




Gowin goConfig I2C IP

用户指南

IPUG795-2.1,2023-09-27

版权所有 © 2023 广东高云半导体科技股份有限公司

GOWIN高云, , Gowin, 云源以及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, 其所有权利属其所有者所有。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、翻译本文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
2022/03/09	1.0	初始版本。
2022/10/17	2.0	<ul style="list-style-type: none">● 更新支持器件；● 完善 GW1N(R)-2C 系列相关描述；● 完善 GW2AN-9X 系列相关描述。
2023/09/27	2.1	<ul style="list-style-type: none">● 完善部分描述；● 新增第 6 章实例说明。

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.....	2
2 性能简介	3
2.1 升级功能概述	4
2.2 I ² C 接口	4
2.2.1 I ² C 基本特性.....	4
2.2.2 I ² C 时序特性.....	5
2.3 IP 特性	6
2.4 资源占用.....	7
3 功能描述	8
3.1 goConfig I2C 结构框图	8
3.2 Gowin goConfig I2C IP 外部电路连接	9
3.3 Gowin goConfig I2C IP 背景升级控制指令	10
3.3.1 2C 系列	10
3.3.2 9X 系列	15
4 信号定义	21
4.1 信号定义.....	21
5 IP 使用说明.....	24
5.1 2C 系列	24
5.1.1 软件配置.....	24

5.1.2 流程说明.....	25
5.1.3 注意事项.....	27
5.2 9X 系列	27
5.2.1 软件配置.....	27
5.2.2 流程说明.....	29
5.2.3 注意事项.....	30
6 实例说明	32
6.1 2C 系列实例	32
6.1.1 检验 MCU 与 IP 通信是否正常.....	32
6.1.2 背景升级整体流程	33
7 界面配置	38
8 参考设计	41
8.1 参考设计.....	41
8.1.1 FPGA 工程说明.....	41
8.1.2 参考设计文件	42
8.1.3 参考设计注意事项	43

图目录

图 2-1 I ² C 7-bit 典型时序示例	5
图 2-2 I ² C 接口时序	5
图 3-1 结构框图（2C 系列）	8
图 3-2 结构框图（9X 系列）	9
图 3-3 外部电阻电路连接关系	9
图 3-4 读状态寄存器指令示例	12
图 3-5 复位 IP 指令示例	12
图 3-6 回读指令发送示例	13
图 3-7 读 FIFO 数据指令示例	13
图 3-8 等待指令示例	14
图 3-9 发送数据且不回读 Flash 数据指令示例	14
图 3-10 Reconfign 指令示例	14
图 3-11 IP 版本指令	14
图 3-12 CSCR 寄存器使用示例	16
图 3-13 DICN 寄存器写使用示例	16
图 3-14 DICN 寄存器读使用示例	16
图 3-15 DOCN 寄存器写使用示例	17
图 3-16 DOCN 寄存器读使用示例	17
图 3-17 DIFR 寄存器使用示例	17
图 3-18 SRSTR 寄存器使用示例	18
图 3-19 SR 寄存器使用示例	19
图 3-20 DOFR 寄存器使用示例	20
图 5-1 打开 Configuration 选项	24
图 5-2 配置背景升级为 I2C	25
图 5-3 打开 Configuration 选项	28
图 5-4 配置背景升级为 I2C	28
图 5-5 IO 定义示例	31
图 6-1 结构框图	32

图 6-2 检验 MUC 与 IP 通信的流程图.....	33
图 6-3 背景升级流程图	34
图 6-4 擦除内嵌 Flash 流程图.....	35
图 6-5 编程内嵌 Flash 流程图.....	37
图 7-1 IP Core Generator 选项	38
图 7-2 打开 goConfig I2C IP 核	39
图 7-3 goConfig I2C IP 配置界面.....	40
图 8-1 参考设计-基本结构示意图.....	42

表目录

表 1-1 术语、缩略语	2
表 2-1 Gowin goConfig I2C IP 概述	3
表 2-2 2C 系列 I ² C 接口时序说明	5
表 2-3 9X 系列 I2C 接口时序说明	6
表 2-4 2C 系列资源消耗参考	7
表 2-5 9X 系列资源消耗参考	7
表 3-1 2C 系列指令定义	10
表 3-2 状态寄存器	11
表 3-3 寄存器定义	15
表 3-4 CSCR-Data 定义	15
表 3-5 DICN-Data 定义	16
表 3-6 DOCN-Data 定义	17
表 3-7 DIFR-Data 定义	17
表 3-8 SRSTR-Data 定义	18
表 3-9 SR-Data 定义	18
表 3-10 DOFR-Data 定义	19
表 4-1 2C 系列 IO 接口定义	21
表 4-2 9X 系列 IO 接口定义	22
表 5-1 对照表	30
表 8-1 2C 系列参考设计 src 文件夹内容列表	42
表 8-2 9X 系列参考设计 src 文件夹内容列表	42

1 关于本手册

1.1 手册内容

Gowin® goConfig I2C IP 用户指南主要内容包括性能简介、功能描述、信号定义、使用说明、实例说明，界面配置，旨在帮助用户快速了解 goConfig I2C IP 的产品特性、特点及使用方法。

注！

Gowin goConfig I2C IP 为匹配不同器件特性，功能上略有差别。目前器件 GW1N-1P5C/GW1N(R)-2C 系列（以下简称 2C 系列）为一类，GW2AN-9X/GW2AN-18X 系列（以下简称 9X 系列）为一类，文档将分别描述功能与特性。如果文档中没有明确区分器件，则说明为共有特性或功能。

1.2 相关文档

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

- [DS100, GW1N 系列 FPGA 产品数据手册](#)
- [DS117, GW1NR 系列 FPGA 产品数据手册](#)
- [DS971, GW2AN-18X & 9X 器件数据手册](#)
- [SUG100, Gowin 云源软件用户指南](#)
- [SUG283, Gowin 原语用户指南](#)
- [UG290, Gowin FPGA 产品编程配置手册](#)

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	含义
ACK	Acknowledge	响应
FPGA	Field Programmable Gate Array	现场可编程门阵列
I ² C(I2C)	Inter-Integrated Circuit	两线式串行总线
IP	Intellectual Property	知识产权
NACK	Not Acknowledge	不响应
SRAM	Static Random Access Memory	静态随机存储器

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

E-mail：support@gowinsemi.com

Tel: +86 755 8262 0391

2 性能简介

背景升级是近年来用户需求很高的功能。在使用 Gowin FPGA 产品上存在“功能 A”，若用户想将“功能 A”升级或者重新烧录为“功能 B”，且在升级或烧录“功能 B”过程中芯片的“功能 A”不会停止工作，升级完成后“功能 B”可以马上加载到芯片上，这就需要背景升级技术。

Gowin goConfig I2C IP 可实现 I²C 端口的在线升级功能，用户通过 I²C 端口对 FPGA 进行烧录，实现对内部存储的数据替换，实现背景升级。

表 2-1 Gowin goConfig I2C IP 概述

Gowin goConfig I2C IP	
IP核应用	
逻辑资源	请参见表2-4，表2-5。
交付文件	
设计文件	Verilog (encrypted)
参考设计	Verilog
TestBench	Verilog
测试设计流程	
综合软件	GowinSynthesis [®]
应用软件	Gowin Software (V1.9.8.07及以上)

注！

可登录[高云半导体网站](#)查看芯片支持信息。

2.1 升级功能概述

实现升级功能包括两个过程：编程（Program），配置（Configure）。

- 编程（Program Flash）：将高云半导体云源®软件生成的比特流数据，写入到 Flash 的过程（下文简称“写入 Flash”）。
- 配置（Configure SRAM）：将 Flash 中的比特流数据，加载到 FPGA 的 SRAM 存储区的过程（下文简称“载入 SRAM”）。

注！

- 普通升级：进行编程（Program）时，需要先擦除 SRAM（即停止了正在运行的功能），再“写入 Flash”。
- 背景升级：进行编程（Program）时，不会擦除 SRAM（即不影响正在运行的功能），直接“写入 Flash”。
- 配置（Configure）时，“普通升级”与“背景升级”操作过程一致。

背景升级功能示例如下：

用户一般通过正在运行的“功能 A”，将待更新的“功能 B”写入 Flash；然后用户可在需要更新功能时，通过一系列操作，将 Flash 内的“功能 B”载入 SRAM 内，完成功能的更新。

注：

- “功能 A”按照背景升级功能操作时，必须带有背景升级功能的模块。本文中，即“功能 A”必须带 Gowin goConfig I2C IP。
- “功能 B”可以不带背景升级功能的模块，但是后续不能通过“功能 B”进行背景升级。
- “写入 Flash”过程，“功能 A”仍然正常运行；“载入 SRAM”过程，“功能 A”会停止运行。

2.2 I²C 接口

Gowin goConfig I2C IP 的 I²C 接口通讯方式，符合 I²C 官方手册 7-bit 地址协议，IP 详细的使用请参考第 3 章 功能描述以及第 5 章 IP 使用说明描述。

2.2.1 I²C 基本特性

I²C 的 7-bit 地址协议的典型时序示例如图 2-1，相关说明如下：

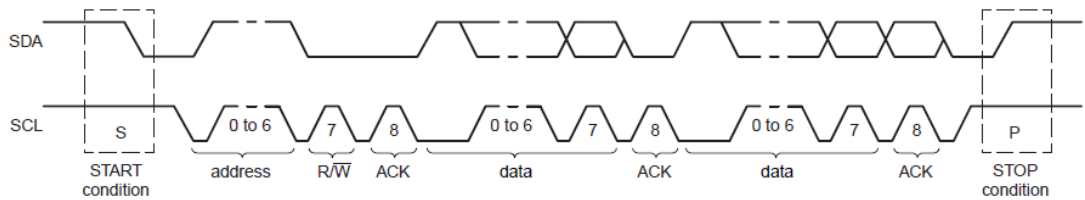
- S: START condition，通讯开始标志。
- address: 从机地址，位宽 7bit。
- R/W: 读写控制 bit。0->写,1->读。
- ACK: 从机响应位。0->响应(ACK),1->不响应(NACK)。

- data: 读/写的有效数据，位宽 8 比特。
- S: STOP condition，通讯结束标志。

注！

多位宽时，高比特先发送。例如：address=7'h58, 按比特发送顺序为 101_1000。
data=8'hB1,按比特发送顺序为 1011_0001。

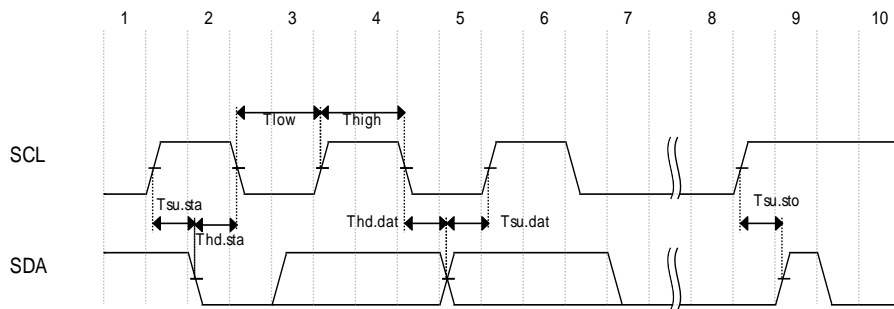
图 2-1 I2C 7-bit 典型时序示例



2.2.2 I2C 时序特性

Gowin goConfig I2C IP 在 2C 系列与 9X 系列的时序特性存在差别。说明如图 2-2 所示。

图 2-2 I2C 接口时序



2C 系列接口时序

Gowin goConfig I2C IP 在 2C 系列的接口时序，可以兼容 I2C 官方手册的定义，详细说明如表 2-2 所示。已验证的 I2C 的最高速率可达 400KHz。

