Arora V 15K FPGA Products

Programming and Configuration Guide

UG720-1.0.1E, 10/11/2025

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

GOWIN is the trademark of Guangdong Gowin Semiconductor Corporation and is registered in China and/or other countries or regions. 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 denotes, 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
08/01/2024	1.0E	Preliminary version published.
06/12/2025	1.0E	Initial version published.
10/11/2025	1.0.1E	● 6.1 Multi Boot Process updated. ● Figure 3-27 The Flow Chart of SRAM Configuration via SSPI, Figure 4-2 Power Recycle Timing, and Figure 4-5 POR Timing updated.

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 3

2 Glossary 4

3 Configuration Interface 6

3.1 Configuration Mode 6
3.2 Configuration Pins 6
3.3 Configuration Clock for Master Modes 15
3.4 JTAG Configuration Mode 15
3.4.1 Connection Diagram for the JTAG Configuration Mode 15
3.4.2 JTAG Configuration Timing 16
3.4.3 JTAG Configuration Process 17
3.5 SSPI Configuration Mode 28
3.5.1 Connection Diagram for SSPI Configuration Mode 28
3.5.2 SSPI Configuration Timing 30
3.5.3 SSPI Configuration Instructions 31
3.5.4 SRAM Configuration via SSPI 34
3.5.5 Flash Programming via SSPI 34
3.5.6 SRAM Configuration via SSPIx4 36
3.6 MSPI Configuration Mode 37
3.6.1 Connection Diagram for MSPI Configuration Mode 37
3.6.2 MSPI Configuration Timing 39
3.6.3 MSPI 40
3.7 CPU Configuration Mode 41
3.7.1 Connection Diagram for CPU Configuration Mode 41
3.7.2 Configuration Timing 42
3.7.3 Continuous Data Loading 43
3.7.4 Non-continuous Data Loading 43

3.8 SERIAL Configuration Mode 44
3.8.1 Connection Diagram for SERIAL Configuration Mode 44
3.9 Daisy Chain 45
3.9.1 Serial Daisy Chain 45
3.9.2 Parallel Daisy Chain 46

4 Configuration Details 47

4.1 Configuration Notes 47
4.2 Configuration Sequence 50
4.2.1 Power-Up Timing 51
4.2.2 Initialization 51
4.2.3 Configuration 52
4.2.4 Wake-up 52
4.2.5 User Modes 52
4.3 Safety Precautions 52
4.4 Bitstream File Configuration 54
4.4.1 Configuration Options 54
4.4.2 Configure Data Encryption 55
4.4.3 Configuration File Size 63
4.4.4 Configuration File Loading Time 65
4.5 SPI Flash Selection 66

5 Status Register and Efuse Definition 67

5.1 Status Register 67
5.2 OTP Efuse 69

6 Multi Boot and Background Update 71

6.1 Multi Boot Process 71
6.2 Background Upgrade and Hotboot 72

List of Figures

Figure 3-1 Configuring Pin Reuse 14
Figure 3-2 Configuration Clock Structure in Master Modes 15
Figure 3-3 Connection Diagram for JTAG Configuration Mode 15
Figure 3-4 Connection Diagram of JTAG Programming External Flash 16
Figure 3-5 JTAG Configuration timing 16
Figure 3-6 TAP State Machine 17
Figure 3-7 Instruction Register Access Timing 18
Figure 3-8 Data Register Access Timing 19
Figure 3-9 Flow Chart of Reading ID Code State Machine 20
Figure 3-10 Access Timing of Reading ID Code Instruction- 0x11 21
Figure 3-11 Access Timing of Reading ID Code Data Register 21
Figure 3-12 SRAM Configuration Flow 23
Figure 3-13 Process View of Programming SPI Flash 24
Figure 3-14 Timing diagram of JTAG- SPI Sending 0x06 (Arora V) 25
Figure 3-15 Process of Program ming Flash in SVF Mode 25
Figure 3-16 OTP Configuration Flow 26
Figure 3-17 Process of reading OTP 27
Figure 3-18 SSPI Configuration Mode Connection Diagram 28
Figure 3-19 Multiple FPGA Connection Diagram 28
Figure 3-20 Connection Diagram of Programming External Flash via SSPI 29
Figure 3-21 SSPI Configuration Process Timing 30
Figure 3-22 SSPI Configuration Timing 30
Figure 3-23 Read ID Code Timing 32
Figure 3-24 Write Enable (0x15) Timing 32
Figure 3-25 Write Disable(0x3A00) Timing 32
Figure 3-26 Write Data (0x3B) Timing 33
Figure 3-27 The Flow Chart of SRAM Configuration via SSPI 34
Figure 3-28 The Flow Chart of Programming Flash via SSPI 35
Figure 3-29 The Flow Chart of SRAM Configuration via QSSPI 36
Figure 3-30 QSSPI Configuration Mode Connection Diagram 37
Figure 3-31 QSSPI Write Data (0x6B) Timing 37

Figure 3-32 Connection Diagram for MSPIx1 Configuration Mode 38
Figure 3-33 Connection Diagram for MSPIx2 Configuration Mode 38
Figure 3-34 Connection Diagram for MSPIx4 Configuration Mode 38
Figure 3-35 MSPI Configuration Timing 39
Figure 3-36 The Flow Chart of MSPI Mode 40
Figure 3-37 Connection Diagram for CPU Mode 41
Figure 3-38 CPU Mode Configuration Timing 42
Figure 3-39 Continuous Data Loading Timing 43
Figure 3-40 Non Continuous Data Loading Timing 43
Figure 3-41 Connection Diagram for SERIAL Mode 44
Figure 3-42 SERIAL Configuration Timing 44
Figure 3 - 43 Diagram of Daisy-Chain 45
Figure 3 - 44 Diagram of Parallel Daisy-Chain 46
Figure 4-1 Process of Power on 47
Figure 4-2 Power Recycle Timing 48
Figure 4-3 Trigger Timing 49
Figure 4-4 GOWINSEMI FPGA Configuration Process 50
Figure 4-5 POR Timing 51
Figure 4-6 Configuration Options 55
Figure 4-7 Encryption Key Setting Method 56
Figure 4-8 Setting the Decryption Key 57
Figure 4-9 AES Security Configure 58
Figure 4-10 Prepare 59
Figure 4-11 Read AES Key Flow 60
Figure 4-12 Program AES Key Flow 61
Figure 4-13 Program AES Key2 Flow 62
Figure 4-14 Lock AES Key Flow 63
Figure 4-15 Bitstream Format generation 64
Figure 4-16 Process of Power on 65
Figure 6-1 Multi- Configuration Flow 72

List of Tables

Table 1-1 Terminology and Abbreviations 1
Table 2-1 Glossary 4
Table 3-1 Configuration Modes 6
Table 3-2 Configuration Pin List (1 of 2) 7
Table 3-3 Configuration Pin List (2 of 2) 8
Table 3-4 Pin Definitions 9
Table 3-5 Pin Reuse Options 13
Table 3-6 JTAG Configuration Timing Parameters 17
Table 3-7 IDCODE 19
Table 3-8 Change of TDI and TMS Value in The Process of Sending Instructions 19
Table 3-9 SSPI Configuration Timing Parameters 31
Table 3-10 Configuration Instruction 31
Table 3-11 MSPI Configuration Timing Parameters 39
Table 3-12 CPU Configuration Timing Parameters 42
Table 3-13 SERIAL Configuration Timing Parameters 45
Table 4-1 Timing Parameters of Power on Again and RECONFIG_N triggering for Arora V FPGA Products 49
Table 4-2 Power Rails 51
Table 4-3 Arora V 15K FPGA Products Configuration File Size (Max.) 64
Table 4-4 Loading Time in MSPI Mode 65
Table 4-5 SPI Flash Operation Instruction 66
Table 5-1 Status Register 67
Table 5-2 OTP Definitions 69

1 About This Guide

1.1 Purpose

This guide mainly introduces general features and functions on programming and configuration of Arora V 15K FPGA products. 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:

- DS1113, GW5A Series of FPGA Products (Automotive) Data Sheet
- DS1111, GW5AT series of FPGA Products (Automotive) Data Sheet
- UG1224, GW5AT-15 Pinout
- UG1120, GW5ART-15 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	Description
Background Programming	Embedded / SPI Flash Background Programming	Programs the Embedded / SPI Flash without affecting the current functions and I/O state.
Bitstream	Bitstream Data	The data of configuring the FPGA SRAM
Bscan	Boundary Scan	Boundary Scan
Configuration	Configuration	The process of configuring the FPGA SRAM area
Configuration Data	Configuration Data	The data of configuring the FPGA SRAM
Configuration Mode	Configuration Mode	Configuration mode, determine the Configuration Data source.

Terminology and Abbreviations	Full Name	Description
CPU	Central Processing Unit	Central Processing Unit
CRC	Cyclic Redundancy Check	Cyclic Redundancy Check
Edit Mode	Edit Mode	The FPGA in Configuration mode or Programming mode
FPGA	Field Programmable Gate Array	Field Programmable Gate Array
FS file	Fuses file	ASCII file including the configuration data
GPIO	General Purpose Input Output	General Purpose Input Output
I2C（I^2C, IIC）	Inter-Integrated Circuit	I^2C
ID	Identification	Identification
IEEE	Institute of Electrical and Electronics Engineers	Institute of Electrical and Electronics Engineers
Internal Flash	Internal Flash	Same with Embedded Flash
JTAG	Joint Test Action Group	Joint Test Action Group
LSB	Least Significant Bit	Least Significant Bit (priority)
LUT	Look-up Table	Look-up Table
MSB	Most Significant Bit	Most Significant Bit (priority)
MSPI	Master Serial Peripheral Interface	Master Serial Peripheral Interface
Programming	Programming	The process of programming the Configuration Data to the Embedded Flash or the External Flash memory
SCL	Serial Clock	Clock line of I2C
SDA	Serial Data	Serial Data
Security Bit	Security Bit	Security Bit (readback the SRAM to keep high level)
SPI	Serial Peripheral Interface	Serial Peripheral Interface
SRAM	Static Random Access Memory	Static Random Access Memory
SSPI	Slave Serial Peripheral Interface	Slave Serial Peripheral Interface
TAP	Test Access Port	Test Access Port
User Mode	User Mode	The mode in which the FPGA executes the corresponding logic functions after the programming or configuration has been completed.

1.4 Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly using the information provided below.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

2 Glossary

This chapter presents an overview of the terms that are commonly used in the process of programming and configuring the Arora V 15K 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.
MODE[2:0]	A representation of the three MODE pin values associated with GowinCONFIG.
MSPI Configuration	As a master, FPGA is configured by reading bitstream from the external Flash via 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.
SERIAL Configuration	As a slave device, the bitstream data is written into the FPGA via the serial interface by the external master.
CPU Configuration	As a slave device, the bitstream data is written into the FPGA via the parallel interface (8-bit) by the external master.
MULTI BOOT	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 supports MSPI all support this mode.
Remote Upgrade	After a FPGA starts to work, if an upgrade is required, first write bitstream to an embedded or external Flash through remote operation, and then the 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).

Glossary	Meaning
Edit Mode	FPGA can be programmed and configured in this mode. All configuration pins cannot be reused as GPIOs. All GPIO outputs are high-impedance state except for background upgrades.
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. 32-bit user code can be written to the FPGA device through Gowin programmer.
Security Bit	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	Arora family of FPGA products support this feature. After the encrypted bitstream is written into 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 the matching fails.

3 Configuration Interface

3.1 Configuration Mode

Table 3-1 Configuration Modes

Configuration Mode	MODE[1:0][1]	Bus Width	Description
JTAG	XX[2]	-	Arora V FPGA products are configured by external Host via JTAG interface
MSPI[4]	11	x1, x2, x4	FPGA acts as Master and reads configuration data from external Flash (or other devices) via the SPI interface for configuration
SSPI	01	x1,x4	● FPGA acts as Slave and reads configuration data externally via DIN interface for configuration ● The FPGA automatically enters SSPI mode after MSPI loading, regardless of success or failure (CLKHOLD_N must be held high).
Slave SERIAL	00	x1	Arora V FPGA products are configured by external Host via the DIN interface
Slave CPU	10	x8, x16	Arora V FPGA products are configured by external Host via the DBUS interface

Note!

- [1] JTAG works in any mode with the highest priority.
- [2] The Slave CPU does not support 32-bit configurations and supports a maximum download frequency of 80MHz and a read frequency of <=40MHz.

3.2 Configuration Pins

The programming and configuration pins can be used as configuration pins and also can be reused as GPIOs. Users can configure the pins as required. Users also can configure them according to their configuration functions to meet specific requirements. Table 3-2 and Table 3-3 contain a list of all the configuration pins of the Arora V 15K FPGA products together with the details of the pins used in each configuration mode and the shared pins in chip packages.

Note!

When configuration I/Os are reused, note that some configuration I/Os have status requirements during power-on and configuration. Otherwise, the device programming

may be affected. When the number of common I/Os meets the requirement, it's not suggested to reuse the configuration I/Os.

Table 3-2 Configuration Pin List (1 of 2)

Pin Name	Bank	JTAG (Only)	Slave Serial	Master Serial	Master SPI		
					x1	x2	x4
MODE [1:0]	7	XX	M[1:0]=00	M[1:0]=11	M[1:0]=11	M[1:0]=11	M[1:0]=11
RECONFIG_N	3	√	√	√	√	√	√
READY	2	√	√	√	√	√	√
DONE	3	√	√	√	√	√	√
TCK	4	TCK	TCK	TCK	TCK	TCK	TCK
TMS	4	TMS	TMS	TMS	TMS	TMS	TMS
TDI	4	TDI	TDI	TDI	TDI	TDI	TDI
TDO	4	TDO	TDO	TDO	TDO	TDO	TDO
CCLK	2	-	CCLK	CCLK	CCLK	CCLK	CCLK
EMCCLK	2	-	-	EMCCLK	EMCCLK	EMCCLK	EMCCLK
CSI_B	3/4	-	-	-	-	-	-
DOUT_CSO_B	3/4	-	DOUT	DOUT	-	-	-
RDWR_B	3/4	-	-	-	-	-	-
MCS_N	3/4	-	-	-	MCS_N	MCS_N	MCS_N
D00_MOSI	3/4	-	-	-	MOSI	MOSI/D00	MOSI/D00
D01_DIN	3/4	-	DIN	DIN	MISO	D01	D01
D02	3/4	-	-	-	-	-	D02
D03	3/4	-	-	-	-	-	D03
SSPI_CS_N	3/4	-	-	-	-	-	-
D05_SI	3/4	-	-	-	-	-	-
D06_SSPI_CLK	3/4	-	-	-	-	-	-
D07_SSPI_WPN	3/4	-	-	-	-	-	-
D08_SO	3/4	-	-	-	-	-	-
CLKHOLD_N	3/4	-	-	-	-	-	-
D04	3/4	-	-	-	-	-	-
D[09-15]	3/4	-	-	-	-	-	-

Table 3-3 Configuration Pin List (2 of 2)

Pin Name	Bank	Slave CPU		Slave SPI	
		X8	X16	X1	X4
MODE [1:0]	7	M[1:0]=10	M[1:0]=10	M[1:0]=10	M[1:0]=01
RECONFIG_N	3	√	√	√	√
READY	2	√	√	√	√
DONE	3	√	√	√	√
TCK	4	TCK	TCK	TCK	TCK
TMS	4	TMS	TMS	TMS	TMS
TDI	4	TDI	TDI	TDI	TDI
TDO	4	TDO	TDO	TDO	TDO
CCLK	2	CCLK	CCLK	CCLK	CCLK
EMCCLK	2	-	-	-	-
CSI_B	3/4	CSI_B	CSI_B	-	-
DOUT_CSO_B	3/4	CSO_B	CSO_B	-	-
RDWR_B	3/4	RDWR_B	RDWR_B	-	-
MCS_N	3/4	-	-	-	-
D00_MOSI	3/4	D00	D00	-	-
D01_DIN	3/4	D01	D01	-	-
D02	3/4	D02	D02	-	-
D03	3/4	D03	D03	-	-
SSPI_CS_N	3/4	-	-	SSPI_CS_N	SSPI_CS_N
D05_SI	3/4	D05	D05	SSPI_SI	SSPI_IO0
D06_SSPI_CLK	3/4	D06	D06	SSPI_CLK	SSPI_CLK
D07_SSPI_WPN	3/4	D07	D07	SSPI_WPN	SSPI_IO2
D08_SO	3/4	D08	D08	SSPI_SO	SSPI_IO1
CLKHOLD_N	3/4	-	-	CLKHOLD_N	SSPI_IO3
D04	3/4	D04	D04	-	-
D[09-15]	3/4	-	D[09-15]	-	-

The configuration pins are as shown in Table 3-4.

Table 3-4 Pin Definitions

Pin Name	Functional Description
MODE	Configuration mode selection pin. As the selection pin of configuration modes, MODE is an input pin that has internal weak pull-up. The maximum bit width is 3 bits. After FPGA powers up or a low pulse triggers RECONFIG_N, the device enters the corresponding configuration mode in accordance with the MODE value. The same MODE value of the different GOWIN 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!** 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.
RECONFIG_N	As a configuration pin, RECONFIG_N is an input pin that has an internal weak pull-up. Active low, it is used to reconfigure the FPGA during programming configuration. FPGA can't be configured if RECONFIG_N is set to low. Keep high-level during power-on and configuration until the device configuration is complete. As a configuration pin, a low level signal with pulse width no less than 25ns is required for GowinCONFIG to reload bitstream data according to the MODE setting value. You can control the pin by writing logic to trigger the device to reconfigure as required. In the case of user logic control, ensure that the pin is kept high during power-on and configuration. **Note!** ● The RECONFIG_N pin cannot be reused as a GPIO. ● If you need to trigger RECONFIG_N through internal FPGA logic, you can connect a GPIO (with I/O standard of LVCMOS10/LVCMOS12/LVCMOS15/LVCMOS18/LVCMOS25/LVCMOS33) to RECONFIG_N via external circuitry. During power-up and configuration, this GPIO must remain in a high-impedance or pull-up state.
READY	Inout pins. Internal weak pull-up, open-drain output. Active-high. FPGA can be configured only when the READY signal is pulled up. It indicates that the FPGA can be configured or not. If the FPGA meets the configuration condition, the READY signal is high. If the power-on or reset process is incomplete, READY signal is low. Users can delay the active configuration by pulling down the READY signal externally. As a GPIO, it can be reused as an input or output type. If READY is used as an input GPIO, ensure that it’s not pulled down during the power-on and configuration process. Otherwise, The configuration may not work properly.
DONE	Inout pins. Internal weak pull-up, open-drain output. It indicates FPGA is configured successfully or not. DONE is pulled up after successfully configuring. As an output configuration pin, it indicates whether the FPGA configuration is successful: ● If DONE is high, the device has been awakened and enters into the working state. ● If DONE is low, the configuration process is incomplete or fails.

Pin Name	Functional Description
	Users can delay the wake up process by pulling down the DONE signal externally. 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 pin. If DONE is used as an input GPIO, ensures that it is not pulled down externally during power-on and configuration process. Otherwise, the FPGA will fail to enter the user mode after configuration.
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 pin. **Note!** As a GPIO, ensure that the pin status is stable during the power-on and loading process to prevent the JTAG command from being triggered by mistake.
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 pin.
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 pin.
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 pin.
CCLK	As a configuration clock pin, CCLK runs the synchronous FPGA configuration sequence in all modes except JTAG and SSPI mode. In slave mode: CCLK is an input and requires connection to an external clock source. In master mode: CCLK is an output as configuration source clock. **Note!** CCLK is the key clock signal, so good signal integrity must be ensured.
EMCCLK	As a configuration pin, it is an input pin. Used to configure the optional external clock input in a master mode (versus the internal configuration oscillator). For master modes: FPGA can optionally switch to EMCCLK as the clock source. For JTAG and slave modes: EMCCLK is ignored and can be left unconnected.
CSI_B	As a configuration pin, it is an input pin. It is a chip selection input signal in the CPU mode, active low. For CPU master mode, tie to GND directly or via a 1 kΩ (or less) resistor. For CPU slave mode: An external configuration controller can control CSI_B for selecting the active FPGA on the bus, or in a daisy-chain configuration, connect to the CSO_B pin of the upstream FPGA.
CSO_B	As a configuration pin, it is an output pin. It is a chip selection input signal in the CPU mode. It connects to the CSI_B pin of the downstream FPGA in a daisy-chain configuration.
DOUT	As a configuration pin, it is an output pin. DOUT is the data output for a serial configuration daisy-chain. For Serial and SPI (x1 only) modes: Connect to the DIN of the downstream FPGA in a serial configuration daisy-chain, as the data input of the following device.
RDWR_B	As a configuration pin, it is an input pin. As a Read/Write enable selection pin in CPU mode: it refers to Read when

Pin Name	Functional Description
	RDWR_B is high and Write when RDWR_B is low. As a GPIO, it can be used as an input or output pin.
MCS_N	As a configuration pin, it is an output pin. It is a chip selection signal in MSPI configuration mode, active low. As a GPIO, it can be used as an input or output pin.
MOSI	As a configuration pin, it is an output pin. Connects to the SI pin of the external Flash. The SPIx2 x4 line is bidirectional. Serial data output pin in MSPI configuration mode. As a GPIO, it can be used as an input or output pin.
MISO	As a configuration pin, it is an input pin. Connects to the SO pin of the external Flash. The SPIx2 x4 line is bidirectional. It is a serial data input pin in MSPI configuration mode. As a GPIO, it can be used as an input or output pin.
MI2	As a configuration pin, it is an output pin. The SPIx4 line is bidirectional. MSPI write protection pin: When the output is high, the MSPI operation is valid; When the output level is low, the MSPI operation is invalid. The V_{CCIO} of the corresponding bank can be pulled up through the 4.7k resistor. As a GPIO, it can be used as an input or output pin.
MI3	As a configuration pin, it is an output pin, internal weak pull-down. The SPIx4 line is bidirectional. MSPI clock holding pin: active-low. The V_{CCIO} of the corresponding bank can be pulled up through the 4.7k resistor. As a GPIO, it can be used as an input or output pin.
DIN	As a configuration pin, it is an input pin. It is a serial data input pin. For Serial and MSPI modes: DIN receives serial data from the data source and collects data at the CCLK rising edge in the default configuration. For CPU modes, DIN is a multi-function pin that functions as the D01 data pin. As a GPIO, it can be used as an input or output pin.
D0~D15	Inout pins For CPU mode: D0~D31 are data in/out pins. The FPGA will automatically detect the data width of x8, or x16. 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. A 4.7k pull-up resistor to the corresponding Bank V_{CCIO} is recommended. 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 pin.
SSPI_CLK	As a configuration pin, it is an input pin. A 4.7k pull-up resistor to the corresponding Bank V_{CCIO} is recommended. It is a clock input pin of SSPI configuration mode. As a GPIO, it can be used as an input or output pin.
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 pin.
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 pin.
CLKHOLD_N	As a configuration pin, it is an input pin. The V_{CCIO} of the corresponding bank can be pulled up through the 4.7k resistor.

