



GW2AN-18X & 9X 器件 编程配置手册

UG702-1.2.1, 2025-03-28

版权所有 © 2025 广东高云半导体科技股份有限公司

GOWIN高云、Gowin、晨熙、高云均为广东高云半导体科技股份有限公司注册商标，本手册中提到的其他任何商标，其所有权利属其拥有者所有。未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本档内容的部分或全部，并不得以任何形式传播。

免责声明

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外，高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和 / 或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。高云半导体对文档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，高云半导体保留修改文档中任何内容的权利，恕不另行通知。高云半导体不承诺对这些文档进行适时的更新。

版本信息

日期	版本	说明
2021/09/10	1.0	初始版本。
2022/03/04	1.0.1	更新配置管脚的默认状态。
2022/07/28	1.0.2	增加 I ² C 配置模式的说明。
2022/09/06	1.1	<ul style="list-style-type: none">● 更新 5.8 I²C 配置模式。● 增加 6.5 背景烧录 (Background Programming)。
2022/12/08	1.1.1	<ul style="list-style-type: none">● 更新 CLKHOLD_N 管脚的说明。● 新增图 5-40 CPU 配置模式时序图。● 新增表 5-14 CPU 配置模式时序参数。
2023/04/13	1.1.2	<ul style="list-style-type: none">● 更新图 5-39 CPU 模式配置流程示意图。● 更新图 5-40 CPU 配置模式时序图。● 更新表 5-14 CPU 配置模式时序参数。● 新增使用 QSSPI 模式编程 Flash 的相关描述。
2023/07/25	1.1.3	<ul style="list-style-type: none">● 更新 RECONFIG_N 管脚的注意事项。● 更新表 5-1 高云半导体 FPGA GW2AN-18X/9X 产品重新上电和 RECONFIG_N 触发时序参数。● 新增关于 MSPI 模式时钟频率误差的注释。
2023/11/16	1.1.4	<ul style="list-style-type: none">● 更新表 5-8 Status Register 与配置加载相关的条目。● 修正配置文件加载时长的计算公式。● 修改表 6-1 高云半导体 FPGA GW2AN-18X/9X 产品配置文件大小 (最大情况) 的注释。● 修改图 5-51 I²C 配置 Flash 的流程图。● 更新 5.4.5 SSPI 配置模式连接示意图。● 更新 I²C 写 Flash 的流程。
2024/7/26	1.2	<ul style="list-style-type: none">● 更新“5.2 AUTO BOOT 配置”。● 更新“图 5-16 菊花链连接示意图”。● 新增“附录 A 管脚状态信息”。● 完善 JTAG 配置模式的说明。● 完善 I²C 配置模式的说明。● 更新“表 5-1 高云半导体 FPGA GW2AN-18X/9X 产品重新上电和 RECONFIG_N 触发时序参数”：修正 T_{portready} 的值。
2025/03/28	1.2.1	更新关于菊花链操作的描述。

目录

目录	i
图目录	iii
表目录	v
1 关于本手册	1
1.1 手册内容	1
1.2 相关文档	1
1.3 术语、缩略语	2
1.4 技术支持与反馈	3
2 名词解释	4
3 配置模式概述	6
4 配置管脚介绍	8
4.1 配置管脚列表及复用选项	8
4.1.1 配置管脚列表	8
4.1.2 配置管脚复用	9
4.2 配置管脚功能及应用	11
5 配置模式详述	14
5.1 配置须知	14
5.2 AUTO BOOT 配置	17
5.3 JTAG 配置	18
5.3.1 JTAG 配置模式管脚	18
5.3.2 JTAG 配置模式连接示意图	19
5.3.3 JTAG 配置模式时序图	20
5.3.4 JTAG 相关配置流程	20
5.4 SSPI 配置模式	38
5.4.1 SSPI 配置模式管脚	38
5.4.2 SSPI 配置模式时序图	39
5.4.3 SSPI 配置指令	40
5.4.4 SSPI Configure SRAM 流程图	43
5.4.5 SSPI 配置模式连接示意图	44

5.4.6 SSPI 模式下的多 FPGA 连线示意图	47
5.5 QSSPI 配置模式	48
5.6 CPU 配置模式.....	53
5.6.1 配置时序	54
5.7 SERIAL 配置模式	55
5.8 I ² C 配置模式	57
5.8.1 配置指令	59
5.8.2 I ² C 配置 SRAM 流程.....	61
5.8.3 I ² C 配置 Flash 流程.....	61
6 比特流文件配置.....	65
6.1 配置选项设置	65
6.2 配置数据加密	66
6.2.1 定义.....	66
6.2.2 输入加密密钥.....	67
6.2.3 输入解密密钥.....	67
6.2.4 AES 密钥编程操作	68
6.2.5 AES 密钥编程流程.....	70
6.3 配置文件大小	73
6.4 配置文件加载时长.....	75
6.5 背景烧录 (Background Programming)	76
7 安全性考虑.....	77
8 边界扫描操作	79
附录 A 管脚状态信息	81

图目录

图 4-1 配置管脚复用设置.....	11
图 5-1 固定管脚推荐接法.....	16
图 5-2 重新上电时序图	16
图 5-3 触发时序图	17
图 5-4 JTAG 配置模式连接示意图.....	19
图 5-5 JTAG 菊花链配置模式连接示意图	19
图 5-6 JTAG 配置模式时序图	20
图 5-7 TAP 状态机	21
图 5-8 指令寄存器访问时序	22
图 5-9 数据寄存器访问时序	22
图 5-10 读取 ID Code 状态机流程图	24
图 5-11 读取 ID Code 指令-0x11 访问时序	24
图 5-12 读取 ID Code(0x0000581B)数据寄存器访问时序.....	24
图 5-13 配置 SRAM 流程.....	26
图 5-14 读取 SRAM 的流程	28
图 5-15 采用 Boundary Scan 模式编程 SPI Flash 流程示意图.....	35
图 5-16 菊花链连接示意图.....	37
图 5-17 SSPI 配置模式时序图	39
图 5-18 读取 ID Code 时序示意图	40
图 5-19 Write Enable (0x15) 时序示意图.....	41
图 5-20 Write Disable (0x3A00) 时序示意图	41
图 5-21 Write Data (0x3B) 时序示意图.....	42
图 5-22 SSPI 配置模式连接示意图.....	44
图 5-23 SSPI 编程内置 Flash 连接示意图.....	44
图 5-24 SSPI 读取内置 Flash ID Code 时序示意图.....	45
图 5-25 SSPI 写 WriteDisable (0x04) 指令的时序示意图	45
图 5-26 SSPI 写 WriteEnable (0x06) 指令的时序示意图	46
图 5-27 SSPI 编程一个 page 的时序示意图	46
图 5-28 SSPI 回读一个 byte 的时序示意图	46

图 5-29 SSPI 配置 Flash 流程图.....	47
图 5-30 多 FPGA 连线示意图 1	47
图 5-31 多 FPGA 连线示意图 2	48
图 5-32 QSSPI Write Data (0x6B) 时序图	49
图 5-33 QSSPI 配置 SRAM 流程图	50
图 5-34 QSSPI 编程 Flash 流程图.....	51
图 5-35 QSSPI write instruction 时序图	52
图 5-36 QSSPI write data 时序图.....	52
图 5-37 QSSPI page program 时序图	52
图 5-38 CPU 配置模式连接示意图	53
图 5-39 CPU 模式配置流程示意图	54
图 5-40 CPU 配置模式时序图.....	54
图 5-41 SERIAL 配置模式连接示意图	56
图 5-42 SERIAL 配置模式时序图	56
图 5-43 I ² C 配置模式连接示意图	57
图 5-44 I ² C 配置模式时序图	58
图 5-45 Reinit 时序图	60
图 5-46 配置 SRAM 时序图	60
图 5-47 配置 Flash 时序图.....	60
图 5-48 Reboot 时序图	60
图 5-49 I ² C 配置 SRAM 的流程图.....	61
图 5-50 I ² C 擦除 Flash 的流程图	62
图 5-51 I ² C 配置 Flash 的流程图	64
图 6-1 配置选项	66
图 6-2 加密密钥设置方法.....	67
图 6-3 解密密钥设置方法.....	68
图 6-4 AES 编程对话框	68
图 6-5 Prepare.....	70
图 6-6 Read AES Key Flow	71
图 6-7 Program AES Key Flow.....	72
图 6-8 Lock AES Key Flow	73
图 6-9 比特流格式生成	74
图 6-10 Background Programming 选项	76
图 8-1 边界扫描操作示意图	80

表目录

表 1-1 术语、缩略语	2
表 2-1 名词解释	4
表 3-1 配置模式	6
表 4-1 配置管脚列表	8
表 4-2 配置管脚复用选项	9
表 4-3 管脚功能	11
表 5-1 高云半导体 FPGA GW2AN-18X/9X 产品重新上电和 RECONFIG_N 触发时序参数	17
表 5-2 JTAG 配置模式管脚定义	18
表 5-3 JTAG 配置模式时序参数	20
表 5-4 Gowin 部分 FPGA 产品 ID CODE	23
表 5-5 发送指令过程中 TDI 和 TMS 的值变化	23
表 5-6 器件 SRAM 地址数量和地址长度	27
表 5-7 管脚状态	35
表 5-8 Status Register 与配置加载相关的条目	36
表 5-9 SSPI 配置模式管脚	38
表 5-10 SSPI 配置模式时序参数	39
表 5-11 配置指令	40
表 5-12 QSSPI 配置模式管脚	48
表 5-13 CPU 配置模式管脚	53
表 5-14 CPU 配置模式时序参数	55
表 5-15 SERIAL 配置模式管脚定义	55
表 5-16 SERIAL 配置模式时序参数	56
表 5-17 I ² C 配置模式管脚定义	57
表 5-18 I ² C 配置模式时序参数	58
表 5-19 I ² C 配置模式频率及地址	59
表 5-20 I ² C 配置指令	59
表 6-1 高云半导体 FPGA GW2AN-18X/9X 产品配置文件大小（最大情况）	74
表 6-2 Autoboot 模式数据流文件加载时长	75
表 A-1 GW2AN-18X & 9X 器件的管脚在各个阶段时的状态(MSPI, MODE=00)	81

表 A-2 GW2AN-18X & 9X 器件的管脚在各个阶段时的状态(AUTOBOOT, MODE=01)	82
--	----

1 关于本手册

1.1 手册内容

本手册主要介绍高云半导体晨熙(Arora)家族 FPGA 产品中的 GW2AN-18X/9X 系列在编程配置方面的通用特性及功能，旨在帮助用户更好地使用 Gowin FPGA 产品。

1.2 相关文档

通过登录高云半导体网站 www.gowinsemi.com.cn 可以下载、查看以下相关文档：

- [DS971, GW2AN-18X & 9X 器件数据手册](#)
- [UG973, GW2AN-18X & 9X 器件封装与管脚手册](#)
- [UG972, GW2AN-18X 器件 Pinout 手册](#)
- [UG978, GW2AN-9X 器件 Pinout 手册](#)

1.3 术语、缩略语

表 1-1 中列出了本手册中出现的相关术语、缩略语及相关释义。

表 1-1 术语、缩略语

术语、缩略语	全称	描述
Background Programming	Embedded Flash Background Programming	是指在不影响 FPGA 当前运行的功能和 I/O 状态的情况下，对内置 Flash 的数据流文件进行更新。
Bitstream	Bitstream Data	配置 FPGA SRAM 的数据
Bscan	Boundary Scan	边界扫描测试技术
Configuration	Configuration	配置 FPGA SRAM 区域的过程
Configuration Data	Configuration Data	配置 FPGA SRAM 的数据
Configuration Mode	Configuration Mode	配置模式，决定 Configuration Data 源
CPU	Central Processing Unit	中央处理器
CRC	Cyclic Redundancy Check	循环冗余校验
Edit Mode	Edit Mode	FPGA 处于 Configuration 或 Programming 所在模式
EFlash/EmbFlash	Embedded Flash	FPGA 内置 Flash 存储器
FPGA	Field Programmable Gate Array	现场可编程门阵列
FS file	Fuses file	包含配置数据的 ASCII 文件
GPIO	Gowin Programmable I/O	Gowin 可编程通用管脚
I2C (I ² C、IIC)	Inter-Integrated Circuits	集成电路总线
ID	Identification	身份标识号
IEEE	Institute of Electrical and Electronics Engineers	电气和电子工程师协会
Internal Flash	Internal Flash	同 Embedded Flash
JTAG	Joint Test Action Group	联合测试行动组
LSB	Least Significant Bit	最低有效位（优先）
LUT	Look-up Table	查找表
MSB	Most Significant Bit	最高有效位（优先）
MSPI	Master Serial Peripheral Interface	主串行外设接口
Programming	Programming	将 Configuration Data 烧录到 Embedded Flash 或 External Flash 存储器的过程
SCL	Serial Clock	I2C 上的时钟线
SDA	Serial Data	I2C 上的数据线
Security Bit	Security Bit	安全位（使 SRAM 回读永为高电平）
SPI	Serial Peripheral Interface	串行外设接口
SRAM	Static Random Access Memory	静态随机存储器

术语、缩略语	全称	描述
SSPI	Slave Serial Peripheral Interface	从串行外设接口
TAP	Test Access Port	测试访问口
User Mode	User Mode	FPGA 在 Configuration 或 Programming 完成后，并且逻辑功能执行的模式

1.4 技术支持与反馈

高云半导体提供全方位技术支持，在使用过程中如有任何疑问或建议，可直接与公司联系：

网址：www.gowinsemi.com.cn

E-mail: support@gowinsemi.com

Tel: +86 755 8262 0391

2 名词解释

本章主要介绍高云半导体 FPGA 产品编程配置过程中常用的一些名词及含义，帮助用户熟悉高云半导体 FPGA 产品编程配置相关的概念。

表 2-1 名词解释

名词	释义
编程 (Program)	将 Gowin 云源软件生成的比特流数据写入到 FPGA 内置 Flash 或者与 FPGA 相连的外部 SPI Flash 的过程。
配置 (Configure)	将 Gowin 云源软件生成的比特流数据通过外部接口或内置 Flash 加载到 FPGA 的 SRAM 存储区的过程。
GowinCONFIG	除通用的 JTAG 配置模式外，高云半导体 FPGA 产品支持的其他配置方式，包括自启动配置，双启动配置，MSPI 配置，SSPI 配置，SERIAL 配置，CPU 配置。每款器件支持的 GowinCONFIG 配置模式多少取决于器件型号和封装类型。
MODE[1:0]	与 GowinCONFIG 相关的 2 个 MODE 管脚的值的表示方法。
自启动配置 (AUTO BOOT)	FPGA 从内置 Flash 读取比特流数据进行配置的过程。只有非易失器件支持此模式。
MSPI 配置	FPGA 作为主器件 (master)，通过 SPI 接口主动从外部 Flash 读取比特流数据进行配置的过程。
SSPI 配置	FPGA 作为从器件 (slave)，外部主机 (master) 通过 SPI 接口写入比特流数据进行配置的过程。
QSSPI 配置	FPGA 作为从器件 (slave)，外部主机 (master) 通过 QSPI 接口写入比特流数据进行配置的过程。
SERIAL 配置	FPGA 作为从器件 (slave)，外部主机 (master) 通过串行接口写入比特流数据进行配置的过程。
CPU 配置	FPGA 作为从器件 (slave)，外部主机 (master) 通过并行接口 (数据位宽 8-bit) 写入比特流数据进行配置的过程。
I ² C 配置	FPGA 作为从器件 (slave)，外部主机 (master) 通过 I ² C 接口写入比特流数据进行配置的过程。
多重配置 (MULTI BOOT)	MSPI 配置模式的衍生概念，是指 FPGA 从外部 Flash 的不同地址读取比特流数据进行配置的过程。用户在前一个比特流数

名词	释义
	据中写入后一次配置的比特流数据的加载地址，在器件不掉电的情况下通过触发 RECONFIG_N 切换数据流文件完成配置。支持 MSPI 配置模式的 FPGA 产品均支持此模式。
远程升级	用户的一种应用场景，即在 FPGA 启动工作后，若有升级需求，先通过远程操作将比特流数据写入到外部 Flash 中，通过触发 RECONFIG_N 或重新上电使 FPGA 读取外部 Flash 完成配置的过程。
菊花链	FPGA 器件以串行的方式依次连接起来的一种方式，可以从链首按照连接顺序依次对器件进行配置，只有相邻的器件之间才能传输数据。
用户模式 (User Mode)	FPGA 完成一次配置操作后，将控制权移交给用户的行为。配置管脚复用为普通 I/O 的设置仅在用户模式下生效。
编辑模式 (Edit Mode)	可以对 FPGA 进行编程或配置操作的模式。编辑模式下所有的配置管脚无法作为普通 I/O 使用，所有的普通管脚输出为高阻态（透明传输除外）。
ID CODE	高云半导体 FPGA 器件的身份标识，每一个系列的器件具有独立的编号。
USER CODE	用户为自己所使用的 FPGA 器件进行的身份标识，可以通过 Gowin 编程软件写入到器件中，最高可支持 32-bit。
安全位 (Security Bit)	高云半导体为保护 FPGA 产品配置数据运行时的安全性所做的特殊设计。用户将设置了安全位的比特流数据写入器件 SRAM 后，任何人都将无法进行数据回读操作。云源软件默认为所有 FPGA 产品的比特流数据设置了安全位。
加密 (Encryption)	晨熙(Arora)家族 FPGA 产品支持的特性，加密的比特流数据写入 FPGA 后，器件自行与事先存储的密钥匹配，匹配成功后进行解密并唤醒器件；匹配失败后器件无法工作。

3 配置模式概述

晨熙(Arora)家族 FPGA 产品除了支持业界通用的 JTAG 配置模式外，还支持高云半导体特有的 GowinCONFIG 配置模式。每款器件支持的 GowinCONFIG 配置模式的多少取决于不同型号和封装形式。器件支持比特流数据加密和安全位设置功能，为用户设计提供了安全性保障。晨熙(Arora)家族 FPGA 产品支持比特流数据的解压缩功能，用户可以将比特流数据进行压缩以节约存储空间。

高云半导体 FPGA GW2AN-18X/9X 产品内置 16M-bit Serial Flash (With Quad SPI)存储器，支持最大 100MHz Quad SPI 模式配置，支持 Fixed-Address GODEN-IMAGE 模式。

GW2AN-18X/9X FPGA 产品支持的配置模式如表 3-1 所示。

表 3-1 配置模式

配置模式		MODE[1:0] ^[1]	相关说明
JTAG		XX ^[2]	外部 Host 通过 JTAG 接口对 FPGA 产品进行配置。最高频率 62.5MHz。
GowinCONFIG	MSPI ^[5]	00	FPGA 作为 Master，通过 SPI 协议从内置 Flash 读取配置数据进行配置。最高支持 100MHz。
	Autoboot	01	FPGA 作为 Master，通过 QSPI 协议从内置 Flash 读取配置数据进行配置。支持 GOLDEN IMAGE。最高支持 100MHz。
	SSPI	0X ^[3]	在 Autoboot 或 MSPI 完成后自动支持。外部 Host 通过 SPI 协议对 FPGA 产品进行配置。最高支持 100MHz。
	QSSPI		在 Autoboot 或 MSPI 完成后自动支持。外部 Host 通过 QSPI 协议对 FPGA 产品进行配置。最高支持 100MHz。

配置模式		MODE[1:0] ^[1]	相关说明
	I ² C		在 Autoboot 或 MSPI 完成后自动支持。外部 Host 通过 I ² C 协议对 FPGA 产品进行配置。支持频率范围 100KHz~555KHz。
	SERIAL ^[4]	10	外部 Host 通过 DIN 接口对 FPGA 产品进行配置。
	CPU ^[4]	11	外部 Host 通过 DBUS 接口对 FPGA 产品进行配置。

注!

- [1]对于一些 MODE 管脚没有全部封装出来的器件，未封装出来的 MODE 值请参考对应封装的 pinout 手册；
- [2]JTAG 配置模式与 MODE[1:0]输入值无关；
- [3]SSPI 配置模式与 MODE0 输入值无关；
- [4]CPU 配置模式的 SCLK、WE_N 和 CLKHOLD_N 管脚与 SERIAL 配置模式共用，CPU 配置模式的数据总线管脚与 MSPI 和 SSPI 配置模式的管脚共用。
- [5] MSPI 配置模式的时钟频率存在 ±10% 的误差。

注!

关于配置管脚列表、配置管脚复用及配置管脚功能及应用信息请参考 4 配置管脚介绍。

4 配置管脚介绍

高云半导体 FPGA 产品配置模式多样，包括通用型的 JTAG 配置、主动配置、被动配置、串行配置和并行配置等，可以满足用户不同外设环境下的各种需求。编程配置相关的管脚既能够完成配置功能，又可以设置为普通的 I/O，用户可根据实际使用情况进行选择。用户也可以根据配置管脚的功能对其灵活控制，满足一些特殊需求。

4.1 配置管脚列表及复用选项

4.1.1 配置管脚列表

高云半导体 FPGA 产品所有与配置相关的管脚如表 4-1 所示，表中也标注了每种配置模式使用到的管脚及芯片封装过程中的管脚共用情况。

表 4-1 配置管脚列表

管脚名称	I/O 类型	JTAG	GowinCONFIG					
			AUTO BOOT	I ² C	SSPI	QSSPI	SERIAL	CPU
RECONFIG_N	I	√	√	√	√	√	√	√
JTAGSEL_N	I	√						
TDO	O	√						
TMS	I	√						
TCK	I	√						
TDI	I	√						
READY	I/O	√	√	√	√	√	√	√
DONE	I/O	√	√	√	√	√	√	√
MODE[1:0]	I		√	√	√	√	√	√
SCLK	I				√	√	√	√
CLKHOLD_N/DIN	I				√	√	√	√
WE_N/DOUT	O					√	√	√
D7	I/O							√

管脚名称	I/O 类型	JTAG	GowinCONFIG					
			AUTO BOOT	I ² C	SSPI	QSSPI	SERIAL	CPU
D6	I/O							√
D5	I/O							√
D4	I/O							√
FASTRD_N/D3	I/O							√
SI/D2	I/O				√	√		√
SO/D1	I/O				√	√		√
SSPI_CS_N/D0	I/O				√	√		√
SCL	I			√				
SDA	I/O			√				

注！

- 不同型号和封装的器件支持的配置模式不同，详细信息请参考 3 配置模式；
- 关于不同配置模式下各管脚的定义请参考 5 配置模式。

4.1.2 配置管脚复用

为最大化地提高 I/O 的利用率，高云半导体的 FPGA 产品支持将配置管脚设置为普通 I/O 使用。所有系列的 FPGA 上电后未进行任何配置操作之前，与配置相关的管脚均默认作为配置管脚使用。配置成功后，器件进入用户模式，按照用户选择的复用选项重新分配管脚的功能。

注！

用户设置管脚复用选项时，需确保管脚的外部初始连接状态不影响器件的配置过程。对于影响配置的连接，需要先进行隔离处理，等待 FPGA 进入用户模式后再进行修改。

配置管脚复用选项如表 4-2 所示。

表 4-2 配置管脚复用选项

设置名称	设置选项	说明
JTAG PORT	默认状态	TMS, TCK, TDI, TDO 作为专用配置管脚, JTAGSEL_N 作为 GPIO。
	设置为普通 I/O	JTAGSEL_N 作为专用配置管脚： <ul style="list-style-type: none"> ● JTAGSEL_N=0, TMS,TCK,TDI,TDO 作为配置管脚； ● JTAGSEL_N=1, TMS,TCK,TDI,TDO 在配置结束后作为 GPIO。
I ² C PORT	默认状态	SCL,SDA 作为专用配置管脚。
	设置为普通 I/O	SCL,SDA 在配置结束后作为 GPIO。
SSPI PORT	默认状态	SCLK, CLKHOLD_N, SSPI_CS_N, SI 和 SO 作为专用配置管脚。
	设置为普通 I/O	SCLK, CLKHOLD_N, SSPI_CS_N, SI 和 SO 在配置结束后作为 GPIO。

设置名称	设置选项	说明
QSSPI PORT	默认状态	SCLK, CLKHOLD_N, SSPI_CS_N, SI 和 SO、QSSPI_WPN 作为专用配置管脚。
	设置为普通 I/O	SCLK, CLKHOLD_N, SSPI_CS_N, SI 和 SO、QSSPI_WPN 在配置结束后作为 GPIO。
RECONFIG_N	默认状态	专用配置管脚。
	设置为普通 I/O	配置结束后作为 GPIO。
READY	默认状态	专用配置管脚。
	设置为普通 I/O	配置结束后作为 GPIO。
DONE	默认状态	专用配置管脚。
	设置为普通 I/O	配置结束后作为 GPIO。

注!

