GW2AN-18X & 9X

Programming and Configuration Guide

UG702-1.2.2E, 09/26/2025

Copyright © 2025 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN, Gowin, and GOWINSEMI are trademarks of Guangdong Gowin Semiconductor Corporation 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. 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
09/10/2021	1.0E	Initial version published.
03/04/2022	1.0.1E	The default states of configuration pins updated.
07/28/2022	1.0.2E	Note about I^2C configuration mode added.
09/06/2022	1.1E	● Section 5.8 I2C Configuration Mode updated. ● Section 6.5 Background Programming added.
12/08/2022	1.1.1E	● The description of CLKHOLD_N pin updated. ● Figure 5-40 CPU Mode Configuration Timing added. ● Table 5-14 CPU Configuration Timing Parameters added.
04/13/2023	1.1.2E	● Figure 5-39 CPU Mode Configuration diagram updated. ● Figure 5-40 CPU Mode Configuration Timing updated. ● Table 5-14 CPU Configuration Timing Parameters updated. ● Description of programming Flash using QSSPI mode added.
07/25/2023	1.1.3E	● Note on RECONFIG_N pin updated. ● Table 5-1 Timing Parameters for Power Cycling and RECONFIG_N Triggering updated. ● Note on the frequency tolerance of the MSPI clock added.
11/16/2023	1.1.4E	● Table 5-8 Status Register Definition updated. ● The formula for loading time fixed. ● Note under Table 6-1 Gowin FPGA GW2AN-18X/9X Products Configuration File Size (Max.) modified. ● Figure 5-51 Flow Chart of Configuring the Flash via I2C modified. ● Section 5.4.5 Connection Diagram for SSPI Configuration Mode updated. ● Section Process of Programing the Flash via I2C updated.
07/26/2024	1.2E	● “5.2 AUTO BOOT Configuration”updated. ● “Figure 5-16 Connection Diagram of Daisy-Chain” updated. ● “Appendix A Pin Status Information” added. ● Description of JTAG configuration mode improved. ● Description of I^2C configuration mode improved. ● “Table 5-1 Timing Parameters for Power Cycling and RECONFIG_N Triggering” updated, correcting value of $T_{portready}$.
03/28/2025	1.2.1E	Description of daisy chain operation updated.
09/26/2025	1.2.2E	● Note for“Figure 5-38 Connection Diagram for CPU Mode” modified. ● “Table 4-3 Pin Function” updated：improving description of RECONFIG_N pin. ● “Appendix B Example File Information” added.

Contents

Contents i

List of Figures iii

List of Tables v

1 About This Guide 1

1.1 Purpose 1

1.2 Related Documents 1

1.3 Terminology and Abbreviations 1

1.4 Support and Feedback 2

2 Glossary 3

3 Configuration Mode Overview 5

4 Configuration Pin 7

4.1 Configuration Pin List and Reuse Options 7

4.1.1 Configuration Pin List 7

4.1.2 Configuration Pin Reuse 8

4.2 Configuration Pin Function and Application 10

5 Configuration Mode Details 14

5.1 Configuration Notes 14

5.2 AUTO BOOT Configuration 17

5.3 JTAG Configuration 18

5.3.1 JTAG Configuration Mode Pins 18

5.3.2 Connection Diagram for the JTAG Configuration Mode 19

5.3.3 JTAG Configuration Timing 20

5.3.4 JTAG Configuration Process 20

5.4 SSPI 37

5.4.1 SSPI Mode Pins 38

5.4.2 SSPI Configuration Timing 39

5.4.3 SSPI Configuration Instruction 39

5.4.4 SSPI Configure SRAM Flow 43

5.4.5 Connection Diagram for SSPI Configuration Mode 44

5.4.6 Multiple FPGA Connection View in SSPI Mode 47

5.5 QSSPI Configuration Mode 48

5.6 CPU Configuration Mode 53
5.6.1 Configuration Timing 54
5.7 SERIAL 56
5.8 I^2C Configuration Mode 57
5.8.1 Configuration Instruction 60
5.8.2 The Process of Configuring SRAM via I^2C 61
5.8.3 The Process of Configuring (Programming) Flash via I^2C 62
5.9 Example Files 65

6 Bitstream File Configuration 66

6.1 Configuration Options 66
6.2 Configure Data Encryption 67
6.2.1 Definition 67
6.2.2 Enter Encryption KEY 68
6.2.3 Enter the Decrypt Key 68
6.2.4 Programming Operation 69
6.2.5 Programming Flow 71
6.3 Configuration File Size 74
6.4 Configuration File Loading Time 75
6.5 Background Programming 76

7 Safety Precautions 78

8 Boundary Scan 80

Appendix A Pin Status Information 82

Appendix B Example File Information 85

List of Figures

Figure 4-1 Configuring Pin Reuse 10
Figure 5-1 Recommended Pin Connection 16
Figure 5-2 Power Recycle Timing 16
Figure 5-3 Trigger Timing 17
Figure 5-4 Connection Diagram for JTAG Configuration Mode 19
Figure 5-5 Connection Diagram of JTAG Daisy-Chain Configuration Mode 19
Figure 5-6 JTAG Configuration timing 20
Figure 5-7 TAP State Machine 21
Figure 5-8 Instruction Register Access Timing 22
Figure 5-9 Data Register Access Timing 22
Figure 5-10 Flow Chart of Reading ID Code State Machine 24
Figure 5-11 Access Timing of Reading ID Code Instruction - 0x11 24
Figure 5-12 Access Timing of Reading ID Code Data Register 24
Figure 5-13 SRAM Configuration Flow 26
Figure 5-14 Process of Reading SRAM 28
Figure 5-15 Process of Use Boundary Scan Mode to Program SPI Flash 35
Figure 5-16 Connection Diagram of Daisy-Chain 37
Figure 5-17 SSPI Configuration Timing 39
Figure 5-18 Read ID Code Timing 40
Figure 5-19 Write Enable (0x15) Timing 41
Figure 5-20 Write Disable(0x3A00) Timing 41
Figure 5-21 Write Data (0x3B) Timing 42
Figure 5-22 SSPI Configuration Mode Connection Diagram 44
Figure 5-23 Connection Diagram of Programming Internal Flash via SSPI 44
Figure 5-24 Timing of Reading the Internal Flash ID Code via SSPI 45
Figure 5-25 Timing of Sending the Write Disable(0x04) Instruction via SSPI 45
Figure 5-26 Timing of Sending the Write Enable(0x06) Instruction via SSPI 46
Figure 5-27 Timing of Programming a Page via SSPI 46
Figure 5-28 Timing of Reading Back a Byte via SSPI 46
Figure 5-29 Flow Chart of Programming the Flash via SSPI 47
Figure 5-30 Multiple FPGA Connection Diagram 1 47
Figure 5-31 Multiple FPGA Connection Diagram 2 48

Figure 5-32 QSSPI Write Data (0x6B) Timing 49
Figure 5-33 Flow Chart of Configuring the SRAM via QSSPI 50
Figure 5-34 Flow Chart of Programming the Flash via QSSPI 51
Figure 5-35 QSSPI Write Instruction Timing 52
Figure 5-36 QSSPI Write Data Timing 52
Figure 5-37 QSSPI Page Program Timing 53
Figure 5-38 Connection Diagram for CPU Mode 54
Figure 5-39 CPU Mode Configuration diagram 55
Figure 5-40 CPU Mode Configuration Timing 55
Figure 5-41 Connection Diagram for SERIAL Mode 56
Figure 5-42 SERIAL Configuration Timing 57
Figure 5-43 Connection Diagram for I^2C Configuration Mode 58
Figure 5-44 I^2C Configuration timing 58
Figure 5-45 Reinit Timing Diagram 60
Figure 5-46 SRAM Configuration Timing Diagram 61
Figure 5-47 Flash Configuration Timing Diagram 61
Figure 5-48 Reboot Timing Diagram 61
Figure 5-49 Flow Chart of Configuring SRAM via I^2C 62
Figure 5-50 Flow Chart of Erasing Flash via I^2C： 63
Figure 5-51 Flow Chart of Configuring the Flash via I^2C 64
Figure 6-1 Configuration Options 67
Figure 6-2 Encryption Key Setting Method 68
Figure 6-3 Setting the Decryption Key 69
Figure 6-4 AES Security Configure 70
Figure 6-5 Prepare 71
Figure 6-6 Read AES Key Flow 72
Figure 6-7 Program AES Key Flow 73
Figure 6-8 Lock AES Key Flow 74
Figure 6-9 Bitstream Format generation 75
Figure 6-10 The Background Programming Option 77
Figure 8-1 Boundary Scan Operation Schematic Diagram 81

List of Tables

Table 1-1 Terminology and Abbreviations 1
Table 2-1 Glossary 3
Table 3-1 Configuration Modes 5
Table 4-1 Configuration Pin List 7
Table 4-2 Pin Reuse Options 8
Table 4-3 Pin Function 10
Table 5-1 Timing Parameters for Power Cycling and RECONFIG_N Triggering 17
Table 5-2 Pin Description in JTAG Configuration Mode 18
Table 5-3 JTAG Configuration Timing Parameters 20
Table 5-4 Gowin FPGA Device IDCODE 23
Table 5-5 Change of TDI and TMS Value in The Process of Sending Instructions 23
Table 5-6 Count of Address and Length of One Address 27
Table 5-7 Pin State 35
Table 5-8 Status Register Definition 36
Table 5-9 SSPI Mode Pins 38
Table 5-10 SSPI Configuration Timing Parameters 39
Table 5-11 Configuration Instruction 40
Table 5-12 QSSPI Mode Pins 48
Table 5-13 CPU Mode Pins 53
Table 5-14 CPU Configuration Timing Parameters 55
Table 5-15 Pin Definition in SERIAL Configuration Mode 56
Table 5-16 SERIAL Configuration Timing Parameters 57
Table 5-17 Pin Definition in I^2C Configuration Mode 57
Table 5-18 I^2C Configuration Timing Parameters 58
Table 5-19 I^2C Configuration Mode Frequency and Address 60
Table 5-20 I^2C Configuration Instruction 60
Table 6-1 Gowin FPGA GW2AN-18X/9X Products Configuration File Size (Max.) 75
Table 6-2 Loading Time in Autoboot Mode 76
Table A-1 Pin States at Each Stage for GW2AN-18X & 9X Devices (MSPI, MODE=00) 83
Table A-2 Pin States at Each Stage for GW2AN-18X & 9X Devices (AUTOBOOT, MODE=01) 84
Table B-1 Example File Information 85

1 About This Guide

1.1 Purpose

This guide mainly introduces general features and functions on programming and configuration of the GW2AN-18X/9X device in Arora family. It helps users to use Gowin FPGA products to their full potential.

1.2 Related Documents

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

- DS971, GW2AN-18X and GW2AN-9X Data Sheet
- UG973, GW2AN-18X and GW2AN-9X Package & Pinout User Guide
- UG972, GW2AN-18X Pinout
- UG978, GW2AN-9X Pinout

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	Full Name
Background Programming	Embedded Flash Background Programming
Bitstream	Bitstream Data
Bscan	Boundary Scan
Configuration	Configuration
Configuration Data	Configuration Data
Configuration Mode	Configuration Mode
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
Edit Mode	Edit Mode
EFlash/EmbFlash	Embedded Flash

Terminology and Abbreviations	Full Name
FPGA	Field Programmable Gate Array
FS file	Fuses file
GPIO	Gowin Programmable I/O
I2C（I^2C, IIC）	Inter-Integrated Circuits
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
Internal Flash	Internal Flash
JTAG	Joint Test Action Group
LSB	Least Significant Bit
LUT	Look-up Table
MSB	Most Significant Bit
MSPI	Master Serial Peripheral Interface
Programming	Programming
SCL	Serial Clock
SDA	Serial Data
Security Bit	Security Bit
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SSPI	Slave Serial Peripheral Interface
TAP	Test Access Port
User Mode	User Mode

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 Glossary

This chapter presents an overview of the terms that are commonly used in the process of programming and configuring of Gowin FPGA products to help users get familiar with the related concepts.

Table 2-1 Glossary

Glossary	Meaning
Program	Write the bitstream data generated by Gowin software to the embedded Flash or external SPI Flash of FPGA.
Configure	Load the bitstream data generated by Gowin software to the FPGA SRAM via external interfaces or embedded Flash.
GowinCONFIG	In addition to the generic JTAG configuration mode, Gowin FPGA products support additional configurations, including AUTO BOOT configuration, DUAL BOOT configuration, MSPI configuration, SSPI configuration, SERIAL configuration, and CPU configuration. How many GowinCONFIG configuration modes each device supports depend on the device model and package.
MODE[1:0]	A representation of the two MODE pin values associated with GowinCONFIG.
AUTO BOOT Configuration	FPGA loads bitstream data into the SRAM from an embedded Flash. Only non-volatile devices support this mode.
MSPI Configuration	As a master, FPGA is configured by reading bitstream from the external Flash via the SPI interface automatically.
SSPI Configuration	As a slave device, the bitstream data is written into the FPGA via the SPI interface by the external master.
QSSPI Configuration	As a slave device, the bitstream data is written into the FPGA via the QSPI interface by the external master.
SERIAL Configuration	As a slave device, the bitstream data is written into the FPGA via the serial interface by the external master.
CPU Configuration	The bitstream data is written into the FPGA via the QSPI interface by a parallel interface (8-bit).
I^2C Configuration	As a slave device, the bitstream data is written into the FPGA via the I^2C interface by the external master.

Glossary	Meaning
MULTI BOOT Configuration	The derivative concept of MSPI, it refers to that FPGA reads bitstream data from different addresses of external Flash. The loading address of the latter bitstream data is written in previous bitstream data and the configuration is completed by triggering RECONFIG_N to switch the data stream file under the condition that the device power is on. FPGA products that support MSPI all support this mode.
Remote Upgrade	After FPGA starts to work, if an upgrade is required, first write bitstream to an embedded or external Flash through remote operation, and then FPGA reads the external Flash by triggering RECONFIG_N or powering up again to complete the configuration.
Daisy Chain	FPGA devices are connected sequentially in a serial way. Devices can be configured from the head of the chain in sequence according to the connection order, and data can only be transmitted between adjacent devices.
User Mode	Hands over control to users when the FPGA configuration has been completed. Only in user mode, configuration pins can be reused as GPIOs (Gowin Programmable I/O).
Edit Mode	FPGA can be programmed and configured in this mode. All configuration pins cannot be reused as GPIOs. The output of all GPIOs is high-impedance state, except transparent transmission.
ID CODE	Identification for the the Gowin FPGA device. Each series of devices has a different number.
USER CODE	Used to identify the FPGA device that used. The user code can be written to the FPGA device through Gowin programmer. Up to 32-bit can be supported.
Security Bit	A special design for the configuration data security of Gowin FPGA product. After you write the bitstream with security bit to the device SRAM, no one will be able to read back the data. Gowin software sets a security bit for the bitstream data of all FPGA products by default.
Encryption	The Arora family of FPGA products support this feature. After the encrypted bitstream is written to FPGA, the device will match the pre-stored key automatically, and then decrypt and wake up the device after successful matching. The device cannot work if matching fails.

3 Configuration Mode Overview

Besides the JTAG configuration mode that is commonly used in the industry, the Arora Family of FPGA products also support GOWINSEMI's own configuration mode: GowinCONFIG. The GowinCONFIG configuration modes that are available and supported for each device depend on the device model and package. The Arora Family of FPGA Products support bitstream encryption and security bit setting, which is safety for user design. The Arora Family FPGA products support bitstream decompression; users can compress bitstream to save storage memory.

16M-bit Serial Flash (With Quad SPI) is embedded in GW2AN-18X/9X. Up to 100Mhz Quad SPI configuration mode can be supported, and Fixed-Address GOLDEN-IMAGE mode is supported.

Table 3-1 lists the configuration modes that are supported by GW2AN-18X/ 9X.

Table 3-1 Configuration Modes

Configuration Modes		MODE[1:0][1]	Description
JTAG		XX[2]	FPGA products are configured via JTAG interface by external Host. Supports up to 62.5 Mhz.
GowinCONFIG	MSPI[5]	00	As a Master, FPGA reads data from the internal Flash via the SPI interface for configuration.· Supports up to 100 Mhz.
	Autoboot	01	As a Master, FPGA reads data from the internal Flash via the QSPI interface for configuration.· Supports GOLDEN IMAGE. Supports up to 100 Mhz.
	SSPI	0X[3]	Supported automatically upon completion of Autoboot or MSPI. FPGA products are configured via SPI interface by external Host. Supports up to 100 Mhz.
	QSSPI		Supported automatically upon completion of Autoboot or MSPI. FPGA products are configured via QSPI

Configuration Modes		MODE[1:0][1]	Description
			interface by external Host. Supports up to 100 Mhz.
	I^2C		Supported automatically upon completion of Autoboot or MSPI. FPGA products are configured via I^2C interface by external Host. The supported frequency is 100KHz~555KHz..
	SERIAL[4]	10	FPGA products are configured via DIN interface by external Host.
	CPU[4]	11	FPGA products are configured via DBUS interface by external Host.

Note!

- [1] For the value of unbound mode pins, please refer to the related pinout manuals;
- [2] The JTAG configuration mode is independent of the input values of MODE [1:0];
- [3] The SSPI configuration mode is independent of the input values of MODE0;
- [4] The CPU configuration mode and SERIAL configuration mode share SCLK, WE_N and CLKHOLD_N. The data bus pins for the CPU configuration mode share pins with MSPI and SSPI configuration modes.
- [5] The MSPI configuration mode clock frequency has a tolerance of ±10%.

Note!

For details about configuration pins, pin reuse, and pin functions and application, please refer to 4 Configuration Pin.

4 Configuration Pin

Gowin FPGA products have various configuration modes, including generic JTAG configuration, active configuration, passive configuration, serial configuration, and parallel configuration, etc., which can meet the various peripheral requirements of different users. The programming and configuration pins can be used as configuration pins and also can be reused as GPIO. Users can configure the pins as required. Users also can configure them according to their configuration functions to meet specific requirements.

4.1 Configuration Pin List and Reuse Options

4.1.1 Configuration Pin List

Table 4-1 contains a list of all the configuration pins of Gowin FPGA products together with the details of the pins used in each configuration mode and the shared pins in chip packages.

Table 4-1 Configuration Pin List

Pin Name	I/O	JTAG	GowinCONFIG					
			AUTO BOOT	I^2C	SSPI	QSSPI	SERIAL	CPU
RECONFIG_N	I	Yes	Yes	Yes	Yes	Yes	Yes	Yes
JTAGSEL_N	I	Yes						
TDO	O	Yes						
TMS	I	Yes						
TCK	I	Yes						
TDI	I	Yes						
READY	I/O	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DONE	I/O	Yes	Yes	Yes	Yes	Yes	Yes	Yes
MODE[1:0]	I		Yes	Yes	Yes	Yes	Yes	Yes
SCLK	I				Yes	Yes	Yes	Yes

Pin Name	I/O	JTAG	GowinCONFIG					
			AUTO BOOT	I^2C	SSPI	QSSPI	SERIAL	CPU
CLKHOLD_N/DIN	I				Yes	Yes	Yes	Yes
WE_N/DOUT	O					Yes	Yes	Yes
D7	I/O							Yes
D6	I/O							Yes
D5	I/O							Yes
D4	I/O							Yes
FASTRD_N /D3	I/O							Yes
SI /D2	I/O				Yes	Yes		Yes
SO /D1	I/O				Yes	Yes		Yes
SSPI_CS_N/D0	I/O				Yes	Yes		Yes
SCL	I			Yes				
SDA	I/O			Yes				

Note!

- For the configuration modes supported by different devices, please refer to 3 Configuration Mode Overview.
- Please refer to 5 Configuration Mode Details for the definition of each pin in different configuration modes.

4.1.2 Configuration Pin Reuse

To maximize the utilization of I/O, Gowin FPGA product support for setting the configuration pins as GPIO pins. Before any configuration operation is performed on all series of Gowin FPGA products after power up, all related configuration pins are used as configuration pins by default. After successful configuration, the device enters into user mode and reassigns the pin functions according to the multiplex options selected by the user.

Note!

When setting the pin reuse options, ensure the external initial connection state of the pins does not affect the device configuration. Isolate the connections that affect the configuration first, and then wait to modify them in user mode.

The reuse options for the configuration pins are detailed in Table 4-2.

Table 4-2 Pin Reuse Options

Name	Options	Description
JTAG PORT	Default Status	TMS, TCK, TDI, and TDO are used as dedicated configuration pins. JTAGSEL_N is used as GPIO.
	Set as GPIO	JTAGSEL_N pins are used as dedicated configuration pins: ● JTAGSEL_N=0, TMS, TCK, TDI, and TDO are used as configuration pins: ● JTAGSEL_N = 1, TMS, TCK, TDI, and TDO are used as GPIO after configuration.

Name	Options	Description
I^2C PORT	Default Status	SCL and SDA pins are used as dedicated configuration pins.
	Set as GPIO	SCL and SDA pins are used as GPIO after configuration.
SSPI PORT	Default Status	SCLK, CLKHOLD_N, SSPI_CS_N, SI and SO are used as dedicated configuration pins.
	Set as GPIO	SCLK, CLKHOLD_N, SSPI_CS_N, SI and SO are used as GPIO after configuration.
QSSPI PORT	Default Status	SCLK, CLKHOLD_N, SSPI_CS_N, SI and SO, and QSSPI_WPN are used as dedicated configuration pins.
	Set as GPIO	SCLK, CLKHOLD_N, SSPI_CS_N, SI and SO, and QSSPI_WPN are used as GPIO after configuration.
RECONFIG_N	Default Status	Dedicated configuration pins.
	Set as GPIO	Used as GPIO after configuration.
READY	Default Status	Dedicated configuration pins.
	Set as GPIO	Used as GPIO after configuration.
DONE	Default Status	Dedicated configuration pins.
	Set as GPIO	Used as GPIO after configuration.

Note!

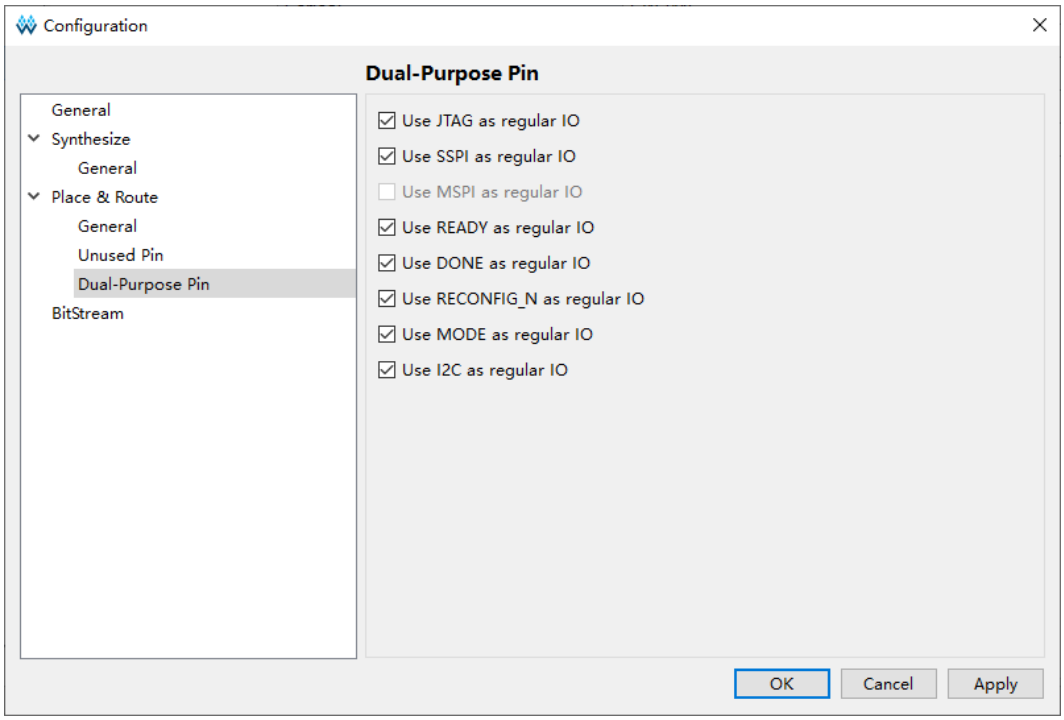
- [1] For the devices with JTAGSEL_N unbound, when debugging the cases with JTAG pin reuse, it's suggested to set MODE value to non-auto configuration mode before power up to avoid the other bit stream data affecting configuration. After the device is power up and JTAG is manually configured, the device enters User Mode,and the JTAG pin is used as GPIO. For the LittleBee® Family of FPGA products, when MODE[2: 0]=001, the JTAGSEL_N pin and the four JTAG Configuration pins (TCK, TMS, TDI, TDO) can be set as GPIOs simultaneously, but the JTAG pin cannot be recovered as a configuration pin by JTAGSEL_N. It can be recovered when the device reenters the edit mode.
- [2] The SERIAL and CPU modes share pins with other configuration modes and cannot be set as GPIO separately, however, the pins can be set as GPIO in non-shared configuration modes.

Configuration Pin Reuse

The steps are as follows:

1. Open the project in Gowin software;
2. Select “Project > Configuration > Dual Purpose Pin” from the menu options, as shown in Figure 4-1;
3. Check the corresponding options to set the configuration pins reuse.

Figure 4-1 Configuring Pin Reuse



4.2 Configuration Pin Function and Application

The RECONFIG_N, READY, and DONE pins are used in all configuration modes. Other pins can be set as dedicated pins or GPIO (Gowin Programmable IO) according to their specific application.

Table 4-3 Pin Function

Pin Name	Functional Description
RECONFIG_N	As a configuration pin, RECONFIG_N is an input pin that has an internal weak pull-up. Active low is used as the reset function for the FPGA programming configuration. The **FPGA can't be configured if RECONFIG_N is set to low**. **It is important to keep this pin high or floating during power-up, initialization, and configuration of the FPGA, and it can be released after configuration is complete.** As a configuration pin, a low-level signal with a pulse width of no less than 25ns is required for GowinCONFIG to reload bitstream data according to the MODE setting value. You can also write logic to control this pin to trigger device reconfiguration: (1) For devices with a bonded-out RECONFIG_N pin, you can control the RECONFIG_N function by constraining the I/O location. (2) For devices without a bonded-out RECONFIG_N pin, you can control the RECONFIG_N function by connecting to the RECONFIG_N node through the net constraint. For details on constraint methods, refer to SUG935, Gowin Design Physical Constraints User Guide, Section A.1 I/O Location Constraints and Section A.10.2 RECONFIG_N Net Constraints. As a GPIO, it can be used as an input or output type. To ensure a smooth configuration, you need to set the initial value of

Pin Name	Functional Description
	RECONFIG_N to high when reusing it.
READY	In-out pins. Active-high. FPGA can be configured only when the READY signal is pulled up. When the READY signal is pulled down, recover the status by powering up or triggering RECONFIG_N. As an output configuration pin, it indicates that the FPGA can be configured or not. If the FPGA meets the configuration condition, the READY signal is high. If the configuration fails, READY signal is low. As an input configuration pin, you can delay the configuration via its own logic or by pulling down the READY signal. As a GPIO, it can be used as an input or output type. **If READY is used as an input GPIO, the initial value needs to be 1 before configuration. Otherwise, the FPGA cannot be configured.**
DONE	In-out pins. A signal which indicates FPGA is configured successfully, DONE is pulled up after successfully configuring. As an output configuration pin, it indicates the current configuration of FPGA: if configured successfully, the DONE signal is high and the device enters into working state. if the configuration fails, the DONE signal keeps low. As an input configuration pin, the user can delay the entering of user mode via its own internal logic or by reducing the DONE signal. When RECONFIG_N or READY signals are low, DONE signal also keeps low. When configuring SRAM using JTAG circuit, it does not need to take DONE signal into account. As a GPIO, it can be used as an input or output type. **If DONE is used as an input GPIO, the initial value of DONE should be 1 before configuring. Otherwise, the FPGA will fail to enter the user mode after being configured.**
MODE	GowinCONFIG modes selection pin. As the selection pin of GowinCONFIG modes, MODE is an input pin that has internal weak pull-down. The maximum bit width is 2 bits. When FPGA powers up or a low level pulse triggers RECONFIG_N, the device enters the corresponding GowinCONFIG mode in accordance with the MODE value. The same MODE value of the different Gowin series of FPGA products may have different configuration MODE. As the number of pins for each package is different, some MODE pins are not all bonded out, and the unbound MODE pins are grounded by default. Please refer to the corresponding PINOUT manual for further details. When MODE pins are used as GPIOs, they can be used as an input or output type. Note that when the MODE value changes, power-on again or providing one low pulse for triggering RECONFIG_N is required for it to take effect.
JTAGSEL_N	As a configuration pin, it is an input pin with internal weak pull-down. If JTAG pins are set as a GPIO in the Gowin software, the JTAG pins can become GPIOs after the device being powered up and successfully configured. The JTAG pin configuration functions can be recovered by pulling down JTAGSEL_N. The JTAG configuration functions are always available if no JTAG pin reuse is set. As a GPIO, it can be used as an input or output type.