Pin Name	Functional Description
	SSPI clock holding pin: active-low. As a GPIO, it can be used as an input or output pin.
SSPI_WPN	As a configuration pin, it is an input pin. The V_{CCIO} of the corresponding bank can be pulled up through the 4.7k resistor. A write protection pin in SSPI mode: SSPI is valid when the input is high, SSPI is invalid when the input is low. As a GPIO, it can be used as an input or output pin.

Configuration Pin Multiplexing

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.

The reuse options for the configuration pins are detailed in Table 3-5.

Table 3-5 Pin Reuse Options

Name	Options	Description
JTAG PORT[1]	Default Status	TMS, TCK, TDI, and TDO are dedicated configuration pins.
	Set as GPIOs	TMS, TCK, TDI, and TDO are used as GPIOs after configuration.
CPU PORT	Default Status	CSI_B, CSO_B, RDWR_B, D[0- 31], and CCLK are dedicated configuration pins.
	Set as GPIOs	CSI_B, CSO_B, RDWR_B, D[0- 31], and CCLK are used as GPIOs after configuration.
SSPI PORT	Default Status	SSPI_CS_N, SSPI_CLK, SI, SO, CLKHOLD_N, and SSPI_WPN are dedicated configuration pins.
	Set as GPIOs	SSPI_CS_N, SSPI_CLK, SI, SO, CLKHOLD_N, and SSPI_WPN are used as GPIOs after configuration.
MSPI PORT	Default Status	MCS_N, DIN, MOSI, D02, D03, and CCLK are dedicated configuration pins.
	Set as GPIOs	MCS_N, DIN, MOSI, D02, D03, and CCLK are used as GPIOs after configuration.
SERIAL	Default Status	DIN, DOUT, and CCLK are dedicated configuration pins.
	Set as GPIOs	DIN, DOUT, and CCLK are used as GPIOs after configuration.
RECONFIG_N	Default Status	Dedicated configuration pins.
READY	Default Status	Dedicated configuration pins.
	Set as GPIOs	Used as GPIOs after configuration.
DONE	Default Status	Dedicated configuration pins.
	Set as GPIOs	Used as GPIOs after configuration.

Note!

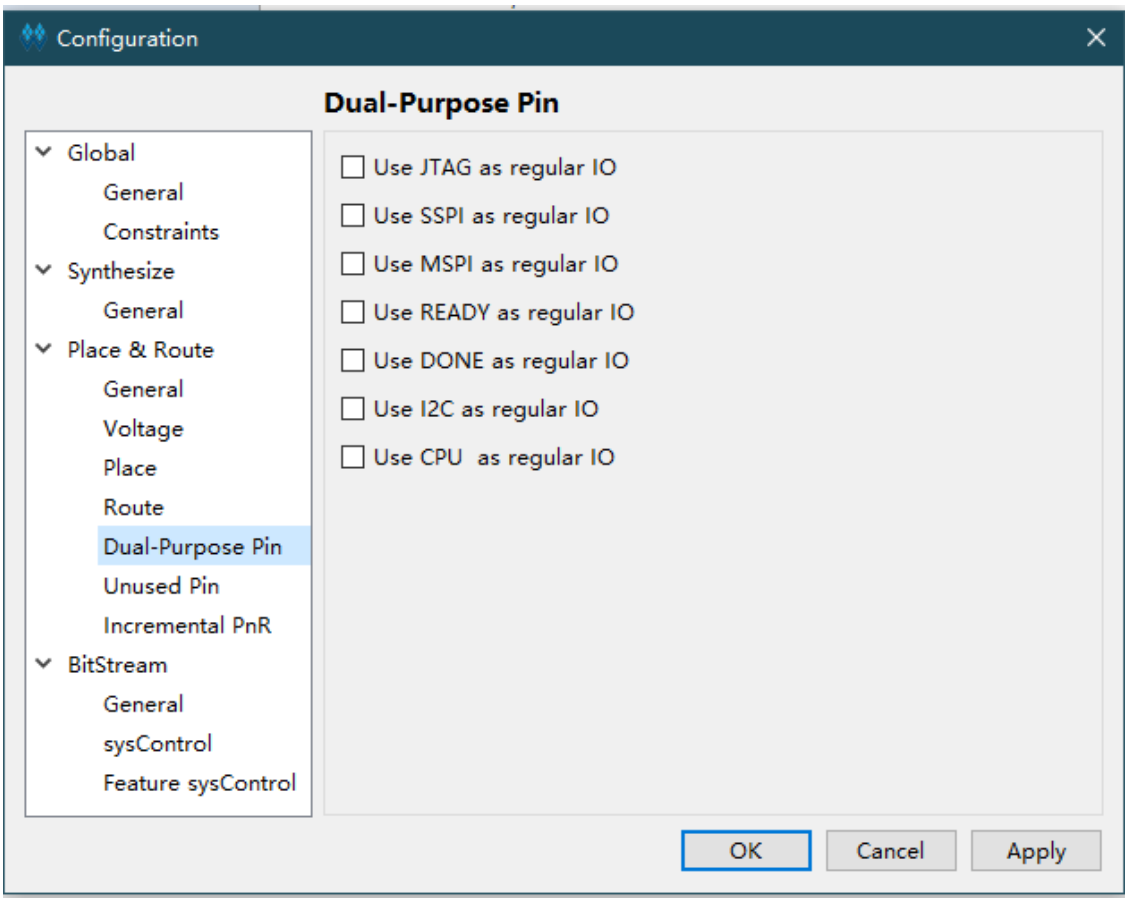
- [1] The JTAG_SEL_N signal can be controlled internally by logic to configure JTAG pins as GPIOs.
- [1] As a GPIO, ensures that the TCK/TMS is stable during the power-on and programming process to prevent the JTAG command from being triggered by mistake.

Pin Reuse Configuration

Users can configure pin reuse via Gowin Software.

1. Open the corresponding project in Gowin Software;
2. Select “Project > Configuration > Dual-Purpose Pin” in the menu bar, as shown in Figure 3-1;
3. Check the corresponding options to set the pin reuse.

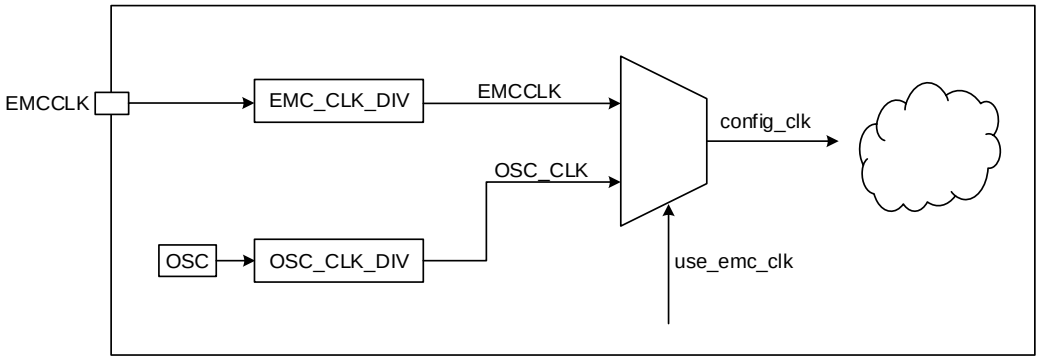
Figure 3-1 Configuring Pin Reuse



3.3 Configuration Clock for Master Modes

There are two options for the configuration clock in master modes (master CPU mode, master SERIAL mode, and master SPI mode): On-chip oscillator and external reference clock. On-chip oscillator is used as the clock source by default. After configured the emc_clk and EMC_CLK_DIV by the EDA tool, external clock can also be used as configuration clock source and contains multiple frequency division options. The configuration clock structure is as shown in the figure below.

Figure 3-2 Configuration Clock Structure in Master Modes



3.4 JTAG Configuration Mode

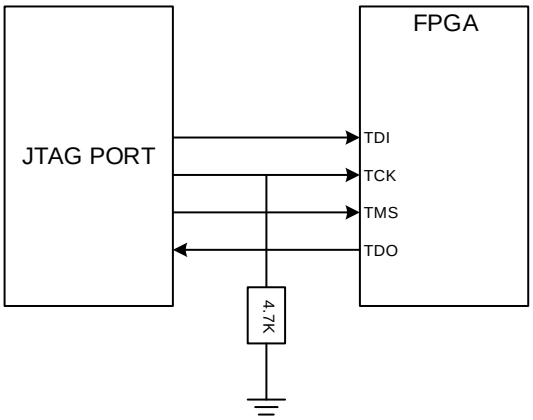
The JTAG configuration mode of Gowin FPGA products conforms to the IEEE1532 standard and the IEEE1149.1 boundary scan standard.

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.

3.4.1 Connection Diagram for the JTAG Configuration Mode

The connection diagram in the JTAG configuration mode is shown in Figure 3-3Figure 3-3 Connection Diagram for JTAG Configuration Mode.

Figure 3-3 Connection Diagram for JTAG Configuration Mode

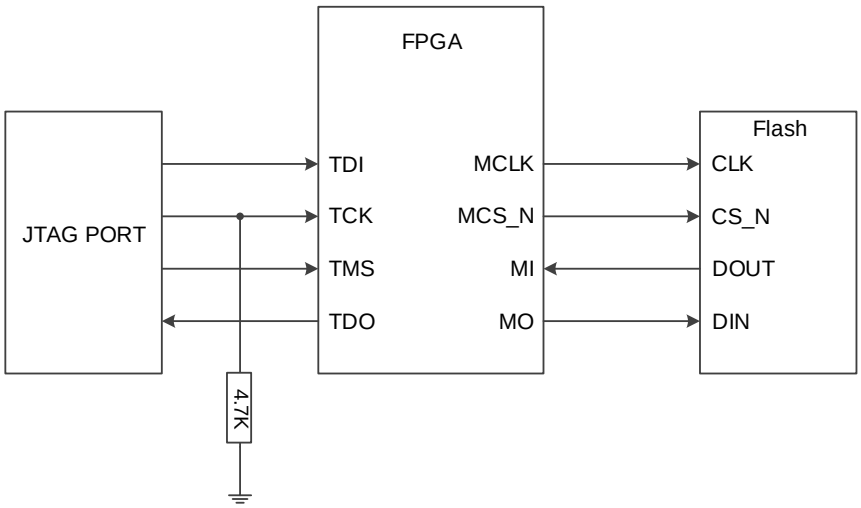


Note!

The clock frequency for JTAG configuration mode cannot be higher than 40MHz.

In addition to configuring SRAM, JTAG mode can also be used to program the external SPI Flash. Figure 3-4 shows the external SPI Flash programming.

Figure 3-4 Connection Diagram of JTAG Programming External Flash



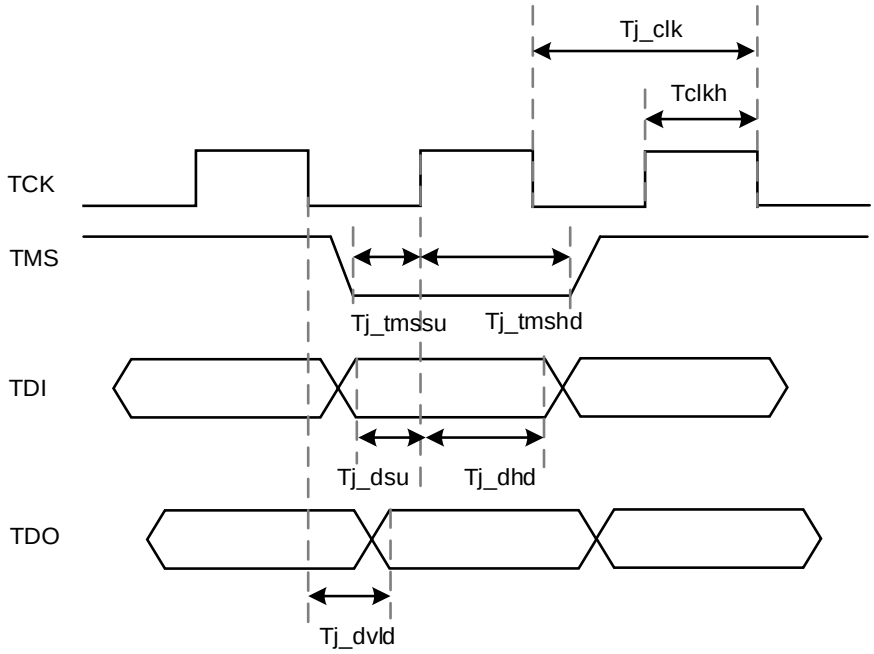
Note!

The figure above is the connection diagram of Programming External Flash via the JTAG interface.

3.4.2 JTAG Configuration Timing

See Figure 3-5 for the timing of JTAG mode.

Figure 3-5 JTAG Configuration timing



See Table 3-6 for the description of timing parameters.

Table 3-6 JTAG Configuration Timing Parameters

Name	Description	Min.	Max.	Unit
T_{j_clk}	JTAG TCK clock period	10	–	ns
T_{j_tmssu}	TMS setup time	3.0	–	ns
T_{j_tmshd}	TMS hold time	0	–	ns
T_{j_dsu}	TDI setup time	3.0	–	ns
T_{j_dhd}	TDI hold time	0	–	ns
T_{j_dvld}	delay of TCK to TDO	–	3.0	ns
T_{clkh}	The time of clock high level	(clock cycle) *45%	(clock cycle) *55%	–

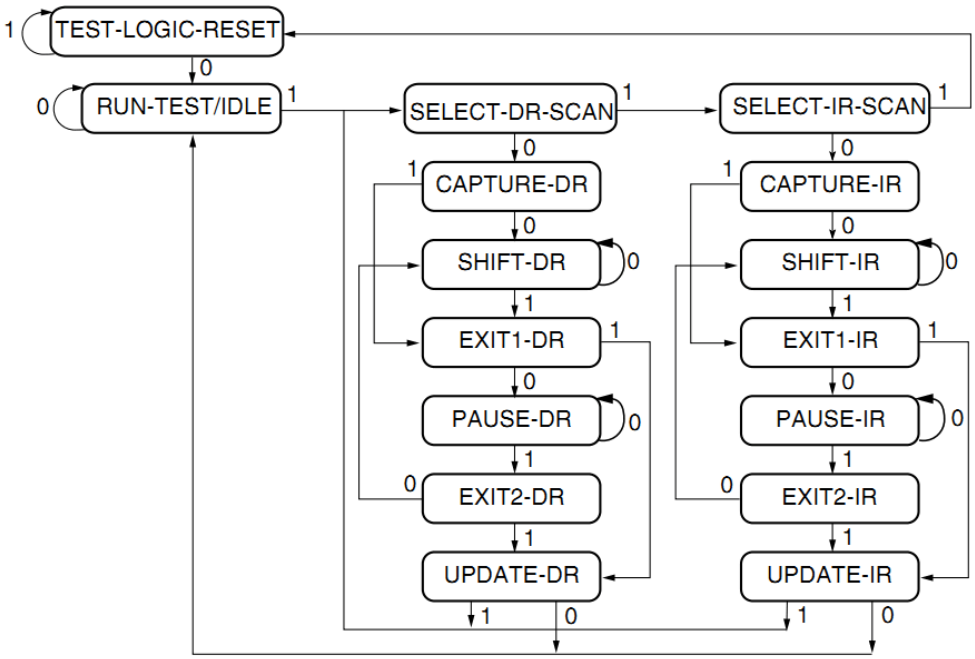
3.4.3 JTAG Configuration Process

JTAG Instructions

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 3-6.

Figure 3-6 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 exiting the Shift_DR or Shift_IR state;
- 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 all be sent when the sate machine returns to Run-Test-Idle, as shown in Figure 3-7.

During the data register scanning operation, the data or instructions are sent to the DR in the Shift_DR state, as shown in Figure 3-8. The data is sent in LSB way or MSB way depending on specific operations.

Figure 3-7 Instruction Register Access Timing

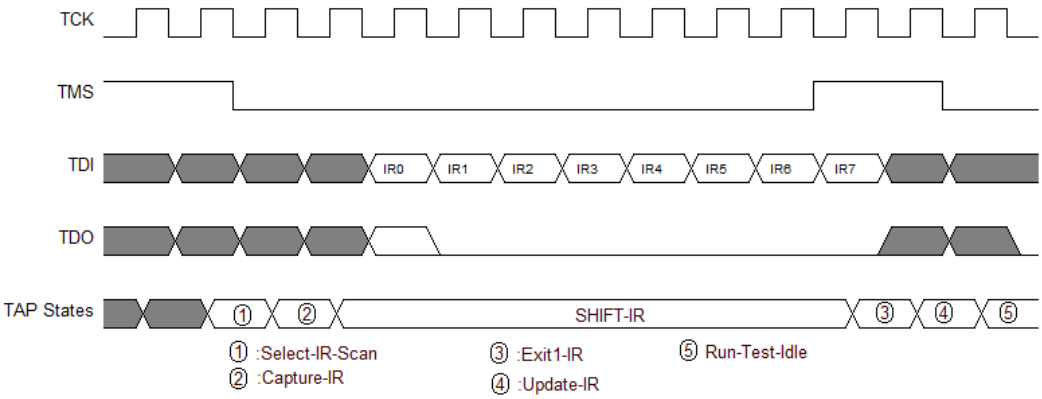
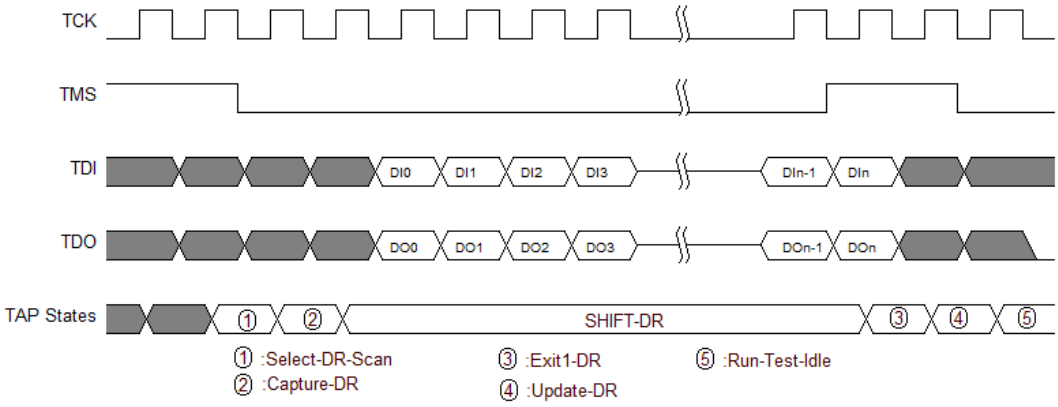


Figure 3-8 Data Register Access Timing



Note!

- The total length of the instruction register is 8 bits in Arora V FPGA products.
- The length of the data register may 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 are listed in the table below.

Table 3-7 IDCODE

Device	IDCODE
GW5AT-15	H0001681B
GW5ART-15	H0001681B

The instruction to read FPGA is 0x11. Take the GW5AT-15 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 the Read ID instruction (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 set high before sending MSB. Table 3-8 lists the change of TDI and TMS value during sending 0x11 in 8 clock cycles, as shown in Figure 3-10.