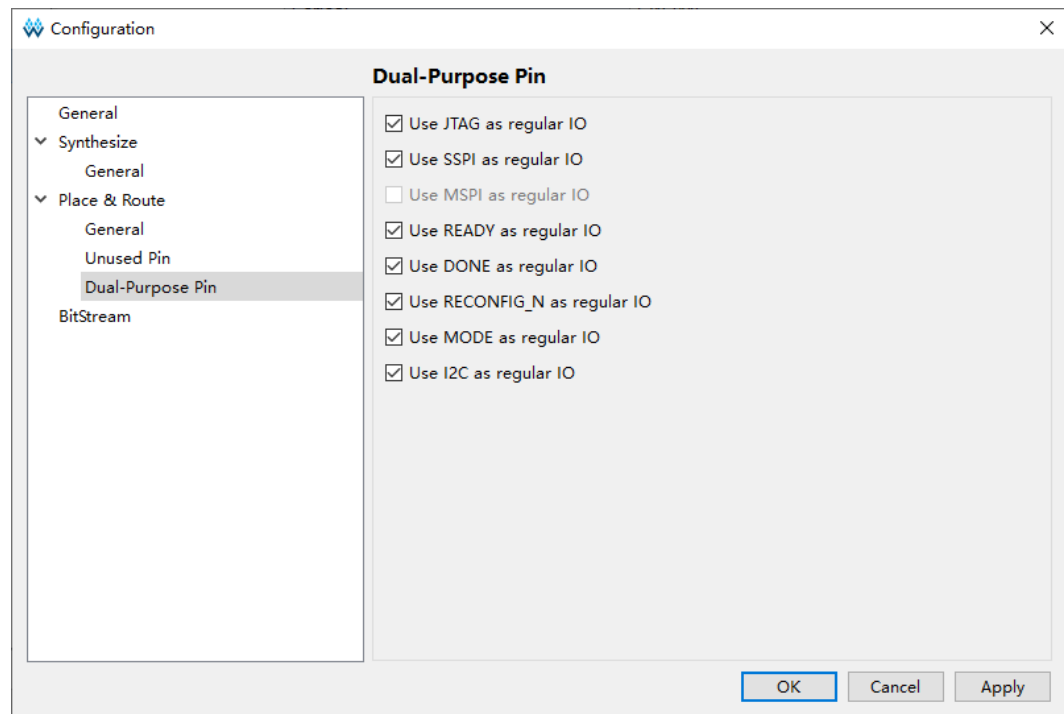
- [1]对于 JTAGSEL_N 未封装出来的器件，在调试 JTAG 管脚复用的案例时，建议用户在 FPGA 上电前确保当前 MODE 值不会使 FPGA 进行配置操作，以避免其他比特流数据影响配置过程。用户上电后手动进行 JTAG 配置后，器件进入用户模式，JTAG 管脚变为 GPIO。
- [2]SERIAL 和 CPU 配置模式的管脚由于与其他配置模式共用，无法单独设置为 GPIO，但是当这些管脚工作在非共用配置模式时可以设置为 GPIO。

配置管脚复用

通过 Gowin 云源软件配置管脚复用：

1. 打开 Gowin 云源软件中相应的工程；
2. 在菜单栏中选择“Project>Configuration>Dual-Purpose Pin”，如图 4-1 所示；
3. 勾选对应选项设置配置管脚的复用情况。

图 4-1 配置管脚复用设置



4.2 配置管脚功能及应用

RECONFIG_N、READY 和 DONE 管脚是每种模式都会用到的管脚。其他配置管脚可根据具体应用设置为配置专用管脚或普通管脚。

表 4-3 管脚功能

管脚名称	功能描述
RECONFIG_N	<p>作为配置管脚时，类型为输入，具有内部弱上拉。低电平有效，相当于 FPGA 编程配置的复位功能，RECONFIG_N 拉低时 FPGA 无法进行任何方式的配置操作。FPGA 上电、初始化、及配置过程中务必保持高电平或悬空，配置完成之后可以释放。</p> <p>作为配置管脚时，需要一个脉冲宽度不少于 25ns 的低电平启动 GowinCONFIG 配置模式，使器件按照 MODE 设置值重新加载比特流数据。用户也可以通过编写逻辑控制此管脚，从而按照自身需求触发器件进行重新配置；作为 GPIO 时，只能用作 output 类型。为保障配置过程顺利进行，用户复用 RECONFIG_N 管脚时需将其初始值置为高电平。</p>
READY	<p>inout 类型管脚。高电平有效，只有 READY 拉高时 FPGA 才能进行配置操作，READY 信号拉低后采用器件上电或触发 RECONFIG_N 的方式恢复状态。</p> <p>作为配置管脚时，若为 output 类型，可以指示 FPGA 当前能否进行配置：当器件具备配置条件时，READY 信号为高电平；若配置失败，则 READY 信号变为低电平。若为 input 类型，用户可通过自身逻辑或在器件外部人为拉低 READY 信号以延迟配置过程。</p> <p>作为 GPIO 时，可用作 input 或 output 类型。READY 用作 input 类型的 GPIO 时需保证配置开始前其初始值为 1，否则 FPGA 无法进</p>

管脚名称	功能描述
	行配置。
DONE	<p>inout 类型管脚。FPGA 配置成功的标志信号，配置成功后 DONE 信号拉高。</p> <p>作为配置管脚时，若为 output 类型，可以指示 FPGA 当前配置过程是否成功：当配置成功时，DONE 信号为高电平，器件进入工作状态；配置过程未完成或配置失败时，DONE 信号保持低电平状态。</p> <p>若为 input 类型，用户可通过自身逻辑或在器件外部人为拉低 DONE 信号以延迟其进入用户模式。RECONFIG_N 或 READY 保持低电平状态时，DONE 信号也会保持在低电平状态。使用 JTAG 电路配置 SRAM 的过程中，DONE 信号的值没有参考意义。</p> <p>作为 GPIO 时，可用作 input 或 output 类型。DONE 用作 input 类型的 GPIO 时需保证配置开始前其初始值为 1，否则配置结束后 FPGA 无法进入用户模式。</p>
MODE	<p>GowinCONFIG 配置模式选择信号。作为配置管脚时，类型为输入，内部弱下拉，最多可达 2-bit 位宽。FPGA 上电或低电平脉冲触发 RECONFIG_N 时，器件根据 MODE 值进入相应的 GowinCONFIG 状态，高云半导体每个系列的 FPGA 产品 MODE 值对应的配置模式略有不同。由于每个封装类型的管脚数目不同，有些器件的 MODE 管脚未完全封装出来，未封装出来的 MODE 值请参考对应的 pinout 手册，具体请参考相应器件的 PINOUT 手册。</p> <p>MODE 管脚作为 GPIO 时，可用作 input 或 output 类型。</p> <p>需要注意的是，当 MODE 值改变时，需要重新上电或低电平触发 RECONFIG_N 才能生效。</p>
JTAGSEL_N	<p>作为配置管脚时，类型为输入，内部弱下拉。如果在云源软件中设置了 JTAG 管脚复用为 GPIO，则器件上电后进行一次成功的配置后 JTAG 管脚变为 GPIO，JTAG 配置功能失效，用户可通过拉低 JTAGSEL_N 进行恢复；如果用户未设置 JTAG 管脚复用，则 JTAG 配置功能一直可用。作为 GPIO 时，可用作 input 或 output 类型。</p> <p>注！</p> <p>JTAGSEL_N 管脚与 JTAG 配置的 4 个管脚（TCK、TMS、TDI、TDO）设置为 GPIO 时存在互斥关系：JTAGSEL_N 设置为 GPIO 时，JTAG 管脚只能作为配置管脚；JTAG 管脚设置为 GPIO 时，JTAGSEL_N 只能作为配置管脚。</p>
TCK	<p>作为配置管脚时，类型为输入。</p> <p>JTAG 配置模式的串行时钟输入管脚。作为 GPIO 时，可用作 input 或 output 类型。</p>
TMS	<p>作为配置管脚时，类型为输入，内部弱上拉。</p> <p>JTAG 配置模式的串行模式输入管脚。作为 GPIO 时，可用作 input 或 output 类型。</p>
TDI	<p>作为配置管脚时，类型为输入，内部弱上拉。</p> <p>JTAG 配置模式的串行数据输入管脚。作为 GPIO 时，可用作 input 或 output 类型。</p>
TDO	<p>作为配置管脚时，类型为输出。</p> <p>JTAG 配置模式的串行数据输出管脚。作为 GPIO 时，可用作 input</p>

管脚名称	功能描述
	或 output 类型。
SCLK	作为配置管脚时，类型为输入。 SSPI、SERIAL 和 CPU 配置模式的时钟输入管脚。作为 GPIO 时，可用作 input 或 output 类型。
CLKHOLD_N	作为配置管脚时，类型为输入，内部弱下拉。 SSPI 和 CPU 配置模式的时钟锁定管脚：在 SSPI 模式下，高电平有效；在 CPU 模式下，低电平有效。作为 GPIO 时，可用作 input 或 output 类型。
SSPI_CS_N	作为配置管脚时，类型为输入，内部弱上拉。SSPI 配置模式的片选信号，低电平有效。作为 GPIO 时，可用作 input 或 output 类型。
SI	作为配置管脚时，类型为输入。SSPI 配置模式的串行数据输入管脚。作为 GPIO 时，可用作 input 或 output 类型。
SO	作为配置管脚时，类型为输出。SSPI 配置模式的串行数据输出管脚。作为 GPIO 时，可用作 input 或 output 类型。
QSSPI_WPN	作为配置管脚时，类型为输出。QSSPI 配置模式的串行数据输出管脚。作为 GPIO 时，可用作 input 或 output 类型。
FASTRD_N	作为配置管脚时，类型为输入。 MSPI 配置模式读取 SPI Flash 速度选择信号：当 FASTRD_N 为高电平时为普通读取模式（指令 0x03）；当 FASTRD_N 为低电平时为高速读取模式，各个厂家的 Flash 高速读取操作指令不同，具体请参考相应 Flash 的数据手册。作为 GPIO 时，可用作 input 或 output 类型。
WE_N	作为配置管脚时，类型为输入。 CPU 配置模式的读写使能信号选择管脚：当 WE_N 为高电平时表示读操作；当 WE_N 为低电平时表示写操作。作为 GPIO 时，可用作 input 或 output 类型。
D0~D7	inout 类型管脚。 CPU 配置模式的数据输入输出管脚，8-bit 位宽。根据 WE_N 的值确定 D0~D7 的输入输出方向。作为 GPIO 时，可用作 input 或 output 类型。
DIN	作为配置管脚时，类型为输入，内部弱下拉。 SERIAL 配置模式的串行数据输入管脚。作为 GPIO 时，可用作 input 或 output 类型。
DOUT	作为配置管脚时，类型为输出。 SERIAL 配置模式的串行数据输出管脚，只在 FPGA 级联时用作后一个器件的输入。作为 GPIO 时，可用作 input 或 output 类型。
SCL	作为配置管脚时，类型为 input。作为 GPIO 时，只可作 input 类型。
SDA	作为配置管脚时，类型为 in/out。作为 GPIO 时，可用作 input 或 output 类型。

5 配置模式详述

高云半导体 FPGA GW2AN-18X/9X 产品属于内置 Flash 的非易失器件。基于 SRAM 工艺的器件掉电后器件内部的配置数据丢失，每次上电后需重新配置；内置 Flash 的非易失器件掉电后数据仍然存储在芯片内部，重新上电时可以通过自启动配置或双启动配置的方式由器件自动进行重新配置。

高云半导体 FPGA 产品封装类型丰富，每一种封装的器件支持的配置模式与封装出来的配置管脚数目有关：所有器件均支持通用的 JTAG 配置；每种配置模式的 MODE 值有所差异。

5.1 配置须知

上电及配置流程

当 FPGA 的 VCC、VCCIO、VCCX 供电电压满足最小供电幅值时，FPGA 进入启动流程：电压稳定且 RECONFIG_N 未被外部电路拉低>FPGA 内部电路拉低 READY 和 DONE 管脚>FPGA 初始化>READY 拉高并采样 MODE 值>根据配置模式读取配置数据并校验>FPGA 唤醒>DONE 拉高>进入用户模式。

FPGA 启动过程中需要保持供电稳定，FPGA 上电、初始化、以及配置过程中 RECONFIG_N 管脚不允许出现低电平，用户可选择将 RECONFIG_N 管脚悬空或外部上拉。从器件上电复位释放到唤醒前，所有 GPIO 为高阻态，内部弱下拉。

高云半导体 FPGA GW2AN-18X/9X 产品按照配置数据的存储和指令的作用位置分为对 SRAM 的操作和对内置 Flash 的操作。

SRAM 操作

对 SRAM 的操作包括读取器件 ID CODE 和 USER CODE，读取器件状态寄存器信息以及 SRAM 配置。器件 ID 验证是配置操作的前提，只有 ID 验证成功的器件才能进行配置；USER CODE 是为方便用户对 ID CODE 相同的多个器件加以区分进行的编号标识；器件的状态寄存器记录着 FPGA

配置前后的状态信息，用户可据此分析器件状态，状态寄存器的含义请参考表 5-8。SRAM 配置操作时需要注意，只有未设置安全位的比特流数据支持验证功能。设置了安全位的数据任何用户无法进行回读验证。

内置 Flash 操作

对内置 Flash 的操作包括擦除、编程和验证操作。可通过 JTAG、SSPI、QSSPI、I²C 等接口操作内置 Flash，JTAG 时钟速率不大于 65MHz，SSPI、QSSPI 时钟速率不大于 100MHz，I²C 时钟速率 100KHz~555KHz。

所有接口模式都支持透明传输的特性，即器件支持在不影响现有工作状态的情况下编程内置 Flash 的操作，编程过程中器件可以按照原有的配置正常工作，编程完成后，低电平触发 RECONFIG_N 或发送 RECONFIG 指令即可完成在线升级。此特性适合应用于在线时间长但又需要不定期升级的场所。

配置管脚复用

用户在使用不同配置模式时，需要根据配置管脚的作用，确保 FPGA 工作在已选择的配置模式下。用户端管脚数目不足时，可以通过其他连接方式灵活处理这些管脚，只保留数据传输相关的管脚即可。MODE[1:0]用来选择 GowinCONFIG 的编程配置模式，用户不需要改变模式时可以使用上拉或下拉的方式将其固定在特定的模式，上拉电阻推荐 4.7K，下拉电阻推荐 1K。

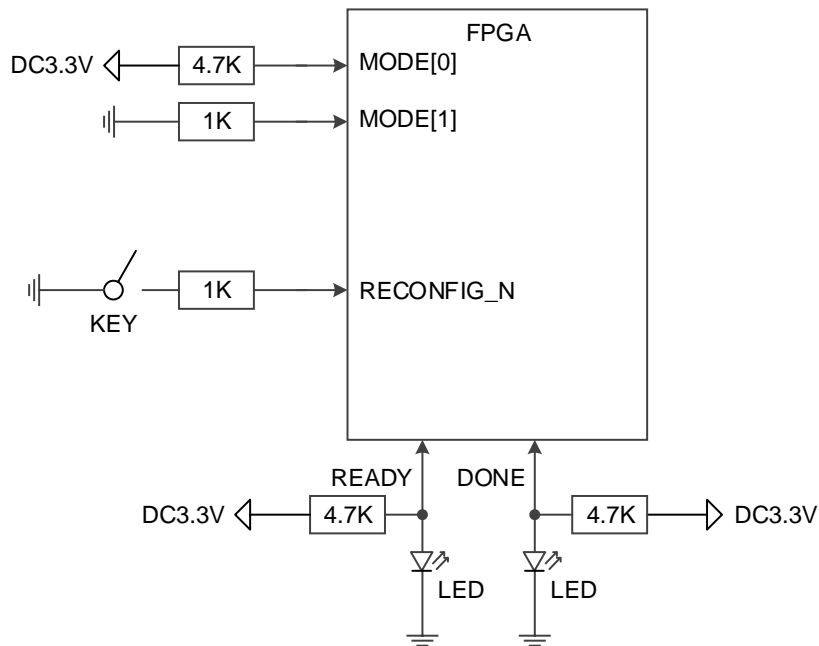
注！

RECONFIG_N、READY 和 DONE 管脚与每种配置模式相关，无论用户是否将它们设置为 GPIO，都需要保证在配置操作完成前它们的初始值或管脚连接状态满足编程配置条件。

固定管脚推荐接法

用户进行电路原理图设计时，推荐接法如图 5-1 所示。

图 5-1 固定管脚推荐接法



注!

- 用户需要改变 MODE 值时可增加拨码开关；部分器件 MODE 管脚未完全封装出来，未封装的 MODE 管脚请查阅 [UG972](#)、[GW2AN-18X 器件 Pinout 手册](#) 及 [UG978](#)、[GW2AN-9X 器件 Pinout 手册](#)；
- JTAG 配置过程中 READY 和 DONE 信号的值没有参考意义；
- RECONFIG_N、READY 和 DONE 未封装出来的管脚已内部处理，不影响配置功能。

重新上电及低电平脉冲触发 RECONFIG_N 时序图

重新上电和低电平脉冲触发 RECONFIG_N 的时序图如图 5-2 和图 5-3 所示。

图 5-2 重新上电时序图

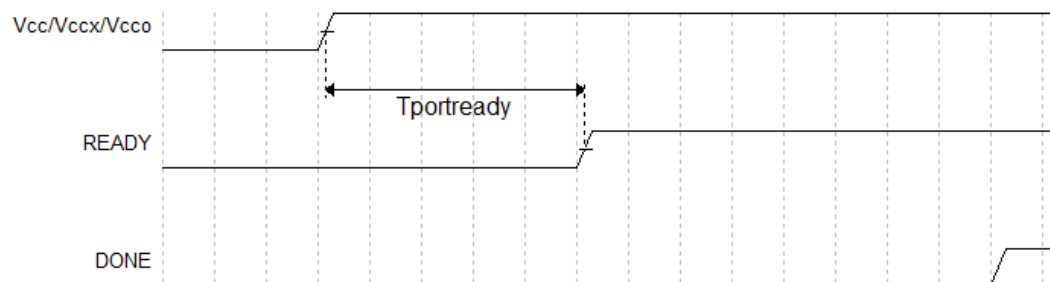
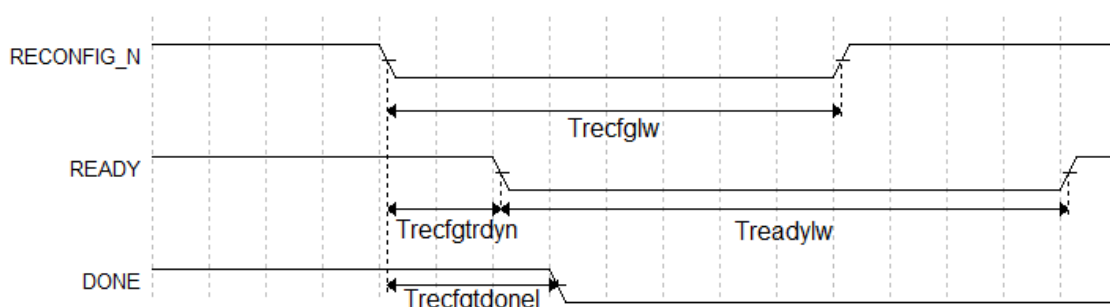


图 5-3 触发时序图



晨熙(Arora)家族 FPGA 产品相关的时序参数如表 5-1 所示。

表 5-1 高云半导体 FPGA GW2AN-18X/9X 产品重新上电和 RECONFIG_N 触发时序参数

参数名称	参数含义	最小值	最大值
$T_{portready}(MODE0=0)$	POR 到 READY 的上升沿的时延 (Time from POR to the rising edge of READY)	-	35ms
$T_{portready}(MODE0=1)$	POR 到 READY 的上升沿的时延 (Time from POR to the rising edge of READY)	-	10ms
$T_{recfglw}$	RECONFIG_N 低电平脉冲宽度 (RECONFIG_N low pulse width)	25ns	-
$T_{recfgtrdyn}$	RECONFIG_N 下降沿到 READY 变低电平的时延 (Time from RECONFIG_N falling edge to READY low)	-	70ns
$T_{readylw}$	READY 低电平脉冲宽度 (READY low pulse width)	TBD	-
$T_{recfgtdonel}$	RECONFIG_N 下降沿到 DONE 变低电平的时延 (Time from RECONFIG_N falling edge to DONE low)	-	80ns

5.2 AUTO BOOT 配置

AUTO BOOT 配置模式，是高云半导体针对非易失 FPGA 产品的瞬时接通特性推出的一种配置模式。自启动模式下，芯片上电后无需连接外部配置接口，FPGA 即可自行从内置 Flash 的地址 0x000000，以 Quad SPI 协议读取比特流数据完成配置。GW2AN-18X/9X 系列 FPGA 支持两次自启动配置，即上电后自启动配置失败时，器件可以自动从 0x100000 地址进行第二次配置操作。导致配置失败的因素包括 ID 验证错误，CRC 校验错误、指令错误、超时错误等。

用户使用自启动模式时，首先需要将配置数据编程到 FPGA 的内置 Flash 中，当系统重新上电或低电平脉冲触发 RECONFIG_N 管脚时，芯片会自动读取比特流数据完成配置过程。内置 Flash 的瞬时接通特性为配置过程节约了下载时间，提高了工作效率。

AUTOBOOT 模式需要将 MODE[1:0] 设置为 “01”，若配置失败，器件

依旧可以使用 JTAG、I²C、SSPI、QSSPI 等协议配置 SRAM 或 Flash。

5.3 JTAG 配置

高云半导体 FPGA 产品的 JTAG 配置模式符合 IEEE1532 标准和 IEEE1149.1 边界扫描标准。

JTAG 配置模式是将比特流数据写入到高云半导体 FPGA 产品的 SRAM 中，掉电后配置数据丢失。高云半导体所有封装的 FPGA 产品均支持 JTAG 配置模式。

5.3.1 JTAG 配置模式管脚

JTAG 配置模式的相关管脚如表 5-2 所示。

表 5-2 JTAG 配置模式管脚定义

管脚名称	I/O 类型	说明
JTAGSEL_N	I,内部弱下拉	将 JTAG 管脚从 GPIO 恢复为配置管脚，低电平有效
TCK ¹	I,内部弱下拉	JTAG 串行时钟输入
TMS	I,内部弱上拉	JTAG 串行模式输入
TDI	I,内部弱上拉	JTAG 串行数据输入
TDO	O,内部弱上拉	JTAG 串行数据输出

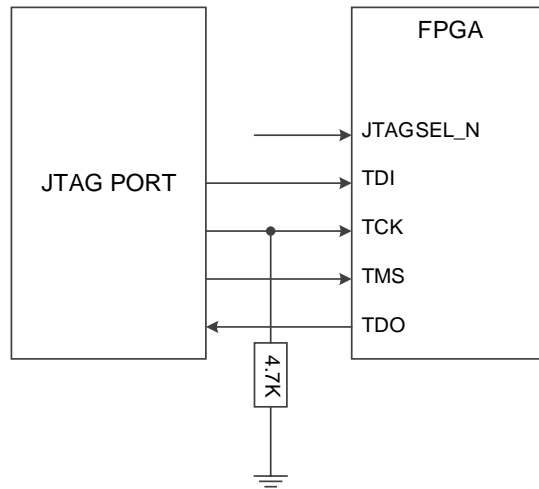
注！

[1] TCK 需在 PCB 上连接 4.7K 下拉电阻。

5.3.2 JTAG 配置模式连接示意图

JTAG 配置模式器件连接关系如图 5-4 所示。

图 5-4 JTAG 配置模式连接示意图



注！

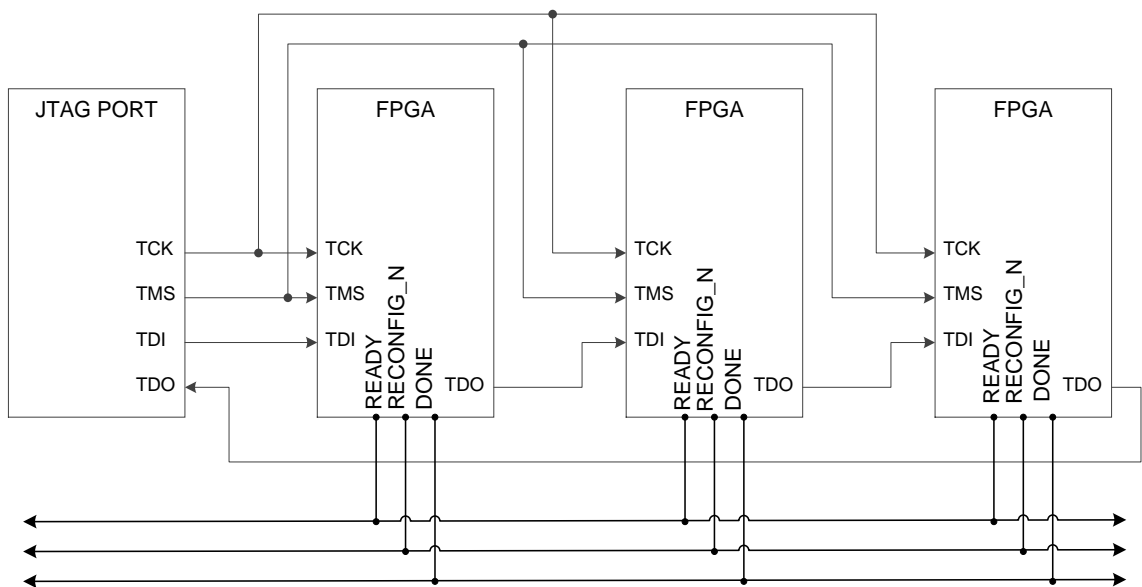
JTAG 配置模式时钟频率不能高于 65MHz。

高云半导体 FPGA 产品支持 JTAG 菊花链操作，即，把一个 FPGA 的 TDO 管脚连接到下一个 FPGA 的 TDI 管脚，Gowin 编程软件会自动识别连接在一起的 FPGA 器件，依次进行配置。菊花链配置的连接示意图如图 5-5 所示。

注！

若要编程菊花链上的 GW2AN-18X/9X 产品内置 Flash，需选择用 GAO-Bridge (sFlash Erase,Program,Verify thru GAO-Bridge)的方式，详情请参考 [SUG502, Gowin Programmer 用户指南](#)。

图 5-5 JTAG 菊花链配置模式连接示意图



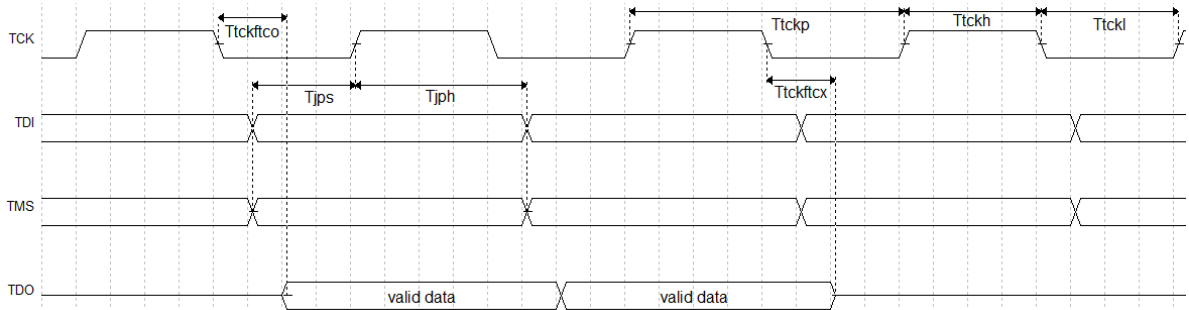
注！

DONE、RECONFIG_N 及 READY 信号视情况决定是否连接。

5.3.3 JTAG 配置模式时序图

JTAG 配置模式的时序图如图 5-6 所示。

图 5-6 JTAG 配置模式时序图



图中各个参数的含义如表 5-3 所示。

表 5-3 JTAG 配置模式时序参数

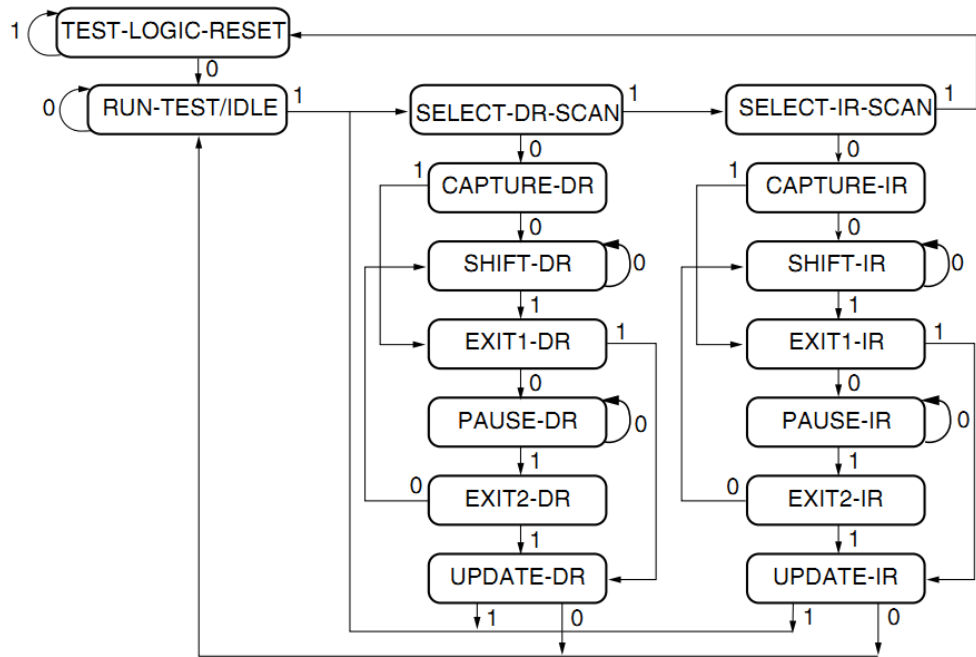
参数名称	参数含义	最小值	最大值
$T_{tckftco}$	TCK 下降沿到输出数据时延 (Time from TCK falling edge to output)	-	10ns
$T_{tckftcx}$	TCK 下降沿到输出高阻时延 (Time from TCK falling edge to high impedance)	-	10ns
T_{tckp}	TCK 时钟周期 (TCK clock period)	40ns	-
T_{tckh}	TCK 时钟高电平时间 (TCK clock high time)	20ns	-
T_{tckl}	TCK 时钟低电平时间 (TCK clock low time)	20ns	-
T_{jps}	JTAG PORT 建立时间 (JTAG PORT setup time)	10ns	-
T_{jph}	JTAG PORT 保持时间 (JTAG PORT hold time)	8ns	-

5.3.4 JTAG 相关配置流程

TAP 状态机

测试访问口状态机旨在选择指令寄存器或数据寄存器，使其连接至 TDI 和 TDO 之间。一般来说，指令寄存器用于选择需扫描的数据寄存器，在状态机框图中，位于箭头一侧的数字表示 TCK 变高时 TMS 的逻辑状态，如图所示。写一个 JTAG 指令后，要在 Run-Test-Idle 状态保持至少 3 个时钟周期。