Pin Name	Functional Description
	Note! The JTAGSEL_N pin and four JTAG pins (TCK, TMS, TDI, and TDO) are exclusive. JTAG pins can only be used as configuration pins if JTAGSEL_N is set as a GPIO. JTAGSEL_N can only be used as a configuration pin if JTAG pins are set as GPIOs. For the LittleBee® Family of FPGA products, when MODE[2: 0]=001, the JTAGSEL_N pin and the four JTAG pins (TCK, TMS, TDI, TDO) can be set as GPIOs simultaneously, but the JTAG pin cannot be recovered as a configuration pin by JTAGSEL_N. It can be recovered when the device reenters the edit mode.
TCK	As a configuration pin, it is an input pin. It is a serial clock input pin in the JTAG configuration mode. As a GPIO, it can be used as an input or output type.
TMS	As a configuration pin, it is an input pin with internal weak pull-up. It is a serial input pin in JTAG configuration mode. As a GPIO, it can be used as an input or output type.
TDI	As a configuration pin, it is an input pin with internal weak pull-up. It is a serial data input pin in JTAG configuration mode. As a GPIO, it can be used as an input or output type.
TDO	As a configuration pin, it is an output pin. It is a serial data output pin in JTAG configuration mode. As a GPIO, it can be used as an input or output type.
SCLK	As a configuration pin, it is an input pin. It is a clock input pin in SSPI, SERIAL, and CPU configuration modes. As a GPIO, it can be used as an input or output type.
CLKHOLD_N	As a configuration pin, it is an input pin with internal weak pull-down. It is a clock-locking pin in SSPI and CPU configuration modes: active high in SSPI mode; active low in CPU mode. As a GPIO, it can be used as an input or output type.
SSPI_CS_N	As a configuration pin, it is an input pin with internal weak pull-up. It is a chip selection signal in the SSPI configuration mode, active low. As a GPIO, it can be used as an input or output type.
SI	As a configuration pin, it is an input pin. It is a serial data input pin in the SSPI configuration mode. As a GPIO, it can be used as an input or output type.
SO	As a configuration pin, it is an output pin. It is a serial data output pin in the SSPI configuration mode. As a GPIO, it can be used as an input or output type.
QSSPI_WPN	As a configuration pin, it is an output pin. It is a serial data output pin in QSSPI configuration mode. As a GPIO, it can be used as an input or output type.
FASTRD_N	As a configuration pin, it is an input pin. In the MSPI mode, FASTRD_N is used to select Flash access speed. High indicates regular Flash access mode(command 0x03). Low indicates high-speed Flash access mode; The high-speed flash access command of each manufacturer is different. Please refer to the corresponding Flash manual. As a GPIO, it can be used as an input or output type.
WE_N	As a configuration pin, it is an input pin.

Pin Name	Functional Description
	Select the data input/output of D[7:0] in CPU mode: Read operation when WE_N is high; write operation when WE_N is low. As a GPIO, it can be used as an input or output type.
D0~D7	In-out pins. Data input/output pins in CPU configuration mode, 8-bit width. Determine the input/output of D0 ~ D7 according to WE_N. As a GPIO, it can be used as an input or output type.
DIN	As a configuration pin, it is an input pin with internal weak pull-down. It is a serial data input pin in the SERIAL configuration mode. As a GPIO, it can be used as an input or output type.
DOUT	As a configuration pin, it is an output pin. It is a serial data output pin in the SERIAL configuration mode, which is only used as the input to the latter device when the FPGA is cascading. As a GPIO, it can be used as an input or output type.
SCL	As a configuration pin, it is an input pin. As a GPIO, it can be used as an input type.
SDA	As a configuration pin, it is an in/out pin. As a GPIO, it can be used as an input or output type.

5 Configuration Mode Details

GW2AN-18X/9X products are non-volatile devices with internal flash. Any configuration data that is stored in the SRAM device is lost after it is powered down; as such, it needs to be reconfigured each time it is powered up. The data stored in non-volatile devices with internal flash is still stored in the chip if the device is powered down, and the device can be automatically reconfigured after power up via the AUTOBOOT or DUALBOOT configuration options.

Gowin FPGA products have abundant packages. The configuration modes supported by each device are related to the number of configuration pins bonded out: All devices support JTAG configuration. The mode value for each configuration is different.

5.1 Configuration Notes

Power Up and Configuration Flow

When the power up voltage of VCC, VCCIO, and VCCX reaches the min. value, FPGA begins to start: stable voltage and RECONFIG_N is not pulled down > The internal circuit of FPGA pulls down READY and DONE pins > FPGA initialization > Pulling up READY and sampling MODE value > Reading and checking the configuration data according to the configuration mode > FPGA waking up > DONE pulling up > Entering user mode.

The power supply needs to be stable during FPGA startup. No low levels are allowed on the RECONFIG_N pin during power-up, initialization, and configuration of the FPGA. You can choose to leave the RECONFIG_N pin floating or pull up the RECONFIG_N pin externally. From the release of Power-on Reset to just before the wake-up of the device, all GPIOs are high-impedance with internal weak pull-downs.

GW2AN-18X/9X operates on SRAM and embedded Flash according to the configuration data storage location and instructions.

SRAM Operation

The SRAM operations include read device ID CODE and USER CODE, read device status register information and SRAM configuration.

The device ID needs to be verified before configuration. Only the device with successful ID verification can be configured. The USER CODE is the identification number for users to distinguish between the devices that share the same ID CODE. The state register of the device records the status information before and after FPGA configuration, and you can use this information to analyze the state of the device accordingly. Please refer to Table 5-10 for the meaning of the status register. During SRAM configuration, only the bitstream data with no security bit setting supports validation. Data with security bit cannot be readback or verified.

Embedded Flash Operation

The embedded flash operations include erasing, programming and verification. The internal flash can only be programmed via JTAG, SSPI, QSSPI, and I^2C interface. The JTAG clock rate is no higher than 65 MHz, the SSPI and QSSPI clock rate is no higher than 100 MHz, and the I^2C clock rate is 100KHz~555KHz.

All the interface modes support transparent transmission. That is to say, the device can program the embedded Flash via the interfaces without affecting the current working state. During programming, the device works normally according to the previous configuration. After programming, the online upgrade can be completed by triggering RECONFIG_N at a low level or sending the RECONFIG command. This feature applies to the applications requiring long online time and irregular upgrades.

Configuration Pin Reuse

In different configuration modes, users need to ensure that FPGA works in the selected configuration mode according to the pin functions. If user pins is insufficient, these pins can be configured and used as GPIOs, but pins associated with data transmission need to be kept. MODE [1:0] is used to select the GowinCONFIG programming configuration MODE. MODE can be fixed through pull-up or pull-down resister. It is recommended to use 4.7 K pull-up resister and1 K pull-down resistor.

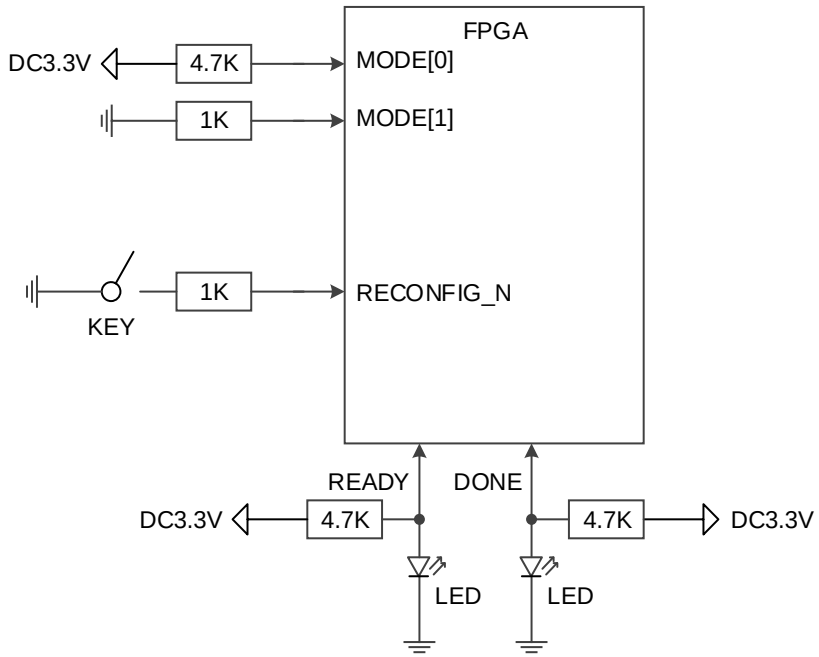
Note!

The RECONFIG_N, READY, and DONE pins are associated with each configuration mode. Whether they are set as GPIO or not, users should ensure that their initial value or pin connection state meets programming and configuration conditions before completing the configuration process.

Recommended Pin Connection

When users are designing a circuit schematic diagram, the recommended connection is as shown in Figure 5-1.

Figure 5-1 Recommended Pin Connection



Note!

- You can add a DIP switch to change the MODE value. Some MODE pins of devices are not all bonded out, please refer to *UG792, GW2AN-18X Pinout* and *UG978, GW2AN-9X Pinout* to get the MODE pin values;
- The values of READY and DONE signals have no meaningful reference in JTAG configuration.
- The unbonded RECONFIG_N, READY, and DONE pins have been internally handled, with no influence on the configuration function.

Timing for Power-on Again and Triggering RECONFIG_N at Low Pulse

Figure 5-2 and Figure 5-3 show the timing for power-on again or triggering RECONFIG_N at low pulse.

Figure 5-2 Power Recycle Timing

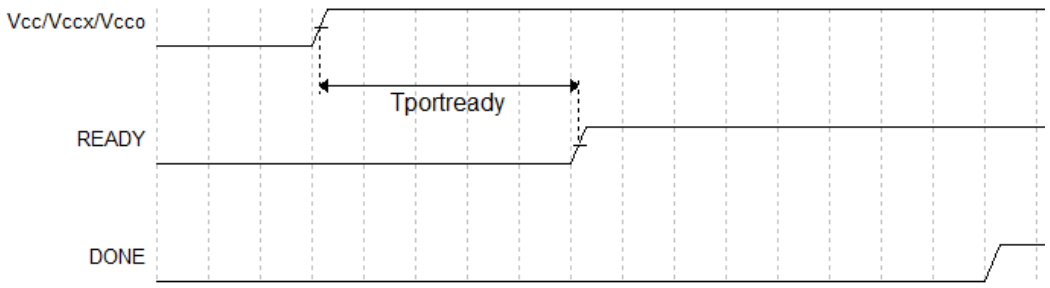
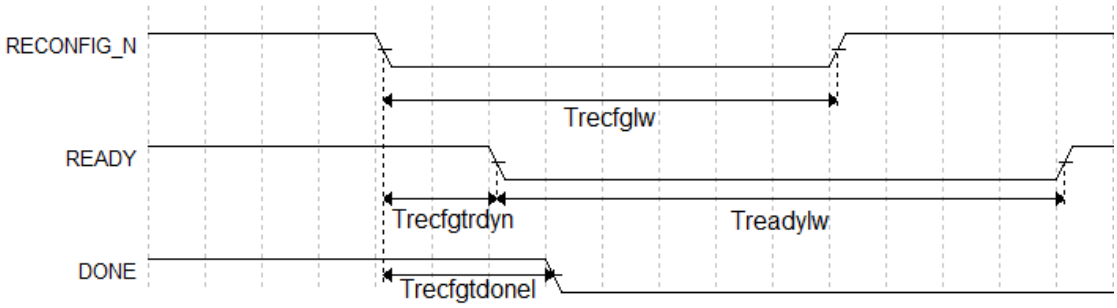


Figure 5-3 Trigger Timing



Timing parameters of the Arora Family of FPGA Products are shown in Table 5-1.

Table 5-1 Timing Parameters for Power Cycling and RECONFIG_N Triggering

Name	Description	Min.	Max.
$T_{portready}$(MODE0=0)	Time from POR to the rising edge of READY	-	35ms
$T_{portready}$(MODE0=1)	Time from POR to the rising edge of READY	-	10ms
$T_{recfglw}$	RECONFIG_N low pulse width	25ns	-
$T_{recfgtrdyn}$	Time from RECONFIG_N falling edge to READY low	-	70ns
$T_{readylw}$	READY low pulse width	TBD	-
$T_{recfgtdonel}$	Time from RECONFIG_N falling edge to READY low	-	80ns

5.2 AUTO BOOT Configuration

The AUTO BOOT mode is a configuration mode for momentary connection feature of non-volatile FPGA Products. In AUTO BOOT mode, after the chip is powered on, there is no need to connect to the external configuration interface, and the FPGA can complete the configuration by reading the bit stream data from the address 0x000000 of the internal Flash with the QuadSPI protocol. GW2AN-18X/9X supports two AUTO BOOT configurations. That is, when the AUTO BOOT configuration fails after power-on, the device automatically performs the second configuration from the address 0x100000. The factors that can lead to a failed configuration include false ID validation error, CRC check error, instruction error, and timeout error.

In the AUTO BOOT MODE, the bitstream data needs to be written to the internal Flash first. The chip will automatically read the bitstream data to complete configuration when it is powered up again or the RECONFIG_N pin is triggered at a low-level pulse. The momentary connection feature of the internal Flash saves download time and improves work efficiency.

MODE [1:0] needs to be set as “01” for AUTO BOOT mode. If the configuration fails, the device can still configure the SRAM or Flash using

JTAG, I2C, SSPI, QSSPI, etc.

5.3 JTAG Configuration

The JTAG configuration mode of Gowin FPGA products conforms to the IEEE1532 standard and the IEEE1149.1 boundary scan standard.

The JTAG configuration mode writes bitstream data to the SRAM of Gowin FPGA products. All configuration data is lost after the device is powered down. All Gowin FPGA products support the JTAG configuration mode.

5.3.1 JTAG Configuration Mode Pins

The relevant pins for the JTAG configuration mode are shown in Table 5-2.

Table 5-2 Pin Description in JTAG Configuration Mode

Pin Name	I/O	Description
JTAGSEL_N	I, internal weak pull-down	Revert JTAG pin from GPIO to configuration pin. Low active
TCK 1	I, internal weak pull-down	JTAG serial clock input
TMS	I, internal weak pull-up	JTAG serial mode input
TDI	I, internal weak pull-up	JTAG serial data input
TDO	O, internal weak pull-up	JTAG serial data output

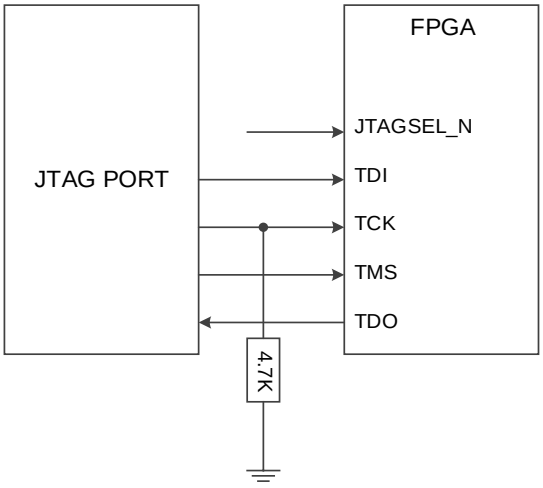
Note!

[1] TCK needs to connect 4.7 K pull down resister on the PCB.

5.3.2 Connection Diagram for the JTAG Configuration Mode

The connection diagram in the JTAG configuration mode is shown in Table 5-4.

Figure 5-4 Connection Diagram for JTAG Configuration Mode



Note!

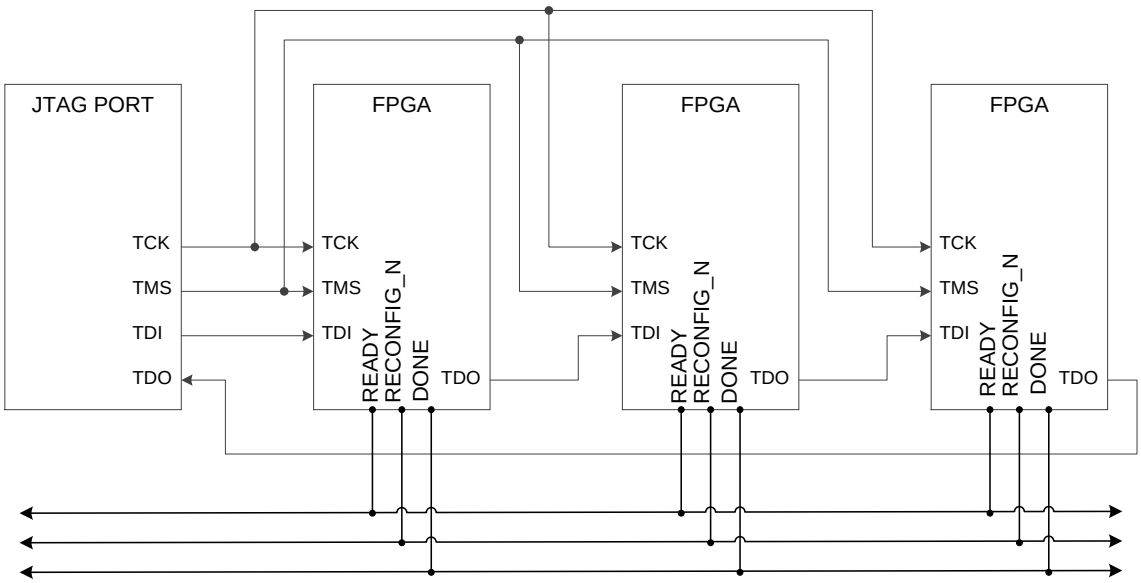
The clock frequency for JTAG configuration mode cannot be higher than 65MHz.

Gowin FPGA products support JTAG daisy chain operation, which connects the FPGA TDO pin to the next FPGA TDI pin. Gowin programming software will identify the connected FPGA devices automatically and configure them in turn. The connection diagram for the daisy chain configuration is shown in Figure 5-5.

Note!

To program the internal Flashes of the GW2AN-18X/9X FPGAs in a daisy chain, you need to use GAO-Bridge mode (sFlash Erase,Program,Verify thru GAO-Bridge). For details, please refer to SUG502, Gowin Programmer User Guide.

Figure 5-5 Connection Diagram of JTAG Daisy-Chain Configuration Mode



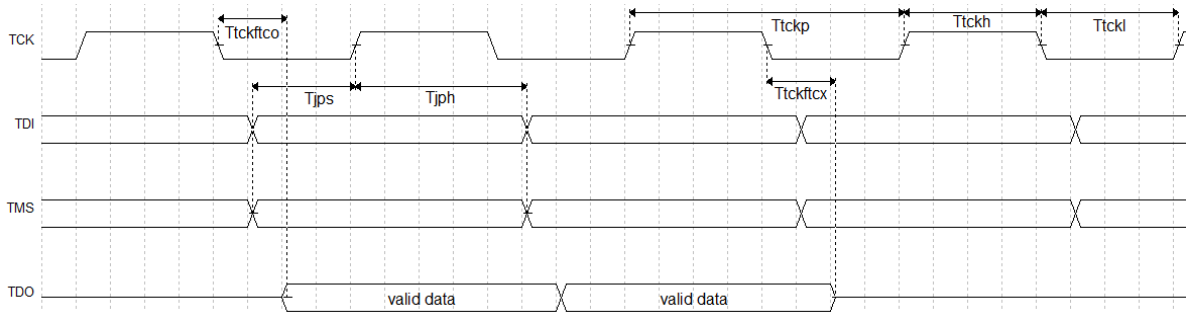
Note!

DONE, RECONFIG_N, and READY can be connected or not as appropriate.

5.3.3 JTAG Configuration Timing

See Table 5-6 for the timing of JTAG mode.

Figure 5-6 JTAG Configuration timing



See Table 5-3 for the description of timing parameters.

Table 5-3 JTAG Configuration Timing Parameters

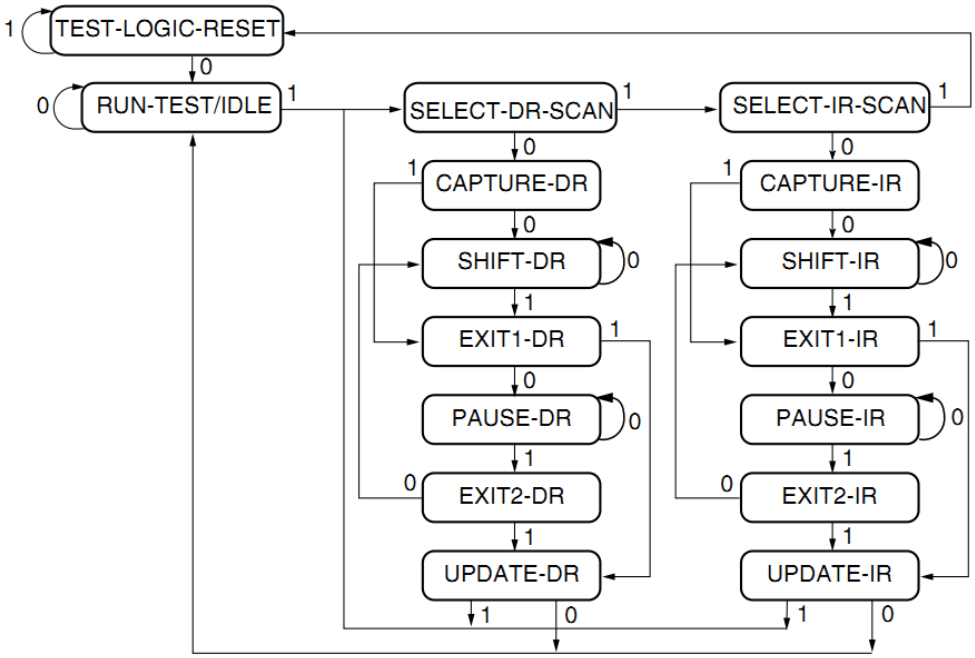
Name	Description	Min.	Max.
$T_{tckftco}$	Time from TCK falling edge to output	-	10ns
$T_{tckftcx}$	Time from SCLK falling edge to high impedance	-	10ns
T_{tckp}	TCK clock period	40ns	-
T_{tckh}	TCK clock high time	20ns	-
T_{tckl}	TCK clock low time	20ns	-
T_{jps}	JTAG PORT setup time	10ns	-
T_{jph}	JTAG PORT hold time	8ns	-

5.3.4 JTAG Configuration Process

TAP State Machine

The state machine for the test access port is designed to select an instruction register or a data register to connect it between TDI and TDO. In general, the instruction register is used to select the data register to be scanned. In the state machine diagram, the number on the side of the arrow indicates the logic state of the TMS when the TCK goes high, as shown in Figure 5-7. After writing a JTAG instruction, you need to move to the Run-Test-Idle state and stay there for at least 3 clock cycles.

Figure 5-7 TAP State Machine



TAP Reset

After TMS keeps high (logic "1") and at least 5 strobes are input (higher and then low) at the TCK terminal, the TAP logic is reset, the TAP state machine in other states is converted into the state of test logic reset, and the JTAG port and the test logic are reset.

Note!

The CPU and peripherals are not reset in this state.

Note!

- The data on the TDO is valid from the falling edge of TCK in the Shift_DR or Shift_IR state;
- The data is not shifted in the Shift_DR or Shift_IR state;
- The data is shifted when leaving the Shift_DR or Shift_IR;
- The first to be shifted is the least significant bit (LSB) of the data;
- Once reset, all instructions will be reset or disabled.

Instruction Register and Data register

In addition to the test logic reset, the state machine can also control two basic operations:

- Instruction register (IR) scan;
- Data Register (DR) scan.

During the IR scanning operation, in Shift_IR state, the data or instructions are sent to the IR in the LSB way. The lower data bits are sent first. The instructions will be all sent when the sate machine returns to Run-Test-Idle, as shown in Figure 5-8.

During the data register scanning operation, the data or instructions are sent to the DR in the Shift_DR state, as shown in Figure 5-9. The data is sent in LSB way or MSB way depending on specific operations.

Figure 5-8 Instruction Register Access Timing

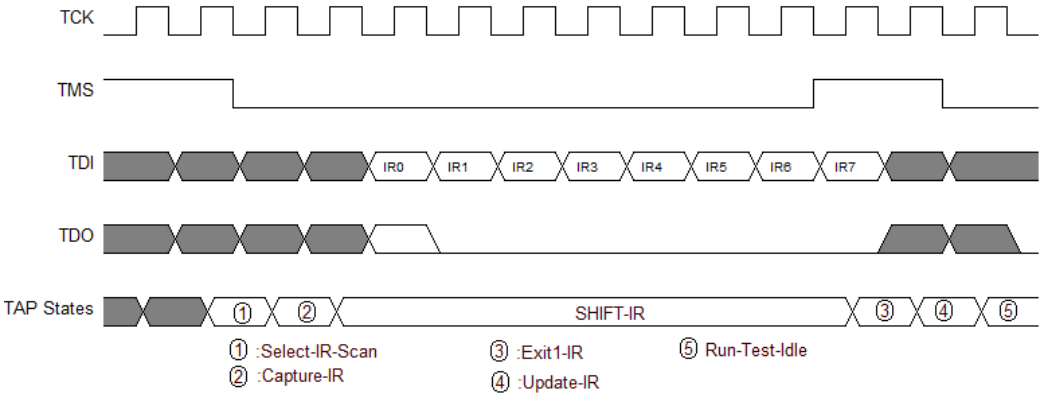
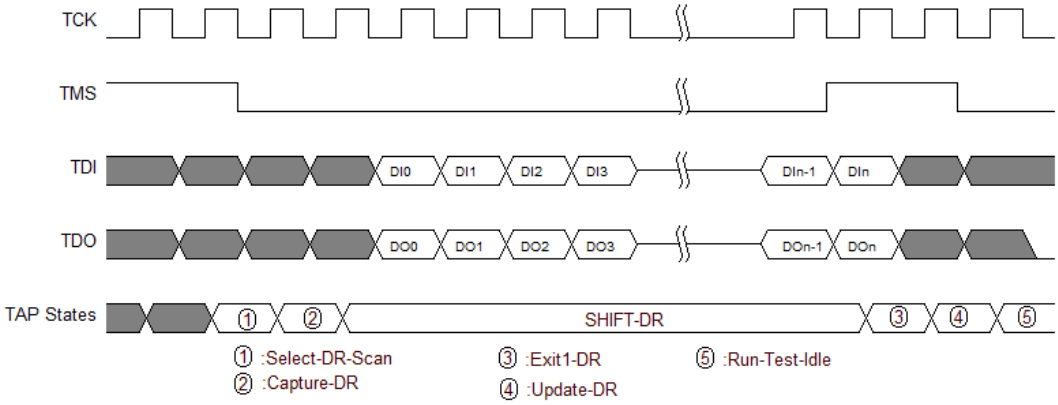


Figure 5-9 Data Register Access Timing



Note!

- The total length of the instruction register is 8 bits;
- The length of the data register can vary depending on the selected register.

Read ID CODE Instance

ID Code, i.e. JEDEC ID Code, is a basic identification of FPGA products.

The length of the Gowin FPGA ID Code is 32 bits. The ID Codes of the FPGA are listed in the following table.

Table 5-4 Gowin FPGA Device IDCODE

Gowin FPGA Device Family IDCODE			
Device Family	Device Part	Manufacturer ID	IDCODE
	Bits 31-12	Bits 11-0	
		h81B	
GW2AN-18X	h00004	h81B	h0000481B
GW2AN-9X	h00005	h81B	h0000581B

The instruction to read FPGA is 0x11. Take the GW2AN-9X ID Code as an example to illustrate the working mode of JTAG, please refer to the following steps:

1. TAP reset: TMS is set to high level and at least 5 clock cycles are continuously transmitted;
2. Move the state machine from Test-Logic-Reset to Run-Test-Idle;
3. Move the state machine to Shift-IR. Send Read ID 0x11 beginning with LSB. When MSB (the last bit) is being sent, move state machine to Exit1-IR at the same time, i.e., TMS should be high level before sending MSB. Table 5-5 lists the change of TDI and TMS value during sending 0x11 in 8-clock cycle, as shown in Figure 5-11.

Table 5-5 Change of TDI and TMS Value in The Process of Sending Instructions

	TCK 1	TCK 2	TCK 3	TCK 4	TCK 5	TCK 6	TCK 7	TCK 8
TDI value (0x11)	1	0	0	0	1	0	0	0
TMS value	0	0	0	0	0	0	0	1

4. Move the state machine, back to Run-Test-Idle after going from Exit1-IR to Update-IR, and then run the state machine at least 3 clock cycles in Run-Test-Idle.
5. Move the state machine to Shift-DR, send 32 clock cycles, and set TMS to high level before the 32nd clock is sent. When the 32 clock cycles are completed, jump from Shift-DR to Exit1-DR. During this period, sending 32 clocks can read 32 bits data, that is, 0x0000581B, as shown in Figure 5-12;
6. Move the state machine back to Run-Test-Idle.