Table 3-8 Change of TDI and TMS Value in The Process of Sending Instructions

Value / TCK	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, 0x0001681B, as shown in Figure 3-11.
6. Move the state machine back to Run-Test-Idle;

Figure 3-9 Flow Chart of Reading ID Code State Machine

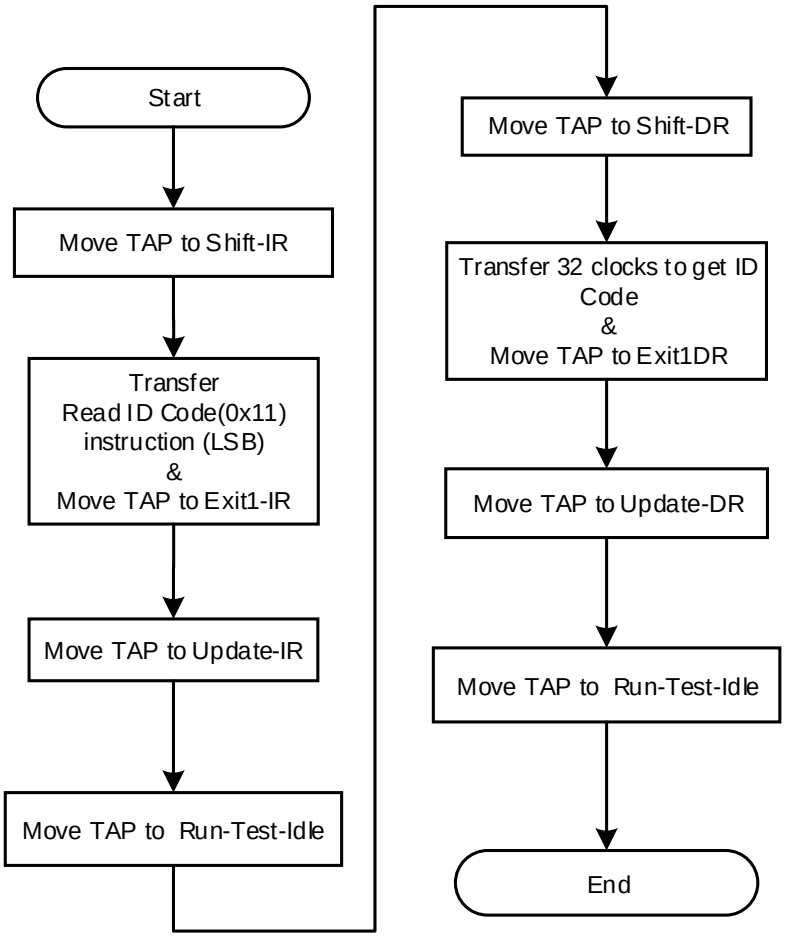


Figure 3-10 Access Timing of Reading ID Code Instruction- 0x11

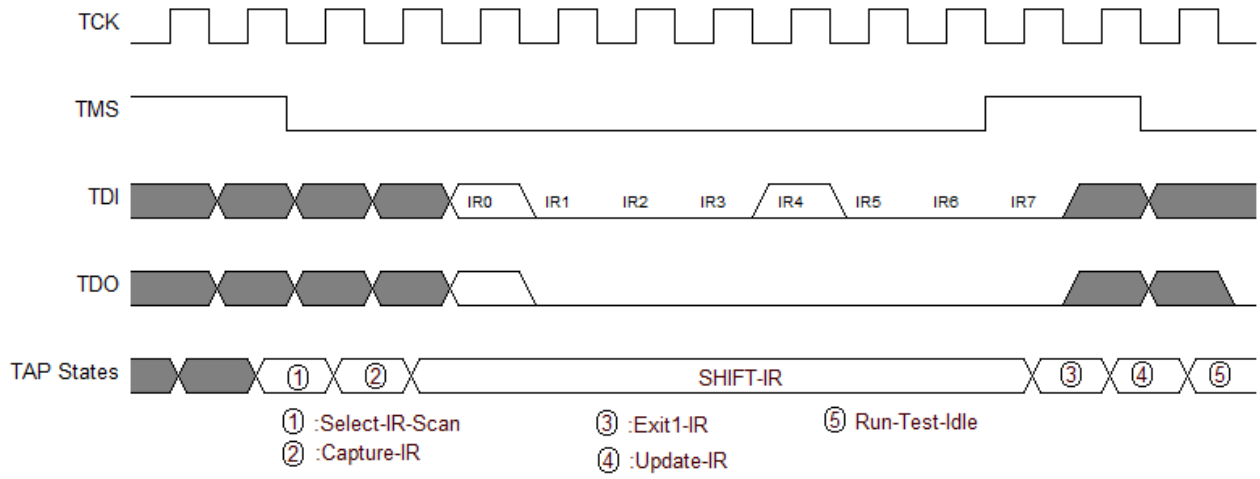
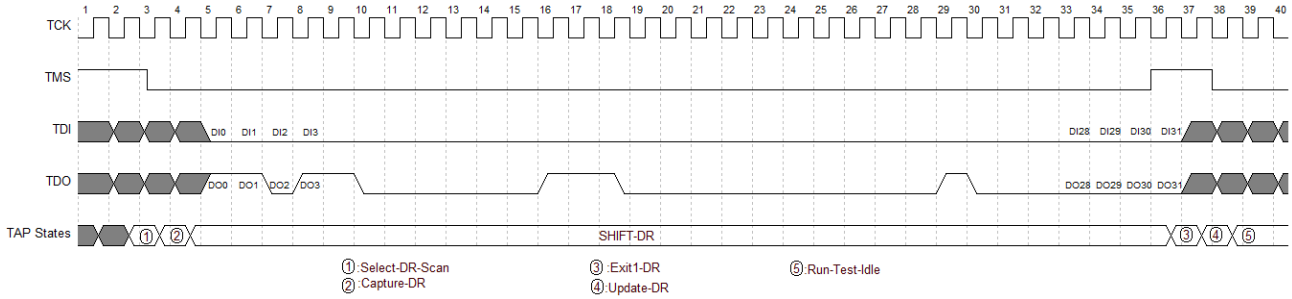


Figure 3-11 Access Timing of Reading ID Code Data Register



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.

For the meaning of the Status Register, please refer to 5.1 Status Register.

Read 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.

Reload 0x3C

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

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

Erase SRAM

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!

Enough time is required for the device to finish erasing after the instructions of EraseSram(0x05) and Noop(0x02) are sent.

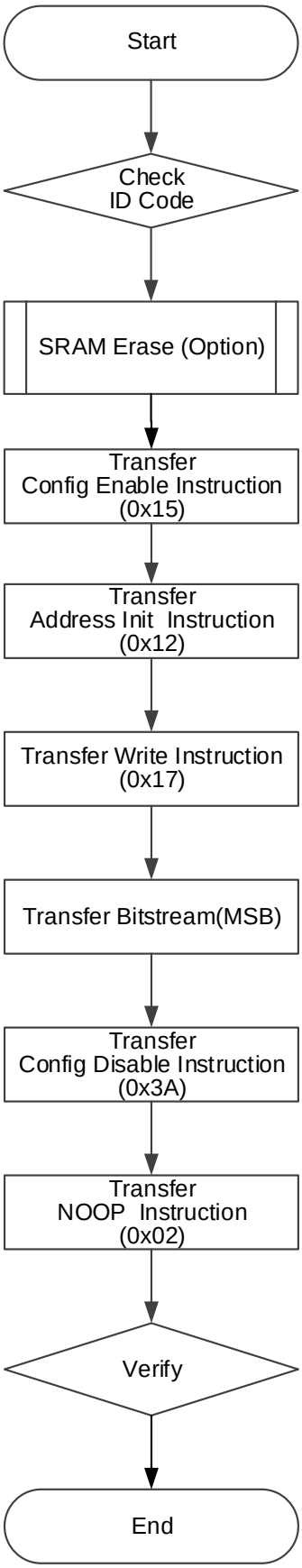
SRAM Configuration

The FPGA SRAM is configured using an external Host to enable the FPGA functions. SRAM is configured via JTAG to avoid the influence of Configuration Mode Pins.

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 3-12.

1. Establish a JTAG link and reset TAP;
2. Read the device ID CODE and check if it matches.
3. Erase the SRAM if it has been configured. Please refer to “Erase SRAM” for the detailed flow.
4. Send the "0x15" instruction of ConfigDisble;
5.
6. Send the "0x17" instruction of Transfer Configuration Data.
8. 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;
9. Send the "0x3A" instruction of ConfigDisabled;
7. Send the "0x02 " instruction of Noop to end the configuration process.

Figure 3-12 SRAM Configuration Flow



ExFlash Programming

Program External Flash via JTAG-SPI

In this mode, users can simulate Master SPI timing to program SPI Flash via the JTAG interface. TMS corresponds to CS signal, TCK to SCLK signal, TDI to SI signal, and TDO to SO signal. Please refer to the figure below for the flow of programming Flash in this mode.

Figure 3-13 Process View of Programming SPI Flash

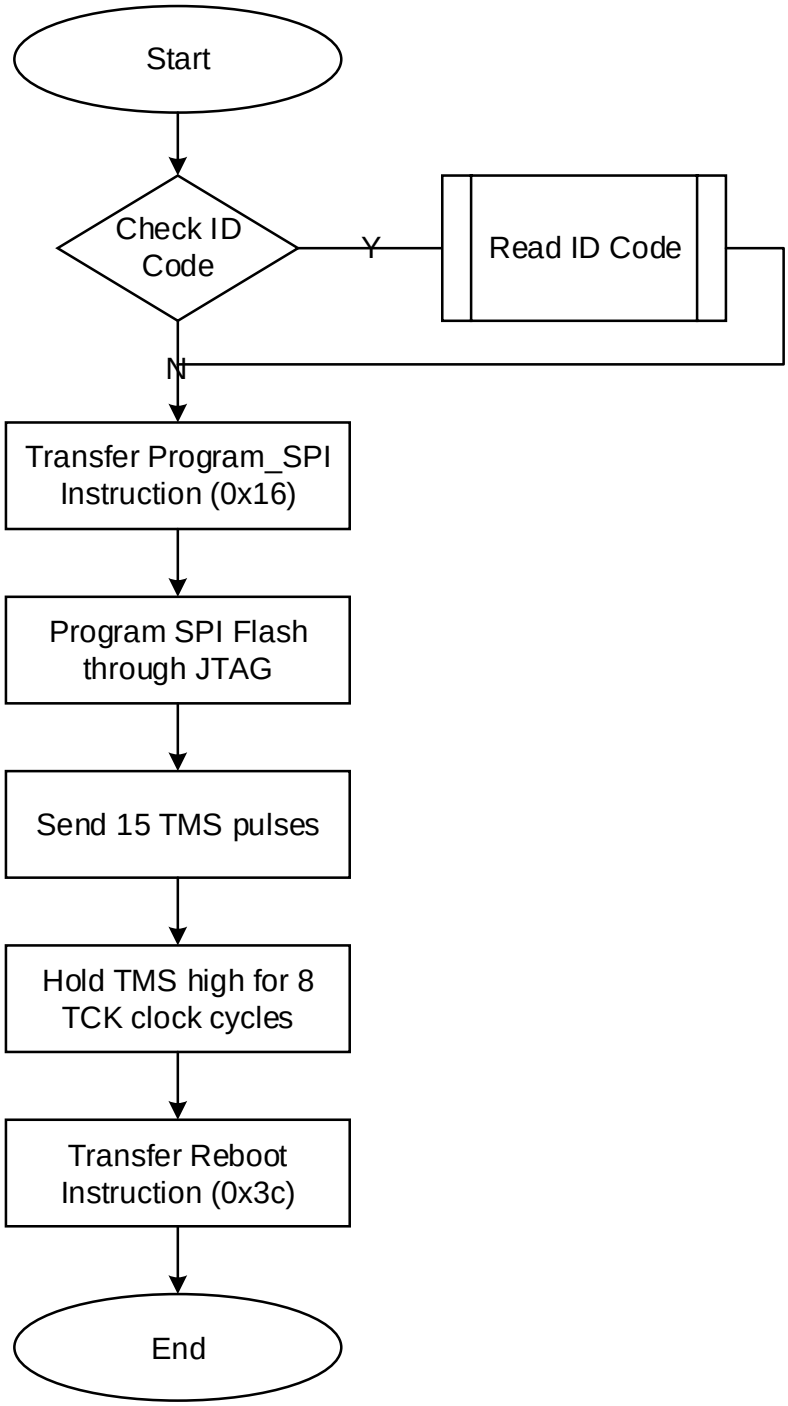
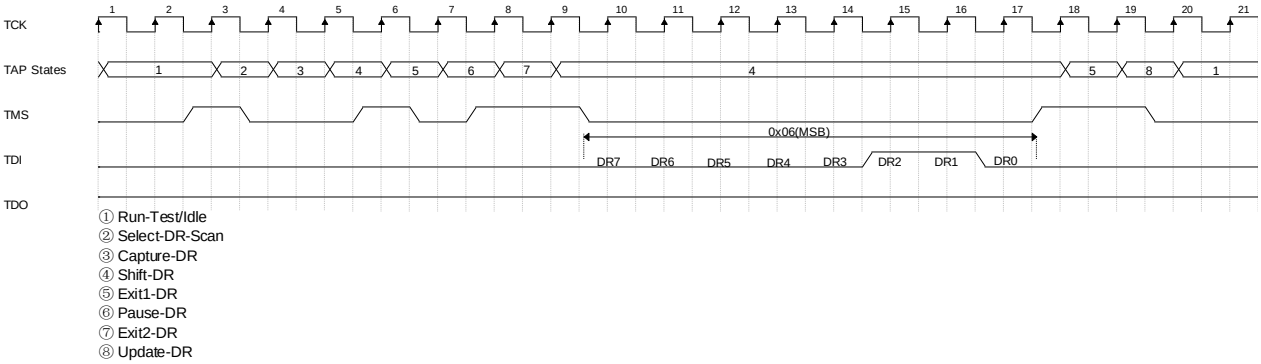


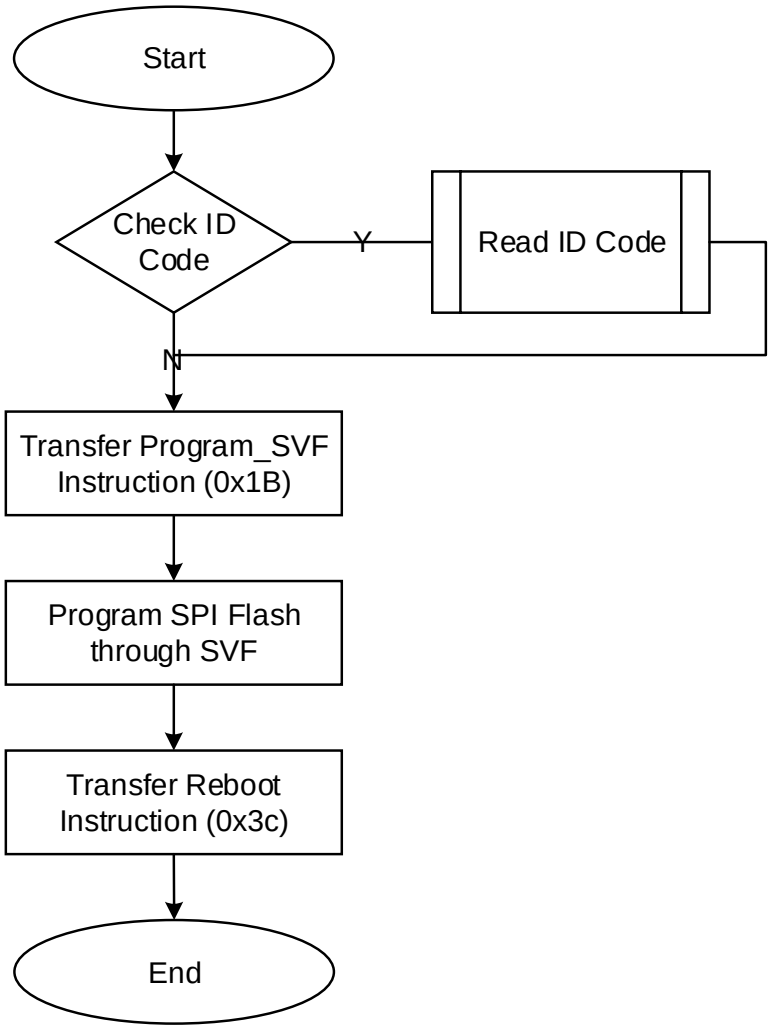
Figure 3-14 Timing diagram of JTAG- SPI Sending 0x06 (Arora Ⅴ)



SVF (Serial Vector Format) Programming External Flash

SVF is a syntax specification describing high-level IEEE 1149.1 (JTAG) bus operations. SVF describes JTAG chain operations in a compact and portable form. SVF files record JTAG operations by describing the information that needs to be moved into the device chain. The programming tool uses an SVF file as input and programs Flash using the information contained in the SVF file. Please refer to the figure below for the flow of programming Flash in this mode.

Figure 3-15 Process of Program ming Flash in SVF Mode



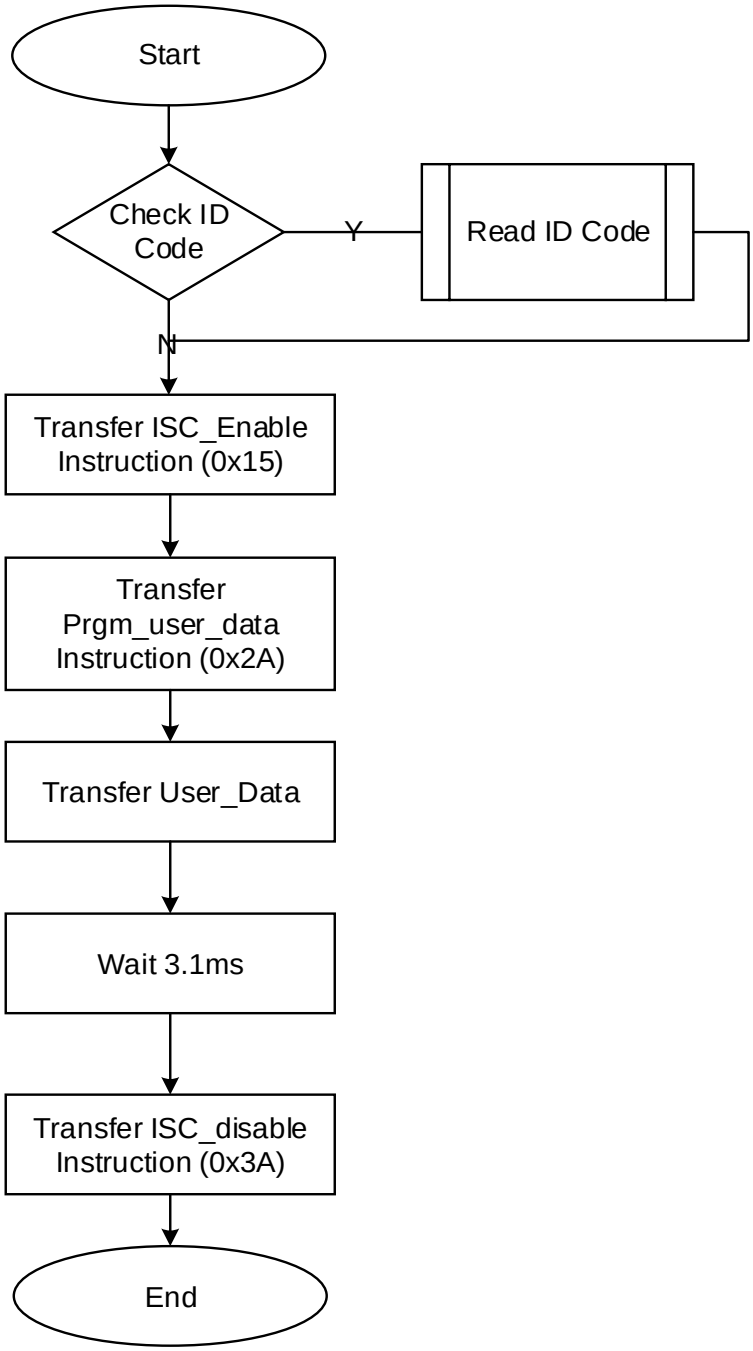
OTP Configuration

Arora V FPGA products supports one-time programming and provides 128 Bit OTP space. Bit0 ~ Bit31 is the user space, which can be used to store security and other important information. Bit32~Bit95 is the DNA space, which stores the 64-bit unique identification information of the device.

For OTP definition, please refer to 5.2 OTP Efuse.

The configuration flow is as shown in the figure below.