图 5-7 TAP 状态机



TAP 复位

通过保持 TMS 为高电平（逻辑“1”）并在 TCK 端输入至少 5 个选通脉冲（变高后再变低）后，复位 TAP 逻辑，从而实现将处于其它状态的 TAP 状态机转换成测试逻辑复位状态，对 JTAG 接口和测试逻辑复位。

注！

该状态不复位 CPU 和外设。

注！

- 在进入 Shift_DR 或 Shift_IR 状态时，TDO 上的数据从 TCK 的下降沿开始有效；
- 在进入 Shift_DR 或 Shift_IR 状态时，数据不移位；
- 在离开 Shift_DR 或 Shift_IR 时，数据被移位；
- 最先移出的是数据的最低位 LSB；
- 一旦复位，所有指令将被重置或失效。

指令寄存器和数据寄存器

除测试逻辑复位外，状态机亦可控制两个基本操作：

- 指令寄存器（IR）扫描；
- 数据寄存器（DR）扫描。

在指令寄存器扫描操作中，在 Shift_IR 状态时，传送数据或指令给指令寄存器，发送时采用 LSB 的方式，低数据位首先被发送，回到 Run-Test-Idle 后指令即被发送完毕，如图 5-8 所示。

在数据寄存器扫描操作中，在 Shift_DR 状态时，传送数据或指令给数据寄存器，如图 5-9 所示。数据发送采用 LSB 还是 MSB 取决于具体操作。

图 5-8 指令寄存器访问时序

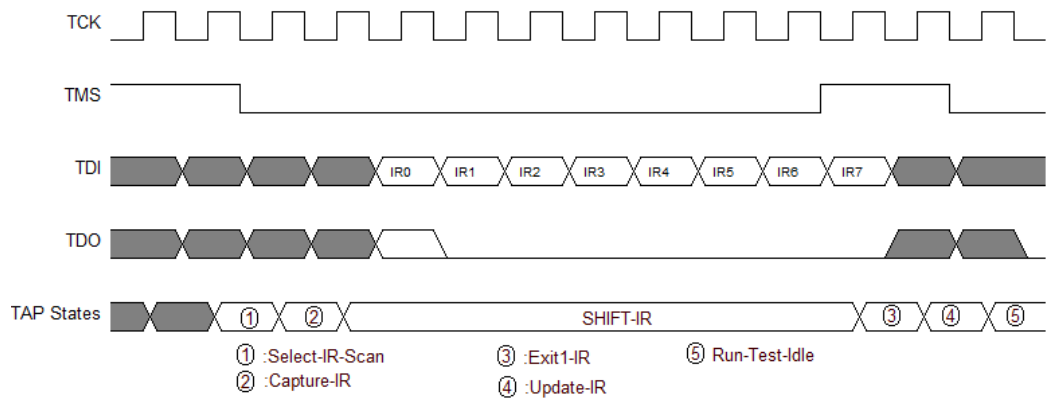
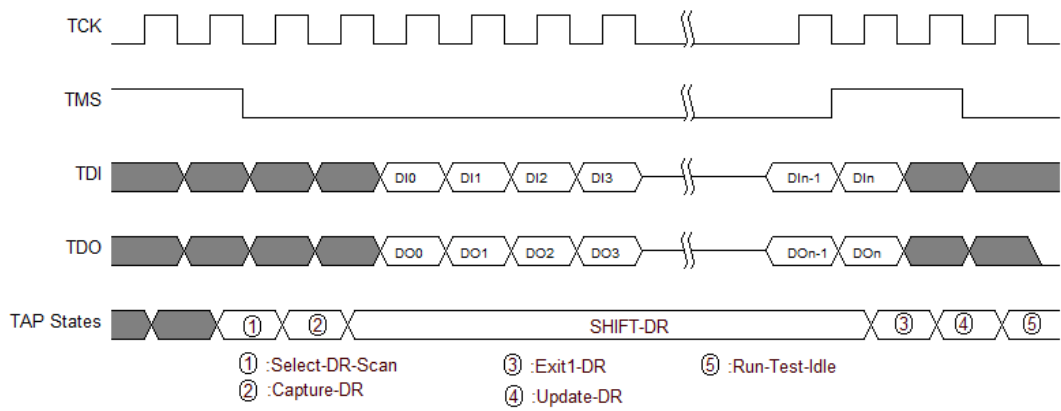


图 5-9 数据寄存器访问时序



注！

- 指令寄存器的总长度为 8 位；
- 根据所选择的寄存器，数据寄存器的长度可变化。

读取 ID CODE 实例

ID Code 即 JEDEC ID Code，是 FPGA 器件的一个基本标识。

高云 FPGA ID Code 长度为 32 位，下表列出了高云部分 FPGA 的 ID Code。

表 5-4 Gowin 部分 FPGA 产品 ID CODE

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

读取 FPGA 的指令是 0x11，以下步骤以读取 GW2AN-9X ID Code 为例说明 JTAG 的工作方式。

1. TAP 复位：TMS 置为高电平，连续发送至少 5 个时钟周期；
2. 移动状态机从 Test-Logic-Reset 到 Run-Test-Idle；
3. 移动状态机到 Shift-IR，从最低位开始发送 Read ID 指令 0x11，最高位（最后一位）发送的同时移动状态机到 Exit1-IR，即最高位发送前 TMS 要置于高电平，表 5-5 给出 8 个时钟周期内发送 0x11 过程中 TDI 和 TMS 的值变化，时序如图 5-11 所示。

表 5-5 发送指令过程中 TDI 和 TMS 的值变化

	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. 移动状态机，从 Exit1-IR 经过 Update-IR 后回到 Run-Test-Idle，并在 Run-Test-Idle 运行至少 3 个时钟周期；
5. 移动状态机到 Shift-DR，发送 32 个时钟周期，并在第 32 个时钟发送前，置 TMS 为高电平，完成 32 个时钟周期的同时，跳出 Shift-DR 到 Exit1-DR。这期间，发送 32 个时钟即可读出 32bits 数据，即为 0x0000581B，如图 5-12 所示；
6. 移动状态回到 Run-Test-Idle。

图 5-10 读取 ID Code 状态机流程图

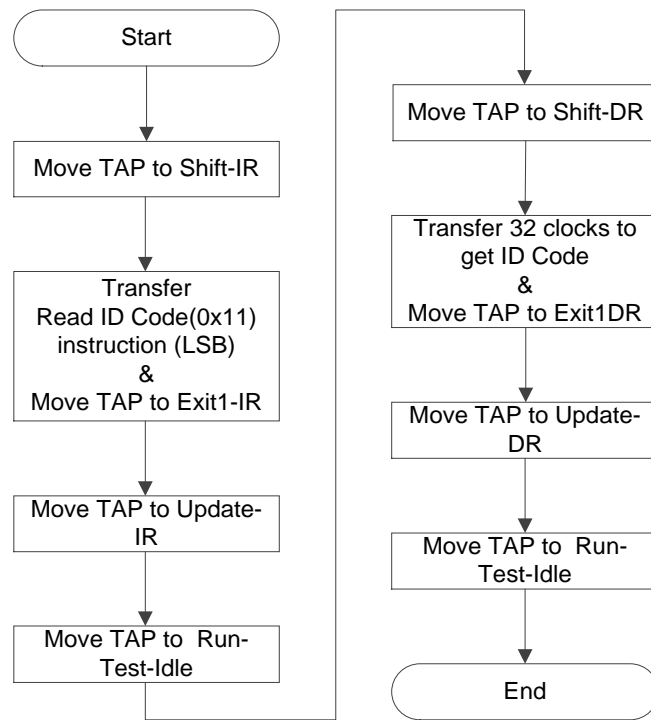


图 5-11 读取 ID Code 指令-0x11 访问时序

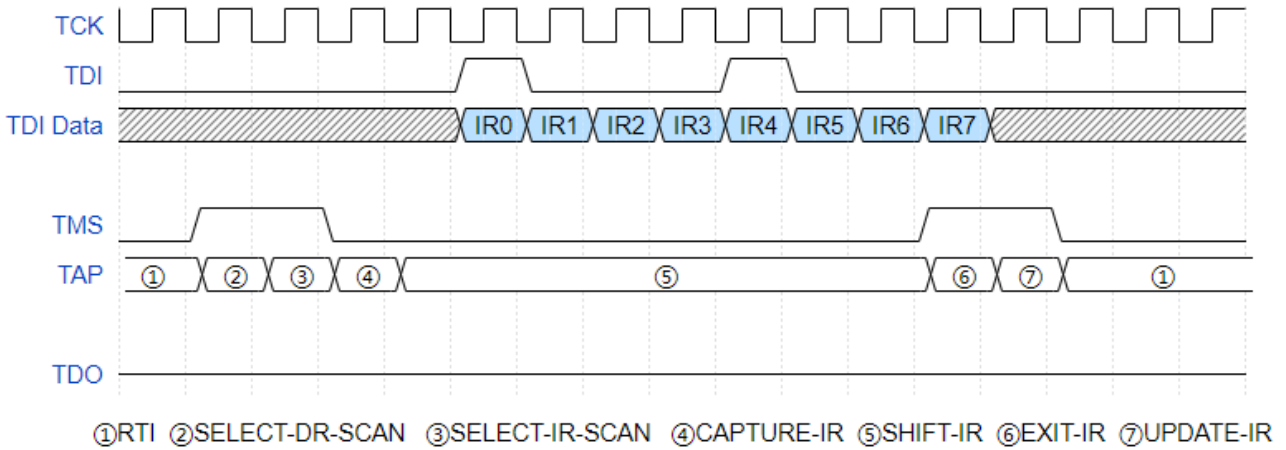
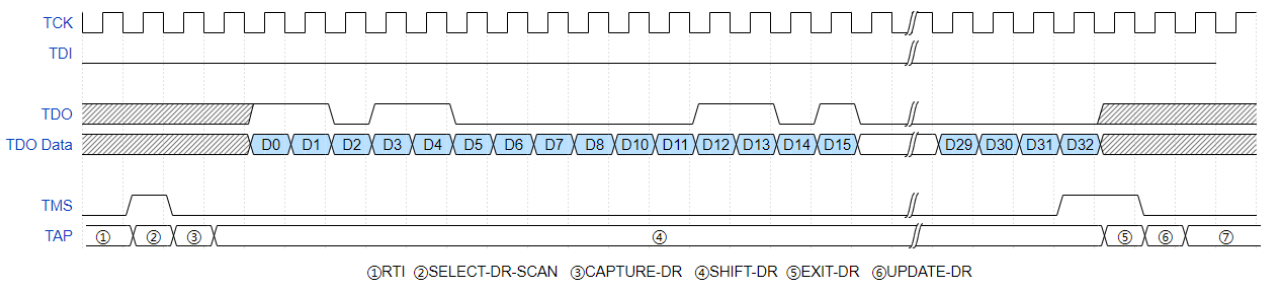


图 5-12 读取 ID Code(0x0000581B)数据寄存器访问时序



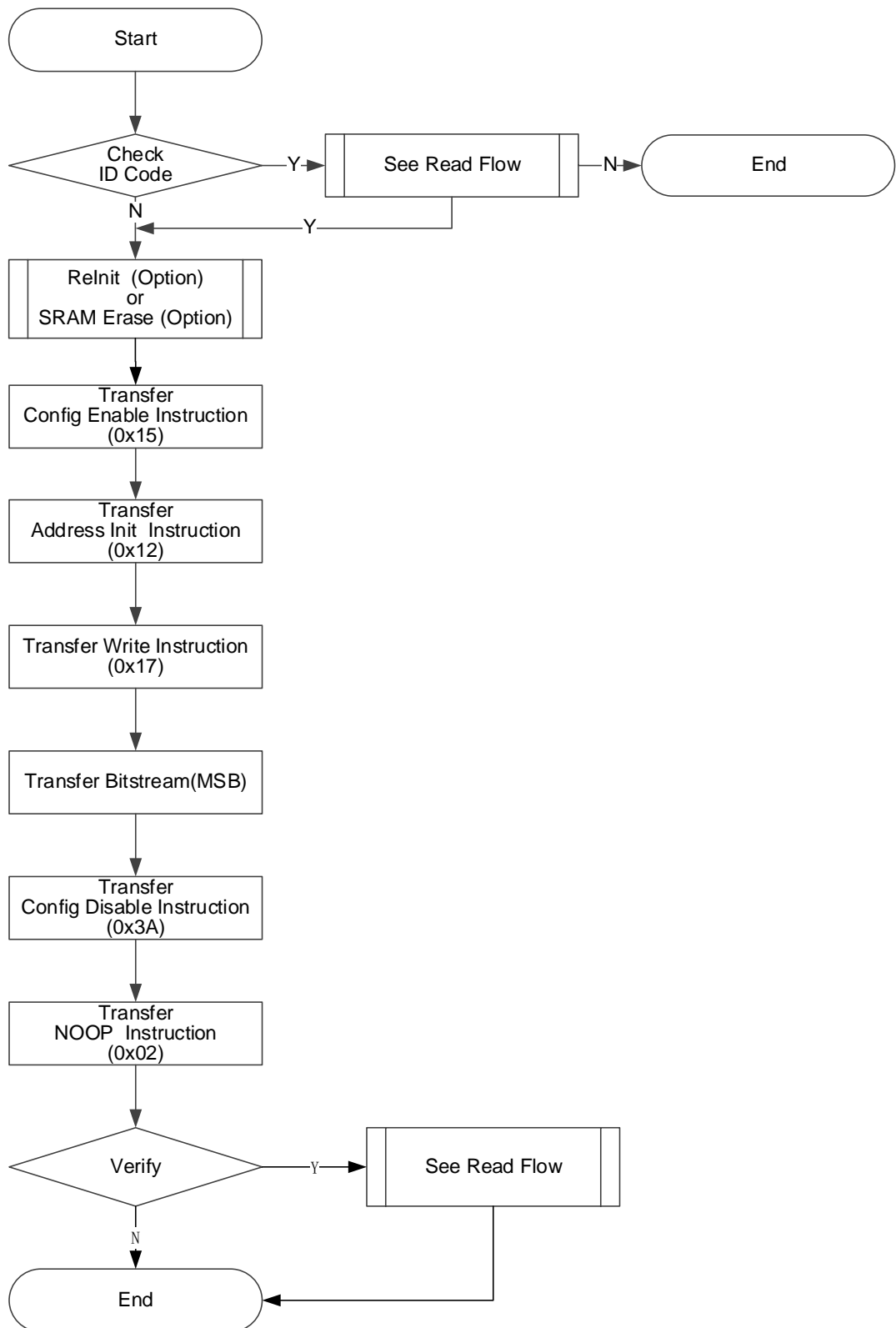
配置 SRAM 的流程

通过外部 Host 配置 FPGA SRAM，使 FPGA 实现功能，通过 JTAG 配置 SRAM 不受 Configuration Mode Pins 的影响。SRAM 写完后等待 60ms，以待 status code 刷新。

通过 Gowin software 设计软件生成数据流文件，利用 JTAG 实现 SRAM 的配置，下面介绍外部 Host 配置 SRAM 的过程，如图 5-13 所示。

1. 建立 JTAG 链路，TAP 复位；
2. 读取设备 ID Code，检查 ID CODE 是否匹配；
3. 读取设备 StatusCode，如 Ready 位为 0，发送 Reinit 指令 0x3F；或者当 SRAM 已被配置，擦除 SRAM，流程参考“[擦除 SRAM 的流程](#)”；
4. 发送 ConfigEnable 指令 0x15。
5. 发送 Address Initialize 指令 0x12；
6. 发送 Transfer Configuration Data 指令 0x17。
7. 移动状态到 Shift-DR（数据寄存器），将 Bitstream Data 从最高位开始（MSB），逐位发送，发送全部数据流文件内容，并回到 Run-Test-Idle 状态；
8. 发送 ConfigDisabled 指令 0x3A；
9. 发送 Noop 指令 0x02，结束配置流程。
10. 如需回读 Configuration Data 进行校验，请参考“[读取 SRAM 的流程](#)”。

图 5-13 配置 SRAM 流程



读取 SRAM 的流程

警告：处于数据保密设计，SRAM 数据默认不被允许回读。
从 FPGA 的 SRAM 区域读取 SRAM 数据，首先应保证写入 SRAM 时

未配置安全位（**Security Bit**），安全位是用于保护运行时数据，保证数据安全。安全位完成设置后，从 **SRAM** 取回的数据均为 1（高电平）。

在加载过程中，**FPGA** 对写入数据进行 **CRC** 校验，以确保数据写入正确，**CRC** 是否报错，可以作为配置 **SRAM** 的校验机制。

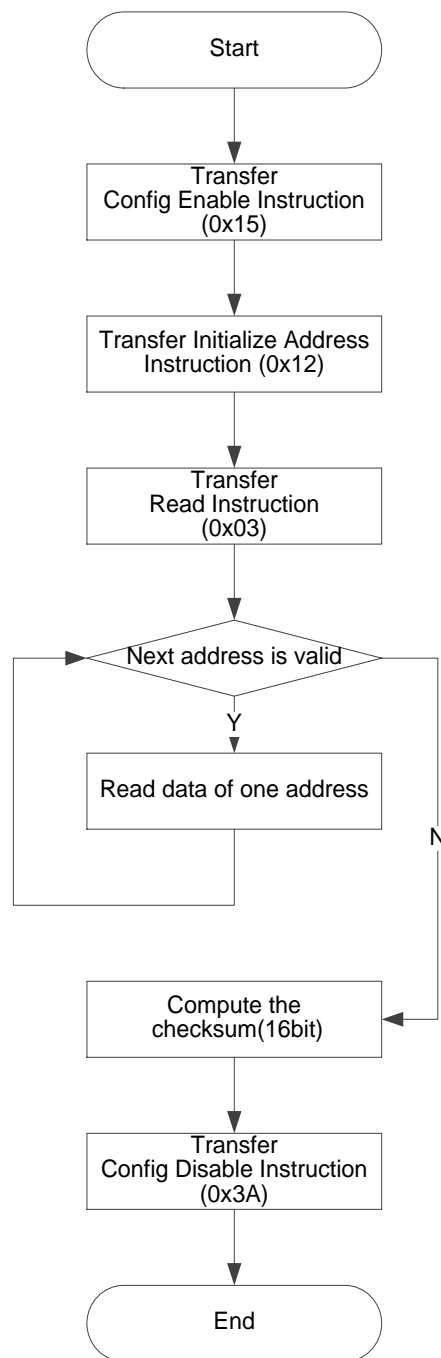
表 5-6 器件 SRAM 地址数量和地址长度

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

下面详细介绍读取流程，如图 5-14 所示。

1. 发送 **ConfigEnable** 指令 0x15；
2. 发送 **Address Initialize** 指令 0x12；
3. 发送 **SRAM Read** 指令 0x03；
4. 移动状态机到 **Shift-DR**（数据寄存器），发送地址长度数量的时钟，请参见表 5-6。在发送最后一个时钟同时拉高 **TMS**，跳到 **Exit1-DR**，此时 **TDO** 读取相应长度的数据。最后回到 **Run-Test-Idle**；
5. 重复步骤 4，每次读取一个地址的数据，其地址会自动累加；
6. 发送 **ConfigDisabled** 指令 0x3A；
7. 发送 **Noop** 指令 0x02，结束读取流程；

图 5-14 读取 SRAM 的流程



擦除 SRAM 的流程

当重新配置 SRAM 时，需要擦除已存在的 SRAM。流程如下：

1. 发送 ConfigEnable 指令 0x15；
2. 发送 SRAM Erase 指令 0x05；
3. 发送 Noop 指令 0x02；
4. 延时或 Run Test 2~10ms；
5. 发送 SRAM Erase Done 指令 0x09；
6. 发送 ConfigDisabled 指令 0x3A；

7. 发送 Noop 指令 0x02，结束流程；

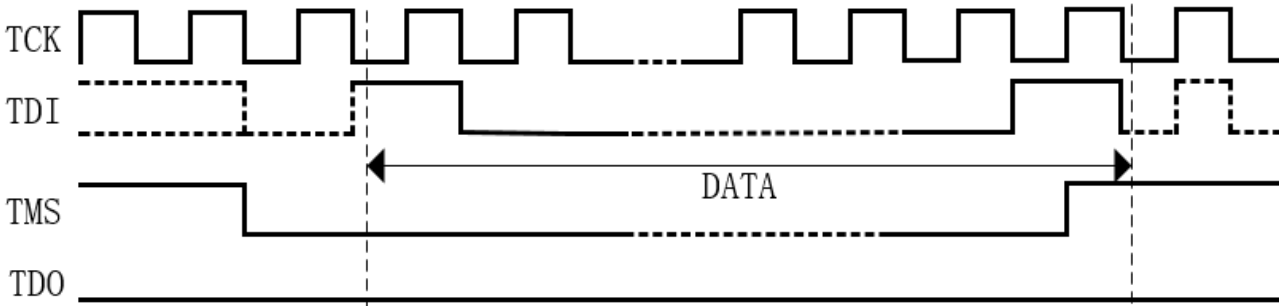
注！

- 在发送 EraseSram (0x05) 指令、Noop (0x02) 之后，要给足够的时间等待其擦除完毕；
- GW2AN-18X/9X 参考时间为 6ms；

内置 Flash 编程模式

高云半导体 FPGA GW2AN-18X/9X 产品内置 16 Mbit Serial Flash 存储器，JTAG 提供类 SPI 协议用来编程内置 Flash，最大操作频率支持 65MHz。

JTAG 提供的类 SPI 协议，总体与标准 SPI 逻辑一致，TMS 对应 ChipSelect (/CS) 信号，TDI 对应 DI 信号，TCK 对应 SerialClock (CLK) 信号，TDO 对应 DO 信号。时序也参考标准 SPI 对应时序，仅 TDI 数据向后位移一个时钟周期。当 TMS 拉低后，在发送一个 Colck 之后，才是有效的 DATA，并且 DATA 的最后一个 BIT 要与 TMS 一起拉高，示意图如下：

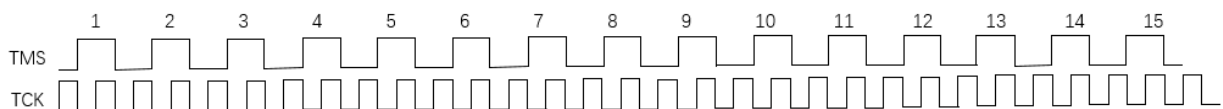


启用 Flash 编程模式

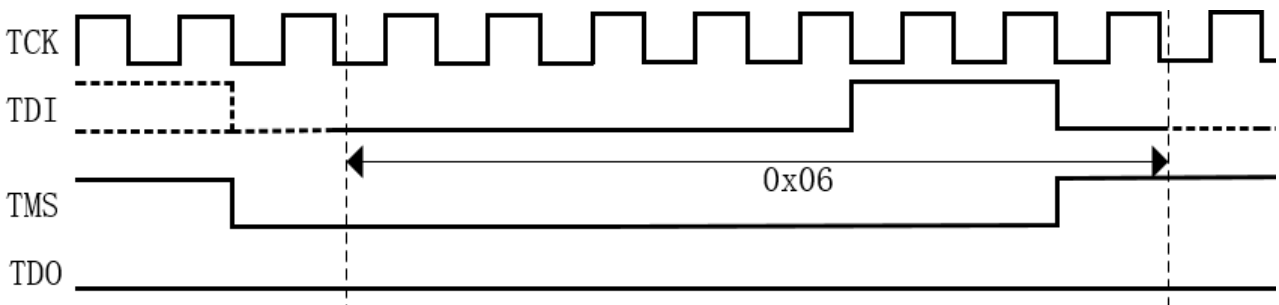
在编程 Flash 之前，需要将 JTAG 接口接通内部 Flash 控制器，发送标准 JTAG 指令 0x16，即可启用该模式，启用之后，JTAG 接口仅工作在 Flash 编程模式下，转为类 SPI 协议，不再支持 JTAG 指令。

退出 Flash 编程模式

通过设置连续的 15 次 TMS 高低电平，即可退出 Flash 编程模式，即退出类 SPI 接口，恢复到 JTAG 标准接口，恢复 JTAG 协议对器件进行配置的模式。如下图：



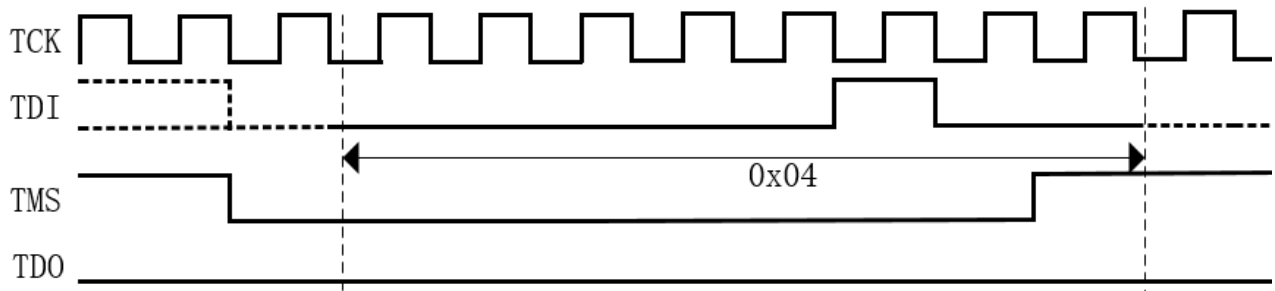
内置 Flash 指令-WriteEnable (0x06)



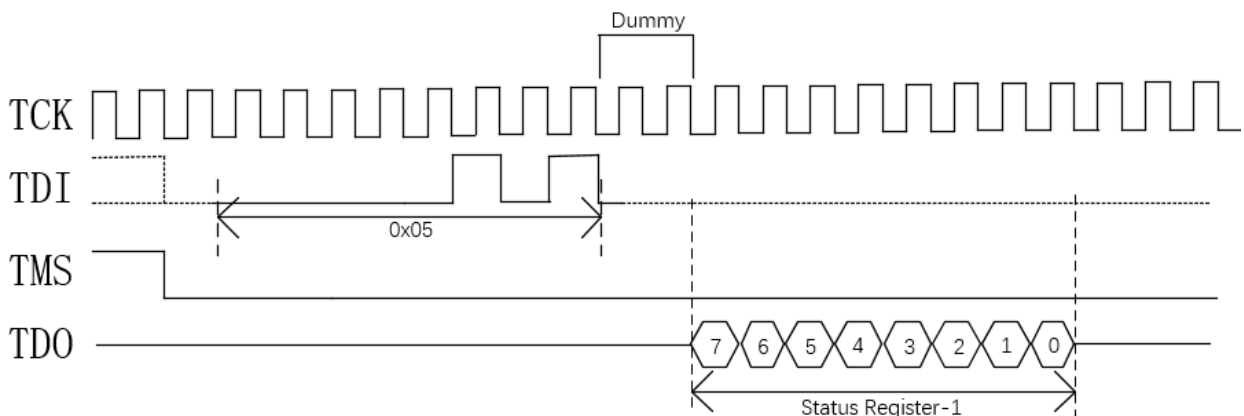
WriteEnable 指令（上图）用来设置 Flash Status Register 的 Write Enable Latch (WEL) 位，WEL 位必须在每次 Page-Program、Sector Erase、Chip Erase 前设置。在 TMS 为低电平并在一个时钟周期 (TCK) 后，将指令“0x06”移入数据输入 (TDI) 引脚上，0x06 的最后一个 bit 位“0”跟随 TMS 在同一个时钟周期内完成。

内置 Flash 指令-WriteDisable (0x04)

WriteDisable 指令用于重置 Flash Status Register 的 Write Enable Latch (WEL) 位，在 TMS 为低电平并在一个时钟周期 (TCK) 后，将指令“0x04”移入数据输入 (TDI) 引脚上，0x04 的最后一个 bit 位“0”跟随 TMS 在同一个时钟周期内完成。



内置 Flash 指令-Read Status Register-1 (0x05)



