



# Gowin FPGA 产品 JTAG 配置手册

TN653-1.05, 2018-08-02

## **版权所有© 2018 广东高云半导体科技股份有限公司**

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档进行适时的更新。

## 版本信息

日期	版本	说明
2017/04/13	1.00	初始版本。
2017/04/24	1.01	完善 JTAG 编程内置 Flash 流程。
2017/06/05	1.02	增加擦除 GW1N-1 内部 Flash 流程。
2017/07/28	1.03	增加擦除 GW1N-2/4/6/9 内部 Flash 流程。
2017/10/22	1.04	完善各配置/编程/读取流程相关描述信息。
2018/08/02	1.05	更新 GW1N4 Background Programming 流程图及 Transfer JTAG Instrction Sample & Extest 流程图等。

# 目录

目录 .....	i
图目录 .....	iii
表目录 .....	iv
<b>1 关于本手册 .....</b>	<b>1</b>
1.1 手册内容 .....	1
1.2 适用产品 .....	1
1.3 相关文档 .....	1
1.4 术语、缩略语 .....	2
1.5 技术支持与反馈 .....	2
<b>2 配置和烧录（Configuration&amp;Programming） .....</b>	<b>3</b>
2.1 JTAG 配置模式 .....	3
2.2 配置流程 .....	5
2.2.1 JTAG 引脚定义 .....	5
2.2.2 TAP 状态机 .....	5
2.2.3 TAP 复位 .....	5
2.2.4 指令寄存器和数据寄存器 .....	6
2.2.5 读取 ID CODE 实例 .....	7
2.2.6 配置 SRAM 流程 .....	9
2.2.7 读取 SRAM 的流程 .....	12
2.2.8 擦除内部 Flash .....	14
2.2.9 编程内部 Flash 流程 .....	17
2.2.10 读取内部 Flash 流程 .....	22
2.2.11 背景烧录（Background Programming） .....	25
2.2.12 编程外部 Flash .....	27
2.2.13 读取 Status Register 0x41 .....	31
2.2.14 读取 User Code 0x13 .....	32
2.2.15 重加载 0x3C .....	32
2.2.16 擦除 SRAM 0x15 .....	32

**3 例程文件 ..... 33**

# 图目录

图 2-1 JTAG 配置模式连接示意图 .....	4
图 2-2 TAP 状态机 .....	5
图 2-3 指令寄存器访问时序 .....	6
图 2-4 数据寄存器访问时序 .....	6
图 2-5 读取 ID Code 状态机流程图.....	8
图 2-6 读取 ID Code 指令-0x11 访问时序 .....	8
图 2-7 读取 ID Code 数据寄存器访问时序.....	8
图 2-8 配置 SRAM 流程.....	10
图 2-9 Transfer Configuration Data 过程示意 .....	11
图 2-10 读取 SRAM 的流程 .....	13
图 2-11 擦除 GW1N-2/4/6/9 内部 Flash 流程.....	16
图 2-12 擦除 GW1N-1 内部 Flash 流程 .....	17
图 2-13 编程内部 Flash 流程图.....	19
图 2-14 X-page 编程流程图 .....	21
图 2-15 Y-page 编程流程图 .....	22
图 2-16 读取内部 Flash 流程图.....	23
图 2-17 读取一个 Y-page 的过程 .....	24
图 2-18 GW1N4 Background Programming 流程图 .....	25
图 2-19 Transfer JTAG Instruction Sample & Extent 流程图 .....	26
图 2-20 JTAG 接口编程外部 Flash 连接示意图 .....	27
图 2-21 采用 config-mode[2:0]=011 模式编程 SPI Flash 流程示意图.....	28
图 2-22 GW2A 系列 JTAG 模拟 SPI 发送 0x06 指令时序图 .....	28
图 2-23 GW1N 系列 JTAG 模拟 SPI 发送 0x06 指令时序图.....	29
图 2-24 采用 Boundary Scan 模式编程 SPI Flash 流程示意图.....	30

# 表目录

表 1-1 术语、缩略语 .....	2
表 2-1 JTAG 配置模式管脚定义 .....	3
表 2-2 Gowin FPGA IDCODE .....	7
表 2-3 TDI 和 TMS 的值变化 .....	7
表 2-4 器件 SRAM 地址数量和地址长度 .....	12
表 2-5 JTAG 的 TC K 频率要求 .....	14
表 2-6 Readback-pattern / Autoboot-pattern .....	18
表 2-7 管脚状态 .....	29
表 2-8 Status Register 含义 .....	31

# 1 关于本手册

## 1.1 手册内容

本手册主要介绍 Gowin FPGA 产品的 JTAG 配置及烧录相关信息，包含 JTAG 配置模式、配置流程及相关例程文件。

## 1.2 适用产品

本手册中描述的信息适用于以下产品：

- GW1N 系列 FPGA 产品：GW1N-1, GW1N-2, GW1N-4(ES), GW1N-6, GW1N-9
- GW1NR 系列 FPGA 产品：GW1NR-4(ES), GW1NR-9
- GW2A 系列 FPGA 产品：GW2A-55, GW2A-18
- GW2AR 系列 FPGA 产品：GW2AR-18

## 1.3 相关文档

阅读本手册前，请登录高云半导体网站 [www.gowinsemi.com.cn](http://www.gowinsemi.com.cn) 参阅以下相关手册，了解 JTAG 1149.1 配置模式相关介绍：

- Gowin FPGA 产品编程配置手册



## 1.4 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	含义
FS file	Fuses file	包含配置数据的 ASCII 文件
Configuration	Configuration	配置 FPGA SRAM 区域的过程
Configuration Data	Configuration Data	配置 FPGA SRAM 的数据
Bitstream	Bitstream Data	配置 FPGA SRAM 的数据
Configuration Mode	Configuration Mode	配置模式，决定 Configuration Data 源
EFlash/EmbFlash	Embedded Flash Memory	FPGA 内置 Flash 存储器
Internal Flash	Internal Flash Memory	同 Embedded Flash
Programming	Programming	将 Configuration Data 烧录到 Embedded Flash 或 External Flash 存储器的过程
Edit Mode	Edit Mode	FPGA 处于 Configuration 或 Programming 所在模式
User Mode	User Mode	FPGA 在 Configuration 或 Programming 完成后，并且逻辑功能执行的模式
LSB	Least Significant Bit	最低有效位（优先）
MSB	Most Significant Bit	最高有效位（优先）
TAP	Test Access Port	测试访问口
Security Bit	Security Bit	安全位（使 SRAM 回读永为高电平）
Bscan	Boundary Scan	边界扫描测试技术

## 1.5 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](http://www.gowinsemi.com.cn)

E-mail: [support@gowinsemi.com](mailto:support@gowinsemi.com)

Tel: +86 755 8262 0391

# 2 配置和烧录 (Configuration&Programming)

## 2.1 JTAG 配置模式

Gowin FPGA 产品的 JTAG 配置模式符合 IEEE1149.1 边界扫描标准。

JTAG 配置模式是将比特数据下载到 Gowin FPGA 产品的 SRAM 中, 掉电后配置数据丢失。JTAG 配置模式的相关管脚如表 2-1 所示。

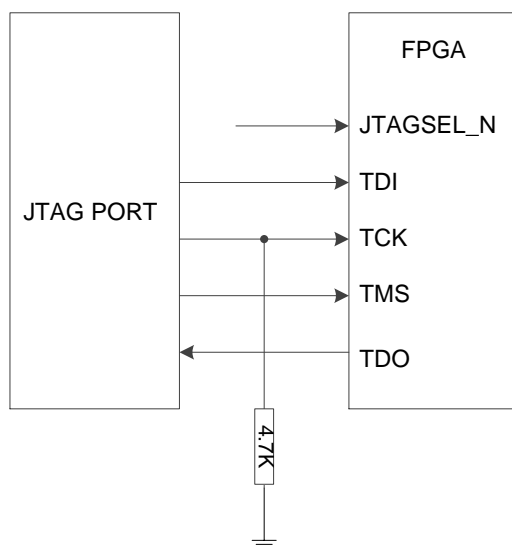
**表 2-1 JTAG 配置模式管脚定义**

管脚名称	I/O 类型	说明
JTAGSEL_N <sup>1</sup>	I, 内部弱上拉	将 JTAG 管脚从 GPIO 恢复为配置管脚, 低电平有效
TCK <sup>2</sup>	I	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	JTAG 串行数据输出

注!