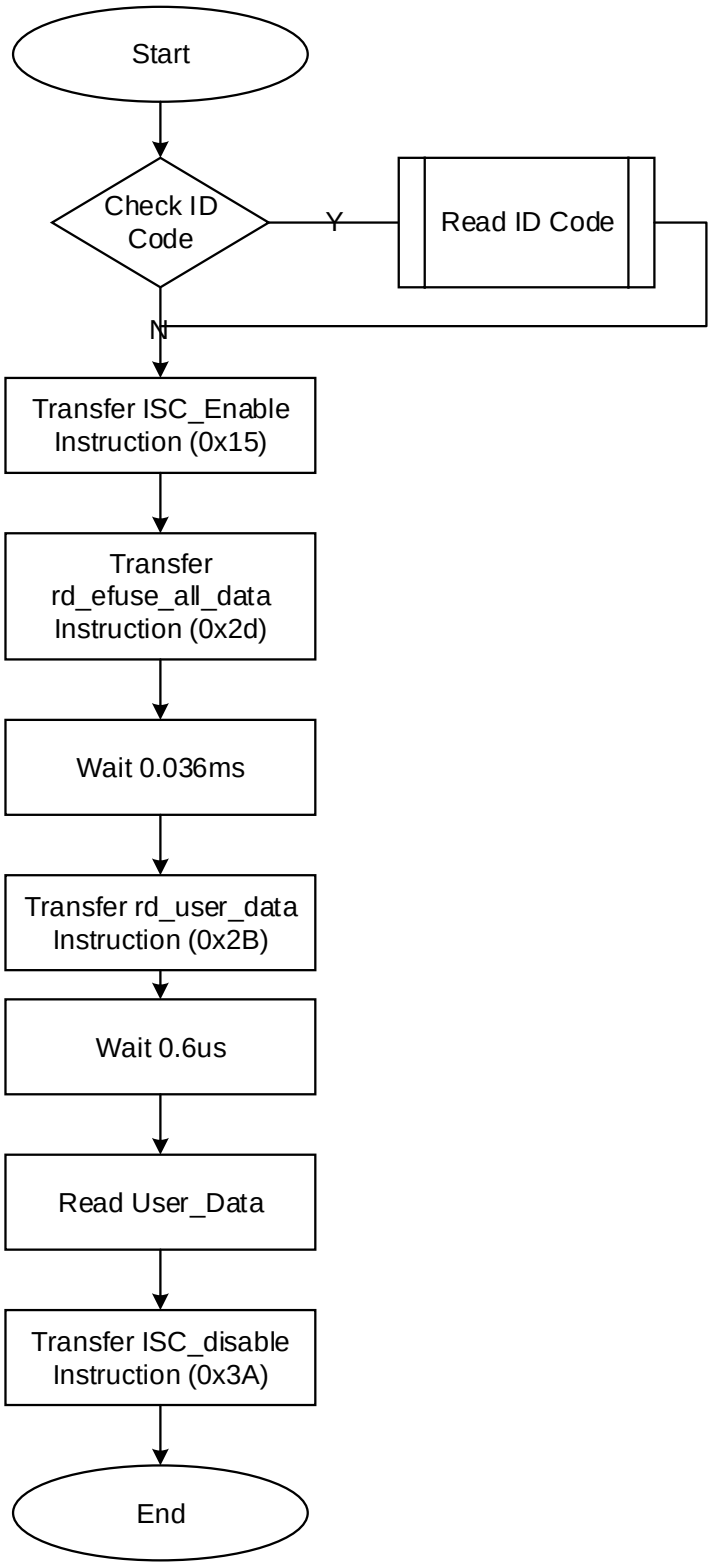
Figure 3-16 OTP Configuration Flow



Read OTP

The process of reading data is as shown in the figure below.

Figure 3-17 Process of reading OTP



Routine File

For the routine file, please contact GOWINSEMI technical support or

the local office.

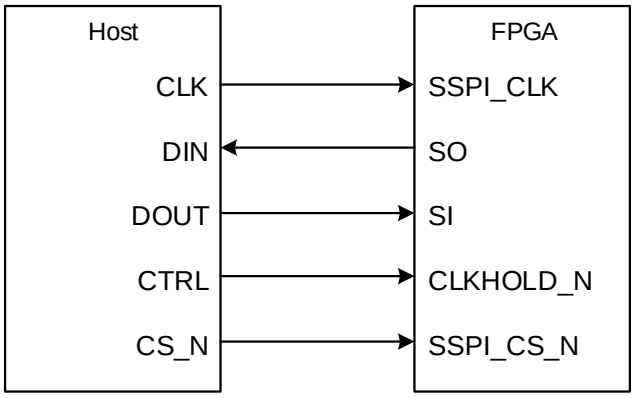
3.5 SSPI Configuration Mode

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

3.5.1 Connection Diagram for SSPI Configuration Mode

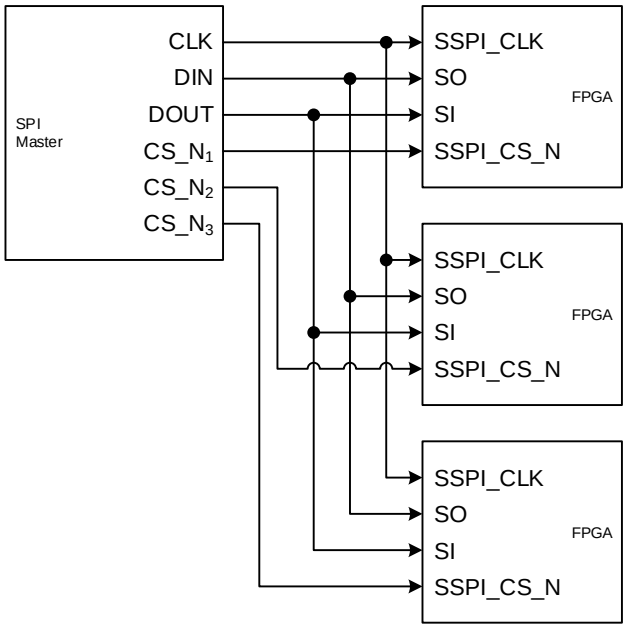
The connection diagram for configuring Gowin FPGA products via SSPI is shown in Figure 3-18.

Figure 3-18 SSPI Configuration Mode Connection Diagram



The connection diagram for configuring multiple FPGA products via SSPI is shown in Figure 3-19.

Figure 3-19 Multiple FPGA Connection Diagram

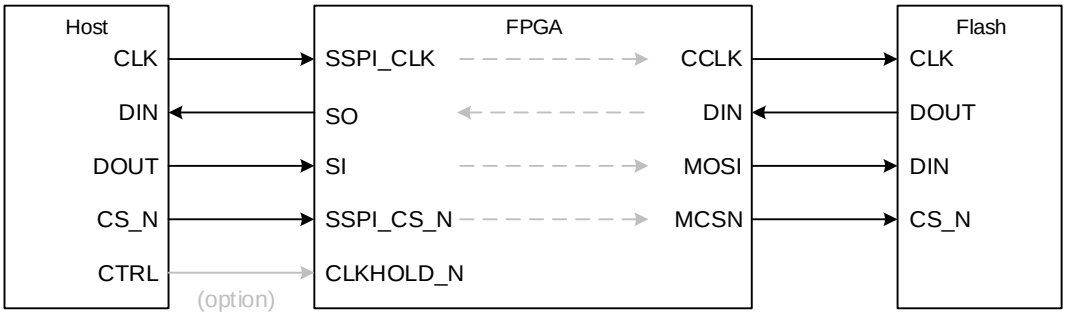


In addition to SRAM, SSPI can be used to program external SPI Flash. The MODE value of the Flash programming is the same as that of SSPI configuration mode. Configuration data can be written to SRAM or an external Flash using Gowin programmer. Before loading from the external Flash, the MODE value should be adjusted to MSPI MODE, and

then the MSPI loading can be triggered by powering on again or triggering RECONFIG_N.

The connection diagram for programming an external Flash via SSPI is shown in Figure 3-20.

Figure 3-20 Connection Diagram of Programming External Flash via SSPI

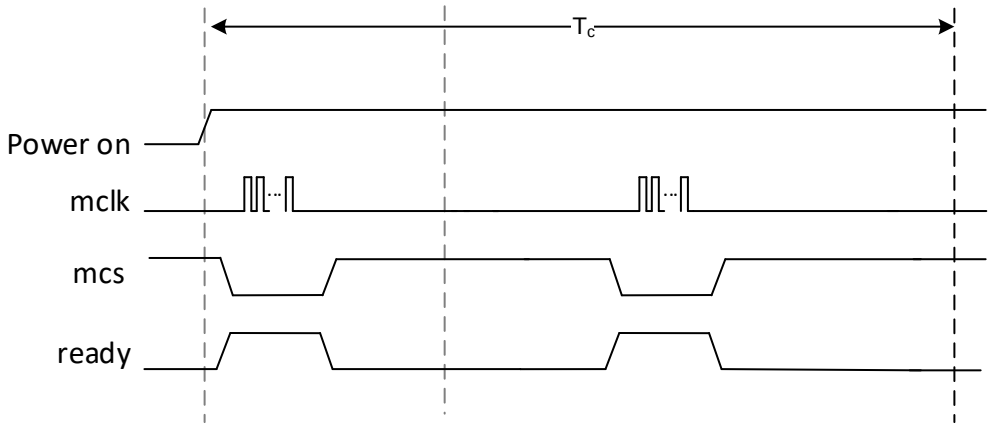


3.5.2 SSPI Configuration Timing

After power-up, when the MODE value selects MSPI configuration mode, the FPGA issues commands to load via MSPI. If the loading fails, it switches to SSPI mode.

It is recommended to use a 4.7k pull-up resistor to the corresponding Bank VCCIO on the SSPI_CLK and SSPI_CS_N pins.

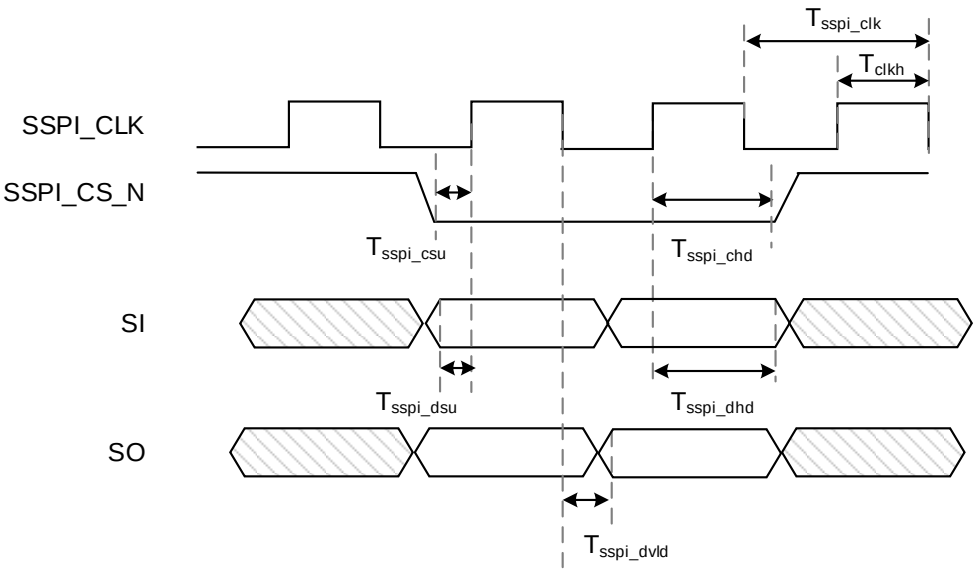
Figure 3-21 SSPI Configuration Process Timing



Name	Description	Typ.	Unit
Tc	SSPI mode process time	20	ms

See Figure 3-22 for the SSPI timing.

Figure 3-22 SSPI Configuration Timing



See Table 3-9 for the SSPI configuration timing parameters.

Table 3-9 SSPI Configuration Timing Parameters

Name	Description	Min.	Max.	Unit
T_{sspi_clk}	SSPI port clock period	10.0	–	ns
T_{sspi_csu}	sspi_enablen setup time	1.0	–	ns
T_{sspi_chd}	sspi_enablen hold time	0	–	ns
T_{sspi_dsu}	SSPI input data setup time	1.0	–	ns
T_{sspi_dhd}	SSPI input data hold time	0	–	ns
T_{sspi_dvld}	delay of SSPI clock to output data	–	1.0	ns
T_{clkh}	The time of clock high level	(clock cycle) *45%	(clock cycle) *55%	–

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

- Initiate a new configuration
  Power up again or trigger the RECONFIG_N pin at one low pulse.

3.5.3 SSPI Configuration Instructions

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 3-10 Configuration Instruction

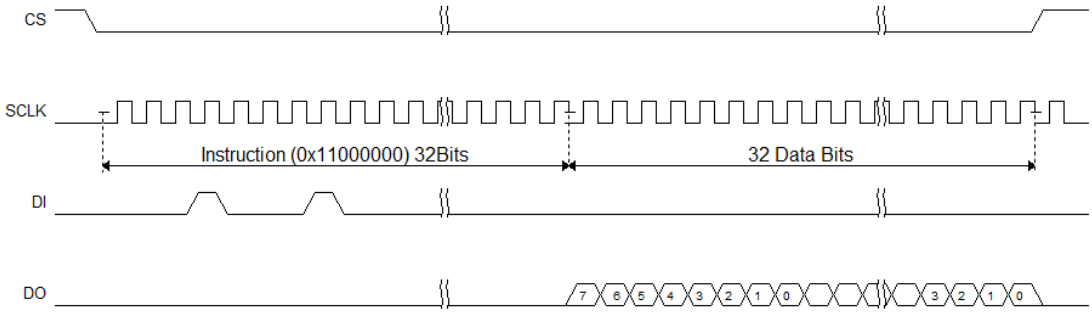
Name	Complete Instruction (Instruction Byte + Redundant Information Byte)
Read ID Code	0x11000000
Read User Code	0x13000000
Read Status Code	0x41000000
Reconfig/Reprogram	0x3C00
Write Enable	0x1500
Write Disable	0x3A00
Write Data	0x3B
Program SPI Flash	0x1600
Init Address	0x1200
Erase SRAM	0x0500
Reinit	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 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 3-23 Read ID Code Timing

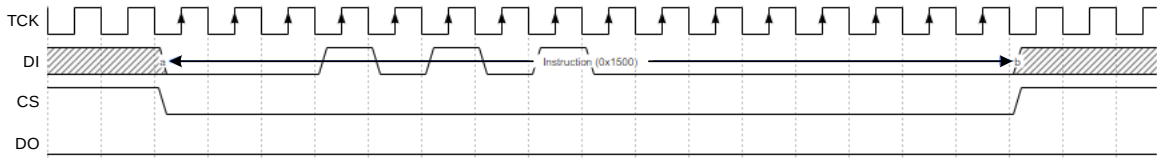


Operations of reading StatusCode / UserCode are similar with the operation of reading ID Code, simply replacing the related instructions.

Write Enable (0x1500)

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

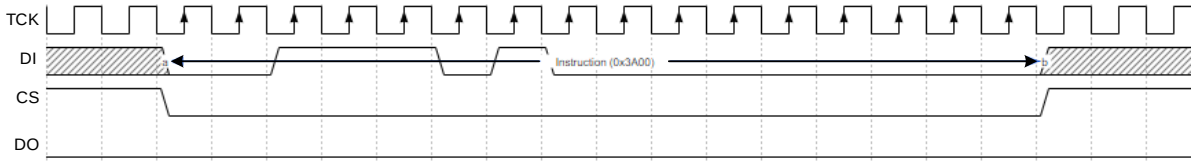
Figure 3-24 Write Enable (0x15) Timing



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 3-25 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).

Erase SRAM(0x0500)

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

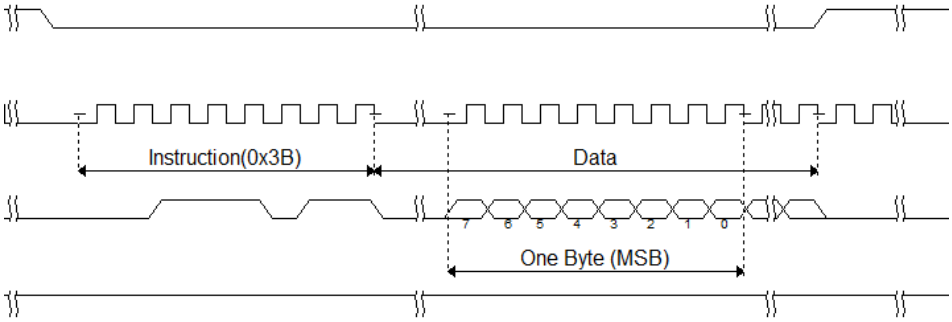
When the instruction is sent, at least 10ms is required to execute it.

Write Data (0x3B)

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

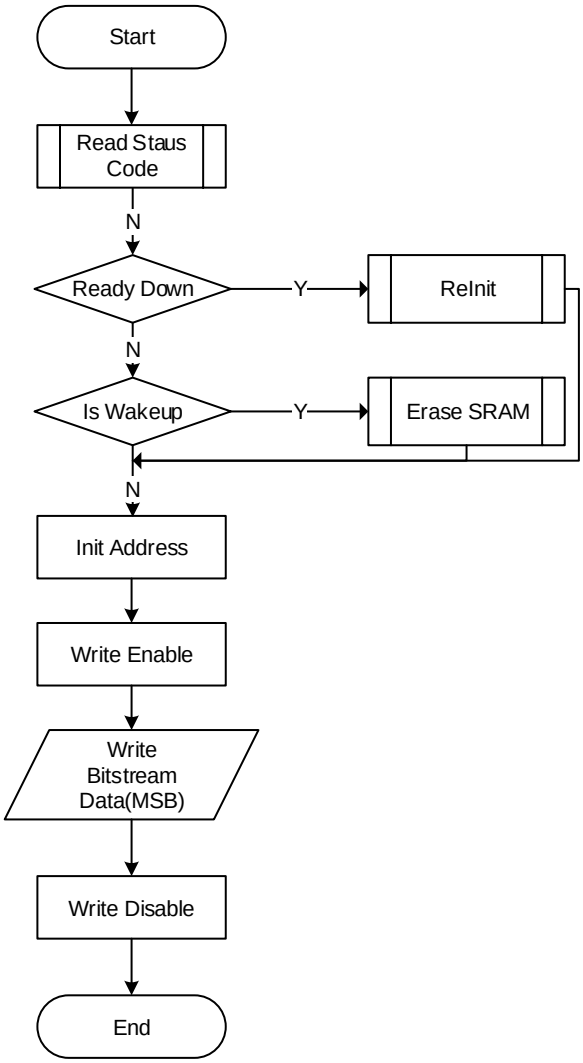
Note that CS keeps low level in the process of data writing.

Figure 3-26 Write Data (0x3B) Timing



3.5.4 SRAM Configuration via SSPI

Figure 3-27 The Flow Chart of SRAM Configuration via SSPI



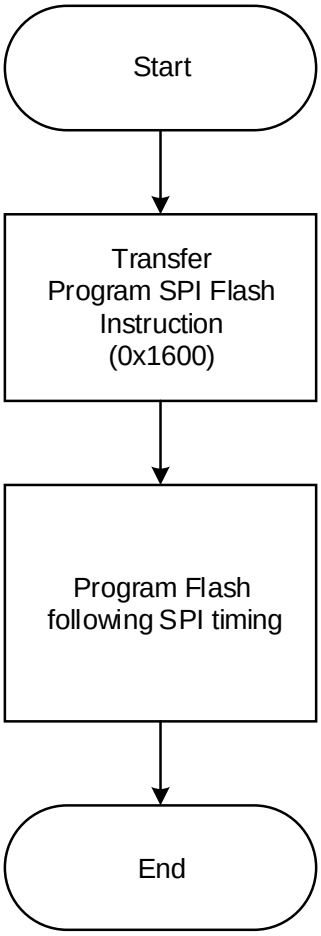
3.5.5 Flash Programming via SSPI

Figure 3-28 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.

It is recommended to use a 4.7k pull-up resistor to the corresponding Bank VCCIO on the SSPI_CLK and SSPI_CS_N pins.

Note that when reading data from Flash, the data being read back is delayed by one Bit. For example, when SSPI reads Flash's IDCode, it needs to send an extra Clock to get the last bit.

