

# **Logos 系列 FPGA 配置（configuration）**

## **用户指南**

**(UG020005, Version 1.10)**

**（2020.8.4）**

深圳市紫光同创电子有限公司

版权所有侵权必究

## 文档版本修订记录

版本号	发布日期	修订记录
V1.0	2018/1/18	初始版本
V1.1	2018/12/18	封面增加最新的版本更新时间；页脚去掉版本信息
V1.2	2019/8/17	增加器件类型区分；增加表格说明 PGL12G 不支持主 BPI 配置模式
V1.3	2019/10/17	<ol style="list-style-type: none"> <li>1. 新增器件 PGL12G、PGL22GS、PGL25G</li> <li>2. 新增“远程升级”、“SEU 检测”参考文件描述。</li> <li>3. 修改“表 1 Logos 系列 FPGA 配置模式”，新增不同封装的配置功能支持情况。</li> <li>4. 修改描述“数据流”，统一为“位流”。</li> <li>5. 修改信号“VDDIOCFG、VCCJ_3.3V、vddiocfg”为“VCCIOCFG”。</li> <li>6. 修改信号“VDDIOL0、vdd11、vddm、vdd33”为“VCCIOL0、VCC、VDDM、VCCAUX”。</li> <li>7. 修改位宽描述“X”统一为小写“x”。</li> <li>8. 修改“表 22 Master SPI 接口信号描述”信号“D[7:0]”I/O 状态为“IO”。</li> <li>9. 修改“表 5 Master BPI 配置接口信号描述”中“CFG_CLK”描述。</li> <li>10. 修改“表 5 Master BPI 配置接口信号描述”中 32 位地址线“D[31:16]、ADR[31: 16]”为 26 位“ADR[25:0]”。</li> <li>11. 修改 BPI 章节中信号“felk”为“CFG_CLK”。</li> <li>12. 修改图 2、9、10、30、33、40，新增建立保持时间。</li> <li>13. 修改图 9、10、30、33、40，新增信号“RST_N，INIT_FLAG_N”。</li> <li>14. 修改图 9、10、30、33、40，新增“T<sub>RSTN</sub>、T<sub>PL</sub>、T<sub>ICCK</sub> 等信号描述”。</li> <li>15. 修改图 1、5、6、7、8、13、14、15、20、25、29、32、36、41、44，删除按键。</li> <li>16. 修改图 7、8、15、20、25，增加必要的上拉电阻。</li> <li>17. 修改图 11，改正信号名。</li> <li>18. 更新图 48、49、50、51、52、53、62、63。</li> </ol>
V1.4	2019/12/5	<ol style="list-style-type: none"> <li>1. 新增“图 2 PGL25G JTAG 配置模式应用接口”。</li> <li>2. 修改图 1、2、6、7、8、9、14、15、16、21、26、30、33、37、42、45，明确上拉的对应电源。</li> <li>3. 新增第八章第（二）部分，UID 介绍。</li> <li>4. 新增从并回读时序图。</li> <li>5. 修改表 7，增加 CS_N 建立保持时间信息。</li> <li>6. 修改图 31、41、44，更正 T<sub>INIT2CS</sub> 数值。</li> <li>7. 修改图 34、38，更正 T<sub>INIT2CLK</sub> 数值。</li> <li>8. 新增第六章。</li> <li>9. 修改第七章 SEU 描述。</li> <li>10. 修改图 46，更正信号名。</li> <li>11. 新增从 SPI 配置流程及指令列表。</li> <li>12. 新增 INIT_FLAG_N 释放给用户所需条件。</li> <li>13. 新增 JTAG 模式 TMS、TDI、TCK 上拉电阻建议。</li> <li>14. 新增 JTAG 级联时仅支持第一级 FPGA 外部 Flash 操作。</li> <li>15. 新增“三（二）3 不同接口位宽下的位序”说明。</li> <li>16. 新增从 SPI 回读时钟频率限制。</li> <li>17. 更新图 22、23、24、27、28。</li> </ol>
V1.5	2020/1/10	<ol style="list-style-type: none"> <li>1. 新增主 SPI 模式转换介绍</li> <li>2. 修改“版本回退”为“多启动”。</li> </ol>
V1.6	2020/2/23	<ol style="list-style-type: none"> <li>1. 修改“三（一）3”唤醒描述，增加 GLOGEN、GRS_N 对寄存器的影响。</li> </ol>
V1.7	2020/4/1	<ol style="list-style-type: none"> <li>1. 修改 Slave Serial、Slave Parallel 时钟要求。</li> </ol>

V1.8	2020/4/20	<ol style="list-style-type: none"><li>1. 修改加密位流使用描述，默认使能密钥保护无需单独操作。</li><li>2. 修改 UID 描述</li><li>3. 删除远程升级应用文档描述。</li><li>4. 更新图 33，Slave SPI 编程时序。</li><li>5. 修改加密描述，PGL25G 不支持加密功能。</li></ol>
V1.9	2020/5/18	<ol style="list-style-type: none"><li>1. 修改 CFG_CLK IO、专用复用描述。</li><li>2. 修改 MODE 专用复用描述。</li><li>3. 链接 SEU IP。</li></ol>
V1.10	2020/8/4	<ol style="list-style-type: none"><li>1. 修改表 3，更正 CFG_CLK IO 方向。</li><li>2. 新增器件 PGL50G、PGL50H。</li><li>3. 新增 Slave Parallel 菊花链介绍。</li><li>4. 新增内部从并行接口介绍。</li><li>5. 修改表 4，更新 spi flash 支持列表。</li></ol>

## 名词术语解释

JTAG: Joint Test Action Group

SED: Soft Error Detection

CCS: Configuration Control System

SPI: Serial Peripheral Interface

BPI: Byte-wide Peripheral Interface

AES: Advanced Encryption Standard

SEU: Single Event Upsets

ECC: Error Control Coding

SECDED: Single Error Correcting and Double Error Detecting

UID: Unique ID

## 目录

一、 概述.....	1
二、 配置模式描述.....	2
(一) JTAG 配置模式.....	3
(二) Master SPI 配置模式 .....	4
(三) Master BPI 配置模式.....	14
(四) Slave SPI 配置模式 .....	20
(五) Slave Serial 配置模式 .....	24
(六) Slave Parallel 配置模式 .....	29
三、 下载配置.....	35
(一) 配置过程 .....	35
(二) 配置文件 .....	41
(三) 配置数据包 .....	43
(四) 位流格式 .....	55
(五) 下载线速率 .....	60
(六) 配置速率 .....	62
(七) 多功能配置 .....	63
(八) 远程升级 .....	64
(九) 多启动 .....	70
四、 回读操作.....	74
(一) JTAG 接口回读.....	74
(二) Slave Parallel 接口回读 .....	74
五、 回读 CRC.....	78
六、 SEU 检测 .....	79
七、 设计保护.....	80
(一) 位流加密 .....	80
(二) UID .....	82
八、 用户逻辑接口.....	85
(一) 内部从并行接口 .....	85
(二) UID 接口 .....	87
免责声明 .....	89

## 图目录

图 1 PGL12G、22G、22GS JTAG 配置模式应用接口.....	3
图 2 PGL25G、50G、50H JTAG 配置模式应用接口.....	3
图 3 JTAG 编程典型时序图.....	4
图 4 JTAG 级联应用示意图.....	4
图 5 Master SPI 配置接口 Master SPI 配置接口信号描述如下表所示。.....	5
图 6 Master SPI 配置模式 x1 应用示意图 .....	6
图 7 Master SPI 配置模式 x2 应用示意图 .....	6
图 8 Master SPI 配置模式 x4 应用示意图 .....	7
图 9 Master SPI 配置模式 x8 应用示意图 .....	7
图 11 2/4bit Master SPI 编程典型时序图 .....	8
图 18 2/4bit Master SPI 编程典型时序图 .....	11
图 20 Master BPI 配置接口 .....	14
图 32 配置/重配流程 .....	23
图 33 编程流程时序 .....	24
图 47 Slave Parallel 配置接口 .....	32
图 52 下载配置流程图 .....	35
图 53 下载配置时序 .....	35
图 54 唤醒时序 .....	40
图 55 生成位流示意图 .....	41
图 56 Configure Generate Bitstream .....	42
图 57 Generate Flash Programing File .....	42
图 58 Fabric Configuration.....	61
图 59 Connect To Cable .....	61
图 60 USB Cable 速率设置 .....	61
图 61 跳转流程 .....	64
图 62 远程升级一个应用位流流程图 .....	66
图 63 远程升级多个应用位流流程图 .....	69
图 64 黄金位流加载流程 .....	70
图 65 应用位流加载流程 .....	71
图 66 无黄金位流时加载流程 .....	72
图 67 回读 CRC 流程 .....	78
图 68 SEU 流程.....	79
图 69 无位流加密场景 .....	80
图 70 位流加密场景 .....	80
图 71 Configure 界面.....	81
图 72 Configure 设置选项 .....	82
图 73 读 UID 时序 .....	83
图 74 扩展 UID 位宽 .....	83
图 75 GTP_IPAL_E1 端口图.....	85
图 76 GTP_IPAL_E1 内部从并行接口读写接口时序.....	86
图 77 读 UID 时序 .....	88
图 78 扩展 UID 位宽 .....	88

## 表目录

表 1 Logos 系列 FPGA 配置模式 .....	2
表 2 Master SPI 接口信号描述 .....	5
表 3 Master SPI 接口信号描述 .....	9
表 4 支持的 SPI Flash 型号 .....	12
表 5 Master BPI 配置接口信号描述 .....	14
表 6 Master BPI 配置接口信号描述 .....	17
表 7 Slave SPI 配置接口信号描述 .....	20
表 8 配置相关的从 SPI 指令集 .....	22
表 9 Slave Serial 配置接口信号描述 .....	24
表 10 Slave Serial 配置接口信号描述 .....	27
表 11 Slave Parallel 配置接口信号描述 .....	29
表 12 Slave Parallel 配置接口信号描述 .....	32
表 13 GOUTEN 逻辑功能 .....	40
表 14 PDS 配置文件描述 .....	41
表 15 Logos 系列各器件的位流大小 .....	43
表 16 不同位宽下同步字发送顺序 .....	43
表 17 类型 1 包头格式 .....	43
表 18 类型 2 包头格式 .....	44
表 19 配置寄存器说明 .....	44
表 20 器件标识寄存器说明 .....	45
表 21 Logos 系列 FPGA 器件型号 .....	45
表 22 CCS 命令及其描述 .....	45
表 23 控制寄存器 0 说明 .....	46
表 24 控制寄存器 1 说明 .....	46
表 25 选项寄存器 0 说明 .....	47
表 26 选择寄存器 1 .....	48
表 27 状态寄存器说明 .....	49
表 28 看门狗寄存器说明 .....	50
表 29 帧地址寄存器说明 .....	50
表 30 SBPI 寄存器说明 .....	51
表 31 RCR 控制寄存器说明 .....	51
表 32 SEU 控制寄存器说明 .....	52
表 33 SEU 状态寄存器说明 .....	52
表 34 热启动控制寄存器说明 .....	52
表 35 热启动地址寄存器说明 .....	52
表 36 历史状态寄存器说明 .....	53
表 37 位流 0 地址寄存器说明 .....	54
表 38 位流 1 地址寄存器说明 .....	54
表 39 位流 2 地址寄存器说明 .....	54
表 40 位流 3 地址寄存器说明 .....	54
表 41 版本回退寄存器说明 .....	54
表 42 普通位流格式 .....	55
表 43 压缩位流格式 .....	56
表 44 加密位流格式 .....	59
表 45 VS[1:0]的值与位流的对应关系 .....	63
表 46 多功能跳转程序 .....	63
表 47 远程升级位流 .....	65

表 48 多个应用位流格式 .....	67
表 49 热启动流程 .....	72
表 50 有 busy 端口的回读流程.....	74
表 52 有 busy 端口的回读配置寄存器流程.....	76
表 54 GTP_UDID 端口列表.....	83
表 55 GTP_UDID 参数列表.....	83
表 56 GTP_IPAL_E1 端口列表.....	85
表 57 GTP_IPAL_E2 参数列表.....	86
表 58 GTP_UDID 端口列表.....	87
表 59 GTP_UDID 参数列表.....	87



## 一、概述

配置（configuration）是把用户的设计数据（位流）写入FPGA的内部存储器中的过程。配置数据可以由芯片主动从外部Flash获取，也可通过外部处理器/控制器将配置数据下载到芯片中。Logos系列FPGA使用SRAM单元存储配置数据，掉电后配置数据丢失，所以每次上电需要重新对FPGA进行配置。

Logos系列FPGA支持如下六种配置模式：（不同器件支持模式详情见第二章）

- JTAG模式，符合IEEE 1149.1和IEEE 1532标准
- Master SPI模式，支持数据位宽1/2/4/8bits
- Master BPI模式，支持数据位宽8/16bits
- Slave SPI模式，支持数据位宽1bit
- Slave Parallel模式，支持数据位宽8/16/32bits
- Slave Serial模式，支持数据位宽1bit

此外，Logos系列FPGA还提供以下功能：

- 看门狗，支持超时检测
- 支持通过JTAG接口、Slave Parallel接口进行SEU 1-bit纠错和2-bit检错
- 在Master BPI / Master SPI模式下，支持配置位流版本回退功能
- 配置位流压缩可有效减小位流的大小、存储空间和编程时间
- 配置位流加密可防止恶意抄袭，有效保护客户设计知识产权
- JTAG 模式提供专用接口，支持在线调试和边界扫描测试
- 每个器件在出厂前被写入与之对应的唯一编码 — 64bits的UID保护

## 二、配置模式描述

用户可以通过设置MODE的值来选择Logos系列FPGA的配置模式，如表1所示。其中JTAG配置模式优先级最高，MODE设置为任意值都能进行配置。配置时钟CFG\_CLK的方向由配置模式决定，主模式时，CFG\_CLK为输出，由FPGA输出给外部存放位流的设备，如Flash；从模式时，CFG\_CLK为输入，由外部设备（如微处理器，CPLD或其他FPGA）输出给FPGA。本章主要详细描述表1中列出的Logos系列FPGA支持的六种配置模式的应用接口。

表 1 Logos 系列 FPGA 配置模式

序号	配置模式	数据位宽	不同器件支持情况						CFG_CLK 方向
			PGL12G		PGL22G		PGL22GS	PGL25G/50G/50H	
			LPG144	FBG256	FBG256	MBG324	LPG176	ALL	
1	JTAG	1	√	√	√	√	√	√	Input (TCK)
2	Master SPI	1	√	√	√	√	√ <sup>(1)</sup>	√	Output
		2							
		4							
		8						×	
3	Master BPI	8 (异步)	×	×	×	√	×	√	Output
		16 ( 异步)						×	
		16 ( 同步)							
4	Slave SPI	1	√	√	√	√	×	×	Input
5	Slave Parallel	8	√	√	√	√	×	√	Input
		16	×		×				
		32							
6	Slave Serial	1	√	√	√	√	×	√	Input

注：1. 22 GS主SPI模式不支持串行菊花链。

## (一) JTAG 配置模式

JTAG配置模式应用接口示意图如图1、2所示。

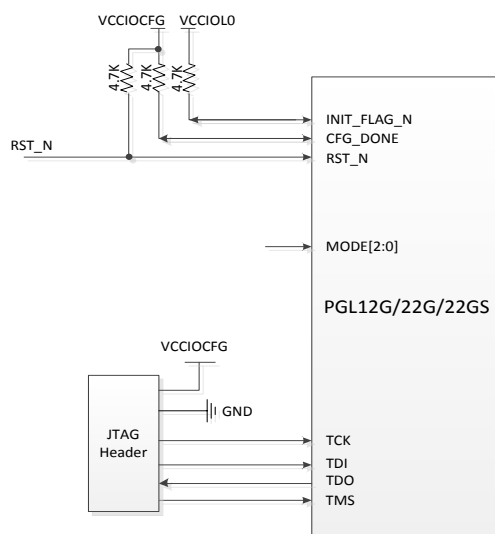


图 1 PGL12G、22G、22GS JTAG 配置模式应用接口

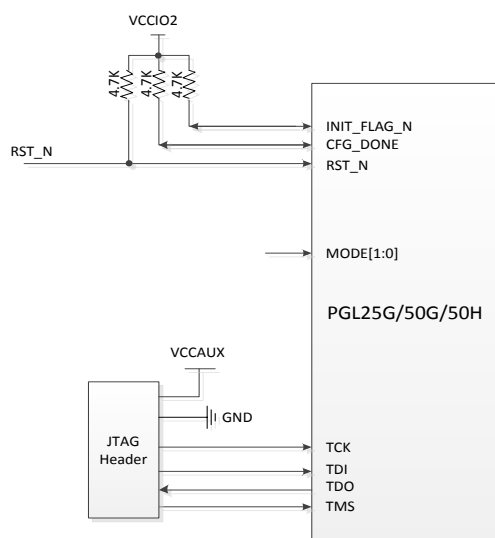


图 2 PGL25G、50G、50H JTAG 配置模式应用接口

RST\_N 低电平将复位配置逻辑。初始化过程中，INIT\_FLAG\_N作为输入时，如果连接低电平将使Logos FPGA停留在初始化阶段；作为输出时，输出高电平指示芯片初始化结束。CFG\_DONE输出高电平表示芯片进入用户模式，外部输入低电平将使Logos FPGA停留在配置阶段。

对于PGL12G、22G、22GS建议将RST\_N、CFG\_DONE采用4.7K的电阻上拉至VCCIOCFG，将INIT\_FLAG\_N采用4.7K的电阻上拉至VCCIOLO。并且将VCCIOCFG、VCCIOLO设置为3.3v。对于PGL25G、50G、50H，建议将RST\_N、CFG\_DONE、INIT\_FLAG\_N采用4.7K的电阻上拉至VCCIO2。

JTAG模式下TCK测试时钟需要由外部提供；外部可以通过改变TMS的状态来控制JTAG内部TAP状态机的跳变，以此选择配置位流的写入（TDI）或者片内数据回读（TDO）。除了配置编程，JTAG接口还常用于在线调试和边界扫描测试。JTAG模式下，建议将TDI、TCK、TMS用15K（±20%）电阻上拉至VCCIOCFG（PGL12G/22G/22GS）或VCCAUX（PGL25G/50G/50H），以提供稳定的初始输入电平。

JTAG编程模式典型时序如下图所示。

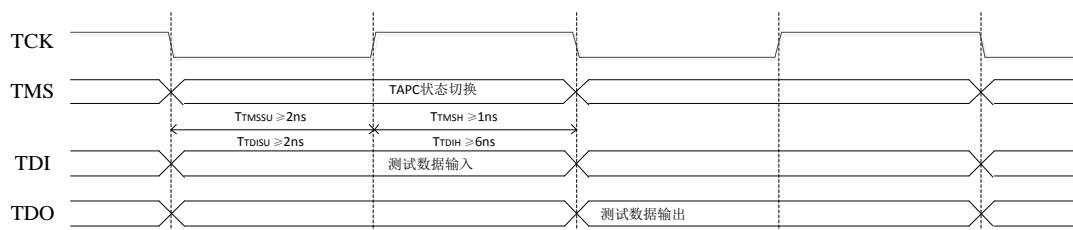


图 3 JTAG 编程典型时序图

## JTAG 级联模式

可用JTAG菊花链配置多个器件，连接如下图所示。

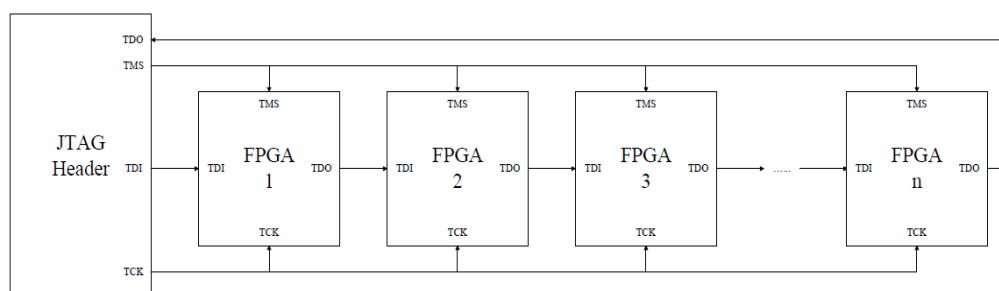


图 4 JTAG 级联应用示意图

此种方式，应将模式管脚设为JTAG模式。连接下载线之后，主机上的PDS软件会扫到在JTAG链上的所有器件，用户可选择对相应的FPGA进行编程下载。

TCK和TMS信号连接了所有JTAG链上的器件，所以这两者的信号质量会影响JTAG配置的最大频率及可靠性。注意，JTAG级联时，仅支持对第一级FPGA的外部Flash进行操作。

## （二）Master SPI 配置模式

Master SPI模式下，位流通常保存在外部SPI Flash中。通过上电或给RST\_N一个低电平脉冲来启动编程，Logos系列FPGA会主动从外部Flash读取位流，CCS自动以初始模式（CFG\_CLK为3.125MHz，数据采样沿为上升沿，fast read x1模式）从Flash的0地址开始取数据。取到SPBI寄存器和操作码后，用操作码规定的模式（CFG\_CLK频率、数据采样沿、数据位宽）继续取数据。PGL12G/22G/22GS支持最多4种功能位流，取到IRSTCTRLR寄存器和操作码后，用操作码规定的模式从Flash取IRSTADRR

寄存器或VS[1:0]管脚指定的位流。而PGL25G、50G、50H不支持多功能位流，取到IRSTADDR寄存器后，从IRSTADDR寄存器地址处读取位流。指定位流全部从Flash取出后，FCS\_N置1，Master SPI模式操作结束。

## 1. PGL12G/22G/22GS

Master SPI配置接口如下所示：

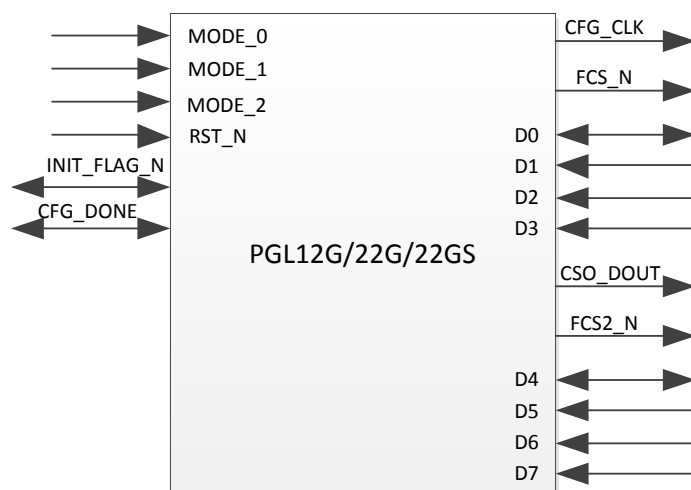


图 5 Master SPI配置接口Master SPI配置接口信号描述如下表所示。

表 2 Master SPI 接口信号描述

名称	I/O	专用/复用	描述
RST_N	I	专用	异步全芯片复位信号，低有效
CFG_CLK	O	复用	配置时钟，用于主 SPI 模式时，最高频率为 50M
MODE[2:0]	I	复用	配置模式管脚 001: 主 SPI 配置模式(x1/x2/x4/x8)
INIT_FLAG_N	开漏	复用	采样MODE[2:0]管脚之前，INIT_FLAG_N是输入， 可以通过保持低电平来延迟配置。 采样MODE[2:0]管脚之后，INIT_FLAG_N是开漏， 指示配置过程是否出错。 0: 错误 1: 正确 FPGA进入用户模式后，如果回读CRC指示使能，该引脚继续做为配置管脚使用。如果回读CRC指示禁止，该引脚释放给用户使用。 <sup>(1)</sup>
FCS_N	O	复用	SPI Flash 芯片选择信号，低有效
FCS2_N	O	复用	x8 模式第二片 SPI Flash 芯片选择信号，低有效
D[7: 0]	IO	复用	串行输入输出数据总线，CFG_CLK 上升沿或下降沿采样 D[7:4]为 x8 模式第二片 SPI Flash 的串行输入输出数据总线
CSO_DOUT	O	复用	菊花链数据输出，CFG_CLK 下降沿发送，PGL22GS 无此管脚，不支持菊花链
CFG_DONE	开漏	专用	指示配置完成 0: FPGA 未配置 1: FPGA 已配置

注：1. 回读CRC指示使能可通过软件进行设置，指示使能不影响回读CRC的功能，用户仍可使

用回读CRC，并通过状态寄存器或内部并行接口查看回读CRC结果。

要选择Master SPI模式，建议将MODE[2:1]通过下拉电阻连接到地，将MODE[0]通过上拉电阻接到VCCIOLO电源。

CFG\_CLK由芯片内部产生并输出；通过监控管脚INIT\_FLAG\_N和CFG\_DONE来判断编程是否结束；Master SPI支持1/2/4/8bit四种数据位宽模式，可以通过对位流的解析来选择数据位宽，针对不同数据位宽，数据bit位选择、数据输入/输出方向会不一样，如下描述：

x1位宽时，FPGA的管脚D[0]作为命令输出连接到SPI Flash的数据输入端，管脚D[1]作为数据输入连接到SPI Flash的数据输出端。应用示意图如下所示：

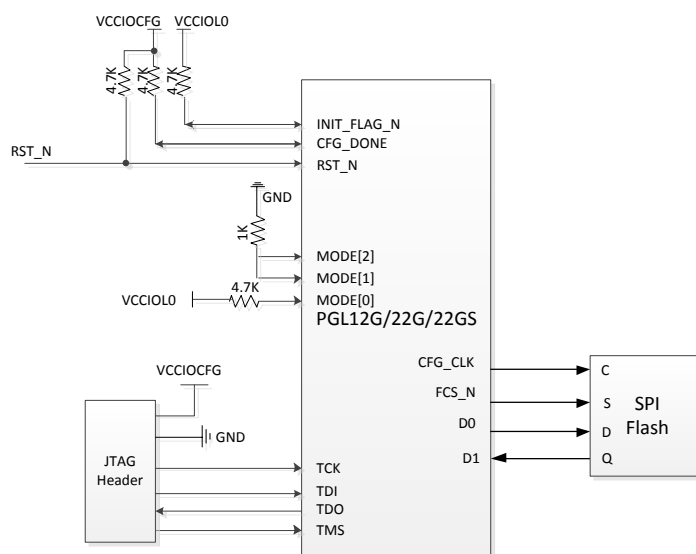


图 6 Master SPI 配置模式 x1 应用示意图

x2位宽时，管脚 D[1:0]为数据总线。应用示意图如下所示：

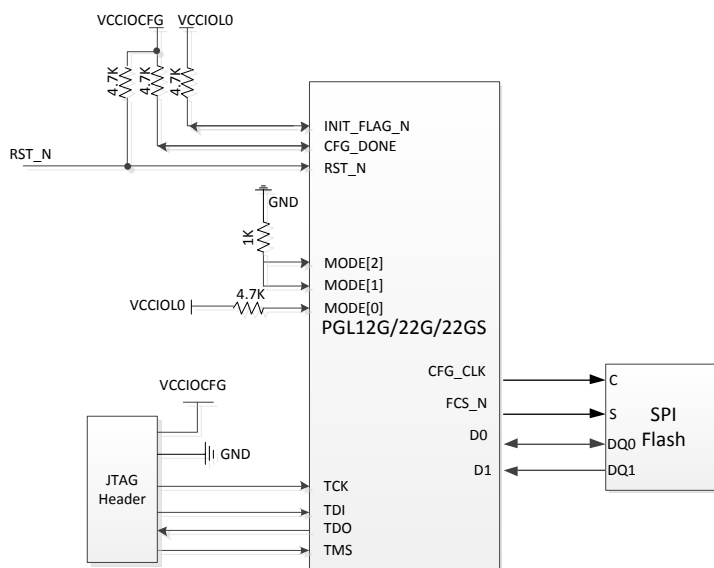


图 7 Master SPI 配置模式 x2 应用示意图

x4 位宽时，管脚 D[3:0]为数据总线。应用示意图如下所示：

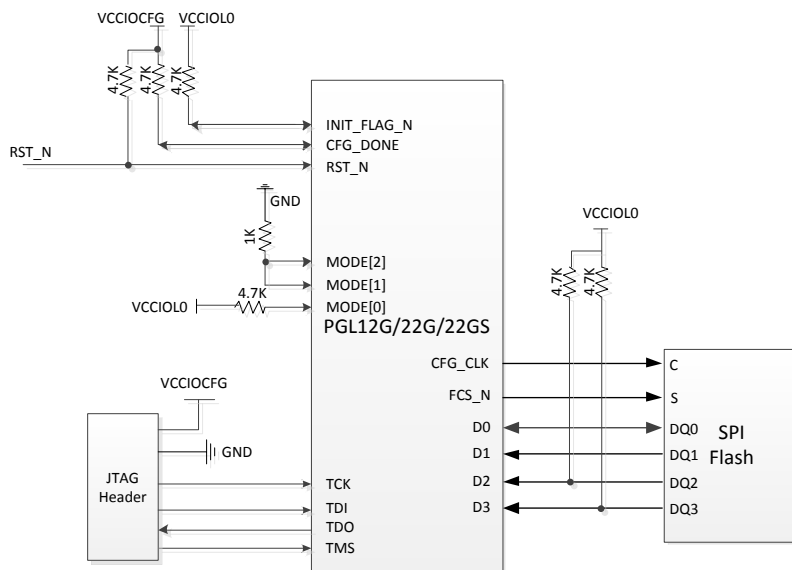


图 8 Master SPI 配置模式 x4 应用示意图

x8 位宽时，管脚 D[3:0]为第 1 片 SPI Flash 的数据总线，管脚 D[7:4]为第 2 片 SPI Flash 的数据总线。应用示意图如下所示：