- [1]只有当 JTAG 管脚设置为 RECOVERY 状态并且器件完成启动过程后, JTAGSEL\_N 信号方有效;
- [2]TCK 信号需在 PCB 上连接 4.7K 下拉电阻。

图 2-1 JTAG 配置模式连接示意图

**注!**

- 对于 JTAGSEL\_N 未封装出的器件，用户在调试 JTAG 管脚复用的案例时，建议上电前将 MODE 值设置为非自动配置的模式（自启动、双启动和 MSPI）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；
- JTAG 配置模式时钟频率上限不应高于 25MHz。

## 2.2 配置流程

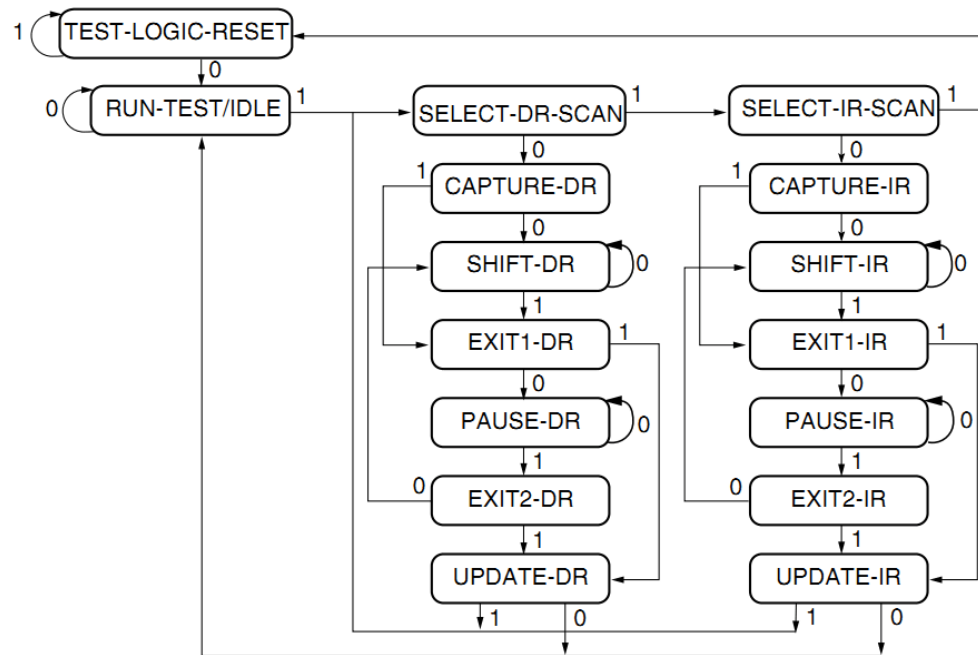
### 2.2.1 JTAG 引脚定义

- TCK: 测试时钟输入。在 TCK 的上升沿采样 TMS 和 TDI 的数据, 在时钟的下降沿将数据输出至 TDO;
- TMS: 测试模式选择, TMS 用来设置 JTAG 口处于某种特定的测试模式;
- TDI: 测试数据输入, 数据通过 TDI 输入 JTAG 口;
- TDO: 测试数据输出, 数据通过 TDO 从 JTAG 口输出。

### 2.2.2 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 2-2 所示。

图 2-2 TAP 状态机



### 2.2.3 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。

## 2.2.4 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，传送指令寄存器，如图 2-3 所示。在数据寄存器扫描操作中，在 Shift\_DR 状态时，传送数据寄存器，如图 2-4 所示。

图 2-3 指令寄存器访问时序

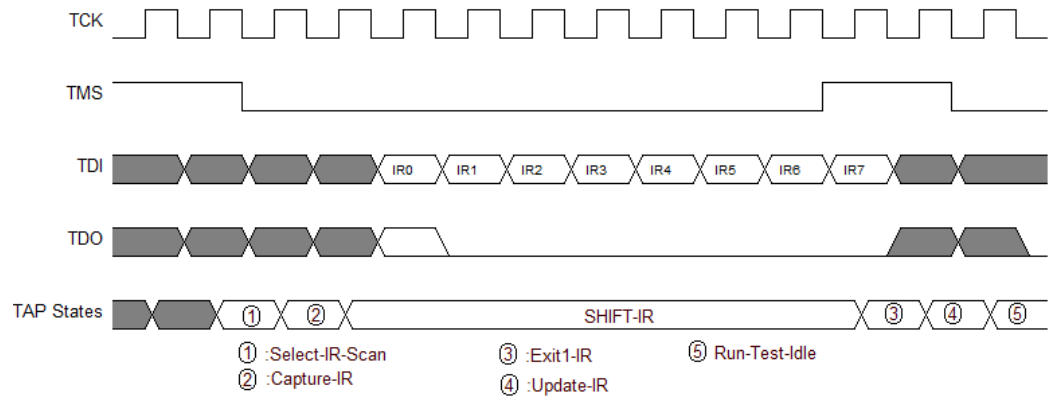
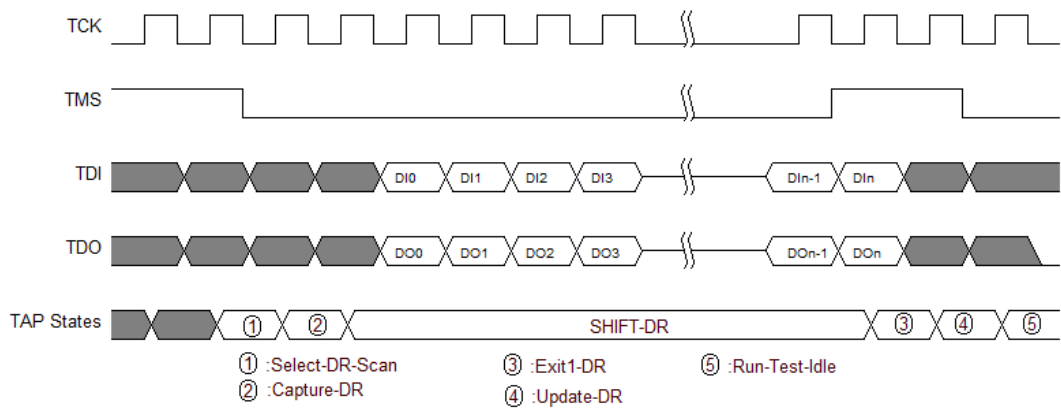


图 2-4 数据寄存器访问时序



注！

- 在高云半导体 GW1N(R)、GW2A(R)系列 FPGA 中，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。

## 2.2.5 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 2-2 Gowin FPGA IDCODE

Gowin FPGA Device Family IDCODE			
Device Family	Device Part	Manufacturer ID	IDCODE
	Bits 31-12	Bits 11-0 h81B	
GW1N-1	h09002	h81B	h0900281B
GW1N-2	h01001	h81B	h0100181B
GW1N(R)-4	h11003	h81B	h1100381B
GW1N(R)-4ES	h01003	h81B	h0100381B
GW1N-6	h01004	h81B	h0100481B
GW1N-9	h01005	h81B	h0100581B
GW2A(R)-18	h00000	h81B	h0000081B
GW2A-55	h00002	h81B	h0000281B

高云 FPGA 读取 FPGA 的指令是 0x11, 以下步骤以读取 GW1N-4ES 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 2-3 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 2-6 所示。

表 2-3 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 (0x01)	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, 32 个时钟移出的数据即为 ID Code 0x0100381B, 如图 2-7 所示;
6. 移动状态回到 Run-Test-Idle。