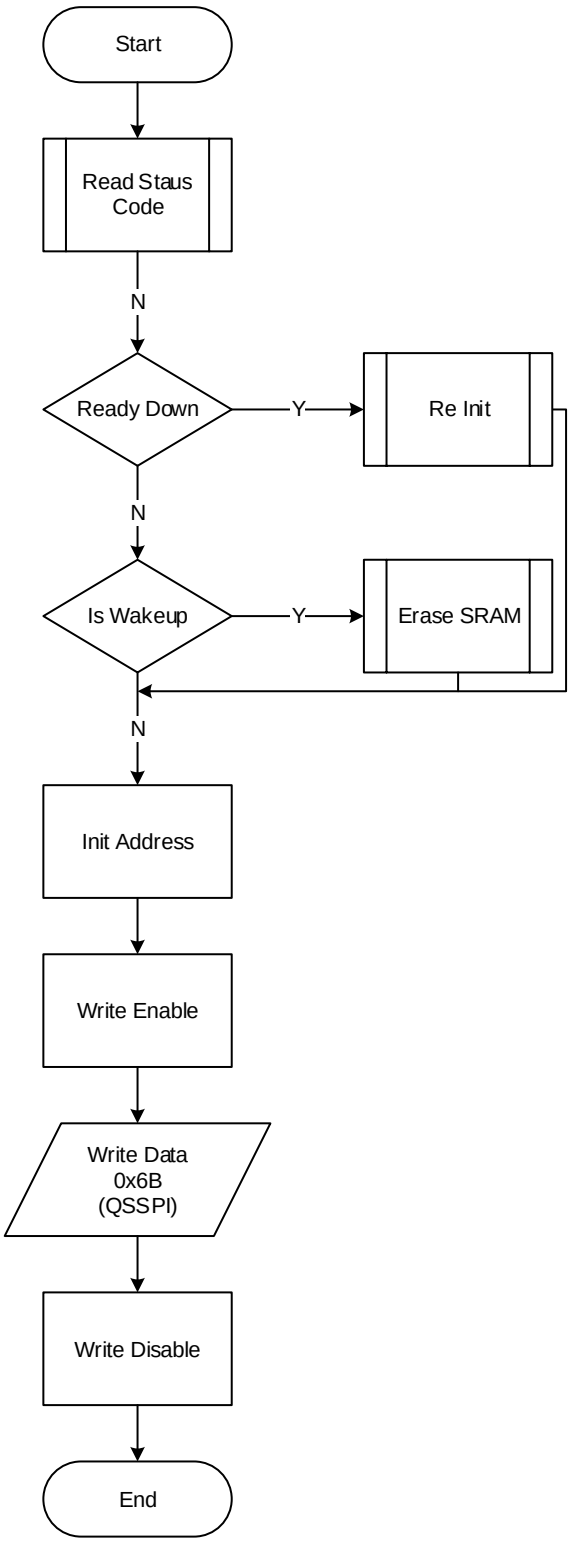
Figure 3-28 The Flow Chart of Programming Flash via SSPI



3.5.6 SRAM Configuration via SSPIx4

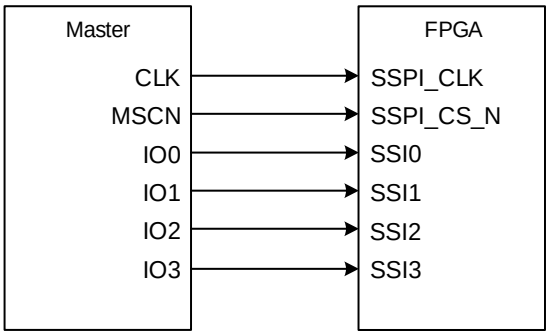
The SRAM configuration process via SSPI mode is as follows. The Read Status, Reinit, EraseSram, InitAddress, WriteEnable, and WriteDisble commands are the same with those of SSPI. Only Write Data uses the QSSPI command.

Figure 3-29 The Flow Chart of SRAM Configuration via QSSPI



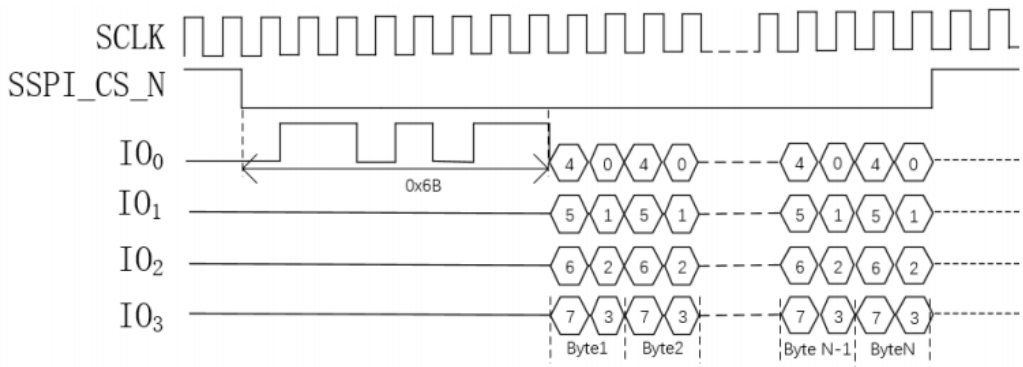
The connection diagram for configuring Gowin FPGA products via QSSPI is shown in Figure 3-30.

Figure 3-30 QSSPI Configuration Mode Connection Diagram



The timing diagram of QSSPI Write Data (0x6B) is as shown in the figure below.

Figure 3-31 QSSPI Write Data (0x6B) Timing



3.6 MSPI Configuration Mode

In MSPI (Master SPI) mode, FPGA is a Master and reads bitstream data from the external Flash via SPI port to complete configuration.

MSPI Configuration Process: Set the MODE pin to MSPI status, power on again or trigger RECONFIG_N at one low-level pulse, and the device will read bitstream data from the external Flash and complete configuration automatically.

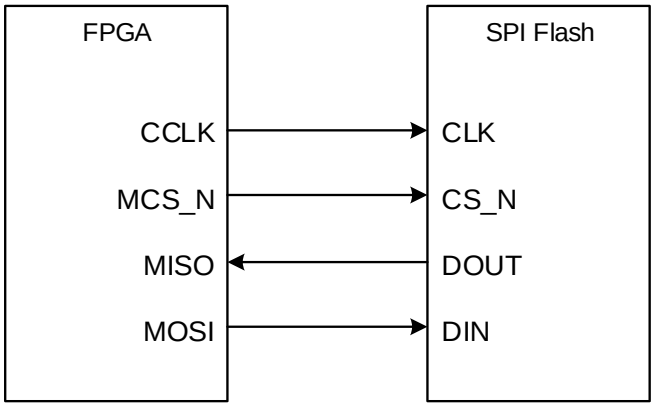
According to the MSPI configuration features, remote upgrade requirements can be implemented. After starting the FPGA, if an upgrade is required, users can remotely write the configuration data into the external Flash, and trigger RECONFIG_N or power up again to upgrade the system if the upgrade conditions are met.

3.6.1 Connection Diagram for MSPI Configuration Mode

The connection diagram for configuring Gowin FPGA products

through MSPI is shown in Figure 3-32~Figure 3-34.

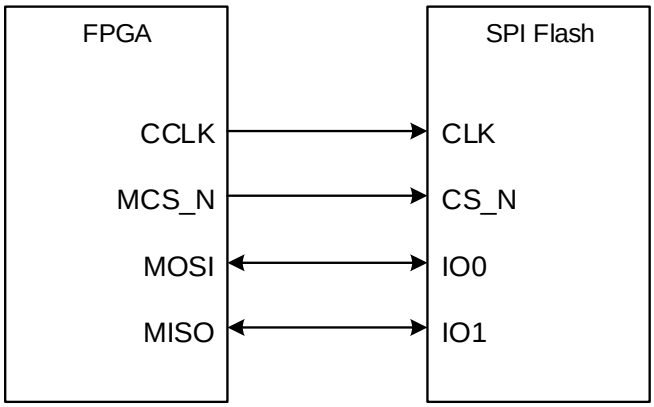
Figure 3-32 Connection Diagram for MSPIx1 Configuration Mode



Note!

- MSPI x1 frequency range: 2.5Mhz~105Mhz
- The default configuration is MSPI x1, 3Byte addressing mode, which can be configured in the EDA software to improve the loading time.

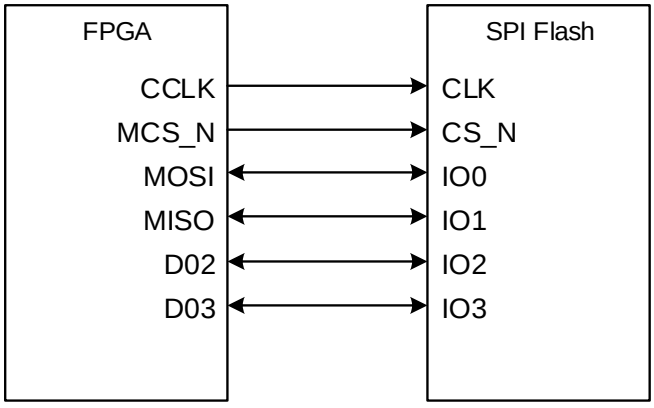
Figure 3-33 Connection Diagram for MSPIx2 Configuration Mode



Note!

- MSPI x2 frequency range: 2.5Mhz~105Mhz
- The default configuration is MSPI x1, 3Byte addressing mode, which can be configured in the EDA software to improve the loading time.

Figure 3-34 Connection Diagram for MSPIx4 Configuration Mode



Note!

- MSPI x2 frequency range: 2.5Mhz~105Mhz

- The default configuration is MSPI x1, 3Byte addressing mode, which can be changed in the EDA software to improve the loading time.

3.6.2 MSPI Configuration Timing

See Figure 3-35 for MSPI timing.

Figure 3-35 MSPI Configuration Timing

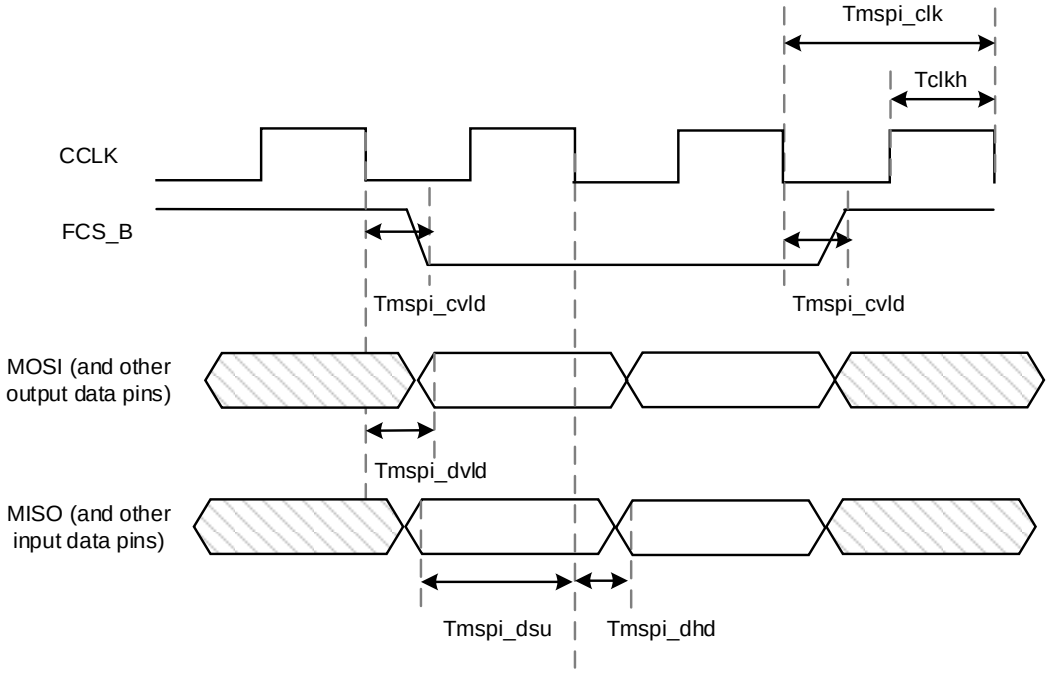


Table 3-11 shows the timing parameters.

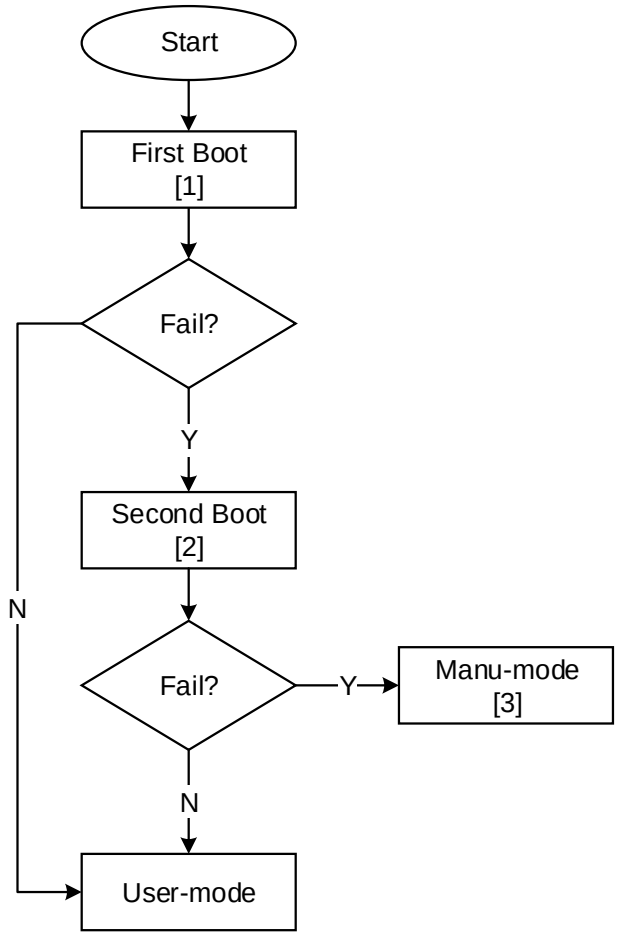
Table 3-11 MSPI Configuration Timing Parameters

Name	Description	Min.	Max.	Unit
Tmspi_clk	MSPI clock period	10.0	–	ns
Tmspi_cvld	delay of MSPI clock falling edge to CSI_B	–	1.0	ns
Tmspi_dvld	delay of MSPI clock falling edge to output data	–	1.0	ns
Tmspi_dsu	the setup time of input data	2.0	–	ns
Tmspi_dhd	the hold time of input data	0	–	ns
Tclkh	The time of clock high level	(clock cycle) *45%	(clock cycle) *55%	–

3.6.3 MSPI

Arora V FPGA products start from the 0x000000 address at the default frequency of 2.5MHz after power on, and read the bit stream data with SPI x1 protocol to complete the configuration. If the first configuration fails, the device automatically performs a second configuration operation from the default address 0x800000. The factors that can lead to a failed configuration include false ID validation error, CRC check error, instruction error, and timeout error. If both configurations fail, the device goes into Manu mode.

Figure 3-36 The Flow Chart of MSPI Mode



Note!

- [1] The default configuration of the first loading is MSPI x1, 3Byte addressing Read mode, which can be configured in the EDA software to improve the loading time.
- [2] The loading address can be set as required in the EDA tool.
- [3] In manu-mode, SSPI and JTAG interfaces can be used for debugging or manually configuration.

In addition, in MSPI mode, jumping to any loading address can be realized by adding the “mspi_jump(0x6E)” instruction at the beginning of the bitstream file. It is convenient to allocate FLASH storage space flexibly in complex system.

3.7 CPU Configuration Mode

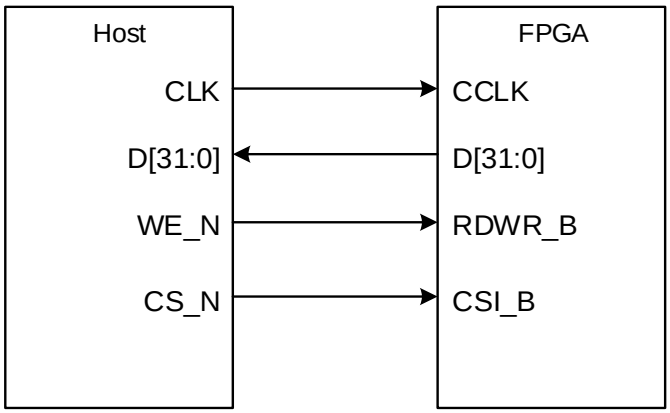
The CPU configuration interface supports 8/16 bit width, which can adjust the bus width adaptively and supports 8-bit width data readback. Slave mode is supported. The only difference between the master mode and the slave mode is the direction of the interface clock. The master mode selects the internal crystal as the configuration clock source by default, and EMCCLK can be selected as the configuration clock source through the EDA tool. The maximum clock speed of the interface is listed as follows:

- 16-bit width: The max. write clock frequency is 80MHz. The max. read clock frequency is 40MHz.
- 8-bit width: The max. write clock frequency is 80MHz. The max. read clock frequency is 40MHz.

3.7.1 Connection Diagram for CPU Configuration Mode

The connection diagram for the CPU mode is shown in Figure 3-37.

Figure 3-37 Connection Diagram for CPU Mode



Note!

CCLK is output in master mode and input in slave mode.

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

- Initiate new configuration
  Power-on again or trigger RECONFIG_N at one low pulse.

3.7.2 Configuration Timing

CPU Timing is as shown in Figure 3-38.

Figure 3-38 CPU Mode Configuration Timing

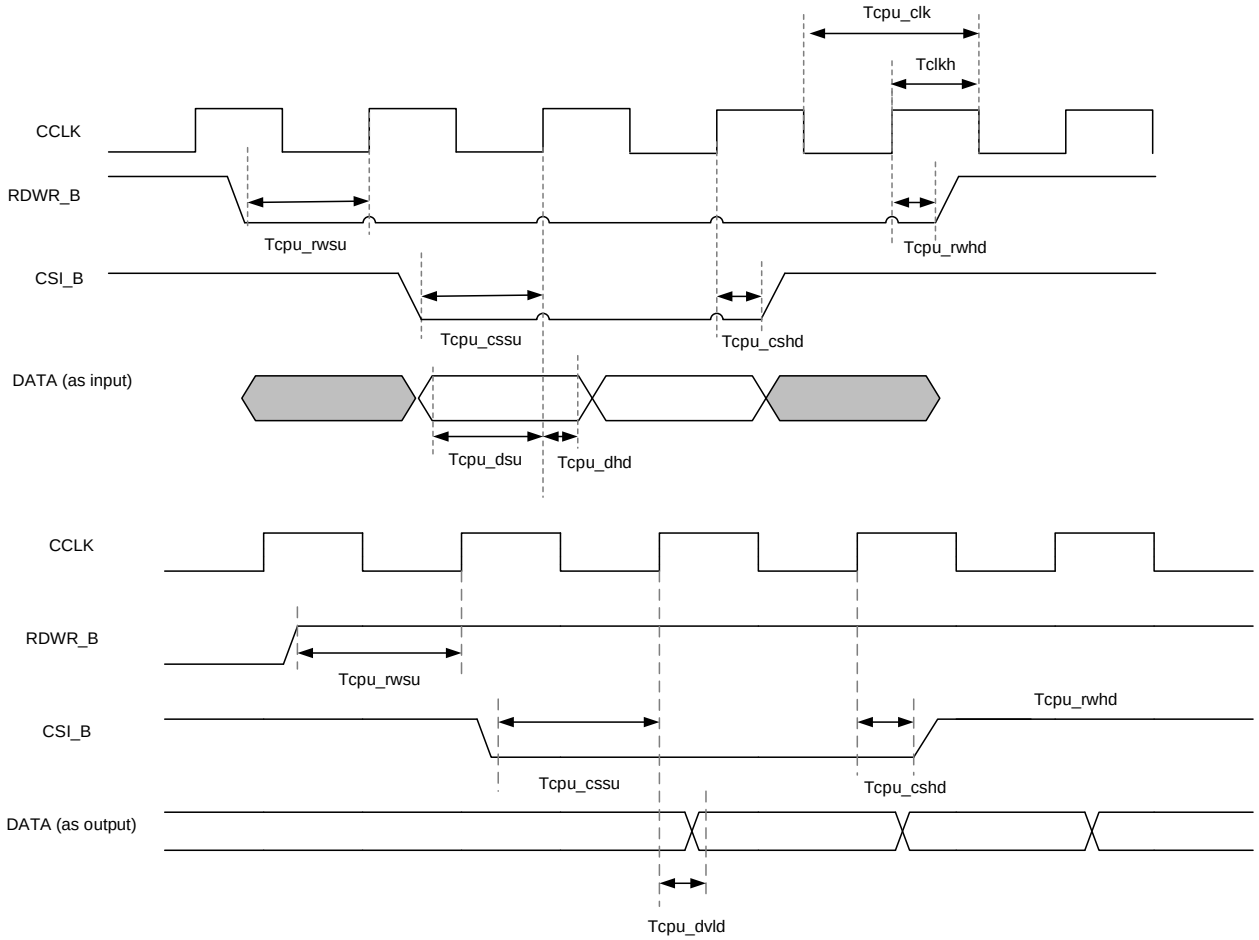


Table 3-12 shows the timing parameters.

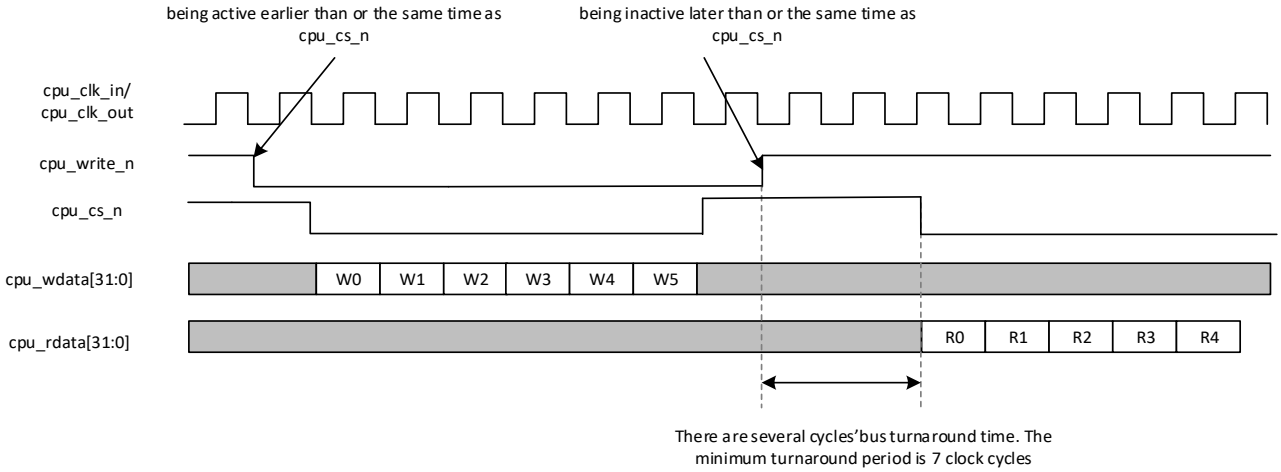
Table 3-12 CPU Configuration Timing Parameters

Name	Description	Min.	Max.	Unit
Tcpu_clk	CPU port clock period	10.0	–	ns
Tcpu_rwsu	RDWR_B setup time	8.0	–	ns
Tcpu_rwhd	RDWR_B hold time	0	–	ns
Tcpu_cssu	CS_B setup time	8.0	–	ns
Tcpu_cshd	CS_B hold time	0	–	ns
Tcpu_dsu	input data setup time	4.0	–	ns
Tcpu_dhd	input data hold time	0.0	–	ns
Tcpu_dvld	delay of CPU clock to output data	–	8.0	ns
Tclkh	The time of clock high level	(clock cycle) *45%	(clock cycle) *55%	–

3.7.3 Continuous Data Loading

Continuous data loading is used by the Host to provide continuous data stream for configuration. After power up, the controller sets the RDWR_B signal to write control (RDWR_B = 0) and sets the CSI_B signal to zero (CSI_B = 0). Note that RDWR_B must be driven to low before CSI_B, otherwise an abort occurs.