表 2-2 2C 系列 I2C 接口时序说明

名称	描述	Min	Max	Unit	备注
F _{SCL}	Clock Frequency	-	400	KHz	-
T _{low}	LOW period of the SCL	1.3	-	μs	
T _{high}	HIGH period of the SCL	0.6	-	μs	
T _{hd.sta}	Start Hold Time	0.6	-	μs	
T _{su.sta}	Start Setup Time	0.6	-	μs	

名称	描述	Min	Max	Unit	备注
T _{hd.dat}	Data In Hold Time	0	-	ns	
T _{su.dat}	Data In Setup Time	100	-	ns	
T _{su.sto}	Stop Setup Time	0.6	-	μs	

9X 系列接口时序

Gowin goConfig I2C IP 在 9X 系列上的接口时序，部分参数不符合 I²C 官方手册的定义，详细的说明如表 2-3 所示。已验证的 I²C 的最高速率可达 400KHz。

表 2-3 9X 系列 I2C 接口时序说明

名称	描述	Min	Max	Unit	备注
F _{SCL}	Clock Frequency	-	400	KHz	-
T _{low}	LOW period of the SCL	1.3	-	μs	
T _{high}	HIGH period of the SCL	0.6	-	μs	
T _{hd.sta}	Start Hold Time	0.6	-	μs	
T _{su.sta}	Start Setup Time	0.6	-	μs	
T _{hd.dat}	Data In Hold Time	50 ^[1]	-	ns	
T _{su.dat}	Data In Setup Time	100	-	ns	
T _{su.sto}	Stop Setup Time	0.6	-	μs	

注！

[1] T_{hd.dat} 与 I²C 官方手册不一致。

2.3 IP 特性

Gowin goConfig I2C (2C 系列) IP 特性包括：

- I²C 接口为 Slave 模式
- 支持 I²C 最高传输速度达 400Kbps
- I²C 接口支持 7-bit 地址模式
- I²C 接口从机地址自由配置
- 支持内部存储的读写
- 支持输出 25 MHz 时钟供用户设计使用
- 仅支持基础的 I²C 读写协议
- 时钟和复位内部产生，无需外部输入

- 完全可综合
 - Gowin goConfig I2C (9X 系列) IP 特性包括:
- I²C 接口为 Slave 模式
- 支持 I²C 最高传输速度达 400Kbps
- I²C 接口支持 7-bit 地址模式
- I²C 接口从机地址自由配置
- 支持内部存储的读写
- 支持输出 20MHz 时钟
- 时钟和复位内部产生, 无需外部输入
- 完全可综合

2.4 资源占用

Gowin goConfig I2C IP (2C 系列), 以高云 GW1N-2C 器件为例, 其资源利用情况如表 2-4 所示。

表 2-4 2C 系列资源消耗参考

器件	编程语言	LUT	REG	BSRAM
GW1N-2C	Verilog	608	241	1

Gowin goConfig I2C IP (9X 系列) 以 GW2AN-9X 器件为例, 其资源利用情况如表 2-5 所示。

表 2-5 9X 系列资源消耗参考

器件	编程语言	LUT	REG	BSRAM
GW2AN-9X	Verilog	444	247	2

3 功能描述

3.1 goConfig I2C 结构框图

Gowin goConfig I2C IP 的应用可分为用户（User）侧和 FPGA 侧。User 侧用于控制 I2C 接口，以及其他用户功能。FPGA 侧连接 Flash，以及用户需要在 FPGA 上实现的功能。若仅考虑 IP，User 侧通过 I2C 接口控制 FPGA 内的 IP，进而控制 Flash，完成背景升级的功能。

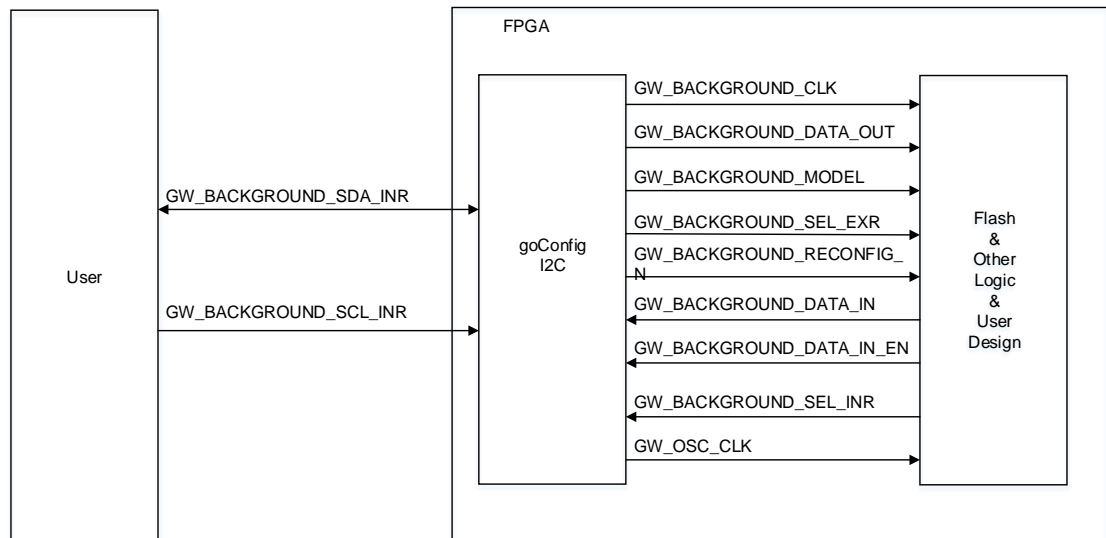
注！

文档中主要描述 User 侧接口操作，本文不涉及 Flash 接口的详细操作与时序。

2C 系列实现框图如图 3-1 所示，I²C 接口

（GW_BACKGROUND_SCL_INR, GW_BACKGROUND_SDA_INR）需要 User 侧进行控制，其他信号根据实际的功能需求，选择控制或使用。

图 3-1 结构框图（2C 系列）

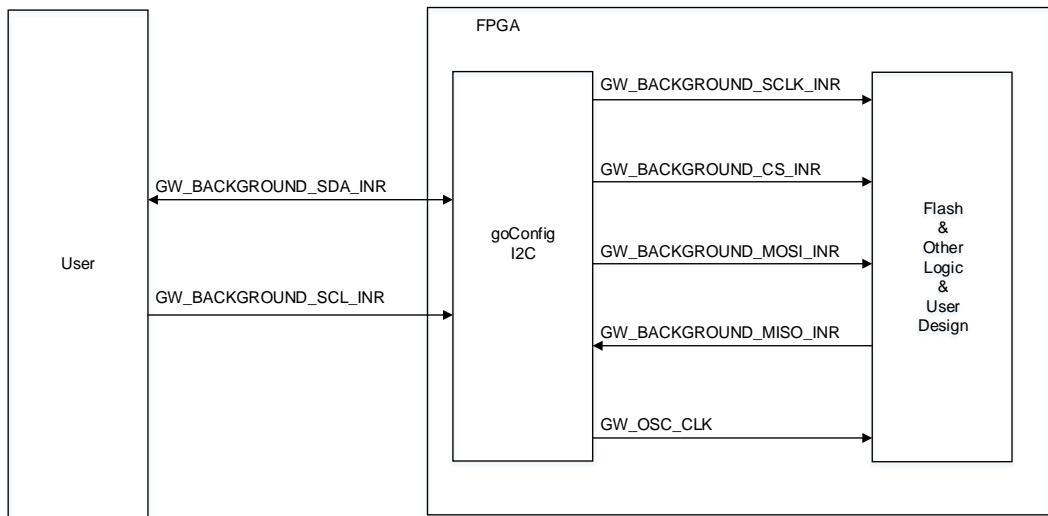


9X 系列实现框图如图 3-2 所示，I²C 接口

（GW_BACKGROUND_SCL_INR, GW_BACKGROUND_SDA_INR）需

要 User 侧进行控制，其他信号根据实际的功能需求，选择控制或使用。

图 3-2 结构框图（9X 系列）



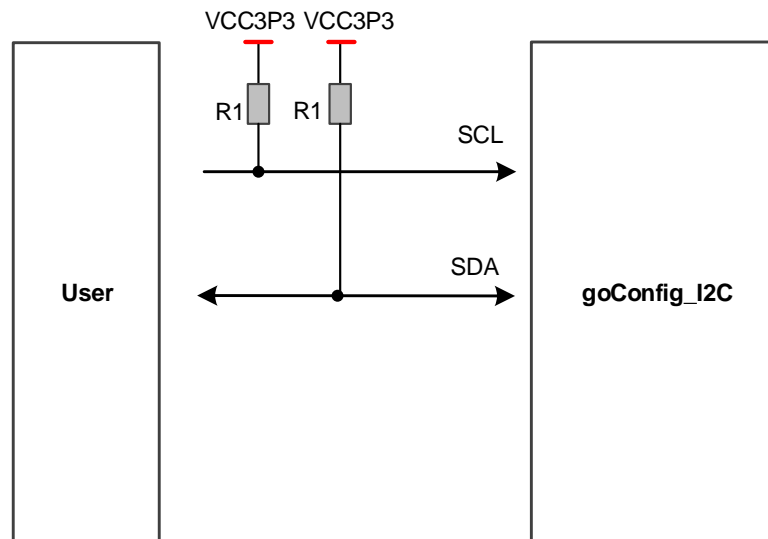
3.2 Gowin goConfig I2C IP 外部电路连接

注！

本小节描述内容同时适用于 2C 系列、9X 系列。

Gowin goConfig I2C IP 外部电阻电路连接如图 3-3 所示。

图 3-3 外部电阻电路连接关系



R1:4.7K

注！

- IO 端口属性约束参考如下：
 - GW_BACKGROUND_SCL_INR: IO_TYPE=LVCMOS33 PULL_MODE= UP

- GW_BACKGROUND_SDA_INR: IO_TYPE=LVC MOS33 PULL_MODE= UP DRIVE=8
- I/O 所在 Bank 的供电电压为 3.3V。

3.3 Gowin goConfig I2C IP 背景升级控制指令

Gowin goConfig I2C IP 在 2C 系列与 9X 系列中，操作方式不通用，下文将按 2C 系列与 9X 系列分别描述。

3.3.1 2C 系列

指令定义

Gowin goConfig I2C IP 2C 系列支持的指令如表 3-1 所示。

表 3-1 2C 系列指令定义

序号	指令名称	指令地址 (CMD)	描述	指令格式	备注
1	Status	0x44	读状态寄存器指令	CMD	-
2	Reset	0x55	复位IP指令	CMD	
3	Send (data back)	0x66	发送数据且回读Flash接口数据指令	CMD + Length + Data +Check_sum	
4	Waiting	0x77	等待指令	CMD + Length + Check_sum	
5	Send (No data back)	0x88	发送数据且不回读Flash接口数据指令	CMD + Length + Data +Check_sum	
6	Reconfign	0x99	Reconfign指令	CMD	
7	IP Version	0xAA	IP版本指令	CMD	

指令格式中，名称的定义与说明如下：

- **CMD**: 支持的指令参考表 3-1;
 - 定义：指令地址；
 - 位宽：8 比特 (bit)。
- **Length**: 最小值为 2
 - 定义：发送的有效字节长度^[1]；
 - 位宽：16 比特 (bit)，低 8 比特 (Length_L) 先发，高 8 比特 (Length_H) 后发；

- 计算公式： $Length = \text{有效数据长度} + 2$ 。

注!