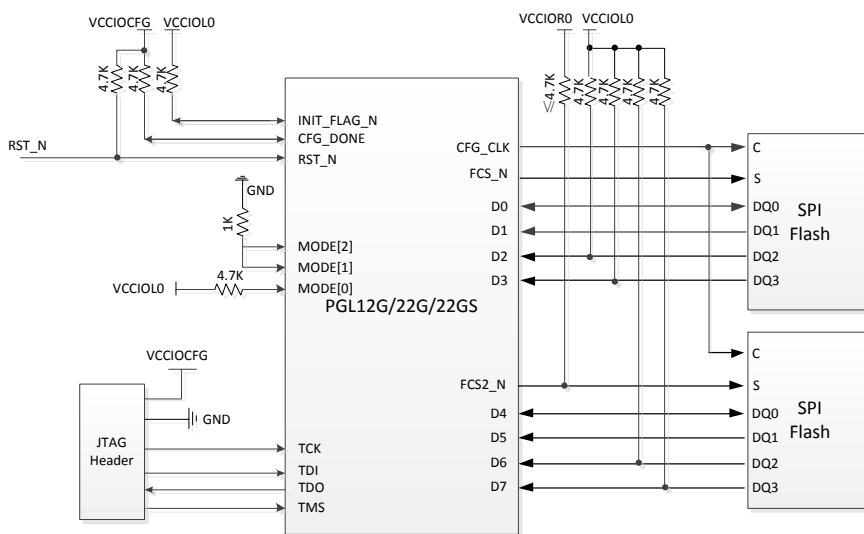


图 9 Master SPI 配置模式 x8 应用示意图

Master SPI编程模式典型时序如下图所示（上升沿采样）：

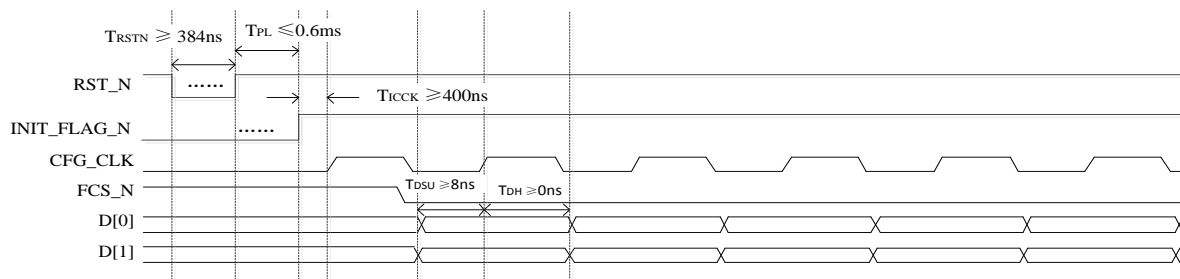


图 10 1bit Master SPI 编程典型时序图

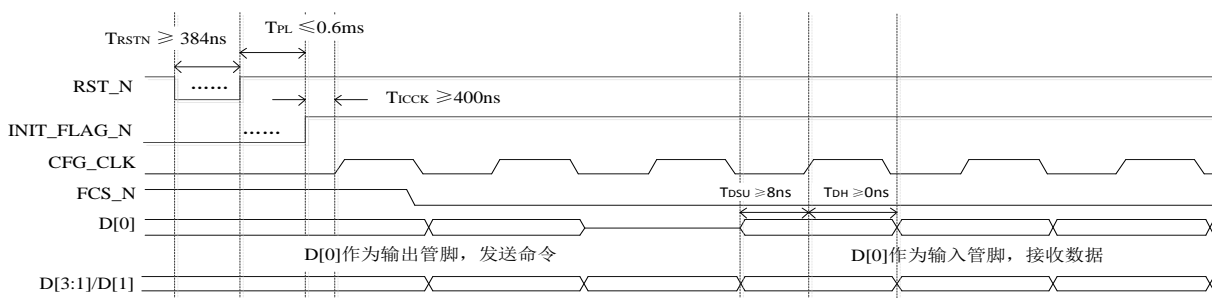


图 11 2/4bit Master SPI 编程典型时序图

## Master SPI 串行菊花链

Master SPI配置模式支持使用Master SPI模式和Slave Serial模式进行SPI串行菊花链配置。SPI串行菊花链中，第一片器件使用Master SPI模式，其它器件使用Slave Serial模式。SPI串行菊花链配置时，从最下级器件开始配置，最后配置第一片器件。逻辑框图如下所示：

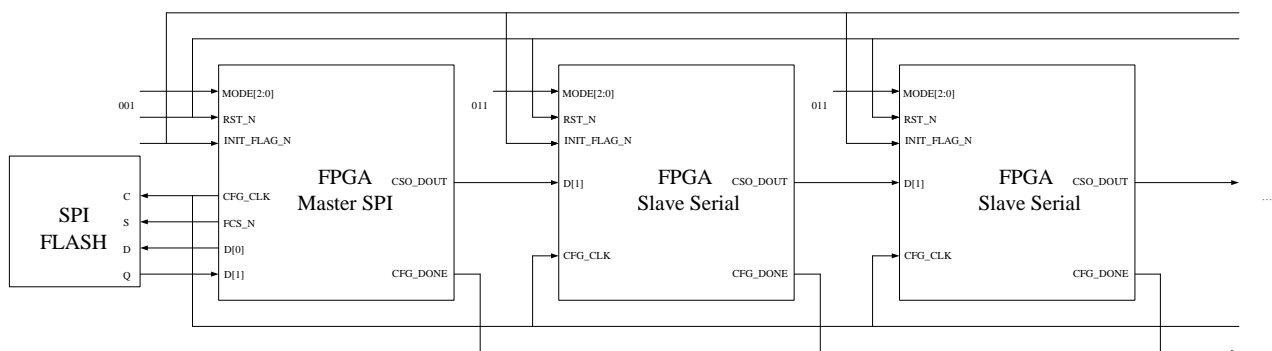


图 12 Master SPI 串行菊花链应用示意图

## 2. PGL25G/50G/50H

Master SPI配置接口如下所示：



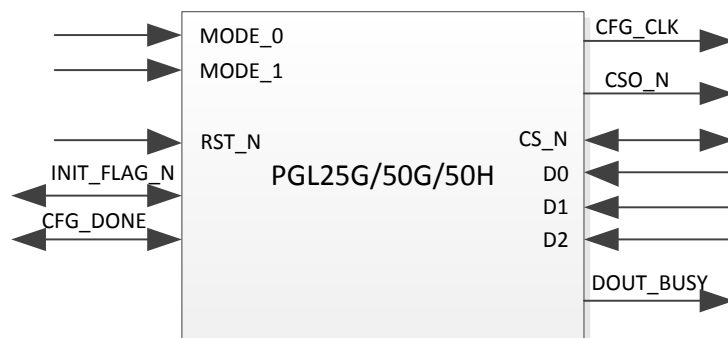


图 13 Master SPI 配置接口

Master SPI配置接口信号描述如下表所示。

表 3 Master SPI 接口信号描述

名称	I/O	专用/复用	描述
RST_N	I	专用	异步全芯片复位信号，低有效
CFG_CLK	O	复用	配置时钟，用于主 SPI 模式时，最高频率为 50M
MODE[1:0]	I	复用	配置模式管脚 01: 主 SPI 配置模式(x1/x2/x4)
INIT_FLAG_N	开漏	复用	采样MODE[1:0]管脚之前，INIT_FLAG_N是输入， 可以通过保持低电平来延迟配置。 采样MODE[1:0]管脚之后，INIT_FLAG_N是开漏， 指示配置过程是否出错。 0: 错误 1: 正确 FPGA进入用户模式后，如果回读CRC指示使能，该引脚继续做为配置管脚使用。如果回读CRC指示禁止，该引脚释放给用户使用。 <sup>(1)</sup>
CSO_N	O	复用	SPI Flash 芯片选择信号，低有效
CS_N	IO	复用	x1 模式时，为主设备输出从设备输入。 x2/x4 配置模式时，为数据总线最低位，CFG_CLK 上升沿或下降沿采样
D[2: 0]	I	复用	串行输入输出数据总线，CFG_CLK 上升沿或下降沿采样 x1 模式时，D[0]为主设备输入从设备输出 x2 模式时，D[0]为数据总线输入高位 x4 模式时，D[2:0]为数据总线输入高 3 位
DOUT_BUSY	O	复用	菊花链数据输出，CFG_CLK 下降沿发送
CFG_DONE	开漏	专用	指示配置完成 0: FPGA 未配置 1: FPGA 已配置

注：1. 回读CRC指示使能可通过软件进行设置，指示使能不影响回读CRC的功能，用户仍可使用回读CRC，并通过状态寄存器或内部并行接口查看回读CRC结果。

要选择Master SPI模式，建议将MODE[1]通过下拉电阻连接到地，将MODE[0]通过上拉电阻连接到VCCIO2电源。

CFG\_CLK由芯片内部产生并输出；通过监控管脚INIT\_FLAG\_N和CFG\_DONE来判断编程是否结束；Master SPI支持1/2/4bit三种数据位宽模式，可以通过对位流的解析来选择数据位宽，针对不同数

据位宽，数据bit位选择、数据输入/输出方向会不一样，如下描述：

x1位宽时，FPGA的管脚CS\_N作为命令输出连接到SPI Flash的数据输入端，管脚D[0]作为数据输入连接到SPI Flash的数据输出端。应用示意图如下所示：

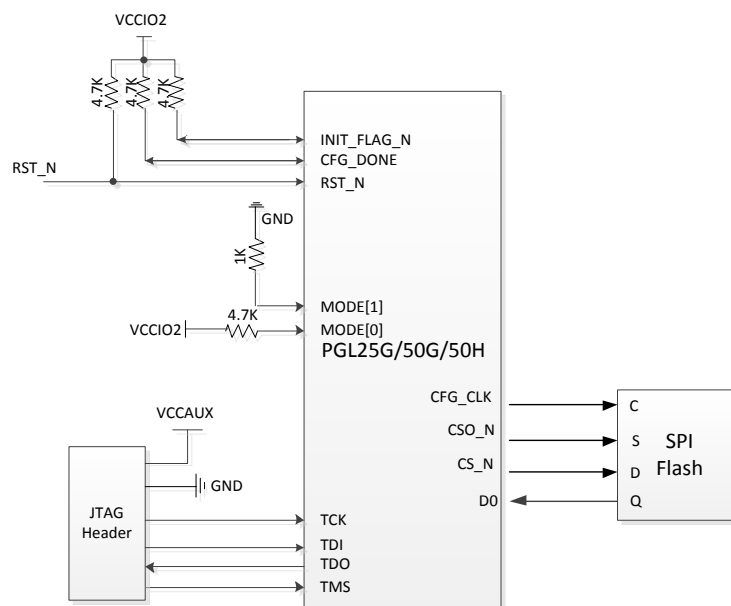


图 14 Master SPI 配置模式 x1 应用示意图

x2 位宽时，管脚 CS\_N、D[0]为数据总线。应用示意图如下所示：

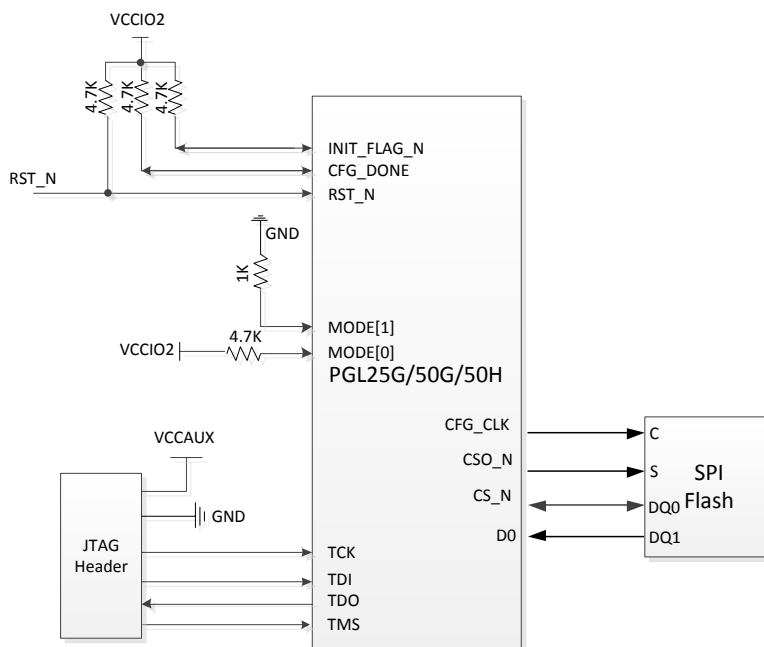


图 15 Master SPI 配置模式 x2 应用示意图

x4 位宽时，管脚 D[3:0]为数据总线。应用示意图如下所示：

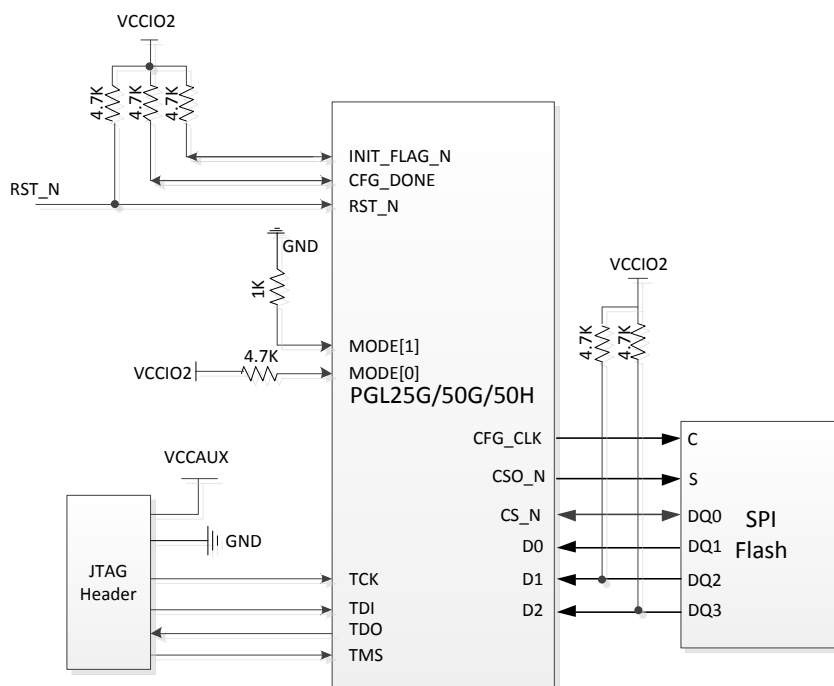


图 16 Master SPI 配置模式 x4 应用示意图

Master SPI编程模式典型时序如下图所示（上升沿采样）：

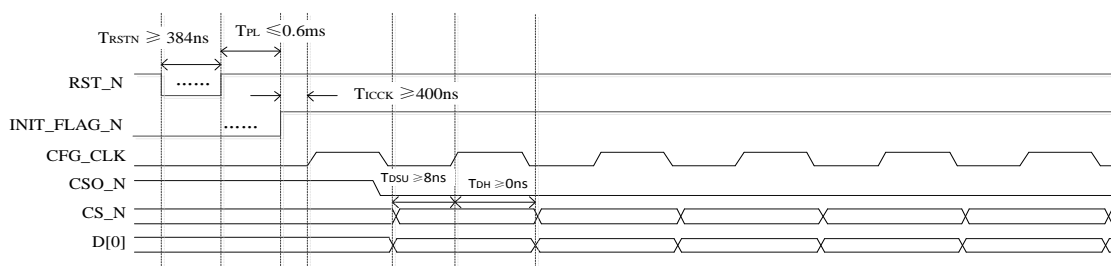


图 17 1bit Master SPI 编程典型时序图

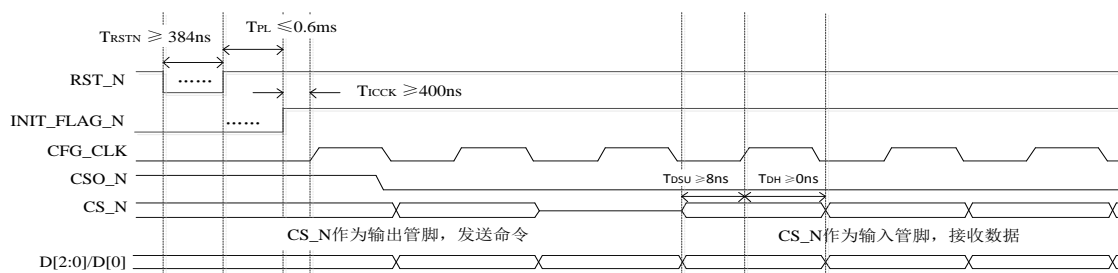


图 18 2/4bit Master SPI 编程典型时序图

## Master SPI 串行菊花链

Master SPI配置模式支持使用Master SPI模式和Slave Serial模式进行SPI串行菊花链配置。SPI串行菊花链中，第一片器件使用Master SPI模式，其它器件使用Slave Serial模式。SPI串行菊花链配置时，

从最下级器件开始配置，最后配置第一片器件。逻辑框图如下所示：

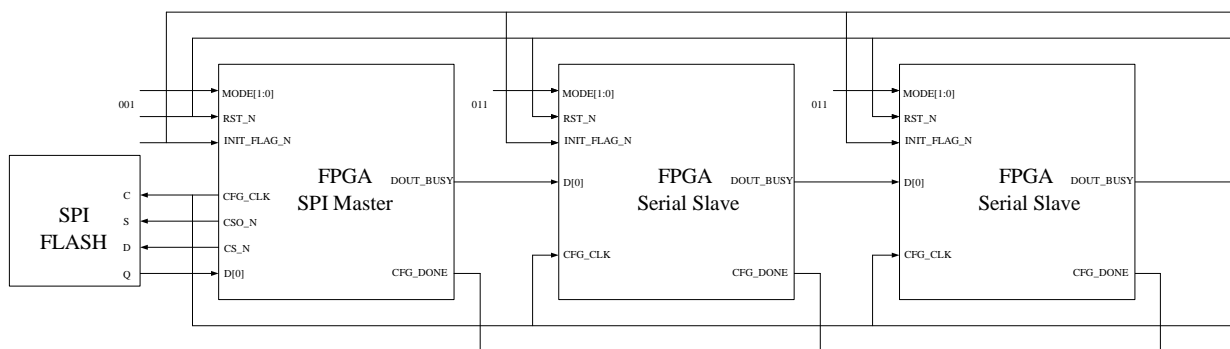


图 19 Master SPI 串行菊花链应用示意图

### 3. SPI Flash 支持列表

Logos系列FPGA支持的部分SPI Flash型号如下表所示。

表 4 支持的 SPI Flash 型号

型号	厂家	容量
AT45DB081E	ADESTO	8Mb
S25FL64	CYPRESS	64Mb
S25FL128	CYPRESS	128Mb
S25FL256	CYPRESS	256Mb
S25FL512	CYPRESS	512Mb
EN25Q80C	ESMT	8Mb
FH25VQ80D	FENTECH	8Mb
M25P16	Micron	16Mb
M25P32	Micron	32Mb
M25P64	Micron	64Mb
M25P128	Micron	128Mb
N25Q32	Micron	32Mb
N25Q64	Micron	64Mb
N25Q128	Micron	128Mb
N25Q256	Micron	256Mb
N25Q512	Micron	512Mb
MD25Q80C	GigaDevice	8Mb
GD25Q80C	GigaDevice	8Mb
GD25Q32C	GigaDevice	32Mb
GD25Q64C	GigaDevice	64Mb
GD25Q128C	GigaDevice	128Mb
GD25Q256D	GigaDevice	256Mb
GD25Q512C	GigaDevice	512Mb
XM25QH16B	武汉新芯	16Mb
XM25QH32B	武汉新芯	32Mb
XM25QH64A	武汉新芯	64Mb
XM25QH128A	武汉新芯	128Mb
XM25QH256B	武汉新芯	256Mb
W25Q40CL	WINBOND	4Mb

W25Q80	WINBOND	8Mb
W25Q16	WINBOND	16Mb
W25Q32	WINBOND	32Mb
W25Q64Q	WINBOND	64Mb
W25Q64M	WINBOND	64Mb
W25Q128Q	WINBOND	128Mb
W25Q128M	WINBOND	128Mb
W25Q256	WINBOND	256Mb
W25Q512	WINBOND	512Mb
IS25LP016D	ISSI	16Mb
IS25LP032D	ISSI	32Mb
IS25LP064D	ISSI	64Mb
IS25LP128D	ISSI	128Mb
IS25LP256D	ISSI	256Mb
IS25LP512D	ISSI	512Mb
IS25WP016D	ISSI	16Mb
IS25WP032D	ISSI	32Mb
IS25WP064D	ISSI	64Mb
IS25WP128D	ISSI	128Mb
IS25WP256D	ISSI	256Mb
IS25WP512D	ISSI	512Mb
MX25L32	MXIC	32Mb
MX25L64	MXIC	64Mb
MX25L128	MXIC	128Mb
MX25L256	MXIC	256Mb
MX25L512	MXIC	512Mb
ZB25VQ16	ZBIT	16Mb
ZB25VQ32	ZBIT	32Mb
SM25QH256M	SSMEC	256Mb

注：表中的型号是当前做过测试的Flash，若使用其它型号需要进行测试验证。选用Flash时需根据实际存储空间需求（如实际位流大小，Logos器件普通位流大小见[表15](#)）进行考虑。

### (三) Master BPI 配置模式

#### 1. PGL22G

Master BPI配置接口如下所示：

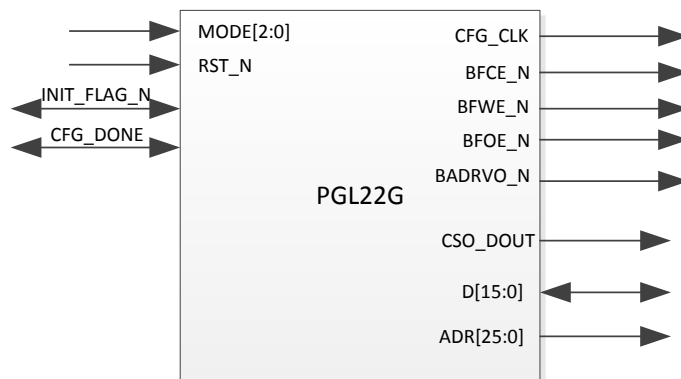


图 20 Master BPI 配置接口

Master BPI配置接口信号描述如下表所示。

表 5 Master BPI 配置接口信号描述

名称	I/O	专用/复用	描述
RST_N	I	专用	异步全芯片复位信号，低有效
CFG_CLK	O	复用	配置时钟，用于主BPI模式时，最高频率为50M
MODE[2:0]	I	复用	配置模式管脚 010：主BPI配置模式（异步x8/异步x16/同步x16）
INIT_FLAG_N	开漏	复用	采样MODE[2:0]管脚之前，INIT_FLAG_N是输入，可以通过保持低电平来延迟配置。 采样MODE[2:0]管脚之后，INIT_FLAG_N是开漏，指示配置过程是否出错。 0：错误 1：正确 FPGA进入用户模式后，如果回读CRC指示使能，该引脚继续做为配置管脚使用。如果回读CRC指示禁止，该引脚释放给用户使用。 <sup>(1)</sup>
BFCE_N	O	复用	BPI Flash片选，低有效
BFWE_N	O	复用	BPI Flash写使能，低有效
BFOE_N	O	复用	BPI Flash输出使能，低有效
BADRVO_N	O	复用	BPI Flash地址有效，低有效，如果外接flash不支持地址有效信号输入，那么不需要连接。
D[15:0]	IO	复用	并行输入输出数据总线
ADR[25:0]	O	复用	BPI Flash26位地址总线，字节寻址
CSO_DOUT	O	复用	菊花链片选输出。上一片器件的菊花链片选输出连接下一片器件的从并行模式片选输入
CFG_DONE	开漏	专用	指示配置完成 0：FPGA未配置 1：FPGA已配置

注：1. 回读CRC指示使能可通过软件进行设置，指示使能不影响回读CRC的功能，用户仍可使用回读CRC，并通过状态寄存器或内部并行接口查看回读CRC结果。

要选择Master BPI模式，建议将MODE[2]和MODE[0]通过下拉电阻连接到地，将MODE[1]通过上拉电阻连接到VCCIO0电源。应用示意图如下所示：

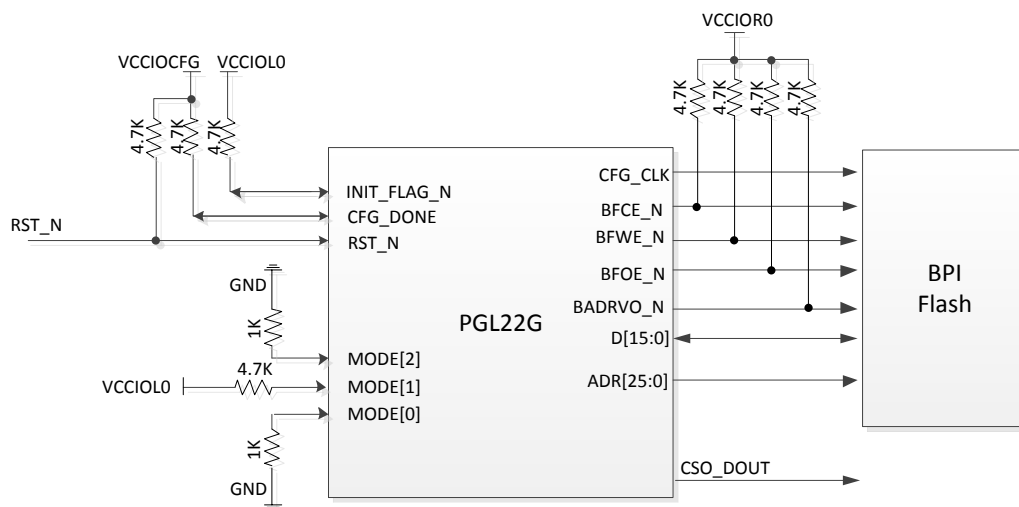


图 21 Master BPI 配置模式应用示意图

Master BPI模式支持异步读，异步页读和同步读。

## 异步读

使用异步模式时，FPGA不需提供时钟给Flash，但是FPGA依然通过CFG\_CLK产生以及采样信号。FPGA在CFG\_CLK的下降沿产生Flash片选等控制信号，发送操作码，数据和地址。FPGA根据设置，在CFG\_CLK的上升沿或下降沿采样数据。时钟频率不高于25M时，使用CFG\_CLK的上升沿采样数据，时钟频率高于25M时，使用CFG\_CLK的下降沿采样数据。异步读时序图如下所示：

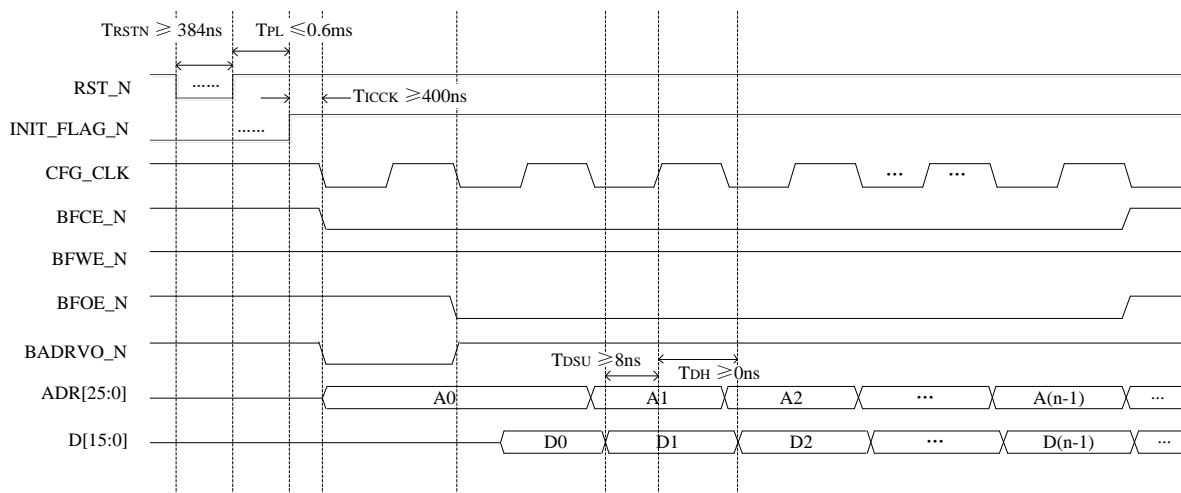


图 22 Master BPI异步读时序

Master BPI模式选中后，CCS自动使用异步读模式从BPI Flash的0地址开始取数据。取到SBPI寄存

器后，如果SBPI寄存器设置为异步读模式，用异步读模式从BPI Flash取IRSTCTRLR寄存器或VS[1:0]管脚指定的位流。指定位流全部从Flash取出后，BFCE\_N置1，Master BPI模式操作结束。