Figure 5-10 Flow Chart of Reading ID Code State Machine

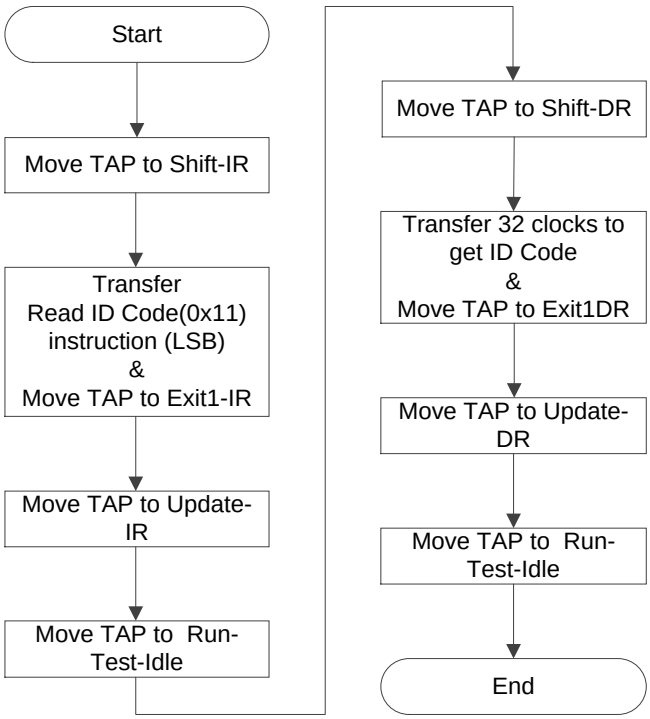


Figure 5-11 Access Timing of Reading ID Code Instruction - 0x11

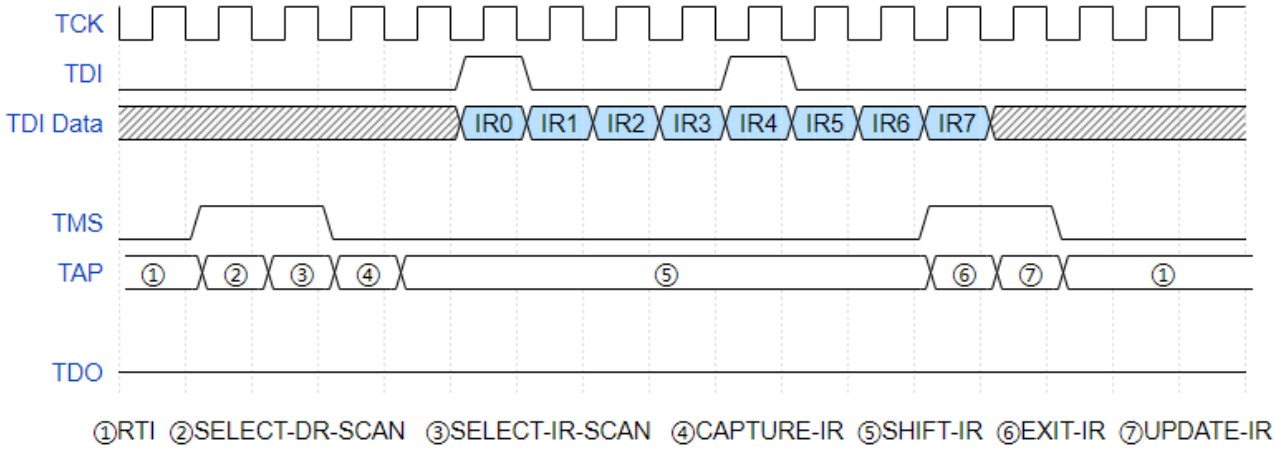
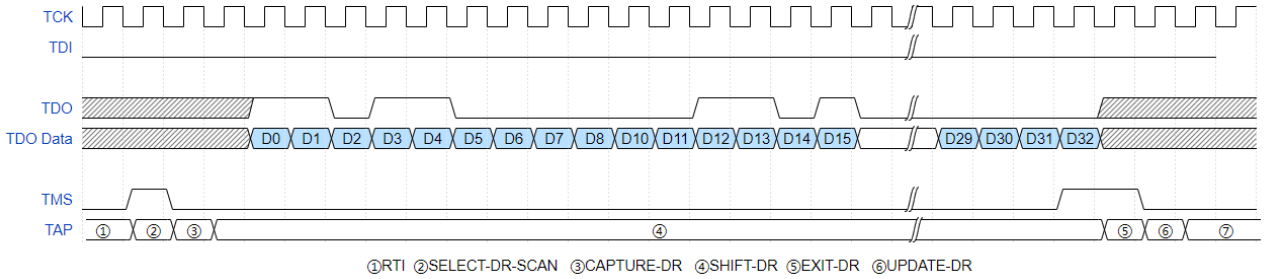


Figure 5-12 Access Timing of Reading ID Code Data Register



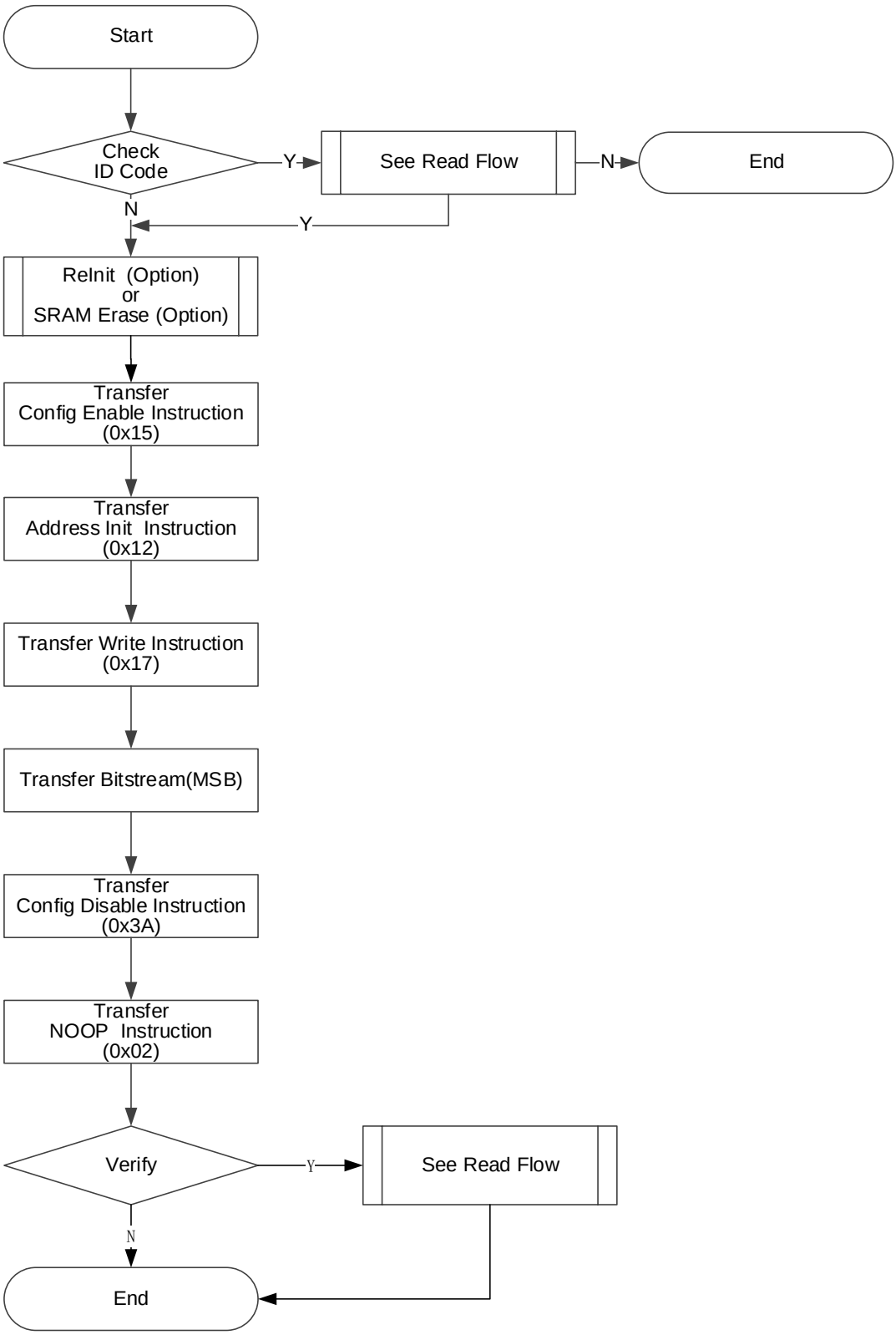
SRAM Configuration Process

You can use an external Host to configure the FPGA's SRAM through the JTAG interface, and JTAG configuration mode is supported regardless of the MODE value. After writing to the SRAM, you need to wait for 60ms for the status code to be refreshed.

Generate the FS file using Gowin software. Configure SRAM using JTAG. The process of SRAM configuration using the external Host is as follows, as shown in Figure 5-13.

1. Establish a JTAG link and reset TAP;
2. Read the device ID CODE and check if it matches.
3. Read the device StatusCode. If the Ready bit is 0, send the Reinit instruction "0x3F"; or
   erase the SRAM if it has been configured. Please refer to "SRAM Erasure Process".
4. Send the "0x15" instruction of ConfigDisble;
5. Send the "0x12" instruction of Address Initialize;
6. Send the "0x17" instruction of Transfer Configuration Data.
7. Move the state to Shift-DR (Data Register). Send Bitstream Data from the MSB bit by bit till all the bitstream file content is sent, and then return to run-test-idle state;
8. Send the "0x3A" instruction of ConfigDisabled;
9. Send the "0x02 " instruction of Noop to end the configuration process.
10. Please refer to Process of Reading SRAM (The process of reading SRAM) if read back Configuration Data is required for verification.

Figure 5-13 SRAM Configuration Flow



Process of Reading SRAM

Warning: SRAM data is not allowed to be read back by default for data security.

Read the SRAM data from the SRAM area of the FPGA. First ensure that the security bit is not configured when the data are written to the SRAM. The security bit is used to protect the runtime data and ensure the data security. After the safety bit is set, the data received from the SRAM are 1 (high level).

During loading, FPGA performs CRC check on the written data to ensure that the data is written correctly, and whether CRC reports an error can be used as a check mechanism to configure SRAM.

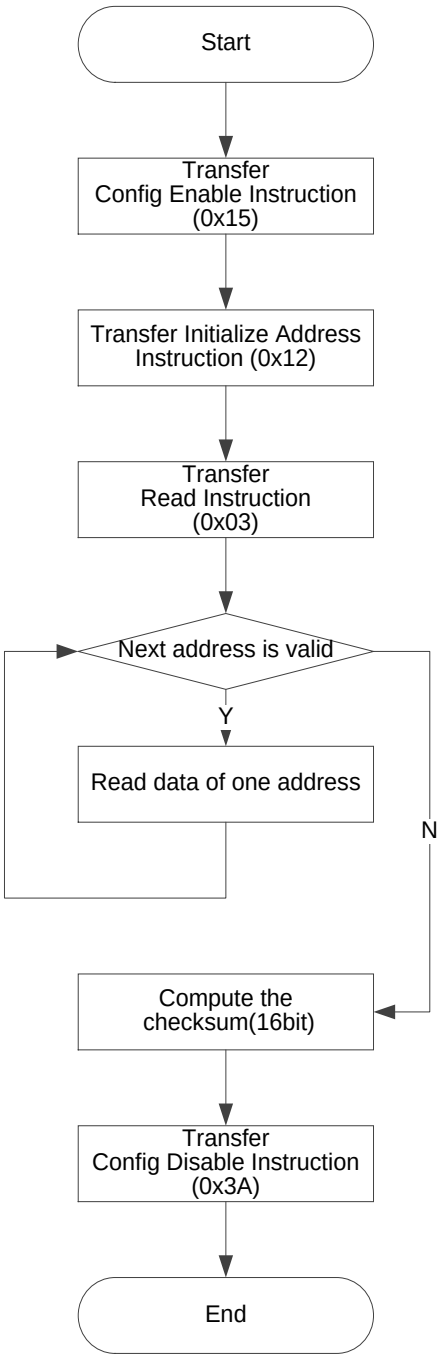
Table 5-6 Count of Address and Length of One Address

Device	Length of One Address (bits/address)	Count of Address
GW2AN-18X/9X	3376	1342

The reading process is described in detail below, as shown in Figure 5-14.

1. Send the "0x15" instruction of ConfigDisble;
2. Send the "0x12" instruction of Address Initialize;
3. Send the "0x 03" instruction of SRAM Read;
4. Move the state machine to Shift-DR (data register) and send it to one clock period with one SRAM length, see Table 5-6. When the last clock is sent, pull up TMS at the same time. The state machine jumps to Exit1-DR, and TDO reads data with corresponding length. The state machine will back to Run-Test-Idle state finally.
5. Repeat the step 4, the address will be automatically accumulated when the data of an address are read each time;
6. Send the "0x3A" instruction of ConfigDisabled;
7. Send the "0x02 " instruction of Noop to end the configuration process.

Figure 5-14 Process of Reading SRAM



SRAM Erasure Process

When reconfiguring SRAM, the existing SRAM needs to be erased. The flow is as follows:

1. Send the "0x15" instruction of ConfigDisble;
2. Send the "0x05" instruction of SRAM Erase;
3. Send the "0x02 " instruction of Noop;
4. Delay or Run Test 2~10ms;
5. Send the "0x09" instruction of SRAM Erase Done;
6. Send the "0x3A" instruction of ConfigDisabled;

7. Send the "0x02 " instruction of Noop to end the configuration process.

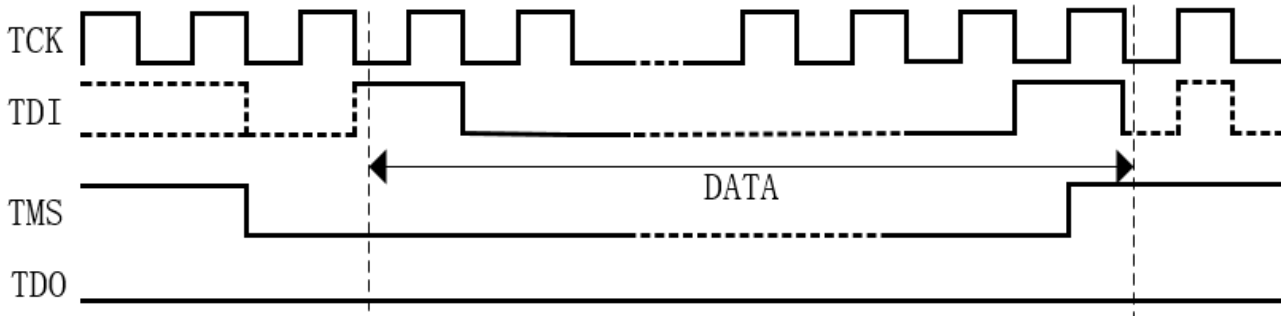
Note!

- You need to wait enough time for the device to finish erasing after the instructions of EraseSram(0x05) and Noop(0x02) are sent.
- The reference time for GW2AN-18X/9X is 6ms.

Embedded Flash Programming Modes

16 Mbit Serial Flash is embedded in GW2AN-18X/9X devices. JTAG provides an SPI-like protocol to program the embedded Flash. The maximum operating frequency is 65Mhz.

The SPI-like protocol provided by JTAG has the same logic with the standard SPI protocol. TMS corresponds to ChipSelect (/CS), TDI corresponds to DI, TCK corresponds to SerialClock (CLK), and TDO corresponds to DO. For the timing, you can also refer to that of the standard SPI protocol, with only TDI data shifted backward by one clock cycle. When TMS is pulled down, the DATA is valid only after a clock is sent, and the last BIT of DATA should be pulled up together with TMS, as shown in the figure below:

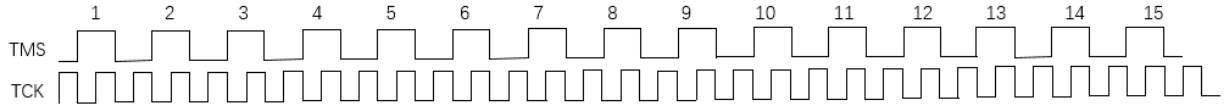


Enable Flash Programming Mode

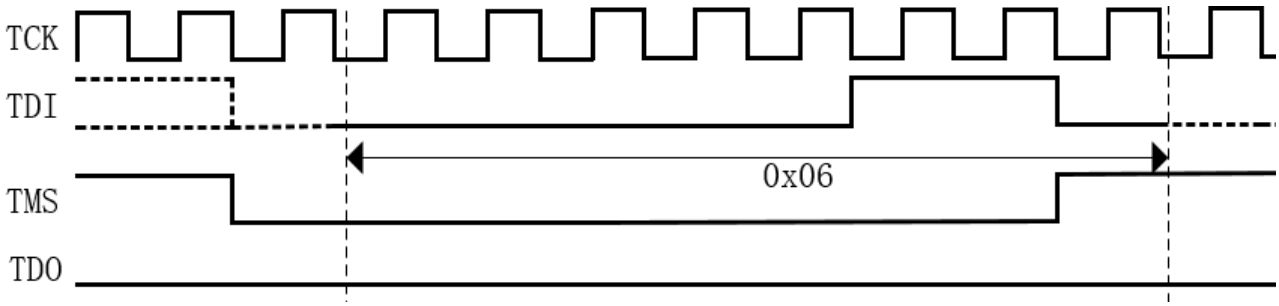
Before Flash configuration, connect the JTAG interface to the internal Flash controller and send the standard JTAG instruction 0x16 to enable the mode. After this mode is enabled, the JTAG interface works only in the Flash configuration mode and is converted to an SPI-like protocol, and does not support JTAG instructions.

Exit Flash Programming Mode

After you set the TMS high and low level for 15 times continuously, you can exit the Flash programming mode, that is, exit the SPI-like interface, back to the JTAG standard interface, and back to the JTAG configuration mode. As shown in the figure below:



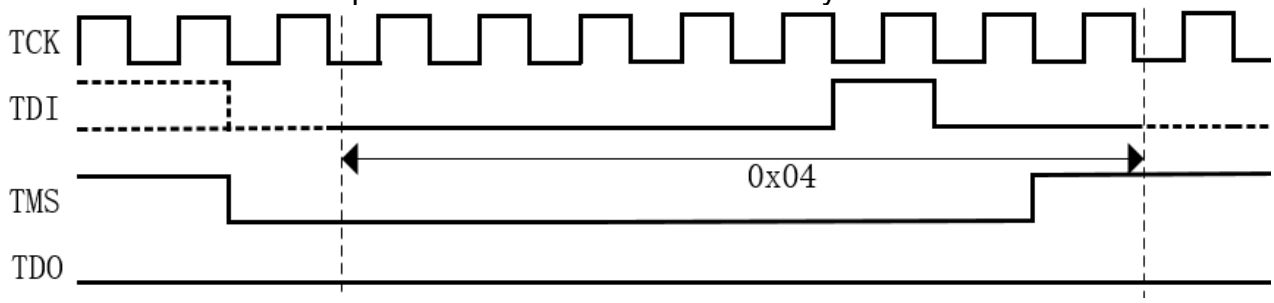
Embedded Flash Instruction -WriteEnable（0x06）



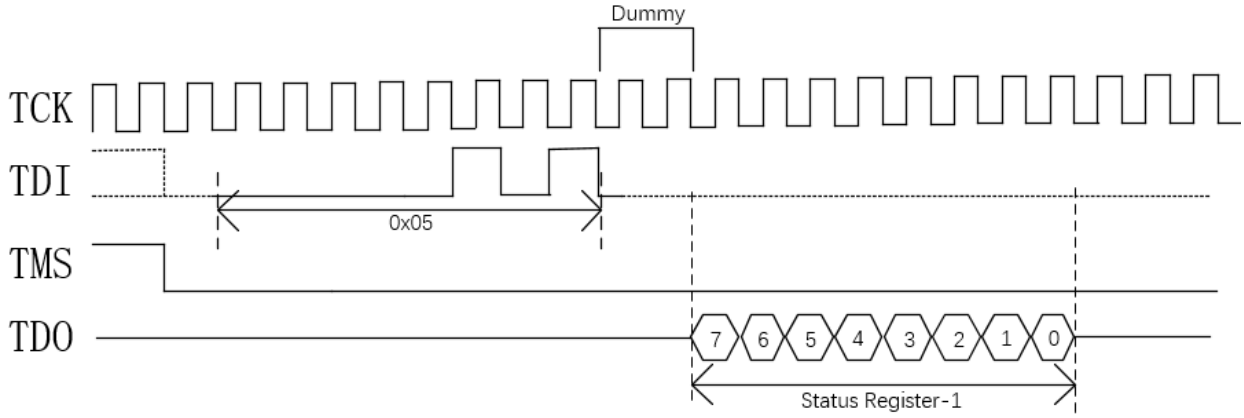
WriteEnable (above) is used to set WriteEnable Latch (WEL) bit of Flash Status Register. The WEL bit must be set before each Page -program, Sector Erase, and Chip Erase. When TMS is low and after one clock cycle (TCK), the instruction "0x06" is shifted to the data input (TDI) pin. The last bit "0" of 0x06 completes to shift in the same clock cycle with the TMS.

Embedded Flash Instruction -WriteDisable（0x04）

WriteEnable (below) is used to reset WriteEnable Latch (WEL) bit of Flash Status Register. When TMS is low and after one clock cycle (TCK), the instruction "0x04" is shifted to the data input (TDI) pin. The last bit "0" of 0x04 completes to shift in the same clock cycle with the TMS.



Embedded Flash Instruction - Read Status Register-1（0x05）



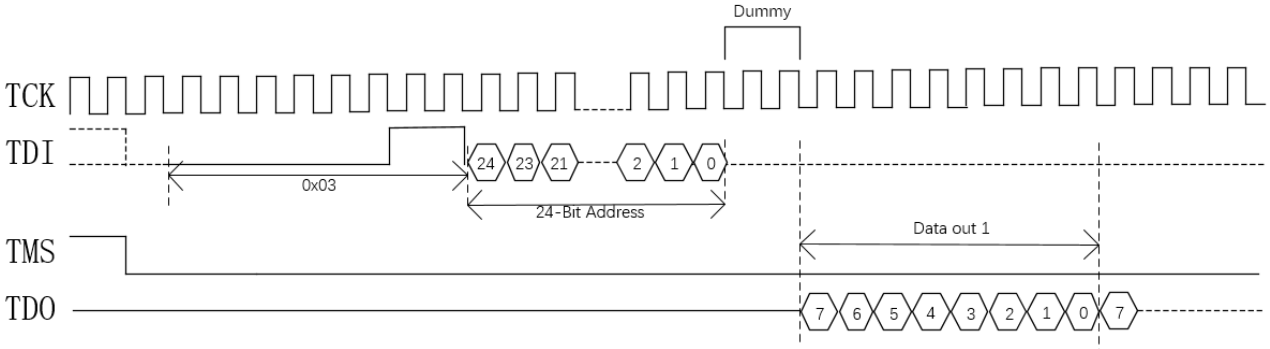
Read Status Register-1 is used to read an 8-bit status register. When TMS is low and after one clock cycle (TCK), the "0x05" instruction will be shifted to the data input (TDI) pin at the rising edge of TCK. After two more

dummy clocks, the data of Status Register-1 will be outputted from the TDO pin at the falling edge of TCK, with the most significant bit (MSB) first. As shown in the figure below.

Read Status Register-1 contains eight bits, represented by S[7:0]. S[0] indicates the BUSY bit. If the Flash is in the process of Page-Program, Chip Erase, or Sector Erase, bit 1 is automatically set to S[0]. In this state, Flash is BUSY and does not respond to other instructions. When the operation is complete, S[0] automatically resets the bit to 0 and allows to send other instructions.

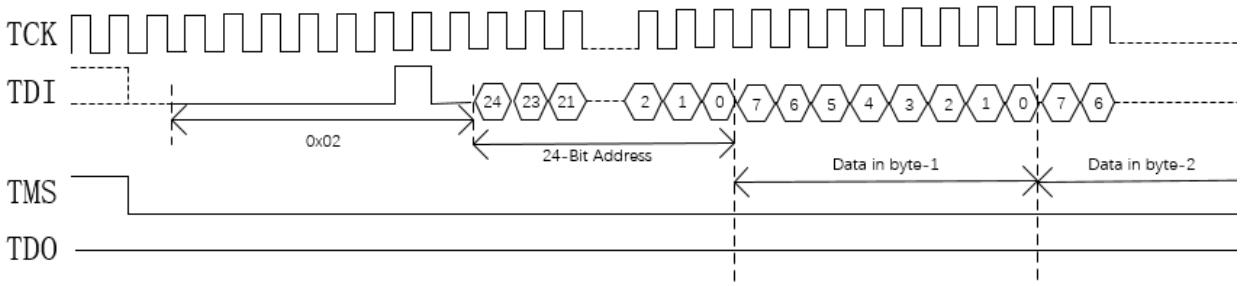
Embedded Flash Instruction - Read Data（0x03）

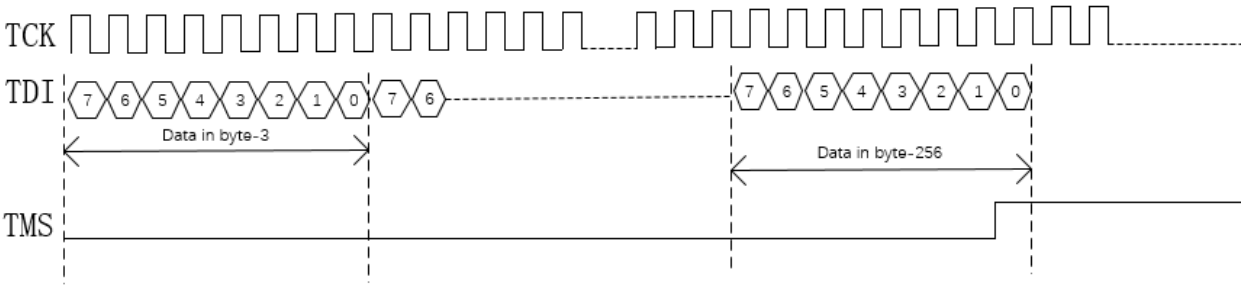
Read Data is used to read continuous data from Flash. When TMS is low and after one clock cycle (TCK), the "0x03” instruction will be shifted to the data input (TDI) pin at the rising edge of TCK, and then shift the 24 bits address to the data pin (DI). After two more dummy clocks, the data of the corresponding address will be outputted from the TDO pin at the falling edge of TCK, with the most significant bit (MSB) first. When the number of data read back exceeds one address, the address is automatically increments, allowing continuous data streams to be read back. That is, a single instruction can read the entire Flash data. As shown in the figure below.



Embedded Flash Instruction - Page Program（0x02）

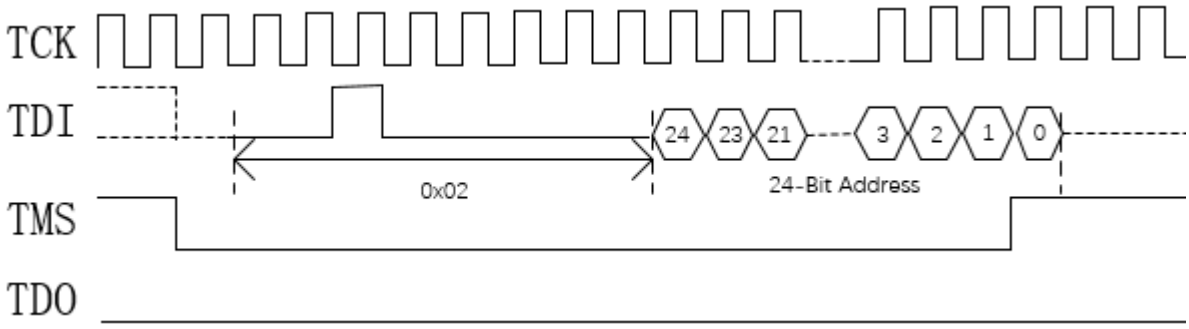
Page Program is used to program one or more bytes on a Page that have been erased (0xFF). The WriteEnable directive must be completed before configuration. When TMS is low and after one clock cycle (TCK), the "0x02” instruction will be shifted to the data input (TDI) pin at the rising edge of TCK, shift the 24 bits address to the data pin (DI), and then the data is shifted into the data (DI) pin in MSB way. TMS stays low during this period and the last bit data of the last byte will shift to the data pin (DI) in the same clock cycle as the TMS is pulling up. As shown in the figure below.