图 2-5 读取 ID Code 状态机流程图

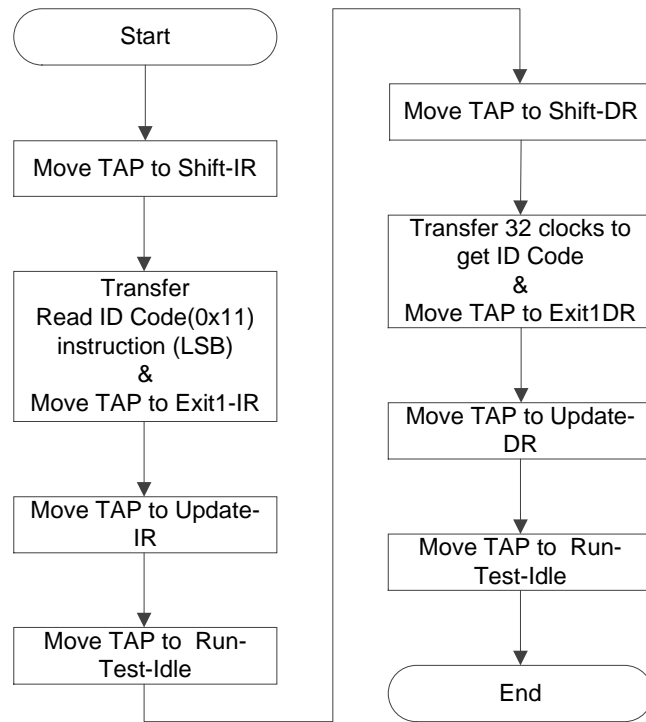


图 2-6 读取 ID Code 指令-0x11 访问时序

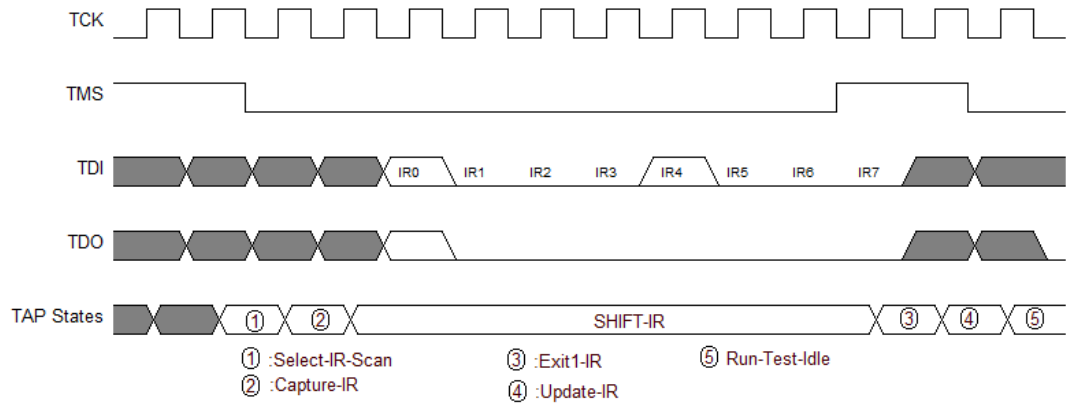
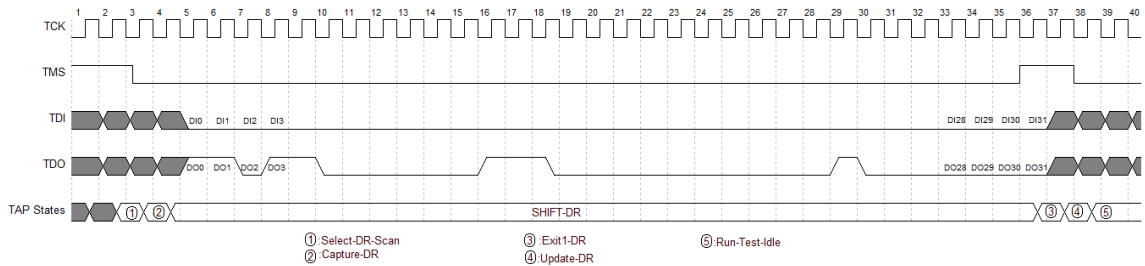


图 2-7 读取 ID Code 数据寄存器访问时序



## 2.2.6 配置 SRAM 流程

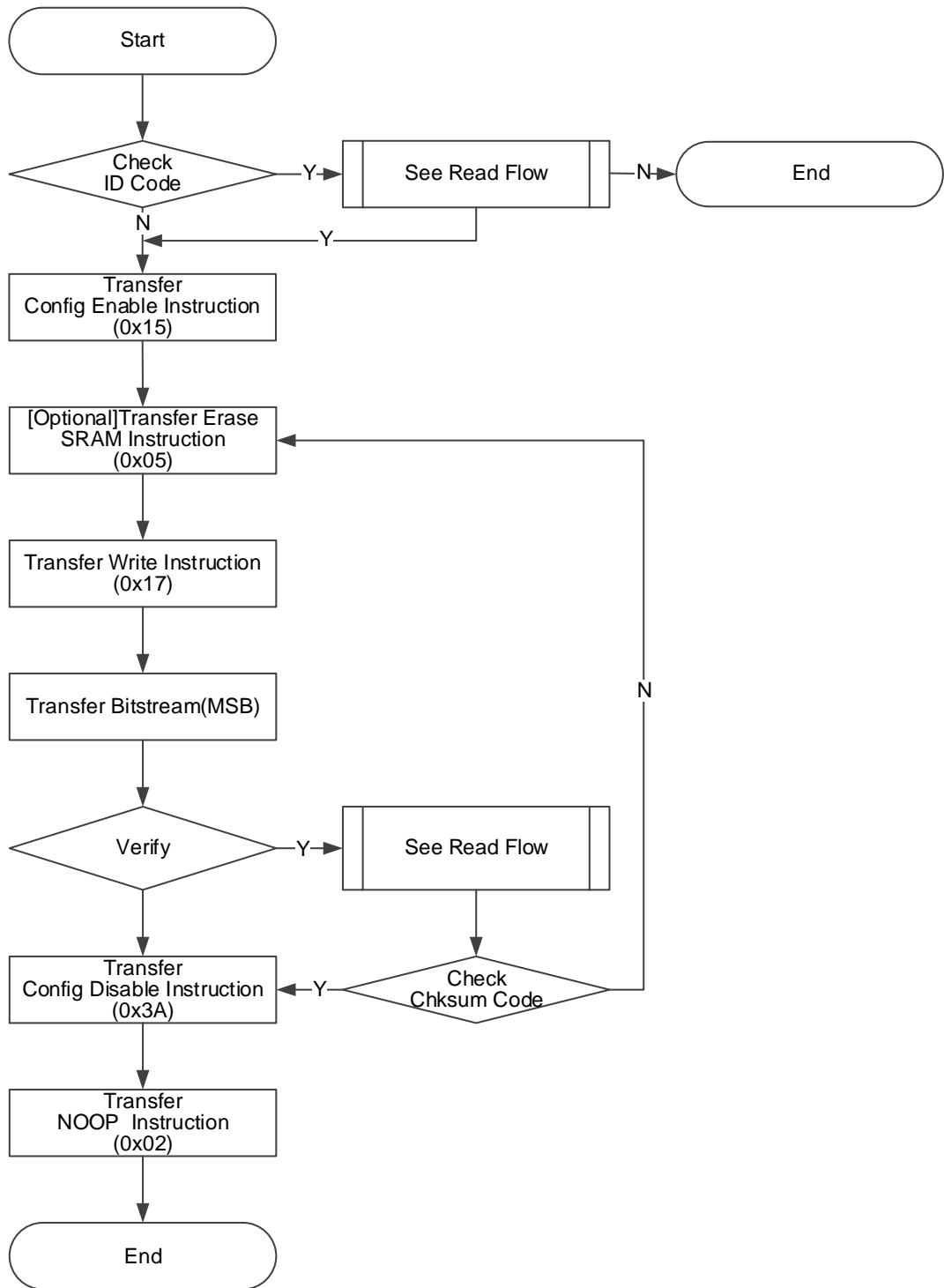
通过外部 Host 配置 FPGA SRAM, 使 FPGA 实现功能, 配置 SRAM 不受 Configuration Mode Pins 的影响。

通过 Gowin Designer 设计软件生成包含 Configuration Data 的 FS 文件, 利用 JTAG 1149.1 实现 SRAM 的配置, 下面介绍外部 Host 配置 SRAM 的过程, 如图 2-8 所示。

1. 建立 JTAG 链路, TAP 复位;
2. 读取设备 IDCODE, 检查是否匹配, 此步可忽略;
3. 移动状态机到 Shift-IR (指令寄存器), 发送 ConfigEnable 指令 0x15, 并回到 Run-Test-Idle, 这一步同 2.1.5 节里提到的发送 ReadIDCode 指令 h11 时序完全相同。发送完成后, FPGA 将进入 Edit Mode, 此时 Config Done 信号将被置为高电平;
4. 移动状态机到 Shift-IR, 发送 Transfer Configuration Data 指令 0x17, 时序同上。最后状态机回到 Run-Test-Idle 状态;
5. 移动状态到 Shift-DR (数据寄存器), 将 Configuration Data 从最高位开始, 逐位发送, 最后一位发送前, 拉高 TMS, 使状态机移动到 Exit1-DR;
6. 移动状态机回到 Run-Test-Idle 状态;
7. 如需回读 Configuration Data 进行校验, 请参考 2.2.7 读取 SRAM 的流程;
8. 移动状态机到 Shift-IR, 发送 ConfigDisabled 指令 0x3A, 完成后状态机回到 Run-Test-Idle;
9. 移动状态机到 Shift-IR, 发送 Noop 指令 0x02, 结束配置流程。