远程升级，多功能配置和版本回退时，如果SBPI寄存器设置为异步读模式，用异步读模式从BPI Flash取IRSTCTRLR寄存器或VS[1:0]管脚指定的位流。指定位流全部从Flash取出后，BFCE\_N置1，Master BPI模式操作结束。

## 异步页读

以一页4个字节/半字为例。每页第一个字节/半字的读时间比页中其它字节/半字的读时间长。可通过软件设置第一个字节的读取时钟数为1~4个时钟。时序如下所示：

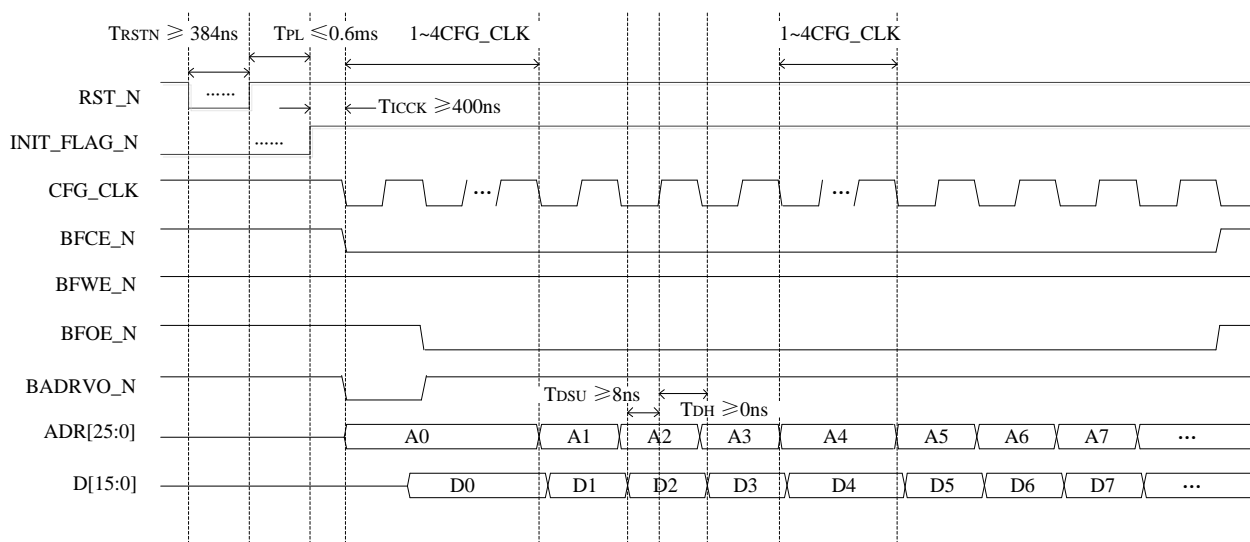


图 23 Master BPI 异步页读时序图

## 同步读

FPGA在CFG\_CLK下降沿产生Flash片选等控制信号，发送操作码，数据和地址。

Flash在CFG\_CLK的上升沿采样操作码和地址。

同步回读时，Flash在CFG\_CLK的上升沿产生数据。FPGA根据设置，在CFG\_CLK的上升沿或下降沿采样数据。时钟频率不高于25M时，使用CFG\_CLK的下降沿采样数据，时钟频率高于25M时，使用CFG\_CLK的上升沿采样数据。同步读时序如下所示：



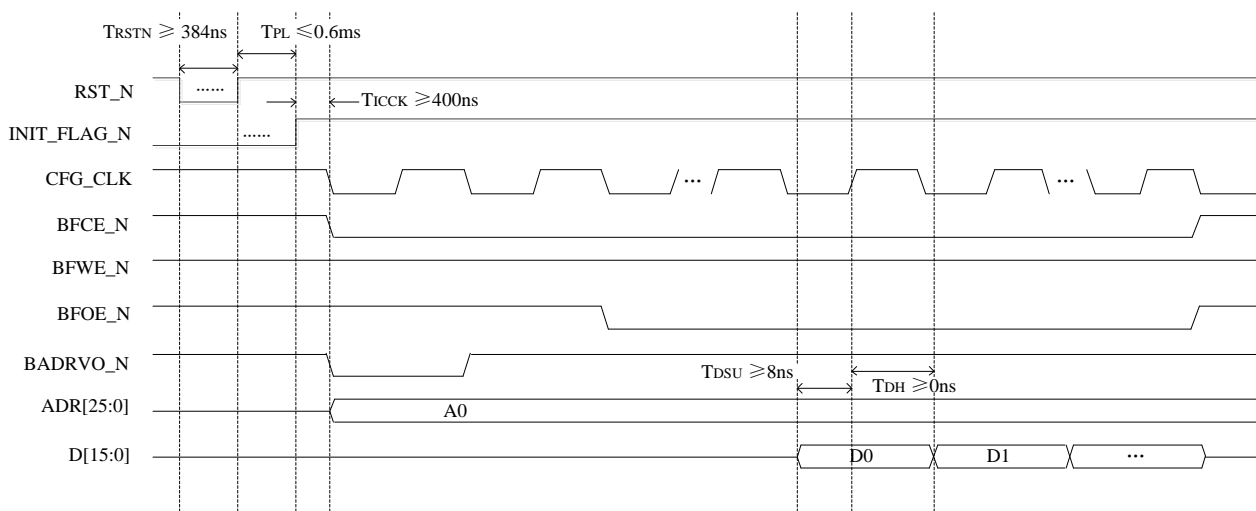


图 24 Master BPI 同步读时序图

Master BPI模式选中后，CCS自动使用异步读模式从BPI Flash的0地址开始取数据。取到SBPI寄存器后，如果SBPI寄存器设置为同步读模式，接下来CCS首先向BPI Flash发送编程读配置寄存器指令，然后用同步读模式从BPI Flash取IRSTCTRLR寄存器或VS[1:0]管脚指定的位流。指定位流全部从Flash取出后，BFCE\_N置1，Master BPI模式操作结束。

远程升级，多启动和版本回退时，如果SBPI寄存器设置为同步读模式，CCS首先向BPI Flash发送编程读配置寄存器指令，然后用同步读模式从BPI Flash取IRSTCTRLR寄存器或VS[1:0]管脚指定的位流。指定位流全部从Flash取出后，BFCE\_N置1，Master BPI模式操作结束。

## 2. PGL25G/50G/50H

Master BPI配置接口如下所示：

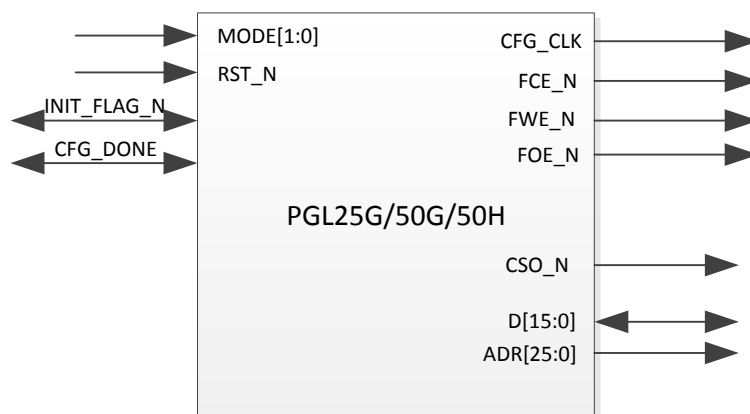


图 25 Master BPI 配置接口

Master BPI配置接口信号描述如下表所示。

表 6 Master BPI 配置接口信号描述

名称	I/O	专用/复用	描述
RST_N	I	专用	异步全芯片复位信号，低有效
CFG_CLK	O	复用	配置时钟，用于主BPI模式时，最高频率为33M

名称	I/O	专用/复用	描述
MODE[1:0]	I	复用	配置模式管脚 00: 主BPI配置模式（异步x8/异步x16）
INIT_FLAG_N	开漏	复用	采样MODE[1:0]管脚之前，INIT_FLAG_N是输入，可以通过保持低电平来延迟配置。 采样MODE[1:0]管脚之后，INIT_FLAG_N是开漏，指示配置过程是否出错。 0: 错误 1: 正确 FPGA进入用户模式后，如果回读CRC指示使能，该引脚继续做为配置管脚使用。如果回读CRC指示禁止，该引脚释放给用户使用。 <sup>(1)</sup>
BFCE_N	O	复用	BPI Flash片选，低有效
BFWE_N	O	复用	BPI Flash写使能，低有效
BFOE_N	O	复用	BPI Flash输出使能，低有效
D[15:0]	IO	复用	并行输入输出数据总线
ADR[25:0]	O	复用	BPI Flash 26位地址总线，字节寻址
CSO_N	O	复用	菊花链片选输出。上一片器件的菊花链片选输出连接下一片器件的从并行模式片选输入
CFG_DONE	开漏	专用	指示配置完成 0: FPGA未配置 1: FPGA已配置

注：1. 回读CRC指示使能可通过软件进行设置，指示使能不影响回读CRC的功能，用户仍可使用回读CRC，并通过状态寄存器或内部并行接口查看回读CRC结果。

要选择Master BPI模式，建议将MODE[1]和MODE[0]通过下拉电阻连接到地。应用示意图如下所示：

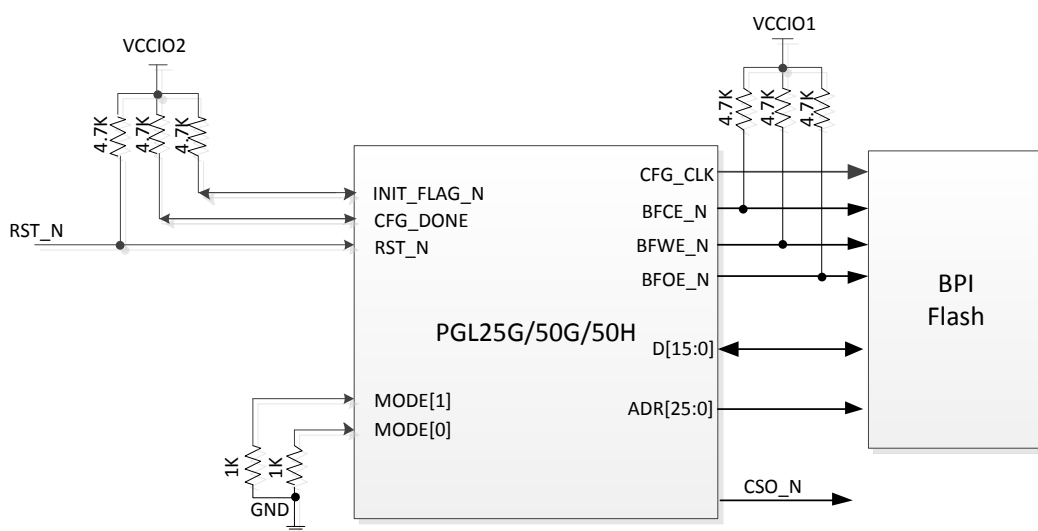


图 26 Master BPI 配置模式应用示意图

Master BPI模式支持异步读，异步页读和同步读。

## 异步读

使用异步模式时，FPGA不需提供时钟给Flash，但是FPGA依然通过CFG\_CLK产生以及采样信号。FPGA在CFG\_CLK的下降沿产生Flash片选等控制信号，发送操作码，数据和地址。FPGA根据设置，在CFG\_CLK的上升沿或下降沿采样数据。时钟频率不高于25M时，使用CFG\_CLK的上升沿采样数据，时钟频率高于25M时，使用CFG\_CLK的下降沿采样数据。异步读时序图如下所示：

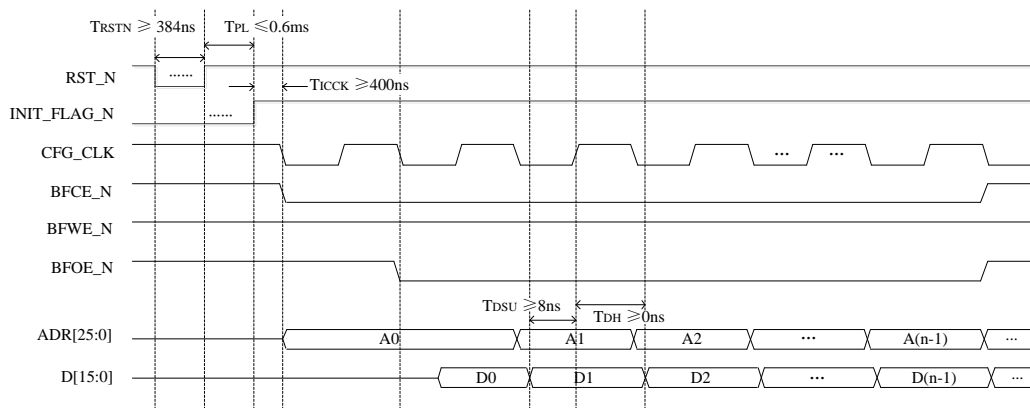


图 27 Master BPI 异步读时序

Master BPI模式选中后，CCS自动使用异步读模式从BPI Flash的0地址开始取数据。取到SBPI寄存器后，如果SBPI寄存器设置为异步读模式，用异步读模式从BPI Flash取IRSTADRR寄存器指定的位流。指定位流全部从Flash取出后，BFCE\_N置1，Master BPI模式操作结束。

远程升级和版本回退时，如果SBPI寄存器设置为异步读模式，用异步读模式从BPI Flash取IRSTADRR寄存器指定的位流。指定位流全部从Flash取出后，BFCE\_N置1，Master BPI模式操作结束。

## 异步页读

以一页4个字节/半字为例。每页第一个字节/半字的读时间比页中其它字节/半字的读时间长。可通过软件设置第一个字节的读取时钟数为1~4个时钟。时序如下所示：

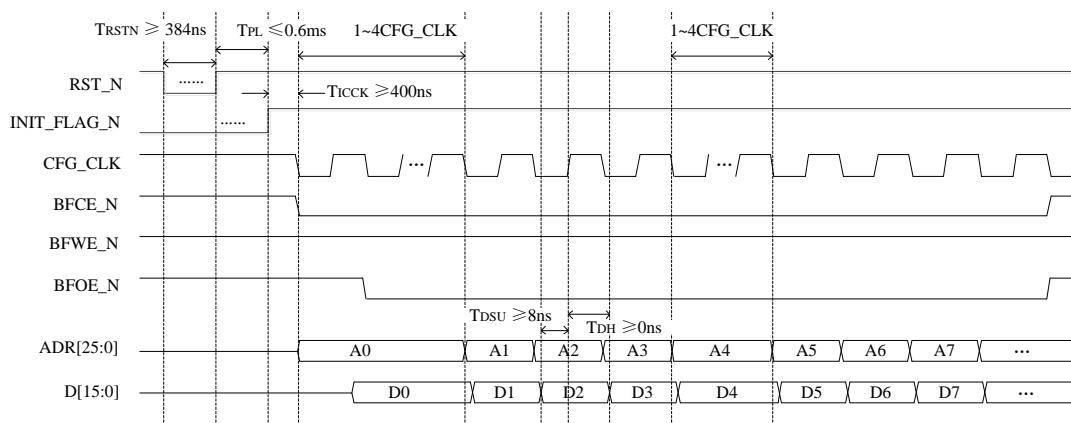


图 28 Master BPI 异步页读时序图

#### (四) Slave SPI 配置模式

仅PGL12G/22G支持Slave SPI配置模式，Slave SPI配置接口如下所示：

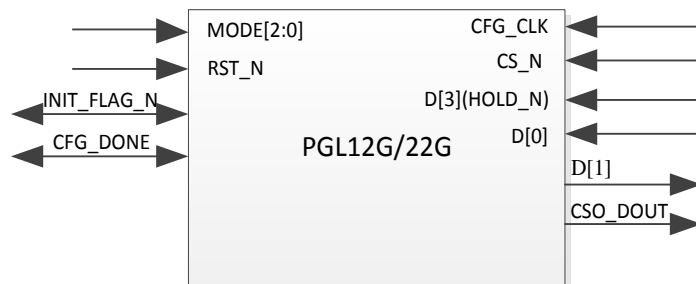


图 29 Slave SPI 配置接口

Slave SPI配置接口描述如下表所示。

表 7 Slave SPI 配置接口信号描述

名称	I/O	专用/复用	描述
RST_N	I	专用	异步全芯片复位信号，低有效
CFG_CLK	I	复用	配置时钟，用于从SPI模式时，写入最高频率为100M，回读最高频率为60M
MODE[2:0]	I	复用	配置模式管脚 101: 从SPI配置模式(x1)
INIT_FLAG_N	开漏	复用	采样MODE[2:0]管脚之前，INIT_FLAG_N是输入， 可以通过保持低电平来延迟配置。 采样MODE[2:0]管脚之后，INIT_FLAG_N是开漏， 指示配置过程是否出错。 0: 错误 1: 正确 FPGA进入用户模式后，如果回读CRC指示使能，该引脚继续做为配置管脚使用。如果回读CRC指示禁止，该引脚释放给用户使用。 <sup>(1)</sup>
CS_N	I	复用	FPGA芯片选择信号，低有效，CFG_CLK上升沿采样，建立保持时间分别为3ns和1ns。
D[3] (HOLD_N)	I	复用	保持信号，低有效，用于暂停主设备与FPGA的通信
D[1]	O	复用	串行输出数据总线
D[0]	I	复用	串行输入数据总线
CSO_DOUT	O	复用	菊花链数据输出，CFG_CLK下降沿发送
CFG_DONE	开漏	专用	指示配置完成 0: FPGA未配置 1: FPGA已配置

注：1. 回读CRC指示使能可通过软件进行设置，指示使能不影响回读CRC的功能，用户仍可使用回读CRC，并通过状态寄存器或内部并行接口查看回读CRC结果。

Slave SPI配置模式时，管脚D[0]为主设备输出从设备输入，管脚D[1]为主设备输入从设备输出，管脚D[3]为芯片保持。

选择Slave SPI配置模式，建议将MODE[2]和MODE[0]通过电阻上拉到VCCIO\_L0，将MODE[1]通过下拉电阻连接到地。Slave SPI配置模式应用示意图如下所示：

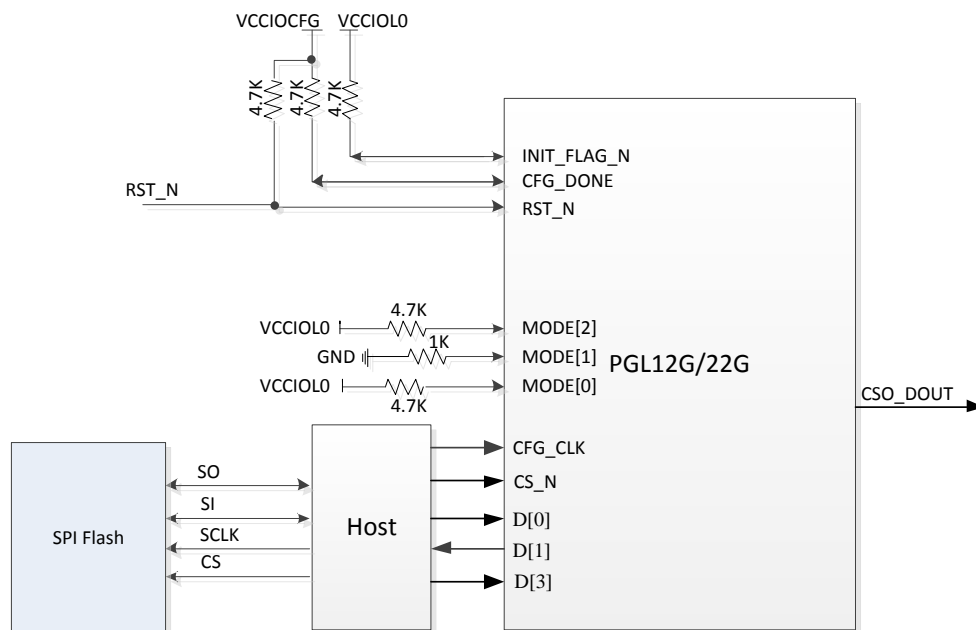


图 30 Slave SPI 模式应用示意图

Slave SPI模式下，可以通过一个主控芯片（Host）控制板上多个芯片的上电启动，Host可以是微处理器、CPLD或者其他FPGA；Slave SPI模式下CFG\_CLK由外部提供。

该模式下可以通过上电或给RST\_N一个低电平脉冲来启动编程；

通过监控管脚INIT\_FLAG\_N和CFG\_DONE来判断编程是否结束；

Slave SPI编程模式典型时序如下图所示。

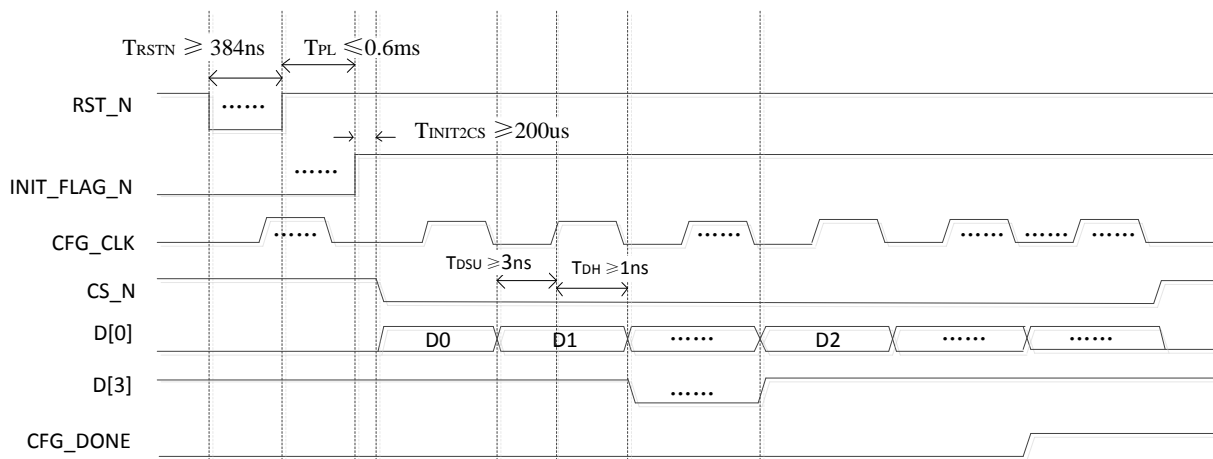


图 31 Slave SPI 配置时序图

## 指令集

表 8 配置相关的从 SPI 指令集

指令	描述	操作码
NOP	无操作	FF
RDID	读IDCODE	A1
RDSR	读状态寄存器	A3
PROGRAM	配置位流	50
WREN	写使能	51
WRDIS	写禁止	52
ERASE	复位FPGA	60

## 配置/重配流程

- 1, 将值3'b101输入给配置管脚MODE[2:0], 并且使配置管脚的值3'b101在整个配置过程中保持不变
- 2, 上电
- 3, 待INIT\_FLAG\_N的值为1后, 写RDID指令, 读出器件IDCODE
- 4, 检查IDCODE的值
- 5, 如果IDCODE不匹配, 终止操作。如果IDCODE匹配, 写WREN指令
- 6, 写入PROGRAM指令, 载入数据流
- 7, 数据流发送完成后, 如果用户模式下从SPI接口释放给用户使用 (persist为0), 察看INIT\_FLAG\_N和CFG\_DONE的值, 操作结束。如果用户模式下从SPI接口做为配置接口使用, 写入WRDIS指令
- 8, 写入RDSR指令, 读出器件状态寄存器

上电后的任何时刻, 可以对 FPGA 进行冷启动 (将 RST\_N 置 0)。冷启动后, 可以进入步骤 3, 对 FPGA 进行重配。

(MODE[2:0] = 101)

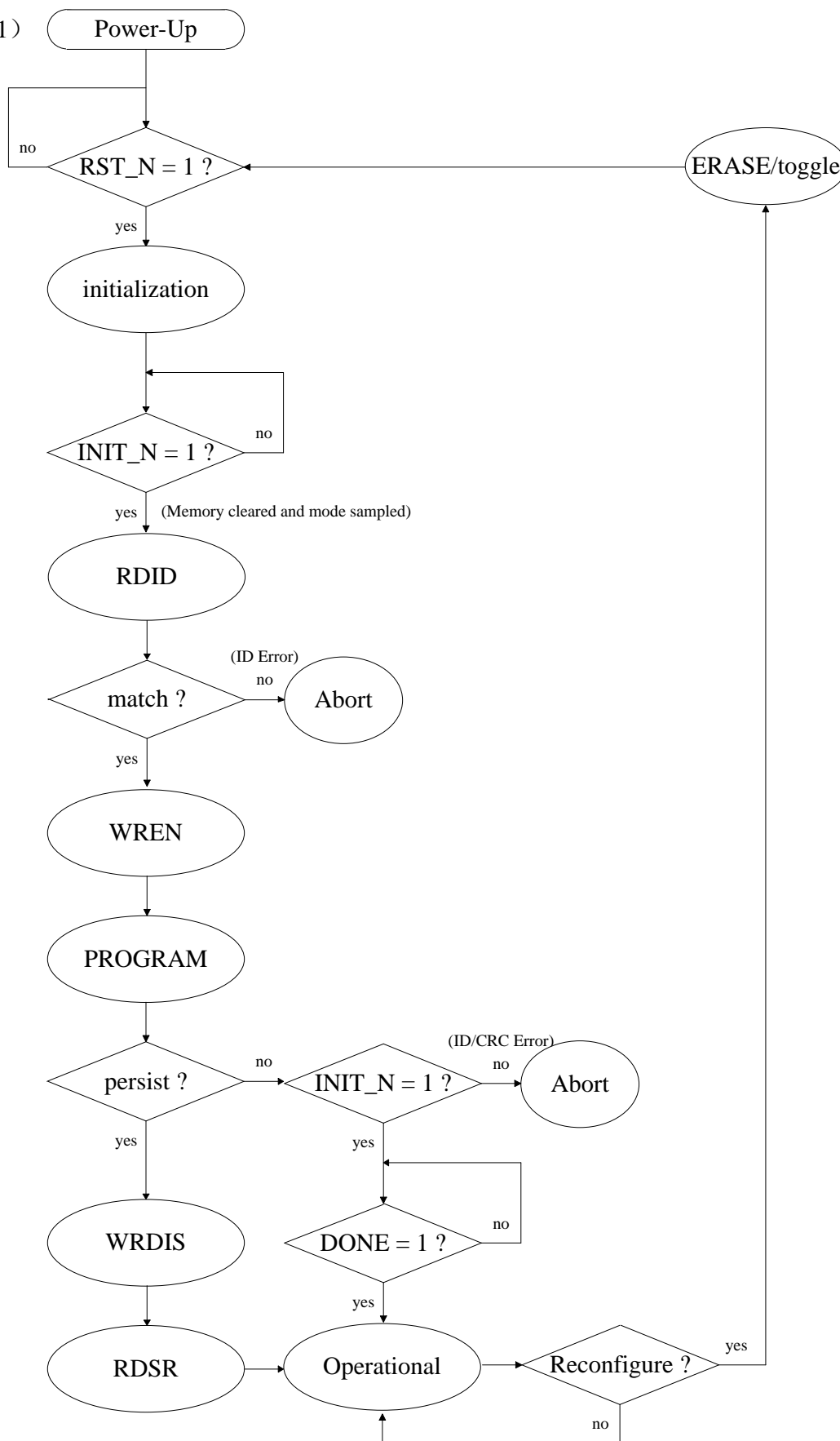


图 32 配置/重配流程

指令与指令之间需要拉高 CS\_N 至少一个周期，指令均为高位先进。编程流程时序如下图所示。

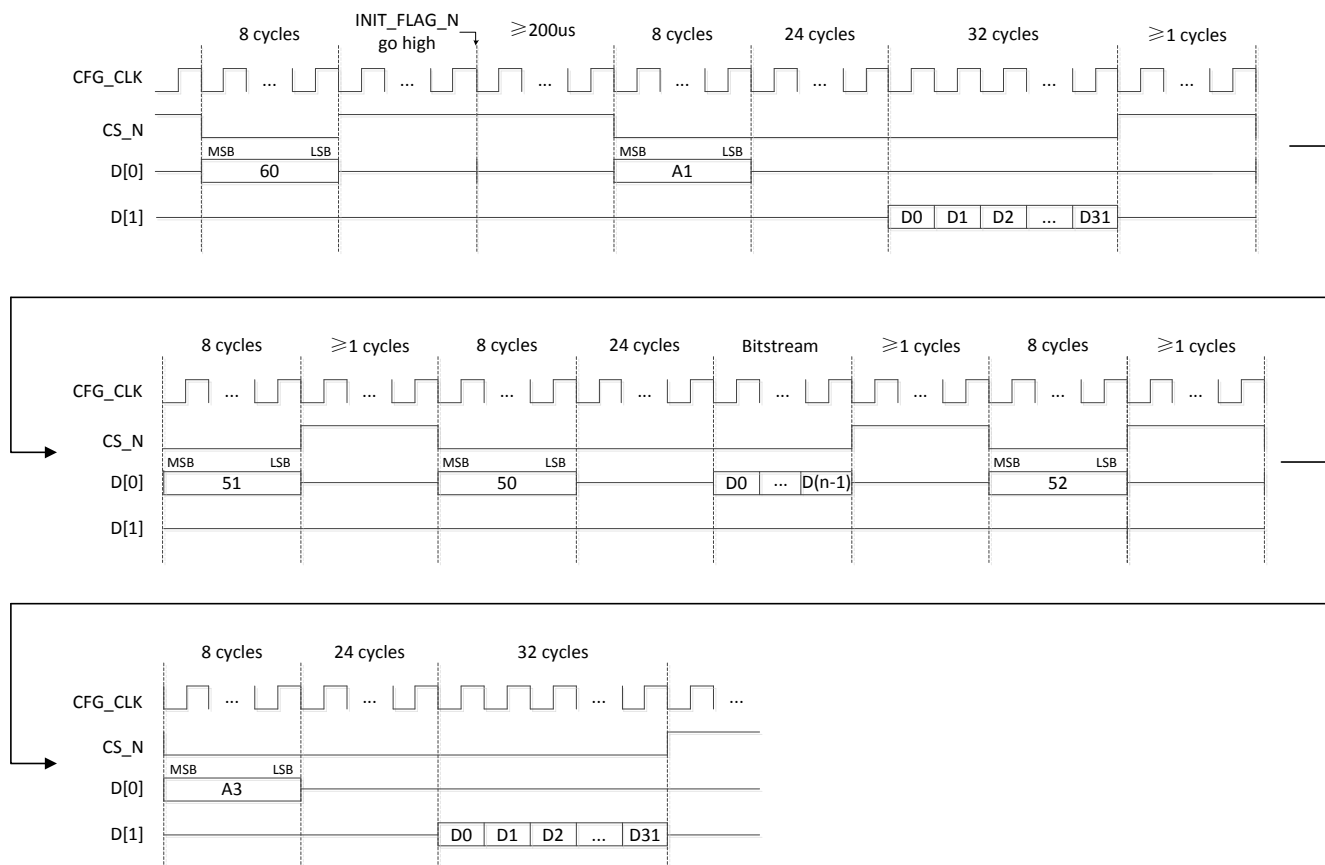


图 33 编程流程时序

## （五）Slave Serial 配置模式

### 1. PGL12G/22G

Slave Serial配置接口如下所示：



图 34 Slave Serial 配置接口

Slave Serial配置接口信号描述如下表所示。

表 9 Slave Serial 配置接口信号描述

名称	I/O	专用/复用	描述
RST_N	I	专用	异步全芯片复位信号，低有效
CFG_CLK	I	复用	配置时钟，用于串行从模式时，最高频率为100MHz，串行菊花链最高频率为50MHz
MODE[2:0]	I	复用	配置模式管脚