- [1] 有效字节长度即有效数据 (Data) 的字节数+2; 以 0x66 指令为例, $Length = \text{Data 的字节长度} + 2$;
- 例如预期输入字节长度为 10, 则 $Length = 10 + 2 = 12$ 。
- Data:
 - 定义: 发送的有效数据;
 - 位宽: 一个数据 8 比特;
 - 长度: 实际发送的数据长度根据 $Length - 2$ 决定; Data_0 表示第一个有效数据, Data_1 表示第二个有效数据, ... Data_N 表示最后一个有效数据。
- Check_sum:
 - 定义: 校验
 - 位宽: 16 比特, 高 8 比特 (Check_sum_H) 先发, 低 8 比特 (Check_sum_L) 后发;
 - 计算公式: $Check_sum = \text{CMD} + Length_L + Length_H + \text{Data}_0 + \dots + \text{Data}_N$;

注!

当 Check_sum 计算溢出时, 只取低 16 位, 其余位舍弃。假设计算结果, $Check_sum = 0x10F11$; 则 $Check_sum_H = 0x0F$; $Check_sum_L = 0x11$. 其余的舍弃。

Status(0x44)

读状态寄存器指令, 只读。读回的 Data 为状态寄存器值, 定义如下表 3-2 所示。

表 3-2 状态寄存器

比特位	默认值	描述	备注
Bit[7:3]	0	保留	
Bit [2]	0	1: Flash 接口工作中 0: Flash 接口空闲	-
Bit [1]	0	1: 校验出错 0: 校验正确	
Bit [0]	0	1: I ² C 状态出错 0: I ² C 状态正常	-

读状态寄存器指令示例如图 3-4 所示。

图 3-4 读状态寄存器指令示例

S	Address	W	A	0x44	Sr	Address	R	A	Data	N	P
---	---------	---	---	------	----	---------	---	---	------	---	---

注！

下文中，相似图例，未作特殊说明，则含义与本例中一致。

- 图中“白色底纹”为“I2C 主机”发送至“I2C 从机”，“灰色底纹”为“I2C 从机”发送至“I2C 主机”。
- 图示中会使用到的名称定义如下：
 - S: I²C 总线的“开始 (START)”；
 - Address: I²C 总线的“从机地址 (Slave Address)”；
 - W: I²C 总线的“写 (WRITE)”；
 - A: I²C 总线的“响应 (ACK)”；
 - Sr: I²C 总线的“再开始 (Repeated START)”；
 - R: I²C 总线的“读 (READ)”；
 - Data: I²C 总线中，传输的有效数据；
 - N: I²C 总线的“不响应 (NACK)”；
 - P: I²C 总线的“停止 (STOP)”；
 - Length_L: Length 的低 8 bits；
 - Length_H: Length 的高 8 bits；
 - Data_0: 第一个有效数据；
 - Data_N: 最后一个有效数据；
 - Check_sum_H: Check_sum 的高 8 bits；
 - Check_sum_L: Check_sum 的低 8 bits。

Reset(0x55)

复位 IP 指令。将复位 IP 内部的状态，以及清空内部的 FIFO。

注！

此过程只涉及 IP，不涉及背景升级流程

图 3-5 复位 IP 指令示例

S	Address	W	A	0x55	A	P
---	---------	---	---	------	---	---

Send(0x66,data back)

发送数据且回读 Flash 数据指令。I2C 接口发送将要读回数据的字节（byte）数目（Length），同时发送数据；若校验无误 IP 将转换为 Flash 接口，并将数据回读到 FIFO 中（图 3-6 步骤）。此时，若用户侧若需要此数据，可通过 I2C 接口发送读指令，可以将 FIFO 内的数据读出（图 3-7 步骤）。

简单来说，在回读的过程中，分为两步：

1. 将待读数据读至 IP 的 FIFO 内；
2. 将 FIFO 内的数据通过 I²C 总线读出。

注！

- 第 1 步与第 2 步中间可以插入部分指令，例如读状态指令。
- 在图 3-6 中，Data_0~Data_N，可以为任意值，但一般建议写 0x00 或 0xFF。
- 在图 3-7 中，Data_0~Data_N 是读回的实际值。

图 3-6 回读指令发送示例

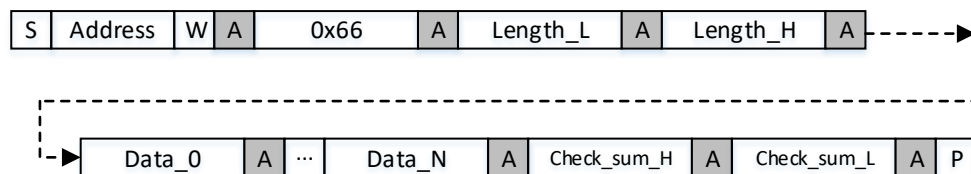
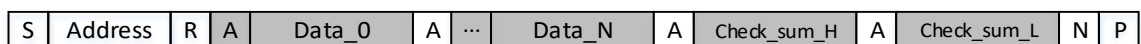


图 3-7 读 FIFO 数据指令示例



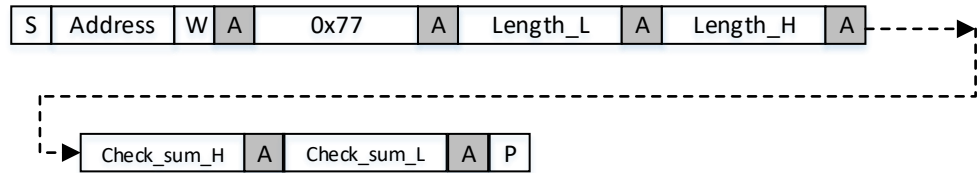
Waiting(0x77)

等待指令。I2C 接口发送等待时长的字节数目（Length），校验无误，Flash 接口将会持续发送空闲的时钟（将不会有数据传输），直到“等待时长的字节数目”计数完成。

注！

2C 系列中，Flash 接口时钟为 2.5MHz，即一个周期 400ns。1 字节为 $8 \times 400 \text{ ns} = 3200 \text{ ns} = 3.2 \mu\text{s}$ 。例如，等待 150ms，计算出有效的字节数为 $46875 = 0xB71B$ ；输入的字节数为 $0xB71B + 2 = 0xB71D$ ，则 Length_L = 0x1D；Length_H = 0xB7。

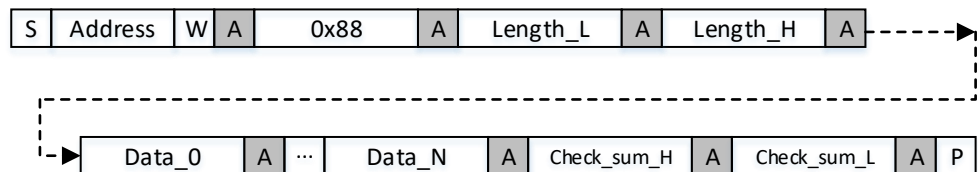
图 3-8 等待指令示例



Send(0x88, No data back)

发送数据且不回读 Flash 数据指令。I2C 接口先将待发送数据存放到 FIFO，在数据计数完成后，且校验无误，FIFO 内的数据一次性通过 Flash 接口发出。

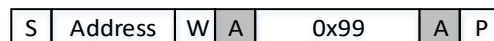
图 3-9 发送数据且不回读 Flash 数据指令示例



Reconfig(0x99)

Reconfig 指令。I2C 接口发送此指令，可实现拉低“Reconfig_N”管脚的功能。

图 3-10 Reconfig 指令示例



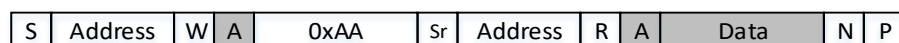
IP Version(0xAA)

IP 版本指令。I2C 接口发送此指令，读回的 Data 为固定值。

注！

当前版本，Version_Data=0x20。

图 3-11 IP 版本指令



3.3.2 9X 系列

9X 系列与 2C 系列器件存在明显差异。使用时请特别注意区别。9X 系列主要以寄存器的方式进行通讯。

寄存器定义

相关的寄存器定义如表 3-3 所示。

表 3-3 寄存器定义

序号	寄存器名称	寄存器地址 (ADDR)	访问类型	描述	格式	备注
1	CSCR	0x00	写	\overline{CS} 信号控制寄存器	ADDR+Data	-
2	DICN	0x01	写/读	写操作字节数寄存器	ADDR+Data	
3	DOCN	0x02	写/读	读操作字节数寄存器	ADDR+Data	
4	DIFR	0x03	写	写FIFO寄存器	ADDR+Data	
5	SRSTR	0x08	写	软复位寄存器	ADDR+Data	
6	SR	0x80	读	状态寄存器	ADDR+Data	
7	DOFR	0x81	读	读FIFO寄存器	ADDR+Data	

其格式为：寄存器地址（ADDR）+数据（Data）。其中根据寄存器地址不同，数据 Data 的长度（单位为字节，Byte）存在差别。

下文中将以“寄存器名称-Data”区分各个寄存器的数据。例如 CSCR-Data 表示 \overline{CS} 信号控制寄存器的 Data。

CSCR(0x00)

\overline{CS} 信号控制寄存器，格式为：ADDR+ CSCR-Data（1 字节）。其中 CSCR-Data 的长度为 1（字节，Byte），定义如下。

表 3-4 CSCR-Data 定义

比特位	默认值	访问类型	描述	备注
Bit[7:0]	0x00	写	0xFF: CS 信号置低（Flash 接口将进入操作状态） 0x00: CS 信号置高（Flash 接口将进入空闲状态） 其他：保留	-

\overline{CS} 信号控制寄存器使用示例，如图 3-12 所示，以置低 cs 为例，其中 CSCR-Data=0xFF。

图 3-12 CSCR 寄存器使用示例

S	Address	W	A	0x00	A	Data	A	P
---	---------	---	---	------	---	------	---	---

DICN(0x01)

写操作字节数寄存器，格式为：ADDR+ DICN-Data（2 字节）。其中 DICN-Data 的长度为 2（字节，Byte）；值的有效范围为 1~300，0x0001 表示 1 个字节，0x00FF 表示 255 个字节，其他值类推。

表 3-5 DICN-Data 定义

比特位	默认值	访问类型	描述	备注
Bit[15:0]	0x0000	写/读	1~300: 写操作字节数 其他: 保留	-

DICN-Data 的两字节发送时，先发送低字节（Bit[7:0]），再发送高字节（Bit[15:8]）。因此，当 DICN-Data=0x0001 时，先发送 0x01，再发送 0x00。

DICN 寄存器写使用示例，如图 3-13 所示。若 DICN-Data=0x0001；则 Data_0 = 0x01，Data_1 = 0x00。

图 3-13 DICN 寄存器写使用示例

S	Address	W	A	0x01	A	Data_0	A	Data_1	A	P
---	---------	---	---	------	---	--------	---	--------	---	---

DICN 寄存器读使用示例，如图 3-14 所示。若 DICN-Data=0x0001；则读回的 Data_0 = 0x01，Data_1 = 0x00。

图 3-14 DICN 寄存器读使用示例

S	Address	W	A	0x01	Sr	Address	R	A	Data_0	A	Data_1	N	P
---	---------	---	---	------	----	---------	---	---	--------	---	--------	---	---