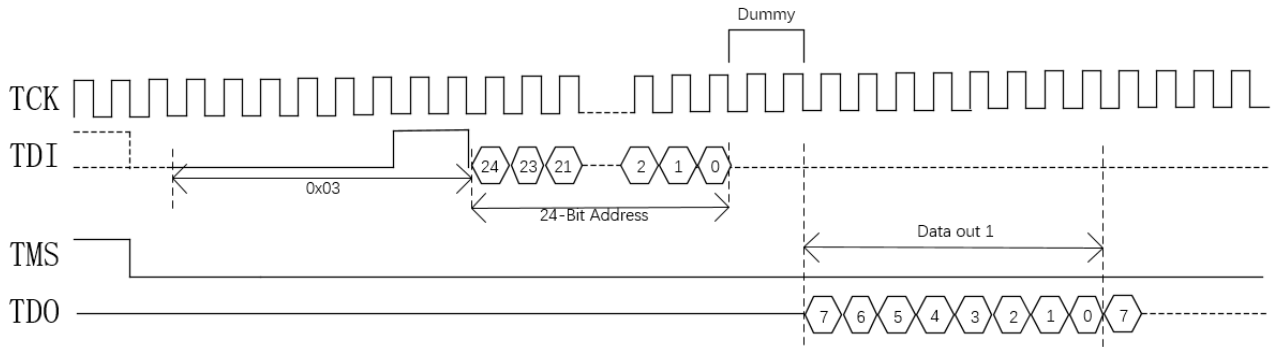
Read Status Register-1 指令用于读取一个 8bit 状态寄存器，在 TMS 为低电平并在一个时钟周期 (TCK) 后，将指令“0x05”在 TCK 上升沿时候移入数据输入 (TDI) 引脚，再经历 2 个 dummy clock 后，Status Register-1 的数据将在 TCK 的下降沿从 TDO 管脚送出，最高有效位

(MSB) 在前。如下图。

Read Status Register-1 共 8 位，用 S[7:0]表示，其中 S[0]表示 BUSY 位，当 Flash 处于 Page-Program、Chip Erase、Sector Erase 等过程时，S[0]将被自动设置位 1。这个状态下 Flash 处于 BUSY 状态，不响应其他指令，当操作结束后，S[0]将自动重置为 0，可以继续发送其他指令。

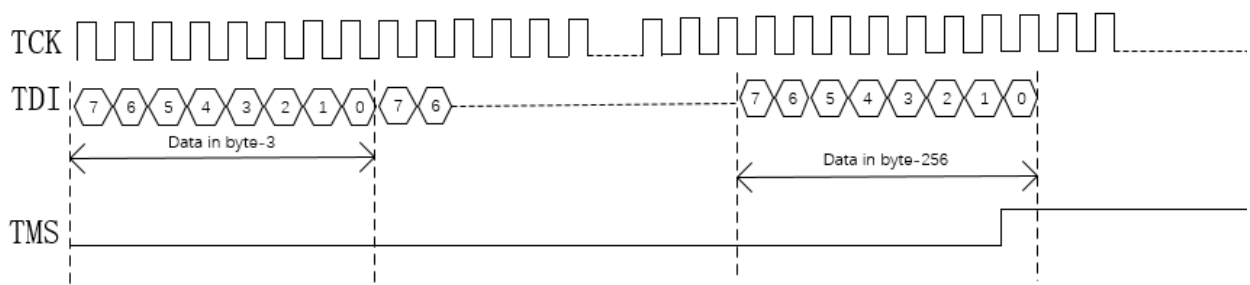
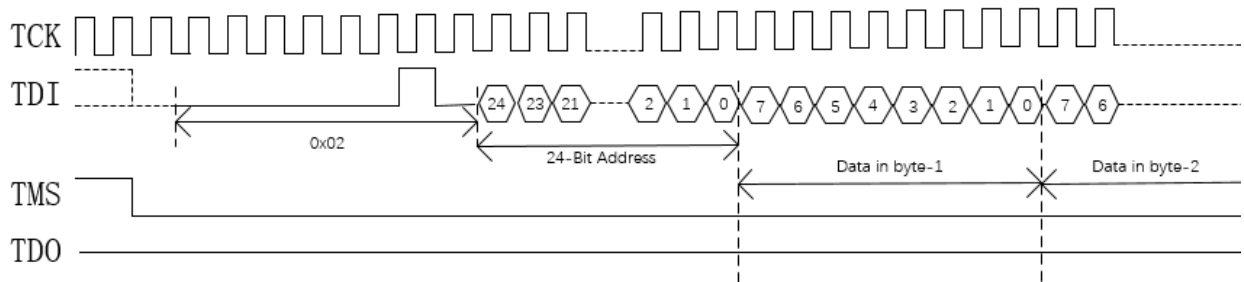
内置 Flash 指令-Read Data (0x03)

Read Data 指令用于从 Flash 中读取连续的数据，在 TMS 为低电平并在一个时钟周期 (TCK) 后，将指令 “0x03” 在 TCK 上升沿时移入数据输入 (TDI) 引脚，然后将 24 位地址移入数据 (DI) 引脚，再经历 2 个 dummy clock 后，对应地址的数据将在 TCK 的下降沿从 TDO 管脚送出，最高有效位 (MSB) 在前。当回读的数据超过一个地址时，地址会自动递增，也就是说一条指令可以读取整个 Flash 的数据。如下图。



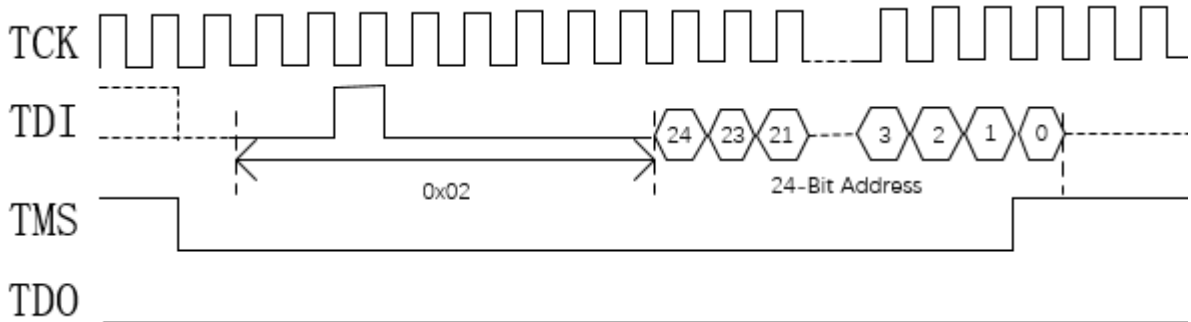
内置 Flash 指令-Page Program (0x02)

Page Program 指令用于烧录一个 page 上的一个或多个已被擦除 (0xFF) 的字节。在配置之前必须完成 WriteEnable 指令。指令在 TMS 为低电平并在一个时钟周期 (TCK) 后，将指令 “0x02” 在 TCK 上升沿时候移入数据输入 (TDI) 引脚，然后将 24 位地址移入数据 (DI) 引脚，然后按 MSB 方式将数据移入数据 (DI) 引脚。TMS 在此期间保持低电平，并在最后一个字节的最后一个 bit 位数据跟随 TMS 拉高时在同一个时钟周期内移入 DI 引脚。如下图：



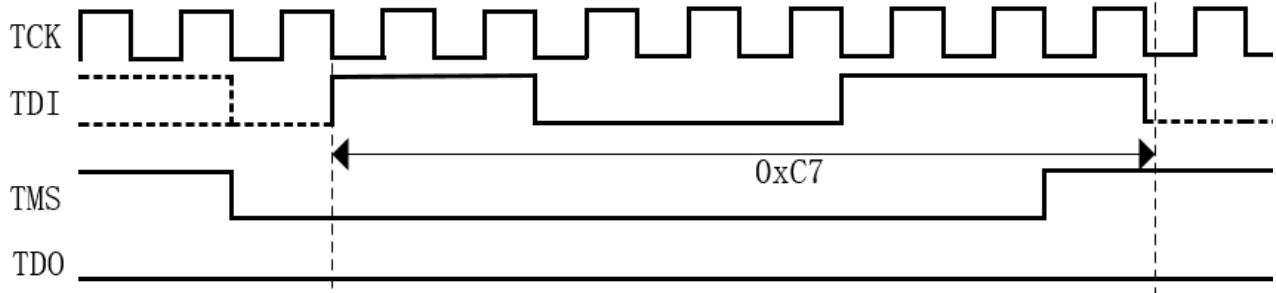
内置 Flash 指令-Sector Erase (0x20)

Sector Erase 指令用于擦除指定扇区 (4Kbytes) 内所有数据, 擦除完成后 Flash 数据恢复到 0xFF 状态。在发送该指令之前, 必须完成 WriteEnable 指令。指令在 TMS 为低电平并在一个时钟周期 (TCK) 后, 将指令 “0x20” 在 TCK 上升沿时候移入数据输入 (TDI) 引脚, 然后将 24 位地址移入数据 (DI) 引脚, 在地址的最后一个 bit 时, 同时将 TMS 设置为高电平。

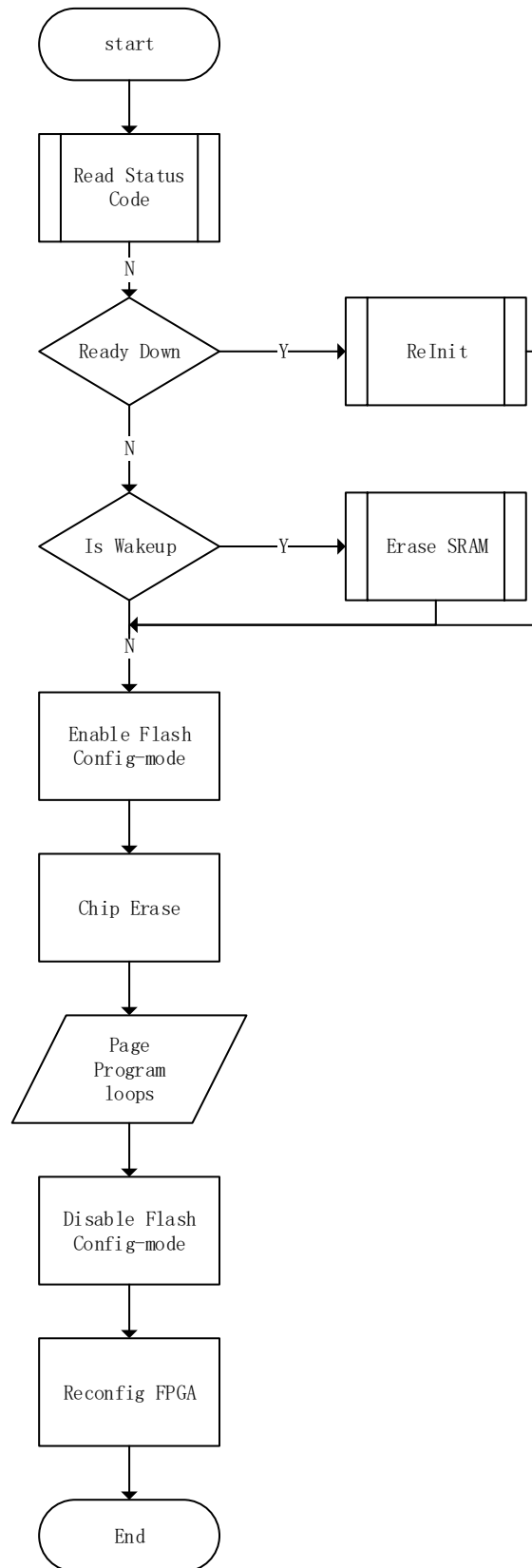


内置 Flash 指令-Chip Erase (0xC7/0x60)

Chip Erase 指令将擦除 Flash 内所有数据, 擦除后所有数据为 1 (0xFF)。在发送该指令之前, 必须完成 WriteEnable 指令。指令在 TMS 为低电平并在一个时钟周期 (TCK) 后, 将指令 “0xc7” 在 TCK 上升沿时候移入数据输入 (TDI) 引脚, 0xC7 的最后一个 bit 位 “1” 跟随 TMS 在同一个时钟周期内完成。



配置内置 Flash 流程图



JTAG Boundary Scan 模式烧录 SPI Flash

该模式的原理，是使用 Boundary Scan 的方式改变与 SPI 相接管脚的状态来实现 SSPI 时序，从而编程内部 Flash。

该模式采用的 Boundary Scan Chain 长度为 8 位，每 2 位组合对应管脚的状态，如表 5-7 所示，每发送两次 Boundary Scan Chain 完成一次 SCLK 驱动。

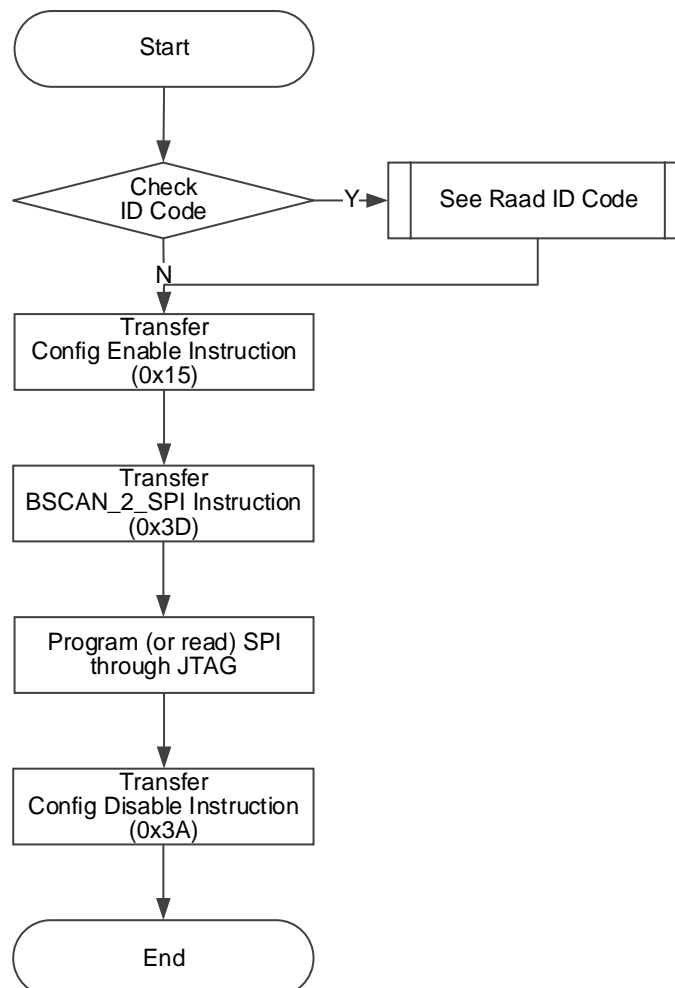
表 5-7 管脚状态

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

注！

- ctrl:0 表示输出，1 表示输入；
- data:0 表示低电平，1 表示高电平。

图 5-15 采用 Boundary Scan 模式编程 SPI Flash 流程示意图



读取 Status Register (0x41)

Status Register 在器件调试和观察器件状态有很大帮助，通过读取 Status Register，可初步判断器件的状态，如判断是否成功 wakeup、是否存在加载错误等。Status Register 共有 32 位，读取指令是 0x41，时序与 Read ID Code 一致。

Status Register 的含义如表 5-8 所示。

表 5-8 Status Register 与配置加载相关的条目

Device Status Register[31:0]	GW2AN-18X/9X
0	CRC Error (1 表示发生错误, 0 表示未发生错误)
1	Bad Command Error(1 表示发生错误, 0 表示未发生错误)
2	ID Verify Failed Error(1 表示发生错误, 0 表示未发生错误)
3	Timeout Error(1 表示发生错误, 0 表示未发生错误)
4	Autoboot2nd Failed Error(1 表示发生错误, 0 表示未发生错误)
5	Memory Erase
6	Preamble
7	Edit Mode
8	Program SPI Directly
9	Autoboot1st Failed Error(1 表示发生错误, 0 表示未发生错误)
10	Non-jtag Active
11	Bypass State
12	I ² C Flag
13	Done Final State
14	Security Bit Final State
15	Encryption Format (1 表示使用了加密的数据流文件)
16	Encryption Key Match (1 表示密钥正确, 0 表示密钥错误)
17	sspi_mode,
18-31	0

清除 status code 错误

在配置 SRAM 或烧录 Flash 前，要确保当前 status code 中没有错误，如果有错误，请按以下流程清除错误：

1. 依次发送指令 ISC-NOOP(0x02)，ISC-ENABLE(0x15)，reprogram(0x3C)，ISC-NOOP(0x02)。
2. 延时 100ms。
3. 依次发送指令 ISC-DISABLE(0x3A)，ISC-NOOP(0x02)。
4. 延时 100ms。

读取 User Code 0x13

User Code 共有 32 位，读取指令是 0x13，时序与 Read ID Code 一致。

User Code 默认使用的是 FS 文件的 checksum 值，可在 Gowin Designer 中重新定义。

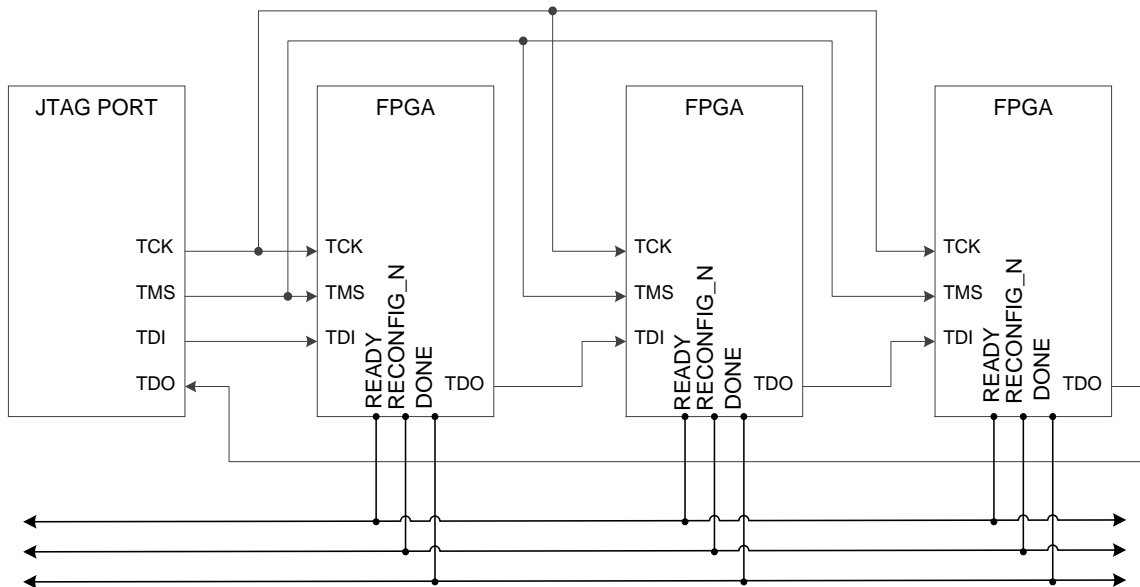
重加载 Reconfig 0x3C

该指令作用是使 FPGA 从 flash 中读取数据流文件，并配置到 SRAM。

通过 JTAG 依次发送 Reconfig (0x3C) 指令、Noop (0x02) 指令，可使器件重加载，效果同触发 Reconfig_N 管脚。

菊花链连接示意图

图 5-16 菊花链连接示意图



例程文件

例程文件，请联系公司技术支持或当地办事处。

5.4 SSPI 配置模式

SSPI (Slave SPI) 配置模式, 即 FPGA 作为从器件, 由外部 Host 通过 SPI 接口对高云半导体 FPGA 产品进行配置的过程。

5.4.1 SSPI 配置模式管脚

SSPI 模式相关的配置管脚如表 5-9 所示。

表 5-9 SSPI 配置模式管脚

管脚名称	I/O 类型	说明
RECONFIG_N	I, 内部弱上拉	低电平脉冲: 启动 GowinCONFIG 配置
READY	I/O, 内部弱上拉	高电平: 当前可以对 FPGA 进行编程配置 低电平: 禁止对 FPGA 进行编程配置
DONE	I/O, 内部弱上拉	高电平: 成功完成编程配置 低电平: 未完成编程配置或编程配置失败
MODE[1:0]	I, 内部弱下拉	配置模式选择, READY 上升沿采样
SCLK	I, 内部弱上拉	输入时钟
CLKHOLD_N	I, 内部弱下拉	高电平有效
SO	O, 内部弱下拉	FPGA 输出数据到 Host
SI	I, 内部弱下拉	Host 向 FPGA 输入数据
SSPI_CS_N	I, 内部弱上拉	SSPI 片选信号, 低电平有效

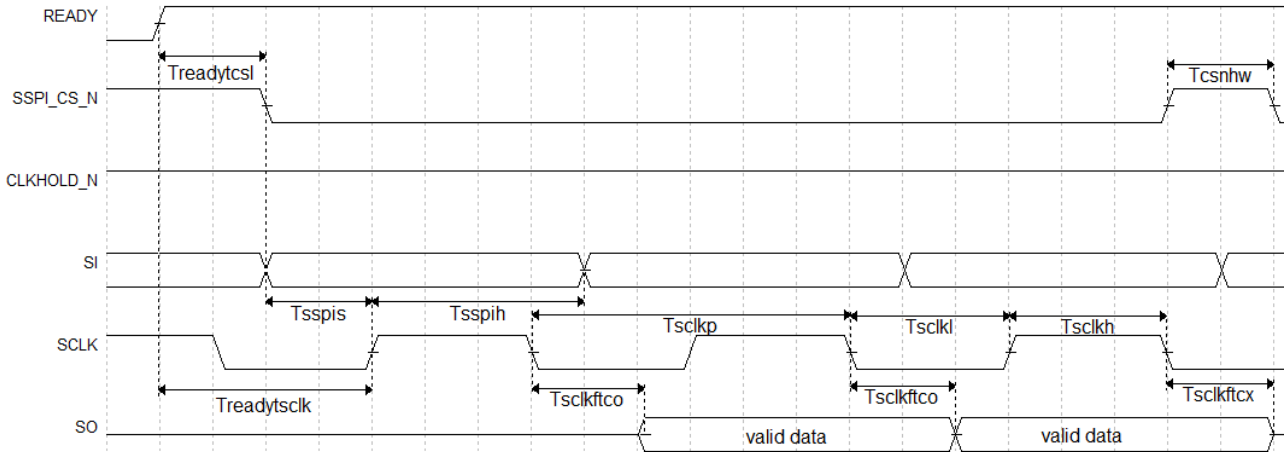
注!

CLKHOLD_N 默认状态为内部弱下拉, 使用 SSPI 时请配置 CLKHOLD_N 为高电平。

5.4.2 SSPI 配置模式时序图

SSPI 配置模式的时序图如图 5-17 所示。

图 5-17 SSPI 配置模式时序图



时序参数如表 5-10 所示。

表 5-10 SSPI 配置模式时序参数

参数名称	参数含义	最小值	最大值
T _{sclkp}	SCLK 时钟周期(SCLK clock period)	15ns	-
T _{sclkh}	SCLK 时钟高电平时间(SCLK clock high time)	7.5ns	-
T _{sckl}	SCLK 时钟低电平时间(SCLK clock low time)	7.5ns	-
T _{sspis}	SSPI PORT 建立时间(SSPI PORT setup time)	2ns	-
T _{sspih}	SSPI PORT 保持时间(SSPI PORT hold time)	0ns	-
T _{sckftco}	SCLK 下降沿到数据输出时延(Time from SCLK falling edge to output)	-	10ns
T _{sckftcx}	SCLK 下降沿到输出高阻时延(Time from SCLK falling edge to high impedance)	-	10ns
T _{csnhw}	CSN 高电平脉冲宽度 (CSN high time)	25ns	-
T _{readytcsi}	READY 上升沿到 CSN 低电平时间(Time from READY rising edge to CSN low)	TBD	
T _{readytsclk}	READY 上升沿到第一个 SCLK 沿时间(Time from READY rising edge to first SCLK edge)	TBD	-

除满足上电要求外，SSPI 模式对高云半导体 FPGA 产品进行配置，还需满足以下条件：

- SSPI 接口使能
上电后初次配置或前一次配置时 RECONFIG_N 未设置为普通 I/O 状态。
- 启动新的配置
重新上电或低电平脉冲触发 RECONFIG_N 管脚。

5.4.3 SSPI 配置指令

当 FPGA 处于 SSPI 模式时，通过 SSPI 可以烧录 FPGA SRAM 或者读取 ID CODE\USER CODE\STATUS CODE 等信息，也可以烧录外部存储设备（例如 SPI Flash）。

FPGA 的 SSPI 指令一般由 1-4 个字节组成，至少包含 1 个指令类字节和多个冗余信息字节，没有指定信息字节的情况下，冗余信息字节可以是任意数（下表用 0x00 表示）。

表 5-11 配置指令

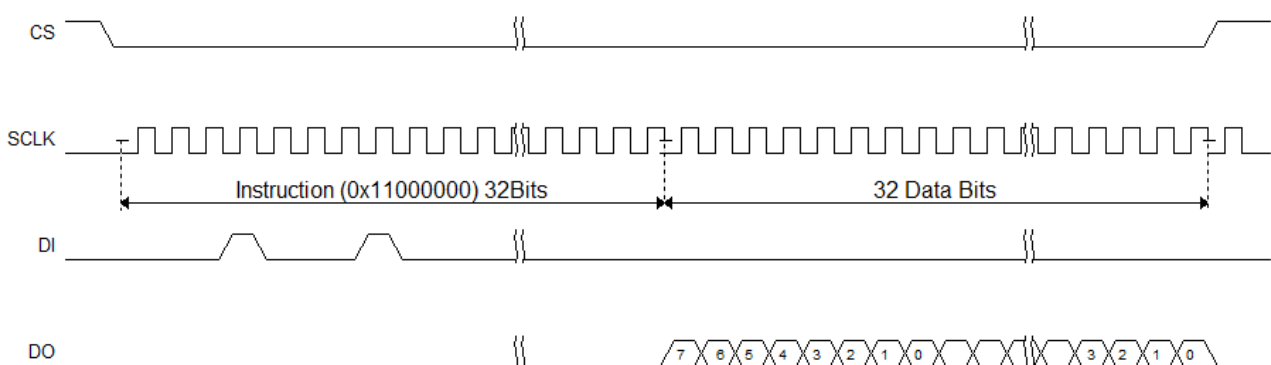
指令名称	完整指令（指令字节+冗余信息字节）
Read ID Code	0x11000000
Read User Code	0x13000000
Read Status Code	0x41000000
Reconfig/Reprogram	0x3C00
Write Enable	0x1500
Write Disable	0x3A00
Write Data	0x3B
Write Data with Quad SPI	0x6B
Program SPI Flash	0x1600
Init Address	0x1200
Erase SRAM	0x0500
Reinit	0x3F00

Read ID Code

FPGA 的 ID Code 长度是 32bits，读取 ID 的指令是四个字节，即 0x11000000。在指令发送之前，使 CS 处于高电平状态，并且要在此状态产生若干时钟（两个时钟以上），以驱动 FPGA 获取 CS 状态。

当 CS 拉低之后，数据以 MSB 的方式写入指令 0x11000000，在写完四字节的指令后要继续产生 32 个时钟，此时 ID CODE 数据将以 MSB 的形式从 DO 依次位移出来。

图 5-18 读取 ID Code 时序示意图

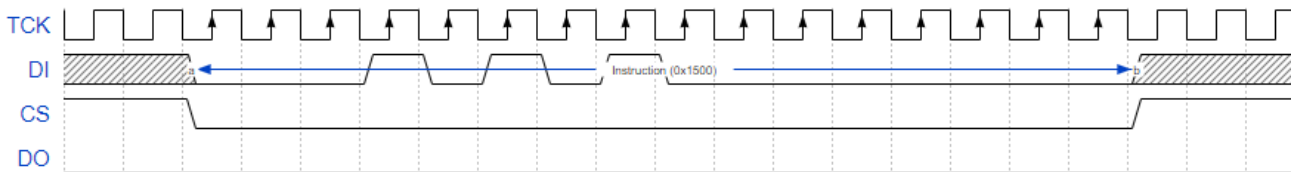


读取 StatusCode/UserCode 与读取 ID Code 的操作类似，替换对应的指令即可。

Write Enable (0x1500)