TCK
TDI
TMS
TDO
0x02
24-Bit Address
Data in byte-1
Data in byte-2


TCK
TDI
TMS
Data in byte-3
Data in byte-256

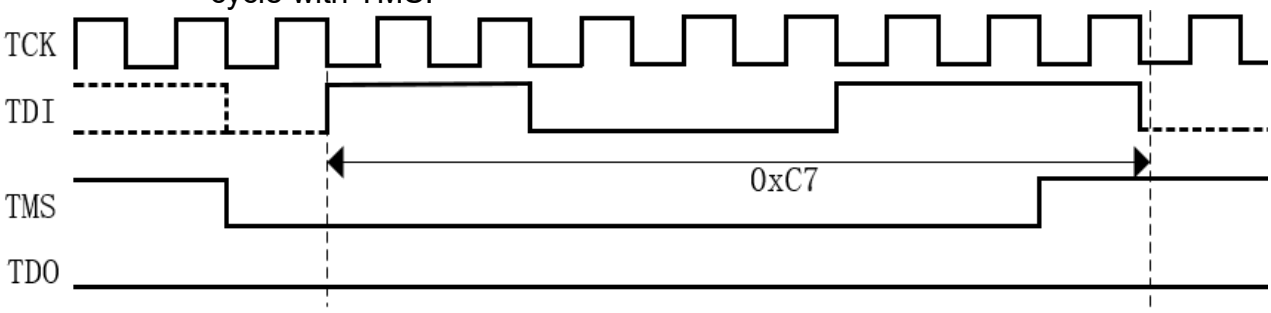
Embedded Flash Instruction - Sector Erase（0x20）

Sector Erase is used to erases all data in a specified Sector (4Kbytes). After the erasing is complete, the Flash data is restored to the 0xFF state. The WriteEnable instruction must be completed before sending this instruction. When TMS is low and after one clock cycle (TCK), the "0x20” instruction will be shifted to the data input (TDI) pin at the rising edge of TCK, shift the 24 bits address to the data pin (DI), and then set TMS as high at the last bit of the address.

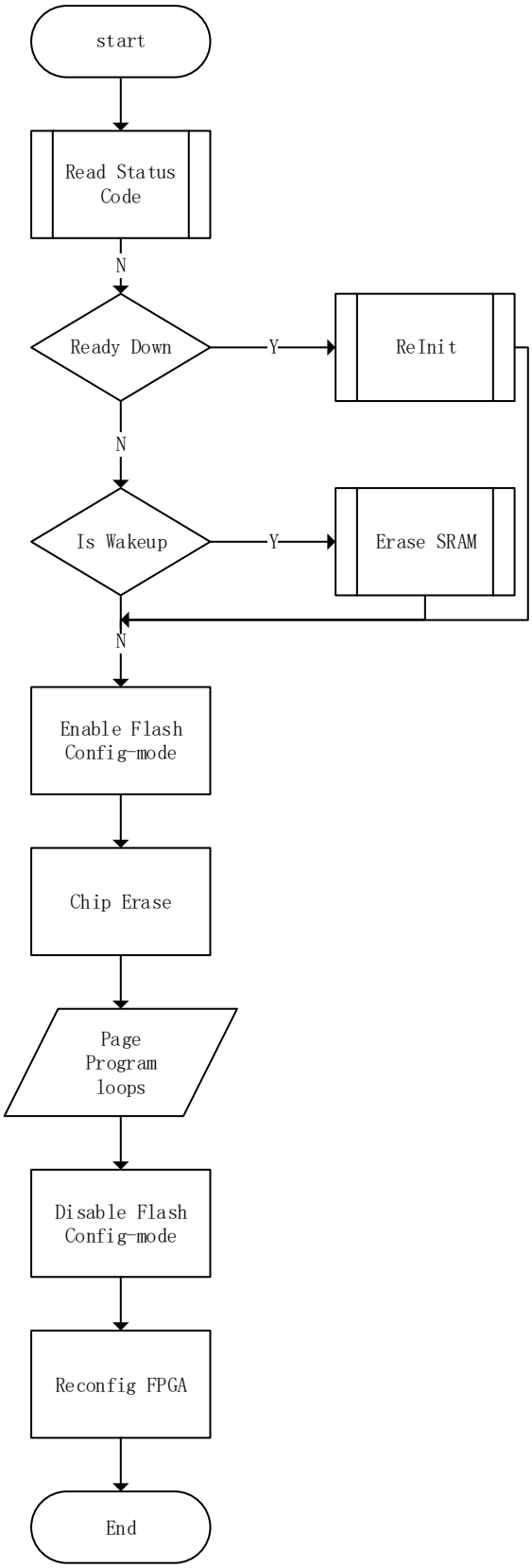


Embedded Flash Instruction - Chip Erase（0xC7/0x60）

Chip Erase is used to erase all data in Flash. After erasing, all data is 1 (0xFF). The WriteEnable instruction must be completed before sending this instruction. When TMS is low and after one clock cycle (TCK), the "0xc7” instruction will be shifted to the data input (TDI) pin at the rising edge of TCK, and the last bit "1" of 0xC7 is completed in the same clock cycle with TMS.



Flow Chart of Configuring the Embedded Flash



Program SPI Flash in JTAG Boundary Scan Mode

The principle of this mode is changing the state of the pins connected to SPI by using Boundary Scan method to implement SSPI timing, and then to program the internal Flash.

The length of the Boundary Scan Chain used in this mode is 8 bits. Every 2 bits combination corresponds to the pin state, as shown in Table 5-7. Completes one SCLK drive once the Boundary Scan Chain is sent twice.

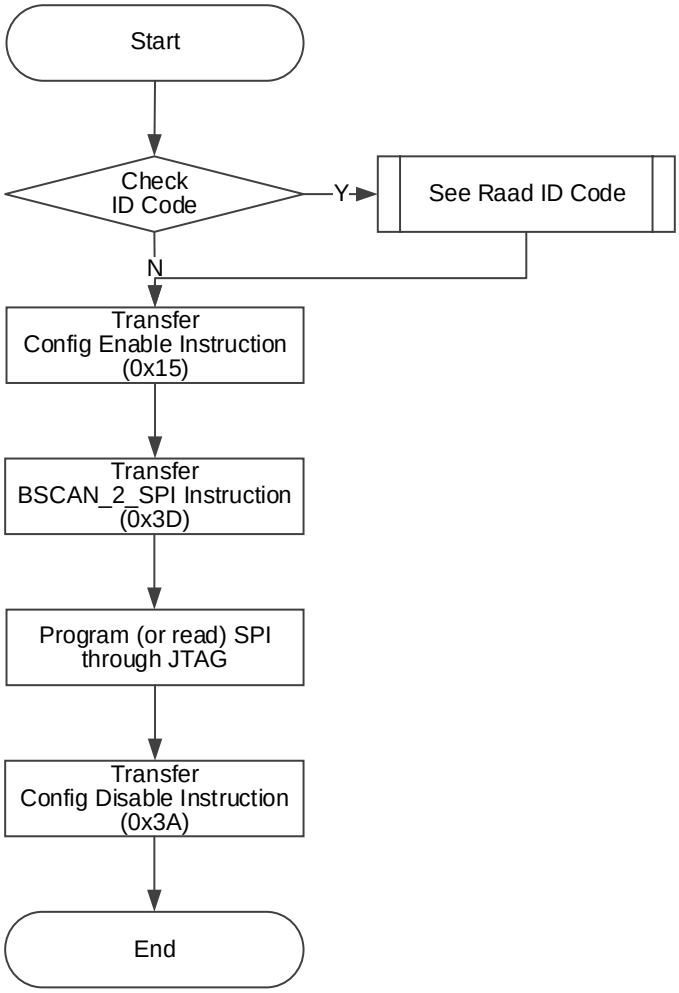
Table 5-7 Pin State

Pins Name of SPI Flash	SCLK		CS		DI		DO	
Bscan Chain[7:0]	7	6	5	4	3	2	1	0
(ctrl & data)	0		0		0		1	

Note!

- ctrl:0 means output, 1 means input;
- data:0 means low, 1 means high.

Figure 5-15 Process of Use Boundary Scan Mode to Program SPI Flash



Read Status Register 0x41

Status Register is of great help in device debugging and observing device Status. Reading Status Register can preliminarily judge the Status of devices, such as whether wakeup is successful or not, whether there is a loading error, etc. The Status Register is 32 bits, the read instruction is 0x41 and the timing is the same as that of Read ID Code.

The meaning of the Status Register is shown in Table 5-8.

Table 5-8 Status Register Definition

Device / Status Register[31:0]	GW2AN-18X/9X
0	CRC Error Flag(Active High)
1	Bad Command Error Flag(Active High)
2	Verify IDCODE Failed Flag(Active High)
3	Timeout Error Flag(Active High)
4	Autoboot2nd Failed Error Flag(Active High)
5	Memory Erase
6	Preamble
7	Edit Mode
8	Program SPI Directly
9	Autoboot1st Failed Error Flag(Active High)
10	Non-jtag Active
11	Bypass State
12	I^2C Flag
13	Done Final State
14	Security Bit Final State
15	Encryption Format(1: Encryption Enabled, 0: Encryption Disabled)
16	Encryption Key Match(1: Correct Key, 0: Wrong Key)
17	sspi_mode,
18-31	0

Clearing Status Code Errors

Before configuring the SRAM or programming the Flash, please make sure that there are no status code errors. If there are status code errors, follow the steps below to clear them:

1. Send ISC-NOOP(0x02), ISC-ENABLE(0x15), reprogram(0x3C), and

ISC-NOOP(0x02) instructions in sequence.
2. Delay 100ms.
3. Send ISC-DISABLE(0x3A) and ISC-NOOP(0x02) instructions in sequence.
4. Delay 100ms.

Reading User Code (0x13)

The user code is 32 bits, the read instruction is 0x13 and the timing is the same as that of Read ID Code.

The user code adopts the checksum value in the FS file by default. It can be redefined using Gowin Designer.

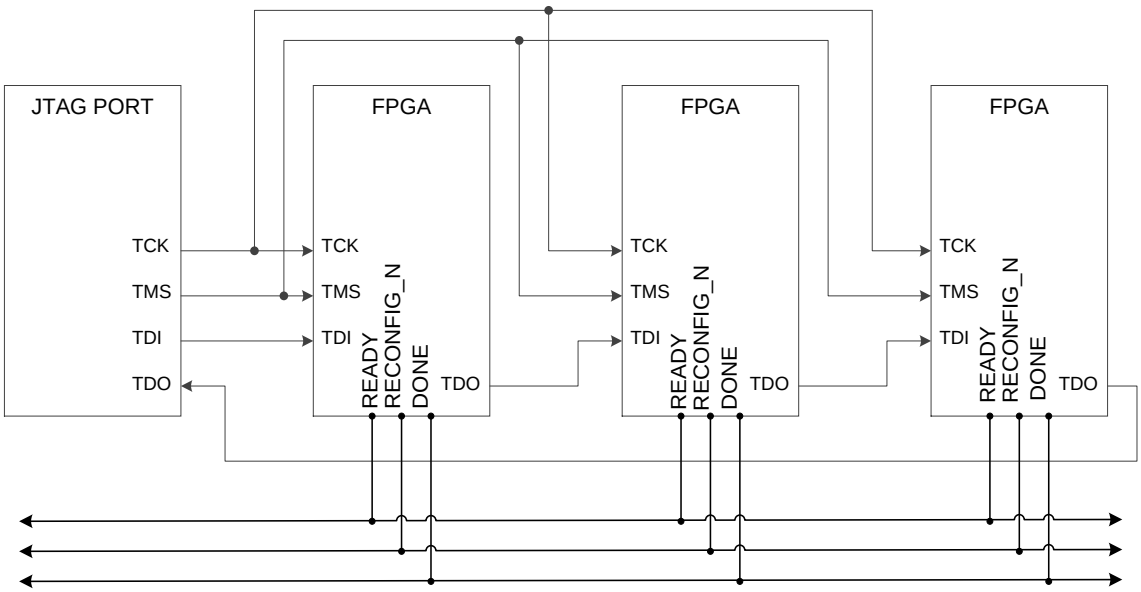
Reprogram (0x3C)

This instruction is used to read the bitstream files from Flash and write to SRAM.

Send the instructions of Reprogram/Reconfig (0x3C) and Noop (0x02) to reload the device via JTAG. You can also reload the device by triggering the Reconfig_N pin.

Connection Diagram of Daisy-Chain

Figure 5-16 Connection Diagram of Daisy-Chain



5.4 SSPI

In SSPI (Slave SSPI) mode, FPGA is as a slave device and is configured via SPI by an external Host.

5.4.1 SSPI Mode Pins

The SSPI configuration pins are shown in Table 5-9.

Table 5-9 SSPI Mode Pins

Pin Name	I/O	Description
RECONFIG_N	I, Internal weak pull-up	Low level pulse: Start GowinCONFIG
READY	I/O, Internal weak pull-up	High level: FPGA can be programmed and configured Low level: Programming configuration for FPGA is prohibited
DONE	I/O, Internal weak pull-up	High-level: Successfully programmed and configured; Low-level: Programming and configuration uncompleted or failed.
MODE[1:0]	I, Internal weak pull-down	Configuration mode selection, READY rising edge sampling
SCLK	I, Internal weak pull-up	Input clock
CLKHOLD_N	I, Internal weak pull-down	Active high
SO	O, Internal weak pull-down	FPGA outputs data to Host
SI	I, Internal weak pull-down	Input data to FPGA from Host
SSPI_CS_N	I, Internal weak pull-up	SSPI Chip selection signal, active low.

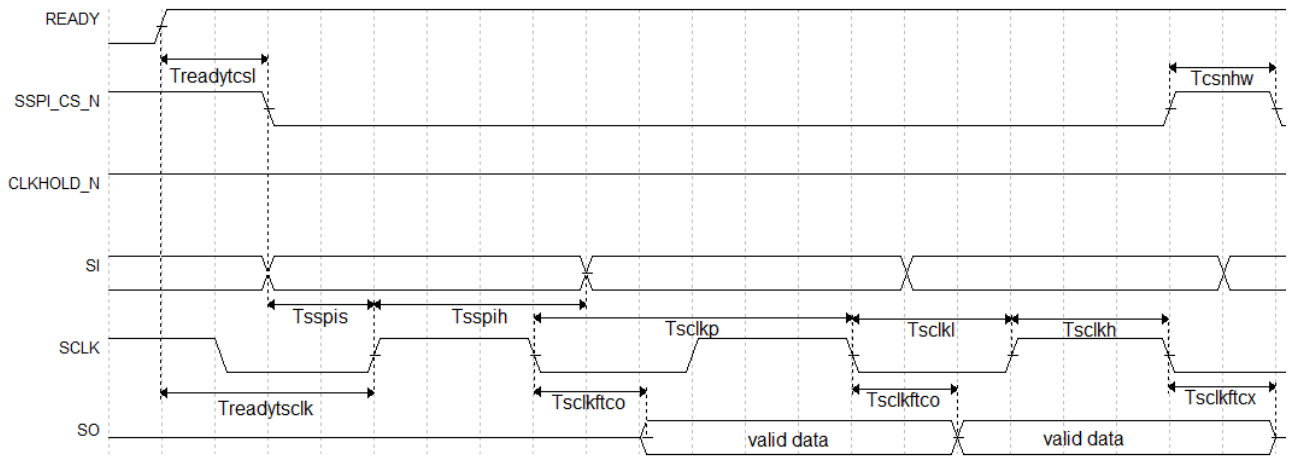
Note!

CLKHOLD_N is internal weak pull-down by default. Set CLKHOLD_N to high when using SSPI.

5.4.2 SSPI Configuration Timing

See Figure 5-17 for the SSPI timing.

Figure 5-17 SSPI Configuration Timing



See Table 5-10 for the SPPI configuration timing parameters.

Table 5-10 SSPI Configuration Timing Parameters

Name	Description	Min.	Max.
T_{sclkp}	SCLK clock period	15ns	-
T_{sclkh}	SCLK clock high time	7.5ns	-
T_{sclkl}	SCLK clock low time	7.5ns	-
T_{sspis}	SSPI PORT setup time	2ns	-
T_{sspih}	SSPI PORT hold time	0ns	-
$T_{sclkftco}$	Time from SCLK falling edge to output	-	10ns
$T_{sclkftcx}$	Time from SCLK falling edge to high impedance	-	10ns
T_{csnhw}	CSN high time	25ns	-
$T_{readytcsl}$	Time from READY rising edge to CSN low	TBD	
$T_{readytsclk}$	Time from READY rising edge to first SCLK edge	TBD	-

Other than the power requirements, the following conditions need to be met to use the SSPI configuration mode:

- SSPI port enable
  RECONFIG_N is not set as a GPIO during the first configuration after power up or the previous programming.
- Initiate new configutation
  Power up again or trigger RECONFIG_N at one low pulse.

5.4.3 SSPI Configuration Instruction

In Slave SPI mode, FPGA SRAM can be configured via SSPI or reading ID information on ID CODE\USER CODE\STATUS CODE in SRAM. External memory can also be programmed (Such as SPI Flash).

The SSPI instruction of FPGA is generally composed of 1-4 bytes, including at least 1 instruction class byte and multiple redundant information bytes. If there is no specified information byte, the redundant information byte can be any number (0x00 is used in the following table).

Table 5-11 Configuration Instruction

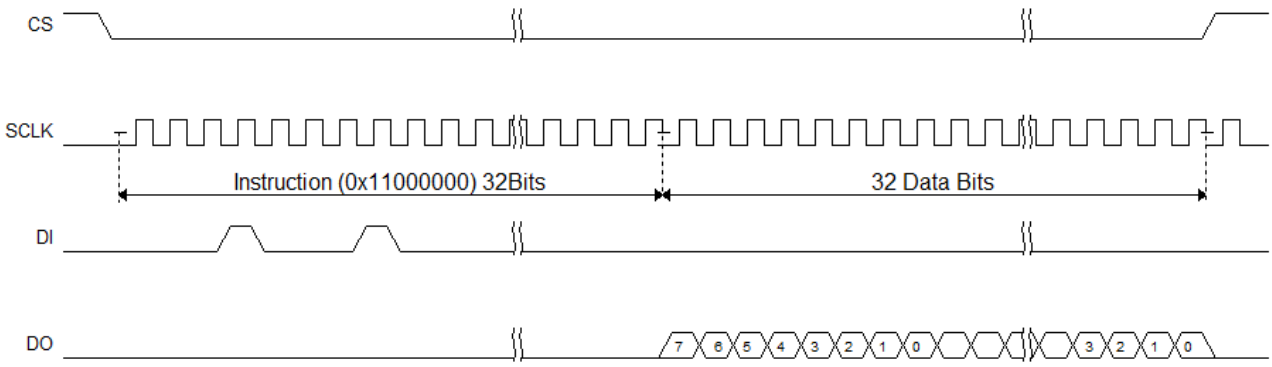
Name	Complete Instruction (Instruction Byte + Redundant Information Byte)
Read ID Code	0x11000000
Read User Code	0x13000000
Read Status Code	0x41000000
Reconfig/Reprogram	0x3C00
Write Enable	0x1500
Write Disable	0x3A00
Write Data	0x3B
Write Data with Quad SPI	0x6B
Program SPI Flash	0x1600
Init Address	0x1200
Erase SRAM	0x0500
Reinit	0x3F00

Read ID Code

The length of FPGA ID Code is 32bits. The instruction to read ID is four Bytes, that is 0x11000000. Before sending instructions, keep CS at a high level and generate multiple clocks (more than two) to let FPGA to get CS state.

After CS is pulled down, the instruction of 0x11000000 is written in in MSB way and after this, 32 clocks are generated continuously. At this time, the ID CODE data will be successively shifted out of DO in the form of MSB.

Figure 5-18 Read ID Code Timing

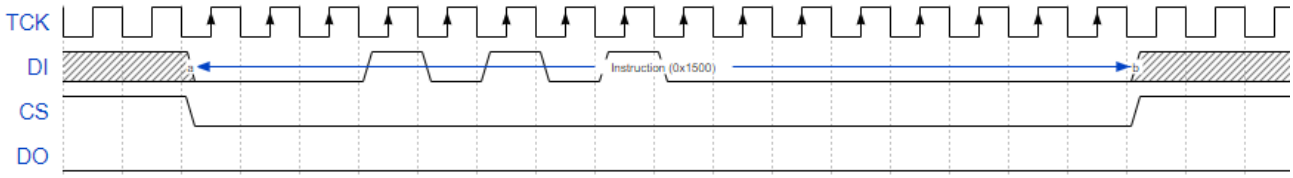


Operations of reading StatusCode / UserCode are similar with the operation of reading ID Code, simply replacing the related instructions.

Write Enable (0x1500)

Before configuring SRAM (write Features), enter programming mode using “Write Enable (0x15)” instruction to receive the “WriteData (0x3B)” write data instructions.

Figure 5-19 Write Enable (0x15) Timing



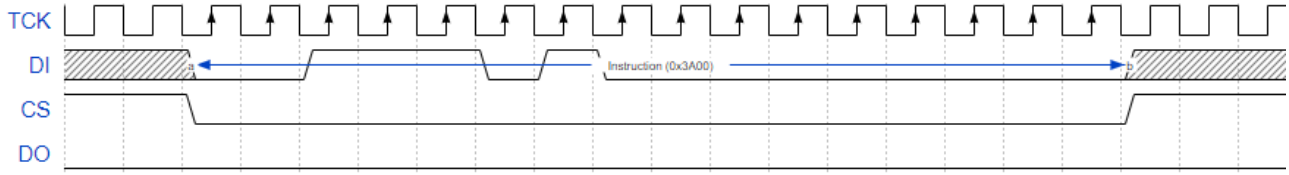
Note!

At CS high level, more than two clocks should be given to SCLK to drive FPGA to identify CS signal. This rule also applies to other instructions.

Write Disable (0x3A00)

After data is sent, you can run the Write Disable command to exit editing mode. After exiting, the device can be awakened to enter the working state.

Figure 5-20 Write Disable(0x3A00) Timing



The timing of 0x1500 and 0x3A00 is basically the same. Instructions start at CS low level and the CS is pulled up after the instruction transmission is completed. Instructions following this timing are as follows: 0x3C00 (Reconfig / Reprogram), 0x1500(Write Enable), 0x3A000 (Write Disable), 0x1600(Program SPI Flash), 0x1200(Init Address), 0x0500(Erase SRAM).

In addition, SSPI is driven by an external clock, so if CS is at high before and after these instructions, more than two clocks are needed to enable FPGA to collect the STATE of CS.

Erase SRAM (0x0500)

The instruction timing is consistent with that of WriteEnable/WriteDisable, and only the instruction content needs to be replaced with “0x0500”.

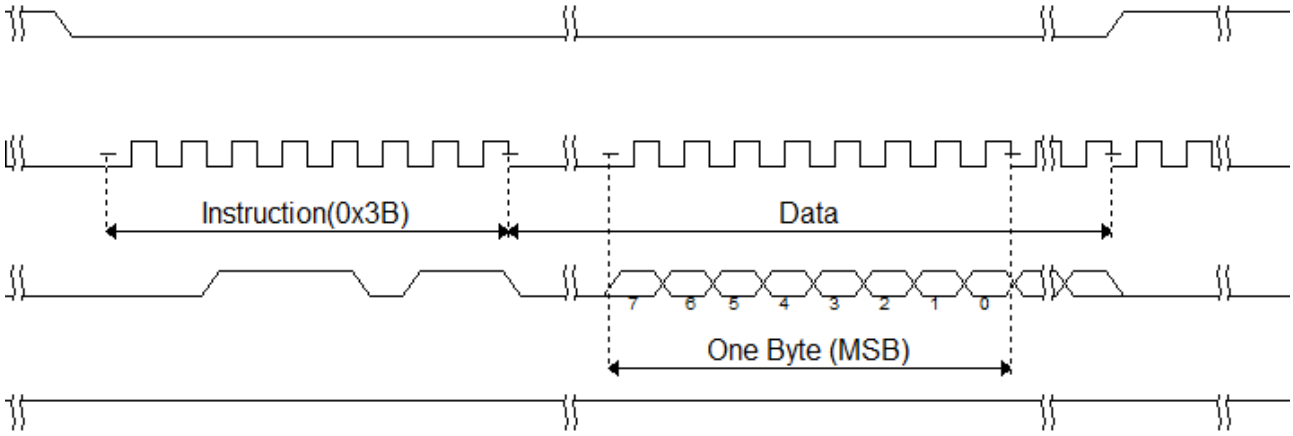
When the instruction is sent, at least 10ms is required to execute it.

Write Data (0x3B)

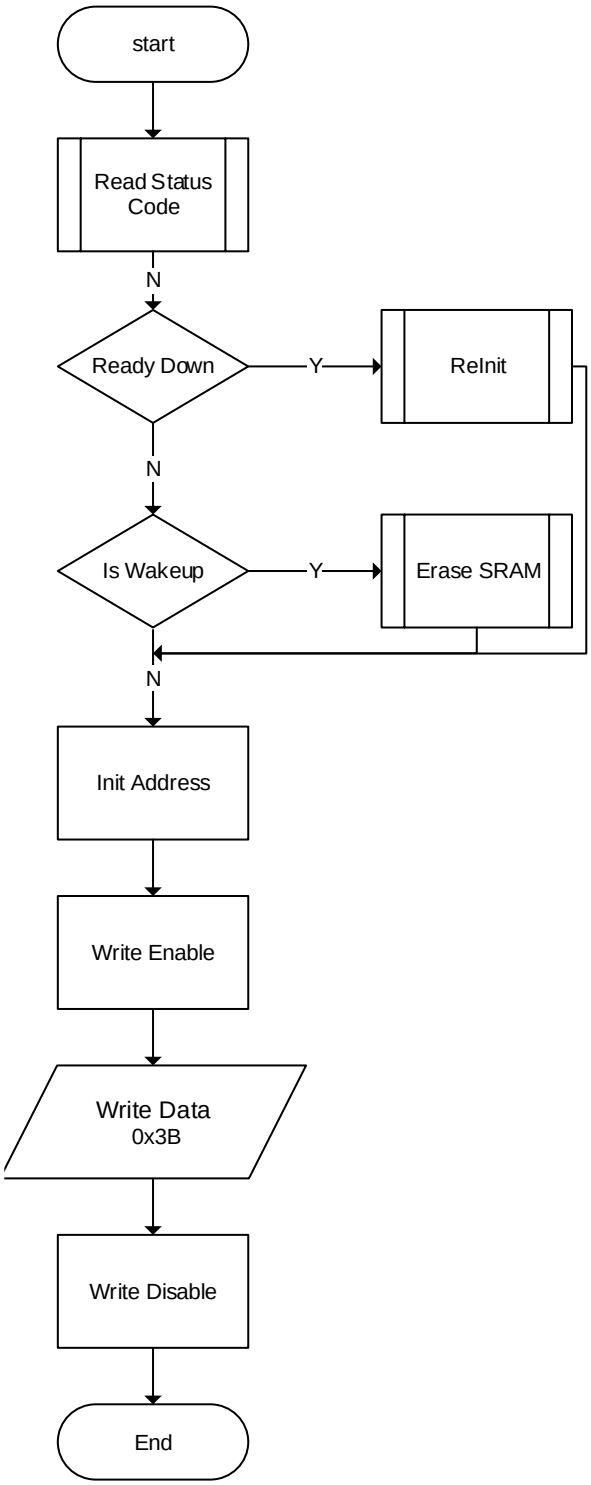
The fs file is sent directly to the FPGA device using the “WriteData (0x3B)” instruction.

Note that CS keeps low level in the process of data writing.

Figure 5-21 Write Data (0x3B) Timing



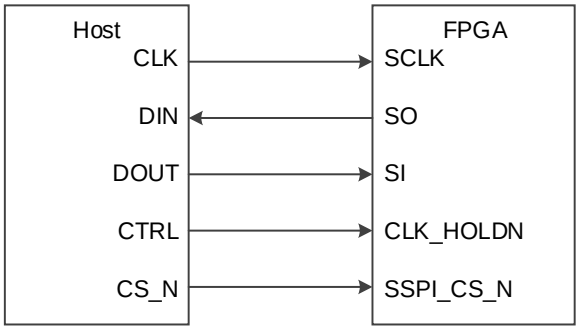
5.4.4 SSPI Configure SRAM Flow



5.4.5 Connection Diagram for SSPI Configuration Mode

The connection diagram for configuring Gowin FPGA products via SSPI is shown in Figure 5-22 .

Figure 5-22 SSPI Configuration Mode Connection Diagram



Note!

The figure above shows the minimum system diagram of the SSPI MODE. The connection for the other fixed pins are shown in Figure 5-1.
CLKHOLD_N is internal weak pull-down by default. Set CLKHOLD_N to high when using SSPI.