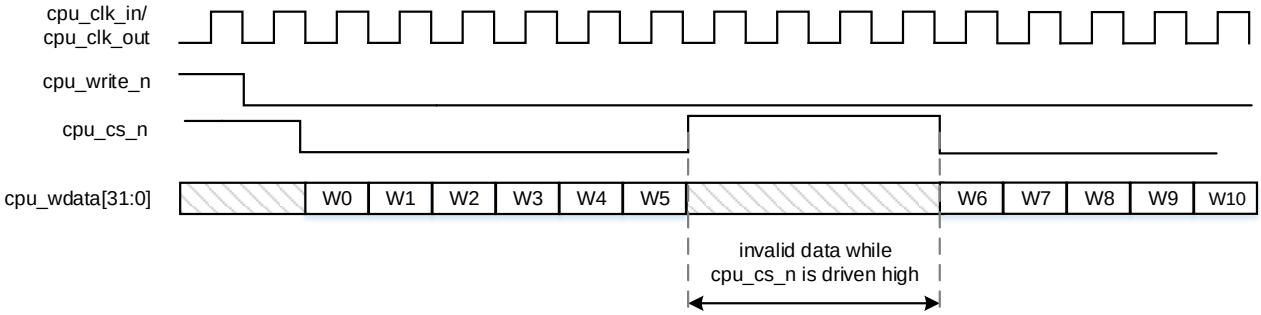
Figure 3-39 Continuous Data Loading Timing



3.7.4 Non-continuous Data Loading

Non-continuous loading is used for that the Host cannot provide continuous data stream for configuration. The Host has two ways to stop the configuration: Pull high the CSI_B signal or pause CCLK.

Figure 3-40 Non Continuous Data Loading Timing



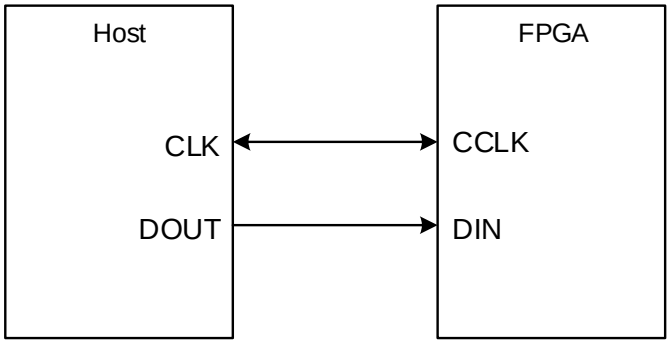
3.8 SERIAL Configuration Mode

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. It supports slave mode. The only difference between the two modes is the different direction of the interface clock. 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.

3.8.1 Connection Diagram for SERIAL Configuration Mode

The connection diagram for the SERIAL mode is shown in Figure 3-41.

Figure 3-41 Connection Diagram for SERIAL Mode



Note!

[1] CCLK is output in the master mode and input in the slave mode.

SERIAL Configuration Timing

See Figure 3-42 for the timing of SERIAL mode.

Figure 3-42 SERIAL Configuration Timing

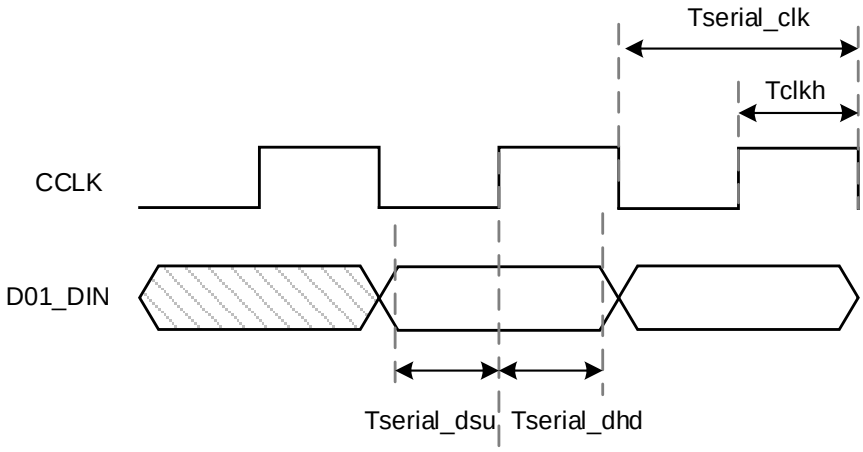


Table 3-13 shows the timing parameters.

Table 3-13 SERIAL Configuration Timing Parameters

Name	Description	Min.	Max.	Unit
Tserial_clk	Serial port clock period	10.0	–	ns
Tserial_dsr	the setup time of input date	4.0	–	ns
Tserial_dhd	the hold time of input data	0	–	ns
Tclkh	The time of clock high level	(clock cycle) *45%	(clock cycle) *55%	–

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

- Initiate new configuration
  Power-on again or trigger RECONFIG_N at one low pulse.

3.9 Daisy Chain

3.9.1 Serial Daisy Chain

In a serial daisy chain, a device receives its configuration data through a DIN pin and transmits the configuration data to the downstream device through a DOUT pin. The most upstream device (the device closest to the data source, i.e., Device_1 in the figure below) can be in one of the following modes:

- Master Serial
- Slave Serial
- Master SPI

The downstream device must be in the mode:

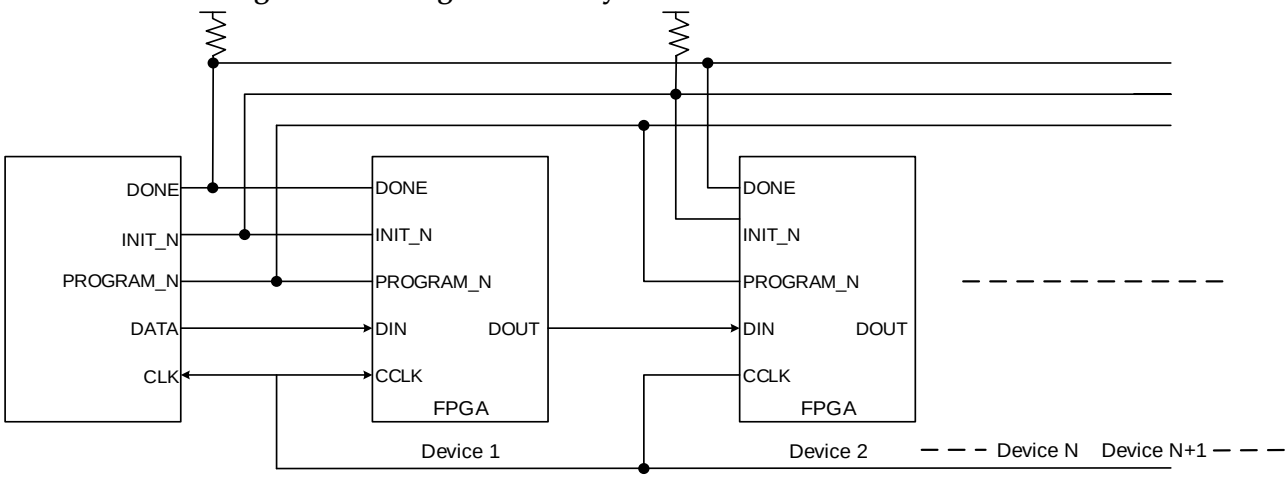
- Slave Serial

Note!

Master SPI must be in the x1 mode.

Below is a diagram of a serial daisy chain, where the Host first configures Device 1 and then bypasses the bitstream to its adjacent Device (Device 2 in the figure) via the DOUT pin.

Figure 3 - 43 Diagram of Daisy-Chain



3.9.2 Parallel Daisy Chain

In the parallel Daisy chain, the "chip selection" signal is transmitted from the upstream device to the downstream device. The most upstream device can be in one of the following modes:

- Slave CPU

  The downstream device must be in the mode:

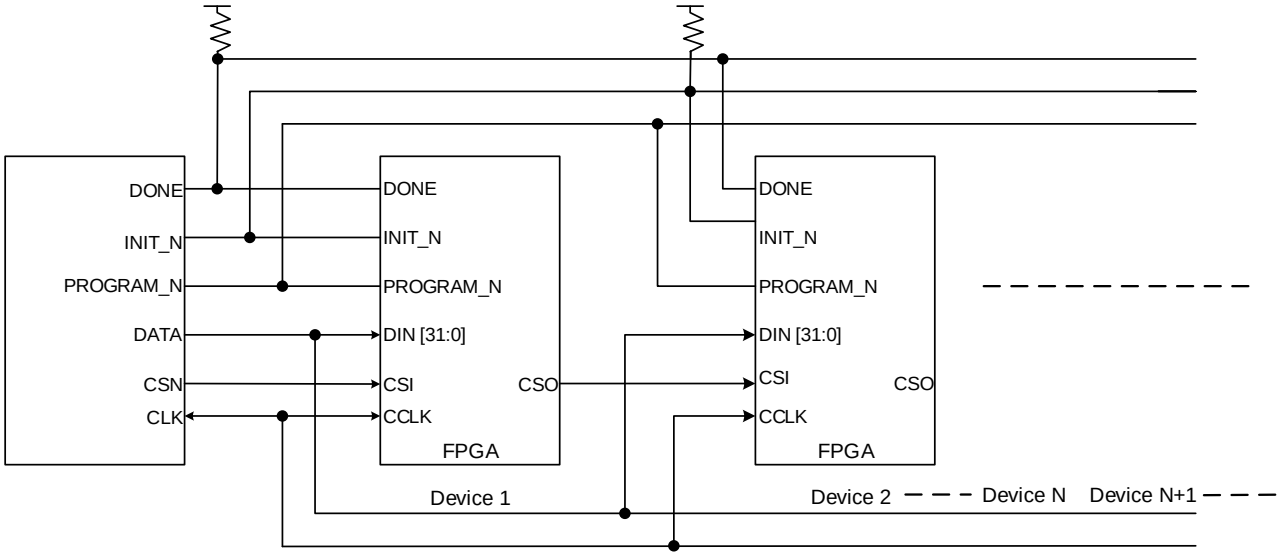
- Slave CPU

  Note!

  The write enable pin (RDWR N) must be pulled low for all devices.

Below is a diagram of a parallel daisy chain, where the Host first configures Device 1 and then bypasses “CSN” to Device 2 via the “CSO” pin. Once the device configuration is done, all downstream devices transmit the "CSN" to the next device adjacent to it.

Figure 3 - 44 Diagram of Parallel Daisy-Chain



4 Configuration Details

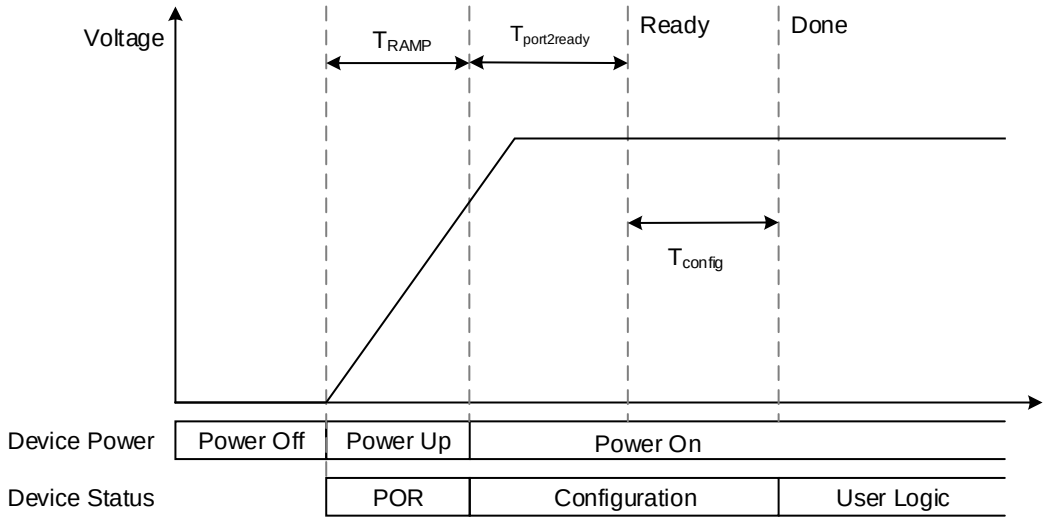
GOWINSEMI Arora V FPGA products are high performance devices based on SRAM technology. The device based on SRAM technology loses the configuration data after power off and needs to be reconfigured after each power on.

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 mode is different.

4.1 Configuration Notes

Power On Requirement

Figure 4-1 Process of Power on



When the power up voltage of VCC, VCCO, 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.

Power supply voltage needs to be stable in the process of FPGA start-up. RECONFIG_N needs to keep high after being powered up until the voltage is stable for 1ms and also in the process of FPGA initialization. RECONFIG_N can be vacant or external pulled up. All GPIOs output high resistance state before FPGA is waken up.

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 are insufficient, these pins can be configured and used as GPIOs, but pins associated with data transmission need to be kept. MODE [2:0] is used to select programming configuration MODE. MODE can be fixed through the pull-up or pull-down resister. It is recommended to use 4.7K pull-up resister and 1K 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

For the recommended pin connection, please refer to the related schematic manuals.

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

Figure 4-2 and Figure 4-3 Trigger Timing show the timing for power-on again or triggering RECONFIG_N at low pulse.

Figure 4-2 Power Recycle Timing

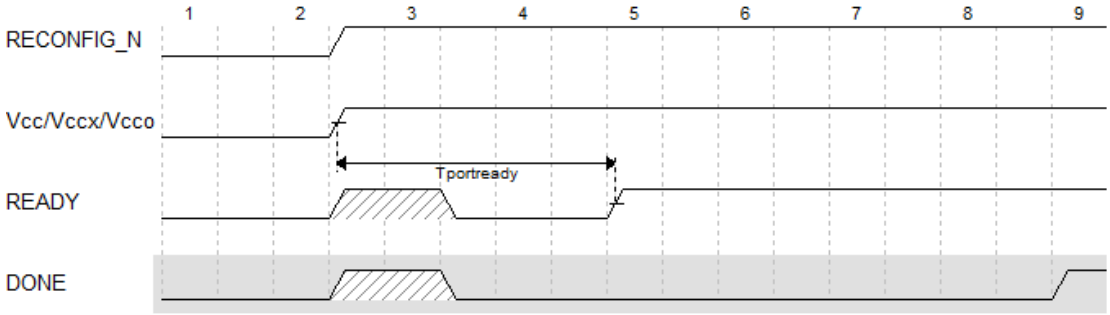


Figure 4-3 Trigger Timing

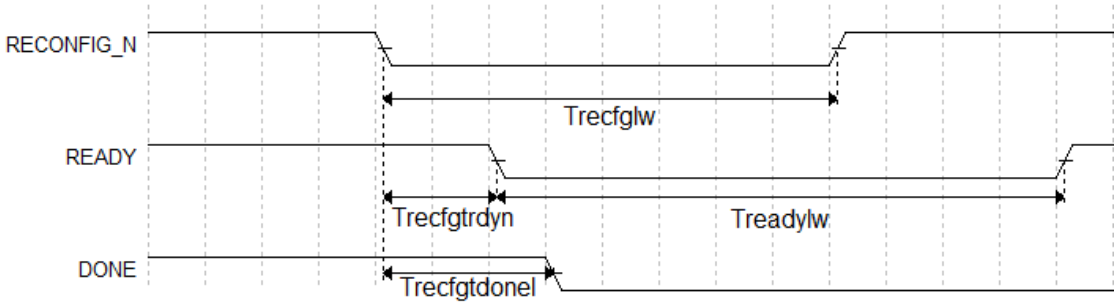


Table 4-1 shows the timing parameters of Arora V FPGA products.

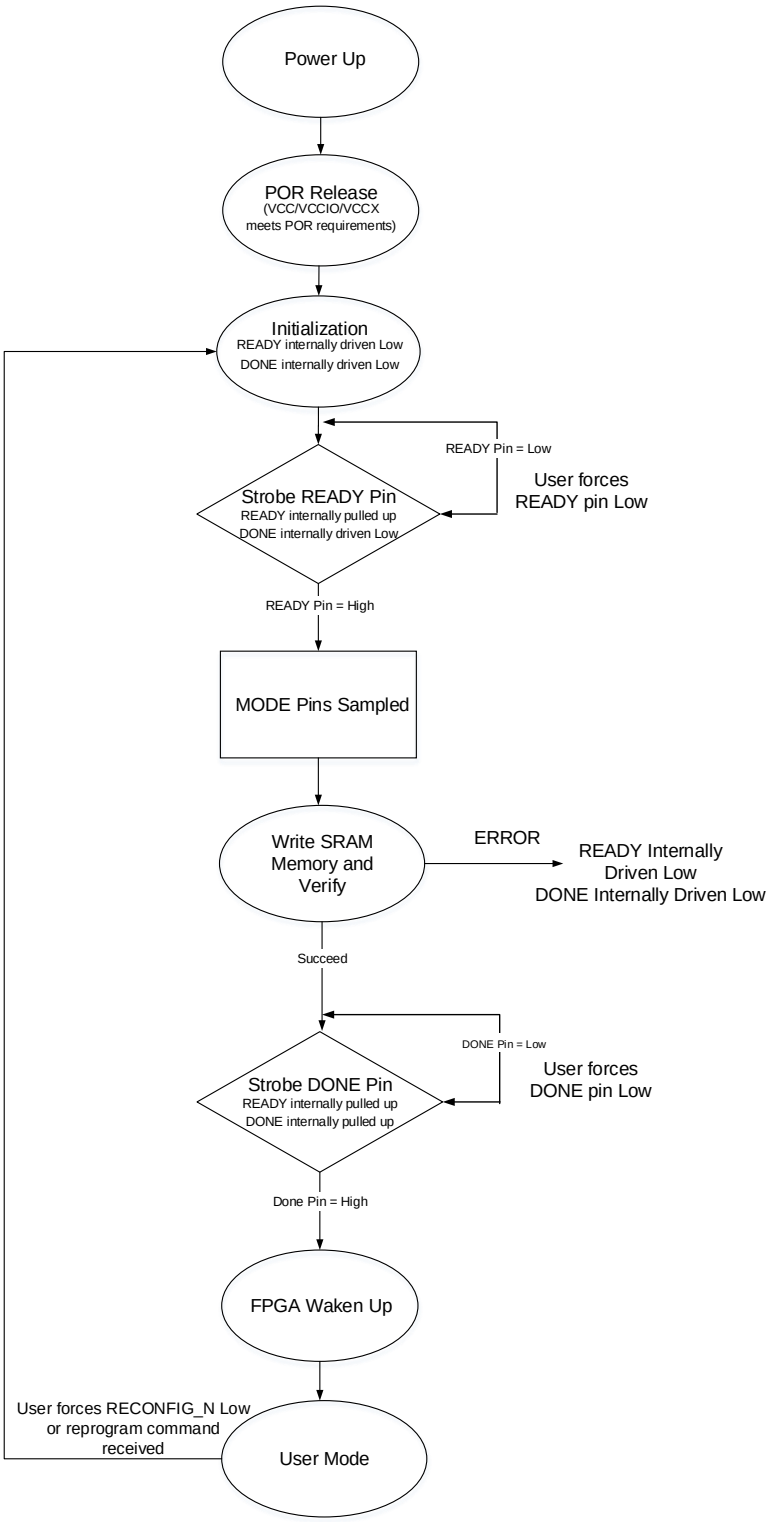
Table 4-1 Timing Parameters of Power on Again and RECONFIG_N triggering for Arora V FPGA Products

Name	Description	Min.	Tpy.	Max.
$T_{portready}$	Time from POR to the rising edge of READY	-	1.024ms	23ms
$T_{recfglw}$	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

4.2 Configuration Sequence

GOWINSEMI FPGA products will go through initialization, SRAM configuration after power-up. Figure 4-4 shows the configuration process.

Figure 4-4 GOWINSEMI FPGA Configuration Process



Note!

- READY, DONE, and RECONFIG_N are bidirectional IOs with open drain output

and internal weak pull-up (the pull-up current is about 100uA).