在配置 SRAM（写 Features）前，使用 Write Enable（0x15）指令进入设备编辑模式，使设备可以接受写数据 Write Data（0x3B）指令。

图 5-19 Write Enable (0x15) 时序示意图



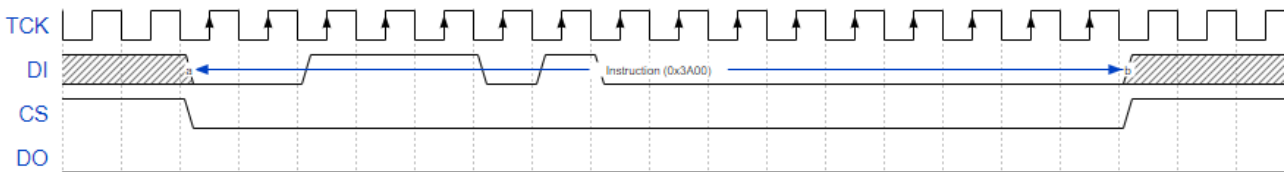
注！

SCLK 驱动规则，在 CS 高电平时，需要给予 SCLK 两个以上时钟，以驱动 FPGA 识别 CS 信号。发送其他指令也遵循这个规则。

Write Disable (0x3A00)

发送数据完成后，需使用 Write Disable 从编辑模式中退出。退出后可唤醒设备，使设备进入工作状态。

图 5-20 Write Disable (0x3A00) 时序示意图



像上述两个指令，0x1500 和 0x3A00 指令的时序基本一致，指令都是在 CS 低电平后开始，并在指令传输完成后拉高 CS，遵循这种时序的指令有 0x3C00（Reconfig/Reprogram）、0x1500（Write Enable）、0x3A000（Write Disable）、0x1600（Program SPI Flash）、0x1200（Init Address）、0x0500（Erase SRAM）。

另外，需要注意的是，因为 SSPI 是由外部时钟驱动，这些指令前后 CS 处于高电平时，需要 2 个以上的时钟以使 FPGA 可以采集到 CS 状态。

Erase SRAM(0x0500)

指令时序与 WriteEnable/WriteDisable 一致，仅替换指令内容为 0x0500。

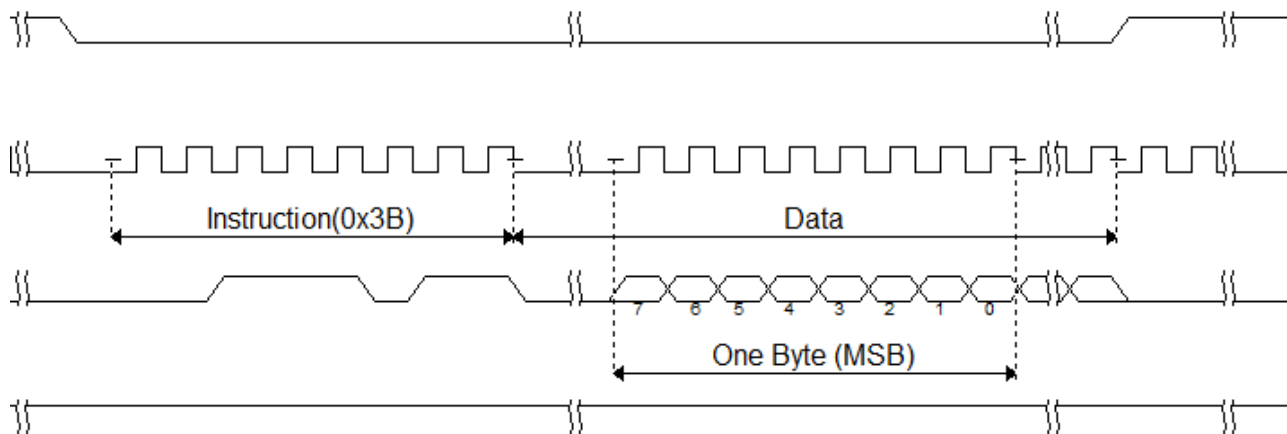
当指令发送后，需延迟至少 10ms 以使指令执行完毕。

Write Data (0x3B)

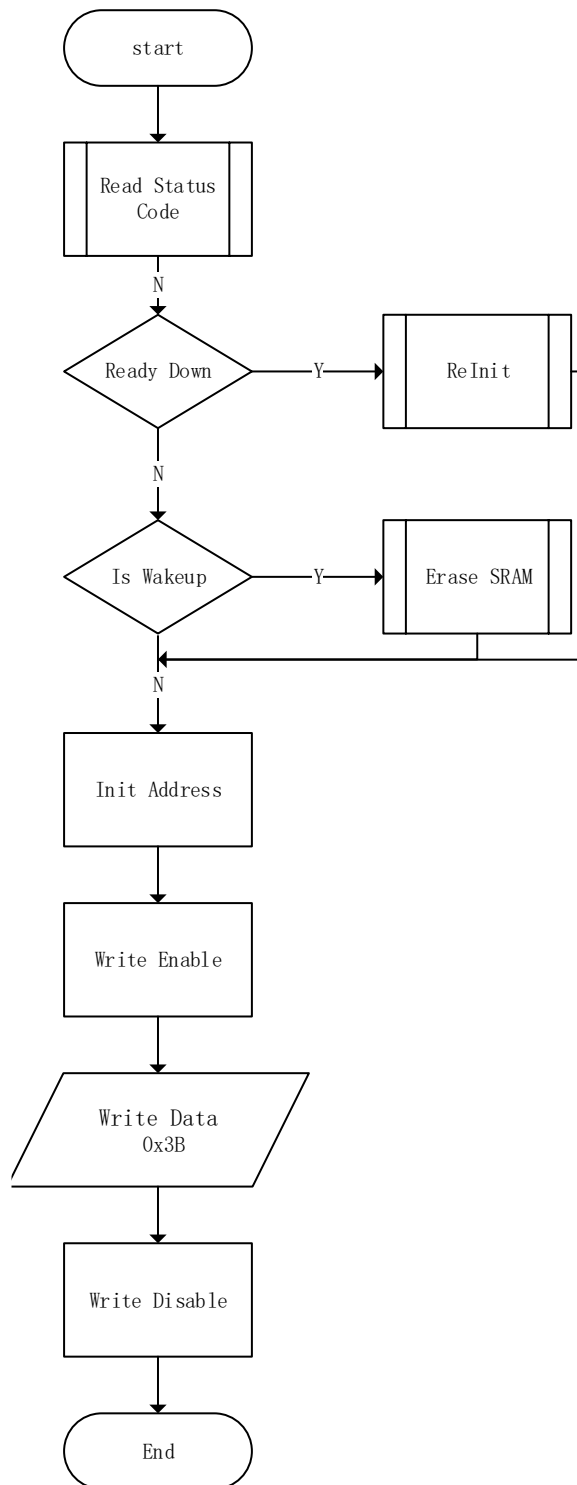
通过使用 WriteData (0x3B) 指令直接向 FPGA 设备发送数据流文件。

注意，在数据写入过程中，CS 一直处于低电平。

图 5-21 Write Data (0x3B) 时序示意图



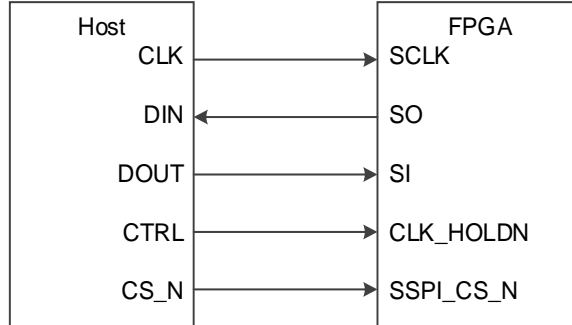
5.4.4 SSPI Configure SRAM 流程图



5.4.5 SSPI 配置模式连接示意图

使用 SSPI 配置模式对高云半导体 FPGA 产品配置的连接示意图如图 5-22 所示。

图 5-22 SSPI 配置模式连接示意图



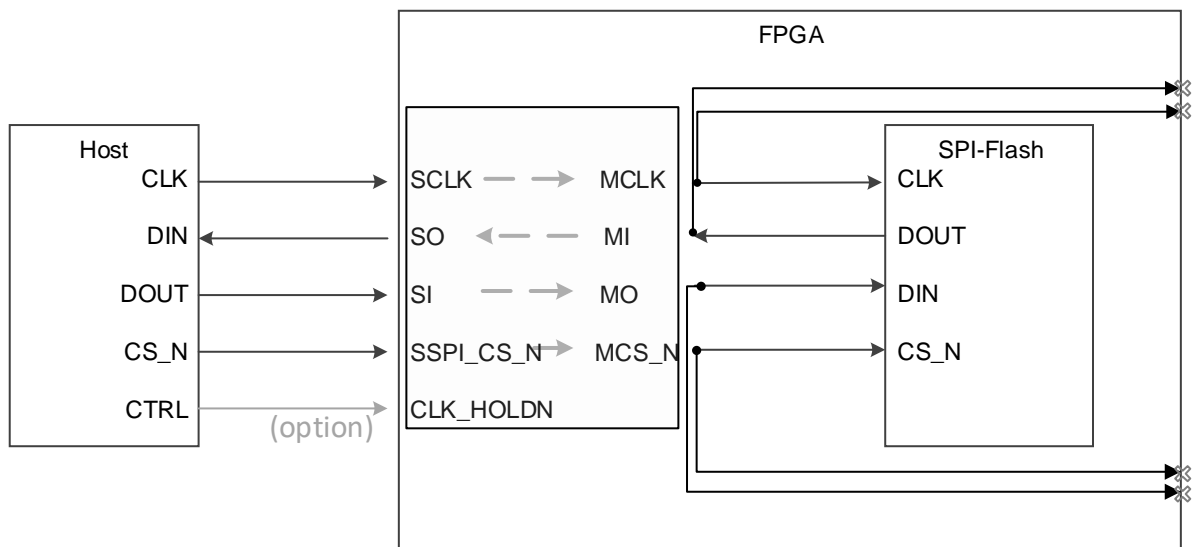
注!

- 此图为 SSPI 配置模式的最小系统图，其他固定管脚的接法请参考图 5-1。
- CLKHOLD_N 默认状态为内部弱下拉，使用 SSPI 时请配置 CLKHOLD_N 为高电平。

SSPI 配置管脚除了进行常规的 SRAM 配置操作外，还可编程 GW2AN9X/18X 内置的 SPI Flash，编程 Flash 操作的 MODE 值与 SSPI 配置模式的 MODE 值相同，用户可以在 Gowin 编程软件中选择将配置数据写入 SRAM 或内置 Flash。需要从内置 Flash 加载前，需要将 MODE 值调整为 MSPI MODE，之后通过重新上电或触发 RECONFIG_N 触发 MSPI 加载。

SSPI 接口编程内置 Flash 的连接示意图如图 5-23 所示。

图 5-23 SSPI 编程内置 Flash 连接示意图



烧录流程示意图见图 5-29。首先通过 SSPI 向 FPGA 发送指令

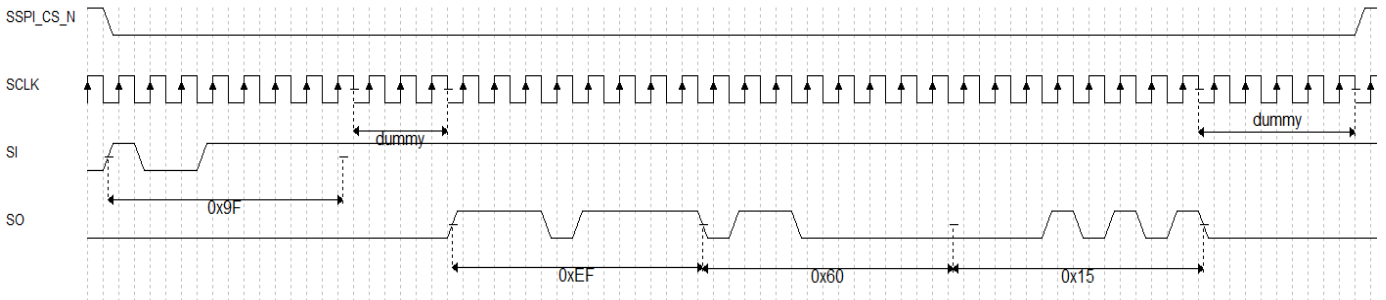
“Program SPI Flash” (0x1600) 指令，完成后 FPGA 可以转发 SSPI 到 Flash，Host 端的 SSPI 可以直接访问 Flash，接下来就可以按照 Flash 的相关时序对其进行编程。

注意，从 Flash 读取数据时，回读的数据会延迟 3 个 Bit。例如 SSPI 读取 Flash 的 ID Code 时，需要额外发送 3 个时钟从而获取最后 3 位。

SSPI 读取内置 Flash ID Code

GW2AN9X/18X 批次不一样，内部 Flash 型号可能存在差异，Flash ID code 也存在差异，但是读取 Flash ID Code 的时序一致。通过 0x9F 指令读取 GW2AN9X 内置 Flash ID Code 的时序：

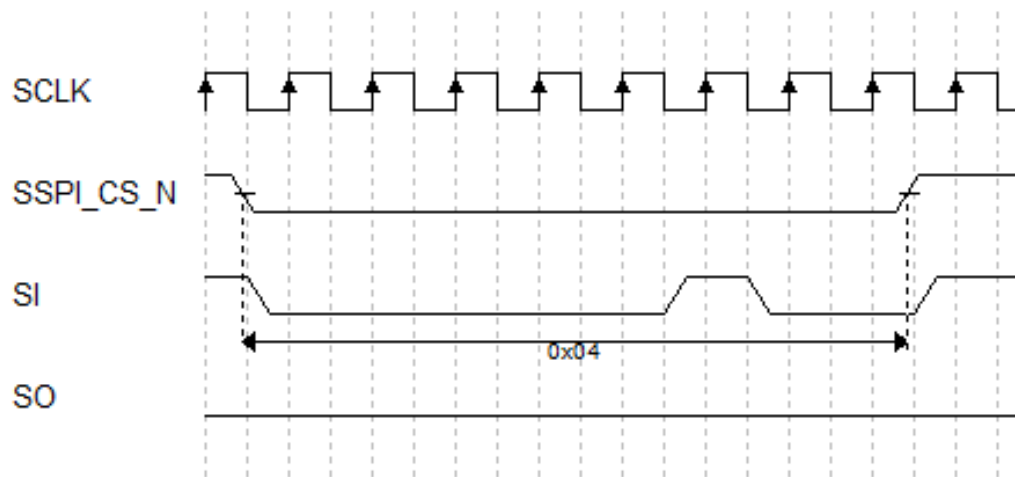
图 5-24 SSPI 读取内置 Flash ID Code 时序示意图



SSPI 写内置 Flash 指令

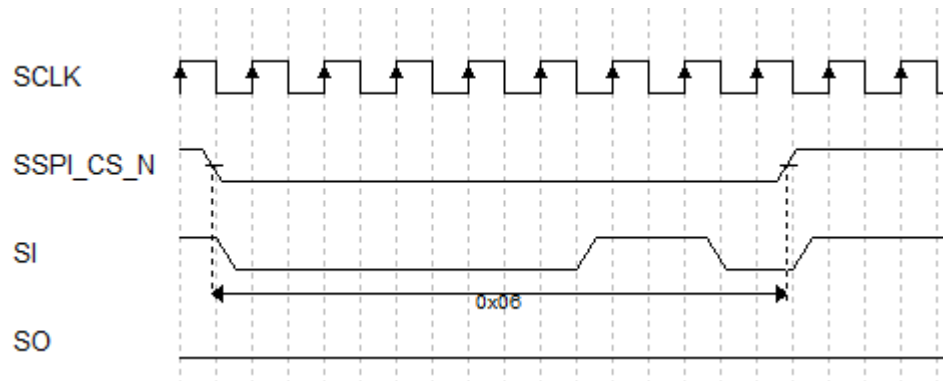
向内置 Flash 写 WriteDisable (0x04) 指令的时序：

图 5-25 SSPI 写 WriteDisable (0x04) 指令的时序示意图



向内置 Flash 写 WriteEnable (0x06) 指令的时序：

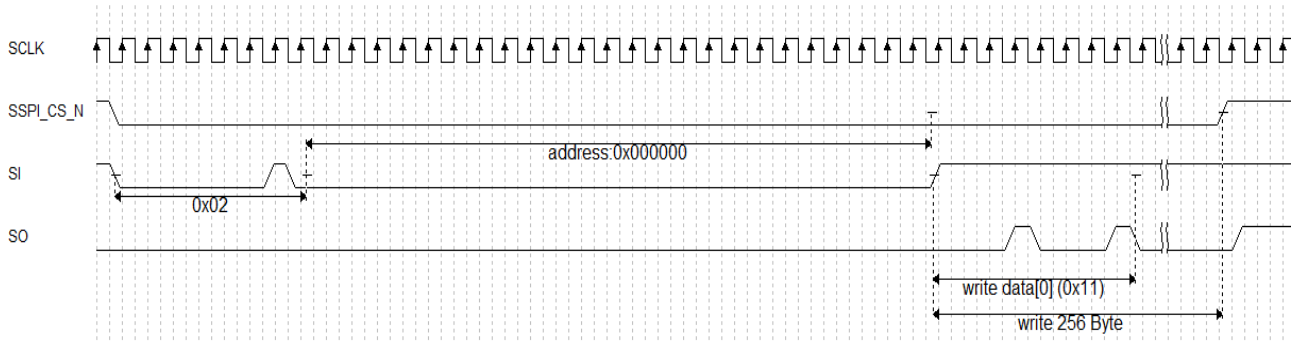
图 5-26 SSPI 写 WriteEnable (0x06) 指令的时序示意图



SSPI 编程内置 Flash

编程内置 Flash 一个 page 时序:

图 5-27 SSPI 编程一个 page 的时序示意图



SSPI 读取内置 Flash 数据

从 Flash 的 0x00 地址，回读一个 byte 时序:

图 5-28 SSPI 回读一个 byte 的时序示意图

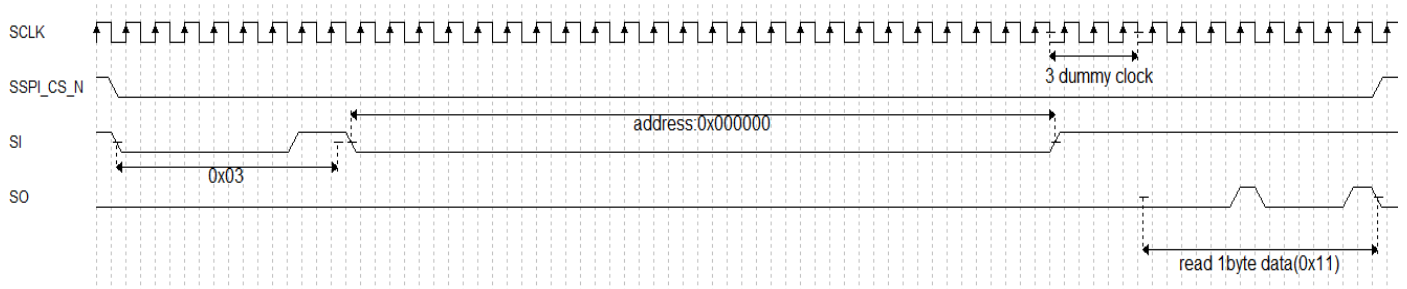
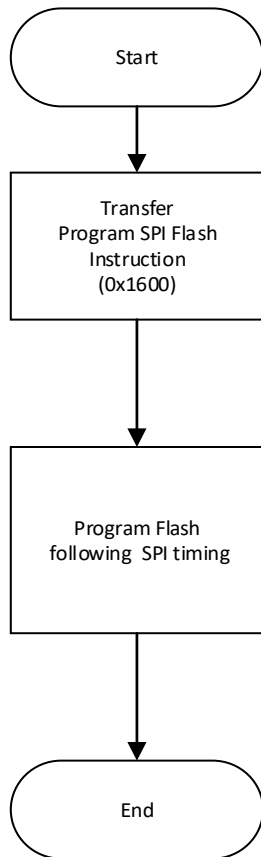


图 5-29 SSPI 配置 Flash 流程图



5.4.6 SSPI 模式下的多 FPGA 连线示意图

图 5-30 多 FPGA 连线示意图 1

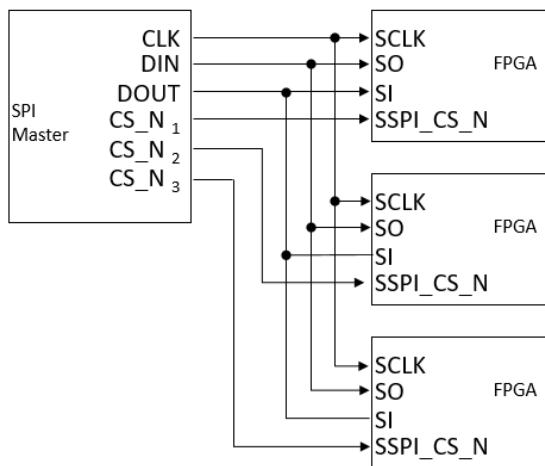
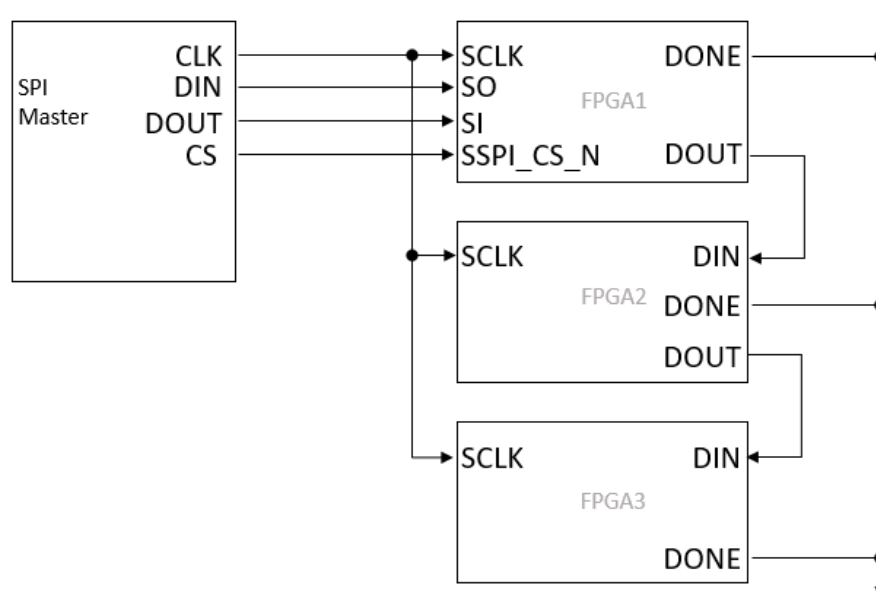


图 5-31 多 FPGA 连线示意图 2



5.5 QSSPI 配置模式

高云内置 Flash 已经默认打开了 Flash 的 Quad Enable Bit(QE), 可以直接使用 Quad Slave SPI(QSSPI)。QSSPI 模式相关的配置管脚如下表 5-12 所示。

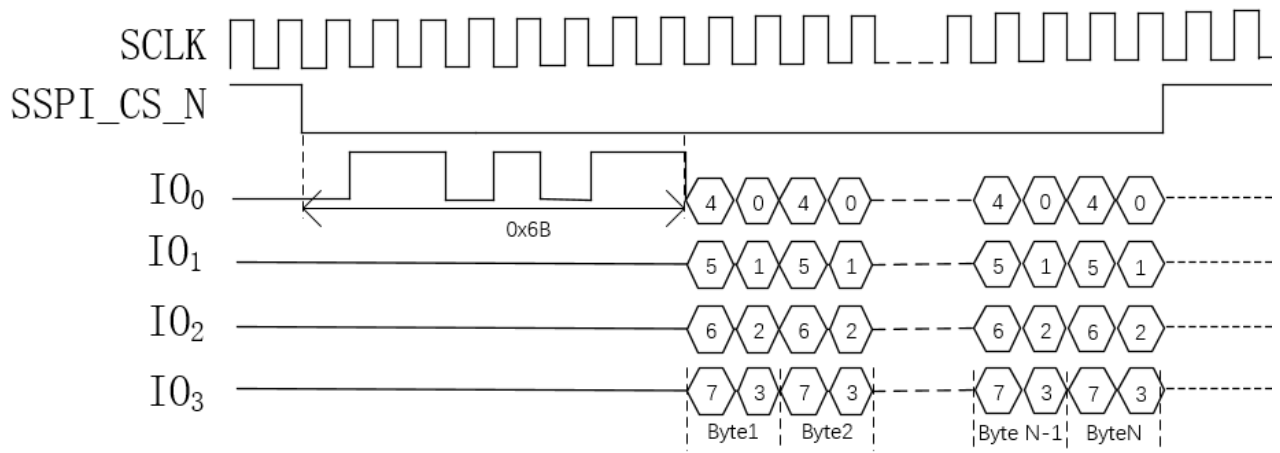
表 5-12 QSSPI 配置模式管脚

管脚名称	I/O 类型	说明
RECONFIG_N	I, 内部弱上拉	低电平脉冲: 启动 GowinCONFIG 配置
READY	I/O, 内部弱上拉	高电平: 当前可以对 FPGA 进行编程配置 低电平: 禁止对 FPGA 进行编程配置
DONE	I/O, 内部弱上拉	高电平: 成功完成编程配置 低电平: 未完成编程配置或编程配置失败
MODE[1:0]	I, 内部弱下拉	配置模式选择, READY 上升沿采样
SCLK	I, 内部弱上拉	输入时钟
IO ₃ (CLKHOLD_N)	I, 内部弱下拉	作为 Qaud SPI 的 IO ₃
IO ₂ (QSSPI_WPN)	I, 内部弱下拉	作为 Qaud SPI 的 IO ₂
IO ₁ (SO)	O, 内部弱下拉	作为 Qaud SPI 的 IO ₁
IO ₀ (SI)	I, 内部弱下拉	作为 Qaud SPI 的 IO ₀
SSPI_CS_N	I, 内部弱上拉	QSSPI 片选信号, 低电平有效

QSSPI 模式下配置 SRAM 流程如下所述, 其中 Read Status、Reinit、EraseSram、InitAddress、WriteEnable、WriteDisble 依旧使用 SSPI 指令, 只有 Write Data 时采用 QSSPI。