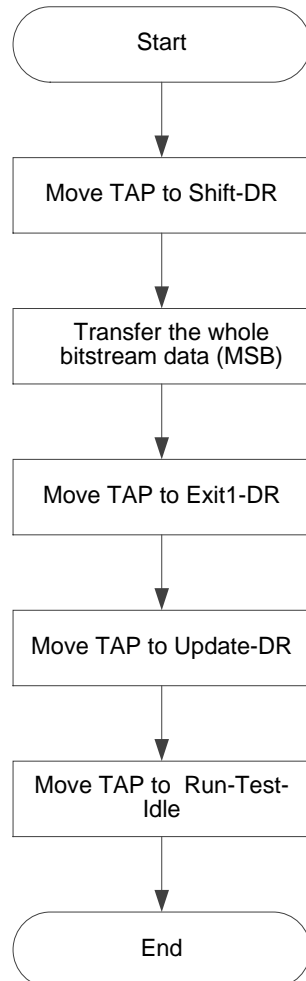
图 2-8 配置 SRAM 流程



### Transfer Configuration Data过程 (Non-JTAG Mode)

将数据流文件(Configuration Data)全部在 SHIFT-DR 中完成传送。

图 2-9 Transfer Configuration Data 过程示意



## 2.2.7 读取 SRAM 的流程

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)。

数据校验采用 Checksum-16 计算方式，即取每 16 位进行累加，最后结果取低 16 位。

**表 2-4 器件 SRAM 地址数量和地址长度**

Device	Length of one address (bits/address)	Count of address
GW1N-1	1216	274
GW1N-2	2296	494
GW1N(R)-4(ES)	2296	494
GW1N-6	2836	712
GW1N(R)-9	2836	712
GW2A(R)-18	3376	1342
GW2A-55	5536	2038

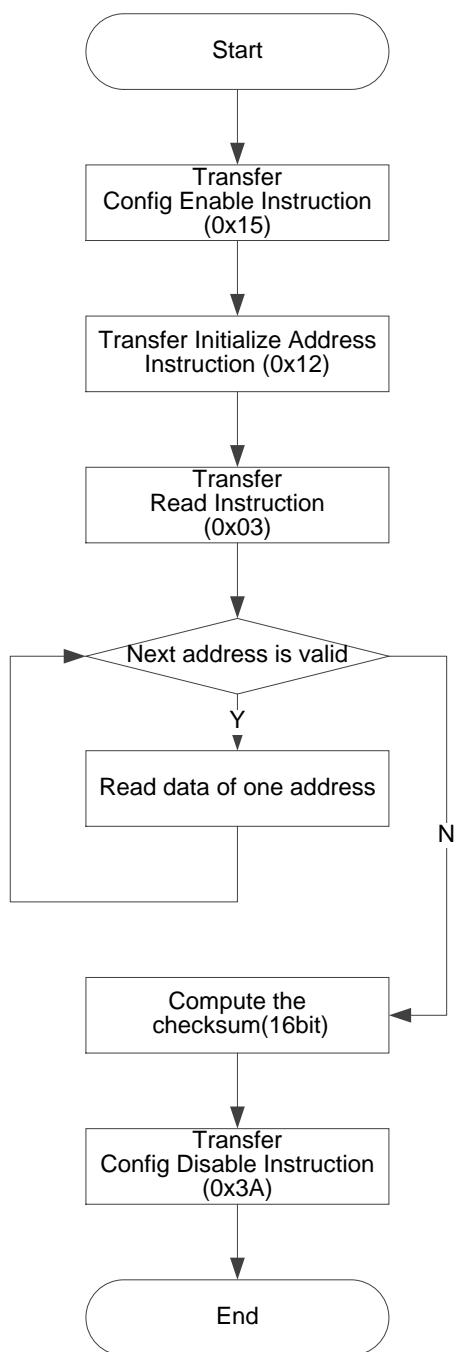
下面详细介绍读取流程，如图 2-10 所示。

1. 如果是接 2.2.6 流程，请跳转至步骤 3；
2. 移动状态机到 Shift-IR (指令寄存器)，发送 ConfigEnable 指令 0x15，并回到 Run-Test-Idle；
3. 移动状态机到 Shift-IR (指令寄存器)，发送 Address Initialize 指令 0x12，并回到 Run-Test-Idle，确保从地址 0 开始读取；
4. 移动状态机到 Shift-IR (指令寄存器)，发送 SRAM Read 指令 0x03，并回到 Run-Test-Idle；
5. 移动状态机到 Shift-DR (数据寄存器)，发送到一个 SRAM 长度单位的时钟周期，请参见表 2-4。在发送最后一个时钟同时拉高 TMS，跳到 Exit1-DR，此时 TDO 产生相应长度的 Configuration Data，这是一个地址长度的数据。最后回到 Run-Test-Idle；
6. 重复步骤 5，每次读取一个地址的数据，其地址会自动累加；
7. 移动状态机到 Shift-IR，发送 ConfigDisabled 指令 0x3A，完成后状态机回到 Run-Test-Idle；
8. 移动状态机到 Shift-IR，发送 Noop 指令 0x02，结束读取流程；
9. 将读取的数据从最高位开始，计算 Checksum，并与相应 FS 文件中的 Checksum 比对。相同表示数据相同。

### 注！

步骤 7、8 与步骤 9 无先后顺序。

图 2-10 读取 SRAM 的流程



## 2.2.8 擦除内部 Flash

高云 GW1N 系列内置 Flash 存储器，在每次编程之前需要先擦除内置 Flash，为数据安全，内置 Flash 只提供整片擦除的操作。

当前，GW1N 系列内置 Flash 因工艺不同，对 JTAG 编程频率有不同要求，请参见表 2-5。

**表 2-5 JTAG 的 TCK 频率要求**

器件	TCK 频率范围
GW1N-1	1.4Mhz ~ 5Mhz
GW1N(R)-2/4(ES)	2.1Mhz $\pm$ 5%
GW1N(R)-6/9	2.5Mhz $\pm$ 5%

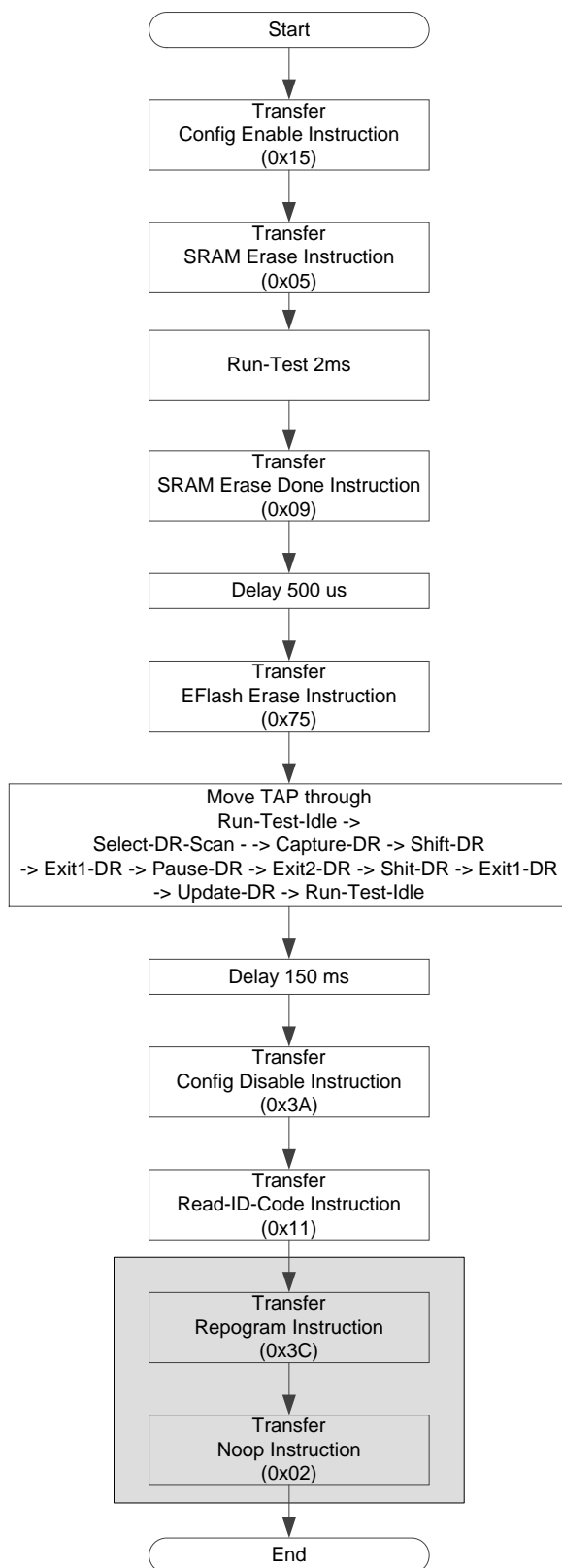
下面详细介绍 GW1N-2/4/6/9 系列芯片的擦除流程，如图 2-11 所示。