DOCN(0x02)

读操作字节数寄存器，格式为：ADDR+ DOCN-Data（2 字节）。其中 DOCN-Data 的长度为 2（字节，Byte）；值的有效范围为 1~300，0x0001 表示 1 个字节，0x00FF 表示 255 个字节，其他值类推。

表 3-6 DOCN-Data 定义

比特位	默认值	访问类型	描述	备注
Bit[15:0]	0x0000	写/读	1~300: 写操作字节数 其他: 保留	-

DOCN-Data 的两字节发送时, 先发送低字节 (Bit[7:0]), 再发送高字节 (Bit[15:8])。因此, 当 DOCN-Data=0x0001 时, 先发送 0x01, 再发送 0x00。

DOCN 寄存器写使用示例, 如图 3-15 所示。若 DOCN-Data=0x0001; 则 Data_0 = 0x01, Data_1 = 0x00。

图 3-15 DOCN 寄存器写使用示例

S	Address	W	A	0x02	A	Data_0	A	Data_1	A	P
---	---------	---	---	------	---	--------	---	--------	---	---

DOCN 寄存器读使用示例, 如图 3-16 所示。若 DOCN-Data=0x0001; 则 Data_0 = 0x01, Data_1 = 0x00。

图 3-16 DOCN 寄存器读使用示例

S	Address	W	A	0x02	Sr	Address	R	A	Data_0	A	Data_1	N	P
---	---------	---	---	------	----	---------	---	---	--------	---	--------	---	---

DIFR(0x03)

写 FIFO 寄存器, 格式为 ADDR+ DIFR-Data (1~300 字节)。其中 DIFR-Data 的长度范围为 1~300 (字节, Byte)。一次发送 1 字节, 实际发送的数据的字节数需要与写操作字节数寄存器的值相匹配。

表 3-7 DIFR-Data 定义

比特位	默认值	访问类型	描述	备注
Bit[7:0]	-	写	0x00~0xFF	-

例如, DICN-Data=0x0011; 则 DIFR 将会发送 0x0011=17 字节的数据。

DIFR 寄存器使用示例如下, Data_0 表示写入的第一个字节数据, Data_N 表示写入的最后一个数据。其中, 写入的字节数应该与 DICN-Data 值一致。

图 3-17 DIFR 寄存器使用示例

S	Address	W	A	0x03	A	Data_0	A	...	Data_N	A	P
---	---------	---	---	------	---	--------	---	-----	--------	---	---

SRSTR(0x08)

软复位寄存器，格式为 ADDR+ SRSTR-Data（1 字节）。其中 SRSTR-Data 的长度范围为 1（字节，Byte）；当 SRSTR-Data =0xFF 时，复位才有效。

表 3-8 SRSTR-Data 定义

比特位	默认值	访问类型	描述	备注
Bit[7:0]	0x00	写	0xFF：软复位有效 其他：保留	-

软复位寄存器使用示例，如图 3-18 所示，其中 SRSTR-Data=0xFF。

图 3-18 SRSTR 寄存器使用示例

S	Address	W	A	0x08	A	Data	A	P
---	---------	---	---	------	---	------	---	---

SR(0x80)

状态寄存器，格式为 ADDR+ SR-Data（3 字节）。其中 SR-Data 的长度范围为 3（字节，Byte）。

表 3-9 SR-Data 定义

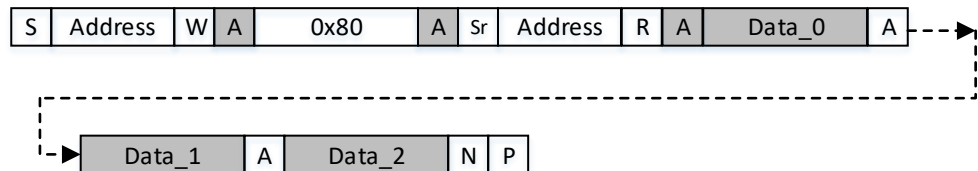
比特位	名称	访问类型	描述	备注
Bit[23:15]	TXLVL	读	TX FIFO有效数据数量。取值范围 1~300。	-
Bit [14:6]	RXLVL	读	RX FIFO有效数据数量。取值范围 1~300。	
Bit [5:4]	FIFO状态	读	Bit [5]: 1: TXLVL中有数据 0: TXLVL中无数据 Bit [4]: 1: RXLVL中有数据 0: RXLVL中无数据	-
Bit [3:0]	Flash接口 操作状态	读	Bit [3:2]: 保留 Bit [1:0]: 11: 保留 10: Flash接口操作完成 01: Flash接口操作中	-

比特位	名称	访问类型	描述	备注
Bit[23:15]	TXLVL	读	TX FIFO有效数据数量。取值范围1~300。	-
Bit [14:6]	RXLVL	读	RX FIFO有效数据数量。取值范围1~300。	
Bit [5:4]	FIFO状态	读	Bit [5]: 1: TXLVL中有数据 0: TXLVL中无数据 Bit [4]: 1: RXLVL中有数据 0: RXLVL中无数据	
			00: Flash接口空闲	

SR-Data 的 3 字节发送时，先发送低字节（Bit[7:0]），接着发送次字节（Bit[15:8]），再发送高字节（Bit[23:15]）。因此，当 SR-Data = 0x000102 时，先发送 0x02，接着发送 0x01，再发送 0x00。

状态寄存器使用示例，如图 3-19 所示。若 SR-Data=0x000102，则 Data_0 = 0x02，Data_1 = 0x01，Data_2= 0x00。

图 3-19 SR 寄存器使用示例



DOFR(0x81)

读 FIFO 寄存器，格式为 ADDR+ DOFR-Data（1~300 字节）。其中 DOFR-Data 的长度范围为 1~300（字节，Byte）。一次读取 1 字节，实际读取的数据的字节数需要与读操作字节数寄存器的值相匹配。

表 3-10 DOFR-Data 定义

比特位	默认值	访问类型	描述	备注
Bit[7:0]	-	读	0x00~0xFF	-

DOFR 寄存器使用示例如下，Data_0 表示读取的第一个字节数据，Data_N 表示读取的最后一个数据。其中，读取的字节数应该与 DOCN-Data

值一致。

图 3-20 DOFR 寄存器使用示例

S	Address	W	A	0x81	A	Data_0	A	...	Data_N	N	P
---	---------	---	---	------	---	--------	---	-----	--------	---	---

4 信号定义

4.1 信号定义

Gowin goConfig I2C IP 在 2C 系列与 9X 系列中，IO 接口定义存在部分差别，下文将按 2C 系列与 9X 系列分别描述，其中 2C 系列 IO 见表 4-1。

表 4-1 2C 系列 IO 接口定义

序号	信号名称	方向	描述	IO分配约束	备注
1	GW_BACKGROUND_SCL_INR	Input	I ² C总线时钟	限名	所有信号输入输出方向均以IP为参考。
2	GW_BACKGROUND_SDA_INR	Inout	I ² C总线数据	限名	
3	GW_BACKGROUND_SEL_INR	Input	背景升级使能输入 1: 使能（建议默认） 0: 不使能	-	
4	GW_OSC_CLK	Output	25 MHz输出时钟	-	
5	GW_BACKGROUND_CLK	Output	IP-Flash接口时钟	自动	
6	GW_BACKGROUND_MODEL	Output	IP-Flash接口模式选择	自动	
7	GW_BACKGROUND_DATA_OUT	Output	IP-Flash接口数据输出	自动	
8	GW_BACKGROUND_DATA_IN	Input	IP-Flash接口数据输入	自动	
9	GW_BACKGROUND	Input	IP-Flash接口输	自动	

序号	信号名称	方向	描述	IO分配约束	备注
	D_DATA_IN_EN		入数据有效信号		
10	GW_BACKGROUND_SEL_EXR	Output	背景升级使能输出	自动	
11	GW_BACKGROUND_RECONFIG_N	Output	RECONFIG_N 信号	自动	

注!

1. IO 分配约束各定义如下:

- “-”: 无限制。可以自由命名, 可以自由约束位置。
- “限名”: 名称限制。不能自由命名, 可以自由约束位置。即顶层 IO 名字需要与 IP 定义的完全一致, 包括“信号名称”与“方向”。
- “自动”: 自动约束位置。不能自由命名, 不能约束位置。即顶层 IO 名字需命名为特定 IO 名称; 且不能约束位置, 高云云源软件会自动约束位置。

2. “GW_BACKGROUND_SEL_INR”: 在没有使能变化时, 需要保持输入为“1”(高电平)。

3. “2C 系列”系列中, “特定 IO 名字”即 IP 的 IO 接口名称。即顶层 IO 名字, 需要与 IP 定义的完全一致。

4. “IO 分配约束”各定义也适用于“9X 系列”的 IO 分配约束。

其中 9X 系列 IO 见表 4-2。

表 4-2 9X 系列 IO 接口定义

序号	信号名称	方向	描述	IO分配 ^[1]	备注
1	GW_BACKGROUND_SCL_INR	Input	I ² C总线时钟	-	所有信号输入输出方向均以 IP 为参考。
2	GW_BACKGROUND_SDA_INR	Inout	I ² C总线数据	-	
3	GW_OSC_CLK	Output	20MHz输出时钟	-	
4	GW_BACKGROUND_SCLK_INR	Output	IP-Flash接口时钟	自动 ^[2]	
5	GW_BACKGROUND_CS_INR	Output	IP-Flash接口 CS输出	自动 ^[2]	
6	GW_BACKGROUND_MOSI_INR	Output	IP-Flash接口数据输出	自动 ^[2]	
7	GW_BACKGROUND_MISO_INR	Input	IP-Flash接口数据输入	自动 ^[2]	

注!

[1] IO 分配约束各定义请参考“2C 系列”相关描述。

[2] 在顶层使用时，部分 IO 接口名称需改成 9X 系列“特定 IO 名字”（Flash 接口专属名称），Flash 接口专属名称定义以及对应关系如下：

IP IO 名称 > Flash 接口专属名称（IO 方向）

GW_BACKGROUND_SCLK_INR > **O_flash_ck(Output)**

GW_BACKGROUND_CS_INR > **O_flash_cs_n(Output)**

GW_BACKGROUND_MOSI_INR > **IO_flash_do(InOut)**

GW_BACKGROUND_MISO_INR > **IO_flash_di(Output)**

5 IP 使用说明

注!