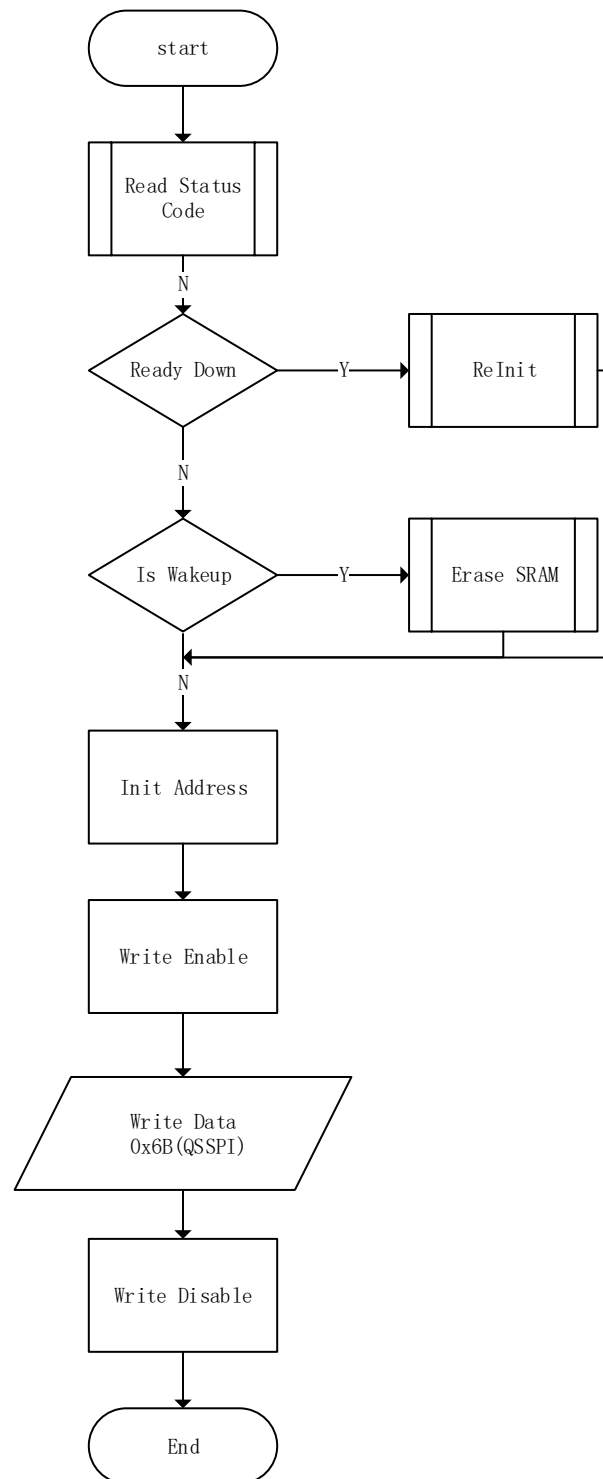
QSSPI Write Data (0x6B) 时序图如图 5-32 所示。

图 5-32 QSSPI Write Data (0x6B) 时序图



QSSPI 配置 SRAM 流程图如图 5-33 所示。

图 5-33 QSSPI 配置 SRAM 流程图



QSSPI 模式下编程 Flash 流程如下所述：

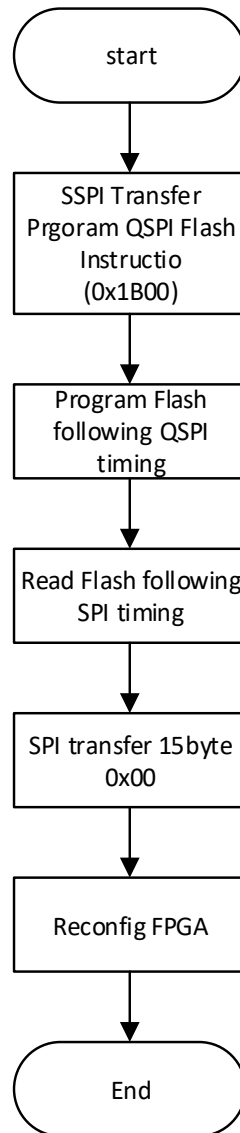
1. Standard SPI 发送 0x1B 指令 (sspi_prgm_qspi), 该阶段的 QSSPI_WPN 和 CLKHOLD_N 两个管脚要保持为 1。
2. QSPI 开始编程 Flash (编程 Flash 时指令 write enable(0x04)和 write disable(0x06)使用 Standard SPI 发送, 写数据时使用 QSPI)。
3. 编程结束后 Standard SPI 需要发送 15 个 0x00 指令。
4. 如果需要回读 Flash, 则需要使用 Standard SPI 回读 Flash, QSSPI 不

支持四线回读 Flash。

5. Standard SPI 发送 Reboot(0x3C)指令或触发 RECONFIG_N 管脚，然后 FPGA 自动加载 bitstream 文件。

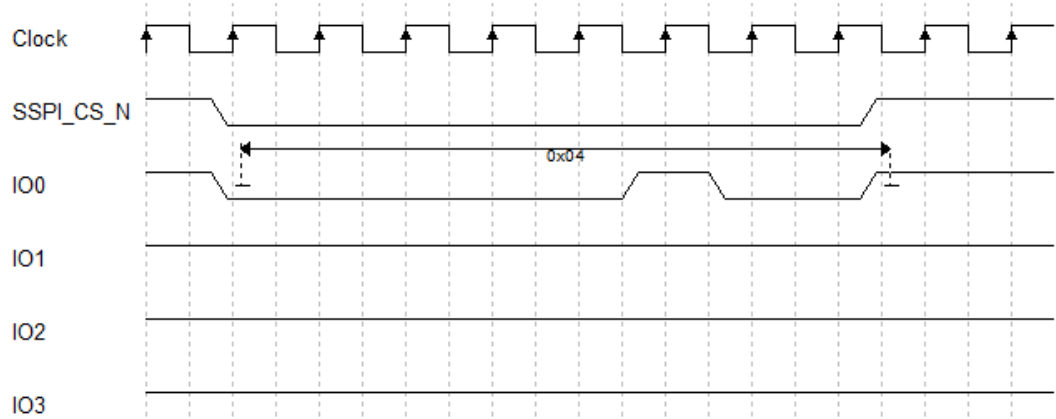
QSPI 编程 Flash 流程图如下：

图 5-34 QSSPI 编程 Flash 流程图



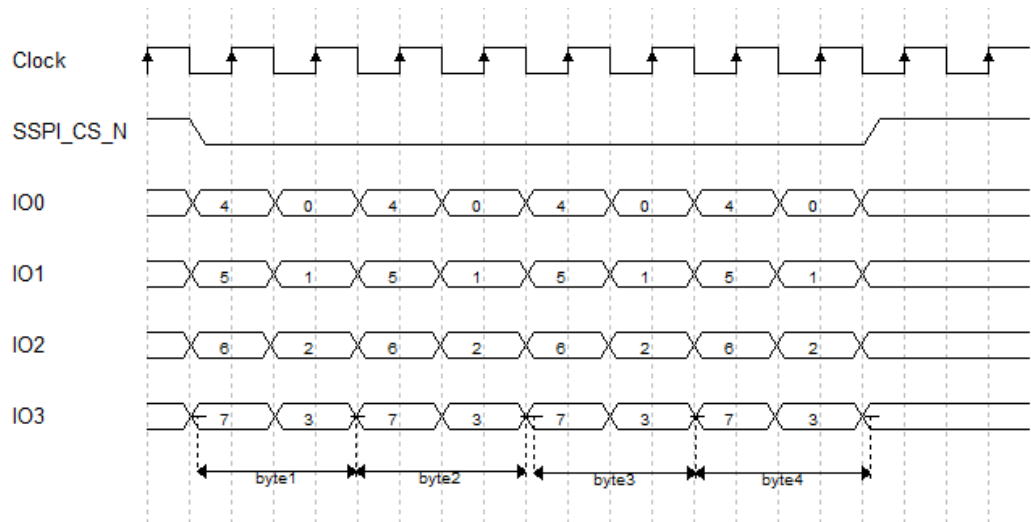
QSSPI write instruction（如 0x04）时序如下图所示。

图 5-35 QSSPI write instruction 时序图



QSSPI write data 时序如下图所示：

图 5-36 QSSPI write data 时序图



QSSPI page program 时序如下图所示：

图 5-37 QSSPI page program 时序图



注！

时序图表示从地址 0x000100 开始写 2Byte 数据 0x15 0x16。

5.6 CPU 配置模式

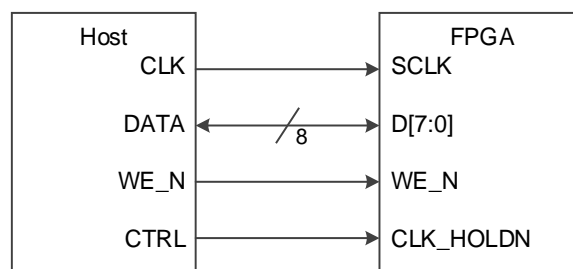
CPU 配置模式下，Host 通过 8-bit 位宽的数据总线接口对高云半导体 FPGA 产品进行配置，CPU 配置模式的管脚如表 5-13 所示。

表 5-13 CPU 配置模式管脚

管脚名称	I/O 类型	说明
RECONFIG_N	I, 内部弱上拉	低电平脉冲：启动 GowinCONFIG 配置
READY	I/O, 内部弱上拉	高电平：当前可以对器件进行编程配置 低电平：禁止对器件进行编程配置
DONE	I/O, 内部弱上拉	高电平：成功完成编程配置 低电平：未完成编程配置或编程配置失败
MODE[1:0]	I, 内部弱下拉	配置模式选择信号，READY 上升沿采样
SCLK	I, 内部弱上拉	输入时钟
CLKHOLD_N	I, 内部弱下拉	CPU 模式下的片选信号，低电平有效。即，若要在 CPU 模式下配置 FPGA，此信号需为低电平。
WE_N	I, 内部弱下拉	读写使能 0：写 1：读
D[7:0]	I/O	数据输入输出端口：CPU 配置模式时作为输入管脚，配置结束后可以转换为输出管脚进行验证

CPU 配置模式的连接示意图如图 5-38 所示。

图 5-38 CPU 配置模式连接示意图



注！

此图为 CPU 配置模式的最小系统图，MODE 值设置为“111”，其他固定管脚的接法请参考图 5-1。

除满足上电要求外，使用 CPU 模式对高云半导体 FPGA 产品进行配置，还需满足以下条件：

- CPU 接口使能
上电后初次配置或前一次配置时 RECONFIG_N 未设置为普通 I/O 状态。
- 启动新的配置
重新上电或者低电平脉冲触发 RECONFIG_N 管脚。

5.6.1 配置时序

在配置之前确保 $MODE[1:0]=11$ ，在配置完成后 $DONE$ 会拉高。如果 $DONE$ 或者 $READY$ 被拉低，说明配置未成功。

配置过程中，数据总线 $D[7:0]$ 按大端模式（MSB）表示，FPGA 在 $SCLK$ 下降沿读取数据。

图 5-39 CPU 模式配置流程示意图

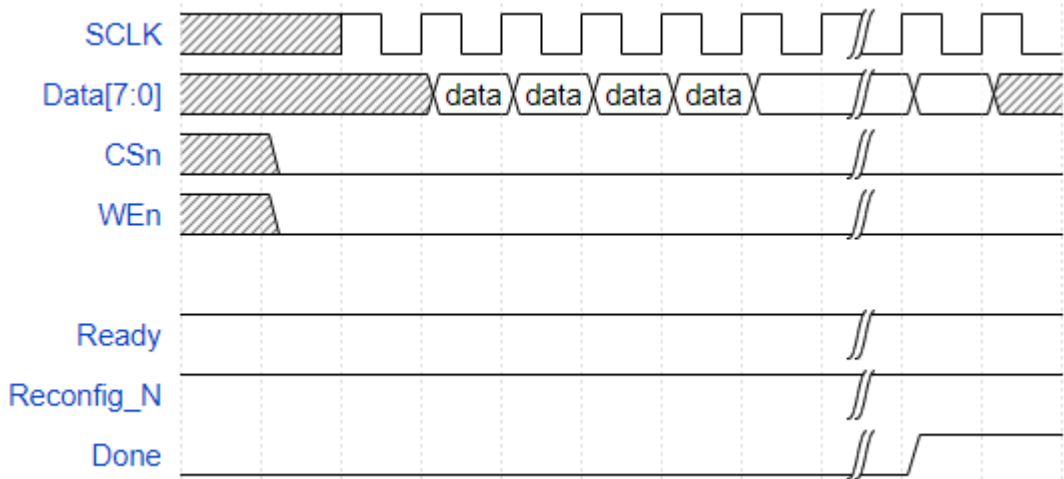


图 5-40 CPU 配置模式时序图

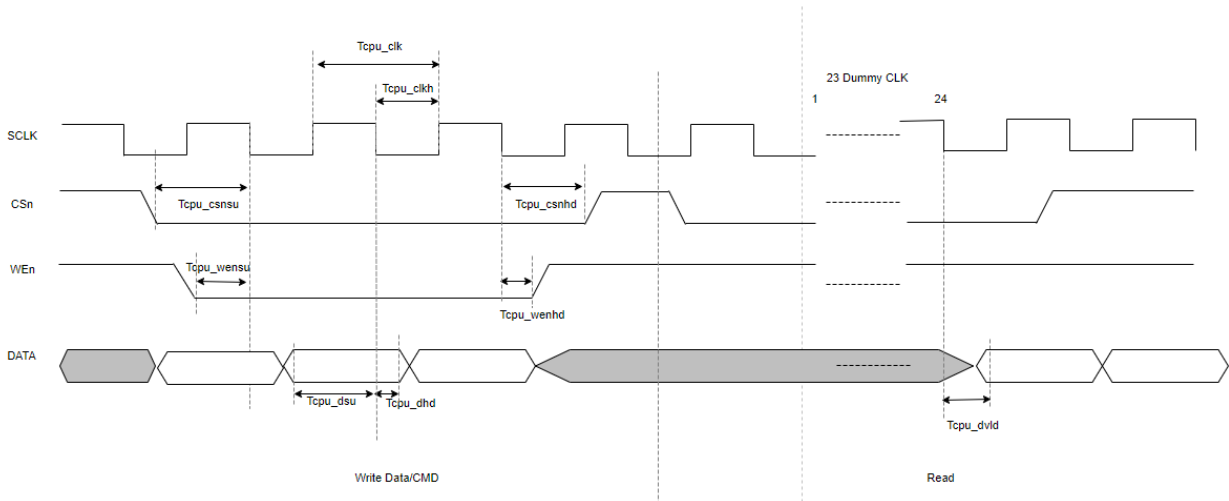


表 5-14 CPU 配置模式时序参数

名称	描述	最小值	最大值	单位
T _{cpu_clk}	CPU input clock period	40	—	ns
T _{cpu_csnsu}	CLKHOLD_N(CSn) setup time to SCLK falling	8	—	ns
T _{cpu_csnhd}	CLKHOLD_N(CSn) hold time from SCLK falling	0	—	ns
T _{cpu_wensu}	WE_N setup time to SCLK falling	8	—	ns
T _{cpu_wenhd}	WE_N hold time from SCLK falling	0	—	ns
T _{cpu_dsu}	Write data input setup time to SCLK falling	10	—	ns
T _{cpu_dhd}	Write data input hold time from SCLK falling	0	—	ns
T _{cpu_dvld}	SCLK falling to read data output valid	—	10	ns
T _{cpu_clkh}	CPU input clock high duration	(clock cycle) *45%	(clock cycle) *55%	—

5.7 SERIAL 配置模式

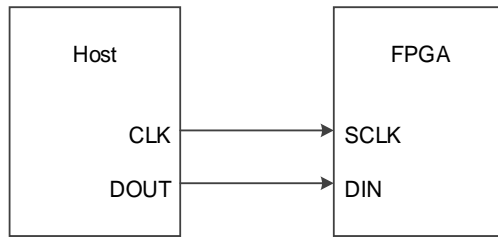
SERIAL 配置模式，Host 通过串行接口对高云半导体 FPGA 产品进行配置。SERIAL 配置模式是使用管脚数量最少的配置模式之一。SERIAL 配置模式只能将比特流数据写入 FPGA，无法从 FPGA 器件回读数据，因此，SERIAL 配置模式无法读取 ID CODE 和 USER CODE 以及状态寄存器信息。SERIAL 配置模式的管脚定义如表 5-15 所示。

表 5-15 SERIAL 配置模式管脚定义

管脚名称	I/O 类型	说明
RECONFIG_N	I, 内部弱上拉	低电平脉冲：启动 GowinCONFIG 配置
READY	I/O, 内部弱上拉	高电平：当前可以对器件进行编程配置 低电平：禁止对器件进行编程配置
DONE	I/O, 内部弱上拉	高电平：成功完成编程配置 低电平：未完成编程配置或编程配置失败
MODE[1:0]	I, 内部弱下拉	配置模式选择信号，READY 上升沿采样
SCLK	I, 内部弱上拉	输入时钟
DIN	I, 内部弱下拉	输入数据
DOUT	O, 内部弱下拉	输出数据，只用于 FPGA 级联时 SERIAL 配置模式

SERIAL 配置模式的连接示意图如图 5-41 所示。

图 5-41 SERIAL 配置模式连接示意图



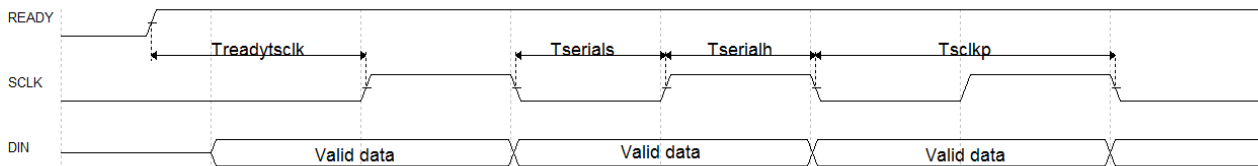
注！

此图为 SERIAL 配置模式的最小系统图，MODE 值为“10”，其他固定管脚的接法请参考图 5-1。

SERIAL 配置模式时序图

SERIAL 配置模式时序如图 5-42 所示。

图 5-42 SERIAL 配置模式时序图



相关的时序参数如表 5-16 所示。

表 5-16 SERIAL 配置模式时序参数

参数名称	参数含义	最小值	最大值
T _{scclkp}	SCLK 时钟周期(SCLK clock period)	15ns	-
T _{serials}	SERIAL PORT 建立时间(SERIAL PORT setup time)	2ns	-
T _{serialh}	SERIAL PORT 保持时间(SERIAL PORT hold time)	0ns	-
T _{readytsclk}	READY 上升沿到第一个 SCLK 沿时间(Time from READY rising edge to first SCLK edge)	TBD	-

除满足上电要求外，使用 SERIAL 模式对高云半导体 FPGA 产品进行配置，还需满足以下条件：

- SERIAL 接口使能
上电后初次配置或前一次配置时 RECONFIG_N 未设置为普通 I/O 状态。
- 启动新的配置
重新上电或者低电平脉冲触发 RECONFIG_N 管脚。

5.8 I²C 配置模式

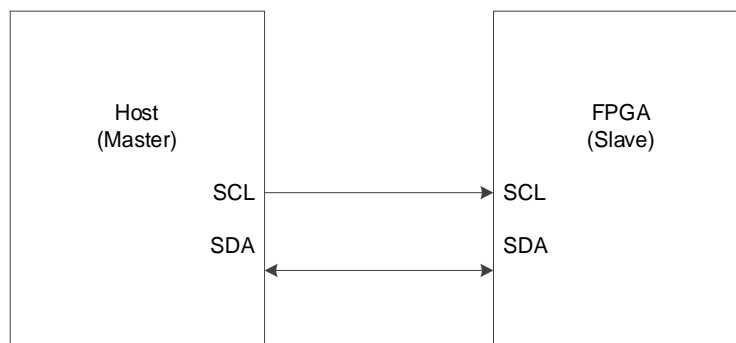
I²C 配置模式，Host 通过 I²C 接口对高云半导体 FPGA 产品进行配置。I²C 配置模式是使用管脚数量最少的配置模式之一。I²C 配置模式只能将比特流数据写入 FPGA，无法从 FPGA 器件回读数据，因此，I²C 配置模式无法读取 ID CODE、USER CODE、状态寄存器和回读校验。I²C 配置模式的管脚定义如表 5-17 所示。

表 5-17 I²C 配置模式管脚定义

管脚名称	I/O 类型	说明
RECONFIG_N	I, 内部弱上拉	低电平脉冲：启动 GowinCONFIG 配置。
READY	I/O, 内部弱上拉	高电平：当前可以对器件进行编程配置。 低电平：禁止对器件进行编程配置。
DONE	I/O, 内部弱上拉	高电平：成功完成编程配置。 低电平：未完成编程配置或编程配置失败。
MODE[1:0]	I, 内部弱下拉	配置模式选择信号，READY 上升沿采样。
SCL	I, 内部弱下拉	输入时钟。
SDA	I/O, 内部弱下拉	输入数据，或输出 ACK。 当配置模式支持 I ² C 时，SDA 管脚需要外上拉。

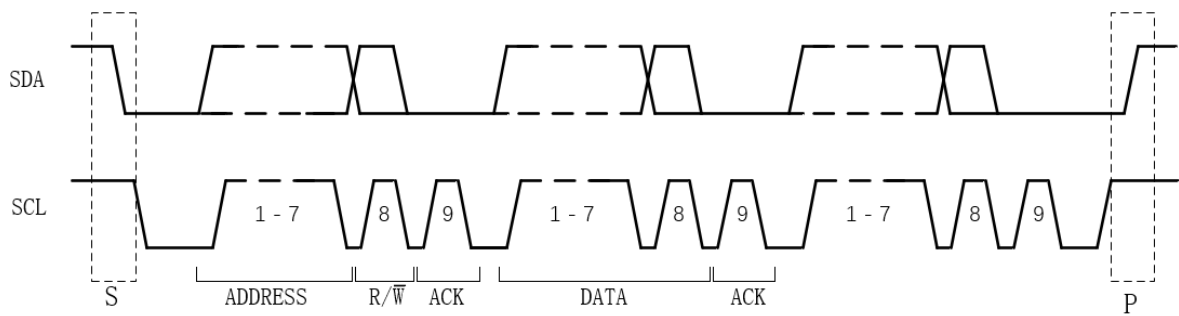
I²C 配置模式的连接示意图如图 5-43 所示。

图 5-43 I²C 配置模式连接示意图



注！

- 此图为 I²C 配置模式的最小系统图，其他固定管脚的接法请参考图 5-1。
- GW2AN-18X/9X 不支持标准 I²C 总线，仅支持 I²C 协议下单一设备配置。

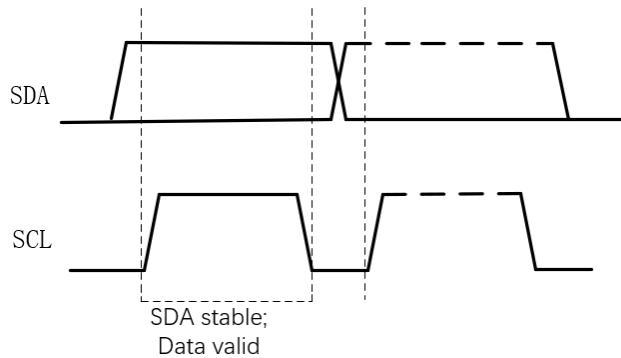
图 5-44 I²C 配置模式时序图

I²C 是串行传输总线，按照上图所示协议进行数据传输，正常状态下，SDA 和 SCL 都处于高电平。

表 5-18 I²C 配置模式时序参数

参数	参数含义	
S	启动条件	SCL 是高电平时，SDA 从高电平向低电平切换。
P	停止条件	SCL 是高电平时，SDA 由低电平向高电平跳变。
ADDRESS	地址帧	每个从属设备唯一的 7 位或 10 位序列，用于在主设备要与之通信时标识该从属设备。
R/W	读/写位	指定主设备是向从设备发送数据（0）还是向从机读取数据（1）。
ACK	ACK/NACK 位	消息中的每个帧后均带有一个 ACK/NACK 位，高云 FPGA 正确时返回 0。
DATA	数据	一个数据有 8bits，并以最高有效位优先发送。

I²C 总线上的所有 DATA 都是以 8 位字节传送的，发送器每发送一个字节，就在时钟脉冲 9 期间释放数据线，由接收器反馈一个应答信号。应答信号为低电平时，规定为有效应答位（ACK 位），表示接收器已经成功地接收了该字节；应答信号为高电平时，规定为非应答位（NACK），一般表示接收器接收该字节没有成功，对于反馈有效应答位 ACK 的要求是，接收器在第 9 个时钟脉冲之前的低电平期间将 SDA 线拉低，并且确保在该时钟的高电平期间为稳定的低电平。如果接收器是主控器，则在它收到最后一个字节后，发送一个 NACK 信号，以通知被控发送器结束数据发送，并释放 SDA 线，以便主控接收器发送一个停止信号。在 I²C 总线上传送的每一位数据都有一个时钟脉冲相对应（或同步控制），即在 SCL 串行时钟的配合下，在 SDA 上逐位地串行传送每一位数据。进行数据传送时，在 SCL 呈现高电平期间，SDA 上的电平必须保持稳定，低电平为数据 0，高电平为数据 1。只有在 SCL 为低电平期间，才允许 SDA 上的电平改变状态。逻辑 0 的电平为低电压，而逻辑 1 则为高电平，如下图所示。



高云器件支持的 I²C 配置模式信息如表 5-19 所示。

表 5-19 I²C 配置模式频率及地址

模式	器件	频率	地址
SRAM	GW2AN-18X/9X 系列	100KHz~555K	7'b1010 <u>000</u> ^[1]
内置 Flash ^[2]			

注!