1. 建立 JTAG 链路，TAP 复位；
2. 读取设备 IDCODE，检查是否匹配，此步骤可选；
3. 移动状态机到 Shift-IR（指令寄存器），发送 ConfigEnable 指令 0x15，并回到 Run-Test-Idle，这一步同 2.2.5 节所述的发送 ReadIDCode 指令 h11 时序完全相同。发送完成后，FPGA 将进入 Edit Mode，此时 Config Done 信号将被置为高电平；
4. 如果 SRAM 未被配置过（包括从内外 Flash 配置），步骤 5、6 可以忽略；
5. 移动状态机到 Shift-IR（指令寄存器），发送 SRAM Erase 指令 0x05，并回到 Run-Test-Idle；
6. 移动状态机到 Shift-IR（指令寄存器），发送 Noop 指令 0x02，并回到 Run-Test-Idle。使前一指令生效；
7. 在 Run-Test-Idle 持续产生时钟，持续时间为 2ms；
8. 移动状态机到 Shift-IR（指令寄存器），发送 SRAM Erase Done 指令 0x09，并回到 Run-Test-Idle；
9. 移动状态机到 Shift-IR（指令寄存器），发送 Noop 指令 0x02，并回到 Run-Test-Idle。使前一指令生效；
10. 在 Run-Test-Idle 持续产生时钟，持续时间为 500  $\mu$ s；
11. 移动状态机到 Shift-IR（指令寄存器），发送 EFlash Erase 指令 0x75，并回到 Run-Test-Idle；
12. 依次移动状态机：Run-Test-Idle -> Select-DR-Scan-> Update-DR -> Capture-DR -> Shift-DR -> Transfer 32 bits-> Exit1-DR -> Update-DR -> Run-Test-Idle；
13. 在 Run-Test-Idle 持续产生时钟，持续时间为 120ms；
14. 移动状态机到 Shift-IR，发送 ConfigDisabled 指令 0x3A，完成后状态机回到 Run-Test-Idle；
15. 移动状态机到 Shift-IR，发送 Noop 指令 0x02；
16. 移动状态机到 Shift-IR，发送 Repogram 指令 0x3C，完成后状态机回到 Run-Test-Idle；
17. 移动状态机到 Shift-IR，发送 Noop 指令 0x02。

GW1N-1 芯片的擦除流程,与 GW1N-2/4/6/9 有所不同,如图 2-12 所示,该过程如下:

1. 从 1 到 11 步与 GW1N-2/4/6/9 的流程相同;
2. 略;
3. 略;
4. 略;
5. 略;
6. 略;
7. 略;
8. 略;
9. 略;
10. 略;
11. 略;
12. 移动状态机,从 Run-Test-Idle 到 Shift-DR,产生 32 个时钟 (TDI 信号保持低电平即可)。在第 32 个时钟同时移动状态机到 Exit1DR,再经过 Update-DR 回到 Run-Test-Idle;
13. 重复步骤 12,重复 65 次;
14. 在 Run-Test-Idle 持续产生时钟,持续时间为 150ms;
15. 与 GW1N-2/4/6/9 流程的 14 相同,依次类推,至流程结束。

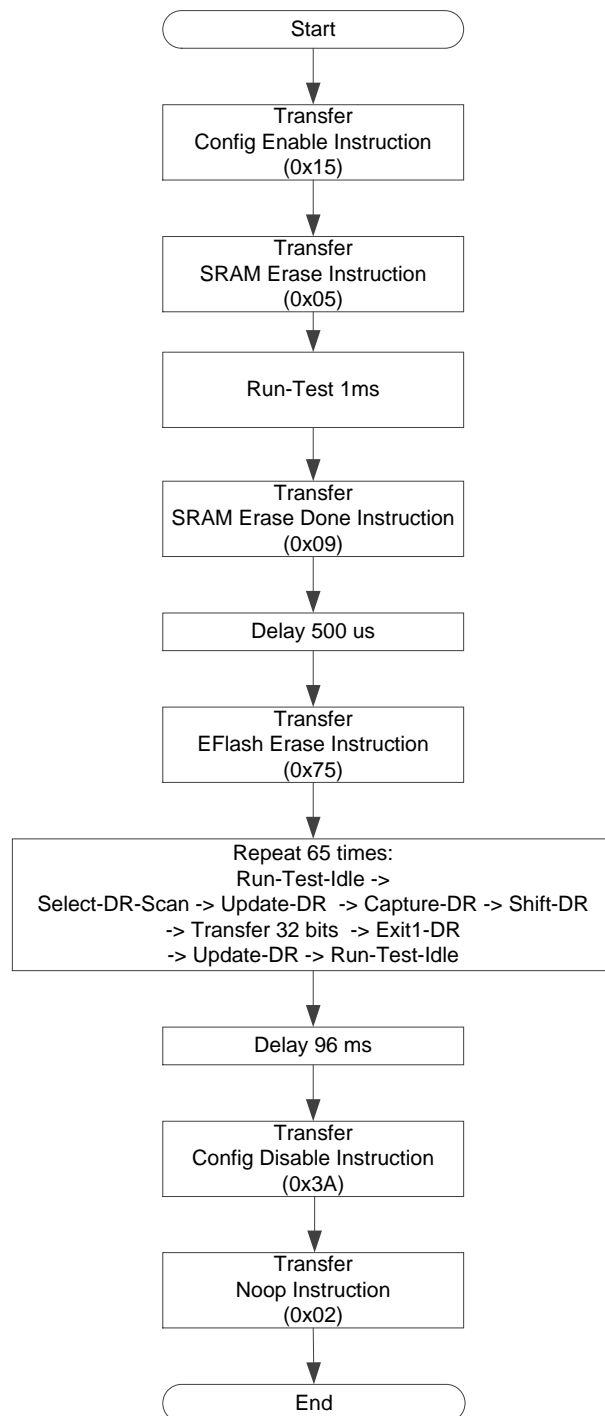
图 2-11 擦除 GW1N-2/4/6/9 内部 Flash 流程



注!

Background Programming 时, 忽略底纹区域操作。

图 2-12 擦除 GW1N-1 内部 Flash 流程



## 2.2.9 编程内部 Flash 流程

内置 Flash 以 256Bytes 为一个 X-page, 每个 X-page 分成 64 个 Y-page, 每 Y-page 包含 4Bytes。

第一个 X-page 的第一个 Y-page, 用于标识 Flash 是否可以具备 Autoboot (自动加载) 功能或回读功能。如表 2-6 所示。当第一个 Y-page 写入 Readable-pattern 后, 可读取 Flash 数据; 当第一个 Y-page 写入



Autoboot-pattern 后，器件在 autoboot mode 下会自动把 Flash 数据加载到 SRAM 中；只有写入 Readable-pattern 后才能读取 Flash，其他情况均不能读取。具备 Background programming 功能的器件，仅需使用 Autoboot-pattern。

当前，GW1N 系列内置 Flash 因工艺不同，对 JTAG 编程频率有不同要求，请参见 [2.2.8 擦除内部 Flash](#)> [表 2-5 JTAG 的 TCK 频率要求](#)。

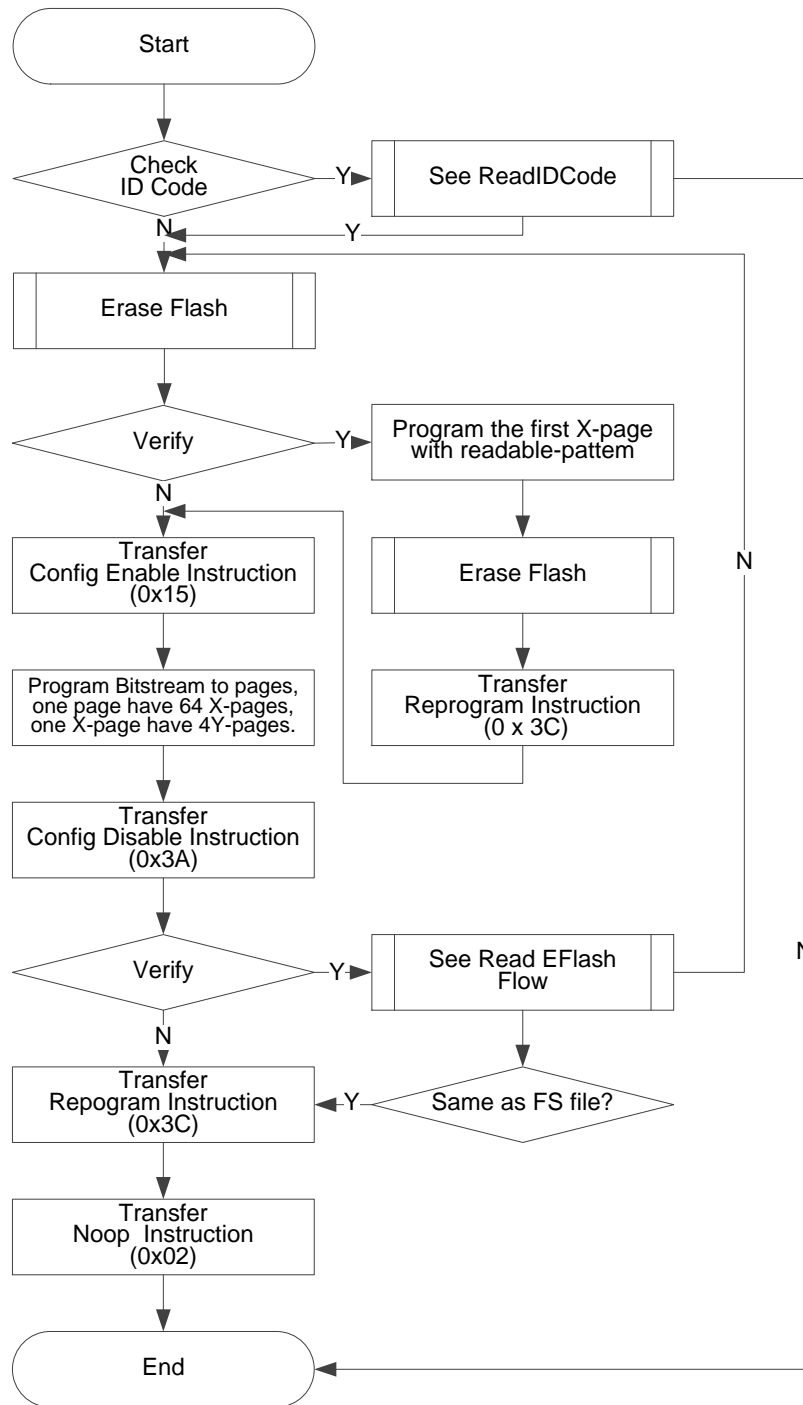
**表 2-6 Readback-pattern / Autoboot-pattern**