本章节内容以 IP 为描述对象，说明仅限于操作 IP。

Gowin goConfig I2C IP 在使用时，2C 系列与 9X 系列存在差别，本章将以系列做区分，分别描述云源软件配置、推荐的 IP 流程、注意事项。

5.1 2C 系列

5.1.1 软件配置

在综合（Synthesizes）带背景升级 IP 的工程前，需要先更改云源软件配置，Gowin goConfig I2C IP 才能正常工作。

在云源软件中，器件选择 2C 系列后，点击“Project > Configuration > BitStream > Background Programming > I2C”如图 5-1 所示。

图 5-1 打开 Configuration 选项

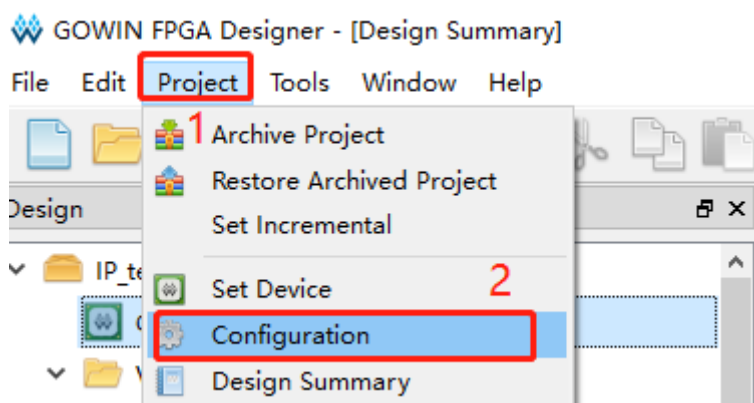
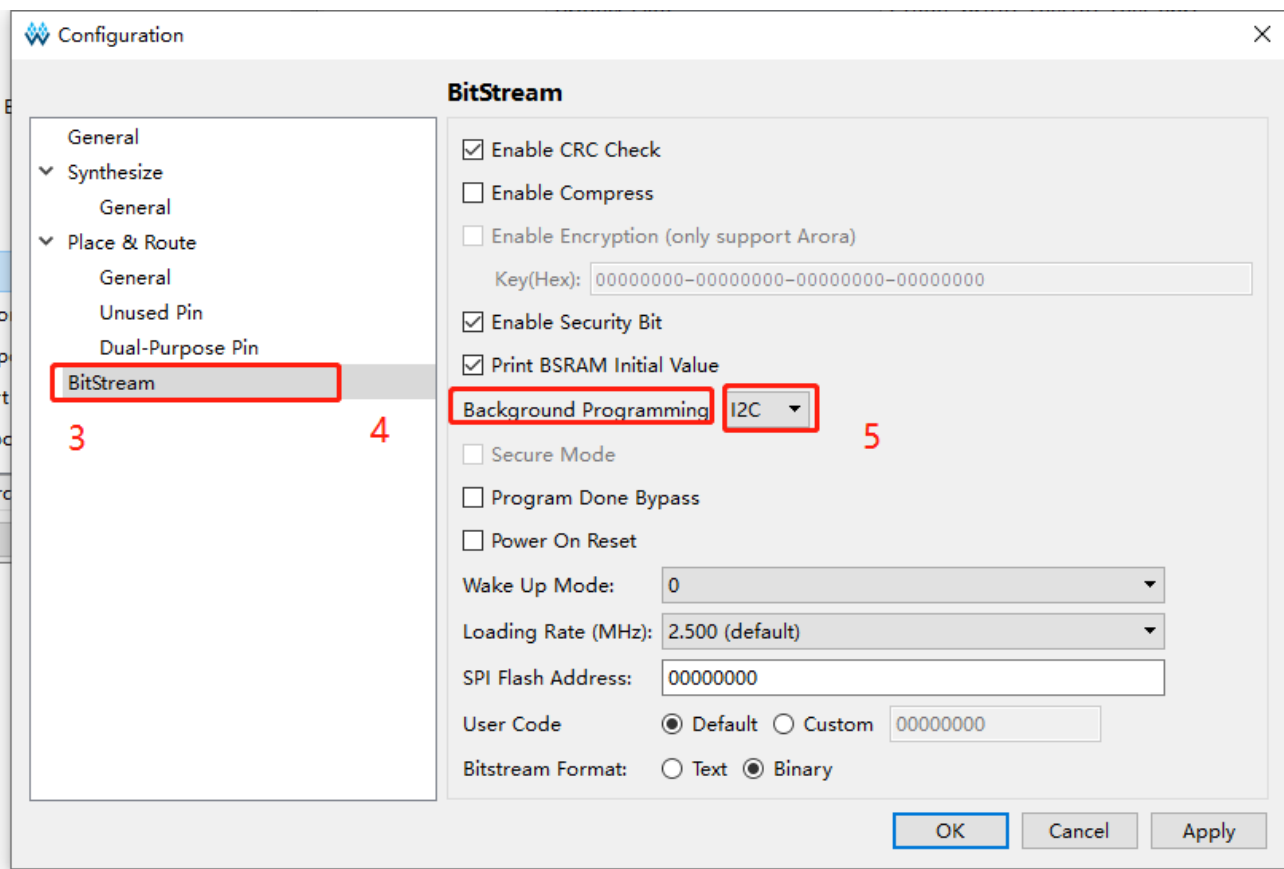


图 5-2 配置背景升级为 I2C



5.1.2 流程说明

Gowin goConfig I2C IP 根据 2C 系列操作，可以将指令分为：读写指令；操作指令。

- 读写指令：需要发送 CMD 指令，还需要发送长度（Length）来控制读/写的指令。例如“Send(data back)”，“Waiting”，“Send(No data back)”。
- 操作指令：只需要发送 CMD 指令，不需要发送长度（Length）来控制读/写的指令。例如“Reset”、“Reconfign”指令。

2C 系列操作可分为：

1. 前期准备；
2. 写流程；
3. 读流程；
4. 触发 RECONFIG_N。

注！

IP 在“读写指令”完成后，会直接操作 Flash 接口。“操作指令”主要控制/查看 IP 的状

态。

前期准备

IP 的前期准备，用于检验/确保 IP 工作正常，可按照下面的描述进行：

1. 发送“Reset”指令（0x55），确保 IP 位于初始状态；
2. 发送“IP Version”指令（0xAA），查看读回的版本值（Version_Data=0x20），确保 IP 版本符合预期；
3. 发送“Status”指令（0x44），查看读回的状态值，确保为 0x00。

操作完成后，读回的数据都符合预期，则说明可以开始与 IP 通讯。

写指令流程

当确定“前期准备”正常后，可以开始写指令流程。写指令流程一般按照下面的步骤：

1. 发送“Send (No data back)”（0x88）指令，写数据；
2. 发送“Status”（0x44）指令，确保 IP 操作完成，可以进行其他操作。

操作完成后，读回的状态数据都符合预期，则说明可以开始与 IP 进行其他通讯。

回读指令流程

当确定前期准备正常后，可以开始回读指令流程。回读指令流程一般按照下面的步骤：

1. 发送“Send(data back)”（0x66）指令，写待读回数据的长度；
2. 发送“Status”（0x44）指令，确保 Flash 侧操作完成；
3. 通过 I2C 接口，读数据（参考 [Send\(0x66,data back\)](#) 步骤）；
4. 发送“Status”（0x44）指令，确保 IP 操作完成，可以进行其他操作。

操作完成后，读回的状态数据都符合预期，则说明可以开始与 IP 进行其他通讯。

触发 RECONFIG_N

触发“RECONFIG_N”。其功能为：在背景升级的第一步（写入 Flash）完成后，可以通过触发“RECONFIG_N”，执行第二步（将 Flash 的内容载入 SRAM），完成背景升级。

5.1.3 注意事项

Gowin goConfig I2C IP 在使用时的注意事项如下。

IO 约束

如“信号定义”中说明，Gowin goConfig I2C IP 的 2C 系列 IO 信号可以分为两类：无限制 IO 以及有限制 IO（包括限名与自动）。

- 无限制 IO：与常规使用相同，可以自由选择名字，是否放置到顶层等。
- 有限制 IO：限定 IO 名称，且必须通过顶层输入或输出。

注！

云源软件会对此“有限制 IO”进行检测，若选择了背景升级-I2C，但 IO 不存在或不对应，软件执行“Place & Route”时会报错。

OSC 输出

Gowin goConfig I2C IP 的使用了器件内的 OSC 模块。同时，IP 可以提供 25MHz 的时钟输出，可根据需求选择是否使用。

缓存限制

Gowin goConfig I2C IP 使用了 1 个 BSRAM 进行数据缓存，因此一次传输的数据总量不能超过 1 个 BSRAM。

5.2 9X 系列

5.2.1 软件配置

在综合（Synthesizes）带背景升级 IP 的工程前，需要先更改云源软件配置，Gowin goConfig I2C IP 才能正常工作。

在云源软件中选择 9X 系列后，点击“Project > Configuration > BitStream > Background Programming > Internal”。

图 5-3 打开 Configuration 选项

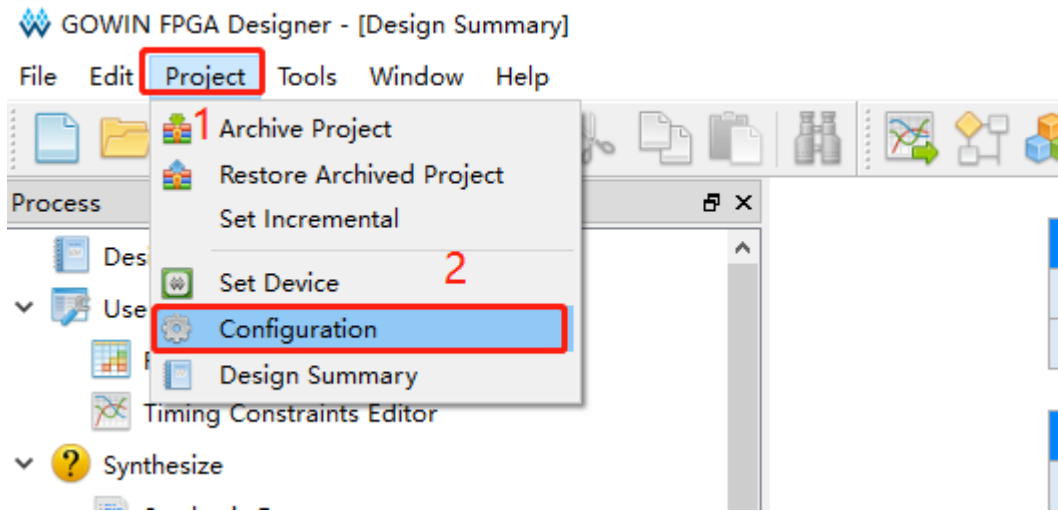
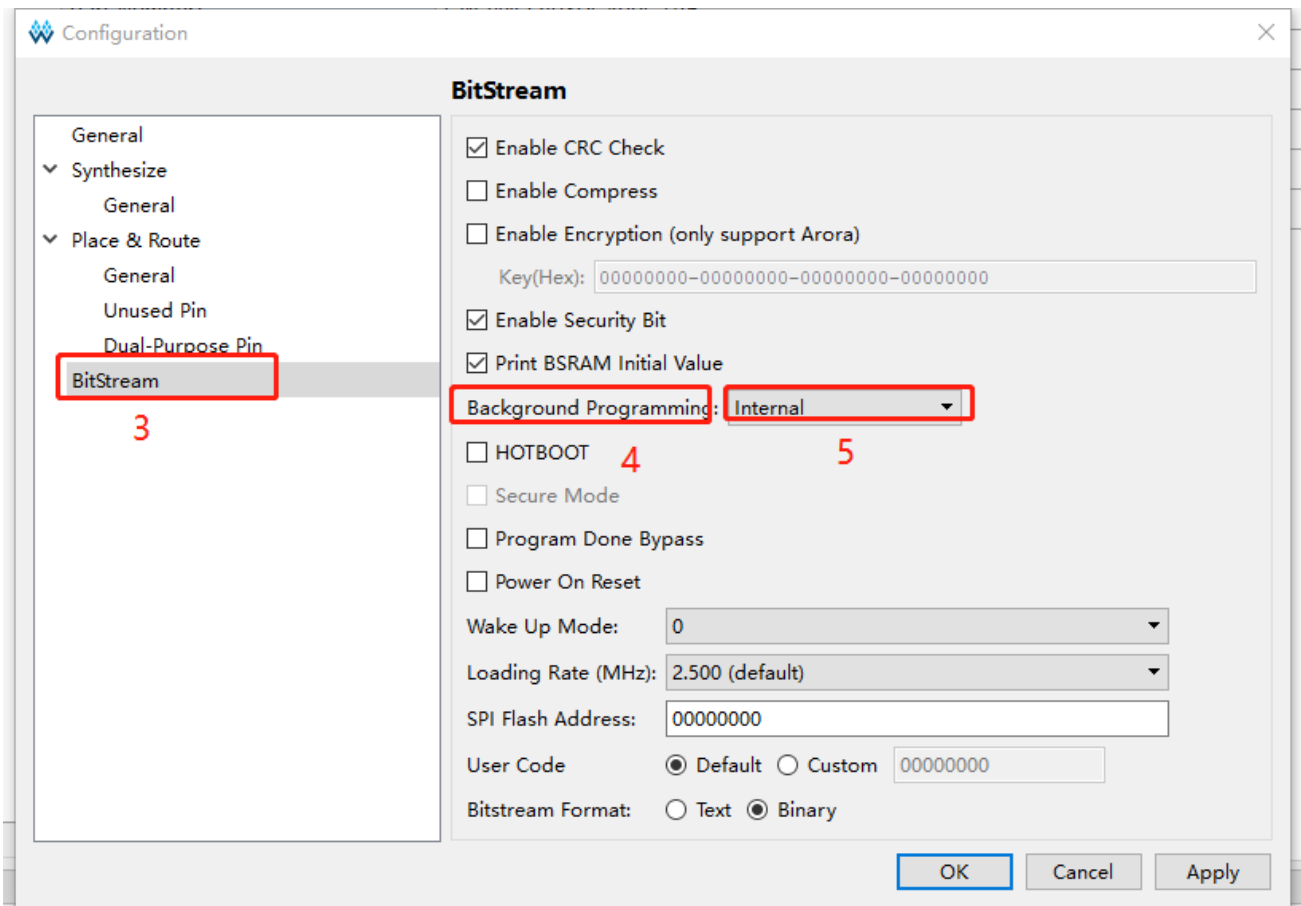