- You can control the timing of the device starting to load by forcing the READY pin low.
- You can control the timing of the device waking up by forcing the DONE pin low.
- The RECONFIG_N pin needs to be held high from power-up to full loading of the device.

4.2.1 Power-Up Timing

During power-up, the Power On Reset (POR) circuit starts to work. The POR circuit ensures that the external I/O pins are in a high-resistance state and monitors the VCC/VCCX/VCCIOn power rail. When the VCC/VCCX/VCCIOn meets the minimum reset level (The reset level may vary for different devices, and different devices monitor different power rails.), the POR circuit releases the internal reset signal, allowing the FPGA to begin its initialization process. When the READY and DONE signals are pulled down, the device moves to the initialized state as shown in Figure 4-5.

Figure 4-5 POR Timing

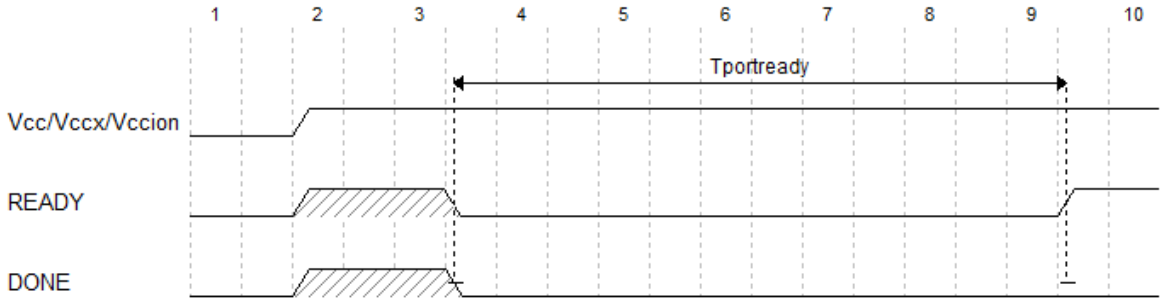


Table 4-2 lists the power rails monitored by POR circuits of different devices.

Table 4-2 Power Rails

Device	POR Rails
GW5AT-15	VCC/VCCX/VCCIO2
GW5AR-15	VCC/VCCX/VCCIO2

4.2.2 Initialization

After the power on reset circuit drives the READY and DONE status pins low, the GOWINSEMI FPGA immediately enters the SRAM initialization state. The initialization is to clear the configuration SRAM inside the FPGA.

The FPGA remains in the initialization state until all of the following conditions are met:

- The TINITL time period has elapsed.
- The RECONFIG_N pin is high.
- The READY pin is not driven low by an external drive.

The READY pin has two functions during the initialization phase:

- Indicates that the FPGA is currently clearing the internal configuration SRAM.
- Acts as an input preventing the FPGA from exiting the initialization state when it's driven low by an external driver.

4.2.3 Configuration

The FPGA enters the configuration state at the rising edge of the READY pin. The internal configuration SRAM of FPGA can be configured via multiple modes according to the MODE pin values. During the time the FPGA receives its configuration data, the READY pin can indicate its internal state. When READY is high, configuration proceeds without issues. If READY is low, an error has occurred and the FPGA does not operate.

4.2.4 Wake-up

When the configuration is done, the FPGA enters a wake-up state and the DONE pin is pulled up. In the wake up state, the FPGA performs the following operations in sequence:

1. Enables the global output signal (GOE), and then the FPGA I/O exits the high resistance state and and takes on its programmed functions. Sets the global set/reset signal (GSR) to prevent the input signal from affecting the Flip-Flop state inside the FPGA.
2. Releases the GSR and the global Write stop signal (GWDISn). Enables the GWDISn to prevent the FPGA from mistakenly rewriting the initialization data of SRAM.
3. Enables the external DONE pin. The DONE pin is a bidirectional, open-drain I/O when enabled. Keep the FPGA wake-up by externally driving the DONE pin low. When the DONE pin is driven high, the FPGA wake-up pahse is complete and enters user mode.

4.2.5 User Modes

After entering the user mode, the FPGA will perform the logical operations you designed immediately. The FPGA will remain in this user mode until the following three events are triggered:

- The RECONFIG_N pin is pulled down externally.
- A REFRESH command is received through the configuration port.
- Power cycle.

Once one of the above three events occurs, the FPGA will re-enter the configuration process.

4.3 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 the 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 GOWINSEMI Software contains an ID verification directive, 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, a CRC code is added. GOWINSEMI devices generate a CRC code 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

according to the user mode selected. If the data is loaded to the SRAM, GOWINSEMI software sets the security bit automatically in the process of bitstream generation, and no user can read the SRAM data.

Note!

GOWINSEMI takes no responsibility for the security of the external Flash.

4.4 Bitstream File Configuration

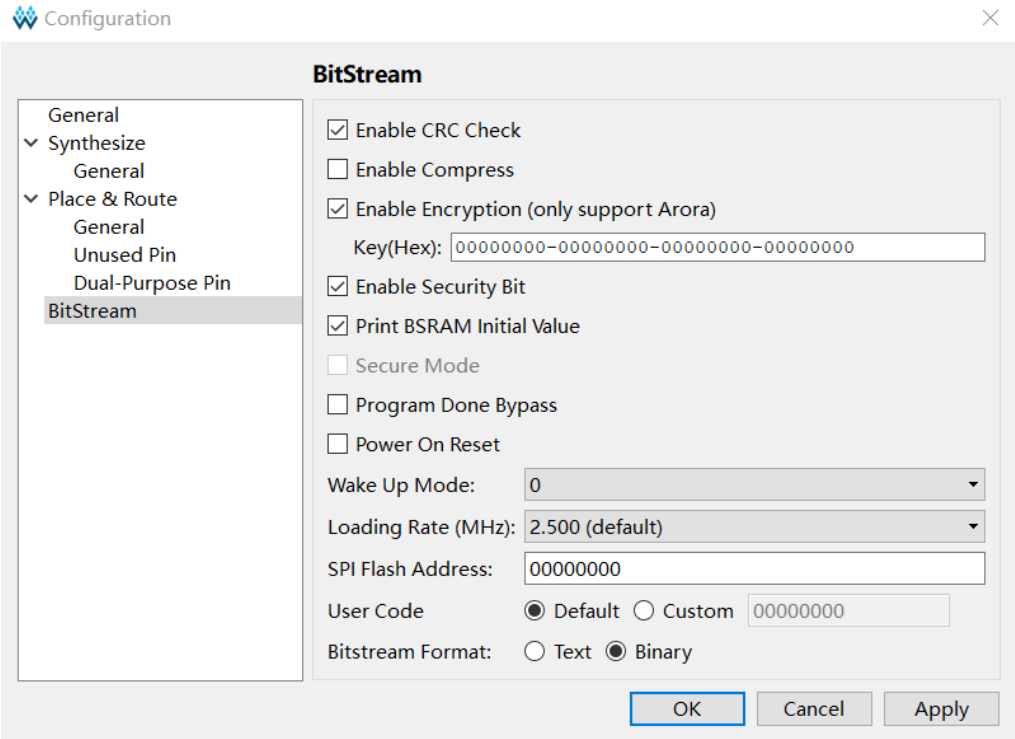
The features of Gowin FPGA products need to be configured and programmed using Gowin software. The settings mainly include the options of configuration pin multiplexing and bitstream data configuration. This chapter describes the bitstream file configuration. For the details about the configuration pin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

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, the input data is checked in real time. The wrong data cannot wake up 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.

4.4.1 Configuration Options

Please refer to Figure 4-6 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.

Figure 4-6 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.

4.4.2 Configure Data Encryption

GOWIN Arora V FPGA products 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 the 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.

Definition

- AES encryption key: AES private key used in AES encryption algorithm, specified by users and referred to as "key” in this manual.
- AES encryption key length: 128 bits;
- Key: An abbreviation for AES encryption key. Arora V FPGA products

offers a 128-bit length space to store Key;

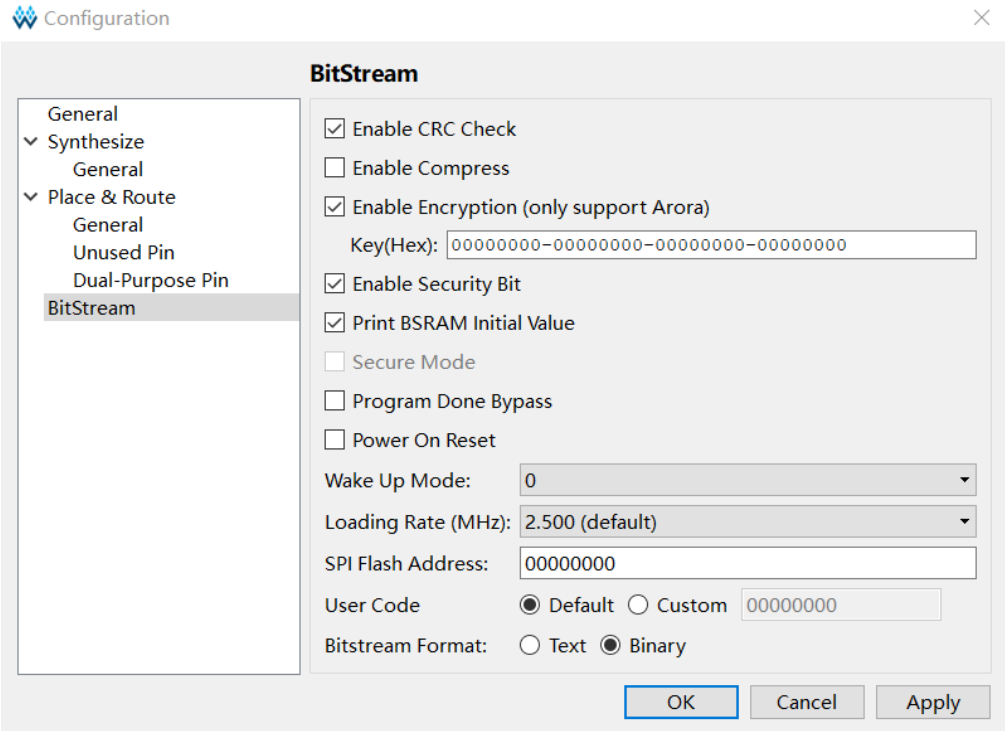
- Lock: To ensure the security of AES Key, it is used to control the read permissions for the Key. This operation is named as "lock" in this manual. When it's locked, all the read back data is 1.

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 Arora)" and input the key value, as shown in Figure 4-7.

Figure 4-7 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.

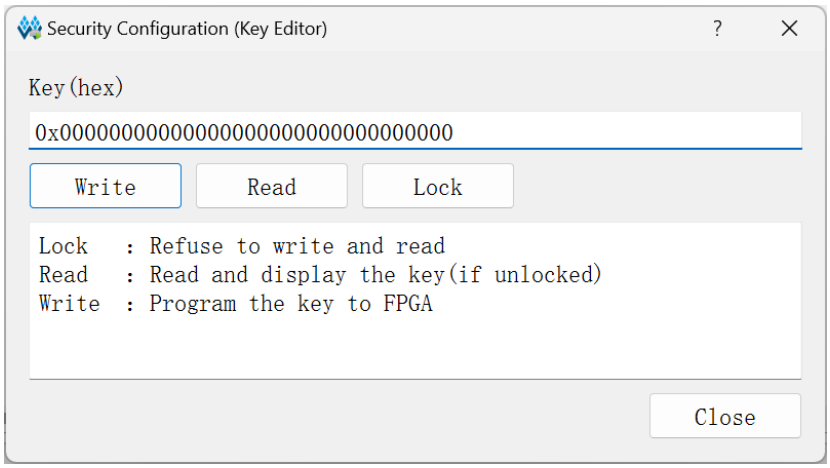
Enter the Decrypt Key

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

1. Open Gowin Programmer;
2. Scan the FPGA device;
3. Right-click on the device name and select " Security Key Setting";
4. Enter the encrypted key value in the pop-up interface, click "Write" to

write the value to the FPGA, as shown in Figure 4-8.

Figure 4-8 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 "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.

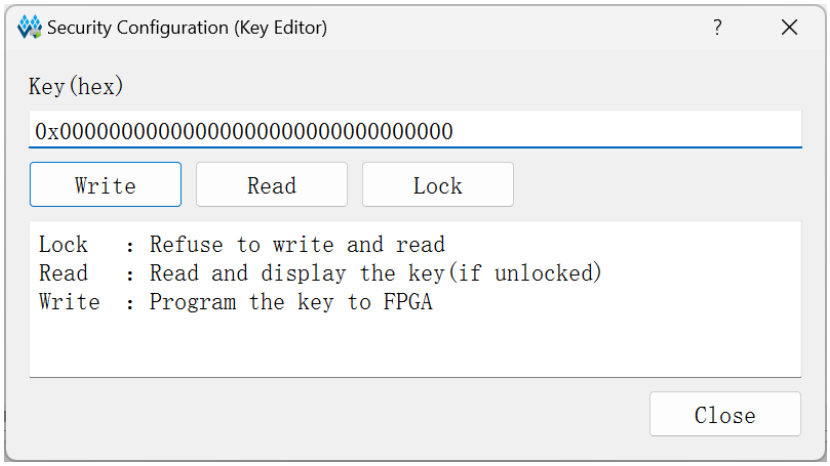
Note!

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.

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 4-9.

Figure 4-9 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 (AES 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.

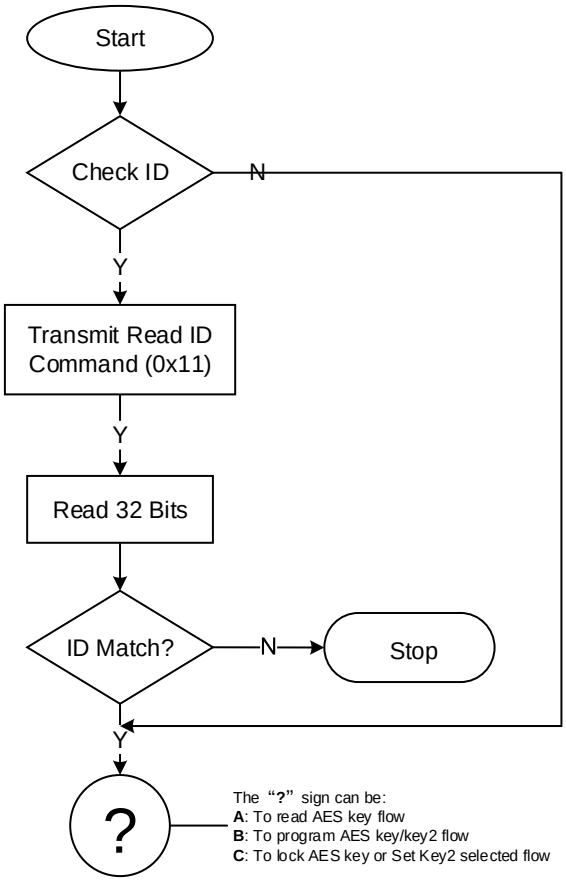
Programming Flow

Figure 4-10 ~ Figure 4-14 show the flow of how to program or lock the AES key. All the flows are based on JTAG protocol.

ID CODE

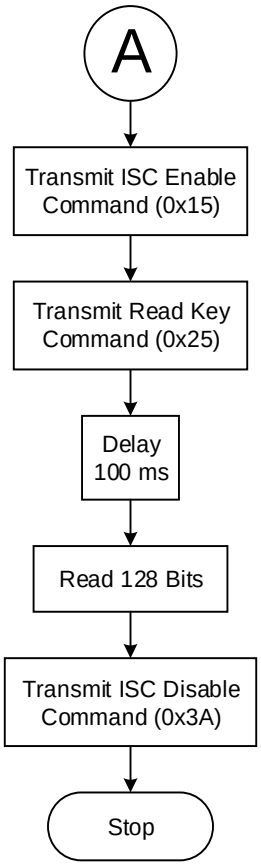
Check the device ID to determine whether the JTAG protocol works properly and whether the programming object is correct to avoid misoperation.

Figure 4-10 Prepare



Read AES Key

Figure 4-11 Read AES Key Flow



Program AES Key

Figure 4-12 Program AES Key Flow

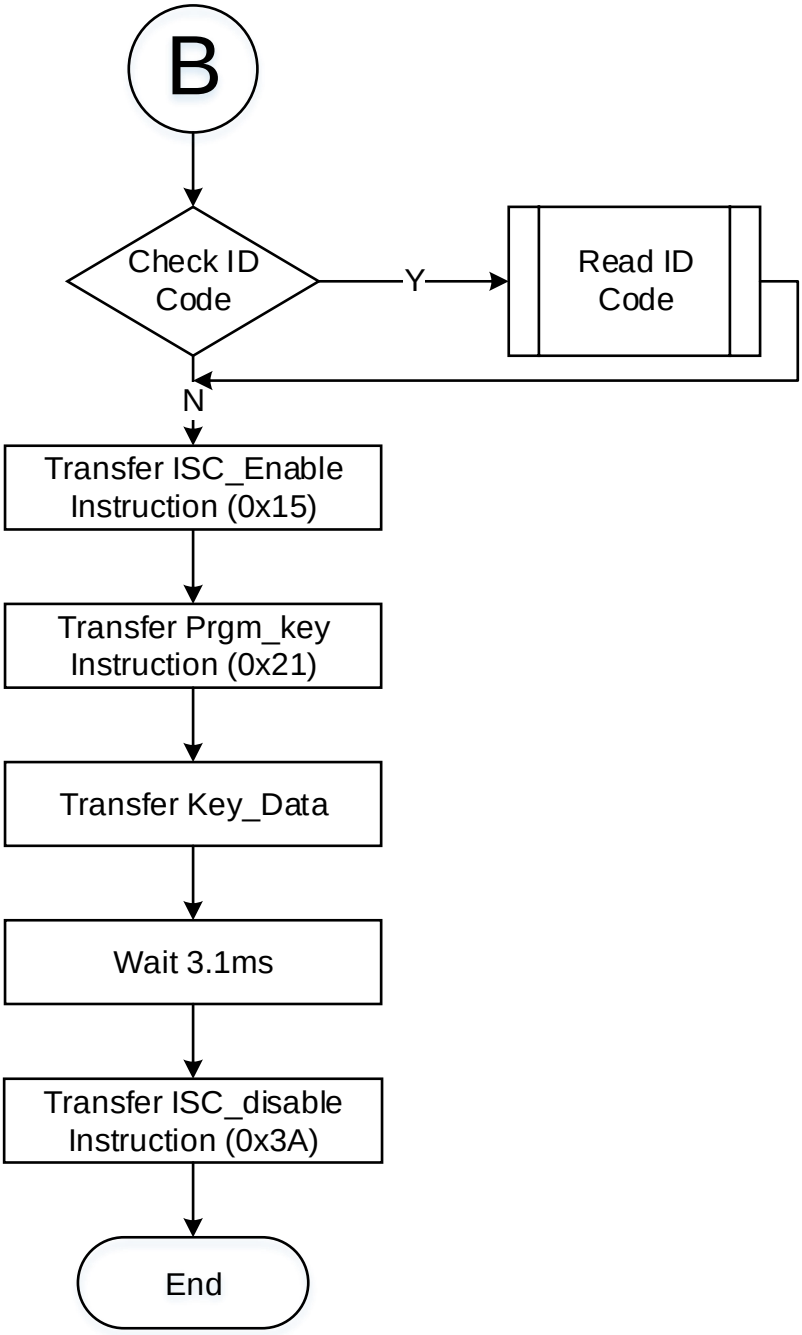
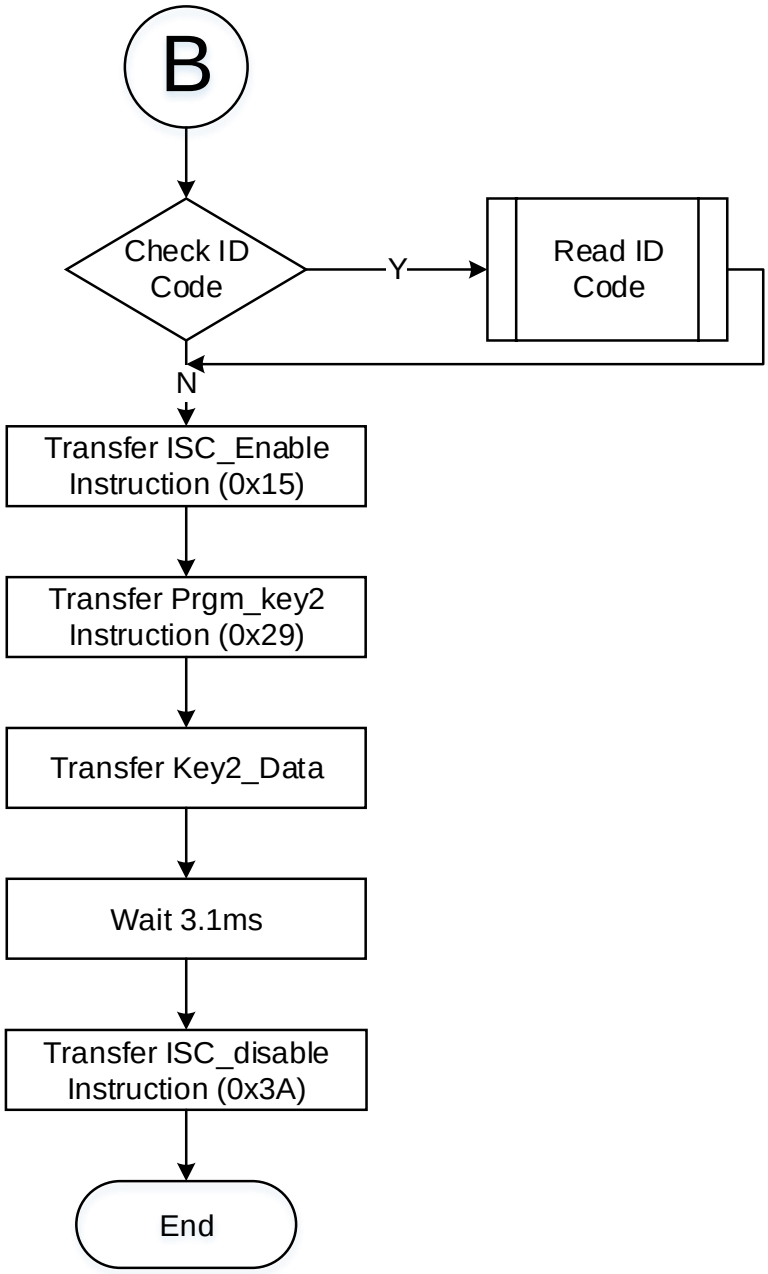