Device	Readable-pattern(4 Bytes)	Autoboot-pattern(4 Bytes)
GW1N-1	0x07,0x07,0x30,0x40	0x47,0x57,0x31,0x4E
GW1N(R)-2/4/6/9	0xF7,0xF7,0x3F,0x4F	

编程内部 Flash 流程如图 2-13 所示：

1. 检查 IDcode 是否匹配；
2. 擦除 Flash；
3. 验证是否擦除成功，可通过读取 Status 寄存器，看器件是否已还原为裸片的初始状态；
4. 发送 ConfigEnable 指令，进入 EditMode；
5. 以 X-page 为单位，地址递增，直至烧录完成；
6. 发送 ConfigDisable 指令，退出 EditMode；
7. 通过读取 StatusCode 验证；
8. 发送 Reprogram 指令，使器件加载 Flash 的数据到 SRAM；
9. 读取 UserCode 验证是否加载成功。

图 2-13 编程内部 Flash 流程图



## 编程一个X-page流程

编程一个 X-page 流程如下描述，如

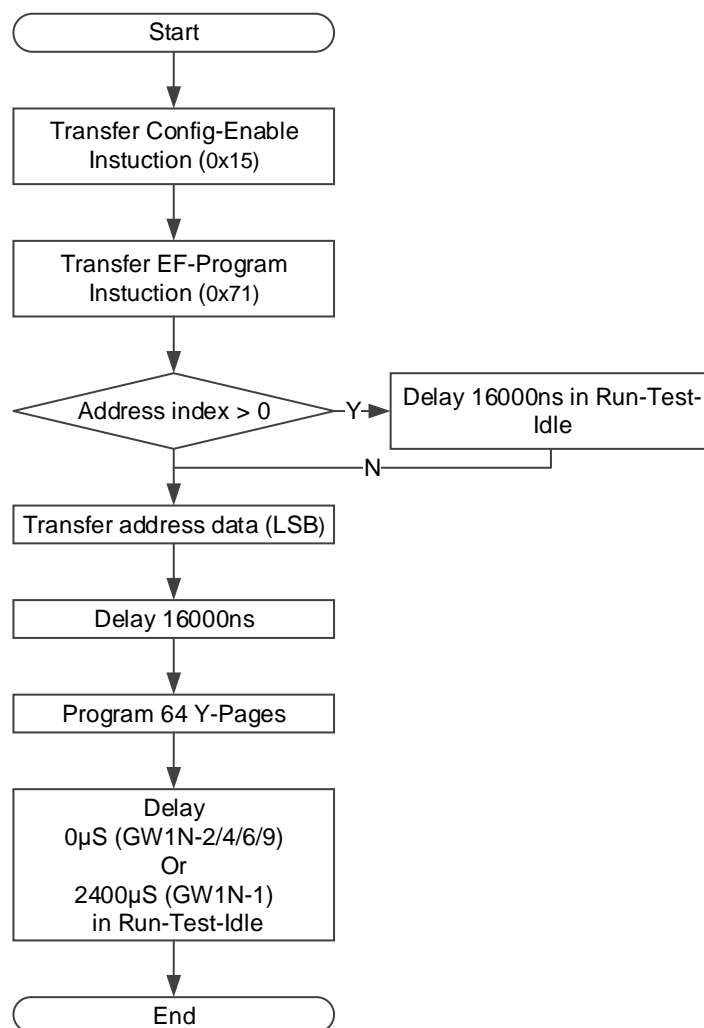
图 2-14 所示。

1. 移动状态机到 Shift-IR (指令寄存器)，发送 ConfigEnable 指令 0x15，并回到 Run-Test-Idle，这一步同 2.2.5 节所述的发送 ReadIDCode 指令 h11 时序完全相同。发送完成后，FPGA 将进入 Edit Mode，此时 Config Done 信号将被置为高电平；
2. 发送 EF-Program 指令；
3. 写入 X-page 地址的数据 (如果地址大于 0，需要 Run-Test 16000ns)；
4. 编程 256Bytes，分 64 次，每次编程 4Bytes，每一次构成了编程一个 Y-page；
5. GW1N-1 期间在此 Run-Test 2400us；  
本次 X-page 编程完毕。

### 注!

地址数据格式共 32bits，其中低 6 位保留，例如地址为 b'00010011 (0x13) 时，写入的地址为 b' 00000000000000000000000010011000000，该地址数据遵循 LSB 方式写入。

图 2-14 X-page 编程流程图



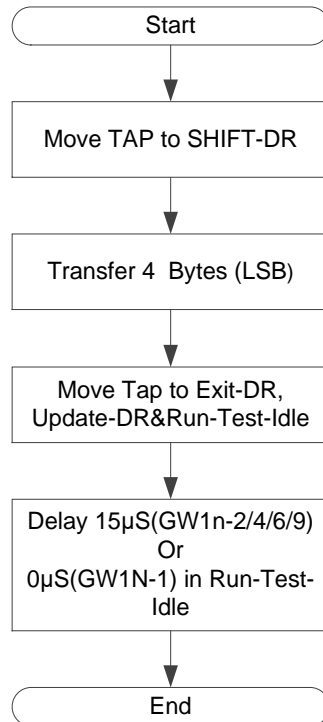
### 编程一个Y-page流程

Y-page 编程是烧录过程的最小单位,每次写入 4Bytes,如图 2-15 所示。

注!

数据从 Configuration Data 取高位 4Bytes,在 Shift-DR 写数据时要从最低位开始写入(LSB)。

图 2-15 Y-page 编程流程图



### 2.2.10 读取内部 Flash 流程

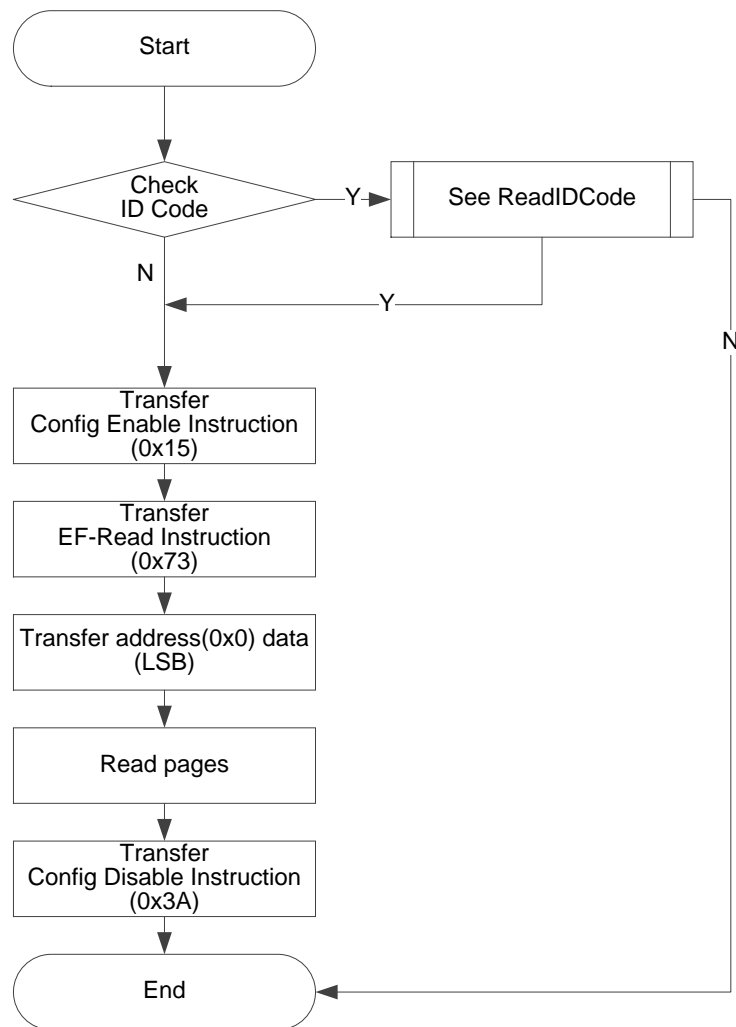
读取内部 Flash 流程概览,对 JTAG 的 TCK 没有速率要求。如图 2-16 所示。