图 5-4 配置背景升级为 I2C



5.2.2 流程说明

Gowin goConfig I2C IP 9X 系列通过操作寄存器，完成对 Flash 的读或者写。其示例流程如下。

写操作流程

Gowin goConfig I2C IP 9X 系列操作进行时，需要先对部分寄存器完成配置，写操作的示例流程如下：

1. 查询 SR 状态寄存器（0x80），当 SPI 端口空闲时（SR-Data[1:0]=00），可以发起操作；
2. 向 DICN（0x01）写入需要发送的字节数；
3. 把待写入的数据依次写入 DIFR（0x03）寄存器；
4. 设置 CSCR（0x00）寄存器值为 0xFF，启动操作（Flash 接口将开始操作）；
5. 查询 SR（0x80）状态寄存器，读取操作状态；
6. 当 Flash 接口操作完成（SR-Data[1:0]=10）后；设置 CSCR（0x00）寄存器值为 0x00，Flash 接口将进入空闲状态，结束本次操作。

读操作流程

Gowin goConfig I2C IP 9X 系列操作进行时，需要先对部分寄存器完成配置，读操作的示例流程如下：

1. 查询 SR 状态寄存器状态（0x80），当 SPI 端口空闲时（SR-Data[1:0]=00），可以发起操作；
2. 向 DICN（0x01）写入需要发送的字节数；
3. 向 DOCN（0x02）写入需要接收的字节数；
4. 待写入的数据依次写入 DIFR（0x03）寄存器；
5. 设置 CSCR（0x00）寄存器值为 0xFF，启动操作；
6. 查询 SR（0x80）状态寄存器，读取操作状态；
7. 当 Flash 接口操作完成（SR-Data[1:0]=10）后；通过读取 DOFR（0x81）寄存器读出数据；
8. 设置 CSCR（0x00）寄存器值为 0x00，Flash 接口将进入空闲状态，结束本次操作。

5.2.3 注意事项

Gowin goConfig I2C IP 9X 系列在使用时的相关注意事项说明如下。

IO 约束

如表 4-2 中说明，Gowin goConfig I2C IP 9X 系列的 IO 信号可以分为两类：无限制 IO 以及有限制 IO（自动）。

- 无限制 IO：与常规使用相同，可以自由选择名字，是否放置到顶层等。
- 有限制 IO：限定 IO 名称，且必须通过顶层输入或输出。

在顶层模块使用“有限制 IO”时，9X 系列的“IO 接口名称”需改成“特定 IO 名字”（Flash 接口专属名称）。Flash 接口专属名称定义以及对应的“有限制 IO”如表 5-1 所示。

表 5-1 对照表

序号	IP 接口名称	Flash接口		描述	备注
		专属名称	方向		
1	GW_BACKGROUND_SCLK_INR	O_flash_ck	Output	IP-Flash接口时钟	顶层使用时，接口方向定义需以Flash接口为准。
2	GW_BACKGROUND_CS_INR	O_flash_cs_n	Output	IP-Flash接口CS输出	
3	GW_BACKGROUND_MOSI_INR	IO_flash_do	Inout	IP-Flash接口数据输出	
4	GW_BACKGROUND_MISO_INR	IO_flash_di	Output	IP-Flash接口数据输入	

顶层模块定义“有限制 IO”时，IO 方向应该与 Flash 接口方向一致。IO 定义示例如图 5-5 所示。

图 5-5 IO 定义示例

```

//
module goConfig_I2C_Top_wrap
(
    //i2c
    inout  GW_BACKGROUND_SDA_INR,
    input  GW_BACKGROUND_SCL_INR,
    //flash IO
    output O_flash_ck, //
    output O_flash_cs_n, //
    inout  IO_flash_do, //
    output IO_flash_di, //
    //Others
    output GW_OSC_CLK
);

goConfig_I2C_Top inst_goConfig_I2C(
    .GW_BACKGROUND_SDA_INR(GW_BACKGROUND_SDA_INR),
    .GW_BACKGROUND_SCL_INR(GW_BACKGROUND_SCL_INR),
    .GW_BACKGROUND_SCLK_INR(O_flash_ck),
    .GW_BACKGROUND_CS_INR(O_flash_cs_n),
    .GW_BACKGROUND_MISO_INR(IO_flash_do),
    .GW_BACKGROUND_MOSI_INR(IO_flash_di),

    .GW_OSC_CLK(GW_OSC_CLK)
);
endmodule

```

注!

云源软件会对此“有限制 IO”进行检测，若选择了背景升级-Internal，但 IO 不存在或不对应，“Place & Route”后结果与预期不一致。

OSC 输出

Gowin goConfig I2C IP 使用了器件内的 OSC 模块。同时，IP 可以提供 20MHz 的时钟输出，可根据需求选择是否使用。

缓存限制

Gowin goConfig I2C IP 使用了 2 个 BSRAM 进行数据缓存，分别作为接收数据与发送数据的缓存。一次读/写的的数据限制为 1~300（单位，字节）。

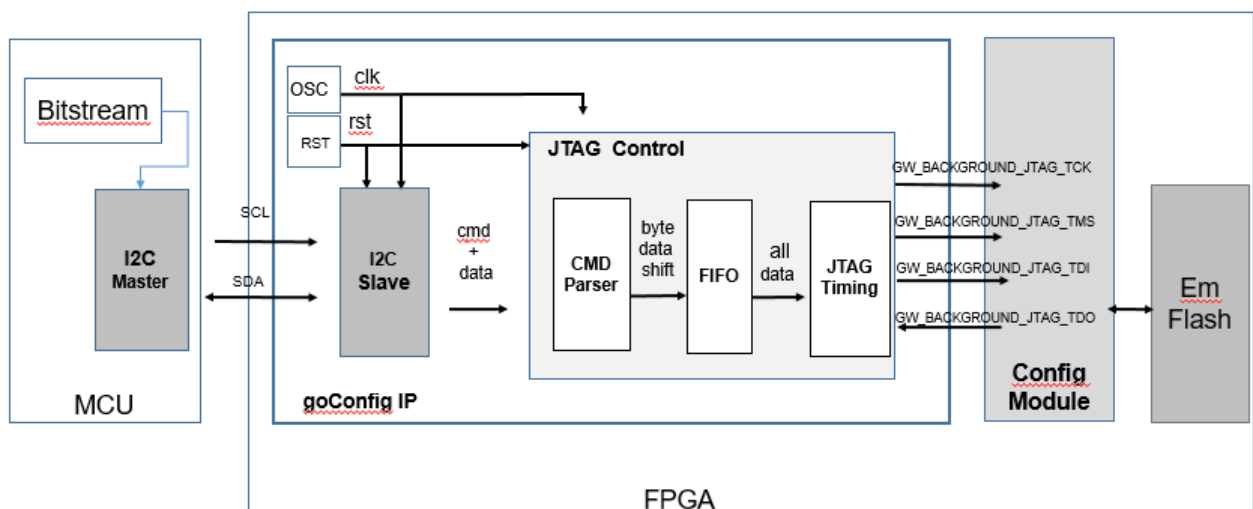
6 实例说明

更详细的实例使用可以参考文档 [TN715, Gowin goConfigIP\(I2C\) 示例代码使用说明](#)，可以联系本公司 FAE 获取示例代码。

6.1 2C 系列实例

本实例的示例代码是在 MCU 中执行，通过 I²C 接口借助 goConfigIP 可以实现对内嵌 Flash 的擦除与编程操作也可以读取 FPGA 的 ID code、user code、status code。

图 6-1 结构框图



6.1.1 检验 MCU 与 IP 通信是否正常

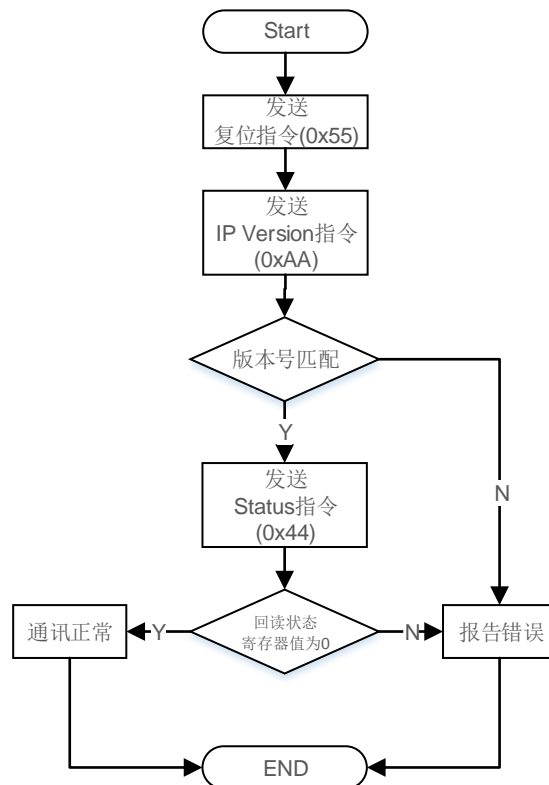
在开始进行背景升级前，先确认 MCU 与 IP 间的通信是否正常，检验流程如下：

1. 发送复位指令（0x55）确保 IP 处于初始状态（API：`gw_i2c_send_reset`）；

2. 发送 IP Version 指令 (0xAA) 来检查读回版本值 (如 Version Data=0x20)以确保 IP 版本与预期的一致 (API:gw_i2c_read_version);
3. 发送读状态寄存器指令 (0x44), 查看读回的状态值 (API: gw_i2c_read_reg_status);
4. 当操作完成并且读回的数据符合预期, 表明与 IP 的通信可以开始。