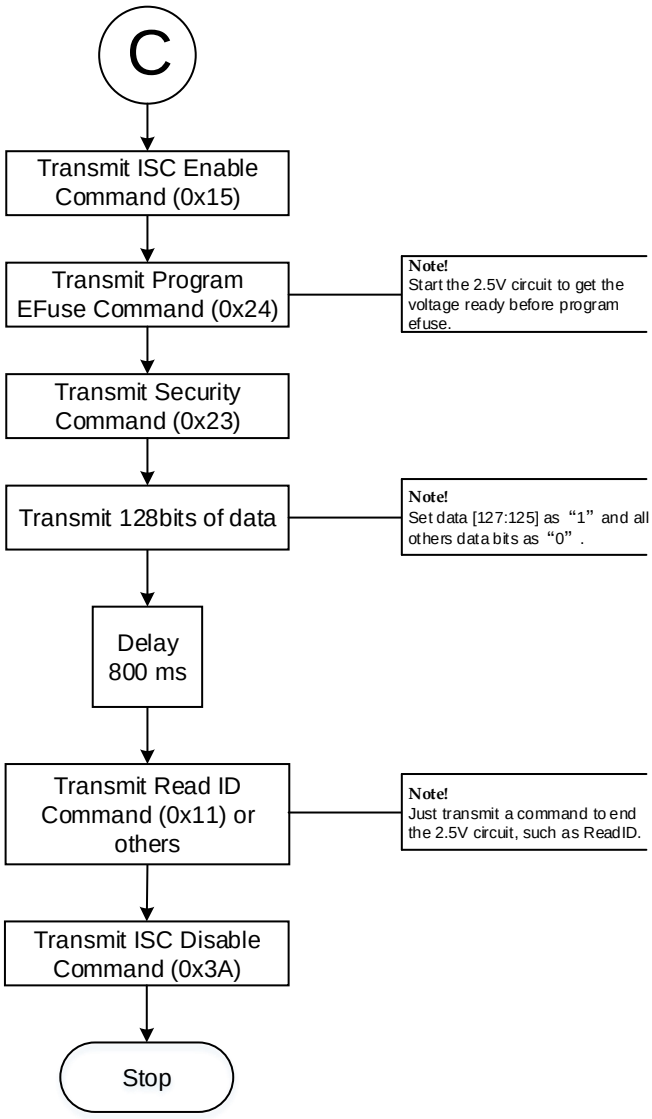
Figure 4-13 Program AES Key2 Flow



Lock AES Key

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

Figure 4-14 Lock AES Key Flow

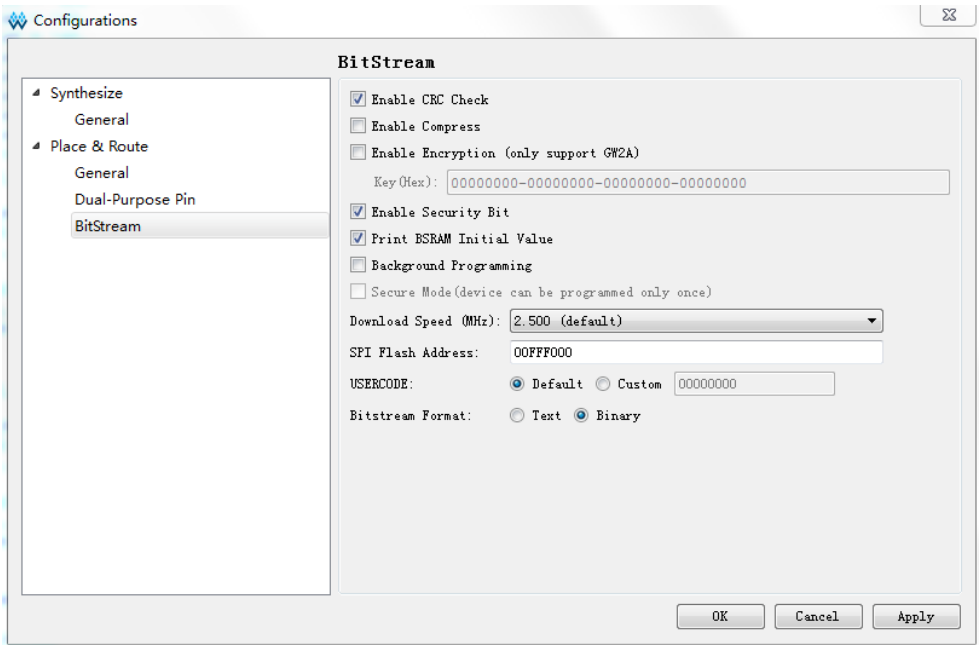


4.4.3 Configuration File Size

The Gowin bitstream format can be 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 are 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 the 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 4-15.

Figure 4-15 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 4-3.

Table 4-3 Arora V 15K FPGA Products Configuration File Size (Max.)

LUT	Max. Configuration File Size
15120	4.8Mb

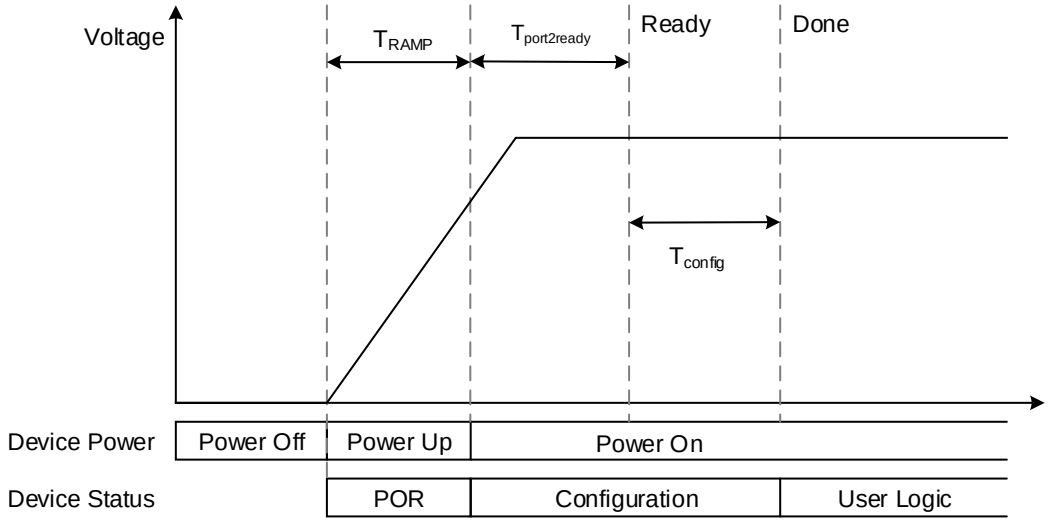
Note!

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

4.4.4 Configuration File Loading Time

Gowin FPGA can be used as Master to read bitstream files from Flash and configure SRAM. 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 Figure 4-16.

Figure 4-16 Process of Power on



Arora V FPGA products support MSPI mode, that is, it can read data from the SPI Flash and configure FPGA automatically. The default frequency of reading configuration file is 100Mhz. 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 125Mhz.

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

The loading time in MSPI mode is as shown in Table 4-4.

Table 4-4 Loading Time in MSPI Mode

LUT	Max. Configuration File Size	Max. value (ms, SPI x 1 Loading frequency = 2.5MHz)	Max. value (ms, SPI x 1 Loading frequency= 25MHz)	Max. value (ms, SPI x 1 Loading frequency = 104MHz)	Max. value (ms, SPI x 4 Loading frequency = 104MHz)
15120	4.8Mb	2021	202	48	12.1

What is listed above is the reference of loading time. From power on to configuration completion of the device, in addition to the configuration time, there are also the power on time (Tramp) and initialization time of the device. The specific power on time is related to the power supply device. Therefore, the approximate time of FPGA from power on to loading completion can be calculated according to the following formula:

MSPI Mode:

$T_{loading\ time} = T_{ramp} + T_{portready}$ + Number of Data Stream Bits / Loading frequency

4.5 SPI Flash Selection

The external SPI Flash device operation instructions supported by Gowin FPGA products are shown in Table 4-5. The Mxic and Winbond products are all in accordance with the requirements. In principle, as long as the read instruction is ordinary or fast, Gowin FPGA can all read data from this Flash.

Table 4-5 SPI Flash Operation Instruction

Operation	Instruction
Read	8’h03
Fast read	8’h0B
Dual output fast read	8’h3B
Quad output fast read	8’h6B
4-byte read	8’h13
4-byte fast read	8’h0C
4-byte dual output fast read	8’h3C
4-byte quad output fast read	8’h6C

Note!

At least one of the Flash read instruction supported by GOWIN FPGAs must be 03 or 0B, and the capacity must be not less than 64Mb.

5 Status Register and Efuse Definition

5.1 Status Register

A Status Register is provided inside the device for debugging. By reading the Status Register, the state of the device can be preliminarily determined, such as whether the wakeup is successful, whether there is a loading error, etc.

Table 5-1 Status Register

Bit	Domain	Description
31	auto_erase	1= bulk erase is ongoing
30	wakeup	1= wakeup status (all global signals, i.e. gsr, gwd, goe and done, are driven high)
29	init_r	1= initialization is done This signal is equal to the status of “INIT_N” pin
28	mfg_done	1= OTP reading is done and MFG pattern is verified
27	decomp_fail	1 – no corresponding dictionary item is found during decompressing a code-word
26:25	sync_det_retry	The retry time of sync pattern detection in MSPI mode 00 – no retry 01 – retry one time 10 – retry two times 11- no “sync pattern” is found after three times detection
24:23	cpu_bus_width	the detected bus width of CPU interface 00 – no BWD pattern is detected 01 – 8-bit mode 10 – 16-bit mode 11 – 32-bit mode
22	ser_running	1: CMSER is ongoing 0: CMSER is in IDLE state
21	ser_ecc_uncorr	1: Uncorrectable ECC error has been detected 0: no uncorrectable ECC error was detected
20	ser_ecc_corr	1: Correctable ECC error has been detected

Bit	Domain	Description
		(whether or not to be corrected is dependent on register setting) 0: no correctable ECC error was detected
19	ser_crc_err	1: CRC error has been detected 0: no CRC error has been detected
18	ser_crc_done	1: at least one time of CRC comparison has been done 0: no CRC comparison was done
17	sspi_mode	
16	key_right	
15	encrypted_format	
14	security_final	
13	done_final	
12	i2c_sram_f	
11	cmd_bypass_state	
10	nj_active_r	
9	auto_boot_1st_fail	
8	prgm_spi	
7	edit_mode	
6	preamble	
5	memory_erase	
4	auto_boot_2nd_fail	
3	time_out_r	Timeout Error (1 indicates that an Error occurred and 0 indicates that no Error occurred)
2	id_fail_r	ID Verify Failed Error (1 indicates that an Error occurred and 0 indicates that no Error occurred)
1	bad_cmd_r	Bad Command Error (1 indicates that an Error occurred and 0 indicates that no Error occurred)
0	crc_error_r	CRC Error (1 indicates that an Error occurred and 0 indicates that no Error occurred)

5.2 OTP Efuse

Arora V FPGA products supports one-time programming and provides 128 Bit OTP space. Bit0 ~ Bit31 is the user space, which can be used to store security and other important information. Bit32~Bit95 is the DNA space, which stores the 64-bit unique identification information of the device.

The definition of OTP is as shown in Table 5-2.

Table 5-2 OTP Definitions

Name	Bit Index	Description
User MISC	0~9bit (6bits)	0bit i2c_addr_2bit
		1bit i2c_addr_0bit
		5~2bit(4bits) otp_2nd_boot_addr[3:0]
		9~6bit(4bits) otp_1st_boot_addr[3:0]
User Define	10~31bit (26bits)	user defined
Device DNA	32~95bit (64bits)	57bit DNA
USER Control	96~127bit(32bits)	96~98bit cfg_aes_only[2:0] When this field is valid, only the encrypted stream files can be used as the configuration data stream.
		99~101bit prgm_key_lock[2:0] When this region is valid, JTAG cannot write to the key in the key/key2 region (as long as two bits of the 3bit data are set to 1, the region is valid).
		102~104bit rd_key_lock[2:0] When this region is valid, JTAG cannot write to the key in the key/key2 region (as long as two bits of the 3bit data are set to 1, the region is valid)
		105~107bit prgm_fuse_user_lock[2:0] When this field is valid, JTAG cannot write to the fuse user field.
		108~110bit rd_fuse_user_lock[2:0] When this field is valid, JTAG cannot read the fuse user field.
		112~114bit prgm_rd_dna_lock[2:0]
		115~117bit lock_sel_key_r[2:0] When the region is valid, it means that the key2 region is selected, otherwise the key region is selected (as long as two bits of the 3bit data are set to 1, this region is valid).
		118~120bit prgm_user_misc_lock[2:0] When this field is valid, JTAG cannot write to the user misc field.

Name	Bit Index	Description
		121-123bit rd_user_misc_lock[2:0] When this field is valid, JTAG cannot read the user misc field.
		124~126bit prgm_user_control_lock[2:0] When this field is valid, JTAG cannot write to the user control field
		127bit reserved

6 Multi Boot and Background Update

The multi boot of GOWINSEMI Arora V FPGA products support flexible dynamic configuration and reliable background upgrade. When an error is detected during configuration, the FPGA can trigger a fallback feature that ensures a Golden firmware can be loaded into the device. The multi boot feature can only be used with the Master SPI mode.

6.1 Multi Boot Process

The Flash space can be divided into multiple regions for storing Multiboot Bitstream1, Multiboot Bitstream2, …, Multiboot BitstreamN, as well as the Golden Bitstream. There is no limit on the number of Bitstreams.

After power-up, the device first loads Multiboot1 Bitstream from address 0x0. Once the loading is completed, the device enters User_Mode to execute the corresponding logic functions.

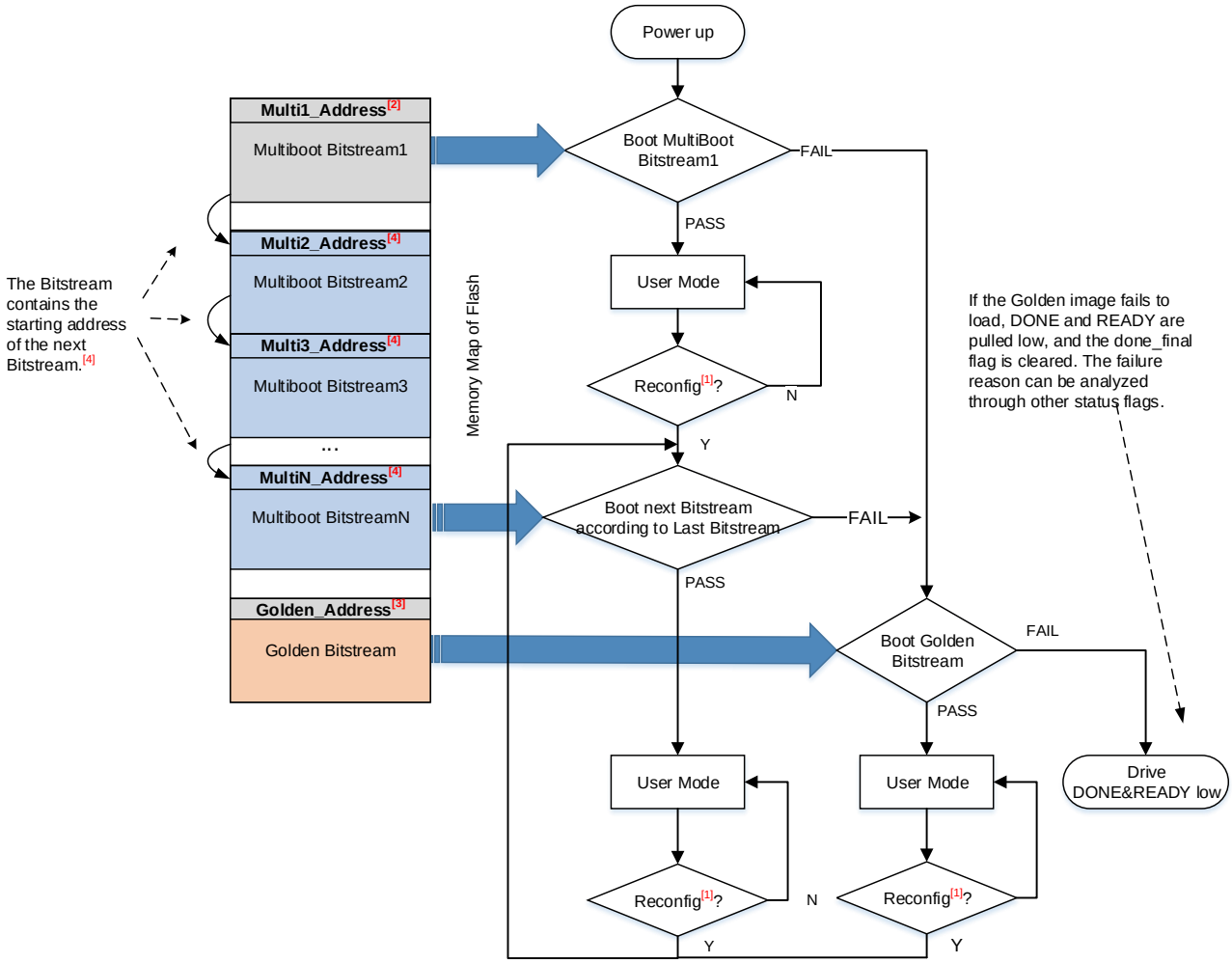
In User_Mode, when a Reconfig command is issued or the RECONFIG_N pin is pulled low, the device loads Multiboot2 Bitstream from the predefined Multi2_Address in MSPI mode. After loading, the device re-enters User_Mode and executes the updated logic functions. Subsequent Reconfig commands or RECONFIG_N assertions will cause the device to load Multiboot3 Bitstream from Multi3_Address, and so on. This allows users to flexibly switch the device logic functions.

If an error occurs during the loading of any Multiboot Bitstream, the device will load the Golden Bitstream from the predefined Golden Address (0x80_0000) to ensure stable system operation. In addition, users can specify the starting address of the next Bitstream to be loaded within the current Bitstream through the EDA tool. For details, refer to Section 3.10 'MSPI 2nd Boot Address Operation' in the SUG502 Gowin Programmer User Guide.

If the Golden Bitstream fails to load, the device drives the DONE and READY pins low, and the done_final flag in the internal status register will be cleared.

The multi- configuration flow is as shown in Figure 6-1.

Figure 6-1 Multi- Configuration Flow



Note!

- [1] Reconfiguration can be triggered by pulling the RECONFIG_N pin low or issuing a Reconfig command.
- [2] The default address is 0x0.
- [3] The Golden_address defaults to 0x800000. Users can modify it through Gowin Programmer. For details, refer to SUG502, Gowin Programmer User Guide, Section 3.10 MSPI 2nd Boot Address Operation.
- [4] The loading address of Multi2–MultiN Bitstreams is stored in the preceding Bitstream. This can be configured in Gowin EDA tool options. For details, refer to SUG100, Gowin EDA Tool User Guide, Section 4.3.3 Project Configuration Editing.

6.2 Background Upgrade and Hotboot

Arora V FPGA products support background upgrade through JTAG/SSPI/QSSPI or UserLogic. Please refer to 3Configuration Interface for accessing Flash through the JTAG/SSPI/QSSPI interface. For accessing Flash through UserLogic, GOWIN official IP needs to be used. To improve system robustness, it is advised to reserve Golden Bistream area during remote upgrades.

In addition, Arora V FPGA products also support Hotboot mode, that is, after background upgrade, if the Reboot command is received or

RECONFIG_N is triggered by the low level pulse from outside, all output IO of the device will be locked in the current state until the wakeup is successful.

GOWIN

PROGRAMMING FOR THE FUTURE