读取内部 Flash 可以理解是烧录 flash 的逆向过程,但首先要确保写入的 Readable-pattern 已经生效。对于 GW1N 而言,写入 Readable-pattern 后依次发送 Reprogram(0x3C)和 Noop(0x02)可使内部 flash 处于 Readable 状态。

流程简述:

1. 校验 IDCode (可选);
2. 发送 ConfigEnable 指令 0x15, FPGA 将进入 Edit Mode;
3. 发送 EF-Read 指令 0x73;
4. 发送读 Flash 起始地址 0x0; 方法同 2.2.9 中写 X-address 相同;
5. 每读 64 个 Y-page 就是一个 X-page;
6. 每次读完一个 X-page 并不需要重新发送地址,其地址会自动递归;
7. 读取完毕后,发送 ConfigDisble 指令 0x3A 结束流程。

图 2-16 读取内部 Flash 流程图

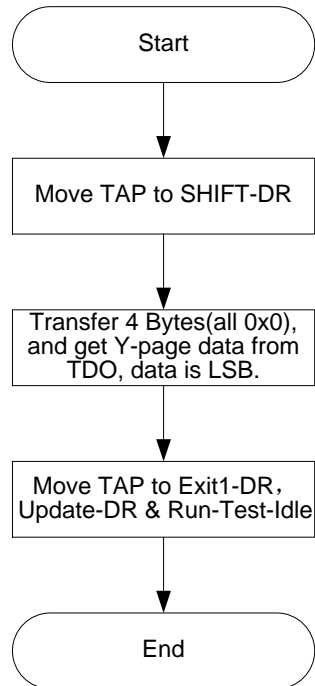


### 读取一个Page (Y-page) 的过程

与写一个 Y-page 相似，但无写入 Flash 的等待时间，如图 2-17 所示。

数据最先输出的是数据最低位。

图 2-17 读取一个 Y-page 的过程



### 2.2.11 背景烧录 (Background Programming)

设备有时需要在不影响当前功能的情况下升级数据文件，对 Flash 进行烧录。并且在加载新的数据流文件时，能够保持 IO 状态。下图是 GW1N4 使用背景烧录技术 (Background Programming) 升级内置 Flash 数据的流程示意图。

图 2-18 GW1N4 Background Programming 流程图

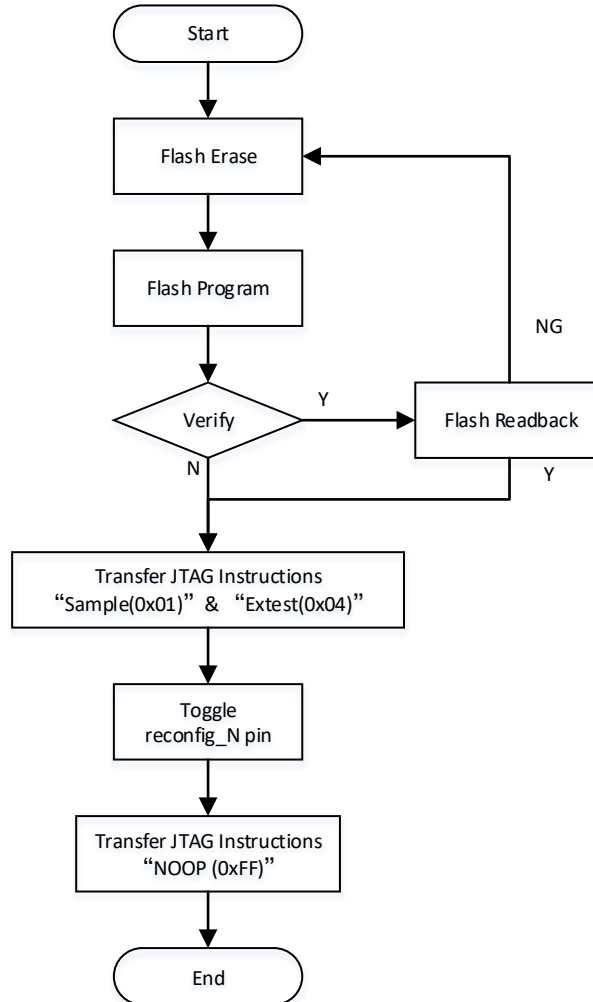
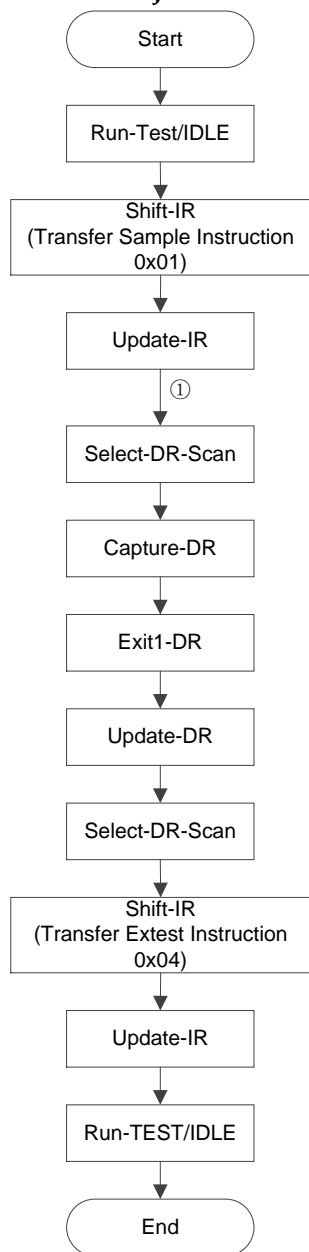




图 2-19 Transfer JTAG Instruction Sample &amp; Extest 流程图



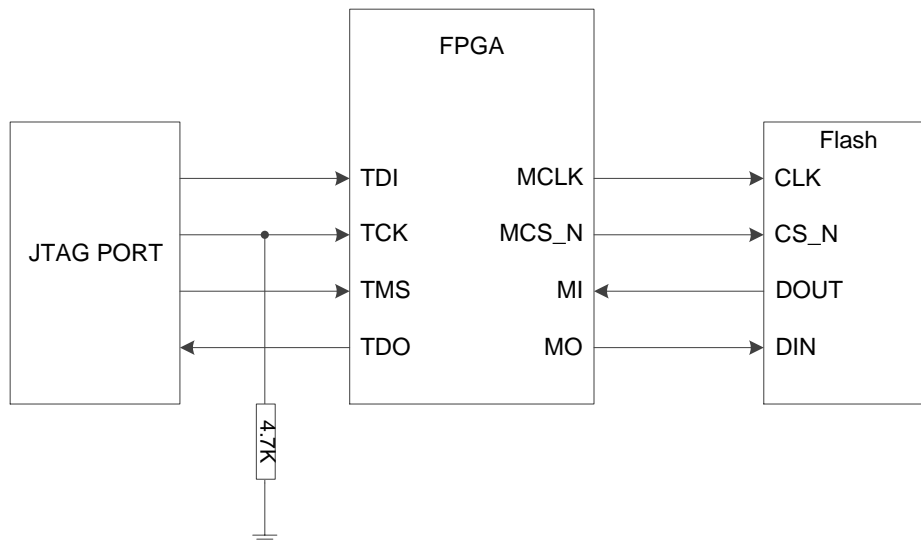
注!