			011: 从串行模式
INIT_FLAG_N	开漏	复用	采样MODE[2:0]管脚之前，INIT_FLAG_N是输入， 可以通过保持低电平来延迟配置。 采样MODE[2:0]管脚之后，INIT_FLAG_N是开漏， 指示配置过程是否出错。 0: 错误 1: 正确 FPGA进入用户模式后，如果回读CRC指示使能，该引脚继续做为配置管脚使用。如果回读CRC指示禁止，该引脚释放给用户使用。 <sup>(1)</sup>
CFG_DONE	开漏	专用	指示配置完成 0: FPGA未配置 1: FPGA已配置
D[1]	I	复用	串行输入数据总线，CFG_CLK上升沿采样
CSO_DOUT	O	复用	菊花链数据输出，CFG_CLK下降沿发送

注：1. 回读CRC指示使能可通过软件进行设置，指示使能不影响回读CRC的功能，用户仍可使用回读CRC，并通过状态寄存器或内部并行接口查看回读CRC结果。

要选择Slave Serial模式，建议将MODE[2]通过下拉电阻连接到地，将MODE[1:0]通过上拉电阻连接VCCIOLO。Slave Serial配置模式的应用示意图如下所示：

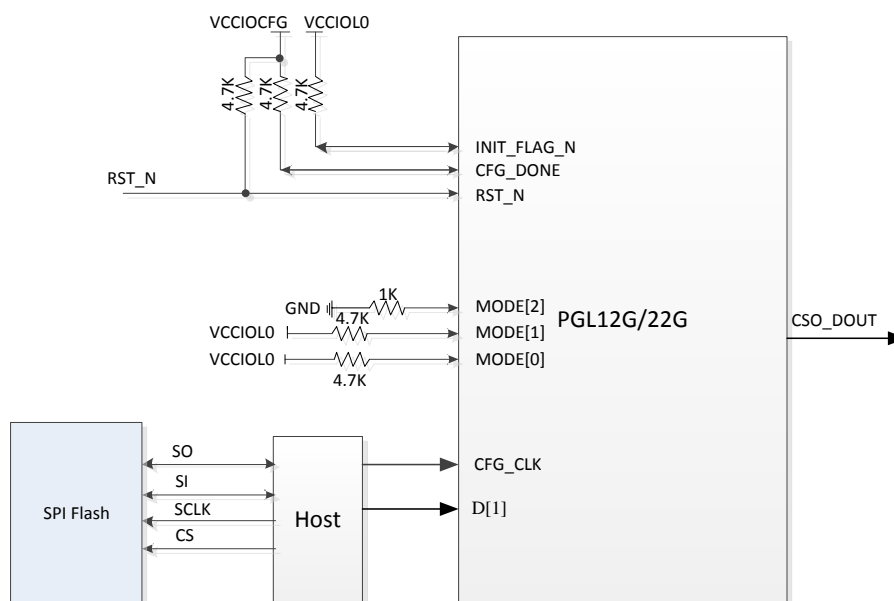


图 35 Slave Serial 模式应用示意图

在Slave Serial配置模式下，可以通过一个主控芯片（Host）控制板上多个芯片的上电启动以及数据加载。Host可以是微处理器、CPLD或者其他FPGA。该模式下可以通过上电或给RST\_N一个低电平脉冲来启动编程；通过监控管脚INIT\_FLAG\_N和CFG\_DONE来判断编程是否结束。

Host向PGL12G/22G器件发送位流时，若不使能等待PLL Lock及DCI Match（默认不使能），器件会在位流结尾的100个无操作类型1包头之间某个时钟周期释放对CFG\_DONE的控制，由于外部上拉电阻的存在，CFG\_DONE会在此时拉高。而CFG\_DONE拉高之后的位流部分用于为器件唤醒提供时钟，因此需要确保将位流发完整后才能终止时钟。若使能等待，则在CFG\_DONE拉高前不能终止时钟，并且在CFG\_DONE拉高后，需多给至少100个时钟。

Slave Serial编程模式典型时序如下图所示，

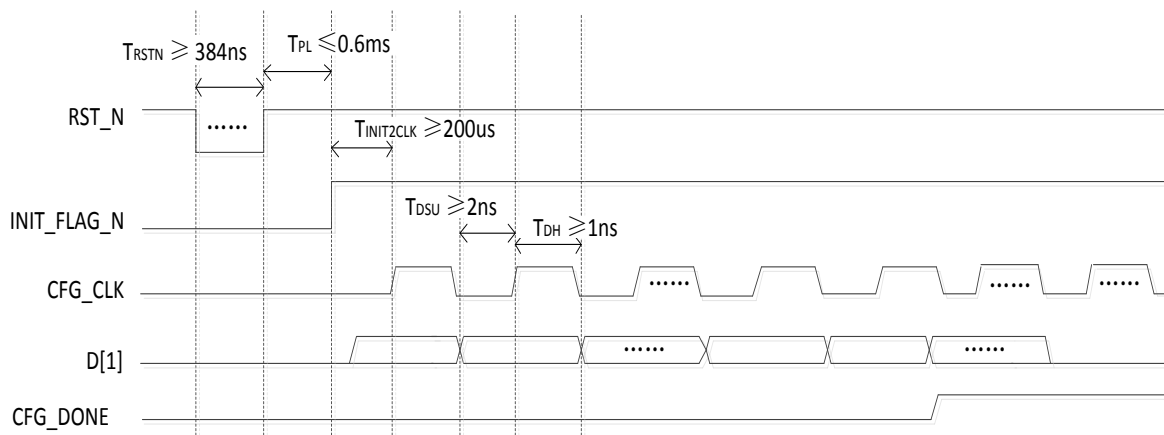


图 36 Slave Serial 编程典型时序图

## Slave Serial 串行菊花链

Slave Serial配置模式的典型应用是用于串行菊花链。所有器件可以全使用Slave Serial模式进行串行级联配置。级联配置时，从最末级器件（最远端）开始配置，最后配置第一片器件。级联配置的级联位流，需要使用Fabric Configuration工具将几个位流合成一个级联位流。

串行级联可以通过一个主控芯片（Host）控制板上多个级联芯片的配置。连接如下图所示。

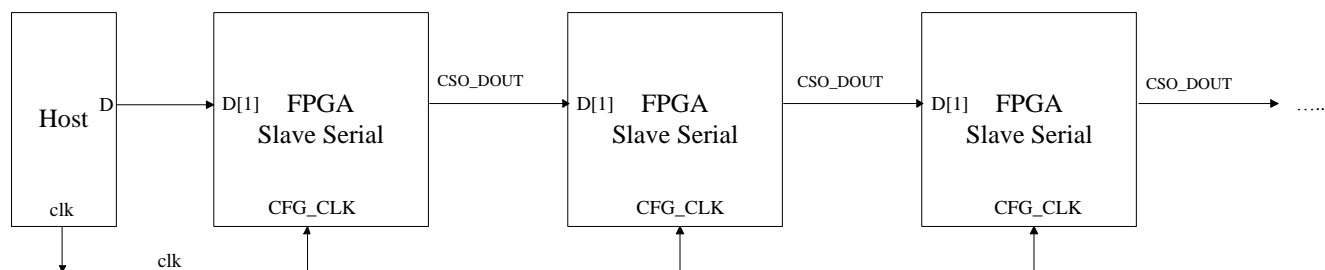


图 37 Slave Serial 配置模式串行菊花链连接

## 2. PGL25G/50G/50H

Slave Serial配置接口如下所示：



图 38 Slave Serial 配置接口

Slave Serial配置接口信号描述如下表所示。

表 10 Slave Serial 配置接口信号描述

名称	I/O	专用/复用	描述
RST_N	I	专用	异步全芯片复位信号，低有效
CFG_CLK	I	复用	配置时钟，用于串行从模式时，最高频率为100MHz，串行菊花链最高频率为50MHz
MODE[1:0]	I	复用	配置模式管脚 11: 从串行模式
INIT_FLAG_N	开漏	复用	采样MODE[1:0]管脚之前，INIT_FLAG_N是输入， 可以通过保持低电平来延迟配置。 采样MODE[1:0]管脚之后，INIT_FLAG_N是开漏， 指示配置过程是否出错。 0: 错误 1: 正确 FPGA进入用户模式后，如果回读CRC指示使能，该引脚继续做为配置管脚使用。如果回读CRC指示禁止，该引脚释放给用户使用。 <sup>(1)</sup>
CFG_DONE	开漏	专用	指示配置完成 0: FPGA未配置 1: FPGA已配置
D[0]	I	复用	串行输入数据总线，CFG_CLK上升沿采样
DOUT_BUSY	O	复用	菊花链数据输出，CFG_CLK下降沿发送

注：1. 回读CRC指示使能可通过软件进行设置，指示使能不影响回读CRC的功能，用户仍可使用回读CRC，并通过状态寄存器或内部并行接口查看回读CRC结果。

要选择Slave Serial模式，建议将MODE[1:0]通过上拉电阻连接VCCIO2。Slave Serial配置模式的应用示意图如下所示：

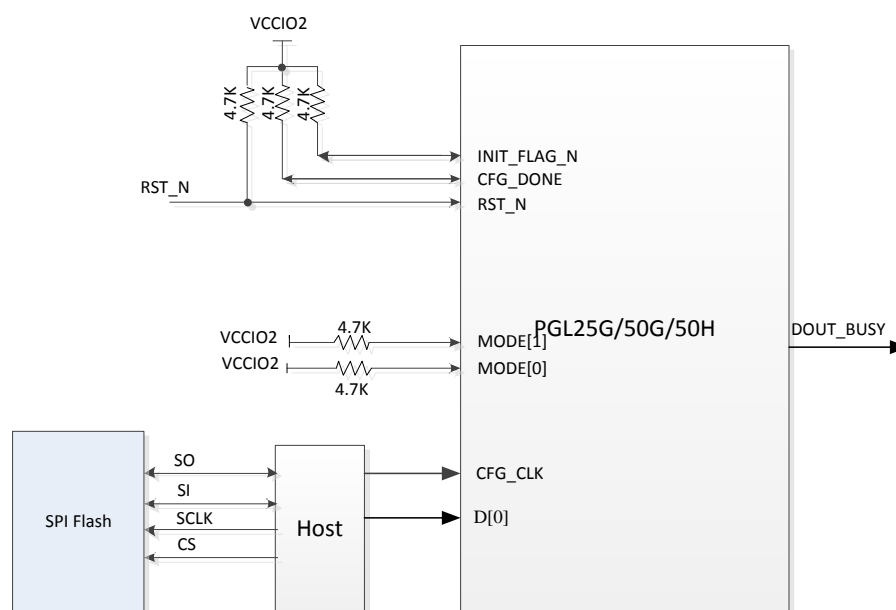


图 39 Slave Serial 模式应用示意图

在Slave Serial配置模式下，可以通过一个主控芯片（Host）控制板上多个芯片的上电启动以及数据加载。Host可以是微处理器、CPLD或者其他FPGA。该模式下可以通过上电或给RST\_N一个低电平脉冲来启动编程；通过监控管脚INIT\_FLAG\_N和CFG\_DONE来判断编程是否结束。

Host向PGL25G/50G/50H器件发送位流时，若不使能等待PLL Lock（默认不使能），器件会在位流结尾的100个无操作类型1包头之间某个时钟周期释放对CFG\_DONE的控制，由于外部上拉电阻的存在，CFG\_DONE会在此时拉高。而CFG\_DONE拉高之后的位流部分用于为器件唤醒提供时钟，因此需要确保将位流发完整后才能终止时钟。若使能等待，则在CFG\_DONE拉高前不能终止时钟，并且在CFG\_DONE拉高后，需多给至少100个时钟。

Slave Serial编程模式典型时序如下图所示，

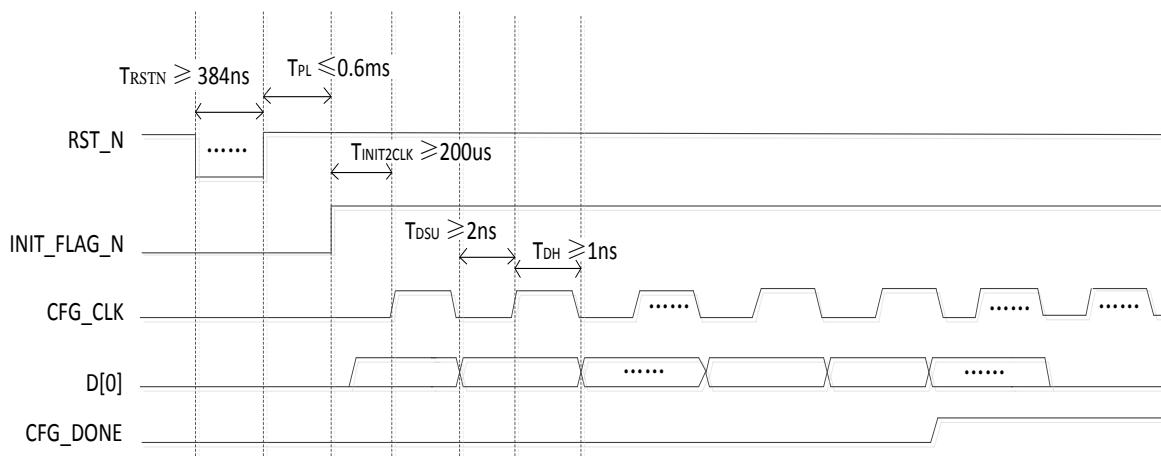


图 40 Slave Serial 编程典型时序图

## Slave Serial 串行菊花链

Slave Serial配置模式的典型应用是用于串行菊花链。所有器件可以全使用Slave Serial模式进行串行级联配置。级联配置时，从最末级器件（最远端）开始配置，最后配置第一片器件。级联配置的级联位流，需要使用Fabric Configuration工具将几个位流合成一个级联位流。

串行级联可以通过一个主控芯片（Host）控制板上多个级联芯片的配置。连接如下图所示。

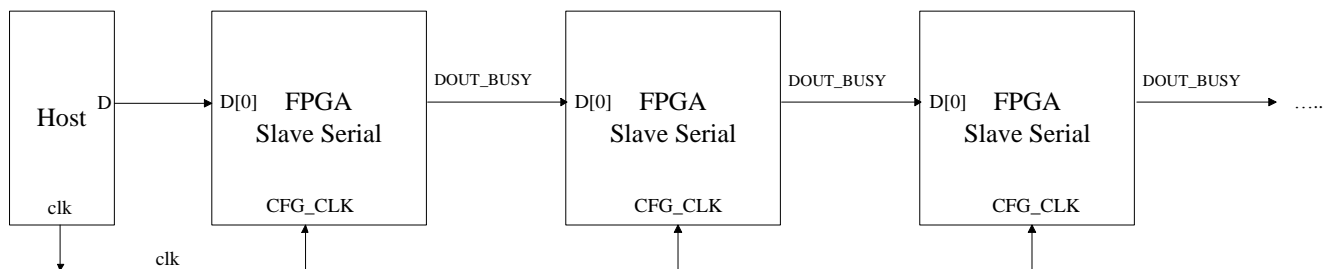


图 41 Slave Serial 配置模式串行菊花链连接

## (六) Slave Parallel 配置模式

### 1. PGL12G/22G

Slave Parallel配置接口如下所示：

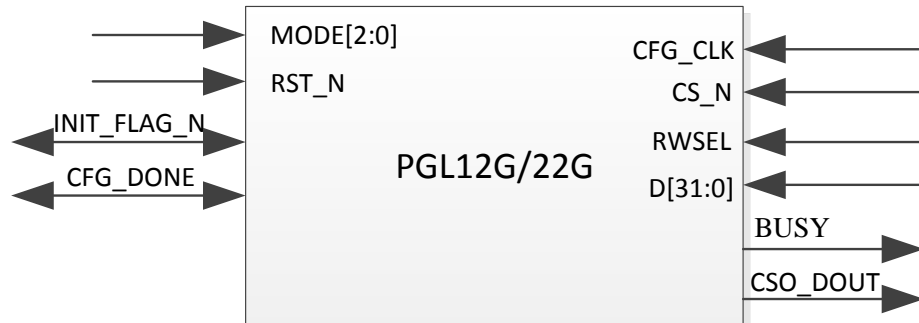


图 42 Slave Parallel 配置接口

Slave Parallel配置接口信号描述如下表所示。

表 11 Slave Parallel 配置接口信号描述

名称	I/O	专用/复用	描述
RST_N	I	专用	异步全芯片复位信号，低有效
CFG_CLK	I	复用	配置时钟，用于串行从模式时，最高频率为100MHz
MODE[2:0]	I	复用	配置模式管脚 100: 从并行模式
INIT_FLAG_N	开漏	复用	采样MODE[2:0]管脚之前，INIT_FLAG_N是输入， 可以通过保持低电平来延迟配置。 采样MODE[2:0]管脚之后，INIT_FLAG_N是开漏， 指示配置过程是否出错。 0: 错误 1: 正确 FPGA进入用户模式后，如果回读CRC指示使能，该引脚继续做为配置管脚使用。如果回读CRC指示禁止，该引脚释放给用户使用。 <sup>(1)</sup>
CFG_DONE	开漏	专用	指示配置完成 0: FPGA未配置 1: FPGA已配置
CS_N	I	复用	外部并行接口芯片选择信号，低有效。CFG_CLK上升沿采样。
RWSEL	I	复用	外部并行接口读写控制信号。CFG_CLK上升沿采样。 0: 写 1: 读
D[31:0]	IO	复用	并行数据总线，CFG_CLK上升沿采样和发送
BUSY	O	复用	指示回读数据是否准备好 0: 准备好 1: 未准备好
CSO_DOUT	O	复用	菊花链数据输出，CFG_CLK下降沿发送

注：1. 回读CRC指示使能可通过软件进行设置，指示使能不影响回读CRC的功能，用户仍可使用回读CRC，并通过状态寄存器或内部并行接口查看回读CRC结果。

Slave Parallel配置接口典型时序图如下图所示。

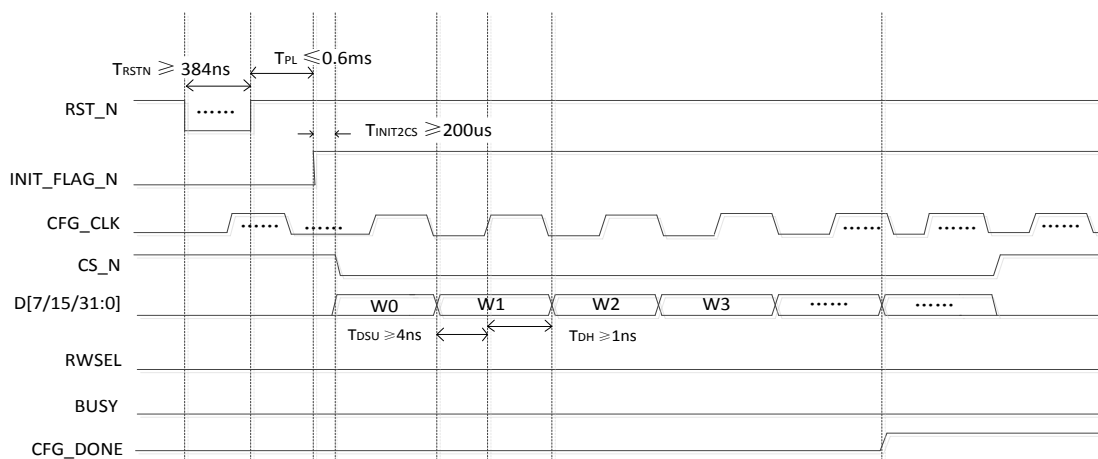


图 43 Slave Parallel 配置时序图

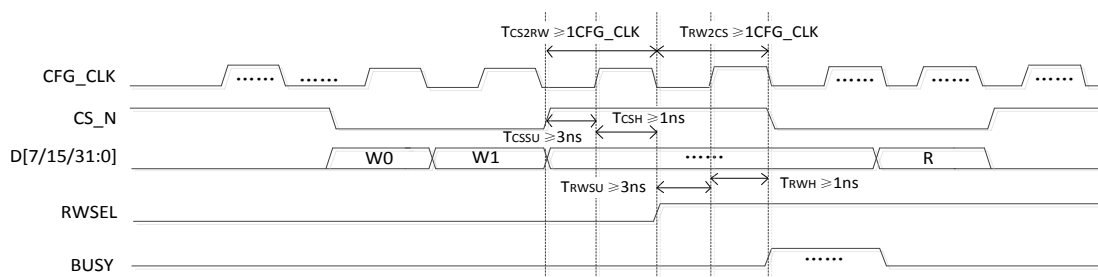


图 44 Slave Parallel 回读时序图

Slave Parallel模式下，可以通过一个主控芯片（Host）控制板上多个芯片的上电启动以及数据加载。Host可以是微处理器、CPLD或者其他FPGA。要选择Slave Parallel模式，建议将MODE[1]和MODE[0]通过下拉电阻连接到地，将MODE[2]通过上拉电阻连接到VCCIOLO电源。应用示意图如下所示：

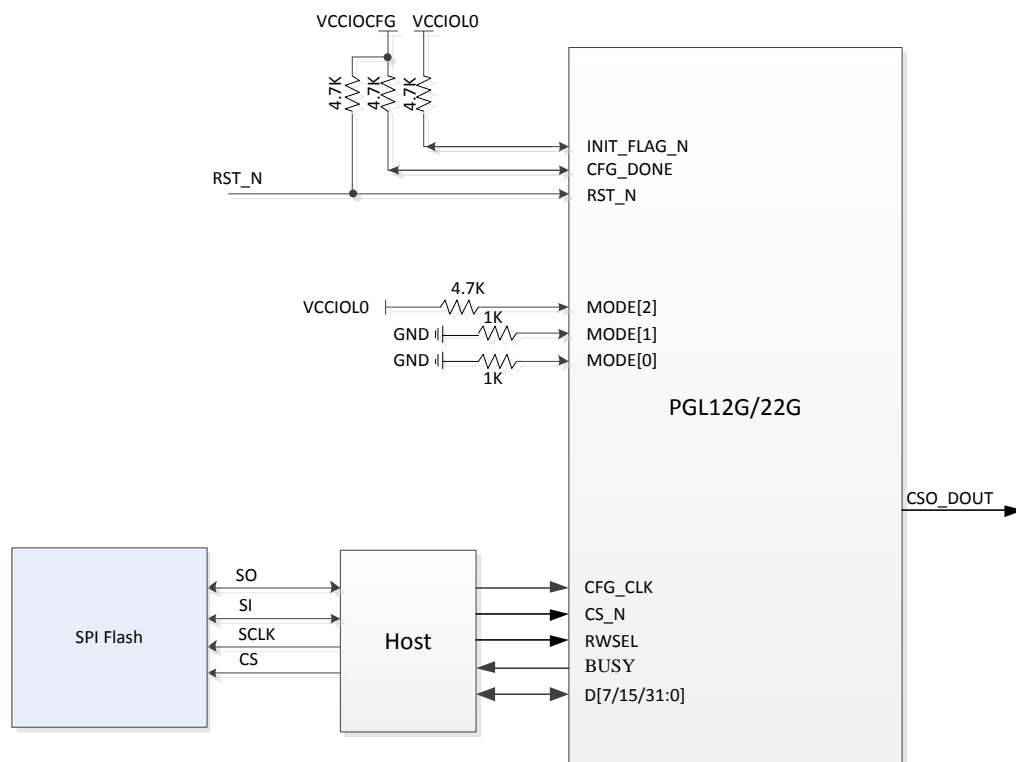


图 45 Slave Parallel 模式应用示意图

通过监控管脚INIT FLAG N和CONFIG DONE来判断编程是否结束:

Slave Parallel模式支持8/16/32bit三种数据位宽模式，可以通过对位流的解析来选择数据位宽，详细说明见[总线位宽自动检测](#)。使用Slave Parallel接口进行配置时，应注意数据总线与数据的位序对应，参考第三章中的[不同接口位宽下的位序](#)。

### Slave Parallel 并行菊花链

并行级联可以通过一个主控芯片（Host）控制板上多个级联芯片的配置。连接如下图所示。

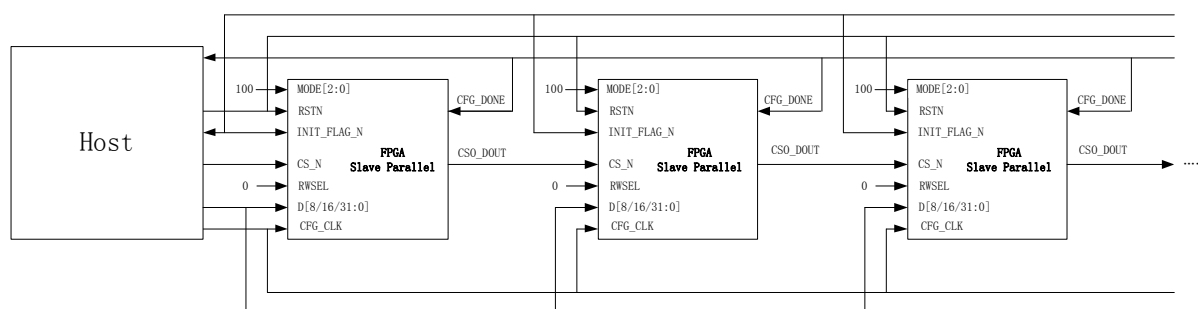


图 46 Slave Parallel 配置模式并行菊花链连接

## 2. PGL25G/50G/50H

Slave Parallel配置接口如下所示:



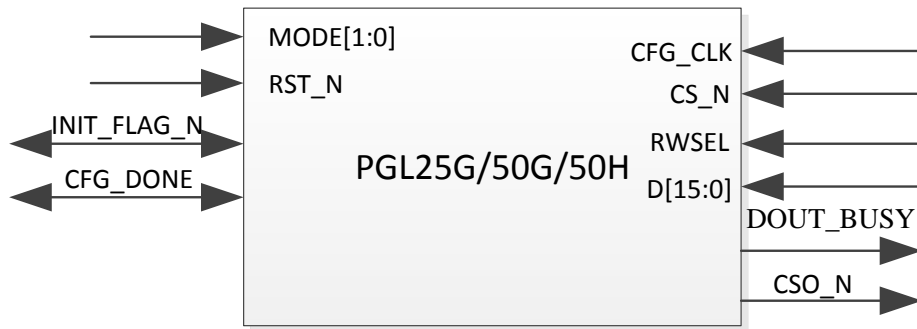


图 47 Slave Parallel 配置接口

Slave Parallel配置接口信号描述如下表所示。

表 12 Slave Parallel 配置接口信号描述

名称	I/O	专用/复用	描述
RST_N	I	专用	异步全芯片复位信号，低有效
CFG_CLK	I	复用	配置时钟，用于串行从模式时，最高频率为100M
MODE[1:0]	I	复用	配置模式管脚 10: 从并行模式
INIT_FLAG_N	开漏	复用	采样MODE[1:0]管脚之前，INIT_FLAG_N是输入， 可以通过保持低电平来延迟配置。 采样MODE[1:0]管脚之后，INIT_FLAG_N是开漏， 指示配置过程是否出错。 0: 错误 1: 正确 FPGA进入用户模式后，如果回读CRC指示使能，该引脚继续做为配置管脚使用。如果回读CRC指示禁止，该引脚释放给用户使用。 <sup>(1)</sup>
CFG_DONE	开漏	专用	指示配置完成 0: FPGA未配置 1: FPGA已配置
CS_N	I	复用	外部并行接口芯片选择信号，低有效。CFG_CLK上升沿采样。
RWSEL	I	复用	外部并行接口读写控制信号。CFG_CLK上升沿采样。 0: 写 1: 读
D[15:0]	IO	复用	并行数据总线，CFG_CLK上升沿采样和发送
DOUT_BUSY	O	复用	指示回读数据是否准备好 0: 准备好 1: 未准备好
CSO_N	O	复用	菊花链数据输出，CFG_CLK下降沿发送