In addition to SRAM, SSPI can be used to program the internal SPI Flash. The MODE value of the Flash programming is the same as the MODE value of SSPI configuration mode. Configuration data can be written to SRAM or an internal Flash using Gowin Programmer. Before loading from the internal Flash, the MODE value should be adjusted to MSPI MODE, and then the MSPI loading can be triggered by power cycling or triggering RECONFIG_N.

The connection diagram for programming an internal Flash via SSPI is shown in Figure 5-23.

Figure 5-23 Connection Diagram of Programming Internal Flash via SSPI

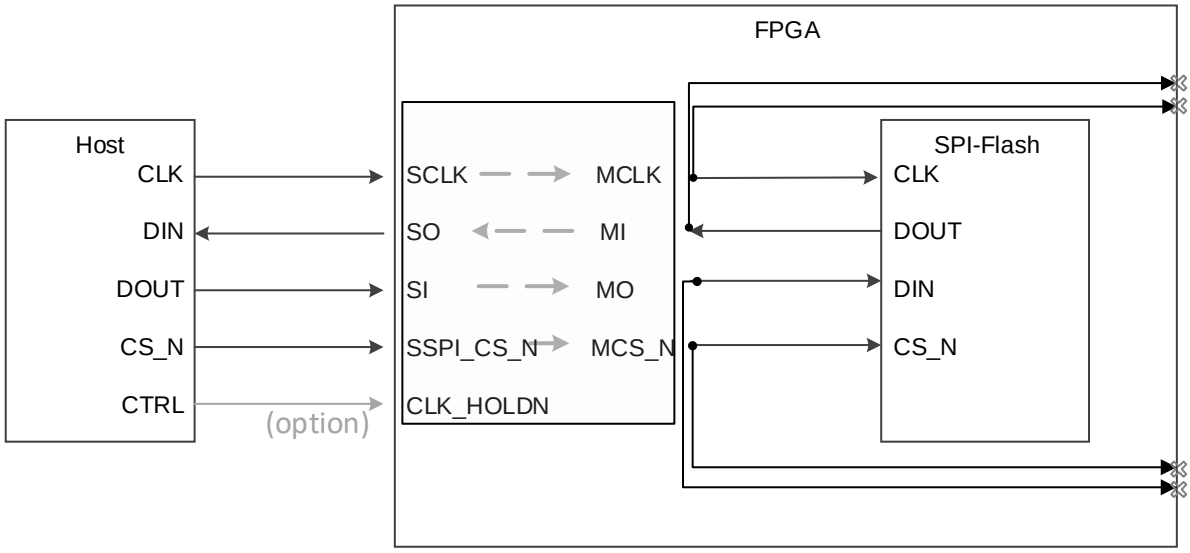


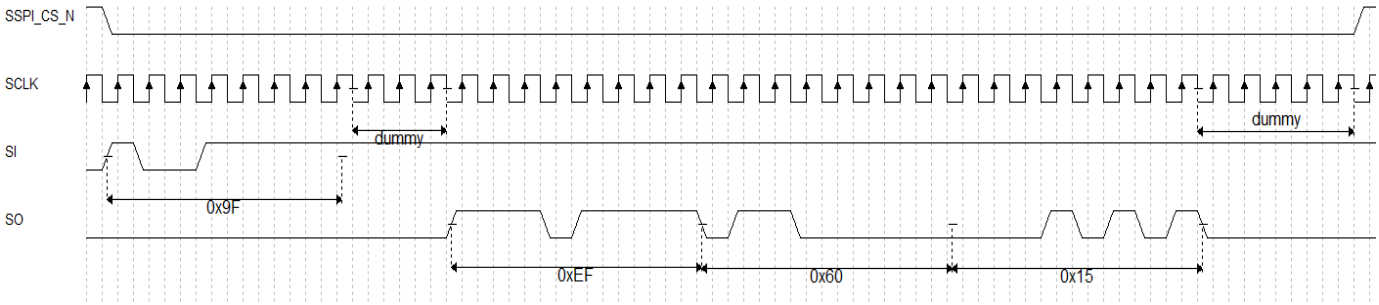
Figure 5-29 shows the programming flow. First, sends the "Program SPI Flash" (0x1600) instruction to FPGA via SSPI. After this, the FPGA can forward SSPI to Flash, and the SSPI on the Host side can directly access Flash. Then, it can be programmed according to Flash timing.

Note that when reading data from the Flash, the data being read back is delayed by three bits. For example, when reading the Flash's ID Code via SSPI, three extra clocks need to be sent to get the last three bits.

Reading the Internal Flash ID Code via SSPI

The internal Flash model of GW2AN-9X/18X may differ across product lots, as can the associated Flash ID code. However, they share the same timing of reading the Flash ID Code. The timing of reading the GW2AN-9X internal Flash ID code with the 0x9F instruction is shown below.

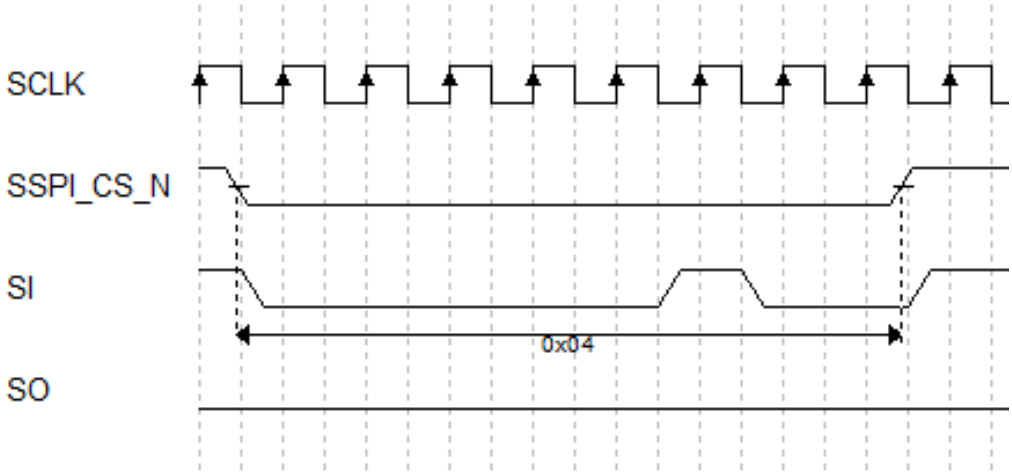
Figure 5-24 Timing of Reading the Internal Flash ID Code via SSPI



Sending instructions to the internal Flash via SSPI

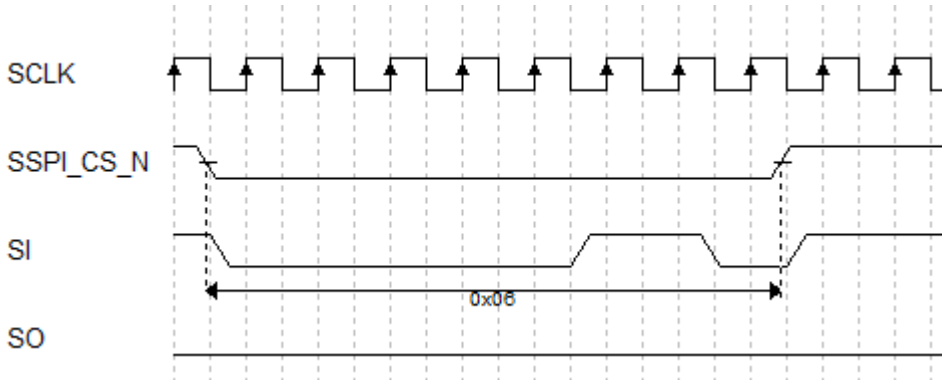
The timing of sending the Write Disable(0x04) instruction to the internal Flash is shown below.

Figure 5-25 Timing of Sending the Write Disable(0x04) Instruction via SSPI



The timing of sending the Write Enable(0x06) instruction to the internal Flash is shown below.

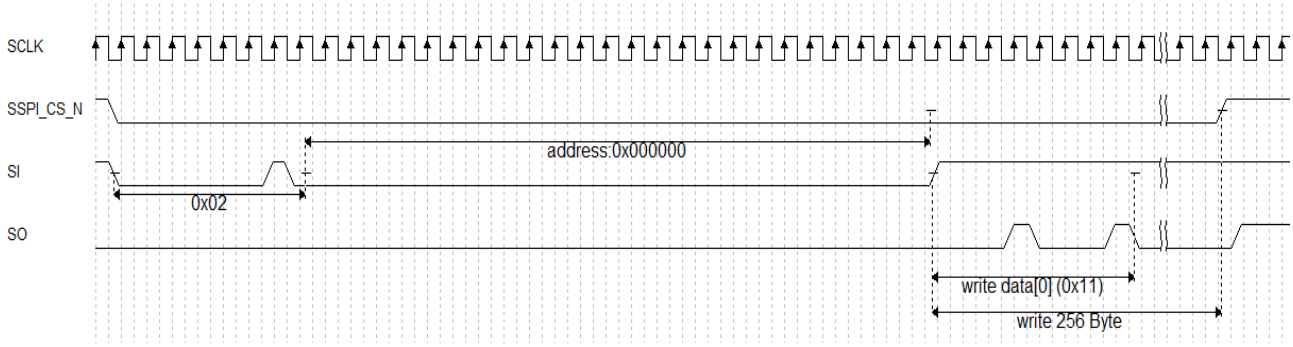
Figure 5-26 Timing of Sending the Write Enable(0x06) Instruction via SSPI



Programming the Internal Flash via SSPI

The timing of programming a page of the internal Flash is shown below.

Figure 5-27 Timing of Programming a Page via SSPI



Reading Data from the Internal Flash via SSPI

The timing of reading back a byte from the 0x00 address of the Flash is shown below:

Figure 5-28 Timing of Reading Back a Byte via SSPI

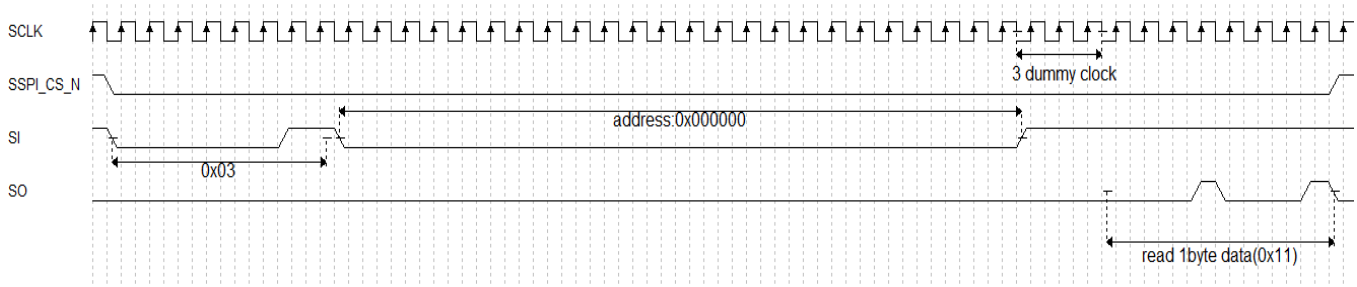
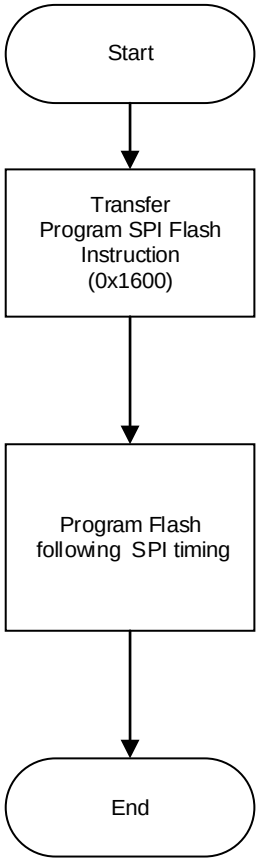


Figure 5-29 Flow Chart of Programming the Flash via SSPI



5.4.6 Multiple FPGA Connection View in SSPI Mode

Figure 5-30 Multiple FPGA Connection Diagram 1

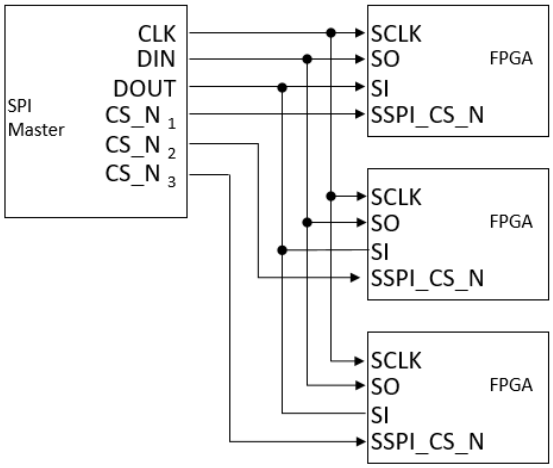
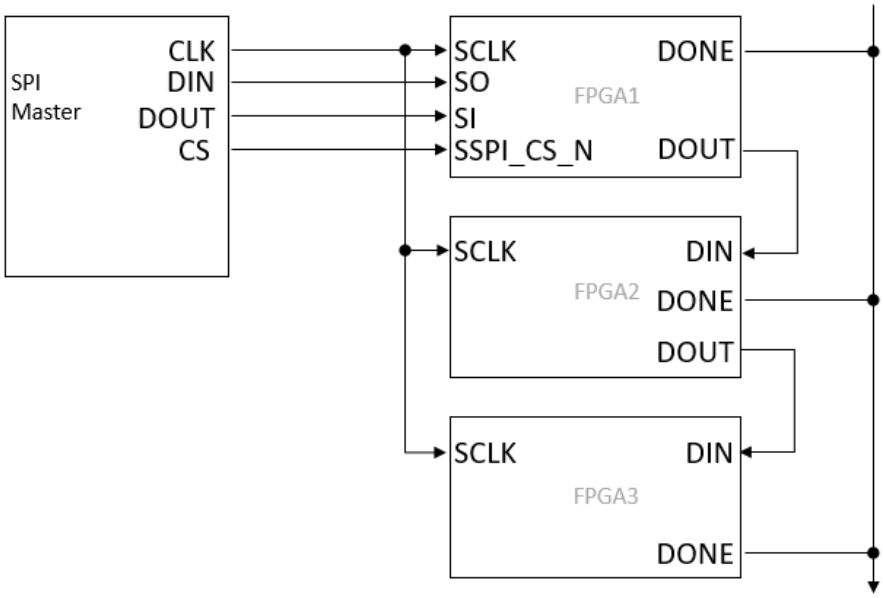


Figure 5-31 Multiple FPGA Connection Diagram 2



5.5 QSSPI Configuration Mode

By default, the Quad Enable Bit(QE) of the embedded Flash is enabled. You can directly use the Quad Slave SPI (QSSPI). The QSSPI configuration pins are shown in Table 5-12.

Table 5-12 QSSPI Mode Pins

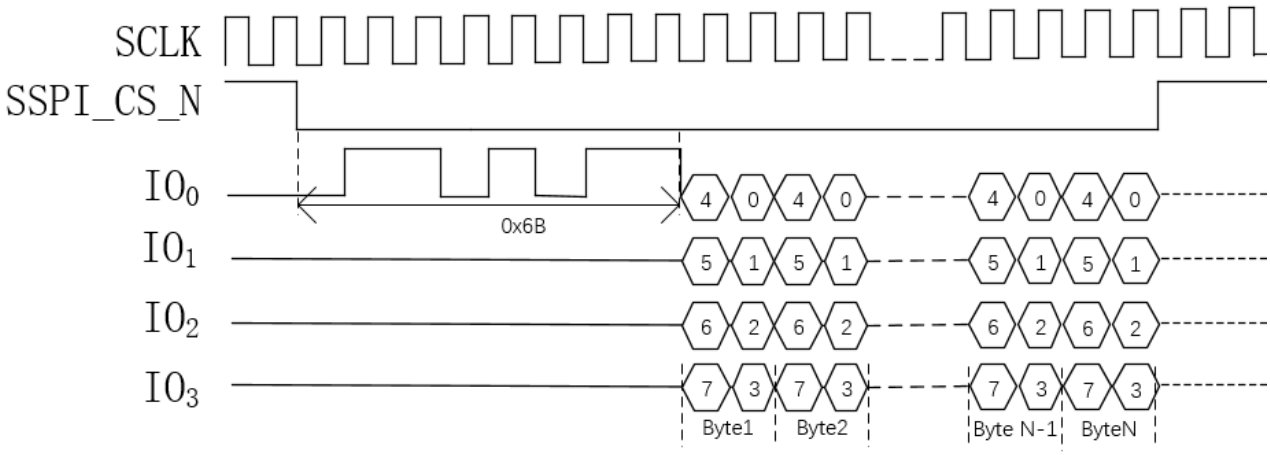
Pin Name	I/O	Description
RECONFIG_N	I, Internal weak pull-up	Low level pulse: Start GowinCONFIG
READY	I/O, Internal weak pull-up	High level: FPGA can be programmed and configured Low level: Programming configuration for FPGA is prohibited
DONE	I/O, Internal weak pull-up	High-level: Successfully programmed and configured; Low-level: Programming and configuration uncompleted or failed.
MODE[1:0]	I, Internal weak pull-down	Configuration mode selection, READY rising edge sampling
SCLK	I, Internal weak pull-up	Input clock
IO_3(CLKHOLD_N)	I, Internal weak	As IO_3 of Quad SPI

Pin Name	I/O	Description
	pull-down	
IO_2(QSSPI_WPN)	I, Internal weak pull-down	As IO_2 of Quad SPI
IO_1(SO)	O, Internal weak pull-down	As IO_1 of Quad SPI
IO_0(SI)	I, Internal weak pull-down	As IO_0 of Quad SPI
SSPI_CS_N	I, Internal weak pull-up	QSSPI Chip selection signal, active low.

The SRAM configuration process in QSSPI mode is as follows. The Read Status, Reinit, EraseSram, InitAddress, WriteEnable, and WriteDisble commands are the same with those of SSPI. Only Write Data uses QSSPI command.

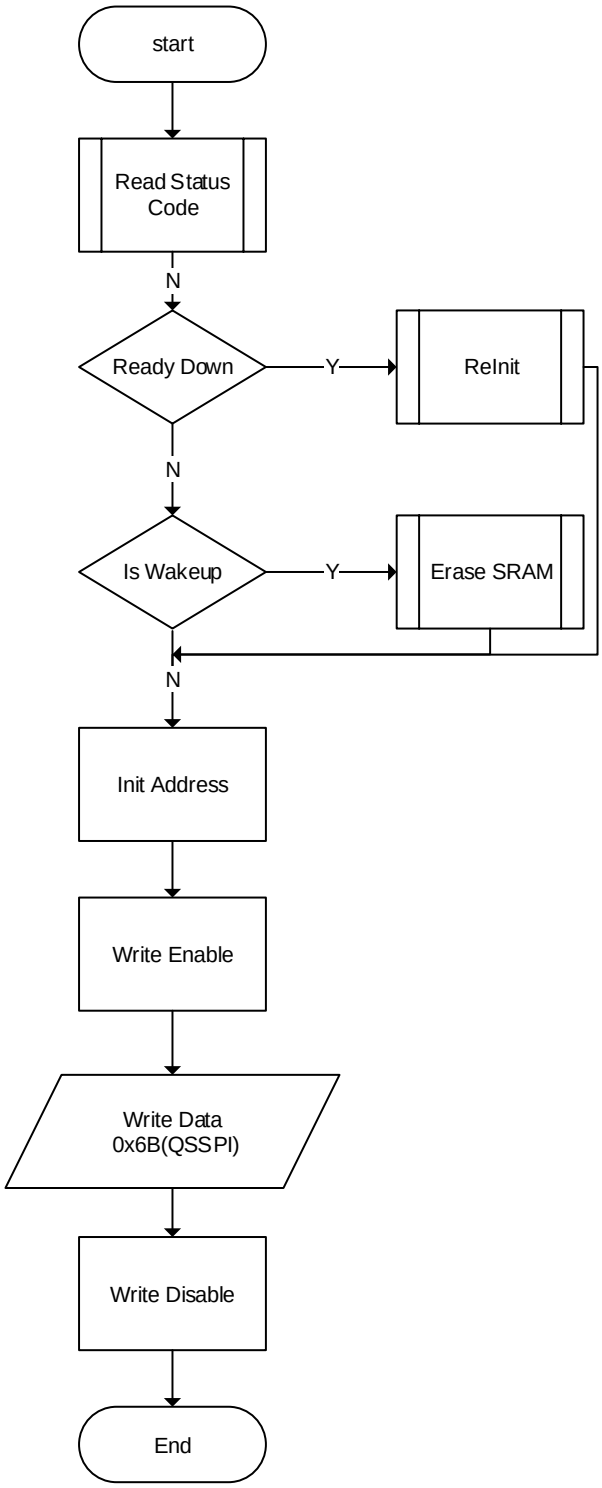
The timing diagram of QSSPI Write Data（0x6B）is shown in Figure 5-32.

Figure 5-32 QSSPI Write Data (0x6B) Timing



The flow of SRAM Configuration in QSSPI mode is shown in Figure 5-33.

Figure 5-33 Flow Chart of Configuring the SRAM via QSSPI



The process of programming the Flash using QSSPI mode is described below.

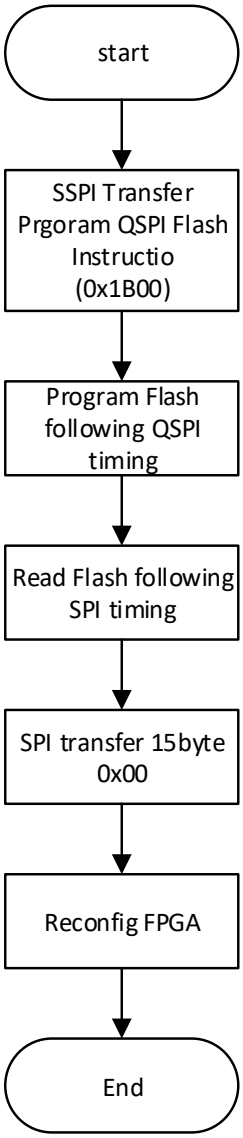
1. Send the 0x1B instruction (sspi_prgm_qspi) via Standard SPI. QSSPI_WPN pin and CLKHOLD_N pin should be held as 1 at this stage.
2. Start programming the Flash via QSPI (when programming the Flash, the write enable (0x04) instruction and the write disable(0x06) instruction are sent using Standard SPI, and QSPI is used when writing

data).

3. After programming, 15 0x00 instructions need to be sent via Standard SPI.
4. If you want to read back the Flash, you need to use Standard SPI. QSSPI does not support reading back the Flash using four lines.
5. Send the Reboot (0x3C) instruction via Standard SPI or trigger the RECONFIG_N pin, and then the FPGA will automatically load the bitstream file.

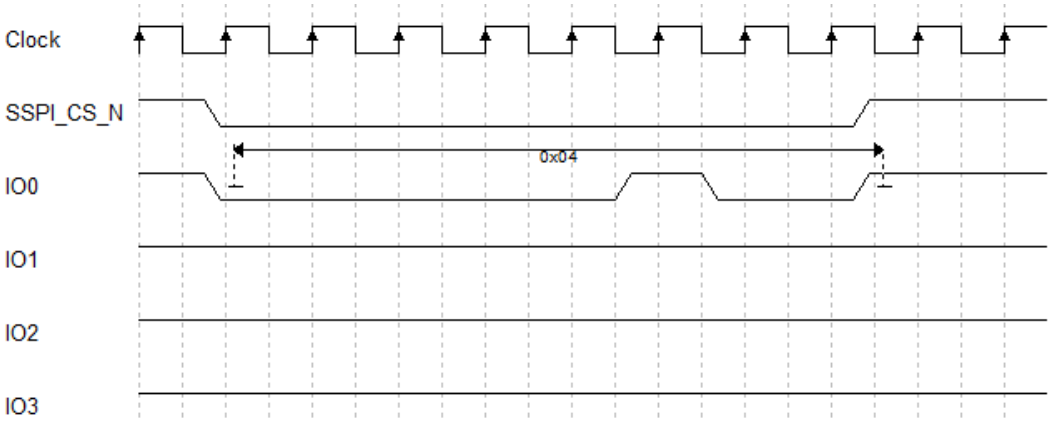
   The flow chart of programming the Flash using QSPI is as follows.

Figure 5-34 Flow Chart of Programming the Flash via QSSPI



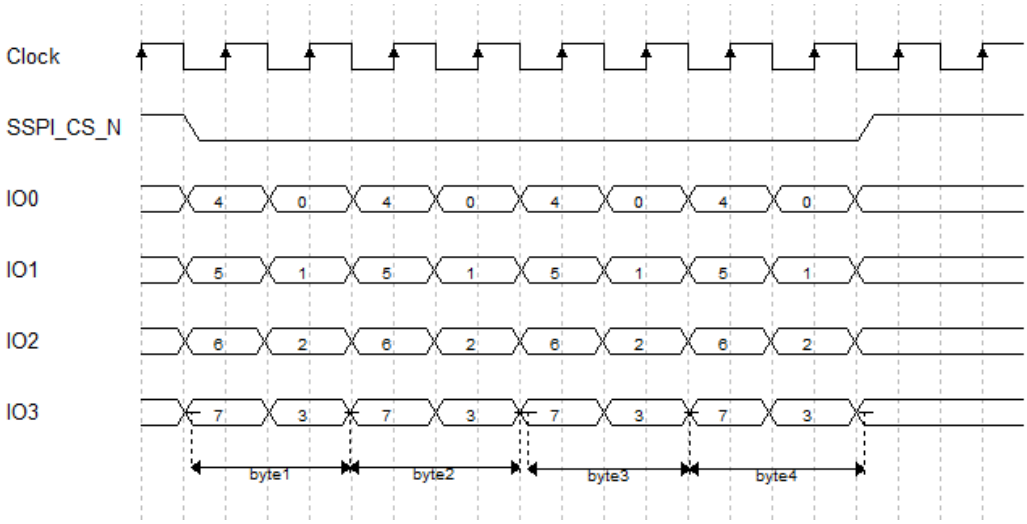
The QSSPI write instruction (e.g., 0x04) timing is as follows.

Figure 5-35 QSSPI Write Instruction Timing



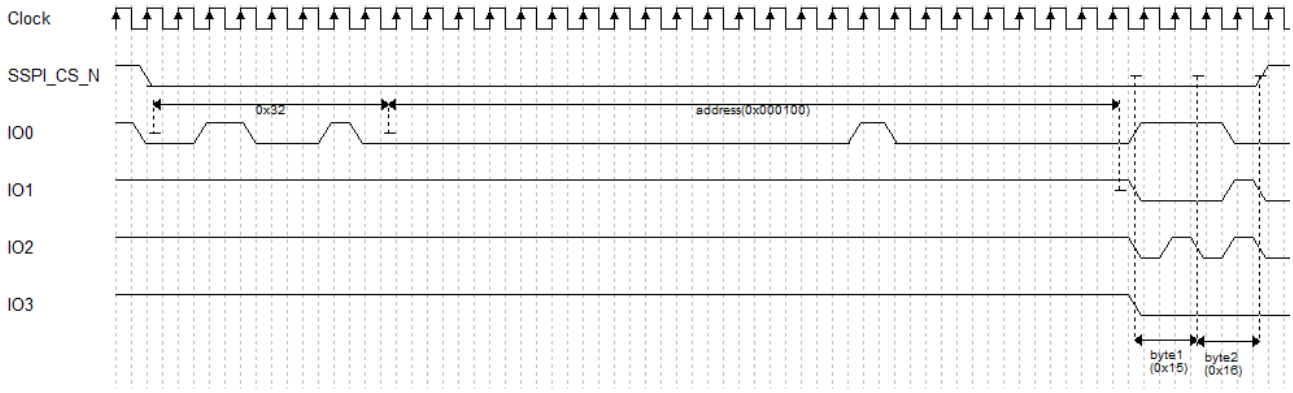
The QSSPI write data timing is as follows.

Figure 5-36 QSSPI Write Data Timing



The QSSPI page program timing is as follows.

Figure 5-37 QSSPI Page Program Timing



Note!

The timing diagram shows the sequence of writing 2-Byte data(0x15, 0x16) from address 0x000100.

5.6 CPU Configuration Mode

In CPU mode, the Host configures Gowin FPGA products through the 8-bit data bus interface. CPU mode pins are shown in Table 5-13.