- [1] I²C 的 slave 地址支持 2bit 配置，默认地址为 7'b1010000，下划线位置为可修改为 1，但不能回改为 0。
- [2] I²C 操作 Flash，需要先将数据流文件转换成特定数据码流，该转换工具包含在 Programmer 中，转换后的文件名后缀以“.i2c”表示。“i2c”是一个二进制文件。

除满足上电要求外，使用 I²C 模式对高云半导体 FPGA 产品进行配置，还需满足以下条件：

- I²C 接口使能
上电后初次配置或前一次配置时 RECONFIG_N 未设置为普通 I/O 状态。
- 启动新的配置
重新上电或者低电平脉冲触发 RECONFIG_N 管脚。

5.8.1 配置指令

I²C 配置模式使用了统一的地址，通过不同的指令指定配置 SRAM 或 Flash。以下是 I²C 指令一览表：

表 5-20 I²C 配置指令

名称	完整指令（指令字节+冗余信息字节）
Reinit	0x3f
Config-SRAM	0x33
Config-Flash	0x55
Reboot/Reconfig	0x3c

图 5-45 Reinit 时序图

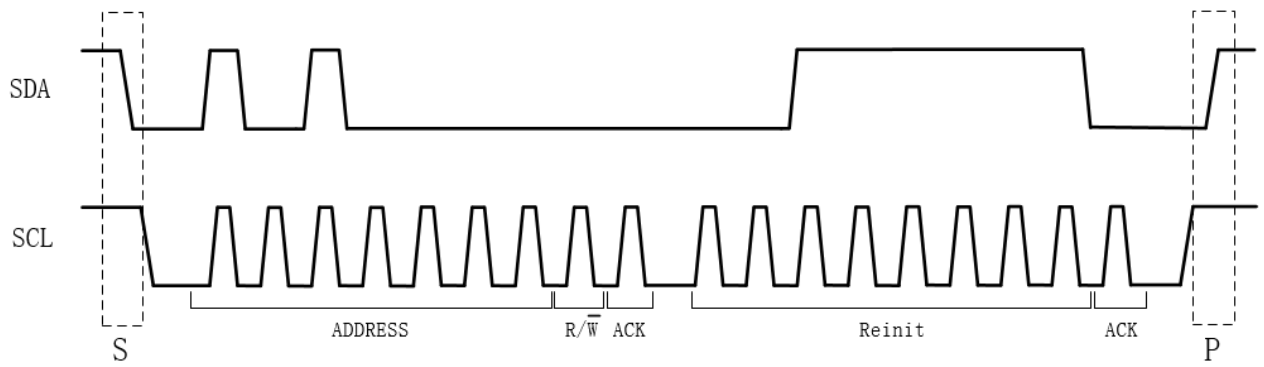


图 5-46 配置 SRAM 时序图

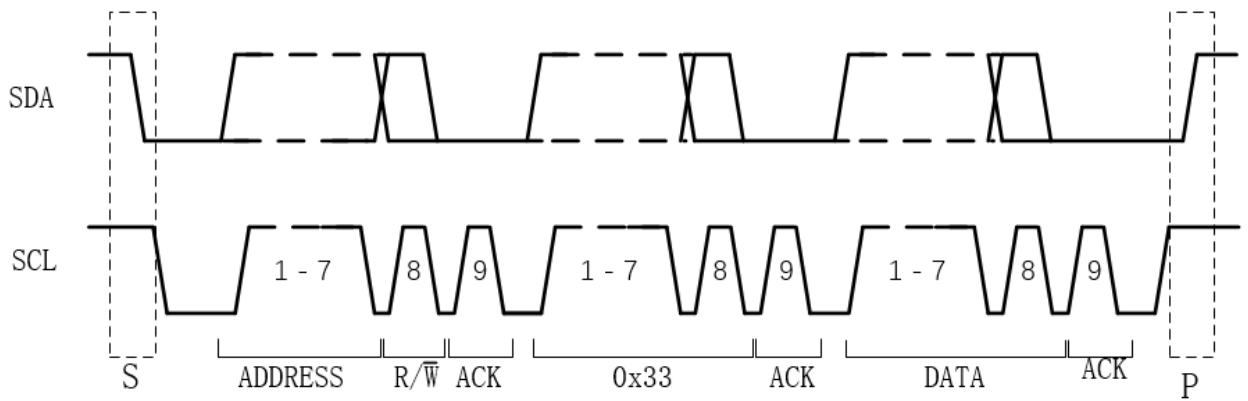


图 5-47 配置 Flash 时序图

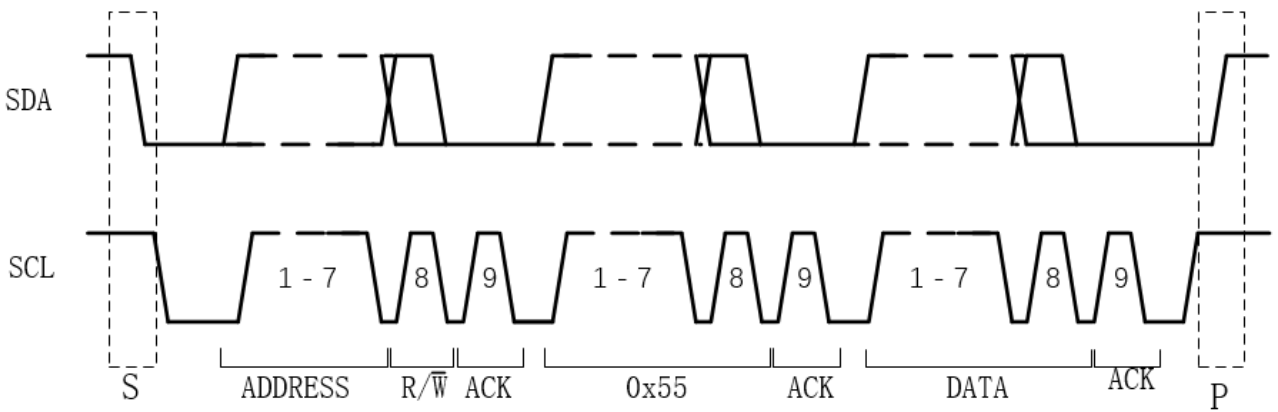
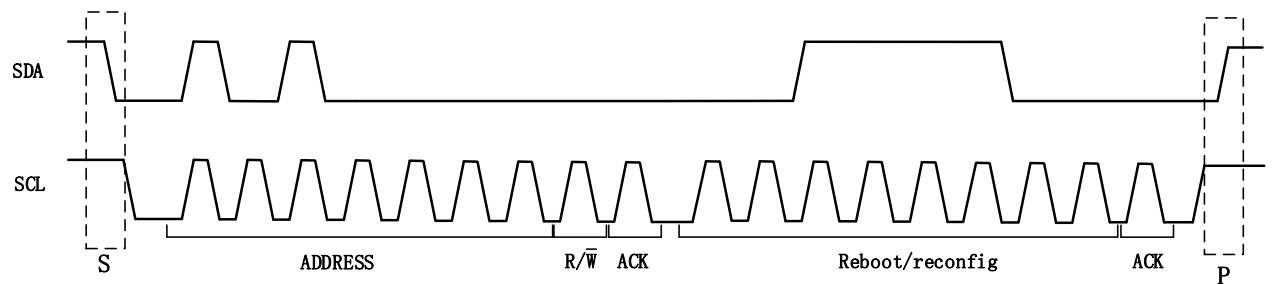


图 5-48 Reboot 时序图



5.8.2 I²C 配置 SRAM 流程

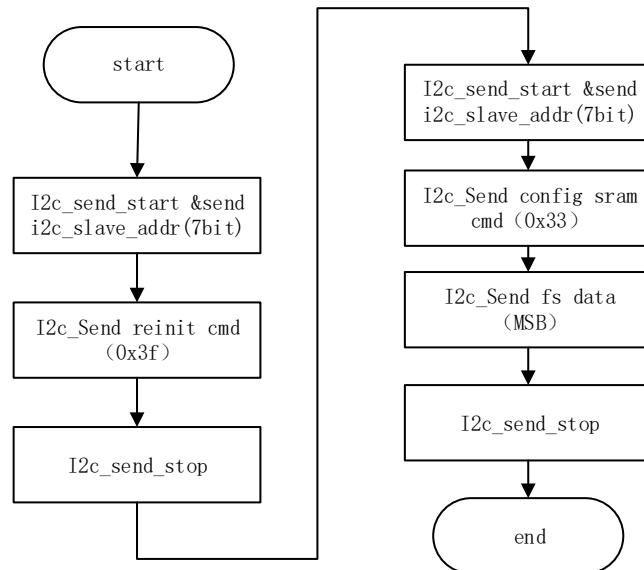
注!

重新配置 SRAM 时, 需要先擦除已存在的 SRAM。可通过重新上电或者低电平脉冲触发 RECONFIG_N 管脚来擦除 SRAM。

I²C 配置 SRAM 的流程如下:

1. 发送 start 信号和 7 位地址+写操作;
2. 发送数据 0x3f(reinit 指令);
3. 发送 stop 信号;
4. 发送 start 信号和 7 位地址+写操作;
5. 发送数据 0x33(config_sram 指令);
6. 发送数据流文件数据, 按照 MSB 方式发送;
7. 数据流文件发送完成后发送 stop 信号。

图 5-49 I²C 配置 SRAM 的流程图



5.8.3 I²C 配置 Flash 流程

I²C 配置 Flash 的 data 形式如下:

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

注!

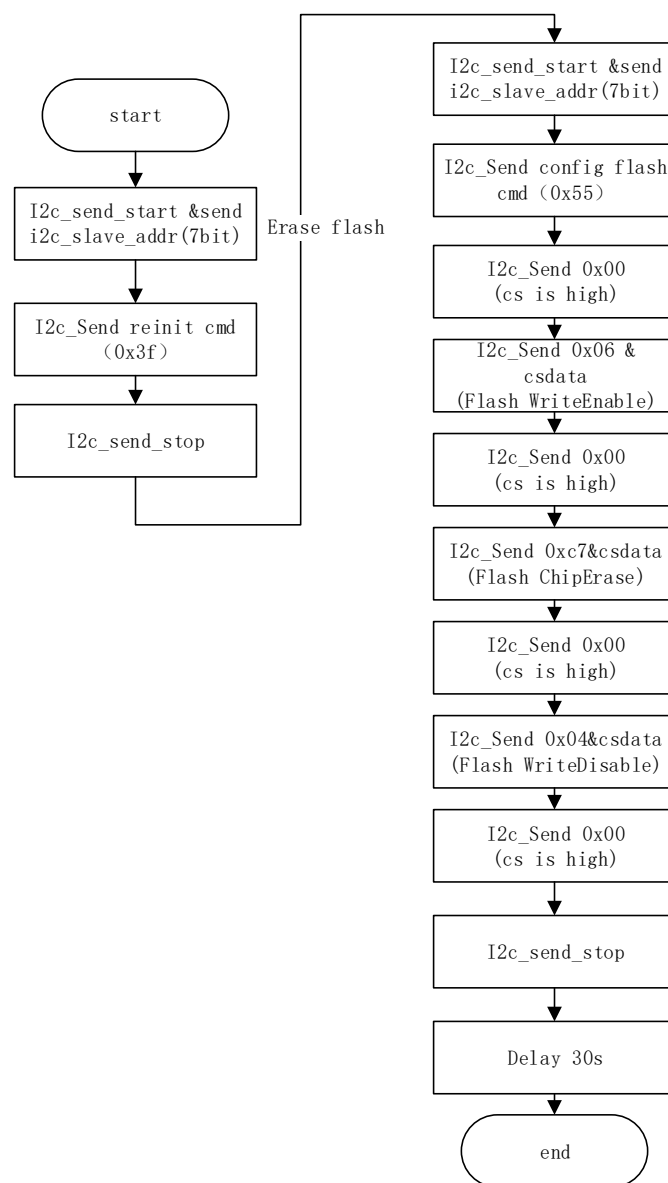
I²C 解析得到 CS 的值同 Flash 需要的 CS 值相反!

I²C 擦除 Flash 的流程

1. 发送 start 信号和 7 位地址+写操作;
2. 发送数据 0x3f(reinit 指令);
3. 发送 stop 信号;

4. 发送 start 信号和 7 位地址+写操作;
5. 发送数据 0x55(config_flash);
6. 发送数据 0x00,拉高 CS 信号;
7. 发送数据 0xaa 0xbe(DI:06&CS:ff), Flash 写使能;
8. 发送数据 0x00,拉高 CS 信号;
9. 发送数据 0xfa 0xbf(DI:c7&CS:ff), Flash 整片擦除;
10. 发送数据 0x00,拉高 CS 信号;
11. 发送数据 0xaa 0xba(DI:04&CS:ff), Flash 写失能;
12. 发送数据 0x00,拉高 CS 信号;
13. 发送 stop 信号;
14. 延时 30s, 等待 flash 擦除完成。

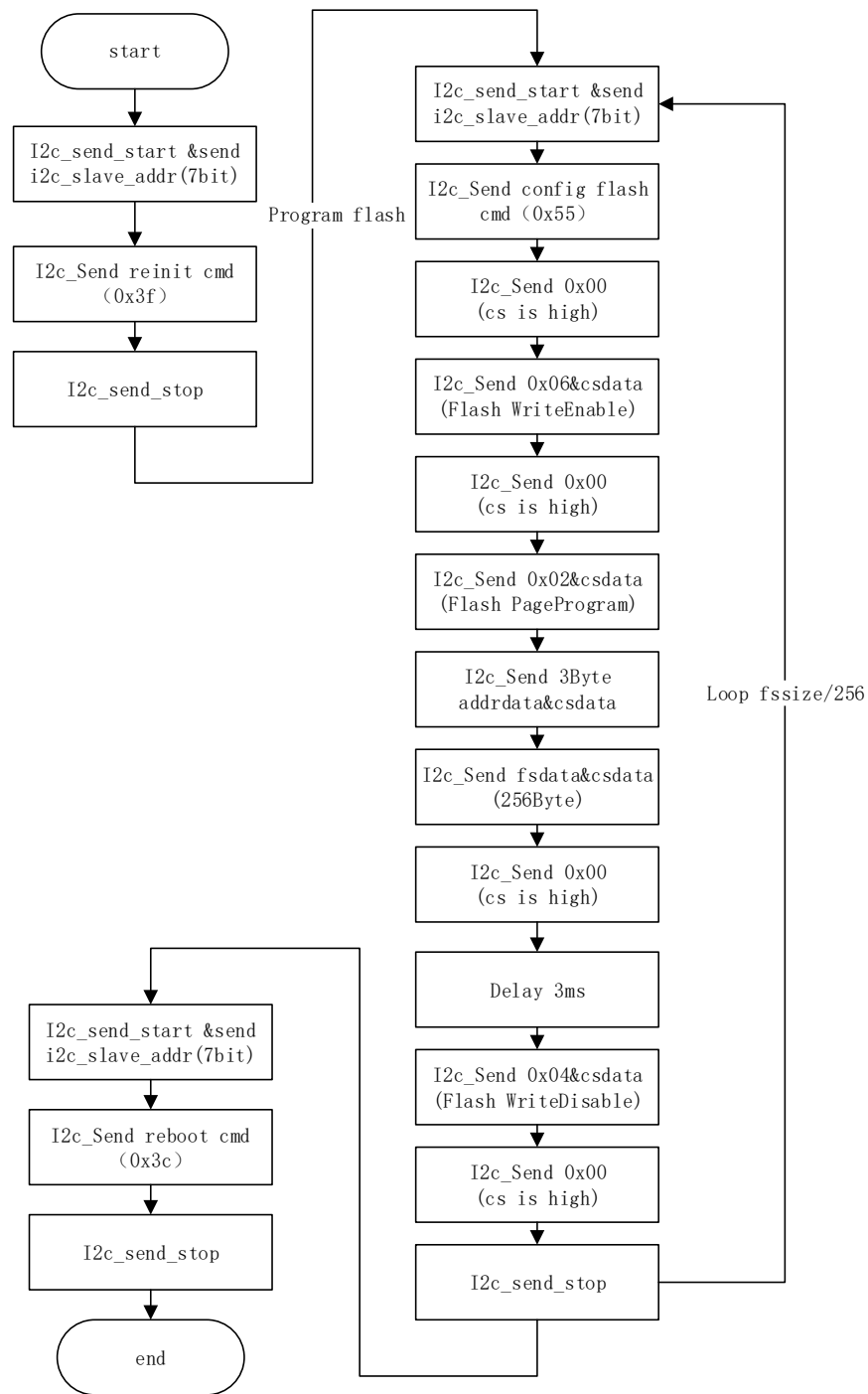
图 5-50 I2C 擦除 Flash 的流程图



I²C 写 Flash 的流程

1. 发送 start 信号和 7 位地址+写操作;
2. 发送数据 0x3f(reinit 指令);
3. 发送 stop 信号;
4. 发送 start 信号和 7 位地址+写操作;
5. 发送数据 0x55(config_flash);
6. 发送数据 0x00,拉高 CS 信号;
7. 发送数据 0xaa 0xbe(DI:06&CS:ff), Flash 写使能;
8. 发送数据 0x00,拉高 CS 信号;
9. 发送数据 0xaa 0xae(DI:02&CS:ff), pageprogram 指令;
10. 发送 24bit 地址(DI:xx&CS:ff);
11. 发送数据流文件数据(DI:xx&CS:ff);
12. 发送数据 0x00, 拉高 CS 信号;
13. delay 3 ms;
14. 发送数据 0xaa 0xba(DI:04&CS:ff), Flash 写失能;
15. 发送数据 0x00, 拉高 CS 信号;
16. 发送 stop 信号;
17. 循环步骤 5-15, 直到数据流文件写完;
18. 发送 start 信号和 7 位地址+写操作;
19. 发送数据 0x3c(reboot 指令);
20. 发送 stop 信号。

图 5-51 I2C 配置 Flash 的流程图



6 比特流文件配置

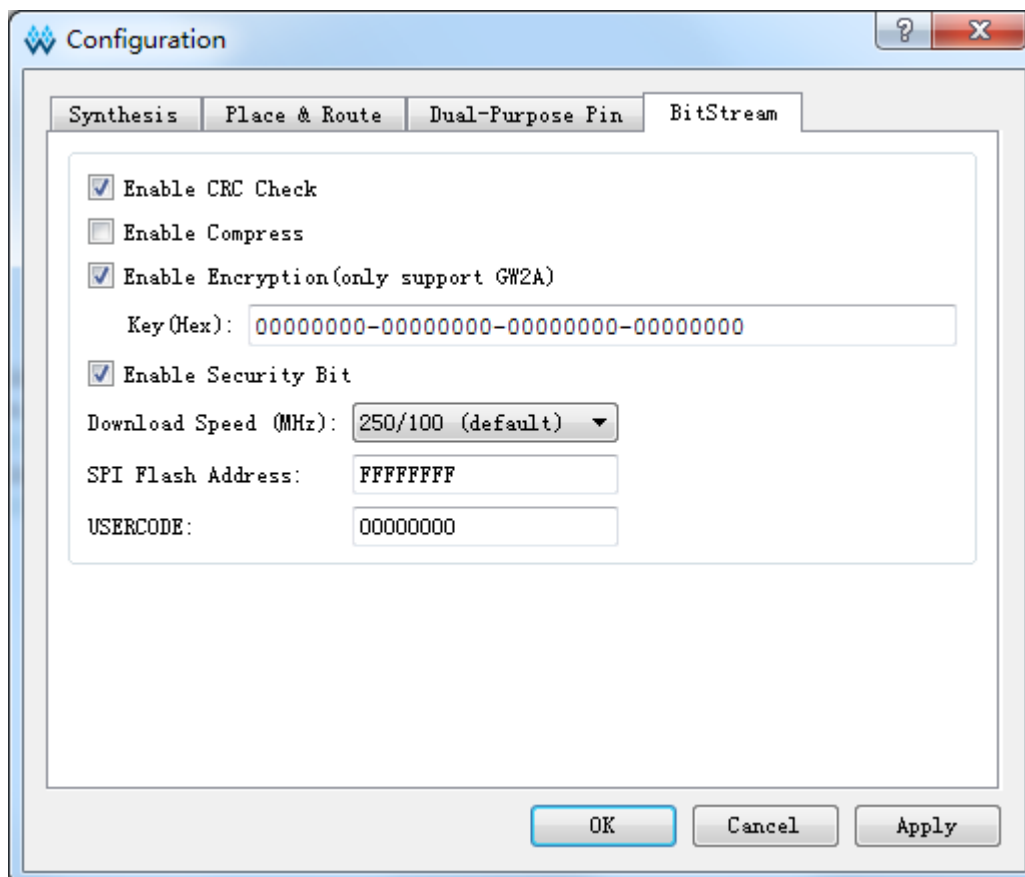
高云半导体 **FPGA** 产品编程配置相关的特性需要结合云源软件进行选项设置。配置主要包括配置管脚复用相关的选项和比特流文件配置相关选项。本章主要描述比特流文件配置相关信息，配置管脚复用相关信息请参考 [4.1.2 配置管脚复用](#)。

高云半导体为保障配置数据的安全准确传输，在 **FPGA** 产品的比特流文件中默认加入了 **CRC** 校验算法并设置了安全位。数据配置过程中实时校验输入数据是否出错，错误的无法唤醒器件，**DONE** 信号拉低。设置了安全位的比特流数据完成配置后，任何用户无法进行回读操作。

6.1 配置选项设置

比特流数据生成相关的设置界面请参考图 6-1，配置选项包括 **CRC** 校验使能、比特流数据压缩、加密密钥设置、安全位设置、**MSPI** 配置频率选择、多重配置模式下 **SPI Flash** 启动地址设置、**USER CODE** 设置等。**SPI Flash** 启动地址低 12 位无效，用户可设置的是 **ADDR[23:12]** 的地址空间。

图 6-1 配置选项



注！

高云半导体云源软件勾选加密密钥设置选项后强制勾选安全位设置选项，用户使用这样的比特流数据进行配置，既可以保证数据传输过程的安全，又能够阻止任何回读操作，最大限度地保障了用户数据的安全性。

6.2 配置数据加密

高云半导体 FPGA GW2AN-18X/9X 产品支持比特流数据加密，采用 128 bits 的 AES 加密算法。加密的比特流数据的配置流程如下：

1. 在 Gowin 云源软件输入加密密钥生成比特流文件；
2. 在 Gowin 编程软件输入解密密钥存入 FPGA；
3. 将加密的比特流数据加载到器件之后，器件会自行读取解密密钥进行数据解析。

数据解析成功后，器件完成配置并正常工作；数据解析失败后，器件无法工作，READY 和 DONE 信号拉低。

6.2.1 定义

- AES 密钥：也称 AES 私钥，AES 加密算法中用到的私钥部分，由算法外指定，本文简称 key；
- AES 密钥长度：128 位；
- Key：AES 密钥的简称，GW2A(R)系列 FPGA 产品中提供一个 128 位长度的空间用于存储 Key；

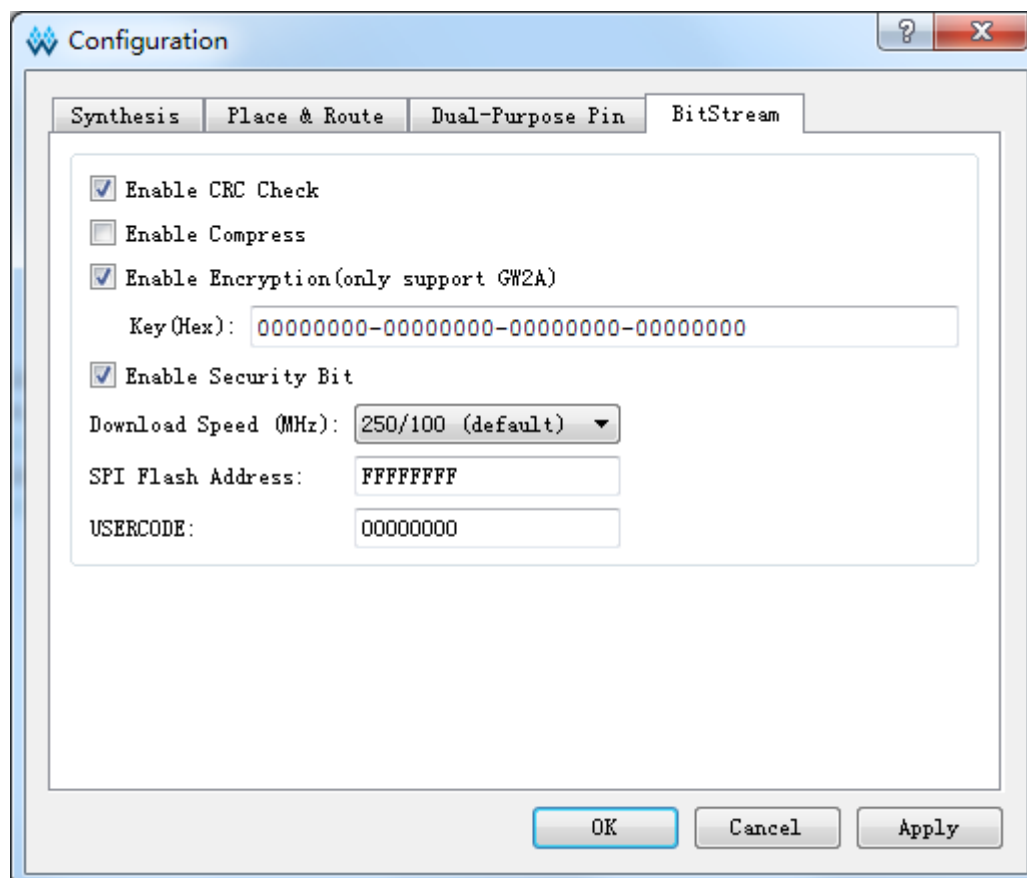
- **Lock:** 为保证 AES 密钥的安全, 该指令用于限制 key 的读权限, 本文将该过程简称 lock, 当处于锁定状态后, 设置后回读数据所有数据都是 1。

6.2.2 输入加密密钥

在云源软件中输入加密密钥的方法如下:

1. 打开云源软件中相应的工程;
2. 在菜单栏中选择“Project>Configuration”;
3. 单击“BitStream”页签, 勾选“Enable Encryption(only support GW2A)”并输入密钥值, 如图 6-2 所示。

图 6-2 加密密钥设置方法



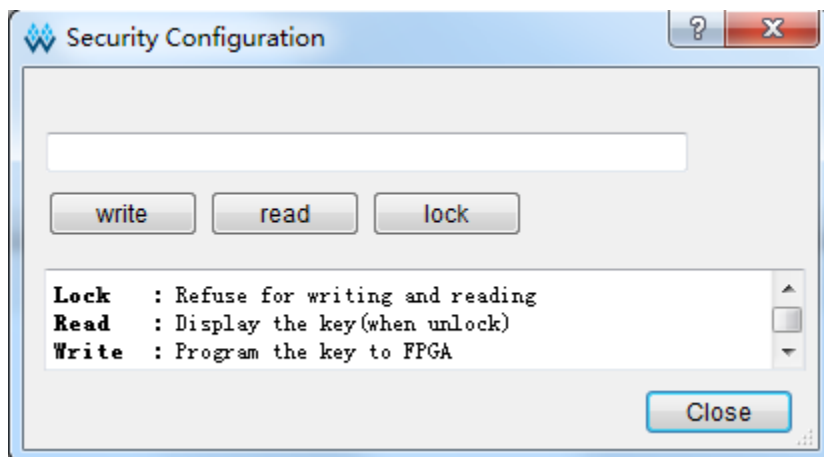
加密密钥设置成功后, 还需要将解密的密钥写入到 FPGA 的密钥存储区, 器件才能解析加密的比特流数据完成配置。

6.2.3 输入解密密钥

解密密钥的写入方法如下:

1. 打开 Gowin 编程软件;
2. 扫描 FPGA 器件;
3. 右键单击器件选择 Configure Security;
4. 在弹出的界面上输入之前加密的密钥值并单击“write”将其写入到 FPGA, 如图 6-3 所示。

图 6-3 解密密钥设置方法



解密密钥写入成功后可以选择界面上的读取指令回读写入的密钥进行验证。

密钥写入成功后，用户也可选择 **lock** 命令将密钥“锁死”在 FPGA 内部，之后任何对密钥的读取和写入操作都将无效：密钥值无法进行修改，读取的密钥所有位全部为“1”。

设置解密密钥后，加密的比特流数据只有与解密密钥匹配成功后才能唤醒。非加密比特流数据的配置不受密钥影响。

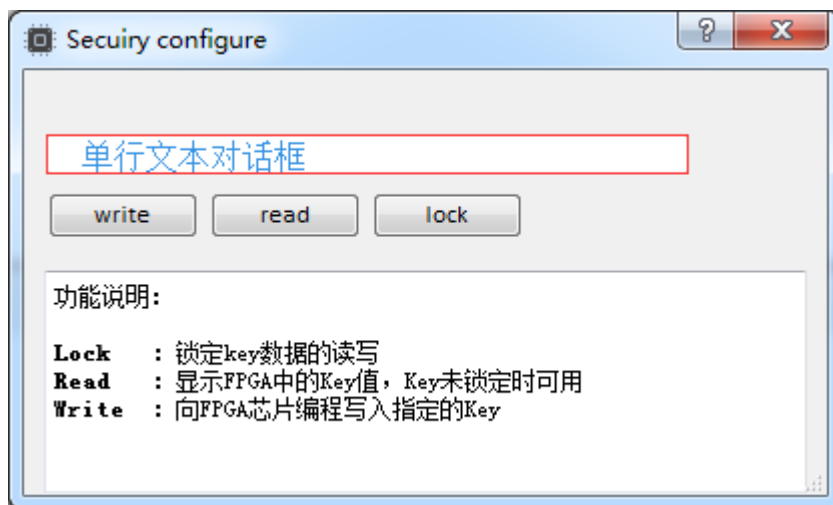
注！

高云半导体 FPGA 的密钥初始值所有位全部为 0，当把密钥值的某一位修改为 1 后便无法改回 0。例如，用户的某次操作中写入的密钥值为 00000000-00000000-00000000-00000001，之后修改此器件的密钥时最低位也必须是 1。

6.2.4 AES 密钥编程操作

Gowin Programmer 工具中提供了 AES 密钥编程工具，在 Gowin Programmer 中单击菜单“Edit”中“Security Key Setting”选项即可开启该工具，，如图 6-4 所示。

图 6-4 AES 编程对话框



该程序包含三个功能，分别是：

- Write: 编程 Key;
- Read: 读取 Key;
- Lock: 锁定 Key 的读写权限。

编程 Key (Write)

1. 将自定义的 Key(AES 密钥)填入“单行文本对话框”中;
2. 单击“write”按钮;
3. 工具运行结束, 返回验证结果。

读取 Key (Read)

单击“read”按钮可对写入的 AES 密钥进行再次验证, 读取出来的 AES 密钥会显示在“单行文本对话框”中。

锁定 Key (Lock)