注：1. 回读CRC指示使能可通过软件进行设置，指示使能不影响回读CRC的功能，用户仍可使用回读CRC，并通过状态寄存器或内部并行接口查看回读CRC结果。

Slave Parallel配置接口典型时序图如下图所示。



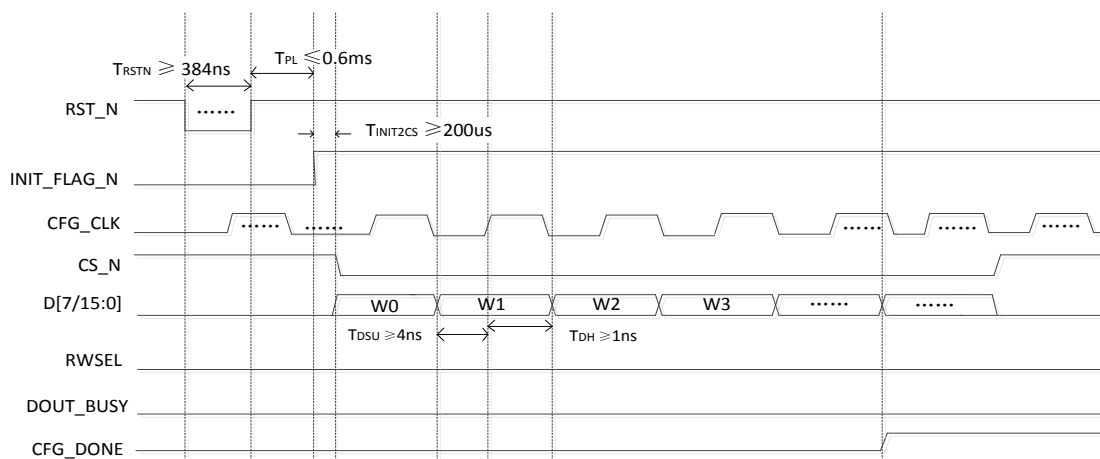


图 48 Slave Parallel 配置时序图

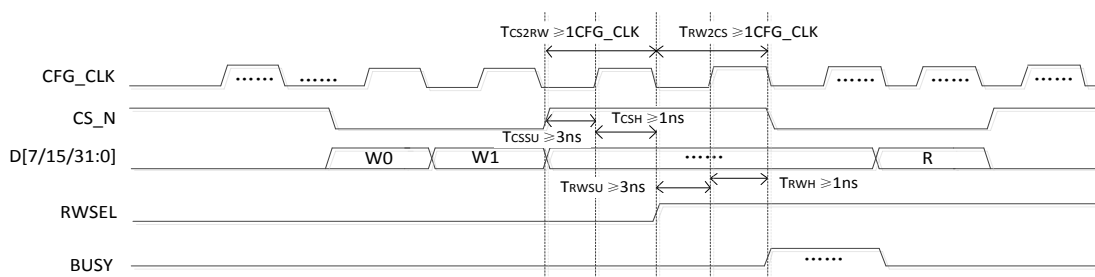


图 49 Slave Parallel 回读时序图

Slave Parallel模式下，可以通过一个主控芯片（Host）控制板上多个芯片的上电启动以及数据加载。Host可以是微处理器、CPLD或者其他FPGA。要选择Slave Parallel模式，建议将MODE[0]通过下拉电阻连接到地，将MODE[1]通过上拉电阻连接到VCCIO2电源。应用示意图如下所示：

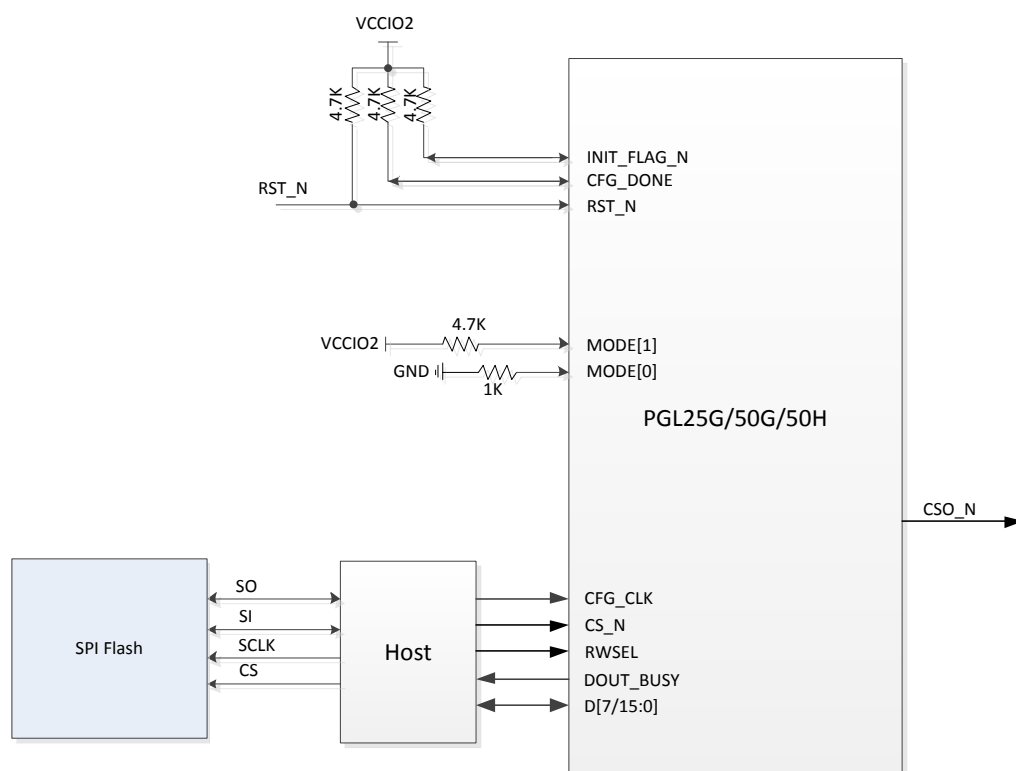


图 50 Slave Parallel 模式应用示意图

通过监控管脚INIT FLAG N和CONFIG DONE来判断编程是否结束:

Slave Parallel模式支持8/16bit三种数据位宽模式，可以通过对位流的解析来选择数据位宽，详细说明见[总线位宽自动检测](#)。使用Slave Parallel接口进行配置时，应注意数据总线与数据的位序对应，参考第三章中的[不同接口位宽下的位序](#)。

### Slave Parallel 并行菊花链

并行级联可以通过一个主控芯片（Host）控制板上多个级联芯片的配置。连接如下图所示。

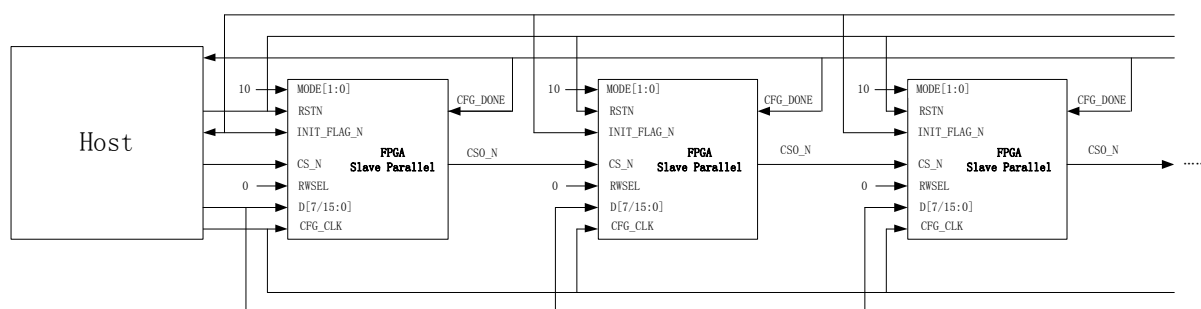


图 51 Slave Parallel 配置模式并行菊花链连接

### 三、 下载配置

#### (一) 配置过程

Logos系列FPGA的下载配置过程如下图所示：

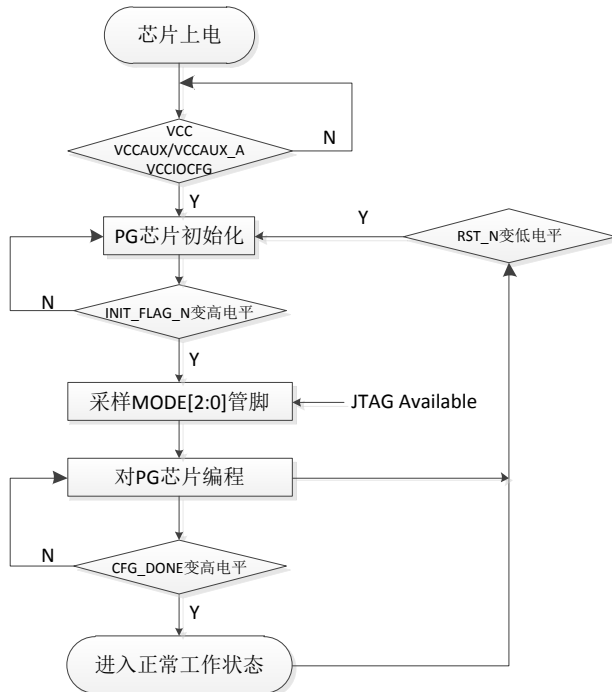


图 52 下载配置流程图

对于所有的配置接口，基本的配置步骤是相同的，包括建立，位流加载以及唤醒。

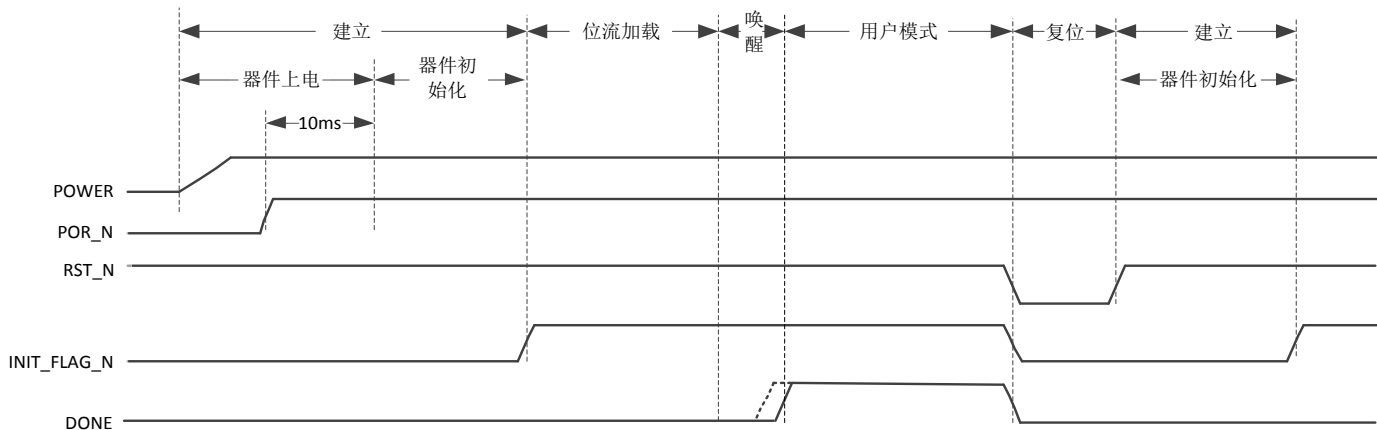


图 53 下载配置时序

Logos FPGA的配置过程包括如下几个阶段。

#### 1. 建立

建立完成FPGA器件初始化和配置模式确定的操作，包括器件上电、器件初始化和配置模式选择。

## 1) 器件上电

器件上电过程由POR\_N信号来标识，当POR\_N信号由低变高，表示配置控制系统的电源VCC已经上升到了0.9V。器件上电后，配置控制系统的上电复位电路部分开始工作，其它电路部分仍处于复位状态。上电复位电路利用器件内部晶振产生的配置控制系统主时钟CLK进行计数，计数10ms后，表示内部晶振产生的时钟已经稳定，此时配置控制系统上电复位释放，配置控制系统开始工作。

## 2) 器件初始化

器件上电，配置控制系统复位释放，配置控制系统开始工作后，首先进行器件初始化操作。

器件初始化完成两个操作：配置存储器清空2次和读eFuse内容。这两个操作同时进行，配置存储器清空2次操作的完成时间长于读eFuse内容的完成时间。

进入器件初始化过程后，配置控制系统从配置存储器0地址开始逐帧清空配置存储器，清空配置存储器完成后，再次从配置存储器0地址开始逐帧清空配置存储器，第2次清空配置存储器完成后，初始化完成标志信号init\_complete变高。

可以通过维持外部的INIT\_FLAG\_N管脚为低电平，来维持配置控制系统继续处于器件初始化过程。

当硬复位，JTAG指令复位，热启动，版本回退中的一种功能触发时，配置控制系统进入器件初始化过程。

器件初始化期间，除专用配置管脚外，其它IO都处于高阻状态。

## 3) 配置模式选择

当INIT\_FLAG\_N信号为高后，采样配置模式管脚MODE[2:0]，选择器件的配置模式。模式采样完成后，CCS选择复用管脚的配置功能。

## 2. 位流加载

不同配置模式具有不同配置接口。配置模式选择完成后，相应的模式所对应的管脚被设置为配置管脚。本阶段包括总线位宽检查、同步、器件标志检查、加载前选项设置、数据加载、CRC、加载后选项设置等操作。

### 1) 总线位宽自动检测

Slave Parallel配置模式时，数据总线位宽由CCS自动进行检测，通过检测由两个字组成的总线位

宽自动检测序列000000AA 08100020，进行位宽匹配。

总线位宽自动检测序列在位流的开头部分，在同步字之前。CCS只检测并行数据总线的低8位。CCS进行检测时，先在数据总线低八位检测0xAA字节，检测到0xAA后，继续检测下一个字节是否是0x08(x8)，0x10 (x16)或0x20(x32)。如果下一个字节不是上述三个中的一个，CCS会重新检测0xAA，直到检测到0xAA + 0x08/0x10/0x20序列。之后，CCS设置相关的数据管脚为配置数据引脚，数据总线切换到检测到的位宽，开始寻找同步字。

#### 8位总线

FF	填充字
FF	
FF	
FF	
00	总线位宽自动检测序列
00	
00	
AA	
08	
10	
00	
20	
FF	填充字
FF	
FF	
FF	
...	

#### 16位总线

FFFF	填充字
FFFF	
0000	总线位宽自动检测序列
00AA	
0810	
0020	
FFFF	
FFFF	填充字
FFFF	
...	

#### 32位总线

FFFFFFFF	填充字
000000AA	总线位宽自动检测序列
08100020	
FFFFFFFF	填充字
...	

## 2) 同步

同步字0x01332D94用于32位字边界对齐。同步后，才能进行后续解帧操作。

当接收到正确的同步字后，CCS才认为数据是有效数据；CCS的包处理器才进行解包操作；除了总线位宽自动检测序列，其它任何在同步字之前的数据都将被忽略。

## 3) 主模式读操作选择

Master SPI/Master BPI配置模式时，对读操作进行选择。

Master SPI配置模式时，选择完成后，器件向SPI Flash重新发送读命令，并根据读命令重新调整数据总线位宽。

Master BPI配置模式时，选择完成后，如果异步读，则重新启动异步读时序。如果同步读，先启动写BPI Flash读配置寄存器时序，将BPI Flash配置为同步读模式，然后启动同步读时序。

## 4) 复位 CRC

复位CRC寄存器。

## 5) 器件标识检查

检查位流中的器件标识是否与硬件匹配，如果不匹配，init\_complete信号变低，ID错误标志存入状态寄存器，跳出配置过程。

## 6) 加载前选项设置

加载配置数据前，在配置寄存器中指定器件的工作状态：

看门狗设置

解密初始向量设置

密钥选择

解密选择

设置是否保留复用的配置端口在配置完成后继续作为配置端口使用

唤醒时钟选择

唤醒时序设置

唤醒时是否等待PLL和DCI

版本回退设置

主模式频率设定

## 7) 加载配置数据

将配置数据帧写入配置存储器。

## 8) 加载后 CRC 检查

配置数据加载完成后,进行CRC校验。软件生成的CRC值存放在CRC寄存器中,与CCS计算的CRC值进行比较。如果两个值一致,表示CRC检查通过,所有配置信息正确写入了配置存储器中。如果两个值不一致,表示CRC检查不同过,那么INIT\_N信号变为低,CRC错误标志存入状态寄存器。

## 9) 加载后选项设置

配置数据加载完成后,在配置寄存器中指定器件的工作状态:

设置是否允许用户逻辑关断OSC

外部端口安全级别设置

## 3. 唤醒

位流加载完毕并通过CRC后,FPGA进入唤醒阶段。配置系统先使能FPGA内部所有功能模块的逻辑输出。唤醒前再次CRC校验,然后进行去同步操作,表示配置操作结束,后续操作需要重新同步。最后完成唤醒,逐步释放相应的全局信号。

### 1) 全局逻辑使能

加载后CRC检查通过后,使能fabric内部所有功能模块的逻辑输出。

### 2) 唤醒启动

全局逻辑使能后,启动唤醒操作。

### 3) 唤醒前 CRC 检查

唤醒操作执行前,进行CRC校验。软件生成的CRC值存放在CRC寄存器中,与CCS计算的CRC值进行比较。如果两个值一致,表示CRC检查通过,唤醒前准备工作完成。如果两个值不一致,表示CRC检查不同过,那么INIT\_N信号变为低,CRC错误标志存入状态寄存器。

#### 4) 去同步

表示配置过程结束。后续操作需要重新同步。

#### 5) 唤醒

去同步完成后，唤醒电路按照设定的时序开始工作，逐步释放相应的全局信号。

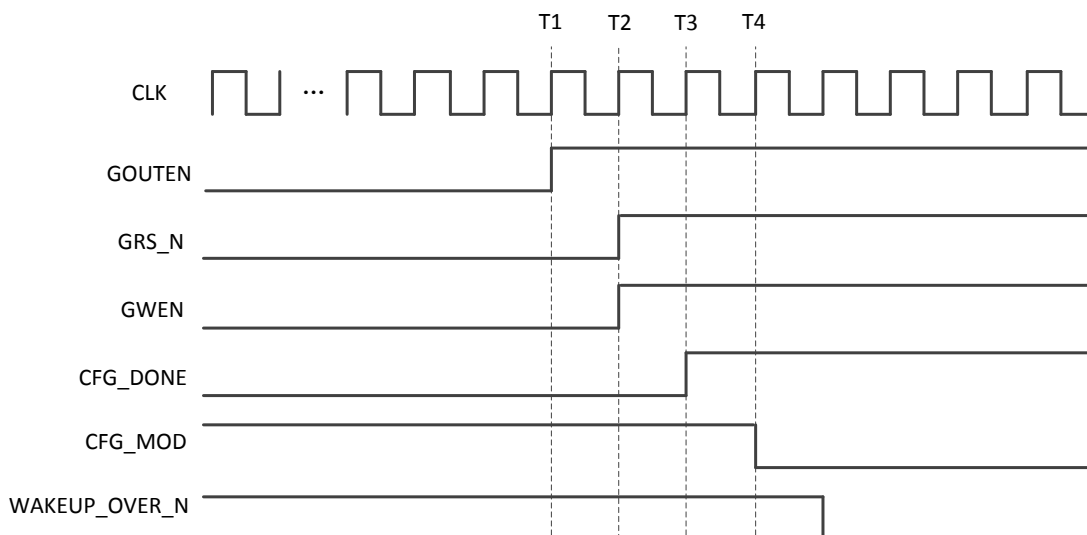


图 54 唤醒时序

全局逻辑使能信号GLOGEN输出到fabric，使能各逻辑块；在最远端反馈回CCS。CCS接收到反馈信号GLOGEN\_FB后，进入唤醒时序。在GLOGEN使能前，全芯片各模块的寄存器输出值均为1，使能后各模块寄存器由GRS\_N控制。

在配置完成之前，全局寄存器置位复位信号GRS\_N为低电平，控制全芯片各模块的GRS\_N端口，使各模块的寄存器处于置位复位状态，用户逻辑中的寄存器处于置位或复位状态，具体取决于用户设计以及软件综合结果；配置完成后，全芯片各模块的GRS\_N信号释放，使各模块的寄存器进入非全局置位复位状态。

在配置完成之前，全局IO输出使能信号GOUTEN为低电平，所有IO的输出处于高阻状态。在配置完成后，使能所有IO的输出。

表 13 GOUTEN 逻辑功能

GOUTEN	0	1
用户使用的 IO	High-Z	用户控制
用户未使用的 IO	High-Z	默认配置点控制 (High-Z)

ISC编程时，GOUTEN不再受唤醒时序控制，而是交由ISC编程控制。在ISC编程过程中或执行ISC指令时，所有IO的输出处于高阻状态。

全局写使能信号GWEN是FPGA芯片内部存储资源的全局写使能。在配置完成之前，该信号为低电平，禁止FPGA存储资源的写操作。

复用管脚控制信号CFGMOD将配置过程中使用的复用管脚释放给用户使用。



唤醒完成指示信号WAKEUP\_OVER\_N指示唤醒过程成功结束。

全局唤醒信号GOUTEN，GRS\_N，GWEN和CFG\_DONE置1的时刻用户可配，可在时刻T1，T2，T3或T4置1。

两个时刻之间的周期数用户可配。

唤醒结束后，FPGA由配置模式进入用户模式。

如果用户需要重新进行位流加载，可以把RST\_N拉低一段时间后释放，FPGA会重复进行上电过程。

## （二）配置文件

### 1. 位流生成

PDS设计软件会产生不同扩展名的配置文件来适应不同的配置方案，具体描述见下表。

表 14 PDS 配置文件描述

配置文件扩展名	描述
.sbit	包含头信息（位流名称、日期等）的二进制配置数据，配置工具会识别头信息，但不会把头信息写入 FPGA。用 Fabric Configuration 工具通过 Cable 可直接把.sbit 文件写入 FPGA
.bin	无头信息的二进制配置数据（纯位流），适用于用户配置方案，如微处理器对 FPGA 进行配置。
.sfc	写入 Flash 的位流文件，由.sbit 文件转换而来

.sbit文件是PDS工具编译完成后生成的默认位流文件，如下图所示，运行Generate Bitstream即可生成.sbit文件。

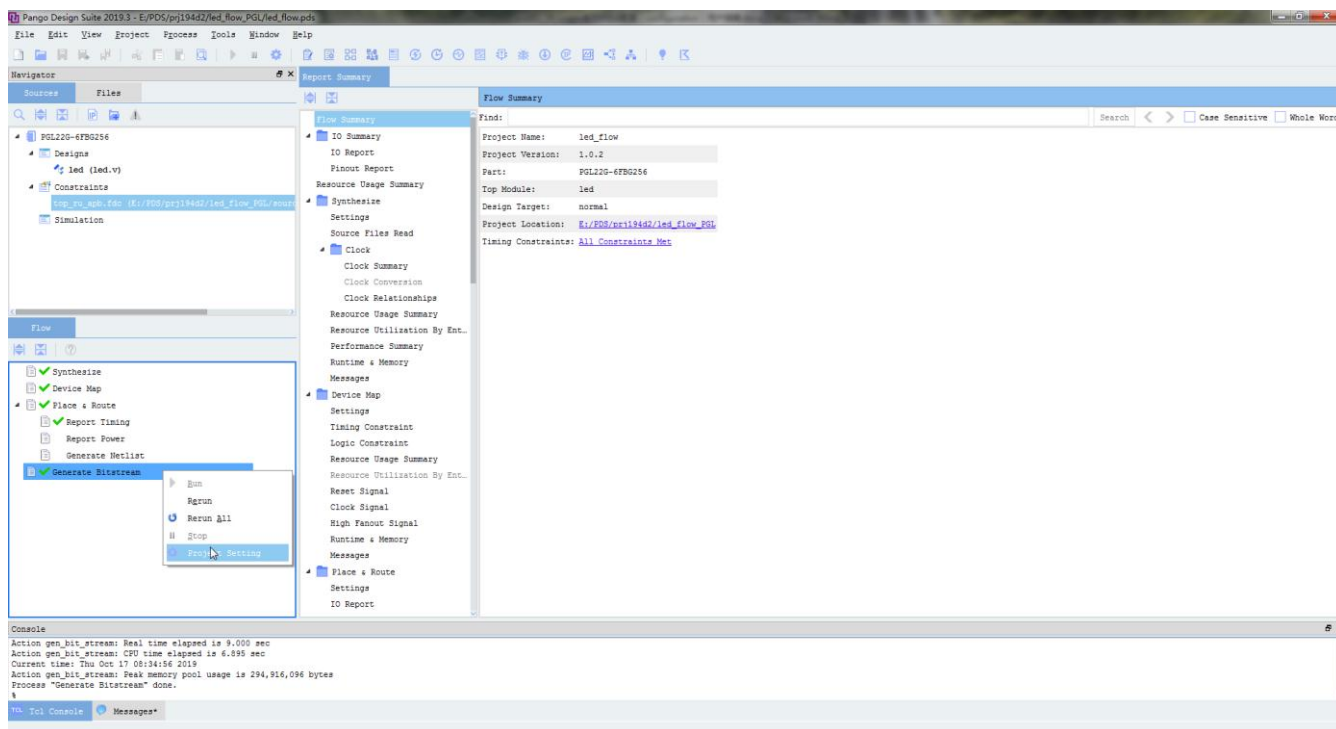


图 55 生成位流示意图

上图中鼠标右击Generate Bitstream，选择configure，弹出如下图所示，勾选Create Bin File，点击

ok后，重新运行Generate Bitstream，则同时生成.sbit和.bin文件。

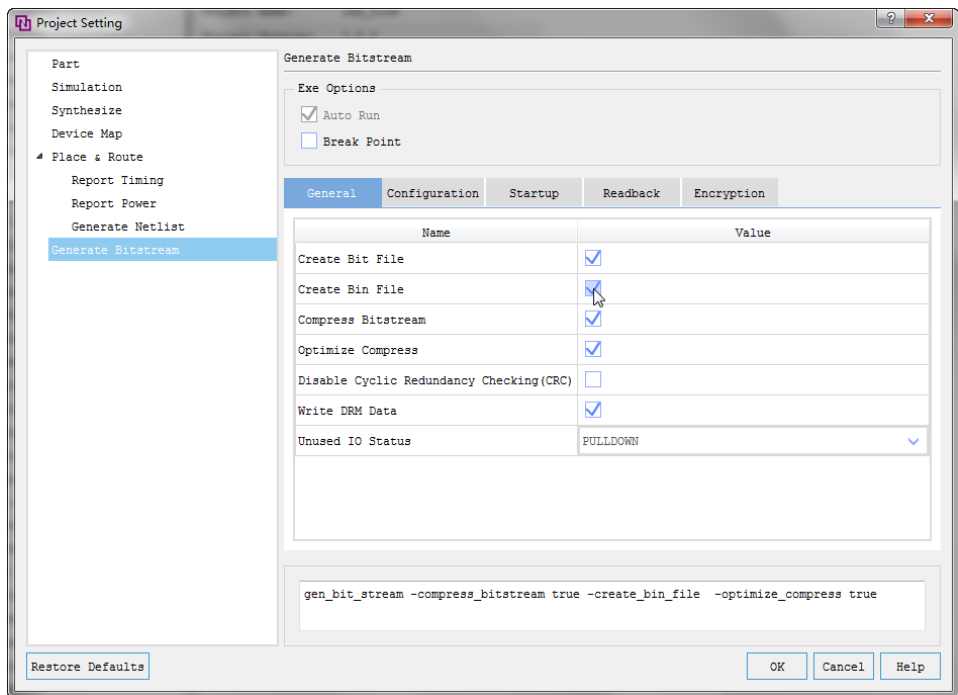


图 56 Configure Generate Bitstream

在Fabric Configuration工具界面，点击菜单栏的Operations，单击下拉菜单中的Generate Flash Programming File，弹出如下图所示界面。