检验 MUC 与 IP 通信的流程图如图 6-2 所示:

图 6-2 检验 MUC 与 IP 通信的流程图



6.1.2 背景升级整体流程

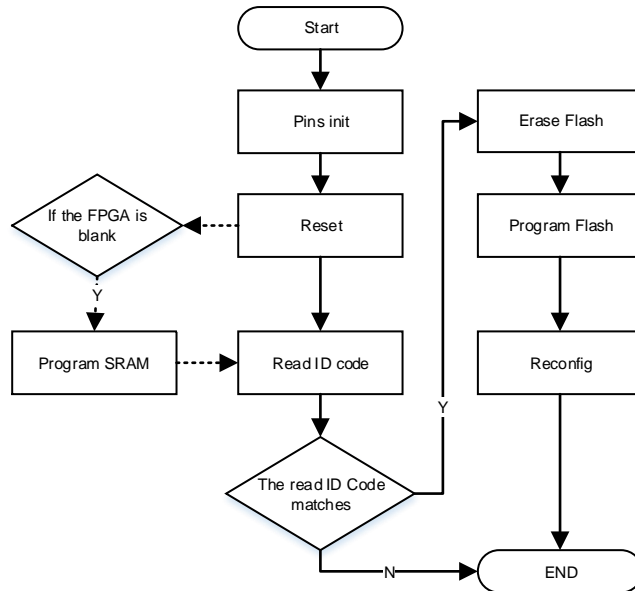
背景升级流程如下所示:

1. MCU 平台引脚初始化 (I2C, debug_uart);
2. IP 复位 (API:gw_i2c_send_reset);
3. 如果芯片不是空白片则忽略此步, 如果芯片是空白片, 先配置 SRAM (API: i2c_configmode_program_sram);
4. 比对 MCU(I2C)读取的 IDcode 与数据流文件的 IDcode 是否一致(API: mach_code), 若一致则继续执行, 不一致结束背景升级;
5. 擦除 Flash (API: gw_i2c_erase_flash_tsmc);
6. 编程 Flash (API: gw_i2c_prog_flash_tsmc);

7. 重新配置 (gw_i2c_send_reconfig_n)。

背景升级流程图如图 6-3 所示。

图 6-3 背景升级流程图



读取 FPGA Code

读取 FPGA code 使用 API: gw_i2c_read_code。当 MCU 与 IP 通讯正常后，可以使用 I2C 读取 FPGA 的 code。

- ID code 即 JEDEC ID Code，是 FPGA 器件的一个基本标识；
- user code 是用户为自己所使用的 FPGA 器件进行的身份标识；
- status code 是 FPGA 器件完成加载数据流文件后的一个状态标识。

例如：

- 读取 ID Code，API 接口为：gw_i2c_read_code(0x11)；
- 读取 user Code，API 接口为：gw_i2c_read_code(0x13)；
- 读取 status Code，API 接口为：gw_i2c_read_code(0x41)。

擦除内嵌 Flash

擦除内嵌 Flash 使用 API: gw_i2c_erase_flash_tsmc。擦除内嵌 Flash 流程如下所示：

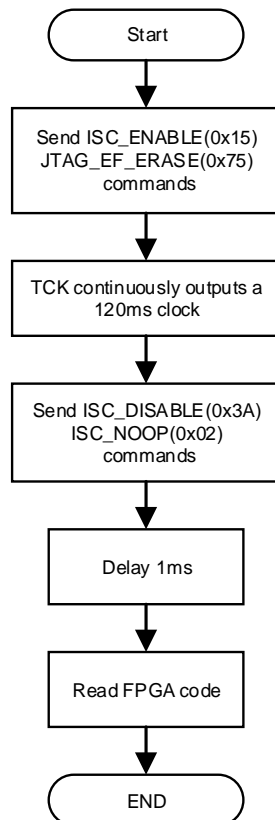
1. 发送 ISC_ENABLE(0x15)和 JTAG_EF_ERASE(0x75)；这两个指令是通过 API (write_inst_and_data) 打包后由 MCU 的 I2C 底层接口 (API: i2c_write_buffer) 发送到 FPGA；
2. TCK 持续输出 120ms clock (API: gw_i2c_send_runttest_xus)；
3. 发送 ISC_DISABLE(0x3A)和 ISC_NOOP(0x02)指令；这两个指令是通

过 API (`write_inst_and_data`) 打包后由 MCU 的 I2C 底层接口 (API: `i2c_write_buffer`) 发送到 FPGA;

4. `delay 1ms` (API: `gw_i2c_send_runttest_xus`);
5. 读取 FPGA code (API: `gw_i2c_read_code`)。

擦除内嵌 Flash 流程图如图 6-4 所示:

图 6-4 擦除内嵌 Flash 流程图



编程内嵌 Flash

编程内嵌 Flash 使用 API: `gw_i2c_prog_flash_tsmc`。编程内嵌 Flash 流程如下所示:

1. 发送 `ISC_DISABLE(0x3A)`和 `ISC_NOOP(0x02)`; 这两个指令是通过 API (`write_inst_and_data`) 打包后由 MCU 的 I2C 底层接口 (API: `i2c_write_buffer`) 发送到 FPGA;
2. 发送 `ISC_ENABLE(0x15)`和 `JTAG_EF_PROGRAM (0x71)`; 这两个指令是通过 API (`write_inst_and_data`) 打包后由 MCU 的 I2C 底层接口 (API: `i2c_write_buffer`) 发送到 FPGA;
3. 发送 X-pgae 地址, 这个地址数据通过 API (`write_one_x_address`)打包后由 MCU 的 I2C 底层接口 (API: `i2c_write_buffer`) 发送到 FPGA;

4. 编程一个 Y-page, 一个 X-page 包含 64 个 Y-page, 这个 Y-page 数据通过 API (write_one_y_page) 打包后由 MCU 的 I2C 底层接口 (API: i2c_write_buffer) 发送到 FPGA;
5. 如果打开回读校验的话需要发送 ISC_ENABLE(0x15) 和 JTAG_EF_READ (0x73); 这两个指令是通过 API (write_inst_and_data) 打包后由 MCU 的 I2C 底层接口 (API: i2c_write_buffer) 发送到 FPGA;
6. 发送 X-page 地址数据 0x00, 这个地址数据通过 API (write_one_x_address) 打包后由 MCU 的 I2C 底层接口 (API: i2c_write_buffer) 发送到 FPGA;
7. 循环从 Flash 回读数据, 并与写入数据流文件做对比 (API: recv_one_y_page_and_verify);
8. 循环回读对比结束再写 autoboot-pattern 数据到第 1 个 X-page, autoboot-pattern 数据通过 API(i2c_prog_xpage) 打包后由 MCU 的 I2C 底层接口 (API: i2c_write_buffer) 发送到 FPGA;
9. 发送 ISC_DISABLE(0x3A)和 ISC_NOOP(0x02)指令; 这两个指令是通过 API (write_inst_and_data) 打包后由 MCU 的 I2C 底层接口 (API: i2c_write_buffer) 发送到 FPGA。

编程内嵌 Flash 流程图如图 6-5 所示:

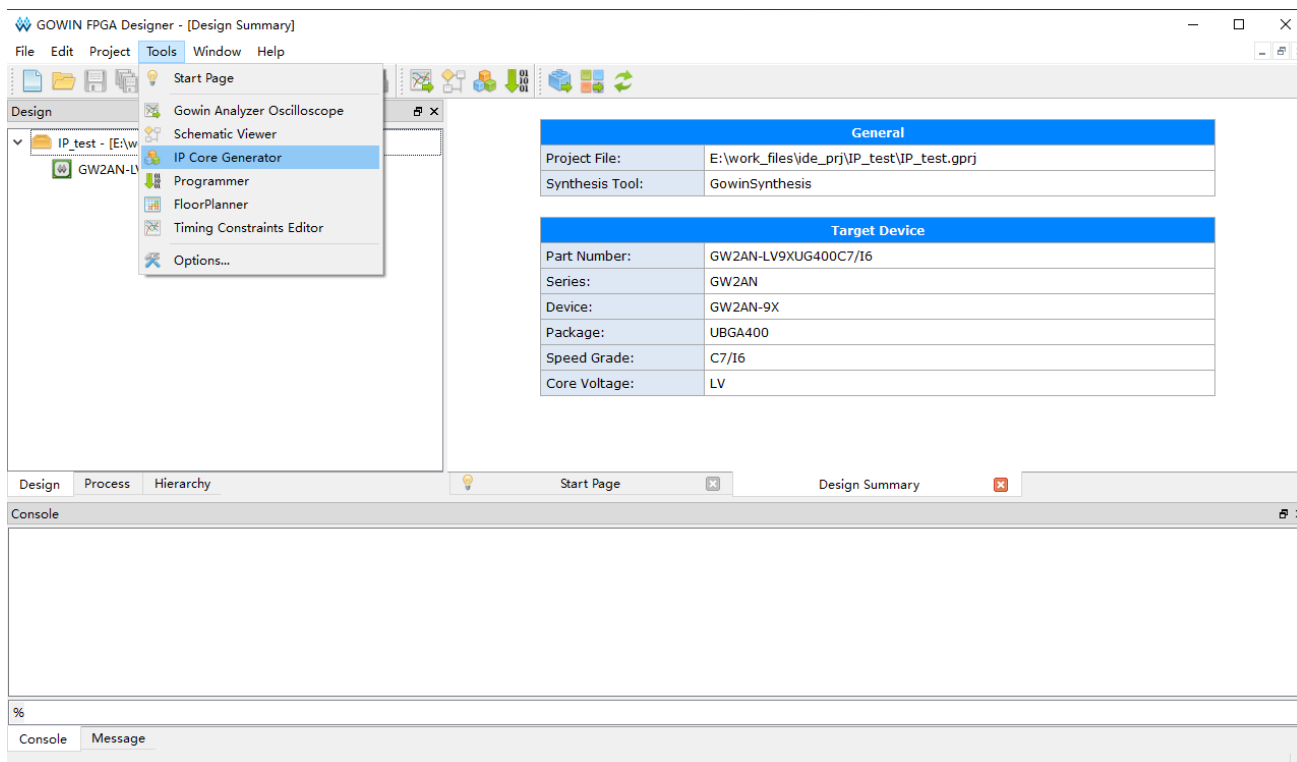
7 界面配置

在高云半导体云源®软件界面菜单栏 Tools 下，可启动 IP Core Generator 工具，完成调用并配置 Gowin goConfig I2C IP。

1. 打开 IP Core Generator

建立工程后，点击左上角“Tools”选项卡，下拉单击“IP Core Generator”选项，就可打开 IP 核产生工具，如图 7-1 所示。

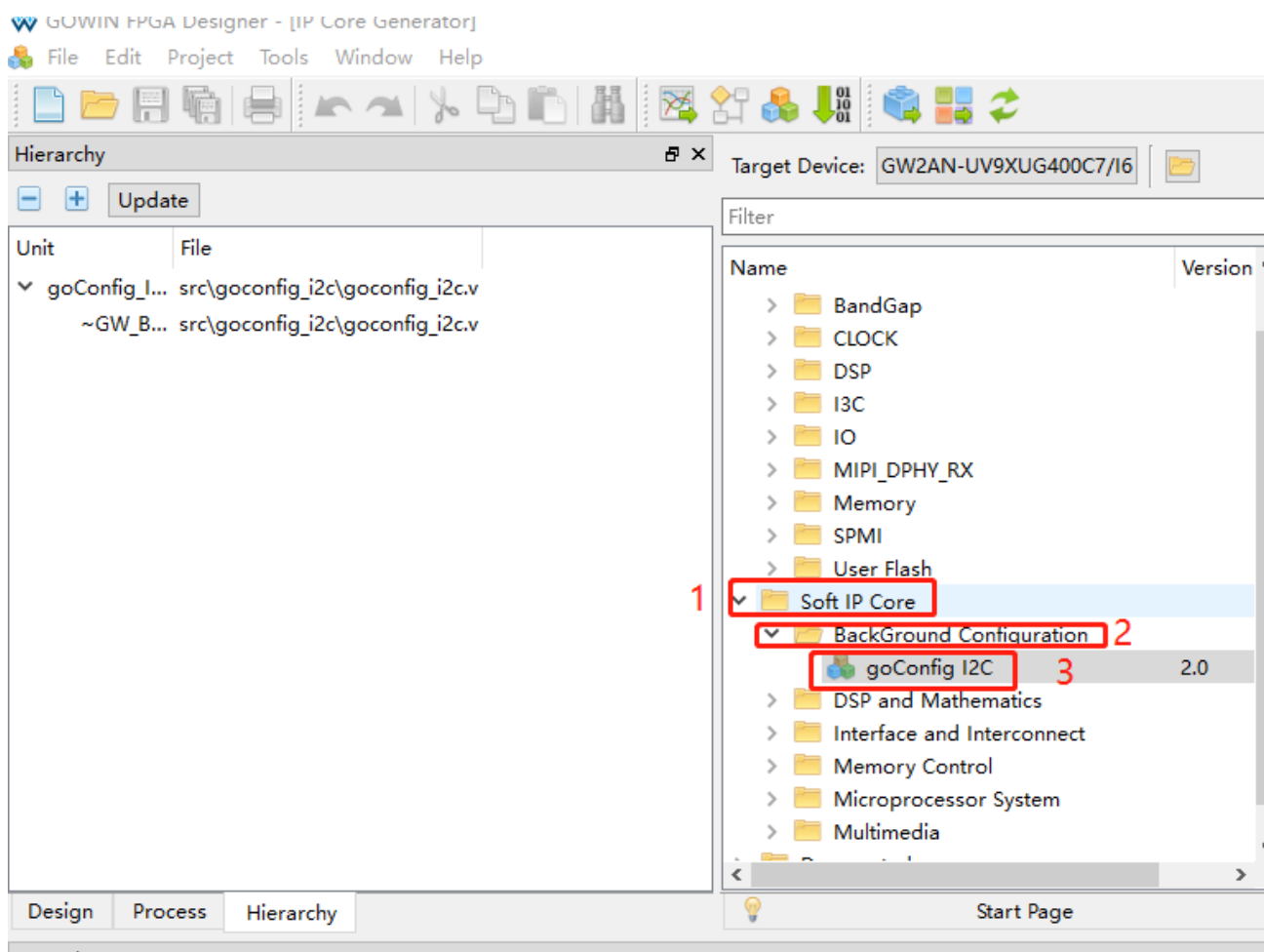
图 7-1 IP Core Generator 选项



2. 打开 Gowin goConfig I2C IP 核

选择“Soft IP Core > BackGround Configuration > goConfig I2C”，如图 7-2 所示，双击即可打开配置界面。

图 7-2 打开 goConfig I2C IP 核

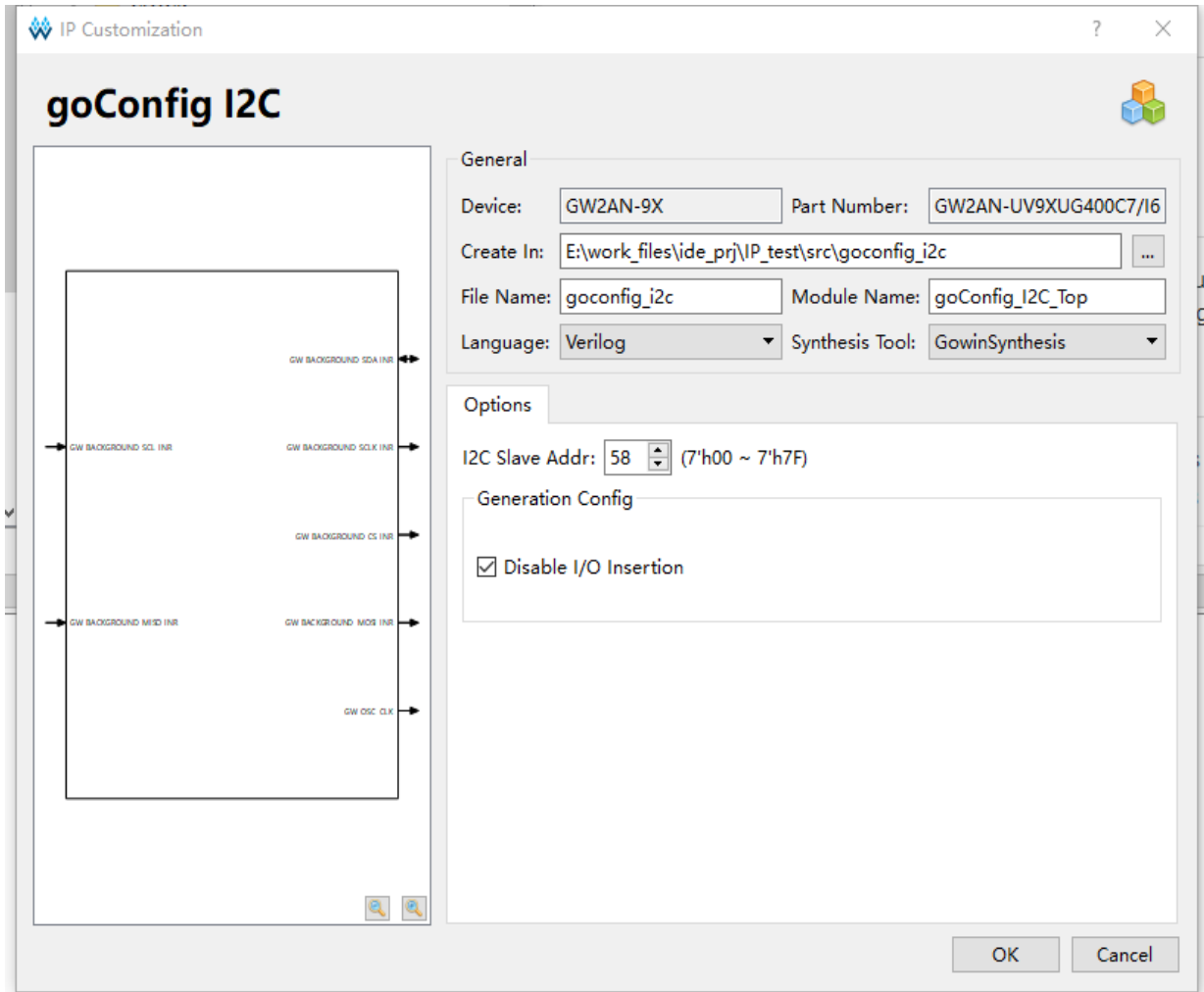


3. Gowin goConfig I2C IP 核配置界面

Gowin goConfig I2C IP 核配置界面如图 7-3 所示。配置界面左侧是 goConfig I2C IP 核的接口示意图，右侧为配置选项。

- 用户可通过修改 File Name，配置产生文件名称；
- 可通过修改 Module Name，配置产生的顶层模块名称；
- 可通过修改 I2C Slave Addr，配置从机地址。

图 7-3 goConfig I2C IP 配置界面



8 参考设计

本节主要介绍 Gowin goConfig I2C IP 参考设计的搭建与使用方法。详细信息请参见高云半导体官网给出的相关参考设计。

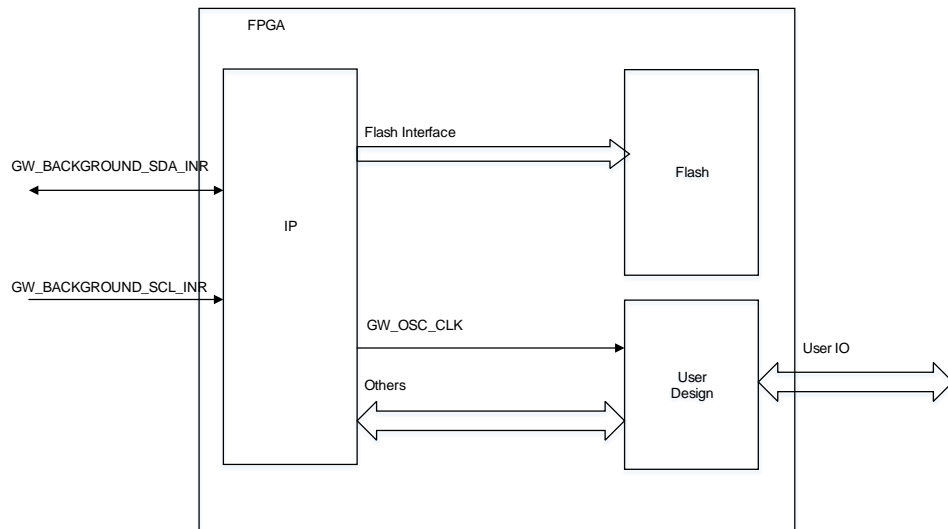
8.1 参考设计

8.1.1 FPGA 工程说明

Gowin goConfig I2C IP 的 FPGA 参考设计主要包含两个组成部分 IP 模块以及用户模块。

- IP 模块：调用 IP，并给出了一种使用方法示例。模块说明如下：
 - IP 的使用需要更改云源软件配置，参考设计中已经包含有对应的示例；详细说明请参考第 5 章 IP 使用说明。
 - IP 的部分 IO 使用时存在限制，可模仿参考设计的使用；详细说明请参考第 4 章 信号定义以及第 5 章的注意事项。
- 用户模块：通过一个简单的 LED 测试，代表用户的设计。

图 8-1 参考设计-基本结构示意图



8.1.2 参考设计文件

Gowin goConfig I2C IP 参考设计中，2C 系列相关文件如表 8-1 所示，9X 系列相关文件如表 8-2 所示。

表 8-1 2C 系列参考设计 src 文件夹内容列表

名称	描述
Top.v	顶层模块，包含LED的测试应用
goConfig_I2C_1p5c.cst	工程物理约束文件
goconfig_i2c	IP文件夹
----goconfig_i2c.v	生成的IP文件，加密。
----goconfig_i2c.vo	生成IP的网表文件，仿真使用。

表 8-2 9X 系列参考设计 src 文件夹内容列表

名称	描述
goConfig_I2C_9X.cst	工程物理约束文件
LED_test	LED 应用文件夹
----LED_TOP.v	顶层模块，
----LED.v	LED 测试模块
goconfig_i2c	IP 文件夹
----goconfig_i2c.v	生成的 IP 文件，加密。
----goconfig_i2c.vo	生成 IP 的网表文件，仿真使用。

8.1.3 参考设计注意事项

Gowin goConfig I2C IP 本次提供的参考设计中，仅包含 FPGA 的工程，以及部分说明文档；不提供仿真环境，以及 I2C 总线的驱动示例。

“I2C 总线的驱动示例”请联系 FAE 获取。