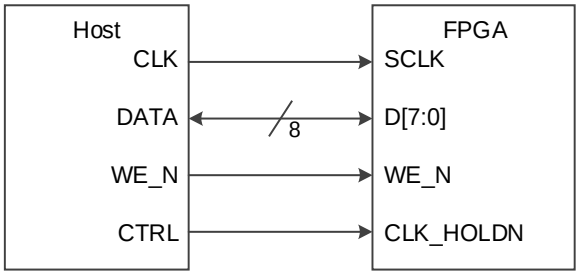
Table 5-13 CPU Mode Pins

Pin Name	I/O	Description
RECONFIG_N	I, Internal weak pull-up	Low level pulse: Start GowinCONFIG
READY	I/O, Internal weak pull-up	High-level pulse: The device can be programmed and configured; Low level: Programming configuration for device is prohibited
DONE	I/O, Internal weak pull-up	High-level: Successfully programmed and configured; Low-level: Programming and configuration uncompleted or failed.
MODE[1:0]	I, Internal weak pull-down	Configuration mode selection, READY rising edge sampling
SCLK	I, Internal weak pull-up	Input clock
CLKHOLD_N	I, Internal weak pull-down	Chip select signal in CPU mode, active low. This signal MUST be low to configure the FPGA in CPU mode.
WE_N	I, Internal weak	Read-write enable 0：Write

Pin Name	I/O	Description
	pull-down	1：Read
D[7:0]	I/O	Data I/O port: Used as input pin in CPU mode, and used as output pin after configuration for verification

The connection diagram for the CPU mode is shown in Figure 5-38.

Figure 5-38 Connection Diagram for CPU Mode



Note!

The figure above shows the minimum system diagram of the CPU MODE. The MODE value is set to "11". The connections are shown in for the other fixed pins are shown in Figure 5-1.

Other than the power requirements, the following conditions need to be met to use the CPU configuration mode:

- CPU port enable
  RECONFIG_N is not set as a GPIO during the first configuration after power up or the previous programming.
- Initiate new configuration
  Power-on again or trigger RECONFIG_N at one low pulse.

5.6.1 Configuration Timing

Before configuration, make sure that MODE[1: 0]=11, and DONE will be pulled up after configuration. If DONE or READY is pulled down, the configuration fails.

In the configuration process, data bus D[7:0] is represented in MSB way, and the FPGA reads the data at the SCLK falling edge.

Figure 5-39 CPU Mode Configuration diagram

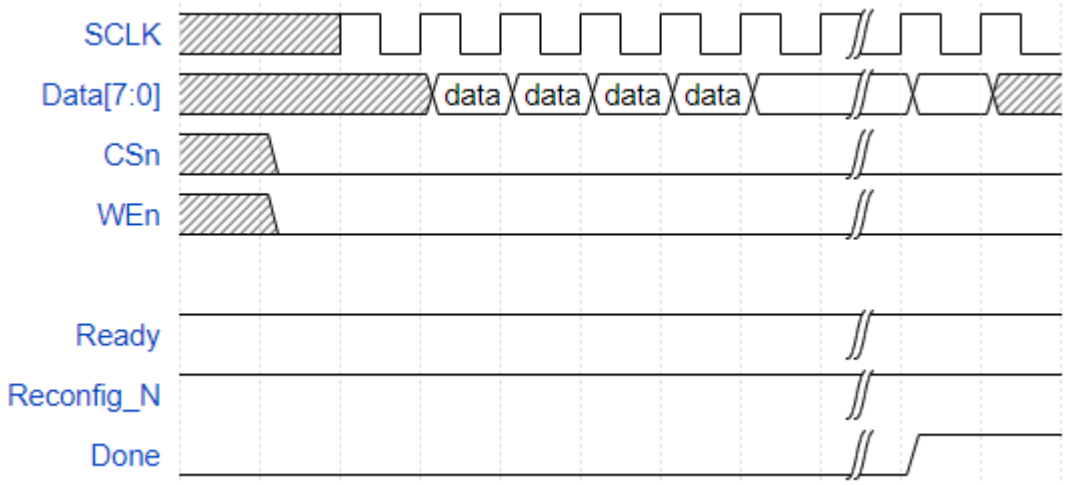


Figure 5-40 CPU Mode Configuration Timing

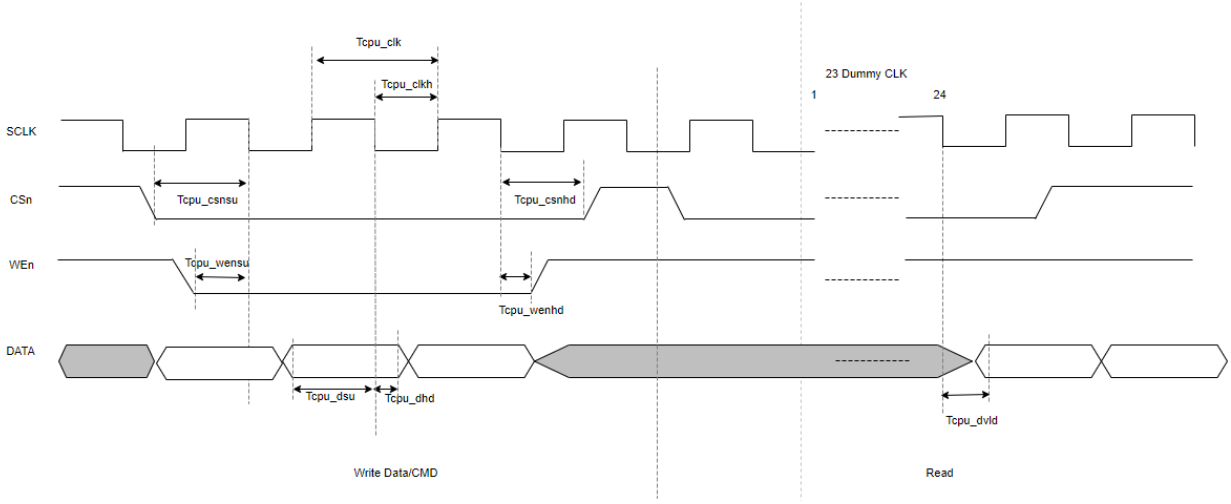


Table 5-14 CPU Configuration Timing Parameters

Name	Description	Min.	Max.	Units
Tcpu_clk	CPU input clock period	40	–	ns
Tcpu_csnsu	CLKHOLD_N(CSn) setup time to SCLK falling	8	–	ns
Tcpu_csnhd	CLKHOLD_N(CSn) hold time from SCLK falling	0	–	ns
Tcpu_wensu	WE_N setup time to SCLK falling	8	–	ns
Tcpu_wenhd	WE_N hold time from SCLK falling	0	–	ns
Tcpu_dsu	Write data input setup time to SCLK falling	10	–	ns
Tcpu_dhd	Write data input hold time from SCLK falling	0	–	ns
Tcpu_dvld	SCLK falling to read data output valid	–	10	ns
Tcpu_clkh	CPU input clock high duration	(clock cycle) *45%	(clock cycle) *55%	–

5.7 SERIAL

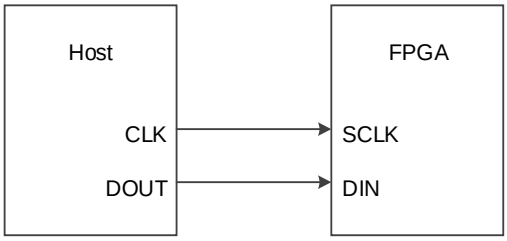
In SERIAL mode, Host configures Gowin FPGA products via serial interface. SERIAL is one of the configuration modes that use the least number of pins. The SERIAL mode can only write bitstream data to FPGA and cannot readback data from FPGA devices; as such, the SERIAL mode cannot read information on the ID CODE and USER CODE and status register. A definition of the pins employed in the SERIAL mode is provided in Table 5-15.

Table 5-15 Pin Definition in SERIAL Configuration Mode

Pin Name	I/O	Description
RECONFIG_N	I, internal weak pull-up	Low level pulse: Start GowinCONFIG
READY	I/O, internal weak pull-up	High-level pulse: The device can be programmed and configured; Low level: Programming configuration for device is prohibited
DONE	I/O, internal weak pull-up	High-level: Successfully programmed and configured; Low-level: Programming and configuration uncompleted or failed.
MODE[1:0]	I, internal weak pull-down	Configuration mode selection, READY rising edge sampling
SCLK	I, internal weak pull-up	Input clock
DIN	I, internal weak pull-down	Input data
DOUT	O, internal weak pull-down	Output data, only used in SERIAL configuration mode when FPGA cascading.

The connection diagram for the SERIAL mode is shown in Figure 5-41.

Figure 5-41 Connection Diagram for SERIAL Mode



Note!

The figure above shows the minimum system diagram of the SERIAL MODE. The MODE value is set to "10". The connection for the other fixed pins are shown in Figure 5-1.

SERIAL Configuration Timing

See Figure 5-42 for the timing of SERIAL mode.

Figure 5-42 SERIAL Configuration Timing

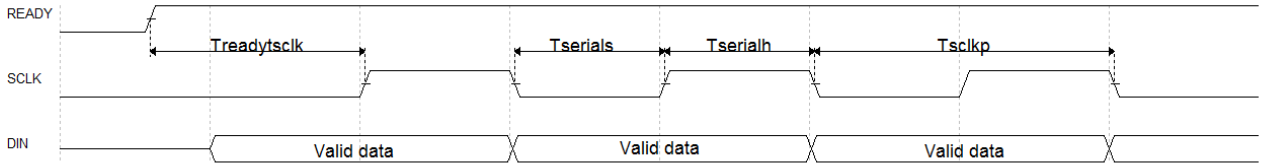


Table 5-16 shows the timing parameters.

Table 5-16 SERIAL Configuration Timing Parameters

Name	Description	Min.	Max.
T_{sclkp}	SCLK clock period	15ns	-
$T_{serials}$	SERIAL PORT setup time	2ns	-
$T_{serialh}$	SERIAL PORT hold time	0ns	-
$T_{readytsclk}$	Time from READY rising edge to first SCLK edge	TBD	-

Other than the power requirements, the following conditions need to be met to use the SERIAL configuration mode:

- SERIAL port enable
  RECONFIG_N is not set as a GPIO during the first configuration after power up or the previous programming.
- Initiate new configuration
  Power-on again or trigger RECONFIG_N at one low pulse.

5.8 I²C Configuration Mode

In I²C Mode, Gowin FPGA products are configured by Host via I2C interface. I²C is one of the configuration modes that use the least number of pins. The I²C mode can only write bitstream data to FPGA and cannot readback data from FPGA devices; as such, the I²C mode cannot read information on the ID CODE, USER CODE, status register, and read-back check. The definition of the I²C mode pins is described in Table 5-17.

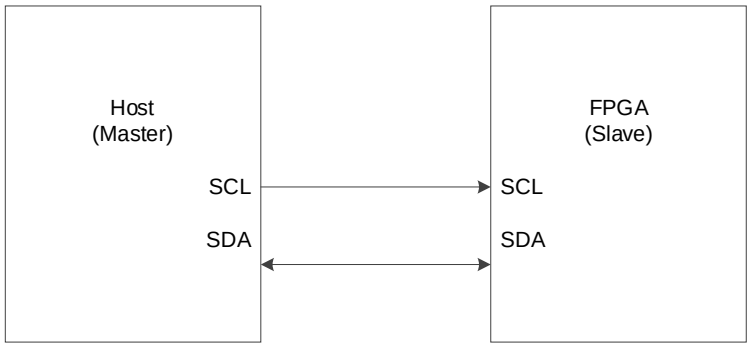
Table 5-17 Pin Definition in I²C Configuration Mode

Pin Name	I/O	Description
RECONFIG_N	I, internal weak pull-up	Low level pulse: Start GowinCONFIG
READY	I/O, internal weak pull-up	High-level pulse: The device can be programmed and configured; Low level: Programming configuration for device is prohibited
DONE	I/O, internal weak pull-up	High-level: Successfully programmed and configured; Low-level: Programming and configuration

Pin Name	I/O	Description
		uncompleted or failed.
MODE[1:0]	I, internal weak pull-down	Configuration mode selection, READY rising edge sampling
SCL	I, internal weak pull-down	Input clock
SDA	I/O, internal weak pull-down	Data output, or output ACK When GowinCONFIG supports I²C, external pull-up is required.

The connection diagram for the I²C mode is shown in Figure 5-43.

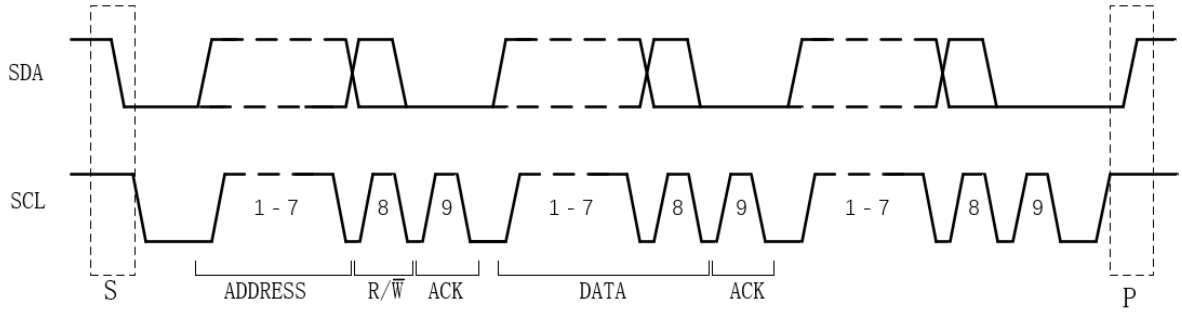
Figure 5-43 Connection Diagram for I²C Configuration Mode



Note!

- The figure above shows the minimum system diagram of the I²C MODE. Figure 5-1 shows the other fixed pin connections.
- GW2AN-18X/9X do not support standard I²C bus, they only support single device configuration based on I2C protocol.

Figure 5-44 I²C Configuration timing



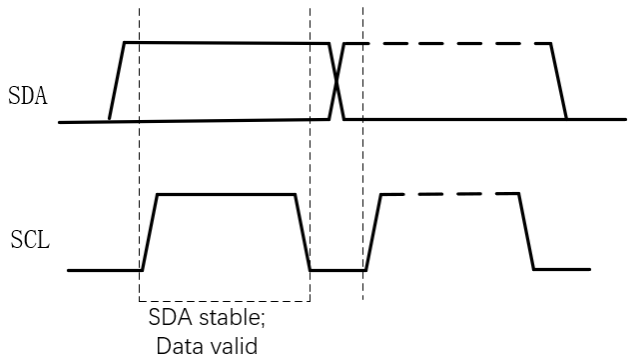
I²C is a serial transmission bus. Data is transmitted according to the protocol shown in the figure above. In normal state, SDA and SCL are both at high level.

Table 5-18 I²C Configuration Timing Parameters

Parameter	Description	
S	Startup Condition	A HIGH to LOW transition on the SDA line while SCL is HIGH.

Parameter	Description	
P	Stop Condition	SDA jumps from low to high while SCL is HIGH.
ADDRESS	Address frame	A unique 7-bit or 10-bit sequence for each slave device that identifies the slave device when the master device is about to communicate with it.
R/W	Read/ Write Bit	Determines whether the master sends data to the slave (0) or reads data from the slave (1).
ACK	ACK/NACK bit	Each frame in the message is followed by an ACK/NACK bit, and Gowin FPGA returns 0 if correct.
DATA	Data	A data has 8bits, and the most significant bit is sent first.

All DATA on the I^2C bus is transmitted in 8-bit bytes. Each byte sent by the transmitter, it releases the DATA line during the clock pulse 9, and the receiver sends back a response signal. The response signal is a valid response bit (ACK bit) if it is low, indicating that the receiver has successfully received the byte. The response signal is a non-acknowledgment bit (NACK) if it is high, which generally indicates that the receiver did not succeed in receiving the byte. The requirement for the ACK feedback is that the receiver pulls the SDA line low during the low level prior to the 9th clock pulse and ensures a stable low level during the high level of the clock. If the receiver is the master, after it receives the last byte, it sends a NACK signal to notify the controlled sender to end the data transmission and releases the SDA line for the master receiver to send a stop signal. Each bit of data transmitted on the I2C bus has a corresponding clock pulse (or synchronous control), that is, each bit of data is transmitted serially on the SDA bit by bit based on the SCL serial clock. During data transfer, the level on the SDA must remain stable, with the low level being data 0 and the high level being data 1, while the SCL is high. The level on the SDA is allowed to change state only while the SCL is low. Logic 0 has a low voltage level and Logic 1 has a high voltage level.As shown in the figure below.



The list of I2C mode supported by Gowin FPGA devices is as shown in Table 5-19.

Table 5-19 I²C Configuration Mode Frequency and Address

Mode	Device	Frequency	Address
SRAM	GW2AN-18X/9X	100KHz~555K	7'b1010000[1]
Embedded Flash[2]			

Note!

- [1] I²C slave address supports 2bit configuration. The default address is 7'b1010000. The underlined bits can be changed to 1, but cannot be changed back to 0.
- [2] When I2C operates the Flash, the data stream file needs to be converted into a specific data stream. The conversion tool is included in the Programmer. The file name suffix after conversion is ".i2c ". “.I2c” is a binary file.

Other than the power requirements, the following conditions need to be met to use the I²C configuration mode:

- I²C port enable
  RECONFIG_N is not set as a GPIO during the first configuration after power up or the previous programming.
- Initiate new configuration
  Power-on again or trigger RECONFIG_N at one low pulse.

5.8.1 Configuration Instruction

The I²C configuration mode uses a uniform address and uses different instructions to specify the configuration of SRAM or Flash. The following is a list of I²C instructions:

Table 5-20 I²C Configuration Instruction

Name	Complete Instruction (Instruction Byte + Redundant Information Byte)
Reinit	0x3f
Config-SRAM	0x33
Config-Flash	0x55
Reboot/Reconfig	0x3c

Figure 5-45 Reinit Timing Diagram

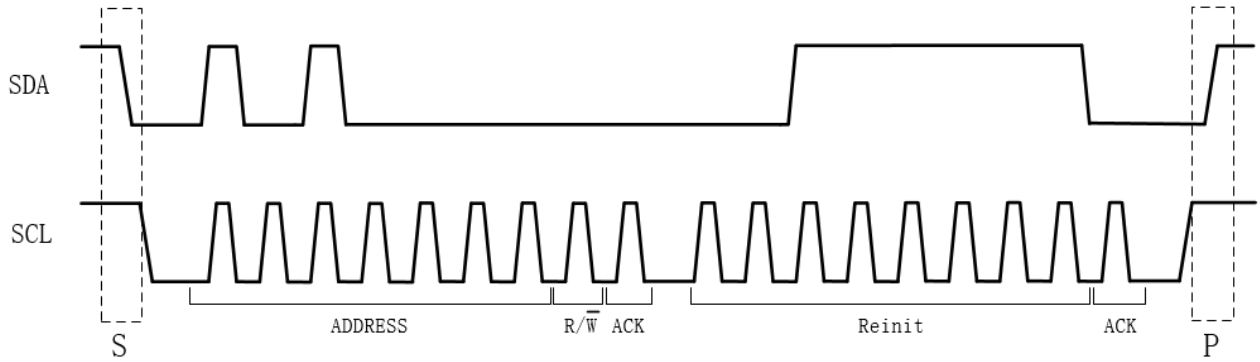


Figure 5-46 SRAM Configuration Timing Diagram

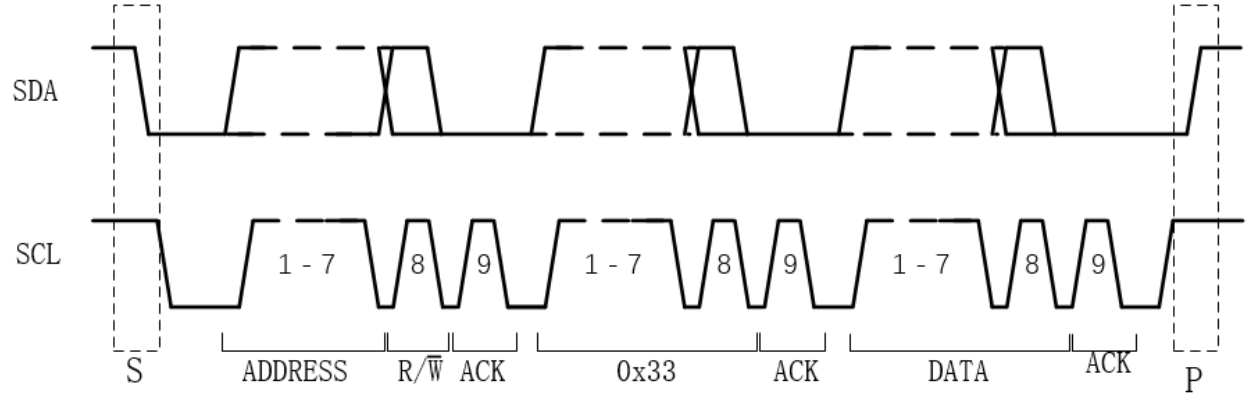


Figure 5-47 Flash Configuration Timing Diagram

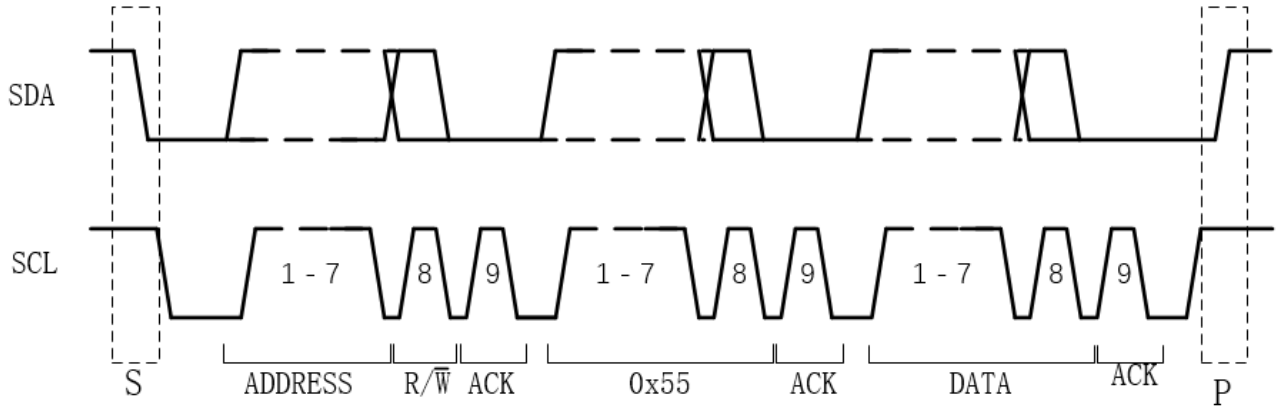
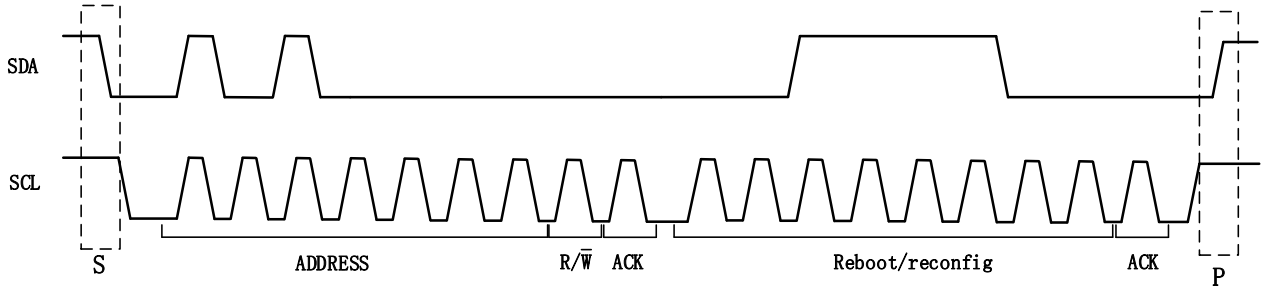


Figure 5-48 Reboot Timing Diagram



5.8.2 The Process of Configuring SRAM via I²C

Note!

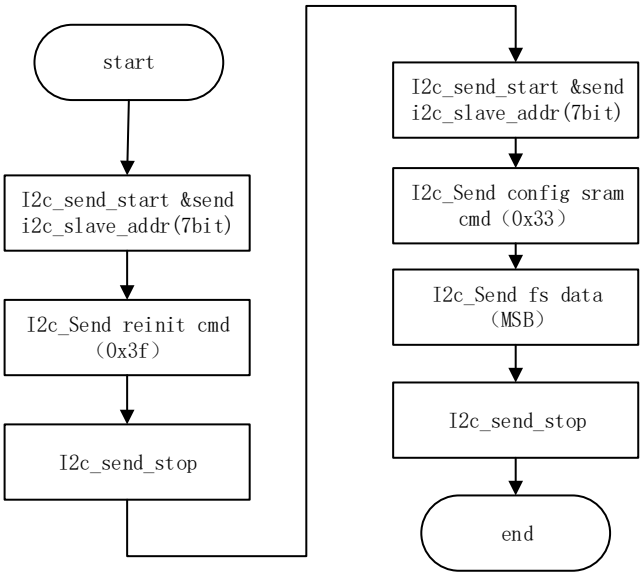
When re-configuring the SRAM, you need to erase the previously configured SRAM first. You can erase the SRAM by power-cycling the FPGA or triggering the RECONFIG_N pin with a low-level pulse.

The process of configuring SRAM via I²C is as follows:

1. Send the start signal and 7-bit address + R/W bit(write);
2. Send data 0x3f (reinit command);
3. Send the stop signal;
4. Send the start signal and 7-bit address +R/W bit(write);
5. Send data 0x33(config_sram);

6. Send data stream in MSB mode;
7. Send the stop signal after the data stream file is sent.

Figure 5-49 Flow Chart of Configuring SRAM via I²C



5.8.3 The Process of Configuring (Programming) Flash via I²C

The data form of configuring Flash via I²C is as follows:

CS	SI	-------	SPI value
bit7	bit6	-------	SPI vector 1
bit5	bit4	-------	SPI vector 2
bit3	bit2	-------	SPI vector 3
bit1	bit0	-------	SPI vector 4

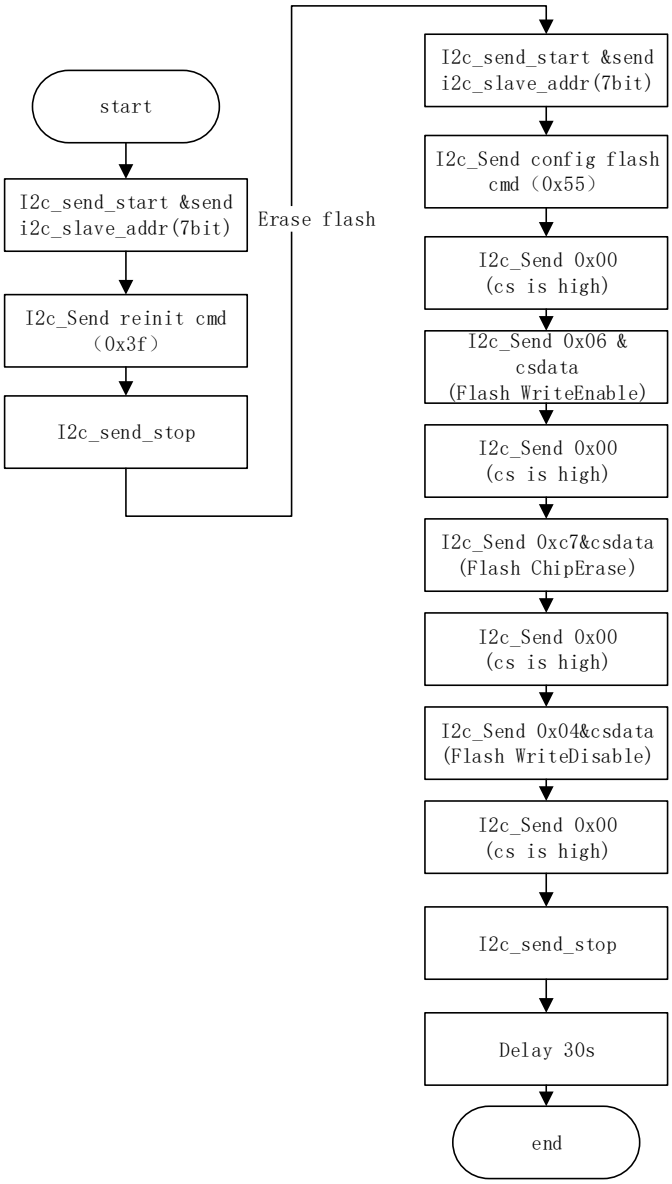
Note!

The value of CS obtained by I²C parsing is opposite to the value of CS required by Flash.