①处直接从 Update-IR 跳入 Select-DR-Scan。

## 2.2.12 编程外部 Flash

高云 FPGA 可从外部 Flash 中加载数据流文件，可以通过 JTAG 直接烧录外部 Flash。

图 2-20 JTAG 接口编程外部 Flash 连接示意图



### 注!

此图为 JTAG 接口编程外部 Flash 的最小系统图，MODE 值设置为“011”（边界扫描操作编程外部 Flash 不必关心 MODE 值）。

### 采用config-mode[2:0]=011模式烧录SPI Flash

此模式通过 JTAG 接口编程外部 Flash 需要设置特定的 MODE[2:0]值为 011。

此模式的原理是将 JTAG 的接口以转发的形式接入 flash 的接口，用户在 JTAG 端操作状态机模拟 SSPI 时序对 SPI Flash 进行编程。

图 2-21 采用 config-mode[2:0]=011 模式编程 SPI Flash 流程示意图

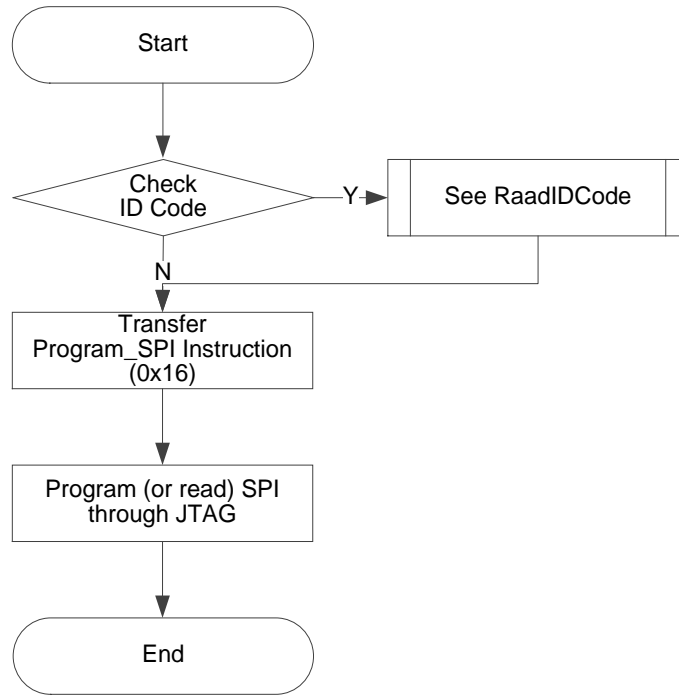


图 2-22 GW2A 系列 JTAG 模拟 SPI 发送 0x06 指令时序图

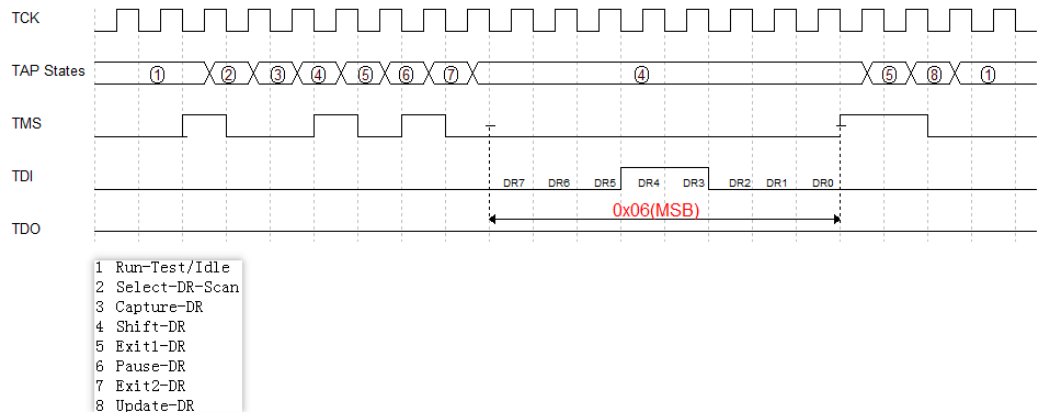
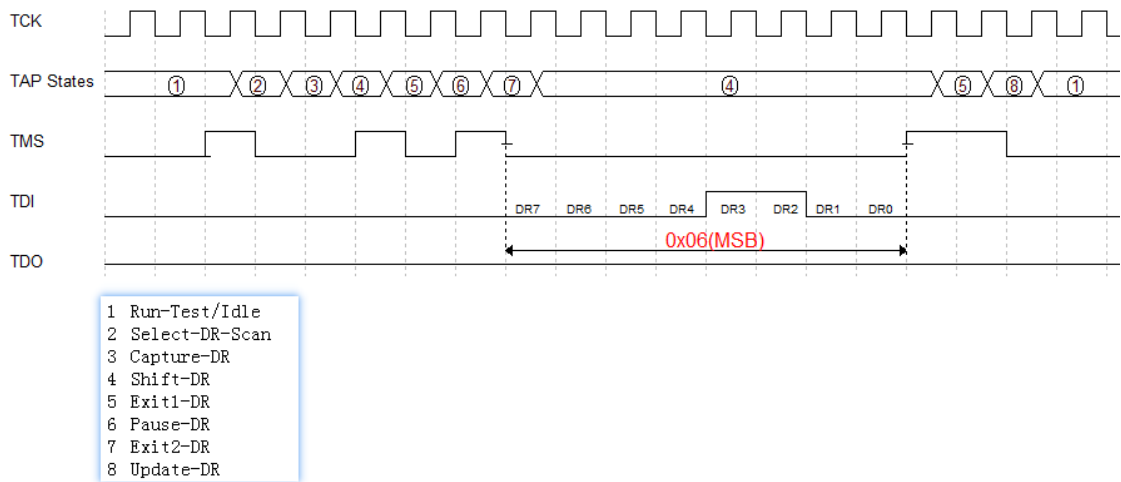


图 2-23 GW1N 系列 JTAG 模拟 SPI 发送 0x06 指令时序图



### 采用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 2-7 所示，每发送两次 Boundary Scan Chain 完成一次 SCLK 驱动。

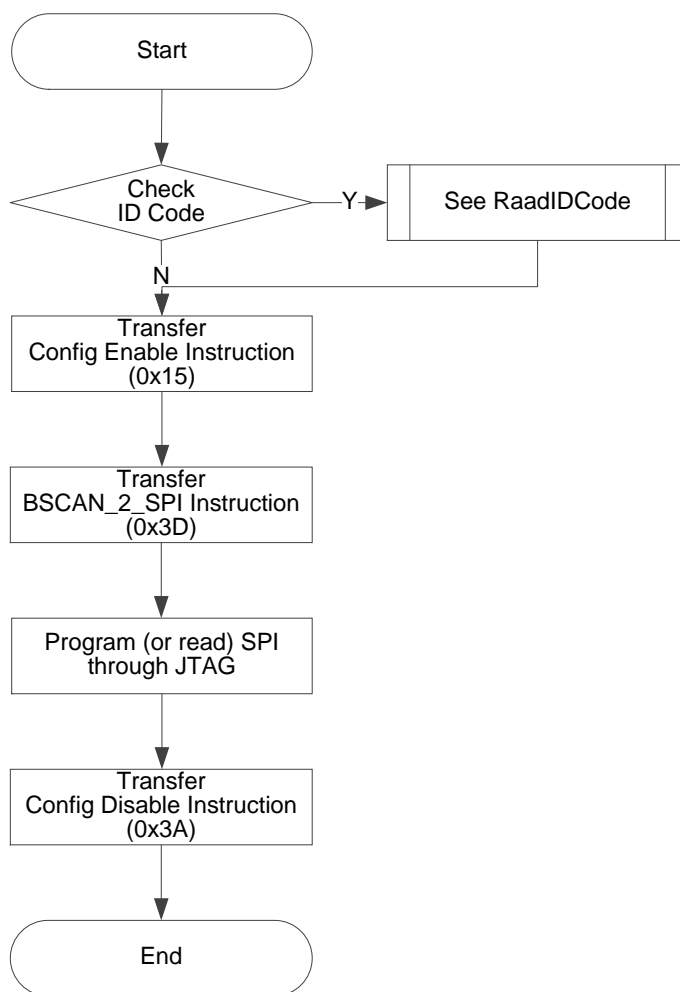
表 2-7 管脚状态

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 2-24 采用 Boundary Scan 模式编程 SPI Flash 流程示意图



### 2.2.13 读取 Status Register 0x41

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 timeout 错误、id 校验错误、crc 校验错误等。

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 2-8 所示。

表 2-8 Status Register 含义

Device Status Register[31:0]	GW1N(R)-1/2/4	GW1N(R)-6/9	GW2A-18/55
0	CRC error		
1	Bad command error		
2	ID verify failed error		
3	Timeout error		
4	0		
5	Memory erase		
6	Preamble		
7	System Edit mode		
8	Program spi flash derectly		
9	0	autoboot state	0
10	Non-jtag configuration is active		
11	Bypass state		
12	Gowin VLD		0
13	Done final state		
14	Security bit final state		
15	Init done		Encrypted format
16	POR end		Encrypted key is right
17	0	Flash lock	0
18-31	0		0

### 2.2.14 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。

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

### 2.2.15 重加载 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 Reprogram (0x3C) 指令、Noop (0x02) 指令, 可使器件重加载, 效果同触发 Reconfig\_N 管脚。

### 2.2.16 擦除 SRAM 0x15

该指令作用是擦除 SRAM。

通过 JTAG 依次发送 ConfigEnable (0x15) 指令、EraseSram (0x05) 指令、Noop (0x02)、ConfigDisable (0x3A) 指令、Noop (0x02) 指令, 可擦除 SRAM 区域。

#### 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:

- GW1N-1 参考时间为 1ms;
- GW1N-2/4 参考时间为 2ms;
- GW1N-6/9 参考时间为 4ms。

# 3 例程文件

关于 JTAG 例程文件，请联系公司技术支持或当地办事处。