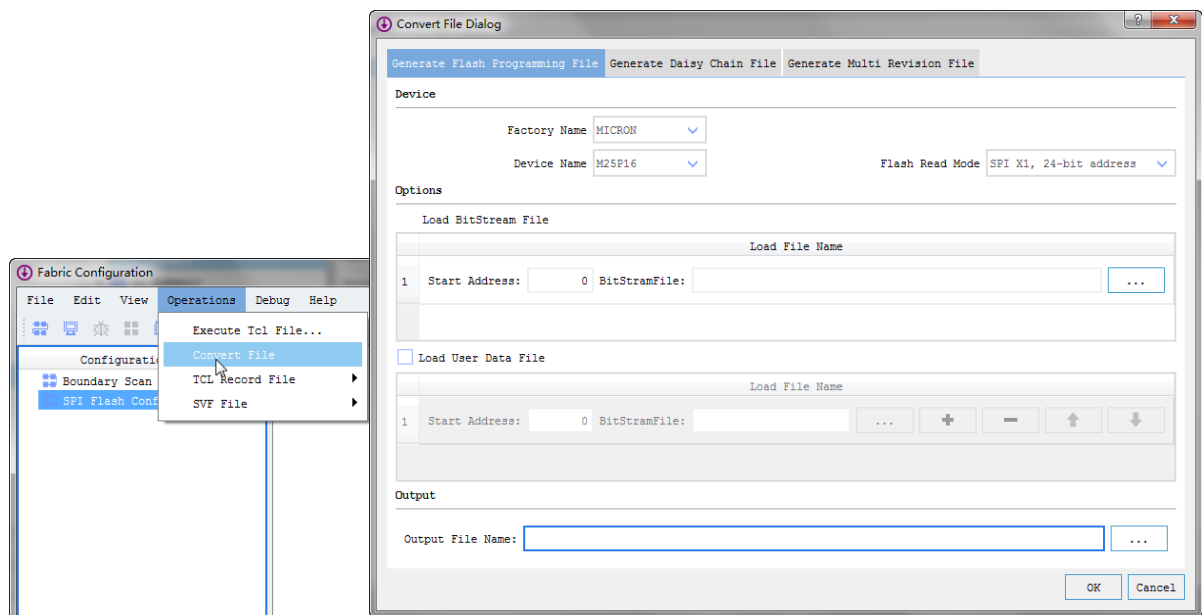


图 57 Generate Flash Programing File

用户根据实际应用情况，选择对应的Flash Device，在Flash Read Mode下拉框中选择Flash数据位宽和地址位宽，然后填入sbit文件路径和sfc文件存放路径，点击ok即可生成Flash配置文件—sfc文件。

## 2. 位流大小

Logos系列各器件的位流大小如下表所示。压缩后的位流文件比压缩前更小，压缩文件的大小跟设计有关，这里不做说明。

表 15 Logos 系列各器件的位流大小

器件	文件名	非压缩文件大小 (Kbyte)
PGL12G	*.sbit	518
PGL22G	*.sbit	745
PGL22GS	*.sbit	745
PGL25G	*.sbit	982
PGL50G	*.sbit	2052
PGL50H	*.sbit	2052

### 3. 不同接口位宽下的位序

Logos系列FPGA配置支持多种位宽，同一位宽的不同配置模式，其位序，即数据总线的高低位与位流顺序的对应关系一致。如Master SPI (x8) 与Master BPI (x8) 模式的位序一样。以位流中的同步字 (32'h01332D94) 为例，数据总线的高低位与位流顺序如下表所示。

表 16 不同位宽下同步字发送顺序

时钟周期	进制	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D[1:0](x2)	2'b	00	00	00	01	00	11	00	11	00	10	11	01	10	01	01	00
D[3:0](x4)	4'h	0	1	3	3	2	D	9	4								
D[7:0](x8)	8'h	01	33	2D	94												
D[15:0](x16)	16'h	0133	2D94														
D[31:0](x32)	32'h	01332D94															

注：表中 D[x:0]仅表示数据总线高低位，具体对应管脚需参考不同接口模式下的端口列表。

### (三) 配置数据包

#### 1. 包格式

Logos系列FPGA配置位流的包有两种类型：类型1和类型2。

类型1包用于寄存器读写。类型1包由包头部分和数据部分组成。

类型1包的包头是一个32位字。包头后面跟数据。数据以32位字为单位。

如果包头部分指定的包头后面跟的32位字的数量为0，那么这个包只有包头部分没有数据部分，是空包。

类型1包头格式如下表所示。

表 17 类型 1 包头格式

[31:29]	[28:27]	[26:22]	[21:0]
包头类型	操作码	寄存器地址	32位字个数
101	00:无操作；01:写；10:读；11:保留	寄存器的地址	包头后面跟的32位字的数量

类型2包用于读写配置存储器。类型2包由包头部分和数据部分组成。

类型2包的包头是一个32位字。包头后面跟数据。数据以32位字为单位。

类型2包必须跟在类型1空包之后。类型2包不指定寄存器地址，用前面的类型1空包指定的地址。

类型2包头格式如下表所示。

表 18 类型 2 包头格式

[31:29]	[28:27]	[26:0]
包头类型	操作码	32位字个数
010	00:无操作；01:写；10:读；11:保留	包头后面跟的32位字的数量

## 2. 配置寄存器

Logos系列FPGA的配置寄存器如下表所列。

表 19 配置寄存器说明

名称	R/W	地址	描述
CRCR	R/W	00000	CRC寄存器
IDR	R/W	00001	器件标识寄存器
CMDR	R/W	00010	命令寄存器
CTRL0R	R/W	00011	控制寄存器0
CTRL1R	R/W	00100	控制寄存器1
CMEMIR	W	00101	帧数据输入寄存器
MFWRITER	W	00110	多帧写寄存器
CMEMOR	R	00111	帧数据输出寄存器
IVR	W	01000	解密初始向量寄存器
STATUSR	R	01001	状态寄存器
CHAINR	W	01010	级联寄存器
ADRR	R/W	01011	帧地址寄存器
SBPIR	R/W	01100	SBPI寄存器
SEUR	R/W	01101	SEU控制寄存器
SEUSTATUSR	R	01110	SEU状态寄存器
IRSTCTRLR	R/W	01111	热启动控制寄存器
IRSTADDR	R/W	10000	热启动地址寄存器
WATCHDOGR	R/W	10001	看门狗寄存器
HSTATUSR	R	10010	历史状态寄存器
FADR0R	R/W	10011	位流0地址寄存器
FADR1R	R/W	10100	位流1地址寄存器
FADR2R	R/W	10101	位流2地址寄存器
FADR3R	R/W	10110	位流3地址寄存器
CMASKR	R/W	10111	控制屏蔽寄存器
FALLBACKR	R/W	11000	版本回退寄存器
OPTION0R	R/W	11001	选项寄存器0
OPTION1R	R/W	11010	选项寄存器1
RCRR	R/W	11011	RCR控制寄存器

各配置寄存器的详细描述如下。

### 1) CRC 寄存器（CRCR）

每写一次CRC寄存器，对应一次位流的CRC检查。如果写入CRC寄存器的值与CCS计算出的当前

CRC值相同，则CRC检查通过。否则init\_complete拉低。

## 2) 器件标识寄存器 (IDR)

载入位流之前要向器件标识寄存器写入IDCODE。写入的IDCODE与器件IDCODE匹配，后续操作才能继续进行。每写一次ID寄存器，对应一次器件ID检查。如果写入ID寄存器低28位的值与器件的ID低28位的值相同，则ID检查通过。否则init\_complete拉低。

表 20 器件标识寄存器说明

位	名称	描述
[31:28]	version	版本
[27:12]	part number	器件型号
[11:1]	manufacturer identity	厂商标识
[0]	1	常数1

器件型号如下表所示。

表 21 Logos 系列 FPGA 器件型号

器件型号	器件标识
PGL12G	X0501899
PGL22G	X0303899
PGL22GS	X0303899
PGL25G	X0511899
PGL50G	X0521899
PGL50H	X0521899

## 3) 命令寄存器 (CMDR)

命令寄存器决定CCS接下来进行的操作。每次操作最多向命令寄存器中写一条命令。不能一个包头后跟多条命令。

表 22 CCS 命令及其描述

命令	编码	描述
NOP	00000	无操作
RSTCRC	00001	复位CRC 复位CRC寄存器
SWITCH	00010	切换主模式配置时钟mclk频率 启动更新主模式配置时钟mclk频率操作，数值由CTRL0R中的oscfsel决定
SWITCHCLK	00011	切换系统时钟clk频率 启动更新系统时钟clk频率操作，数值由CTRL0R中的oscfsel_clk决定
WCMEM	00100	写配置数据 用于经CMEMIR写入配置存储器配置数据之前
MFWRITE	00101	多帧写 将当前写入的一帧数据写入到接下来的连续帧中。帧的数量由MFWRITER决定
RCMEM	00110	读配置数据 用于经CMEMOR回读配置存储器配置数据之前
SWAKEUP	00111	开始唤醒操作

命令	编码	描述
SWAKEDOWN	01000	关闭唤醒操作
GUP	01001	内部逻辑使能
GDOWN	01010	内部逻辑不使能
DESYNC	01011	去同步 用于配置结束时
RWD	01100	重起看门狗
RRBCRC	01101	复位回读CRC
RBCRC	01110	回读CRC检查
IRST	01111	热启动 版本回退时，热启动无效

#### 4) 控制寄存器 0 (CTRL0R)

表 23 控制寄存器 0 说明

位	名称	初始值	描述
[31:5]	保留		
[4]	spersist	1'b0	用户模式下从SPI接口保留选择 0: 释放给用户使用 1: 当用户模式下外部并行接口释放给用户使用时，从SPI接口保留继续用于配置端口。当用户模式下外部并行接口保留继续用于配置端口时，从SPI接口释放给用户使用
[3]	保留		
[2]	tkey_en	1'b0	临时密钥使能 0: 使用密钥 1: 使用临时密钥
[1]	persist	1'b0	用户模式下外部并行接口保留选择 0: 释放给用户使用 1: 保留继续用于配置端口
[0]	dec_en	1'b0	解密使能

#### 5) 控制寄存器 1 (CTRL1R)

表 24 控制寄存器 1 说明

位	名称	初始值	描述
[31:4]	保留		
[3]	mfg_por_off	1'b0	0: 正常监控VCC、VDDM（内部LDO输出电压）、VCCIOCFG、VCCAUX电压 1: 需要测试SRAM retention电压时，当gwen为1时，关闭VCC和VDDM的细探测功能。此时，VCC<0.65V或VDDM<0.5V时，POR才会重新assert。
[2]	osc_off	1'b0	控制是否允许用户逻辑关断OSC 0: 不允许 1: 允许
[1:0]	wrctrl	2'b00	外部端口安全级别。控制关断重配，部分重配，局部动态重配，回读配置存储器。配置为禁止后，不能再更改为使能，除非复位 00: 重配使能，回读使能（默认）

位	名称	初始值	描述
			01: 重配使能, 回读禁止 1x: 重配禁止, 回读禁止

## 6) 控制屏蔽寄存器 (CMASKR)

控制屏蔽寄存器用来屏蔽控制寄存器0和控制寄存器1中的对应位, 默认值为32'd0。0为屏蔽。例如控制屏蔽寄存器的值为32'h0000\_0003时, 控制寄存器0和控制寄存器1的低2位能够写入数据, 其它位不能写入数据。

## 7) 选项寄存器 0 (OPTION0R)

表 25 选项寄存器 0 说明

位	名称	初始值	描述
[31:24]	保留		
[23]	done_syn	1'b0	外部done同步使能。指示外部输入done是否是输出done的同步信号。如果是同步信号, CCS的唤醒模块直接使用。如果是异步信号, 使用前需要同步, 通常用于延迟done功能。 0:不同步 1: 同步
[22:21]	t_sel	2'd0	唤醒周期长度选择, 即T1, T2, T3之间的时间间隔 2'b00: 1 2'b01: 2 2'b10: 4 2'b11: 1
[20:19]	done_sel	2'd0	done信号拉高时刻选择 2'd0: T3 2'd1: T1 2'd2: T2 2'd3: T4
[18:17]	gwen_sel	2'd0	gwen信号拉高时刻选择 2'd0: T2 2'd1: T1 2'd2: T3 2'd3: T4
[16:15]	grsn_sel	2'd0	grsn信号拉高时刻选择 2'd0: T2 2'd1: T1 2'd2: T3 2'd3: T4
[14:13]	gouten_sel	2'd0	gouten信号拉高时刻选择 2'd0: T1 2'd1: T2 2'd2: T3 2'd3: T4

位	名称	初始值	描述
[12]	wait_dci	1'b0	唤醒时等待DCI匹配使能 0: 不需要等待DCI匹配 1: 需要等待DCI匹配
[11]	wait_pll	1'b0	唤醒时等待PLL锁定使能 0: 不需要等待PLL锁定 1: 需要等待PLL锁定
[10:9]	startup_sel	2'b00	唤醒时钟选择 00: MCLK 01: SCLK 10: TCK 11: UCLK 唤醒时钟只能选择配置时使用的时钟或用户时钟
[8:6]	oscfel_clk	3'b010	系统时钟CLK频率选择 3'd2:100M 3'd3:66.67M 3'd4:50M 3'd5:40M 3'd6:33.33M 3'd0:25M
[5:0]	oscfel	6'd0	主模式配置时钟MCLK频率选择 6'd2:100M 6'd3:66.67M 6'd4:50M 6'd5:40M 6'd6:33.33M 6'd8:25M 6'd10:20M 6'd16:12.5M 6'd20:10M 6'd25:8M 6'd32:6.25M 6'd40:5M 6'd50:4M 6'd0:3.125M

## 8) 选项寄存器 1 (OPTION1R)

表 26 选择寄存器 1

位	名称	初始值	描述
[31:2]	保留		
[1]	rbcrc_disable	1'b0	禁止回读CRC错误指示 0: 使能 1: 禁止
[0]	crc_disable	1'b0	禁止CRC检查 0: 使能 1: 禁止



## 9) 帧数据输入寄存器 (CMEMIR)

帧数据输入寄存器是配置存储器的配置数据接口。数据经CMEMIR写入配置存储器。

## 10) 多帧写寄存器 (MFWRITER)

压缩的帧数目，表示后面连续几帧的数据内容与刚写完的一帧数据内容相同。比如配置完一帧数据后，如果后面连续三帧的数据内容与这一帧相同，那么多帧写寄存器内容需要写为3。

## 11) 帧数据输出寄存器 (CMEMOR)

帧数据输出寄存器是配置存储器的回读数据接口。数据经CMEMOR从配置存储器回读出来。

## 12) 解密初始向量寄存器 (IVR)

128位初始向量，用于解密。

## 13) 状态寄存器 (STATUSR)

状态寄存器说明如下所示：

表 27 状态寄存器说明

位	名称	描述
[31:30]	保留	
[29]	over_temp	过温度标志
[28]	flg_x32	并行从模式32位数据位宽指示
[27]	flg_x16	并行从模式16位数据位宽指示
[26]	flg_x8	并行从模式8位数据位宽指示
[25:24]	ipal_m[1:0]	ipal数据位宽选择
[23]	key_lock	密钥锁定标志
[22]	fallback	fallback指示标志
[21]	dci_match	DCI匹配
[20]	pll_lock	PLL锁定
[19]	gwen	gwen
[18]	grsn	grsn
[17]	gouten	gouten
[16]	vddt_n	vddt_n
[15]	glogen_fb	glogen反馈
[14]	glogen	glogen
[13]	done_i	外部done
[12]	done	done
[11]	init_b	init_b
[10]	init_complete	配置完成和错误指示

位	名称	描述
[9:7]	m[2:0]	模式选择
[6]	wakedown_over	唤醒关断结束
[5]	wakeup_over	唤醒结束
[4]	overflow	FIFO溢出
[3]	timeout	Watchdog timeout
[2]	rbcrc_err	回读CRC检测结果 0: CRC正确 1: CRC错误
[1]	crc_err	CRC检测结果 0: CRC正确 1: CRC错误
[0]	id_err	ID检测结果 0: 正确 1: 错误

#### 14) 看门狗寄存器 (WATCHDOGR)

看门狗寄存器用于配置模式和用户模式下的超时检测。其说明如下所示。

表 28 看门狗寄存器说明

位	名称	初始值	描述
[31]	wd_user_en	1'b0	用户模式下看门狗使能
[30]	wd_cfg_en	1'b0	配置模式下看门狗使能
[29:0]	wd_value	30'h3FFF_FFFF	看门狗超时数值 50M时钟计数wd_value*4个周期后，看门狗超时。

#### 15) 帧地址寄存器 (ADRR)

帧地址寄存器提供配置和回读数据的起始地址。只能发有效地址，不能超范围。

表 29 帧地址寄存器说明

位	名称	初始值	描述
[31: 27]	保留		
[26: 25]	type	2'd0	配置存储器内容类型 00: 非DRM内容 01: DRM内容 10: 保留 11: 保留
[24: 20]	addr_region	5'd0	区域地址
[19: 18]	保留		
[17: 10]	addr_column	8'd0	列地址
[9: 8]	保留		
[7: 0]	addr_frame	8'd0	帧地址

#### 16) SBPI 寄存器 (SBPIR)

SBPI寄存器提供Master SPI模式，内部Master SPI模式，Master BPI模式控制和选项信息。

表 30 SBPI 寄存器说明

位	名称	初始值	描述
[31]	emclk_en	1'b0	外部主时钟使能 1'b0: 主模式配置时钟使用内部时钟 1'b1: 主模式时钟使用外部时钟
[30]	sbpi_rf_sel	1'b0	数据采样时钟沿选择 0: 上升沿 1: 下降沿 主SPI模式时, 下降沿采样为快速模式 主BPI异步模式时, 下降沿采样为快速模式 主BPI同步模式时, 上升沿采样为快速模式
[29: 21]	保留		
[20: 19]	bpi_cycle	2'd0	BPI Flash页首地址时钟周期数 00: 1 01: 2 10: 3 11: 4
[18: 17]	bpi_page	2'd0	BPI Flash页大小 00: 1字节/半字 01: 4字节/半字 10: 8字节/半字 11: 16字节/半字
[16]	bpi_mode	1'b0	BPI Flash读模式选择 0: 异步 1: 同步
[15: 11]	保留		
[10]	addrwidth	1'b0	SPI地址位宽 0: 24位地址 1: 32位地址
[9: 8]	datawidth	2'd0	SPI数据位宽 00: x1 01: x2 10: x4 11: x8
[7: 0]	opcode	8'h0B	SPI主模式操作码

## 17) RCR 控制寄存器 (RCRR)

RCR寄存器提供Master BPI模式编程读配置寄存器指令的操作码和数据。

表 31 RCR 控制寄存器说明

位	名称	初始值	描述
[31: 24]	confirm_prcr	8'h03	编程读配置寄存器指令的确认写周期操作码
[23: 16]	setup_prcr	8'h60	编程读配置寄存器指令的建立写周期操作码
[15: 0]	rcr	16'd0	需要编程到BPI Flash的读配置寄存器的内容

## 18) SEU 控制寄存器 (SEUR)

SEU控制寄存器提供SEU检错纠错和回读CRC控制。

表 32 SEU 控制寄存器说明

位	名称	初始值	描述
[31: 2]	保留		
[1]	seu_en	1'b0	SEU使能
[0]	rbcrc_en	1'b0	回读CRC使能

## 19) SEU 状态寄存器 (SEUSTATUSR)

SEU状态寄存器说明如下所示。

表 33 SEU 状态寄存器说明

位	名称	描述
[31:15]	保留	
[14:3]	seu_index	SEU单比特错误在帧中的位置指示
[2]	seu_ded	SEU双比特错误标志
[1]	seu_sec	SEU单比特错误标志
[0]	seu_over	SEU完成标志

## 20) 热启动控制寄存器 (IRSTCTRLR)

热启动控制。

表 34 热启动控制寄存器说明

位	名称	初始值	描述
[31:2]	保留		
[1]	vs_en	1'b0	vs[1:0] pin使能 0: 禁止 1: 使能
[0]	vback_en	1'b0	版本回退使能。控制回退时，回退到黄金位流还是上一版本位流 0: 禁止 1: 使能

## 21) 热启动地址寄存器 (IRSTADDR)

热启动时Flash首地址，字节寻址。

表 35 热启动地址寄存器说明

位	名称	初始值	描述
[31:24]	Flash首地址	8'd0	Flash首地址[31:24]位，用于4-Byte模式
[23:0]	Flash首地址	24'd0	Flash首地址

## 22) 历史状态寄存器 (HSTATUSR)

表 36 历史状态寄存器说明

位	名称	描述
[31:15]	保留	
[14]	timeout1	上个状态timeout 0: 看门狗未超时 1: 看门狗超时
[13]	rbcrc_err1	上个状态回读CRC错误 0: 正确 1: 错误
[12]	crc_err1	上个状态CRC错误 0: 正确 1: 错误
[11]	id_err1	上个状态ID错误 0: 正确 1: 错误
[10]	irst1	上个状态的版本为热启动升级版本指示 0: 非热启动升级版本 1: 热启动升级版本
[9]	fallback1	指示上个状态是否为fallback操作 0: 否 1: 是
[8]	valid1	上个状态有效 0: 无效 1: 有效
[7]	1'b0	保留
[6]	timeout0	当前状态timeout 0: 看门狗未超时 1: 看门狗超时
[5]	rbcrc_err0	当前状态回读CRC错误 0: 正确 1: 错误
[4]	crc_err0	当前状态CRC错误 0: 正确 1: 错误
[3]	id_err0	当前状态ID错误 0: 正确 1: 错误
[2]	irst0	当前状态的版本为热启动升级版本指示 0: 非热启动升级版本 1: 热启动升级版本
[1]	fallback0	指示当前状态是否为fallback操作 0: 否 1: 是
[0]	valid0	当前状态有效 0: 无效

位	名称	描述
		1: 有效

## 23) 位流 0 地址寄存器 (FADR0R)

位流0在Flash中的起始地址，字节寻址。

表 37 位流 0 地址寄存器说明

位	名称	初始值	描述
[31:24]	faddr0[31:24]	8'd0	位流0起始地址高8位，用于4-Byte模式
[23:0]	faddr0[23:0]	24'd0	位流0起始地址

## 24) 位流 1 地址寄存器 (FADR1R)

位流1在Flash中的起始地址，字节寻址。

表 38 位流 1 地址寄存器说明

位	名称	初始值	描述
[31:24]	faddr1[31:24]	8'd0	位流1起始地址高8位，用于4-Byte模式
[23:0]	faddr1[23:0]	24'd0	位流1起始地址

## 25) 位流 2 地址寄存器 (FADR2R)

位流2在Flash中的起始地址，字节寻址。

表 39 位流 2 地址寄存器说明

位	名称	初始值	描述
[31:24]	faddr2[31:24]	8'd0	位流2起始地址高8位，用于4-Byte模式
[23:0]	faddr2[23:0]	24'd0	位流2起始地址

## 26) 位流 3 地址寄存器 (FADR3R)

位流3在Flash中的起始地址，寻址单位为字节。

表 40 位流 3 地址寄存器说明

位	名称	初始值	描述
[31:24]	faddr3[31:24]	8'd0	位流3起始地址高8位，用于4-Byte模式
[23:0]	faddr3[23:0]	24'd0	位流3起始地址

## 27) 版本回退寄存器 (FALLBACKR)

版本回退寄存器用来指定应用位流重试次数和版本回退尝试次数。

表 41 版本回退寄存器说明

位	名称	初始值	描述
---	----	-----	----

[31:8]	num_fallback	24'd1	版本回退尝试次数
[7:0]	num_retry	8'd0	应用位流重试次数

#### （四）位流格式

本节只列举三种常用位流格式，即普通位流、压缩位流与加密位流。

##### 1. 普通位流

普通位流格式如下表所示。

表 42 普通位流格式

FFFFFFFF	填充字（100个）
.....	
FFFFFFFF	
000000AA	总线位宽自动检测
08100020	
FFFFFFFF	填充字（10个）
.....	
FFFFFFFF	
01332D94	同步
AEC00001	类型1包头：写RCR控制寄存器
xxxxxxxx	数据：RCR指令的操作码和数据
AB000001	类型1包头：写SBPIR寄存器
xxxxxxxx	数据：SBPI主模式配置控制选项
A0000000	无操作类型1包头（10个）
.....	
A0000000	
ABC00001	类型1包头：写IRSTCTRLR寄存器
xxxxxxxx	数据：IRSTCTRLR的内容
AC000001	类型1包头：写IRSTADDR寄存器
xxxxxxxx	数据：IRSTADDR的内容
A8800001	类型1包头：写CMDR寄存器
00000000	数据：NOP命令
A8800001	类型1包头：写CMDR寄存器
00000001	数据：RSTCRC命令
A8400001	类型1包头：写IDR寄存器
xxxxxxxx	数据：IDCODE
AC400001	类型1包头：写WATCHDOG寄存器
xxxxxxxx	数据：WATCHDOG
ADC00001	类型1包头：写CMASKR寄存器
xxxxxxxx	数据：CMASKR的内容
A8C00001	类型1包头：写CTRL0R寄存器
xxxxxxxx	数据：CTRL0R的内容
AE400001	类型1包头：写OPTION0R寄存器
xxxxxxxx	数据：OPTION0R的内容

AE800001	类型1包头：写OPTION1R寄存器
xxxxxxx	数据：OPTION1R的内容
AE000001	类型1包头：写FALLBACK寄存器
xxxxxxx	数据：FALLBACK的内容
A8800001	类型1包头：写CMDR寄存器
00000002	数据：SWITCH命令
A0000000	无操作类型1包头（10个）
.....	
A0000000	
AAC00001	类型1包头：写ADRR寄存器
xxxxxxx	数据：ADRR的内容
A8800001	类型1包头：写CMDR寄存器
00000004	数据：WCMEM命令
A9400000	类型1包头：写CMEMIR寄存器
{3'b010, 2'b01, 27'dx}	类型2包头：写操作
xxxxxxx	数据：写入配置存储器的内容
.....	
xxxxxxx	
A0000000	无操作类型1包头（20个）
.....	
A0000000	
A8000001	类型1包头：写CRCR寄存器
xxxxxxx	数据：CRC值
ADC00001	类型1包头：写CMASKR寄存器
xxxxxxx	数据：CMASKR的内容
A9000001	类型1包头：写CTRL1R寄存器
xxxxxxx	数据：CTRL1R的内容
A8800001	类型1包头：写CMDR寄存器
00000009	数据：GUP命令
A8800001	类型1包头：写CMDR寄存器
00000007	数据：SWAKEUP命令
A8000001	类型1包头：写CRCR寄存器
xxxxxxx	数据：CRC值
A8800001	类型1包头：写CMDR寄存器
0000000B	数据：DESYNC命令
A0000000	无操作类型1包头（100个）
.....	
A0000000	

## 2. 压缩位流

一次压缩操作只能在同一个区域中进行，内容为全0的帧不进行配置。压缩位流格式如下表所示。

表 43 压缩位流格式

FFFFFFFF	填充字（100个）
.....	
FFFFFFFF	



000000AA	总线位宽自动检测
08100020	
FFFFFFFF	填充字（10个）
.....	
FFFFFFFF	
01332D94	同步
AEC00001	类型1包头：写RCR控制寄存器
xxxxxxxx	数据：RCR指令的操作码和数据
AB000001	类型1包头：写SBPIR寄存器
xxxxxxxx	数据：SBPI主模式配置控制选项
A0000000	无操作类型1包头（10个）
.....	
A0000000	
ABC00001	类型1包头：写IRSTCTRL寄存器
xxxxxxxx	数据：IRSTCTRL的内容
AC000001	类型1包头：写IRSTADDR寄存器
xxxxxxxx	数据：IRSTADDR的内容
A8800001	类型1包头：写CMDR寄存器
00000000	数据：NOP命令
A8800001	类型1包头：写CMDR寄存器
00000001	数据：RSTCRC命令
A8400001	类型1包头：写IDR寄存器
xxxxxxxx	数据：IDCODE
AC400001	类型1包头：写WATCHDOG寄存器
xxxxxxxx	数据：WATCHDOG
ADC00001	类型1包头：写CMASKR寄存器
xxxxxxxx	数据：CMASKR的内容
A8C00001	类型1包头：写CTRL0寄存器
xxxxxxxx	数据：CTRL0的内容
AE400001	类型1包头：写OPTION0R寄存器
xxxxxxxx	数据：OPTION0R的内容
AE800001	类型1包头：写OPTION1R寄存器
xxxxxxxx	数据：OPTION1R的内容
AE000001	类型1包头：写FALLBACK寄存器
xxxxxxxx	数据：FALLBACK的内容
A8800001	类型1包头：写CMDR寄存器
00000002	数据：SWITCH命令
A0000000	无操作类型1包头（10个）
.....	
A0000000	
AAC00001	类型1包头：写ADRR寄存器
xxxxxxxx	数据：ADRR的内容
A8800001	类型1包头：写CMDR寄存器
00000004	数据：WCMEM命令
A9400000	类型1包头：写CMEMIR寄存器
{3'b010, 2'b01, 27'dx}	类型2包头：写操作
xxxxxxxx	数据：写入配置存储器的内容

.....	
XXXXXXXX	
A0000000	
.....	无操作类型1包头（20个）
A0000000	
AAC00001	类型1包头：写ADRR寄存器
XXXXXXXX	数据：ADRR的内容
A9800001	类型1包头：写MFWRITER寄存器
XXXXXXXX	数据：压缩的帧数
A8800001	类型1包头：写CMDR寄存器
00000005	数据：MFWRITE命令
A0000000	
.....	无操作类型1包头（压缩帧数*总线位宽/2）
A0000000	
A8800001	类型1包头：写CMDR寄存器
00000000	数据：NOP命令
AAC00001	类型1包头：写ADRR寄存器
XXXXXXXX	数据：ADRR的内容
A8800001	类型1包头：写CMDR寄存器
00000004	数据：WCMEM命令
A9400000	类型1包头：写CMEMIR寄存器
{3'b010, 2'b01, 27'dx}	类型2包头：写操作
XXXXXXXX	
.....	数据：写入配置存储器的内容
XXXXXXXX	
A0000000	
.....	无操作类型1包头（20个）
A0000000	
A8000001	类型1包头：写CRCR寄存器
XXXXXXXX	数据：CRC值
ADC00001	类型1包头：写CMASKR寄存器
XXXXXXXX	数据：CMASKR的内容
A9000001	类型1包头：写CTRL1R寄存器
XXXXXXXX	数据：CTRL1R的内容
A8800001	类型1包头：写CMDR寄存器
00000009	数据：GUP命令
A8800001	类型1包头：写CMDR寄存器
00000007	数据：SWAKEUP命令
A8000001	类型1包头：写CRCR寄存器
XXXXXXXX	数据：CRC值
A8800001	类型1包头：写CMDR寄存器
0000000B	数据：DESYNC命令
A0000000	
.....	无操作类型1包头（100个）
A0000000	