单击“lock”按钮, 锁定 Key 数据的读写, AES 密钥将不能再被读取和写入。

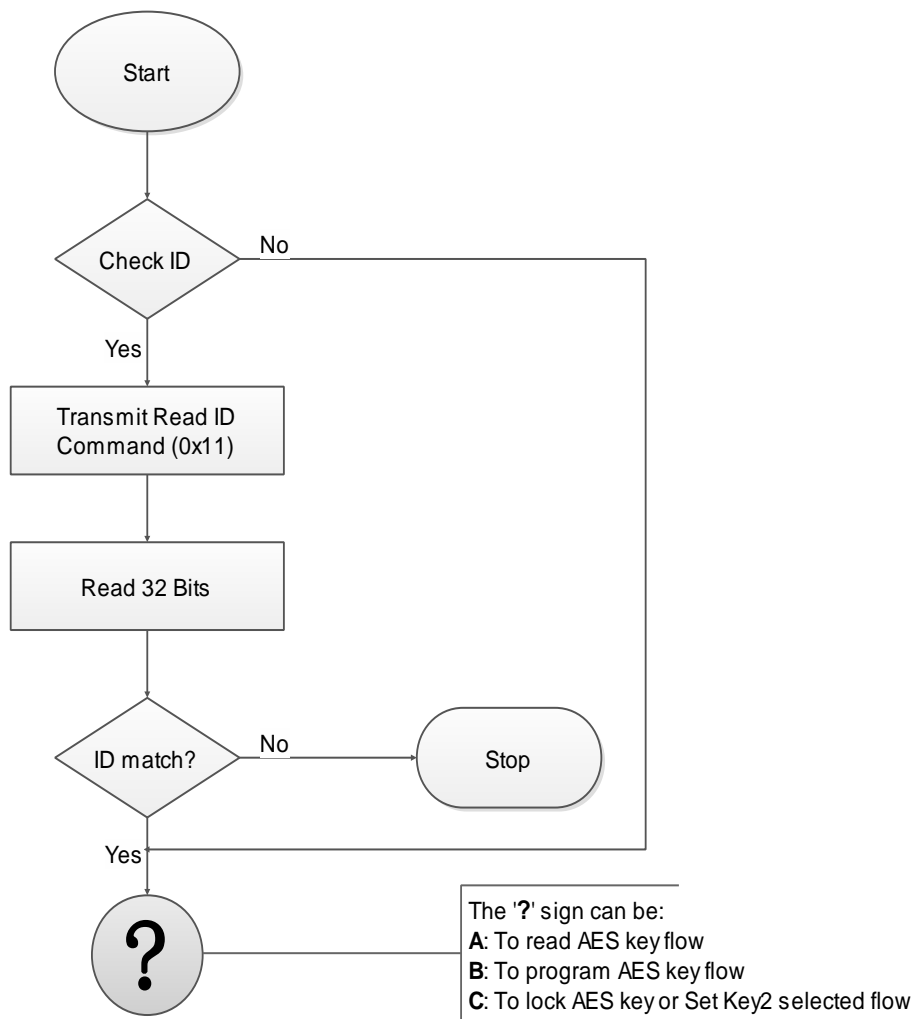
6.2.5 AES 密钥编程流程

图 6-5 ~ 图 6-8 给出了如何编程或锁定 AES 密钥的流程，图示流程均基于 JTAG 协议。

检查 IDCODE

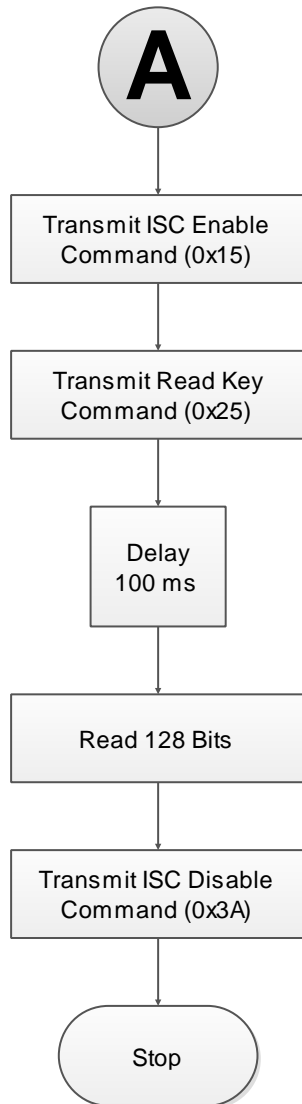
对器件 ID 进行检查，一方面可确定 JTAG 协议是否工作正常，另一方面确定烧录对象是否正确，避免误操作。

图 6-5 Prepare



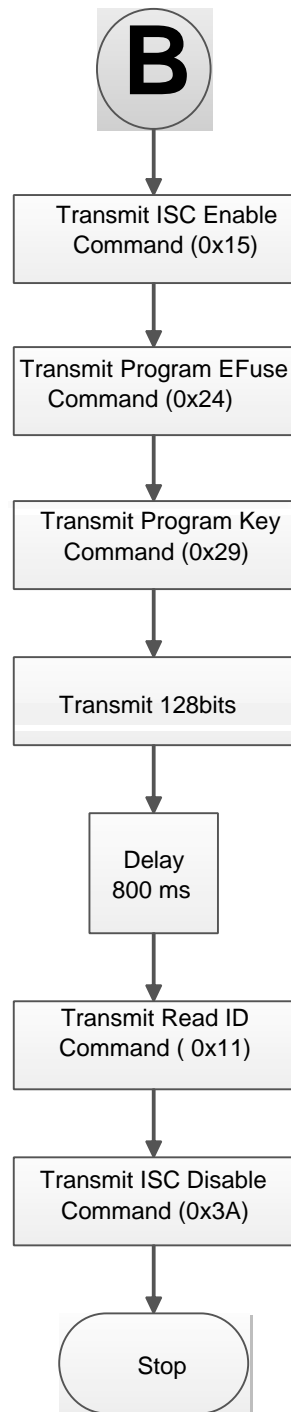
读取 AES Key

图 6-6 Read AES Key Flow



烧录 AES Key

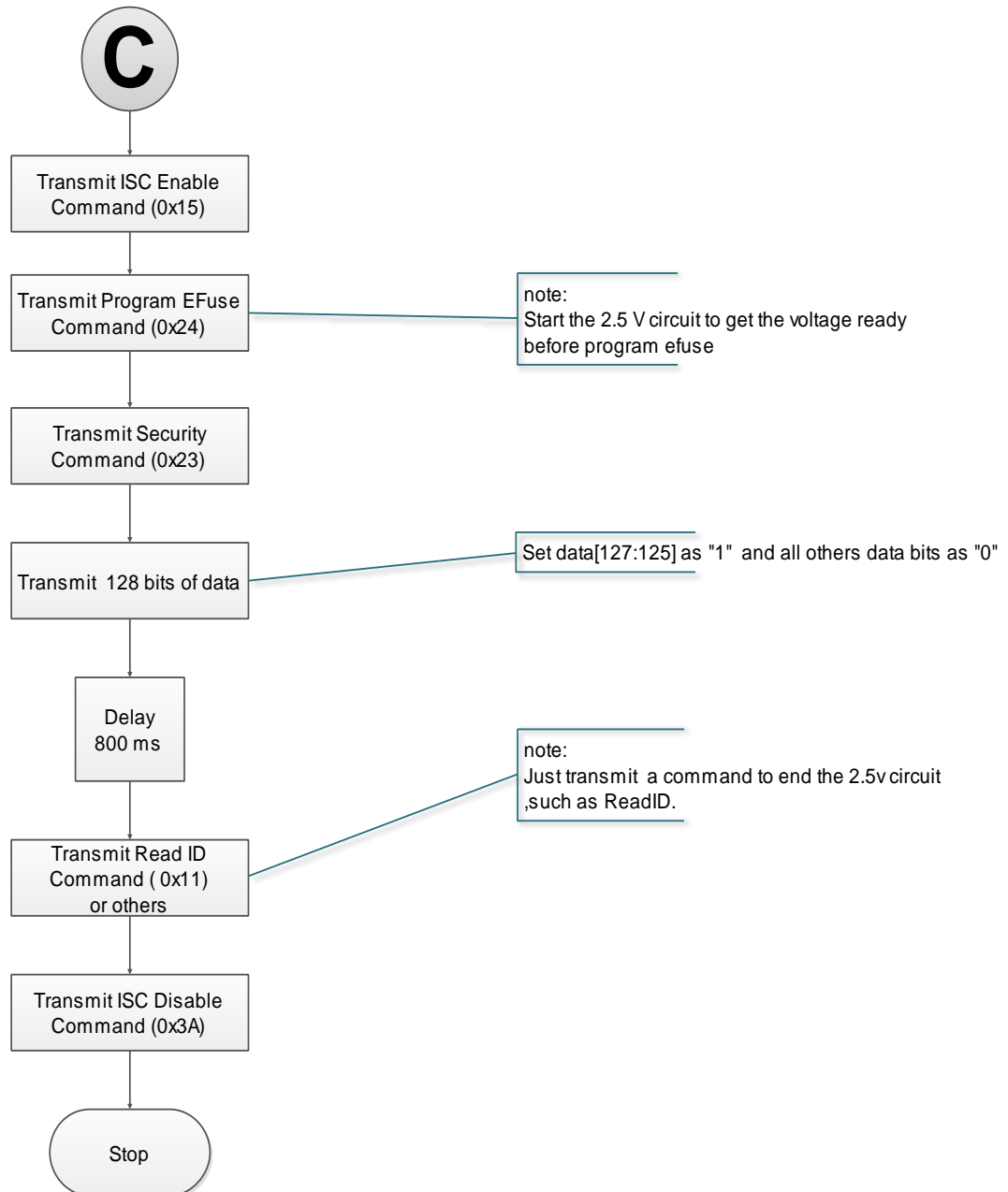
图 6-7 Program AES Key Flow



锁定 AES Key

锁定 AES Key 的作用是防止 Key 泄露。锁定 AES Key 之后将不能读取和配置 AES Key。

图 6-8 Lock AES Key Flow

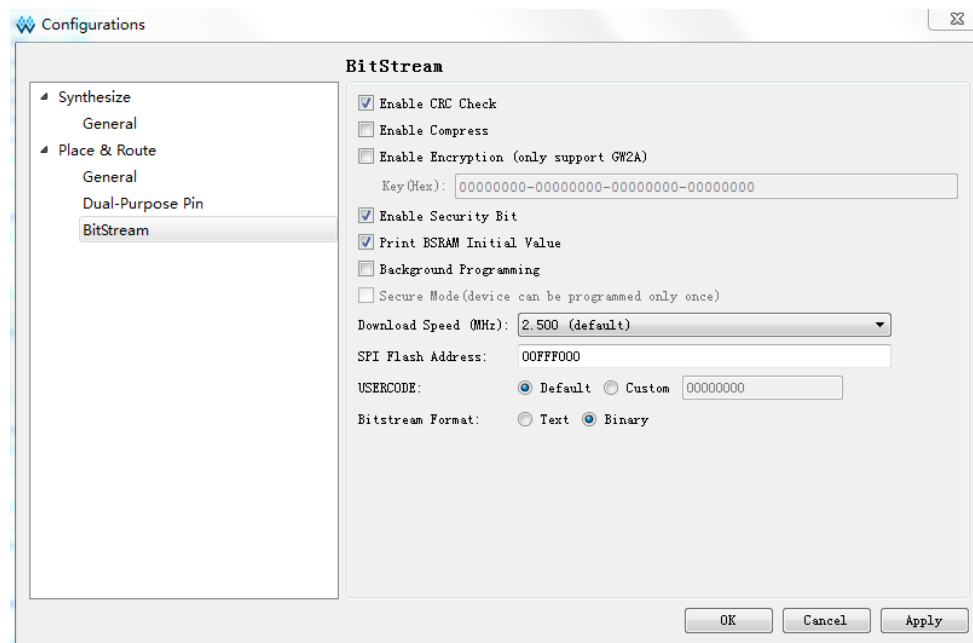


6.3 配置文件大小

高云半导体 FPGA 产品配置数据流文件的存储格式，包括携带注释信息的文本格式(ASCII)的文件和不携带注释信息的二进制格式的文件。文本格式的文件后缀名为.fs，其中以“//”开头的行属于注释信息，其他部分是数据流数据。二进制格式的文件的后缀名为.bin，其不包含注释信息，此文件格式通常用于嵌入式编程。用户可以在 Gowin 云源软件中设置存储格式：

1. 打开 Gowin 云源软件；
2. 在 Process 选项卡上右键单击 Place&Route 选择 Configuration 中的 bitstream；
3. 在 Bitstream Format 选项中选择 Text 或 Binary 格式即可，如图 6-9 所示。

图 6-9 比特流格式生成



高云半导体支持比特流数据的压缩，压缩比例与用户的设计相关，本文档只提供未压缩的配置文件大小，如表 6-1 所示。

表 6-1 高云半导体 FPGA GW2AN-18X/9X 产品配置文件大小（最大情况）

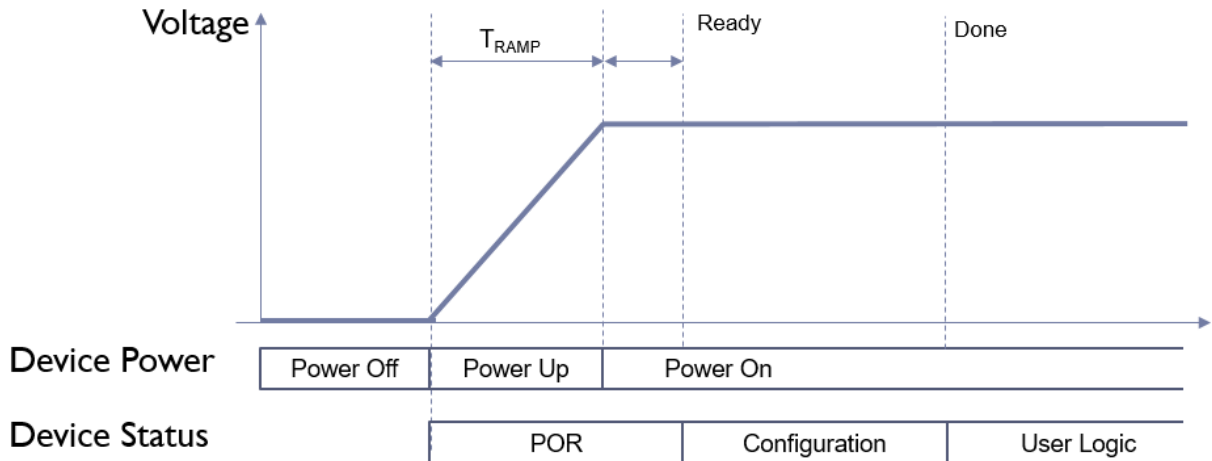
逻辑单元数量	最大配置文件
8,640	435 KBytes
20,736	887 KBytes

注！

表中的数据是二进制格式的文件大小，并且配置文件未进行压缩。

6.4 配置文件加载时长

FPGA 可以作为 Master 从 Flash 读取数据流文件并配置 SRAM，包括 Autoboot 和 MSPI 两种情况，分别对应数据源来自内置和外置 Flash。当 FPGA 上电并在 Ready 后开始尝试主动读取数据流文件，当完成加载后，FPGA 进到 User Logic 状态，如下图所示。



高云半导体 FPGA GW2AN-18X/9X 产品支持 Quad SPI 模式,从内置 Flash 自行读取数据流并配置 FPGA，读取配置文件的频率默认是 100MHz，每一个 SPI 时钟即读取 4bit，根据文件大小可计算加载所需时长。MSPI 读取 SPI Flash 的时钟最大支持 100MHz。注意的是，当使用 Fast Read SPI (0x0B) 时，要同时接地 FastRead_N 管脚。

根据配置文件大小、加载频率和每时钟加载位宽的不同，其加载时间也不一样。

AUTO BOOT 模式数据流文件加载时长如表 6-2 所示。

表 6-2 Autoboot 模式数据流文件加载时长

逻辑单元数量	最大配置文件	加载频率=100MHz 所需时间 (微秒) (Quad SPI)
10,368	252 KBytes	6.4
20,736	887 KBytes	22

以上列出的是加载时间的参考，设备从上电到配置完成，除了配置所占时间，还有设备上电的时间 T_{ramp} ，和设备初始化的时间，具体上电时间与电源器件有关，需要自行测量。所以 FPGA 从上电到加载完成大致时间可按如下公式计算：

Autoboot QMSPI 模式：

$$T_{\text{加载时长}} = \text{POR 时长} + \text{数据流比特数}/4/\text{加载频率}$$

Autoboot MSPI 模式：

$$T_{\text{加载时长}} = \text{POR 时长} + \text{数据流比特数}/\text{加载频率}$$

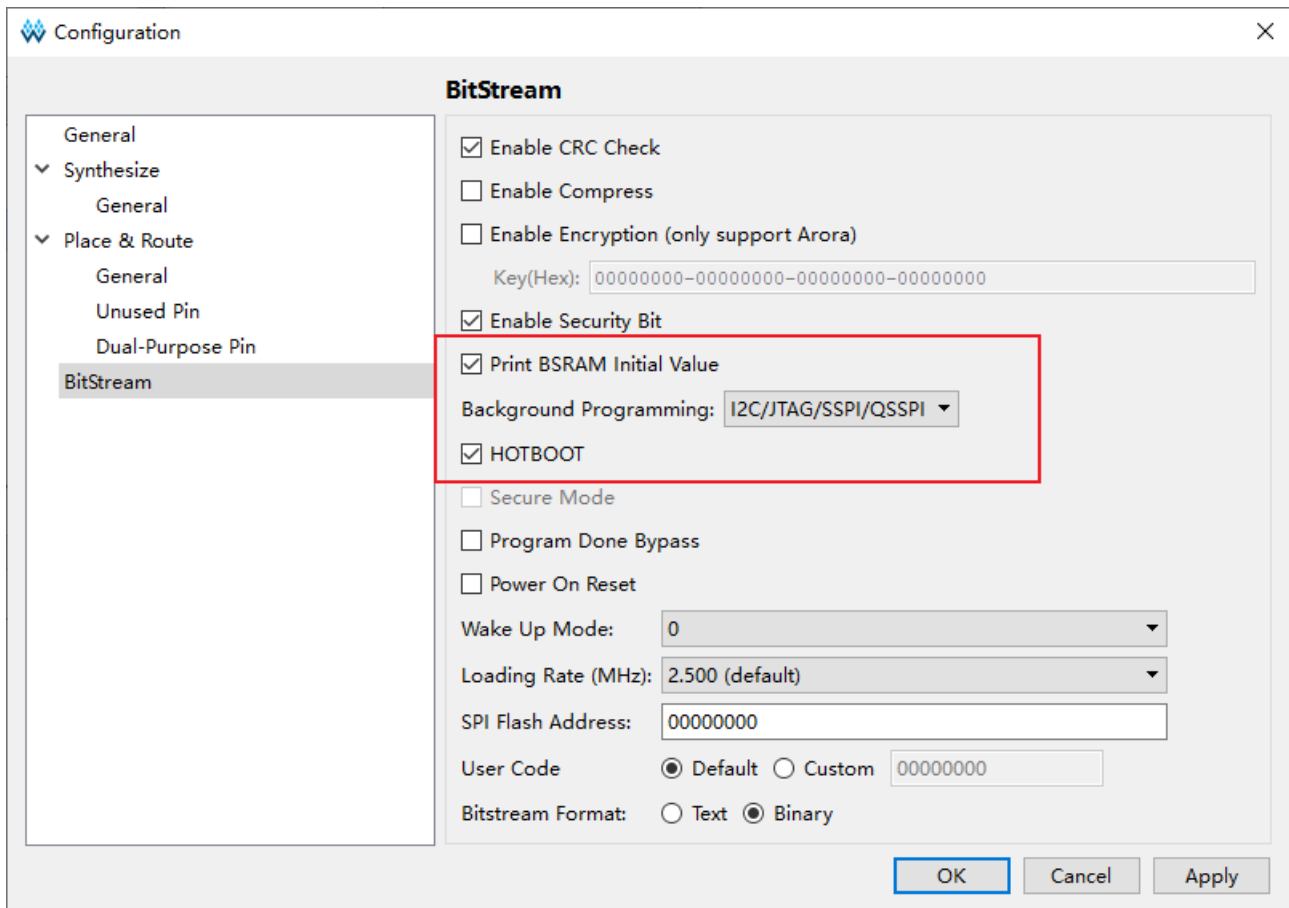
GW2AN-18X、GW2AN-9X 的 POR 时长约为 6.3 毫秒。

6.5 背景烧录 (Background Programming)

在不影响当前功能的情况下升级 Flash 内的数据文件，烧录器件不会中断配置，并且在加载新的数据流文件时，能够保持 IO 状态。

GW2AN-18X/9X 系列支持 JTAG/I2C/SSPI/QSSPI 协议的背景升级，在 IDE 中选择该功能项即可支持，无需修改烧录时序。

图 6-10 Background Programming 选项



7 安全性考虑

用户使用 **FPGA** 进行设计，安全性问题是一个关键的考虑因素，高云半导体的编程软件结合器件特性开发了一系列安全保护措施，为用户的比特流数据提供了完善的安全性保障机制。

安全措施大致分为三个阶段：

- 配置开始前，编程软件自动检查比特流数据的合法性；
- 配置过程中，器件实时校验传输数据的正确性；
- 配置完成后，器件进入工作状态，屏蔽一切形式的回读请求。

三个阶段的详细描述信息如下：

配置开始前

使用高云半导体的编程软件进行配置操作，请参见以下步骤：

1. 进行配置电路的硬件连接；
2. 启动编程软件进行器件扫描，软件自动识别已连接的 **FPGA** 产品；
3. 选择比特流数据和编程配置模式进行器件的编程配置。

上述过程中，编程软件首先读取已连接器件的 **ID**，然后将其与用户选择的比特流数据中的 **ID** 进行比较，只有二者一致才能进行操作，否则，用户选择的比特流数据被判定为非法数据，无法进行编程配置。

注！

高云半导体 **FPGA** 产品具有特定的 **ID**，以便与其他系列产品进行区分。使用 **Gowin** 云源软件生成的比特流数据中自动添加了器件的 **ID** 验证指令，用户只需在建立工程时选择具体器件即可。

配置过程中

配置过程开始后，器件首先读取比特流数据的 **ID** 信息进行校验，校验通过后开始编程配置过程。为防止比特流数据被篡改的情况和传输过程中可能发生的错误，高云半导体器件采用 **CRC** 校验的方式确保比特流文件中的所有数据位正确写入 **FPGA**，具体过程如下：

Gowin 云源软件生成的比特流数据中每段地址后都加入了该段地址对应数据的 CRC 校验码，高云半导体器件在接收数据的过程中也会不断地生成校验码，与器件接收的校验码进行比较，一旦发现校验错误，之后的数据将被忽略，配置完成后 DONE 指示灯不会被点亮，编程软件界面上弹出 CRC 校验出错的提示。

配置完成后

配置完成后，根据用户选择的编程配置模式，器件的比特流数据加载到 SRAM 中完成启动或者存储在内置 Flash 中。

- 对于加载到 SRAM 中的数据，Gowin 云源软件在生成比特流数据的过程中自动设置了安全位，任何用户都无法读取 SRAM 中的数据；
- 对于存储在内置 Flash 中的数据，软件完成 Flash 的编程操作后，Flash 配置为自启动模式，禁止所有的读取请求。
- 高云半导体 FPGA GW2AN-18X/9X 产品推荐在生成比特流数据时，使用 AES 加密。

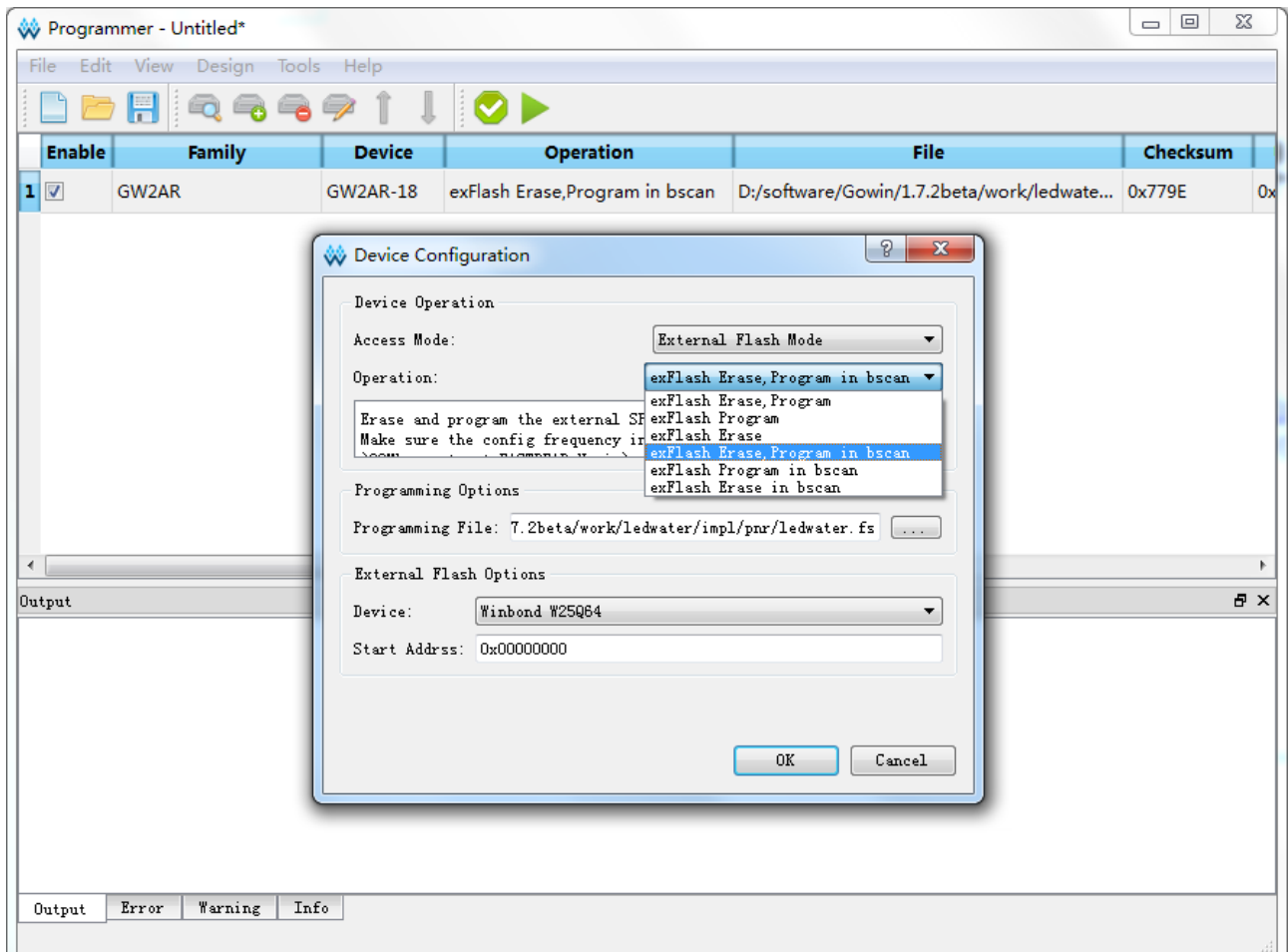
8 边界扫描操作

边界扫描操作是 JTAG 配置模式的扩展，扫描链分为长链和短链：长链主要结合 BSDL 文件进行器件的测试工作；短链主要针对 FPGA 链上的外部 Flash 进行擦除和读写操作。

边界扫描操作步骤如下：

1. 将 FPGA 开发板连接到 PC 并上电；
2. 打开 Gowin 编程软件扫描已连接的器件；
3. 在 Operation 下方双击选择外部 Flash 操作，并选择相关的 bscan 操作，如图 8-1 所示。

图 8-1 边界扫描操作示意图



边界扫描操作只能对 FPGA 外部 Flash 进行操作，无法对内置 Flash 或 SRAM 进行编程配置。边界扫描操作编程外部 Flash 时不必关心 FPGA MODE 值的设置，但是相比于常规 JTAG 编程外部 Flash 的方法，边界扫描操作的编程速度较慢。

附录 A 管脚状态信息

表 A-1 GW2AN-18X & 9X 器件的管脚在各个阶段时的状态(MSPI, MODE=00)

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

表 A-2 GW2AN-18X & 9X 器件的管脚在各个阶段时的状态(AUTOBOOT, MODE=01)

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

