0x0500	0x05FF
☐ slave6	0x0600	0x06FF
☐ slave7	0x0700	0x07FF
☐ slave8	0x0800	0x08FF
☐ slave9	0x0900	0x09FF
☐ slave10	0x0A00	0x0AFF

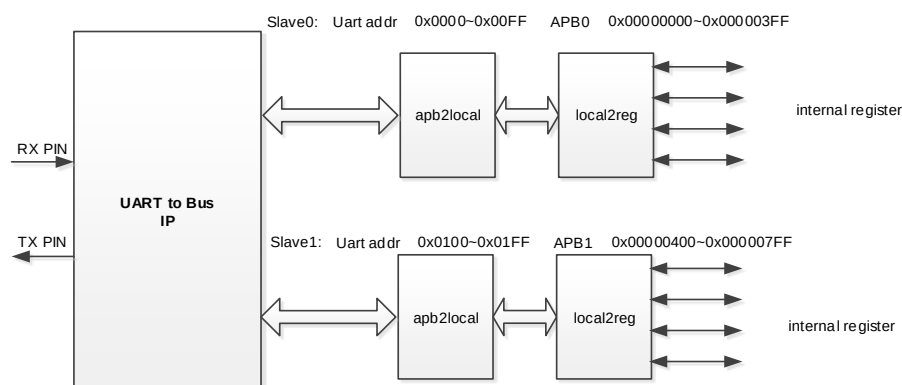
OK Cancel

6 Reference Design

6.1 UART to Bus Reference Design

This chapter is intended to introduce the usage and structure of the reference design of UART to Bus IP. This reference design is a USB-to-UART data loopback design instance. Its basic structure is shown in Figure 6-1.

Figure 6-1 UART to Bus Reference Design Block Diagram



In this design example, the serial assistant from the PC side sends a write instruction to configure the specified registers, then sends a read instruction to read back the specified registers; and control the LEDs on the development board through register 0x0.

6.2 On-board Test

Open the serial debugging assistant software and select the corresponding serial port number. The parity bit is NONE, the data bit is 8, the stop bit is 1, the baud rate is consistent with the setting of IP, and the reference design here is 115200.

Test the address of SLAVE0:

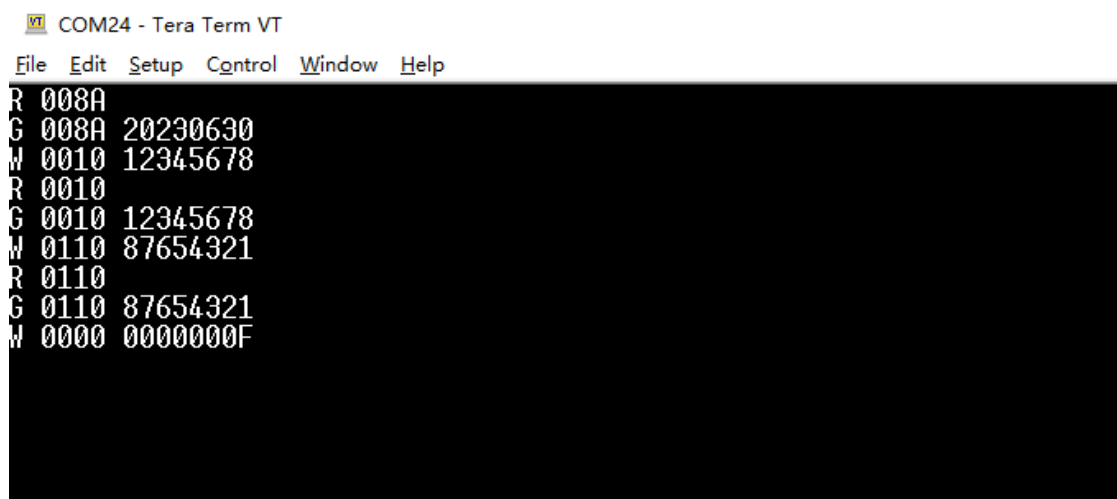
1. Serial assistant sends R 008A
2. Serial assistant displays G 008A 20230630
3. Serial assistant sends W 0010 12345678, then sends R 0010
4. Serial assistant displays G 0010 12345678

Test the address of SLAVE1:

1. Serial assistant sends W 0110 87654321, then sends R 0110
2. Serial assistant displays G 0110 87654321
3. Serial assistant sends W 0000 0000000F, the 4 LEDs on the development board are all on

The test process is as shown in Figure 6-2.

Figure 6-2 Test Process



The screenshot shows a Tera Term VT window titled "COM24 - Tera Term VT". The menu bar includes File, Edit, Setup, Control, Window, and Help. The terminal displays the following sequence of commands and responses:

```
R 008A
G 008A 20230630
W 0010 12345678
R 0010
G 0010 12345678
W 0110 87654321
R 0110
G 0110 87654321
W 0000 0000000F
```

7 File Delivery

The delivery files of Gowin UART to Bus IP includes the documents, the design source code, and the reference design.

7.1 Documentation

The folder mainly contains the user guide and release note in PDF version.

Table 7-1 Documents List

Name	Description
IPUG1022, Gowin UART to Bus IP User Guide	Gowin UART to Bus IP User Guide, that is this manual.
RN1022, Gowin UART IP Release Note	Gowin UART IP Release Note

7.2 Design Source Code (Encryption)

The encryption code folder contains the RTL encryption code of Gowin UART to Bus IP for the GUI to be used with the Gowin Software to generate the IP core required by users.

Table 7-2 Gowin UART to Bus IP Design Source Code List

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

7.3 Reference Design

Gowin UART to Bus IP RefDesign folder contains Gowin UART to Bus IP netlist file, user reference design, constraints file, top-level file, and project file, etc.

Table7-3 Gowin UART to Bus IP RefDesign Folder Content List

Name	Description
top.v	The top module of reference design
top.cst	Project physical constraints file
top.sdc	Project timing constraints file
uart_to_bus.v	Generate UART to Bus IP top-level file, encrypted
apb2local.v	Internal register module
local2reg.v	Internal register module