Process of Erasing the Flash via I²C：

1. Send the start signal and 7-bit address (write operation);
2. Send data 0x3f (reinit command);
3. Send the stop signal;
4. Send the start signal and 7-bit address (write operation);
5. Send data 0x55 (config_flash);
6. Send data 0x00 (pull up the CS signal);
7. Send data 0xaa 0xbe (DI:06&CS:ff) (Flash write enable);
8. Send data 0x00 (pull up the CS signal);
9. Send data 0xfa 0xbf(DI:c7&CS:ff) (Flash chip erase);
10. Send data 0x00 (pull up the CS signal);
11. Send data 0xaa 0xba(DI:04&CS:ff) (Flash write disable);
12. Send data 0x00 (pull up the CS signal);
13. Send the stop signal;
14. Delay 30s and wait for Flash erasure to complete.

Figure 5-50 Flow Chart of Erasing Flash via I²C：

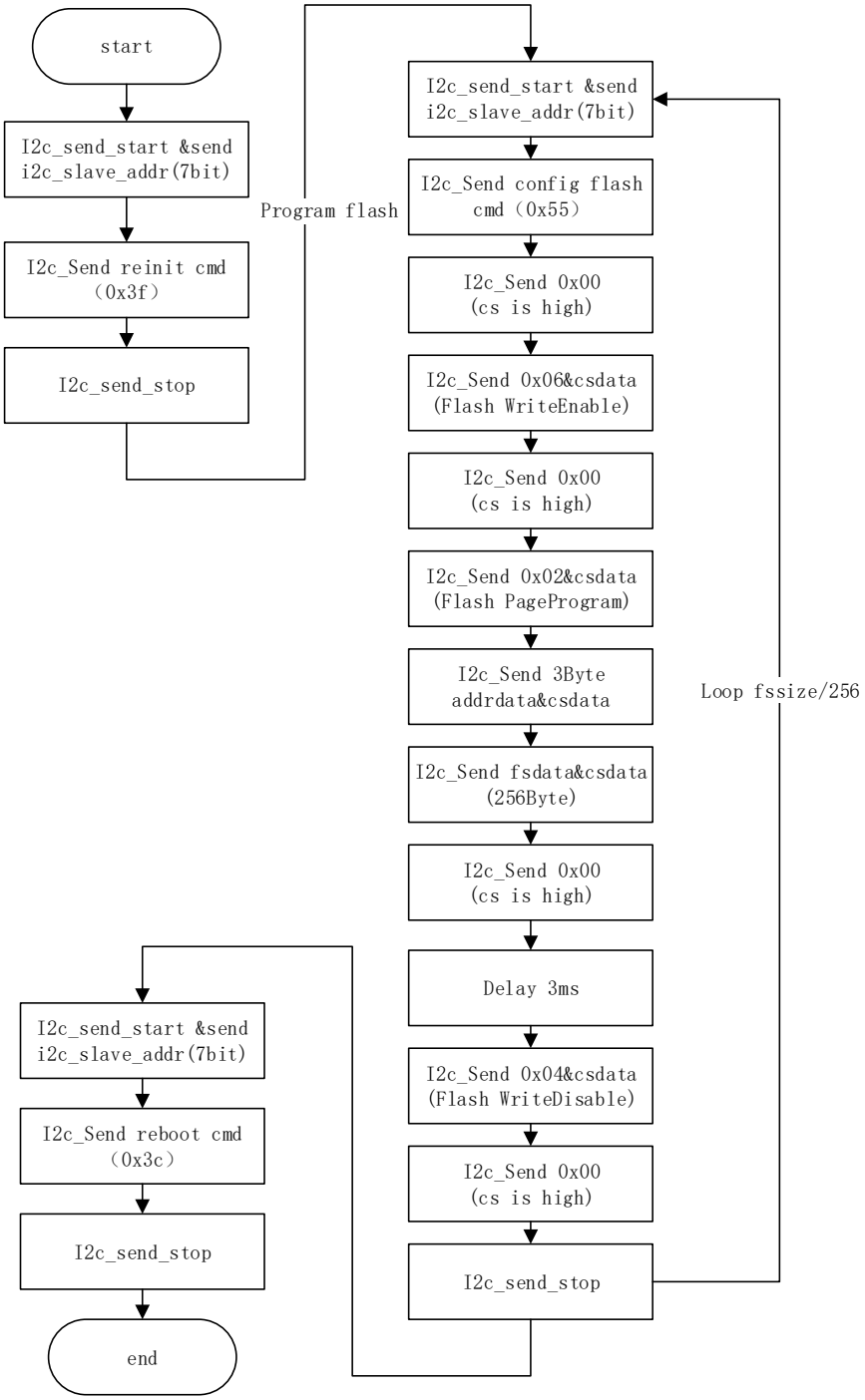


Process of Programing the Flash via I²C

1. Send the start signal and 7-bit address (write operation);
2. Send data 0x3f (reinit command);
3. Send the stop signal;
4. Send the start signal and 7-bit address (write operation);
5. Send data 0x55 (config_flash);
6. Send data 0x00 (pull up the CS signal);
7. Send data 0xaa 0xbe (DI:06&CS:ff) (Flash write enable);
8. Send data 0x00 (pull up the CS signal);
9. Send data 0xaa 0xae (DI:02&CS:ff) (pageprogram command);
10. Send 24-bit address (DI:xx&CS:ff);
11. Send data stream (DI:xx&CS:ff);
12. Send data 0x00 (pull up the CS signal);
13. Delay 3ms;

14. Send data 0xaa 0xba(DI:04&CS:ff) (Flash write disable);
15. Send data 0x00 (pull up the CS signal);
16. Send the stop signal;
17. Repeat steps 5-15 until the data stream file writing is completed;
18. Send the start signal and 7-bit address (write operation);
19. Send data 0x3c (reboot command);
20. Send the stop signal;

Figure 5-51 Flow Chart of Configuring the Flash via I²C



5.9 Example Files

For example files, please contact our technical support or local office. For the configuration modes and test platforms supported by the example files, please refer to Appendix B Example File Information.

6 Bitstream File Configuration

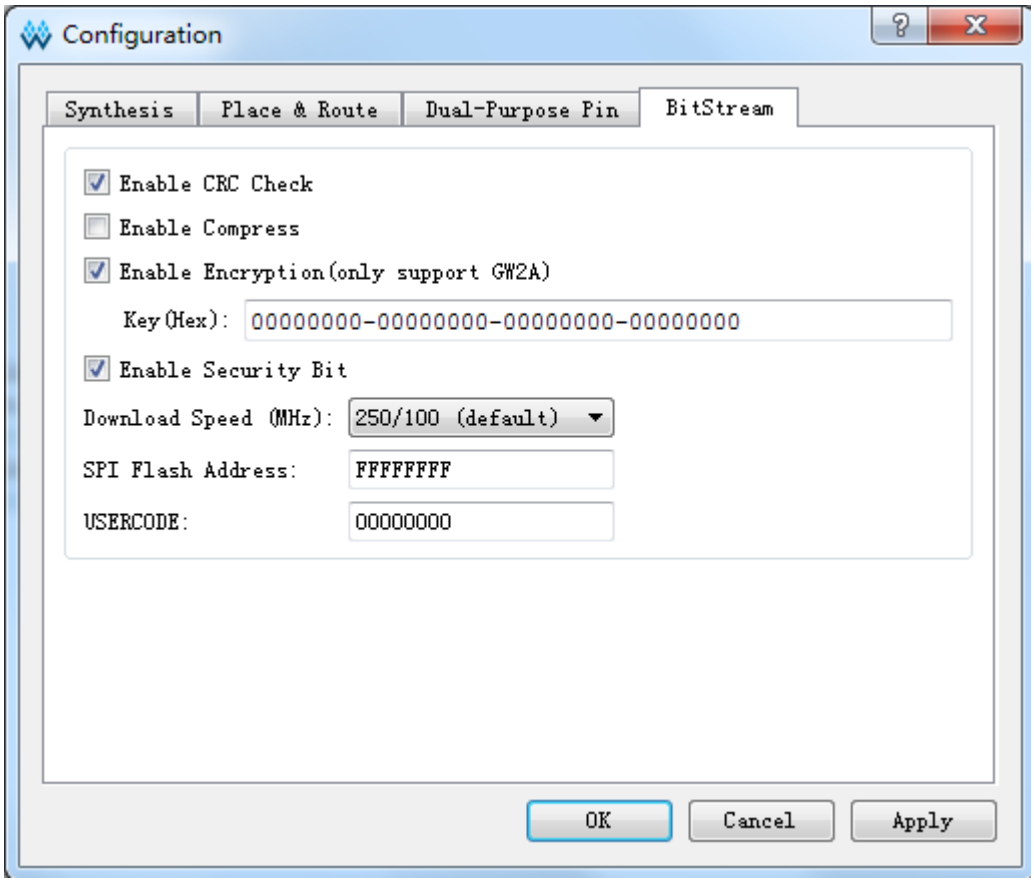
The features of Gowin FPGA products need to be configured and programmed using Gowin software. The settings mainly include configuration pins multiplexing options and bitstream data configuration options. This chapter describes the bitstream file configuration. For the details about the configuration pin reuse, please refer to 4.1.2 Configuration Pin Reuse.

To transfer the configuration data safely and accurately, the CRC calibration algorithm has been incorporated by default in the FPGA bitstream file, and the security bit is set. During the process of data configuration, input data is checked in real time. The wrong data cannot wake the device, and the DONE signal is pulled down. After the configuration of the bitstream with security bit is complete, data readback cannot be performed.

6.1 Configuration Options

Please refer to Figure 6-1 for the related configuration data setting interface. The options include CRC enable, bit stream data compression, encryption key settings, security bit settings, MSPI configuration frequency selection, SPI Flash starting address settings in multiple configuration modes, USER CODE setting, etc. The lower 12 bits of an SPI Flash startup address is invalid and the address space of ADDR [23:12] can be set by users.

Figure 6-1 Configuration Options



Note!

The security bit setting is forcibly checked after Gowin software verifies the encryption key setting option. In addition to ensuring the data is secure during the transmission process, using these bitstream settings during configuration also prevents any readback, thereby ensuring maximum protection of user data.

6.2 Configure Data Encryption

The Gowin GW2AN-18X/9X devices support bitstream data encryption, using the 128 AES encryption algorithm. Please refer to the following steps for the data encryption configuration:

1. Enter the encryption KEY (KEY) in Gowin software interface to generate the bitstream data;
2. Enter the decrypt key in Gowin Programmer;
3. After encrypted bitstream data is loaded into the device, FPGA compares the data that has been loaded with the decrypt key values stored in advance.
   If data parsing succeeds, the device finishes configuration and begins to work; if data parsing fails, the device cannot work, and READY and DONE are pulled down.

6.2.1 Definition

- AES encryption key: AES private key used in AES encryption algorithm, specified by users. Referred to as "key” in this manual.
- AES encryption key length: 128 bits;

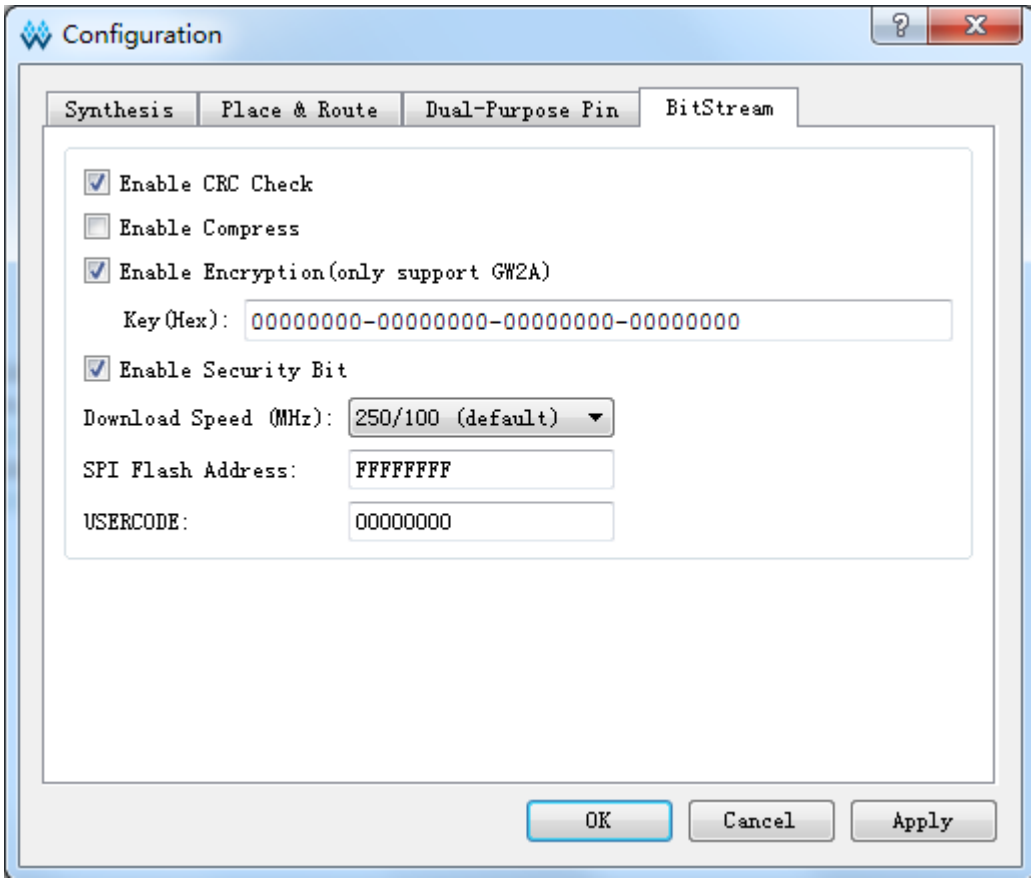
- Key: An abbreviation for AES encryption key. GW2A(R) series of FPGA products offers an address with 128 bits length to store Key;
- Lock: To ensure the security of AES Key, it is used to control the read permissions for the Key. This operation is named as "lock" in this manual. When it's locked, all the read back data is 1.

6.2.2 Enter Encryption KEY

Refer to the steps below to write the encryption keys in Gowin software:

1. Open the corresponding project in Gowin software;
2. Select "Project > Configuration > Dual Purpose Pin" from the available menu options;
3. Click "BitStream", check "Enable Encryption (only support GW2A)" and input the key value, as shown in Figure 6-2.

Figure 6-2 Encryption Key Setting Method



After setting the encryption key successfully, write the decrypted key to the FPGA key storage area for the device to analyze the encrypted bitstream data to complete the configuration.

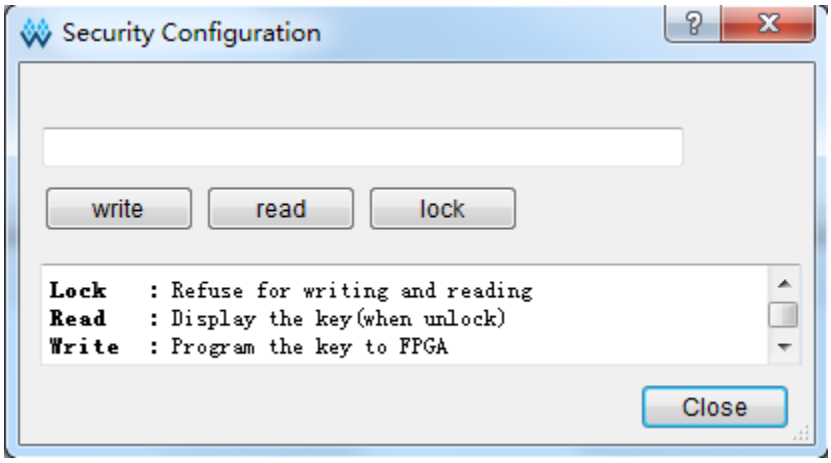
6.2.3 Enter the Decrypt Key

To input the decryption key, refer to the following steps:

1. Open the Gowin programming software;
2. Scan the FPGA device;

3. Right-click on the device name and select "Configure Security";
4. Enter the encrypted key value in the pop-up interface, click "write" and write the value to the FPGA, as shown in Figure 6-3.

Figure 6-3 Setting the Decryption Key



After the decryption key is written successfully, readback the written value via the "Read" button on the interface to verify.

After the key is written successfully, users also can select to "lock" it in FPGA via the Lock command. Once you have performed this action, any read and write key operations will be invalid, the key value cannot be modified, and all read bits are all "1".

After the decryption key is set, the encrypted bitstream data will only work when the data matches the decryption key. The key does not affect the non-encrypted bitstream data.

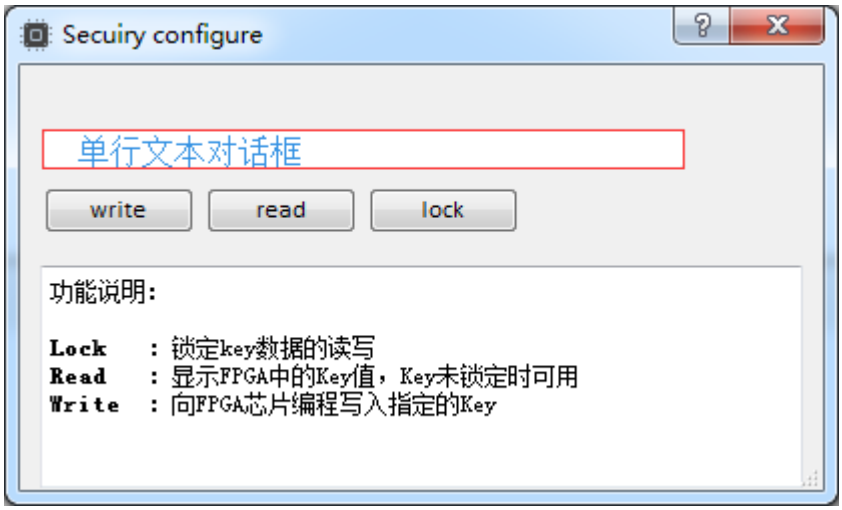
Note!

The initial value of the Gowin FPGA keys is 0. If a key value is changed to 1, it cannot be changed back to 0. For example, the key value written during an operation is 00000000-00000000-00000000-00000001, and the last bit of the modified key must be 1.

6.2.4 Programming Operation

Gowin Programmer offers the tool for programming AES encryption key. Open this tool by clicking "Edit > Security Key Setting " in Gowin Programmer, as shown in Figure 6-4.

Figure 6-4 AES Security Configure



This configuration contains the following three parts:

- Write: Write Key;
- Read: Read Key;
- Lock: Lock read and write access to the Key.

Write

1. Write the user-defined Key to the text box in the figure above;
2. Click "Write" button;
3. Return the validation result after running.

Read

Click "Read" button to validate the written AES encryption key again. The Key that is read from the tool will be displayed in the text box in the figure above;

Lock

Click "Lock" to lock the read and write permission of Key. If it is locked, the Key cannot be read or write.

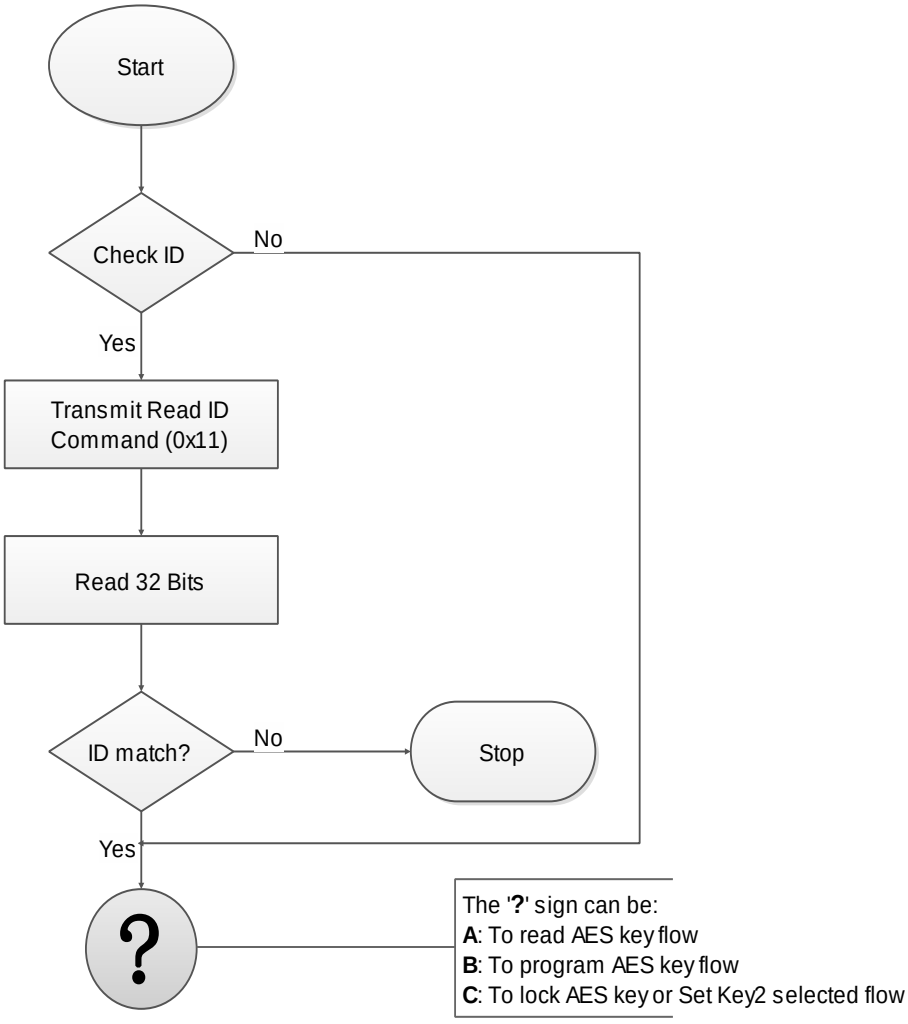
6.2.5 Programming Flow

Figure 6-5 ~ Figure 6-8 show the flow of how to program or Lock the AES key. All the flows are based on JTAG protocol.

Check IDCODE

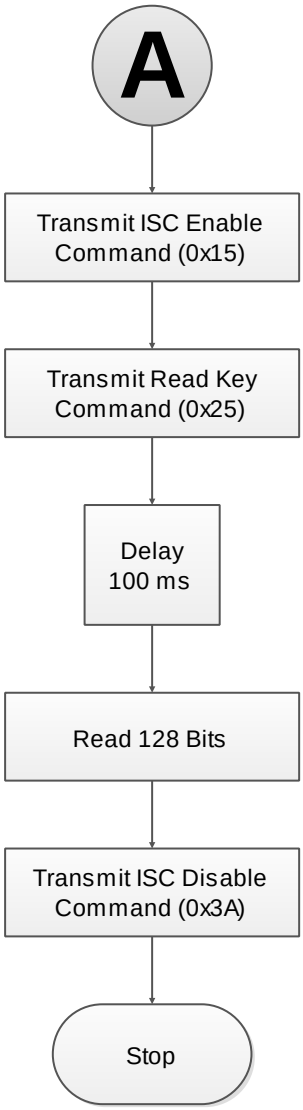
Check the device ID to determine whether the JTAG protocol works properly and whether the programing object is correct to avoid misoperation.

Figure 6-5 Prepare



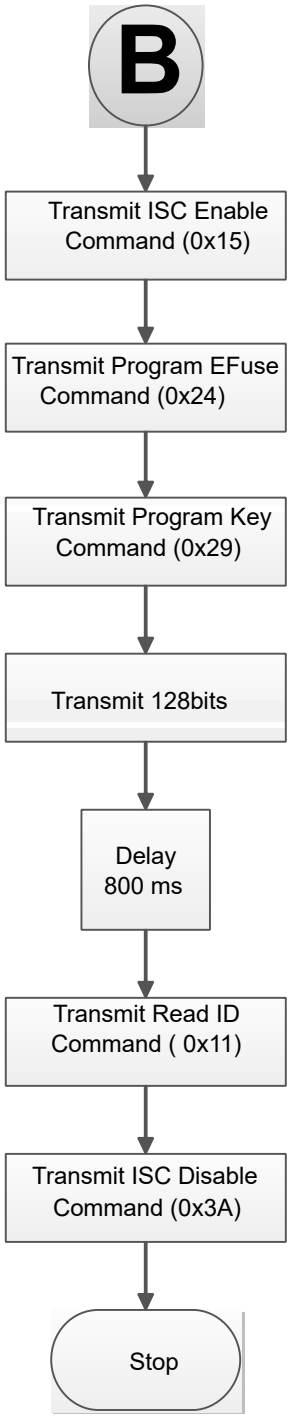
Read AES Key

Figure 6-6 Read AES Key Flow



Program AES Key

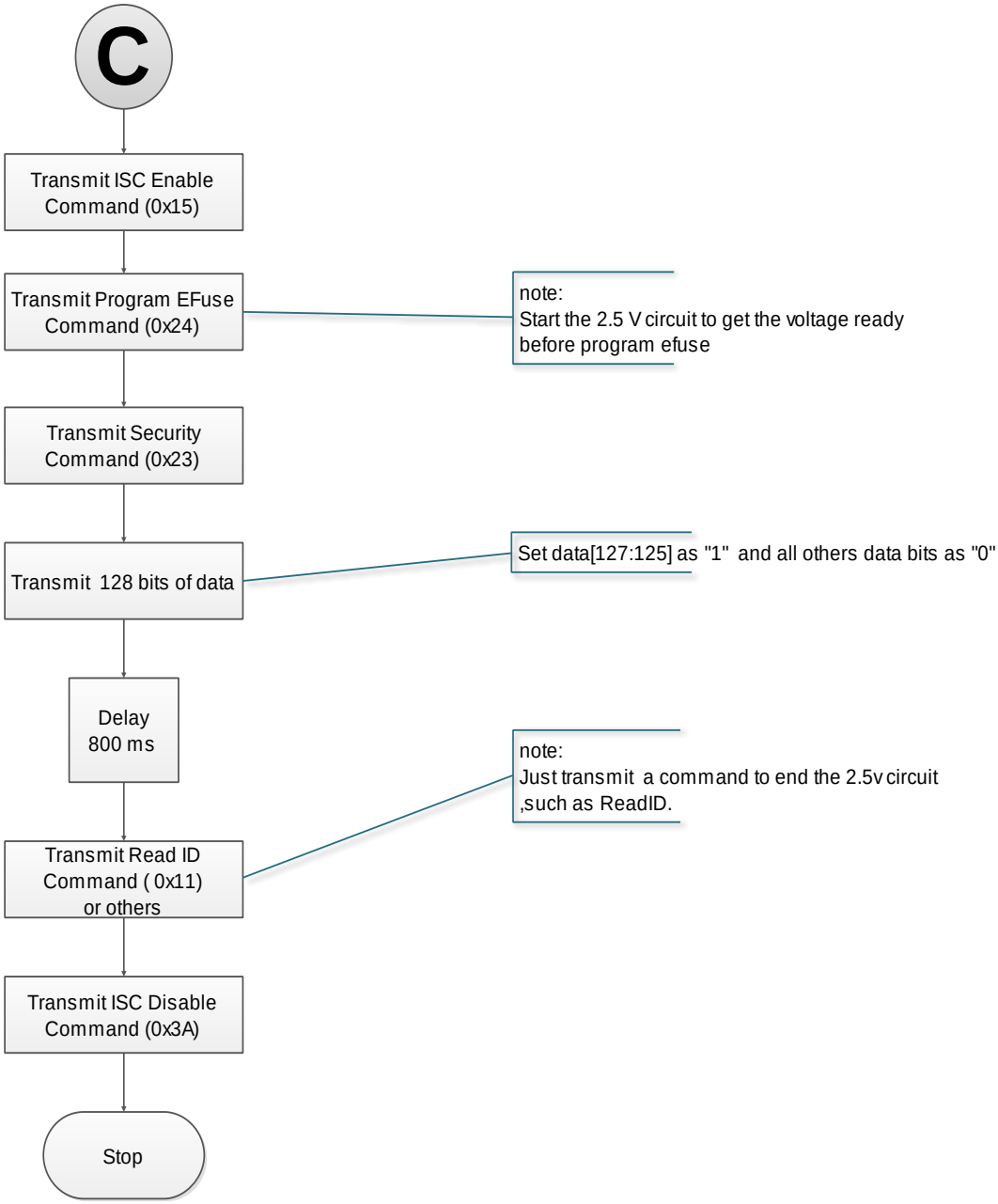
Figure 6-7 Program AES Key Flow



Lock AES Key

Locking the AES Key prevents the Key leakage. After locking the AES Key, you will not be able to read and configure the AES Key.