### 3. 加密位流

只支持完整位流的加密。

表 44 加密位流格式

FFFFFFFF	填充字（100个）
.....	
FFFFFFFF	
000000AA	总线位宽自动检测
08100020	
FFFFFFFF	填充字（10个）
.....	
FFFFFFFF	
01332D94	同步
AEC00001	类型1包头：写RCR控制寄存器
xxxxxxxx	数据：RCR指令的操作码和数据
AB000001	类型1包头：写SBPIR寄存器
xxxxxxxx	数据：SBPI主模式配置控制选项
A0000000	无操作类型1包头（10个）
.....	
A0000000	
ABC00001	类型1包头：写IRSTCTRL寄存器
xxxxxxxx	数据：IRSTCTRL的内容
AC000001	类型1包头：写IRSTADDR寄存器
xxxxxxxx	数据：IRSTADDR的内容
A8800001	类型1包头：写CMDR寄存器
00000000	数据：NOP命令
A8800001	类型1包头：写CMDR寄存器
00000001	数据：RSTCRC命令
A8400001	类型1包头：写IDR寄存器
xxxxxxxx	数据：IDCODE
AC400001	类型1包头：写WATCHDOG寄存器
xxxxxxxx	数据：WATCHDOG
AA000004	类型1包头：写IVR寄存器
xxxxxxxx	数据：IV值
xxxxxxxx	
xxxxxxxx	
xxxxxxxx	
ADC00001	类型1包头：写CMASKR寄存器
xxxxxxxx	数据：CMASKR的内容
A8C00001	类型1包头：写CTRL0寄存器
xxxxxxxx	数据：CTRL0的内容
AE400001	类型1包头：写OPTION0R寄存器
xxxxxxxx	数据：OPTION0R的内容
AE800001	类型1包头：写OPTION1R寄存器
xxxxxxxx	数据：OPTION1R的内容
AE000001	类型1包头：写FALLBACK寄存器

xxxxxxx	数据：FALLBACK的内容
A8800001	类型1包头：写CMDR寄存器
00000002	数据：SWITCH命令
A0000000	无操作类型1包头（10个）
.....	
A0000000	
AAC00001	类型1包头：写ADRR寄存器
xxxxxxx	数据：ADRR的内容
A8800001	类型1包头：写CMDR寄存器
00000004	数据：WCMEM命令
A9400000	类型1包头：写CMEMIR寄存器
{3'b010, 2'b01, 27'dx}	类型2包头：写操作
xxxxxxx	数据：写入配置存储器的内容
.....	
xxxxxxx	
A0000000	无操作类型1包头（20个）
.....	
A0000000	
A8000001	类型1包头：写CRCR寄存器
xxxxxxx	数据：CRC值
ADC00001	类型1包头：写CMASKR寄存器
xxxxxxx	数据：CMASKR的内容
A9000001	类型1包头：写CTRL1R寄存器
xxxxxxx	数据：CTRL1R的内容
A8800001	类型1包头：写CMDR寄存器
00000009	数据：GUP命令
A8800001	类型1包头：写CMDR寄存器
00000007	数据：SWAKEUP命令
A8000001	类型1包头：写CRCR寄存器
xxxxxxx	数据：CRC值
A8800001	类型1包头：写CMDR寄存器
0000000B	数据：DESYNC命令
A0000000	无操作类型1包头（100个）
.....	
A0000000	
A0000000	无操作类型1包头 （凑齐128比特边界）
.....	
A0000000	

### （五）下载线速率

当使用USB Cable进行下载时，包括JTAG下载以及对SPI Flash烧录，默认速率是10MHz。

下载速率可以在Fabric Configuration中设置，点击图标Connect To Server。

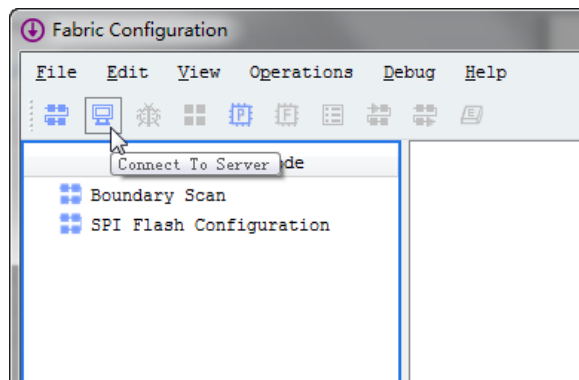


图 58 Fabric Configuration

若未连接，则先连接，如下左图所示，若已连接，则点击Next，如下右图所示：

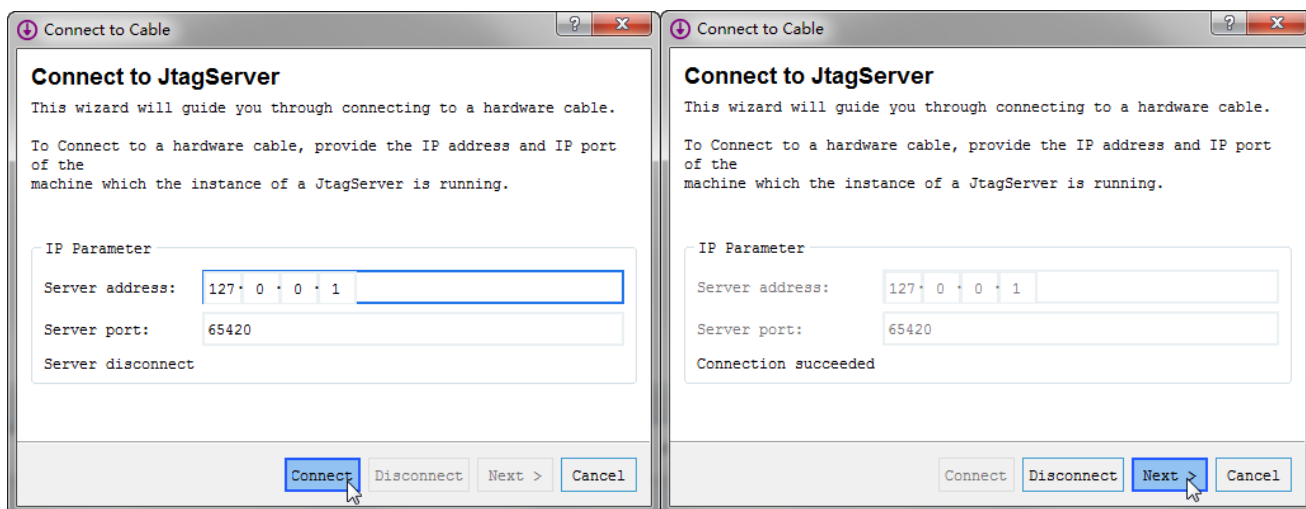


图 59 Connect To Cable

在TCK Frequency下拉框中可以选择USB Cable频率。

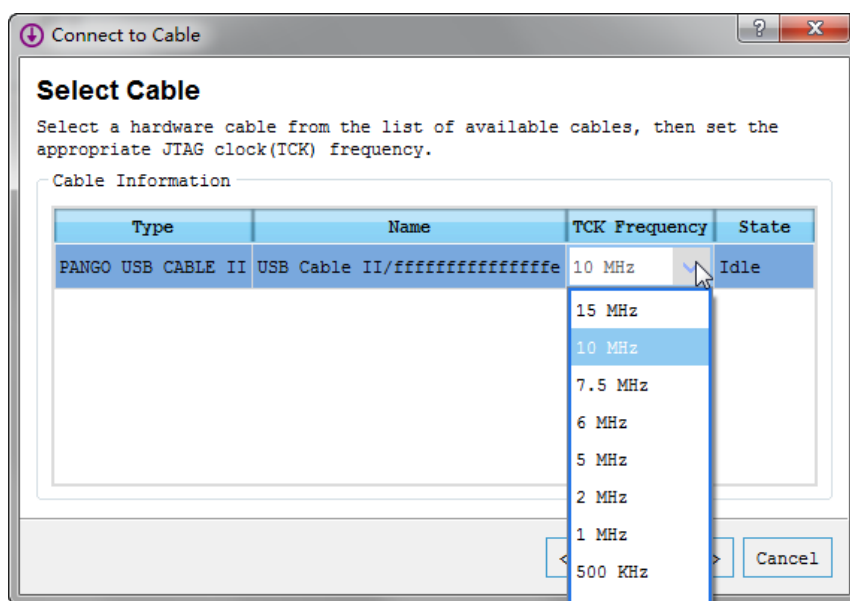


图 60 USB Cable 速率设置

## （六）配置速率

### 1. JTAG 模式

JTAG模式下载，使用下载线下载时，受限于下载线速率，最高配置频率为15MHz；若以其它接口模拟进行JTAG下载，速率最高可为50MHz。

### 2. Master SPI 模式

Master SPI模式，CFG\_CLK时钟频率可配置，默认为3.125MHz。高速模式下（Falling Edge），支持最大配置频率50MHz；低速模式下（Rising Edge），支持最大配置频率20MHz；菊花链模式支持最大配置频率33MHz。

### 3. Master BPI 模式

Master BPI模式，CFG\_CLK时钟频率可配置，默认为3.125MHz。低速模式下（Rising Edge），异步配置支持最大频率为10MHz，同步配置支持最大频率25MHz；高速模式下（Falling Edge），异步配置支持最大频率为33MHz，同步配置支持最大频率50MHz。

### 4. Slave SPI 模式

Slave SPI模式，下载时钟来自外部Host设备，配置时最高可设为100MHz，回读最高频率为60MHz。

### 5. Slave Parallel 模式

Slave Parallel模式下，下载时钟来自外部Host设备，时钟速率最高可为100MHz。

### 6. Slave Serial 模式

Slave Serial模式下，下载时钟来自外部Host设备，最高可设为100MHz；串行菊花链模式最高可支持50MHz。

## （七）多功能配置

在主模式下，Logos系列FPGA（PGL12G/22G/22GS）支持多功能配置。

用户可以最多存储四个独立功能的位流到外部Flash。每种功能可以是任意类型的位流（普通位流、压缩位流或者加密位流）。可通过设置管脚VS[1:0]的值来选择配置哪个功能的位流。具体见下表所示。

表 45 VS[1:0]的值与位流的对应关系

VS[1:0]值	位流
2'b00	功能位流 0
2'b01	功能位流 1
2'b10	功能位流 2
2'b11	功能位流 3

多功能位流的功能跳转程序使能VS[1:0]管脚，控制最多四种功能的起始地址。跳转到指定功能后，禁止VS[1:0]管脚。

表 46 多功能跳转程序

FFFFFFFF	填充字（100 个）	功能跳转程序
.....		
FFFFFFFF		
000000AA	总线位宽自动检测	
08100020		
FFFFFFFF	填充字（10 个）	
.....		
FFFFFFFF		
01332D94	同步	
AB000001	类型 1 包头：写 SPIR 寄存器	
00000xxx	数据：SPI 主模式操作码	
A0000000	无操作类型 1 包头（10 个）	
.....		
A0000000		
ABC00001	类型 1 包头：写 IRSTCTRL 寄存器	
xxxxxxxx	数据：IRSTCTRL 的内容	
AC000001	类型 1 包头：写 IRSTADDR 寄存器	
xxxxxxxx	数据：IRSTADDR 的内容	
ACC00001	类型 1 包头：写 FADDR0 寄存器	
xxxxxxxx	数据：FADDR0 的内容	
AD000001	类型 1 包头：写 FADDR1 寄存器	
xxxxxxxx	数据：FADDR1 的内容	
AD400001	类型 1 包头：写 FADDR2 寄存器	
xxxxxxxx	数据：FADDR2 的内容	
AD800001	类型 1 包头：写 FADDR3 寄存器	
xxxxxxxx	数据：FADDR3 的内容	
A8800001	类型 1 包头：写 CMDR 寄存器	
0000000F	数据：IRST 命令	
	位流 0	功能 0

	位流 1	功能 1
	位流 2	功能 2
	位流 3	功能 3

操作流程如下：

- 1、从Flash0地址载入功能跳转程序。
- 2、从VS[1:0]选中的位流地址寄存器指定的Flash地址载入功能。
- 3、改变VS[1:0]值，触发硬复位。
- 4、重复步骤1，2，3。

如果载入应用位流过程中有错误，VS[1:0]选中的位流地址寄存器指定的Flash地址载入VS[1:0]选中功能的黄金位流或回退到上一版本应用位流。

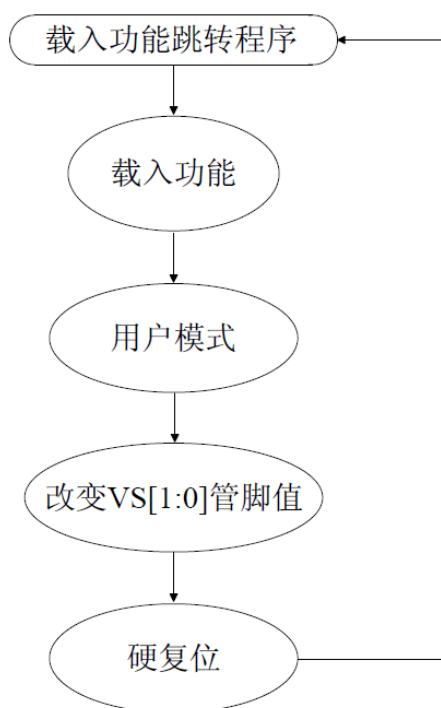


图 61 跳转流程

## （八）远程升级

Logos系列FPGA支持Master SPI和Master BPI接口的远程升级。远程升级时，用户逻辑通过通信协议（例如TCP/IP，PCI，UDP，UART等）或专有接口从远端接收位流。通过用户SPI接口将位流编程到内部或外部Flash。

远程升级时，可以通过JTAG接口将位流编程到内部/外部Flash；一般不使用外部管脚VS[1:0]控制，需要将IRSTCTRLR寄存器的VS[1:0]管脚使能位设为0。



## 1. 一个应用位流

### 1) 位流

Logos FPGA通过同步字（32'h01332D94）来识别有效数据，通过在数据前添加或删除同步字，可使数据被识别或屏蔽。在远程升级位流中，以此作为开关，控制紧接着的跳转程序是否有效。

远程升级时的位流如下所示：

表 47 远程升级位流

FFFFFFFF	填充字（1023 个）	应用位流开关程序 位于 Flash 第 1 个扇区
.....		
FFFFFFFF		
01332D94	应用位流开关	
A0000000	无操作类型 1 包头	应用位流跳转程序 位于 Flash 第 2 个扇区
AB000001	类型 1 包头：写 SPIR 寄存器	
00000xxx	数据：SPI 主模式操作码	
A0000000	无操作类型 1 包头（10 个）	
.....		
A0000000		
ABC00001	类型 1 包头：写 IRSTCTRL 寄存器	
xxxxxxxx	数据：IRSTCTRL 的内容	
AC000001	类型 1 包头：写 IRSTADRR 寄存器	
xxxxxxxx	数据：IRSTADRR 的内容	
A8800001	类型 1 包头：写 CMDR 寄存器	
0000000F	数据：IRST 命令	
A0000000	无操作类型 1 包头（1005 个）	
.....		
A0000000		
	黄金位流	
	应用位流	从黄金位流的下一扇区开始

### 2) 操作流程

- ① 从Flash 0地址载入应用位流开关程序和应用位流跳转程序。
- ② 从IRSTCTRLR寄存器指定的Flash地址载入应用位流。
- ③ JTAG或用户擦除应用位流开关程序。
- ④ JTAG或用户编程Flash，更新应用位流。
- ⑤ JTAG或用户编程Flash，开启应用位流开关。
- ⑥ JTAG热启动或IPAL热启动或冷启动或上电复位或JTAG指令复位。
- ⑦ 如果JTAG热启动或IPAL热启动，重复步骤②，③，④，⑤，⑥。如果冷启动（上电复位/硬复位/JTAG指令复位），重复步骤①，②，③，④，⑤，⑥。

如果载入应用位流过程中有错误，从Flash 0地址载入黄金位流。

流程图如下所示：

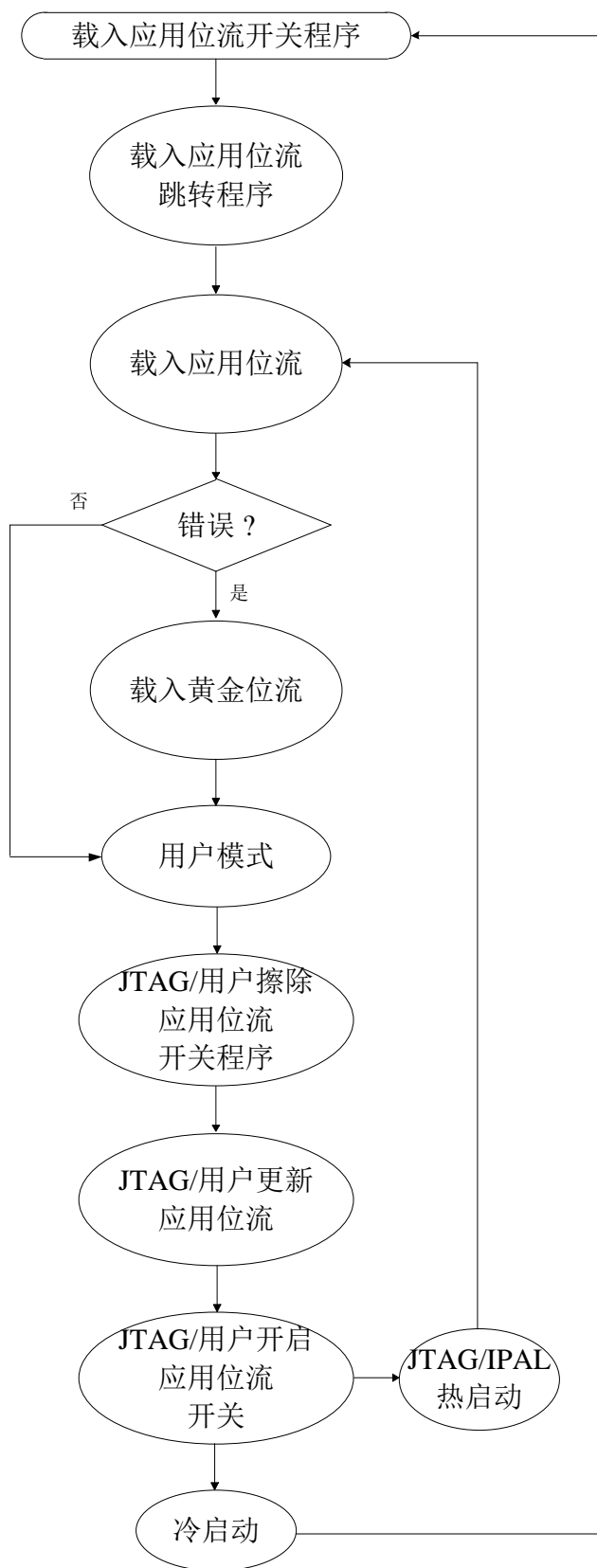


图 62 远程升级一个应用位流流程图

## 2. 多个应用位流

### 1) 位流

远程升级多个应用位流的格式如下所示：

表 48 多个应用位流格式

FFFFFFFF	填充字（1023 个）	应用位流 0 开关程序 位于 Flash 第 1 个扇区
.....		
FFFFFFFF		
01332D94	应用位流 0 开关	应用位流 0 跳转程序 位于 Flash 第 2 个扇区
A0000000	无操作类型 1 包头	
AB000001	类型 1 包头：写 SPIR 寄存器	
00000xxx	数据：SPI 主模式操作码	
A0000000	无操作类型 1 包头（10 个）	
.....		
A0000000		
ABC00001	类型 1 包头：写 IRSTCTRL 寄存器	
xxxxxxxxxx	数据：IRSTCTRL 的内容	
AC000001	类型 1 包头：写 IRSTADRR 寄存器	
xxxxxxxxxx	数据：IRSTADRR 的内容	
A8800001	类型 1 包头：写 CMDR 寄存器	
0000000F	数据：IRST 命令	
A0000000	无操作类型 1 包头（1005 个）	
.....		
A0000000		
FFFFFFFF	填充字（1023 个）	应用位流 1 开关程序 位于 Flash 第 3 个扇区
.....		
FFFFFFFF		
01332D94	应用位流 1 开关	应用位流 1 跳转程序 位于 Flash 第 4 个扇区
A0000000	无操作类型 1 包头	
AB000001	类型 1 包头：写 SPIR 寄存器	
00000xxx	数据：SPI 主模式操作码	
A0000000	无操作类型 1 包头（10 个）	
.....		
A0000000		
ABC00001	类型 1 包头：写 IRSTCTRL 寄存器	
xxxxxxxxxx	数据：IRSTCTRL 的内容	
AC000001	类型 1 包头：写 IRSTADRR 寄存器	
xxxxxxxxxx	数据：IRSTADRR 的内容	
A8800001	类型 1 包头：写 CMDR 寄存器	
0000000F	数据：IRST 命令	
A0000000	无操作类型 1 包头（1005 个）	
.....		
A0000000		
FFFFFFFF	填充字（1023 个）	应用位流 2 开关程序

.....		位于 Flash 第 5 个扇区
FFFFFFFF		
01332D94	应用位流 2 开关	
A0000000	无操作类型 1 包头	应用位流 2 跳转程序 位于 Flash 第 6 个扇区
AB000001	类型 1 包头：写 SPIR 寄存器	
00000xxx	数据：SPI 主模式操作码	
A0000000	无操作类型 1 包头（10 个）	
.....		
A0000000		
ABC00001		
xxxxxxxxxx	数据：IRSTCTRL 的内容	
AC000001	类型 1 包头：写 IRSTADRR 寄存器	
xxxxxxxxxx	数据：IRSTADRR 的内容	
A8800001	类型 1 包头：写 CMDR 寄存器	
0000000F	数据：IRST 命令	
A0000000	无操作类型 1 包头（1005 个）	
.....		
A0000000		
	黄金位流	从 Flash 第 7 个扇区开始 占有整数个扇区
	应用位流 0	从黄金位流的下一扇区开始 使用版本回退功能时，在更新新版本应用位流时，需保留旧版本应用位流。即新旧位流不能使用同一扇区
	应用位流 1	
	应用位流 2	

## 2) 操作流程

- ① 从FLASH 0地址载入应用位流开关程序和应用位流跳转程序。
- ② 从IRSTCTRLR寄存器指定的FLASH地址载入应用位流。
- ③ JTAG或用户擦除旧版本应用位流开关程序。
- ④ JTAG或用户编程FLASH, 更新应用位流。如果新版本应用位流已在FLASH中, 则不需要更新。
- ⑤ JTAG或用户编程FLASH, 更新新版本应用位流跳转程序。如果新版本应用位流起始地址不变, 则不需要更新。
- ⑥ JTAG或用户编程FLASH, 开启新版本应用位流开关。
- ⑦ JTAG热启动/IPAL热启动/冷启动（上电复位/硬复位/JTAG指令复位）。
- ⑧ 如果JTAG热启动/IPAL热启动, 重复步骤②, ③, ④, ⑤, ⑥, ⑦。如果冷启动（上电复位/硬复位/JTAG指令复位）, 重复步骤①, ②, ③, ④, ⑤, ⑥, ⑦。

如果载入应用位流过程中有错误, 从Flash 0地址载入黄金位流或回退到上一版本应用位流。流程

图如下所示：

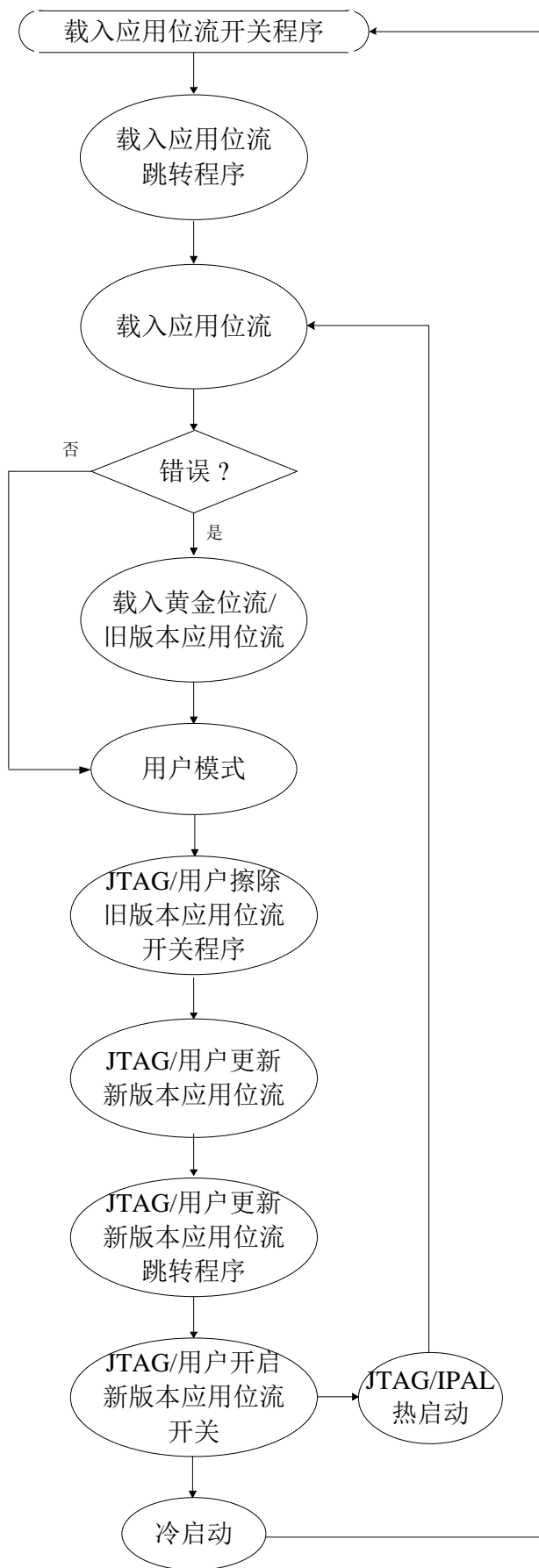


图 63 远程升级多个应用位流流程图

## （九）多启动

使用多启动功能时，外部Flash最少需要支持存放2套位流。位流0为黄金位流版本或应用位流版本，起始地址为0，其它位流为应用位流。配置过程中出现任何错误，复位版本回退电路以外的电路，重新载入黄金位流或上个版本应用位流。如果黄金位流或上一版本应用位流也有错误，那就不再复位了，而是在INIT\_FLAG\_N置0的同时，FCS\_N置1，SPI主模式操作结束。

使用多启动功能时，需要打开看门狗。

配置过程中，以下错误会触发版本回退。

- 1、器件 ID 错误
- 2、CRC 错误
- 3、看门狗超时

### 1. 黄金位流初始化系统

- 1、从Flash 0地址载入黄金位流。
- 2、JTAG或用户进行Flash操作。（可选）如果用于多启动的应用位流已经在Flash中，可以不进行Flash操作。
- 3、JTAG热启动。
- 4、从IRSTCTRLR寄存器指定的Flash地址载入应用位流。
- 5、如果载入应用位流过程中有错误，从Flash 0地址载入黄金位流或回退到上一版本应用位流。

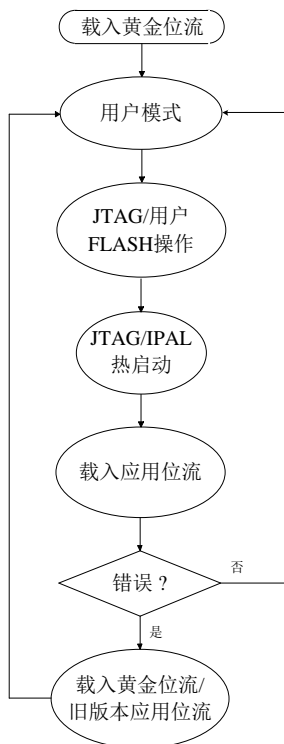


图 64 黄金位流加载流程

## 2. 应用位流初始化系统

- 1、从Flash 0地址载入黄金位流。
- 2、黄金位流内嵌热启动使从IRSTCTRLR指定的Flash地址载入应用位流。
- 3、如果载入应用位流过程中有错误，从Flash 0地址载入黄金位流或回退到上一版本应用位流。