Figure 6-8 Lock AES Key Flow



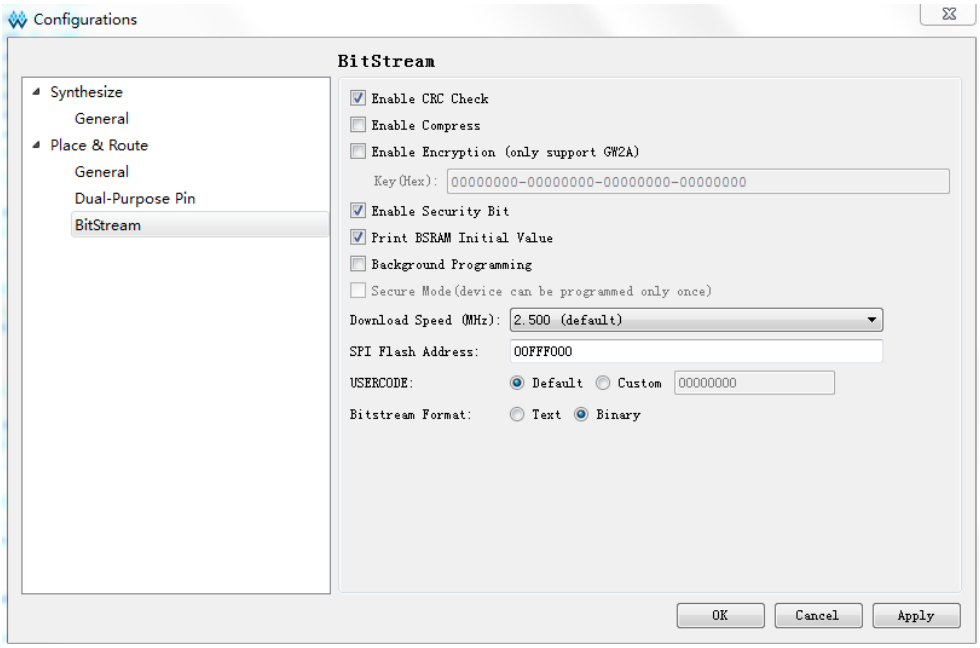
6.3 Configuration File Size

The Gowin bitstream format can be Text (ASCii) with annotations or Binary with no annotations. The file with a .fs suffix is a text format file. Lines beginning with “//” are annotations. The others is the bitstream data. The file with a .bin suffix is a binary format file, with no annotations. This binary format file is commonly used for embedded programming. Users can configure the bitstream file format in Gowin software.

1. Open Gowin software;

2. On the Process tab, right click Place & Route and then click “Configuration > Bitstream”;
3. In the options of Bitstream Format, select Text or Binary, as shown in Figure 6-9.

Figure 6-9 Bitstream Format generation



Gowin supports compressing bitstream data. The compression ratio is related to the user design. This manual only provides uncompressed configuration file sizes, as shown in Table 6-1.

Table 6-1 Gowin FPGA GW2AN-18X/9X Products Configuration File Size (Max.)

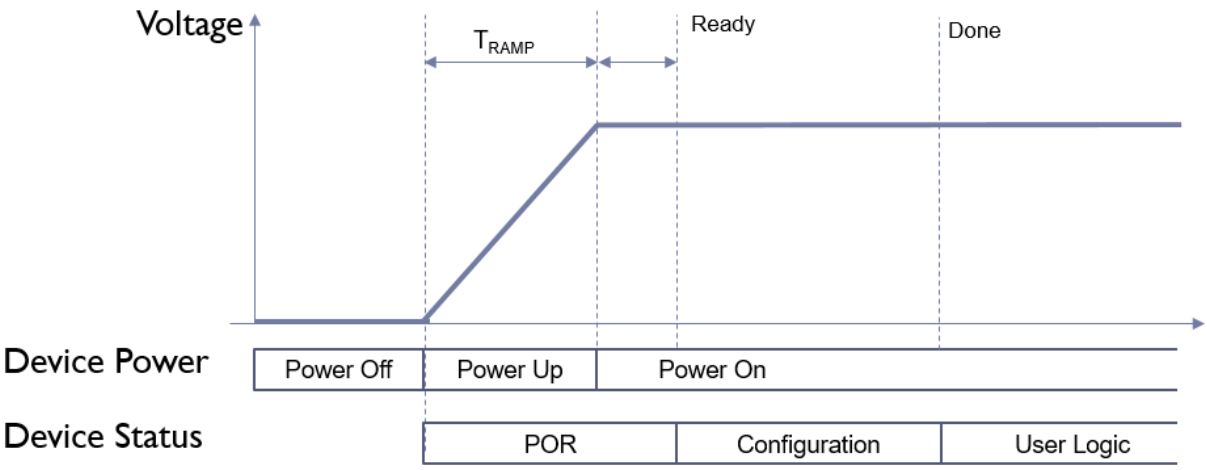
LUTs	Max. Configuration File Size
8,640	435 KBytes
20,736	887 KBytes

Note!

The data in the table is the file size in binary format, and the configuration file is not compressed.

6.4 Configuration File Loading Time

Gowin FPGA can be used as Master to read bitstream files from Flash and configure SRAM, including Autoboot mode and MSPI mode. In Autoboot mode, FPGA reads bitstream files from internal Flash. In MSPI mode, FPGA reads bitstream files from external Flash. When the FPGA is powered on and ready, it starts to read bitstream files, and when the loading is done, the FPGA enters the User Logic state, as shown in the figure below.



GW2AN-18X/9X devices support Quad SPI mode, that is, it can read and configure FPGA from the embedded Flash automatically. The default frequency of reading configuration file is 100Mhz. 4-bit data is loaded per SPI clock, and the required loading time can be calculated according to the file size. The clock frequency for MSPI to read SPI Flash can be up to 100Mhz. Note that the FastRead_n pin should be grounded at the same time when Fast Read SPI（0x0B）is used.

The loading time is different according to the configuration file size, the loading frequency, and the loading bits per clock.

The loading time in AUTO BOOT mode is as shown in Table 6-2.

Table 6-2 Loading Time in Autoboot Mode

LUT	Max. Configuration File Size	Loading frequency =100MHz time required (μs) (Quad SPI)
10,368	252 KBytes	6.4
20,736	887 KBytes	22

The loading time listed above is for reference only. The total time required from the device power-up to the completion of configuration includes the power supply ramp time (Tramp), device initialization time, and configuration time. The power supply ramp time is dependent on the characteristics of the power supply device. Therefore, the time from power-up to the completion of loading of the FPGA can be estimated using the following formula:

Autoboot QMSPI Mode:

$T_{loading\ time}$ = POR time + Number of Data Stream Bits /4/ Loading Frequency

QMSPI MSPI Mode:

$T_{loading\ time}$ = POR time + Number of Data Stream Bits / Loading Frequency

The POR time for GW2AN-18X/9X is about 6.3 ms.

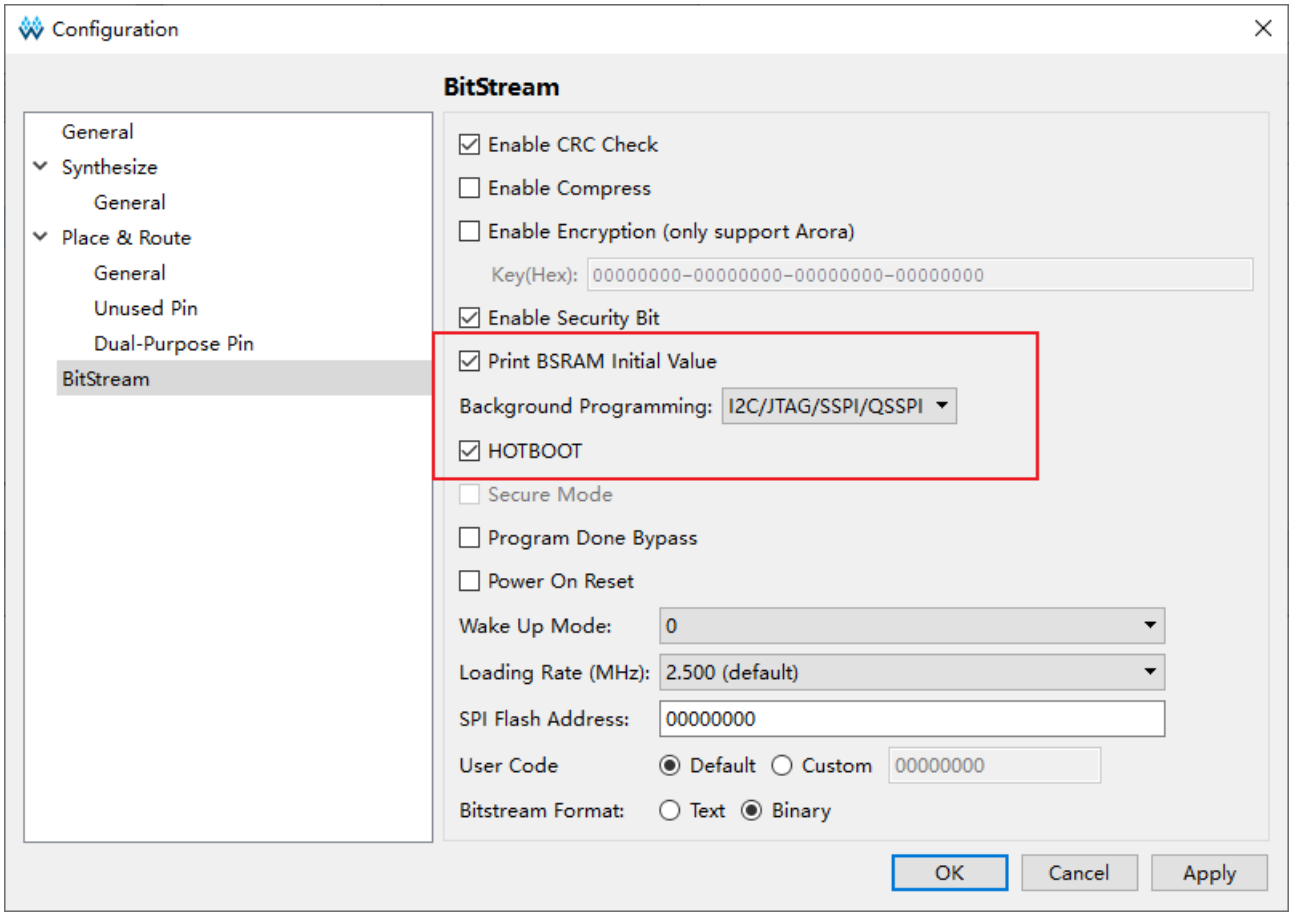
6.5 Background Programming

Background programming allows you to upgrade data files in Flash

without affecting the current operation. The IO states can be maintained when a new data stream file is being loaded.

The GW2AN-18X/9X series supports background programming under the JTAG/I2C/SSPI/QSSPI protocol and it can be supported by selecting this function in the IDE without modifying the programming timing.

Figure 6-10 The Background Programming Option



7 Safety Precautions

Security is a key factor for users to design FPGA. Combined with GOWINSEMI devices features, Gowin programmer offers a series of safety precautions, which provides a perfect security mechanism for users' bitstream data.

Safety precautions consist of three stages:

- Before configuration, Gowin programmer checks the validity of the bitstream;
- During configuration, GOWINSEMI device verifies the accuracy of the transmission data in real time;
- After configuration, GOWINSEMI device enters the working state, masking any readback requests.

The details of the three stages are as follows:

Before Configuration

Gowin programmer can be used to configure GowinFPGA by following the steps outlined below.

1. Connect the device that needs to be configured;
2. Start Gowin programmer to start scanning, and the connected FPGA devices can be identified automatically;
3. Select the bitstream and configuration mode to configure the device.

During the process outlined above, Gowin programmer will read the connected device ID first, and then compare this with the bitstream ID that users selected. The configuration can only proceed when the two IDs are consistent, or the bitstream selected by users will be regarded as illegal data, resulting in configuration failure.

Note!

GOWINSEMI products have specific IDs that distinguish them from the other series of products. The bitstream generated by Gowin Software contains an ID verification directive, as such, users only need to select the specific device when creating a new project.

During Configuration

The device reads and verifies the bit stream ID first, and configuration starts if verification passes. To prevent bitstream modifications or possible transmission errors, GOWINSEMI devices adopt CRC to ensure bitstream is written in correctly. The specific process is outlined below.

Following each address segment of the bitstream generated by Gowin software, CRC is added. GOWINSEMI devices generate CRC in the process of receiving data and compares them with the check codes received. If a CRC error is detected, any data transmitted following this error will be ignored. The "DONE" indicator will not light up after configuration, and the CRC error message will be displayed on the Gowin programmer interface.

After Configuration

After configuration, the device bitstream will be loaded to the SRAM or on-chip Flash according to the user mode selected.

- If the data is loaded to the SRAM, Gowin software will set the security bit automatically in the process of bitstream generation, and no user can read SRAMs.
- If the data is loaded to the on-chip Flash, the Flash will be configured as the AUTO BOOT mode after Flash configuration is complete. Any reading requests will be prohibited.
- It’s recommended to use AES Encryption for GW2AN-18X/9X devices to generate bitstream data.

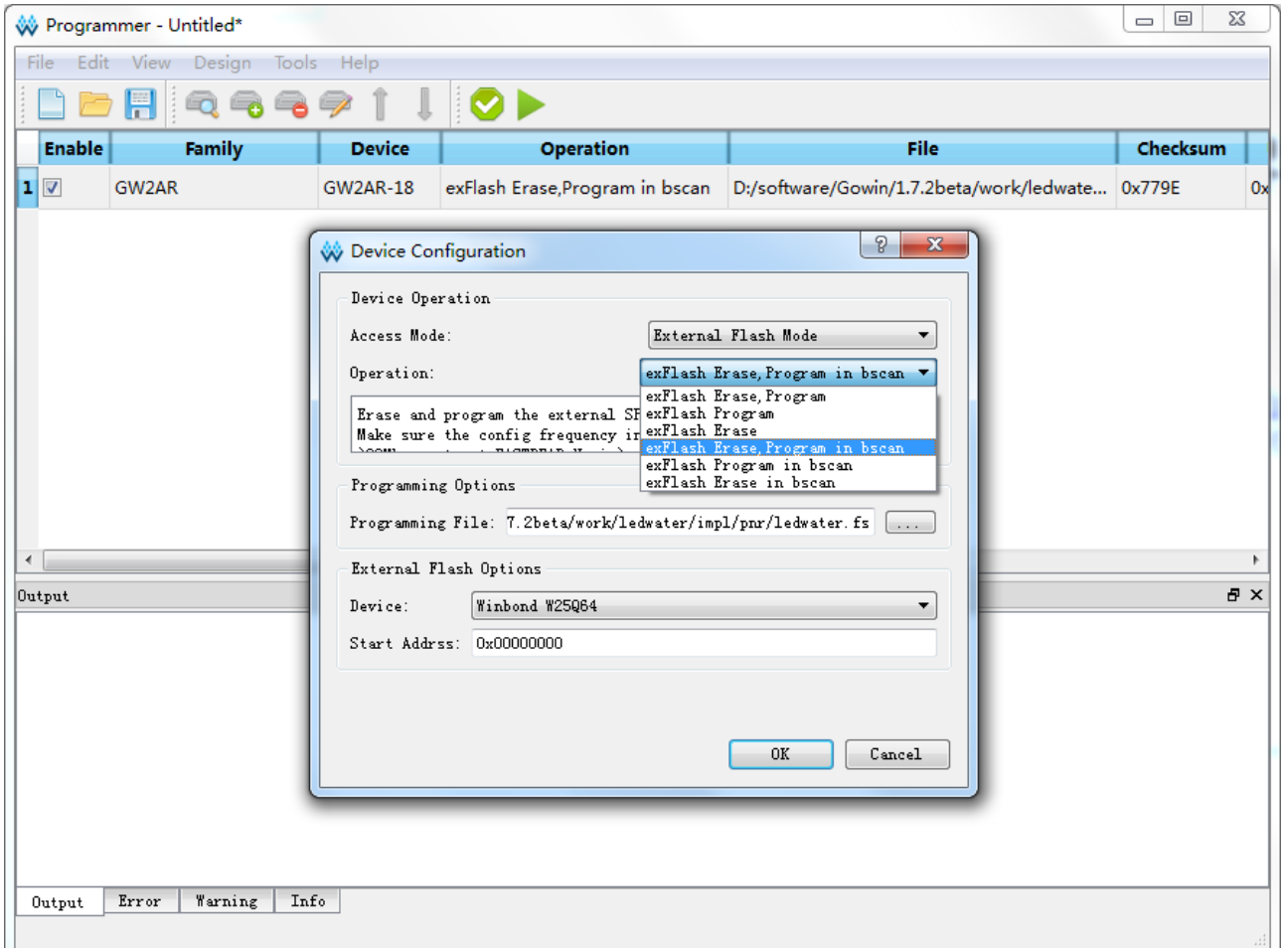
8 Boundary Scan

The boundary scan operation is an extension of the JTAG configuration mode. The scanning chains contain long chain and short chain. The long chain is mainly combined with BSDL file for device testing. The short chain is mainly used to erase and read and write the external Flash on the FPGA chain.

To perform a boundary scan, follow the steps outlined below:

1. Connect the FPGA development board to the PC and then power up;
2. Open Gowin programmer and scan the connected devices;
3. Double-click in the "Operation" field and select "External Flash Mode" and the related bscan operation, as shown in Figure 8-1.

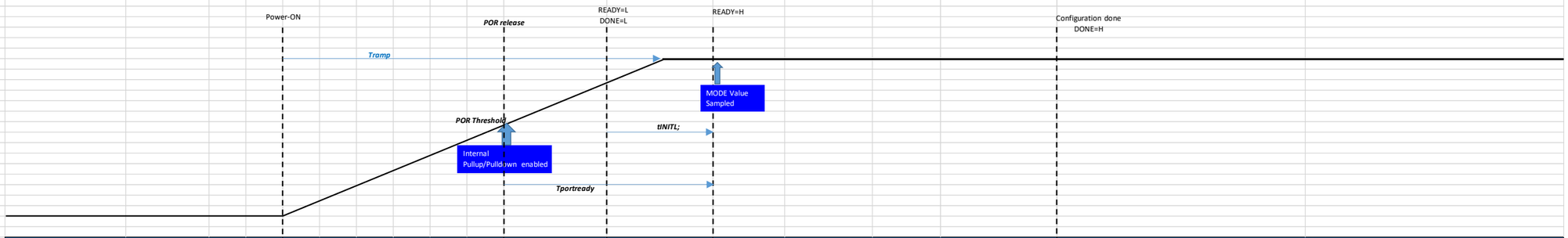
Figure 8-1 Boundary Scan Operation Schematic Diagram



The boundary scan operation can only be performed on the external Flash of FPGA and cannot be used to program the embedded Flash or SRAM. This operation is irrelevant with the FPGA MODE value, but it is slower than that of the external Flash programming via JTAG.

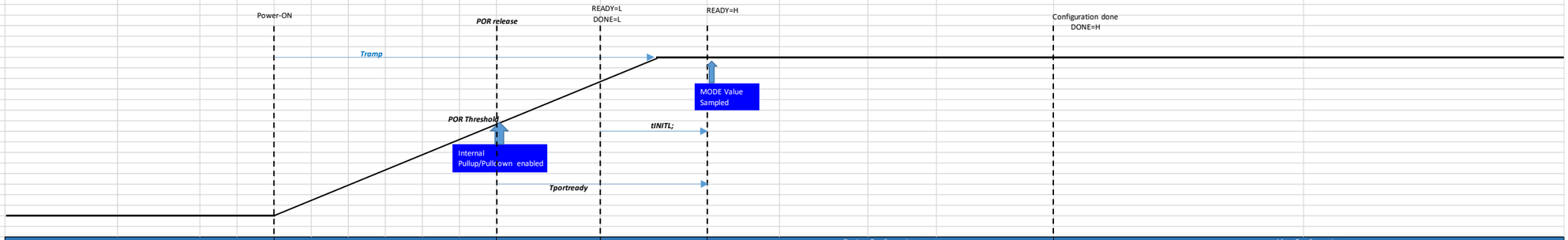
Appendix A Pin Status Information

Table A-1 Pin States at Each Stage for GW2AN-18X & 9X Devices (MSPI, MODE=00)



Stage			Power-ON to POR Voltage	POR to Start initialization	During initialization	During Configuration: Internal State	During Configuration: External Pull	After Configuration: Default	After Configuration: EDA settings for GPIO
State of Used GPIOs			HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Depends on FloorPlanner by user constrints	-
State of Unused GPIOs			HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Hi-Z with Weak pullup	Depends on EDA settings
Configuration IO	**READY**		HI-Z without pull	Open drain output 0 with Weak Pullup		Open drain output 1 with Weak Pullup	Pullup(Recommended)	READY (Open drain output 1 with Weak Pullup)	Input or Output
	RECONFIG_N		HI-Z without pull	Hi-Z with Weak Pullup		Input with Weak Pullup	Pullup(Recommended)	RECONFIG_N (Input with Weak Pullup)	Output
	DONE		HI-Z without pull	Open drain output 0 with Weak Pullup		Open drain output 0 with Weak Pullup	Pullup(Recommended)	DONE (Open drain output 1 with Weak Pullup)	Input or Output
	JTAG	JTAGSEL_N	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	JTAGSEL_N (Input with Weak pullup)	Input or Output
		TCK	HI-Z without pull	Hi-Z with Weak Pulldown		Input with Weak Pulldown	Pulldown	TCK (Input with Weak Pulldown)	Input or Output
		TMS	HI-Z without pull	Hi-Z with Weak Pullup		Input with Weak Pullup	-	TMS (Input with Weak Pullup)	Input or Output
		TDI	HI-Z without pull	Hi-Z with Weak Pullup		Input with Weak Pullup	-	TDI (Input with Weak Pullup)	Input or Output
		TDO	HI-Z without pull	Hi-Z with Weak Pullup		Output	-	TDO	Input or Output
	MSPI	**FASTRD_N**/D3	HI-Z without pull	Hi-Z with Weak Pulldown		Input with Weak Pulldown	-	FASTRD_N (Input with Weak Pullup)	-
	SSPI (Non-SSPI mode)	**SCLK**	HI-Z without pull	Hi-Z with Weak Pullup		Hi-Z with Weak Pullup	-	Input with Weak Pullup	Input or Output
		SSPI_CS_N/D0	HI-Z without pull	Hi-Z with Weak Pullup		Hi-Z with Weak Pullup	-	Input with Weak Pullup	Input or Output
		SI/D2	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Input with Weak Pullup	Input or Output
		SO/D1	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Hi-Z with Weak Pullup	Input or Output
		CLKHOLD_N/DIN	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Input with Weak Pullup	Input or Output
		QSSPI_WPN	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Input with Weak Pullup	Input or Output
	I2C (Non-I2C mode)	SCL	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	External Pullup recommended for packages with fixed MODE1=0	Hi-Z with Weak Pullup	Input
		SDA	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	External Pullup required for packages with fixed MODE1=0	Hi-Z with Weak Pullup	Input or Output
	SERIAL (Non-SERIAL mode)	CLKHOLD_N/**DIN**	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Input with Weak Pullup	Input or Output
		DOUT/WE_N	HI-Z without pull	Hi-Z with Weak PullDown		Output	-	GPIO Input or Output	-
	CPU (Non-CPU mode)	**SCLK**	HI-Z without pull	Hi-Z with Weak Pullup		Hi-Z with Weak Pullup	-	Input with Weak Pullup	Input or Output
		DOUT/**WE_N**	HI-Z without pull	Hi-Z with Weak PullDown		Output	-	GPIO Input or Output	-
		CLKHOLD_N/DIN	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Input with Weak Pullup	Input or Output
		D7-D4	HI-Z without pull	Hi-Z with Weak PullDown		Hi-Z with Weak PullDown	-	GPIO Input or Output	-
		FASTRD_N/**D3**	HI-Z without pull	Hi-Z with Weak Pulldown		Input with Weak Pulldown	-	Input with Weak Pullup	-
		SI/**D2**	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Input with Weak Pullup	Input or Output
		SO/**D1**	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Hi-Z with Weak Pullup	Input or Output
		SSPI_CS_N/**D0**	HI-Z without pull	Hi-Z with Weak Pullup		Hi-Z with Weak Pullup	-	Input with Weak Pullup	Input or Output

Table A-2 Pin States at Each Stage for GW2AN-18X & 9X Devices (AUTOBOOT, MODE=01)



Stage			Power-ON to POR Voltage	POR to Start initialization	During initialization	During Configuration		After Configuration	
						Internal State	External Pull	Default	EDA settings for GPIO
State of Used GPIOs			HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Depends on FloorPlanner by user constrints	-
State of Unused GPIOs			HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	Hi-Z with Weak pullup	Depends on EDA settings
Configuration IO	**READY**		HI-Z without pull	Open drain output 0 with Weak Pullup		Open drain output 1 with Weak Pullup	Pullup(Recommended)	READY (Open drain output 1 with Weak Pullup)	Input or Output
	RECONFIG_N		HI-Z without pull	Hi-Z with Weak Pullup		Input with Weak Pullup	Pullup(Recommended)	RECONFIG_N (Input with Weak Pullup)	Output
	DONE		HI-Z without pull	Open drain output 0 with Weak Pullup		Open drain output 0 with Weak Pullup	Pullup(Recommended)	DONE (Open drain output 1 with Weak Pullup)	Input or Output
	JTAG	JTAGSEL_N	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	JTAGSEL_N (Input with Weak pullup)	Input or Output
		TCK	HI-Z without pull	Hi-Z with Weak Pulldown		Input with Weak Pulldown	Pulldown	TCK (Input with Weak Pulldown)	Input or Output
		TMS	HI-Z without pull	Hi-Z with Weak Pullup		Input with Weak Pullup	-	TMS (Input with Weak Pullup)	Input or Output
		TDI	HI-Z without pull	Hi-Z with Weak Pullup		Input with Weak Pullup	-	TDI (Input with Weak Pullup)	Input or Output
		TDO	HI-Z without pull	Hi-Z with Weak Pullup		Output	-	TDO	Input or Output
	MSPI	**FASTRD_N**/D3	HI-Z without pull	Hi-Z with Weak Pulldown		Input with Weak Pulldown	-	FASTRD_N (Input with Weak Pullup)	-
	SSPI	**SCLK**	HI-Z without pull	Hi-Z with Weak Pullup		Hi-Z with Weak Pullup	-	SCLK (Input with Weak Pullup)	Input or Output
		SSPI_CS_N/D0	HI-Z without pull	Hi-Z with Weak Pullup		Hi-Z with Weak Pullup	-	SSPI_CS_N (Input with Weak Pullup)	Input or Output
		SI/D2	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	SI (Input with Weak Pullup)	Input or Output
		SO/D1	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	SO (output)	Input or Output
		CLKHOLD_N/DIN	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	CLKHOLD_N (Input with Weak Pullup)	Input or Output
		QSSPI_WPN	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	QSSPI_WPN (Input with Weak Pullup)	Input or Output
	I2C	SCL	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	External Pullup recommended for packages with fixed MODE1=0	SCL (Hi-Z with Weak Pullup)	Input
		SDA	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	External Pullup required for packages with fixed MODE1=0	SDA (Hi-Z with Weak Pullup)	Input or Output
	SERIAL (Non-SERIAL mode)	CLKHOLD_N/**DIN**	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	CLKHOLD_N (Input with Weak Pullup)	Input or Output
		DOUT/WE_N	HI-Z without pull	Hi-Z with Weak PullDown		Output	-	GPIO Input or Output	-
	CPU (Non-CPU mode)	**SCLK**	HI-Z without pull	Hi-Z with Weak Pullup		Hi-Z with Weak Pullup	-	SCLK (Input with Weak Pullup)	Input or Output
		DOUT/**WE_N**	HI-Z without pull	Hi-Z with Weak PullDown		Output	-	GPIO Input or Output	-
		CLKHOLD_N/DIN	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	CLKHOLD_N (Input with Weak Pullup)	Input or Output
		D7-D4	HI-Z without pull	Hi-Z with Weak PullDown		Hi-Z with Weak PullDown	-	GPIO Input or Output	-
		FASTRD_N/**D3**	HI-Z without pull	Hi-Z with Weak Pulldown		Input with Weak Pulldown	-	FASTRD_N (Input with Weak Pullup)	-
		SI/**D2**	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	SI (Input with Weak Pullup)	Input or Output
		SO/**D1**	HI-Z without pull	Hi-Z with Weak Pulldown		Hi-Z with Weak Pulldown	-	SO (output)	Input or Output
		SSPI_CS_N/**D0**	HI-Z without pull	Hi-Z with Weak Pullup		Hi-Z with Weak Pullup	-	SSPI_CS_N (Input with Weak Pullup)	Input or Output

Appendix B Example File Information

Table B-1 Example File Information

Gowin FPGA Device	Configuration Mode	sram or flash	Code Testing Platform	Notes
GW2AN-9X/18X	jtag	sram/flash	stm32	-
	sspi/qspi	flash	stm32	-
	i2c	sram/flash	stm32	There are examples of mixed use of goConfig_i2c.
	jtag&spi	flash	stm32	The data programming is implemented via SPI, and the state machine transitions are controlled through JTAG..
others	Huffman Coding	sram	stm32	Software compression algorithm. Suitable for embedded devices with limited storage resources. b3Compress is a compression tool for PC.

GOWIN

PROGRAMMING FOR THE FUTURE