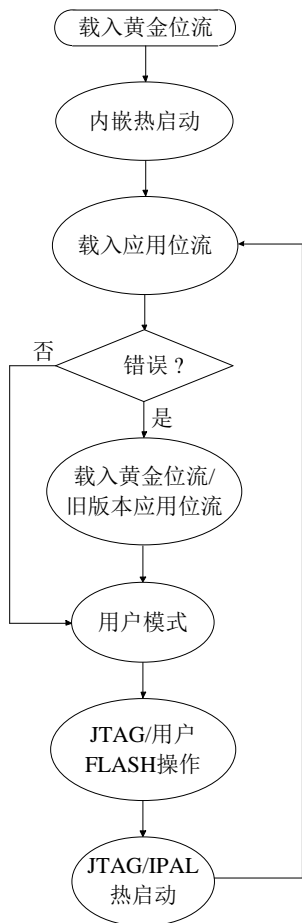


图 65 应用位流加载流程

## 3. 无黄金位流

如果Flash最多支持存放2套位流，使能版本回退功能时，两套位流都为应用位流。

- 1、从Flash 0地址载入应用位流。
- 2、JTAG或用户进行Flash操作。（可选）如果用于多启动的应用位流已经在Flash中，可以不进行Flash操作。
- 3、JTAG热启动或IPAL热启动。
- 4、从IRSTCTRLR寄存器指定的Flash地址载入应用位流。
- 5、如果载入应用位流过程中有错误，回退到上一版本应用位流。

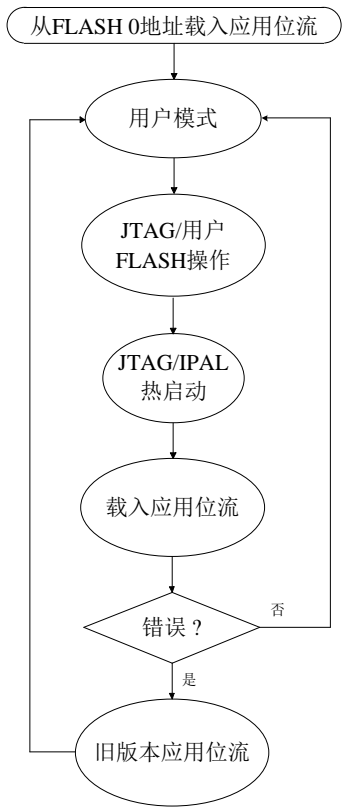


图 66 无黄金位流时加载流程

#### 4. 热启动

向器件发送指令前，需先同步，确保器件能正确解析指令。若通过Slave Parallel接口发送指令，还需发送总线位宽检测指令。在多启动应用中，黄金位流包含写IRSTADDR寄存器指令，写入地址为应用位流起始地址，热启动时无需包含写IRSTADDR寄存器操作。若多启动中不包含黄金位流，则需要用户在热启动流程中往IRSTADDR寄存器写入热启动时载入位流的起始地址。若黄金位流中内嵌热启动指令，即在应用位流初始化系统中，不可通过热启动升级到黄金位流。

热启动完整流程如下表所示。

表 49 热启动流程

FFFFFFFF	填充字（100 个）
.....	
FFFFFFFF	
000000AA	总线位宽自动检测
08100020	
FFFFFFFF	填充字（10 个）
.....	
FFFFFFFF	
01332D94	同步
AC000001	类型 1 包头：写 IRSTADDR 寄存器（可选）
xxxxxxxx	数据：IRSTADDR 的内容（可选）
A8800001	类型 1 包头：写 CMDR 寄存器



0000000F	数据：IRST 命令
A8800001	类型 1 包头：写 CMDR 寄存器
0000000B	数据：DESYNC 命令
A0000000	无操作类型 1 包头（100 个）
.....	
A0000000	

## 5. 看门狗

看门狗用于检测配置过程 and 用户模式下的超时。

每经过512个系统时钟周期，看门狗计数减1。看门狗计数减为0时，产生看门狗超时标志。

主模式配置过程中，看门狗超时引发版本回退。

版本回退过程中和版本回退结束后，看门狗被禁止。版本回退成功结束后，热启动可以取消对看门狗的禁止。

## 四、 回读操作

Logos FPGA支持通过JTAG和Slave Parallel接口回读普通位流和压缩位流。

### （一） JTAG 接口回读

JTAG接口回读操作前需要检测IR中的唤醒结束信号wakeup\_over或状态寄存器中的唤醒结束信号wakeup\_over，确保唤醒后才能回读。通常唤醒结束很长时间以后才会回读。

### （二） Slave Parallel 接口回读

使用Slave Parallel接口进行回读，需确保该接口被使能或保留。

如在配置模式下使用该接口进行回读，应确保当前生效的配置模式为Slave Parallel模式，即MODE管脚设置为Slave Parallel模式。

如在用户模式下使用该接口进行回读，应确保当前生效的位流中设置了保留配置接口（设置位置为PDS中的Project Setting->Generate Bitstream->Readback->Persist Slave Parallel Pins，默认不保留）。并且回读使用的数据位宽必须与配置时的位宽一致，即若使用Slave Parallel x8进行配置，并保留了配置接口，那么可以通过Slave Parallel x8进行回读，而不能以x16/x32位宽进行回读。

使用Slave Parallel接口进行回读时，需要首先按照Slave Parallel接口时序要求，向器件发送一些指令，见下文回读配置存储器流程及回读配置寄存器流程，其中指令流以32bits形式呈现，需根据[不同接口位宽下的位序](#)发送。器件在正确接收到指令后，会在数据准备好后，在时钟上升沿将数据发出，用户即可采样数据。回读配置寄存器和配置存储器流程不同，如下文所述。回读需要满足的时序要求参考第二章的[Slave Parallel配置接口典型时序图](#)。

#### 1. 回读配置存储器（有 busy 端口）

Slave Parallel接口回读操作前需要检测状态寄存器中的唤醒结束信号wakeup\_over，确保唤醒后才能回读。通常唤醒结束很长时间以后才会回读。

有busy端口的回读流程如下所示：

表 50 有 busy 端口的回读流程

流程	指令流（Hex）/操作
100 个填充字	FFFFFFFF ..... FFFFFFFF
总线位宽自动检测	000000AA 08100020
10 个填充字	FFFFFFFF .....

	FFFFFFFF
同步字	01332D94
类型 1 包的包头：空操作	A0000000
写 RCMEM 命令到 CMDR 寄存器	A8800001 00000006
写 ADDR 寄存器	AAC00001 00000000 <sup>(1)</sup>
写类型 1 包的包头，用于读 CMEMOR 寄存器	B1C00000
写类型 2 包的包头，用于读数据	5xxxxxxx <sup>(2)</sup>
读写切换（CS_N 为 1，时钟 CFG_CLK 持续有）	拉高 CS_N，并在 CS_N 为 1 时改变 RWSEL 为读 <sup>(3)</sup>
读 CMEMOR 寄存器	拉低 CS_N，采样数据
读写切换（CS_N 为 1，时钟 CFG_CLK 持续有）	拉高 CS_N，并在 CS_N 为 1 时改变 RWSEL 为写 <sup>(3)</sup>
写 DESYNC 命令到 CMDR 寄存器	A8800001 0000000B
100 个无操作类型 1 包头	A0000000 ..... A0000000

注：（1）向 ADDR 寄存器写入回读起始地址。

（2）回读的 32bits 字数。

（3）RWSEL 改变需满足的时序要求见 [Slave Parallel 回读时序](#)。

## 2. 回读配置存储器（无 busy 端口）

Slave Parallel接口回读操作前需要检测状态寄存器中的唤醒结束信号wakeup\_over，确保唤醒后才能回读。通常唤醒结束很长时间以后才会回读。

不使用busy端口进行回读时，回读过程中需判断是否数据已经回读完，是否已经回读完一帧。如果数据已经回读完，则继续下一步去同步操作。如果数据没回读完，并且当前帧没有回读完，则继续回读。如果数据没有回读完，但当前帧已回读完，则Slave Parallel接口将CS\_N置1，等待至少20个配置时钟周期。PGL22G器件不支持此操作流程。

表 51 无 busy 端口的回读流程

流程	指令流（Hex）/操作
100 个填充字	FFFFFFFF ..... FFFFFFFF
总线位宽自动检测	000000AA 08100020
10 个填充字	FFFFFFFF ..... FFFFFFFF
同步字	01332D94
类型 1 包的包头：空操作	A0000000
写 RCMEM 命令到 CMDR 寄存器	A8800001 00000006
写 ADDR 寄存器	AAC00001

	00000000 <sup>(1)</sup>
写类型 1 包的包头，用于读 CMEMOR 寄存器	B1C00000
写类型 2 包的包头，用于读数据	5xxxxxxx <sup>(2)</sup>
读写切换，并暂停操作至少 200 个配置时钟周期（CS_N 为 1，时钟 CFG_CLK 持续有）	拉高 CS_N，并在 CS_N 为 1 时改变 RWSEL 为读 <sup>(3)</sup>
读 CMEMOR 寄存器（回读过程中，判断是否回读完，是否是帧边界）	拉低 CS_N，采样数据
读写切换（CS_N 为 1，时钟 CFG_CLK 持续有）	拉高 CS_N，并在 CS_N 为 1 时改变 RWSEL 为写 <sup>(3)</sup>
写 DESYNC 命令到 CMDR 寄存器	A8800001 0000000B
100 个无操作类型 1 包头	A0000000 ..... A0000000

注：（1）向 ADDR 寄存器写入回读起始地址。

（2）回读的 32bits 字数。

（3）RWSEL 改变需满足的时序要求见 [Slave Parallel 回读时序](#)。

### 3. 回读配置寄存器（有 busy 端口）

对于有 busy 端口的回读操作，回读配置寄存器的流程如下所示：

表 52 有 busy 端口的回读配置寄存器流程

流程	指令流（Hex）/操作
100 个填充字	FFFFFFFF ..... FFFFFFFF
总线位宽自动检测	000000AA 08100020
10 个填充字	FFFFFFFF ..... FFFFFFFF
同步	01332D94
类型 1 包的包头：空操作	A0000000
写 NOP 命令到 CMDR 寄存器	A8800001 A0000000
写类型 1 包的包头，用于读指定寄存器	Bxxxxxxx <sup>(1)</sup>
读写切换（CS_N 为 1，时钟 CFG_CLK 持续有）	拉高 CS_N，并在 CS_N 为 1 时改变 RWSEL 为读 <sup>(2)</sup>
读指定寄存器	拉低 CS_N，采样数据
读写切换（CS_N 为 1，时钟 CFG_CLK 持续有）	拉高 CS_N，并在 CS_N 为 1 时改变 RWSEL 为写 <sup>(2)</sup>
写 DESYNC 命令到 CMDR 寄存器	A8800001 0000000B
100 个无操作类型 1 包头	A0000000 ..... A0000000

注：（1）按照[包格式](#)，填入需要回读的[寄存器](#)地址，以及回读的 32bits 字数。

（2）RWSEL 改变需满足的时序要求见 [Slave Parallel 回读时序](#)。

#### 4. 回读配置寄存器（无 busy 端口）

对于没有busy端口的回读操作，回读配置寄存器的流程如下所示：

表 53 无 busy 端口的回读配置寄存器流程

流程	指令流（Hex）/操作
100 个填充字	FFFFFFFF ..... FFFFFFFF
总线位宽自动检测	000000AA 08100020
10 个填充字	FFFFFFFF ..... FFFFFFFF
同步	01332D94
类型 1 包的包头：空操作	A0000000
写 NOP 命令到 CMDR 寄存器	A8800001 A0000000
写类型 1 包的包头，用于读指定寄存器	Bxxxxxxx <sup>(1)</sup>
读写切换，并等待最少 20 个配置时钟周期（CS_N 为 1，时钟 CFG_CLK 持续有）	拉高 CS_N，并在 CS_N 为 1 时改变 RWSEL 为读 <sup>(2)</sup>
读指定寄存器	拉低 CS_N，采样数据
读写切换（CS_N 为 1，时钟 CFG_CLK 持续有）	拉高 CS_N，并在 CS_N 为 1 时改变 RWSEL 为写 <sup>(2)</sup>
写 DESYNC 命令到 CMDR 寄存器	A8800001 0000000B
100 个无操作类型 1 包头	A0000000 ..... A0000000

注：（1）按照[包格式](#)，填入需要回读的[寄存器](#)地址，以及回读的 32bits 字数。

（2）RWSEL 改变需满足的时序要求见 [Slave Parallel 回读时序](#)。

## 五、 回读 CRC

回读CRC检查采用32位CRC算法，在回读时进行。回读数据时，32位数据做一次CRC计算。

连续回读两次，每次计算一个CRC值。如果两次回读计算的CRC结果不一致，状态寄存器的回读CRC错误位置高。

将SEU寄存器中的回读CRC控制位设置成允许回读CRC，SEU检错纠错控制位设置成禁止SEU检错纠错。

回读配置存储器2次后，察看回读CRC结果。（JTAG接口通过回读状态寄存器查看。内部并行接口还可以直接采样SEU输出信息）

如果回读CRC错误，用户自行选择下一步操作。比如重配等。

如果回读CRC正确，可以重复回读配置存储器，持续进行回读CRC检查。每回读配置存储器1次后，察看回读CRC结果。

如果不进行回读CRC检查，将SEU寄存器中的回读CRC控制位设置成禁止回读CRC。

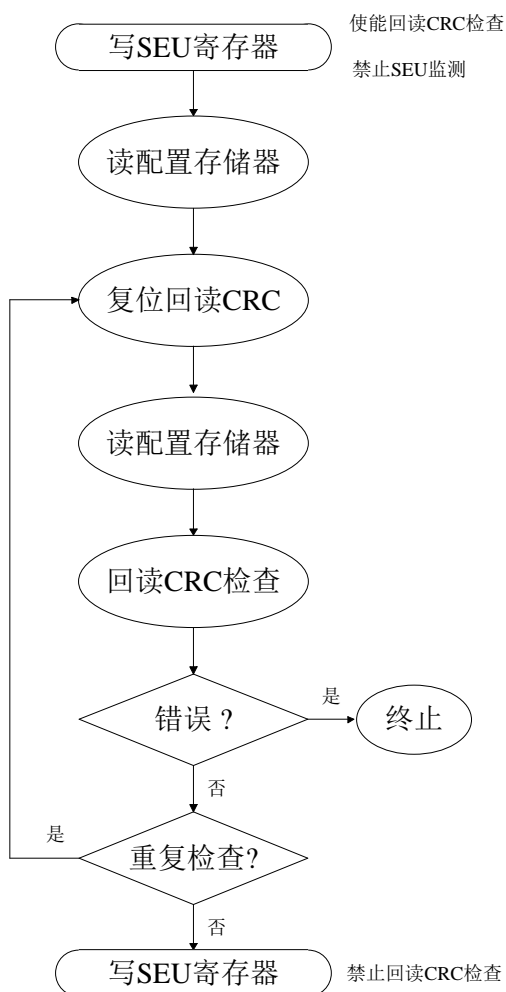


图 67 回读 CRC 流程

具体使用方法见文档《AN020006\_Logos系列FPGA SEU应用指南》。另外，Logos 回读CRC功能可使用IP进行实现，详情参考文档《UG022002\_SEU\_IP》。

## 六、 SEU 检测

Logos FPGA支持SEU 1-bit纠错和2-bit检错。ECC算法使用SECDDED算法，按帧分组。

SEU检错和纠错可以通过JTAG接口，Slave Parallel接口或者内部并行接口进行。

将SEU寄存器中的回读CRC控制位设置成禁止回读CRC，SEU检错纠错控制位设置成允许SEU检错纠错。

回读配置存储器中一帧数据。

查看SEU结果。（JTAG接口和Slave Parallel接口通过回读状态寄存器查看。内部并行接口还可以直接采样SEU输出信息）

如果没有错误，继续回读下一帧。

如果单比特错误，动态重配当前帧后，继续回读下一帧。

如果双比特错误，用户自行选择下一步操作。比如可以动态重配当前帧，也可以重配等。

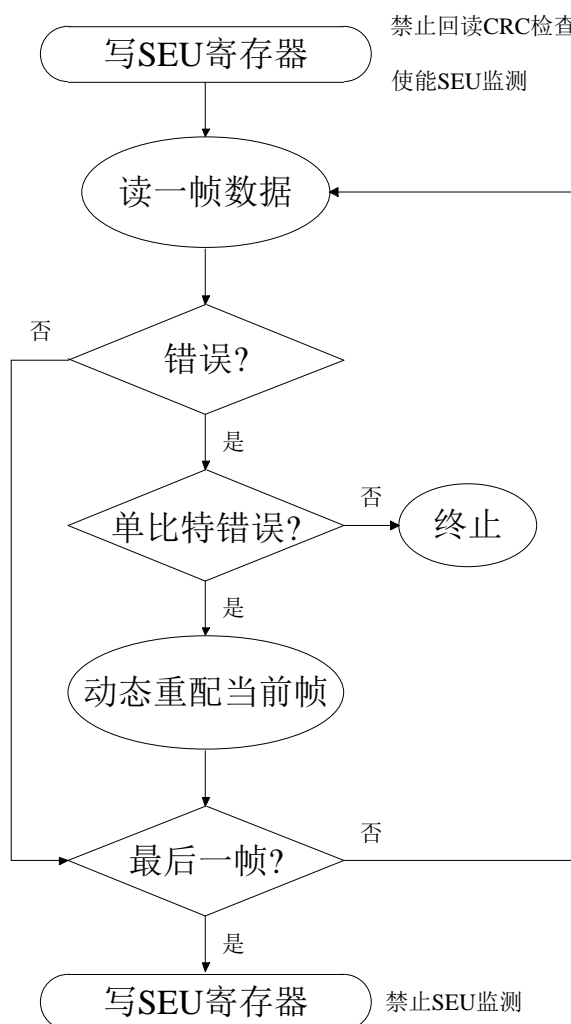


图 68 SEU 流程

具体使用方法见文档《AN020006\_Logos系列FPGA SEU应用指南》。另外，Logos 回读CRC功能可使用IP进行实现，详情参考文档《UG022002\_SEU\_IP》。

## 七、设计保护

### （一）位流加密

用户设计被竞争对手恶意抄袭是 FPGA 使用过程中的常见问题之一。为解决该问题，Logos 系列 PGL12G/22G/22GS/50G/50H 器件中内置了位流加密保护的功能，用于防止用户的设计被恶意克隆。PGL25G 不支持位流加密保护的功能。

#### 1. 不带位流加密保护的场景

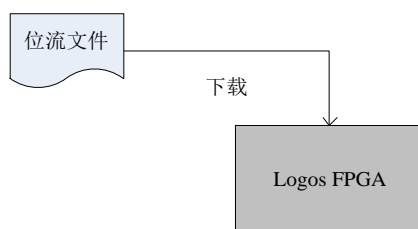


图 69 无位流加密场景

如上图所示，Logos 系列 FPGA 是基于 SRAM 的 FPGA，每次上电都需要从外部加载位流文件。用户的位流文件通常存储于 FPGA 外部的存储单元，例如 Flash 等。

恶意克隆者可以有各种方法获取这个位流文件。获取位流文件后，恶意克隆者把这个位流文件直接加载于他们自己采购的同款 FPGA，即可实现与用户完全相同的 FPGA 功能。

#### 2. 带位流加密保护的场景

为了有效保护用户的知识产权，Logos 系列 FPGA 提供了位流加密的功能。

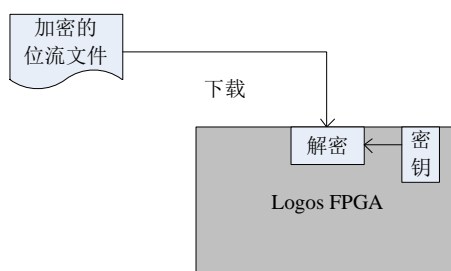


图 70 位流加密场景

如上图所示，位流文件本身是加密的，采用的加密算法是 AES-256。加密后的位流文件加载到 Logos 系列 FPGA 时，必须首先结合预先存储在芯片内部的密钥进行解密。这里，加密的位流文件的密钥和储存在 Logos FPGA 内部的密钥必须匹配，否则 Logos FPGA 无法正常进入工作状态。

Logos FPGA 内部的密钥是由用户通过 JTAG 提前写入的，它是一次可编程（OTP）的；也是非易失的，所以说该密钥信息一经编程会永久保留。此外，Logos FPGA 的密钥提供了读保护功能，使能读保护功能后写入的密钥无法回读。



使用了位流加密保护后,用户不再受恶意抄袭的困扰。恶意抄袭者即使拿到了位流文件(加密的),但是他不知道密钥是什么,所以无法将该位流文件加载到未包含对应密钥的 Logos FPGA 中。

### 3. 位流加密保护的使用流程

位流加密保护的主要使用流程包括:

- 1) 选定密钥, 注意用户需要妥善保护密钥;
- 2) 用户 PDS 生成位流文件时, 使用选定的密钥, 产生加密的位流文件;
- 3) 用户用 PDS 内置的 Fabric configuration, 把密钥烧写到 Logos 系列产品中。之后, 用户可以放心的使用加密的位流文件来加载经过 3) 处理后的 Logos FPGA。

生成加密位流文件(sbit)的步骤如下, 在 PDS 的 Generate Bitstream 选项点鼠标右键, 选择 Configure 项进行设置, 如下图所示。也可以通过菜单中的 Process 进行选择。

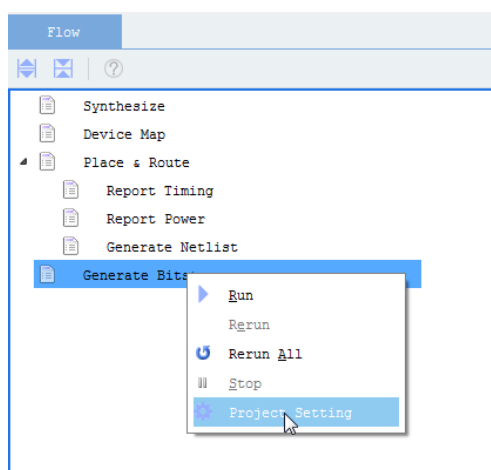


图 71 Configure 界面

弹出窗口如下图所示:

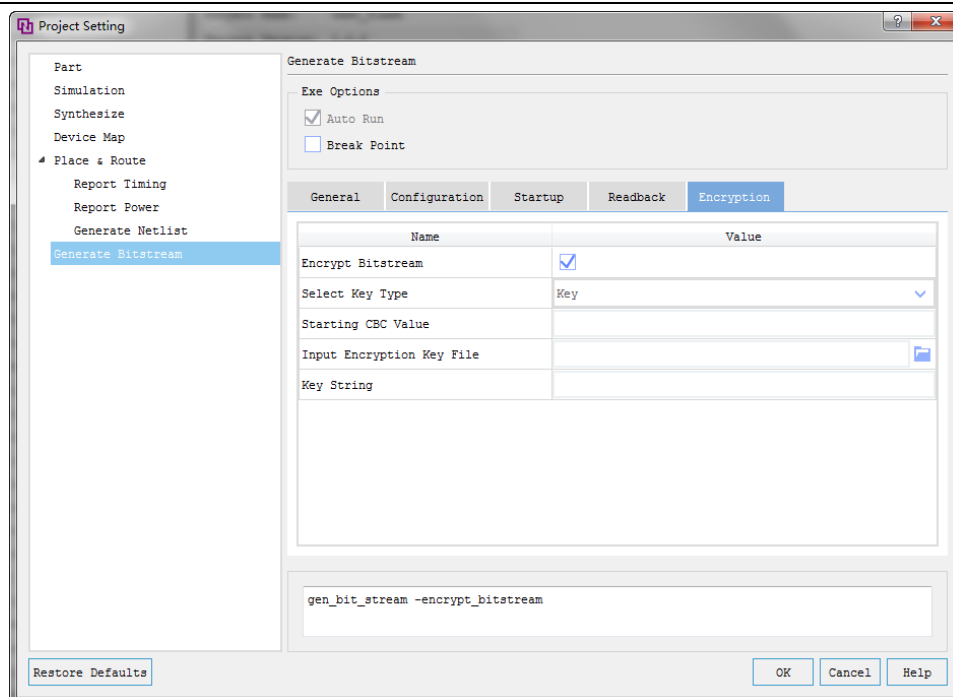


图 72 Configure 设置选项

其中，【Encrypt Bitstream】选择是否加密位流文件，在【General】的【Create Bit File】配置为 Yes 的前提下，勾选【Encrypt Bitstream】生成加密的 sbit 文件及 nky 文件（包含加密所需的 128 比特的初始 CBC 值和 256 比特的密钥）。用户可以编辑 CBC 和密钥字符串或选择 nky 文件，若用户不指定，则随机生成 nky 文件。

【Select Key Type】配置是否使用内部 KEY，默认值为 Key。选择 Temporary Key 时，每次下载位流前需先下载对应的 nky 文件，选择 Key，该操作仅需执行一次，然后直接下载位流即可，器件断电后密钥不丢失；

【Starting CBC Value】用户手动输入初始的 CBC 字符串；

【Input Encryption Key File】选择 nky 文件，如果在此选择了密钥文件，则软件选择以密钥文件进行加密，而不管用户在选项中是否输入了密钥字符串；

【Key String】用户手动输入密钥字符串；

设置完成后，点击 OK 退出配置窗口。再运行 Generate Bitstream 即生成加密的位流文件。

## （二）UID

可重构器件行业存在很大的一个利益安全危害即是设计的抄袭和非法阔产。那么为了避免此类非法行为，UID 即应运而生。每一个器件设备都有一个与之对应的唯一编号，该编号在器件出厂的时候已经唯一确定。用户可以通过 [UID 接口](#)（例化 GTP\_UDID）和 JTAG 接口读取，并且以自己特有的加密算法处理后将得到的结果并入编程数据流。每一次重载数据流后，设备进入用户模式，用户逻辑都会先读取该 UID 以用户独特的加密算法处理后与之前编程数据流中的结果相比对，若有不同。则设备无法正常工作。

64 位 UID 存储在 eFuse 中，在器件出厂时统一编程。每次设备上电，自动将 eFuse 中的 UID 读取到寄存器中，以使用户随时取出。

1. 端口列表

表 54 GTP\_UDID 端口列表

端口名称	方向	功能描述
DI	输入	串行数据输入
DO	输出	串行数据输出
SE	输入	数据移位使能
LOAD	输入	数据寄存器并行加载 UID CODE
CLK	输入	时钟

2. 参数列表

表 55 GTP\_UDID 参数列表

参数名称	参数类型	有效值	默认值	功能描述
UDID_CODE	Binary	0~64'hffffffff ffffffff;	0	芯片身份标识码

3. 接口时序

UID 长度为 64 位。

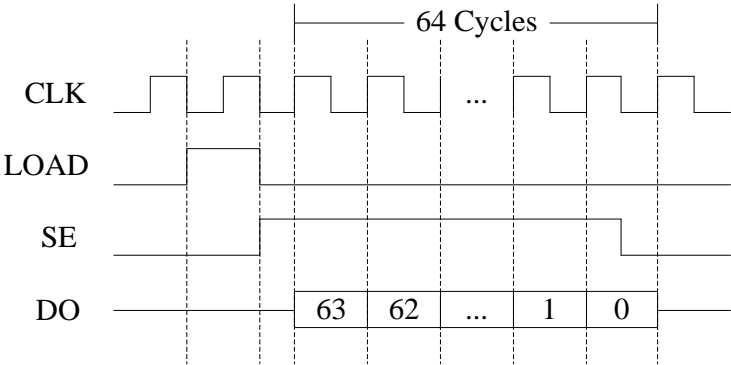


图 73 读 UID 时序

用户可以扩展 UID 的位宽。

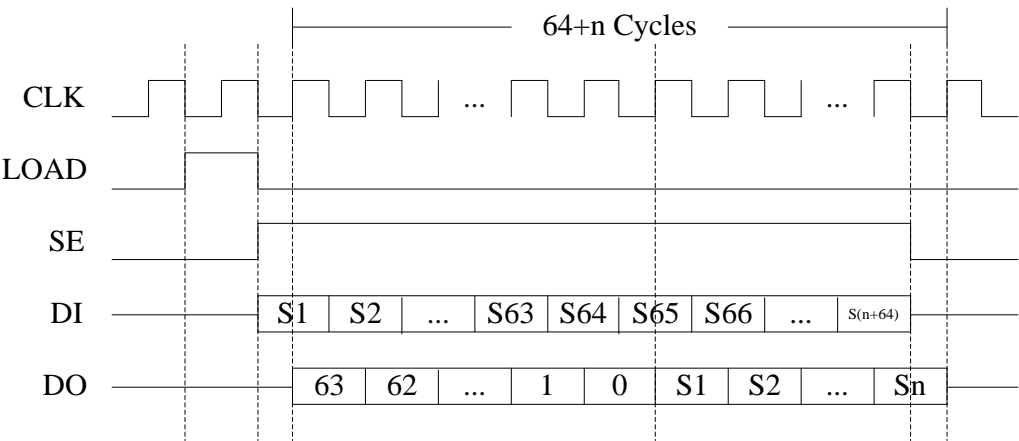


图 74 扩展 UID 位宽

```
GTP_UDID #(
    .UDID_CODE('b0000000000000000000000000000000000000000000000000000000000000000)
) <InstanceName> (
    .DO(),
    .CLK(),
    .DI(),
    .LOAD(),
    .SE()
);
```

## 八、 用户逻辑接口

### （一） 内部从并行接口

GTP\_IPAL\_E1 提供内部从并行接口，以及回读 CRC、SEU 接口。用户可以通过例化该 GTP 进行内部从并行接口操作，也可以使用回读 CRC、SEU 接口很方便地进行回读 CRC 和 SEU 操作。这里主要对内部从并行接口进行介绍，回读 CRC、SEU 的使用请参考文档《AN020006\_Logos 系列 FPGA SEU 应用指南》。

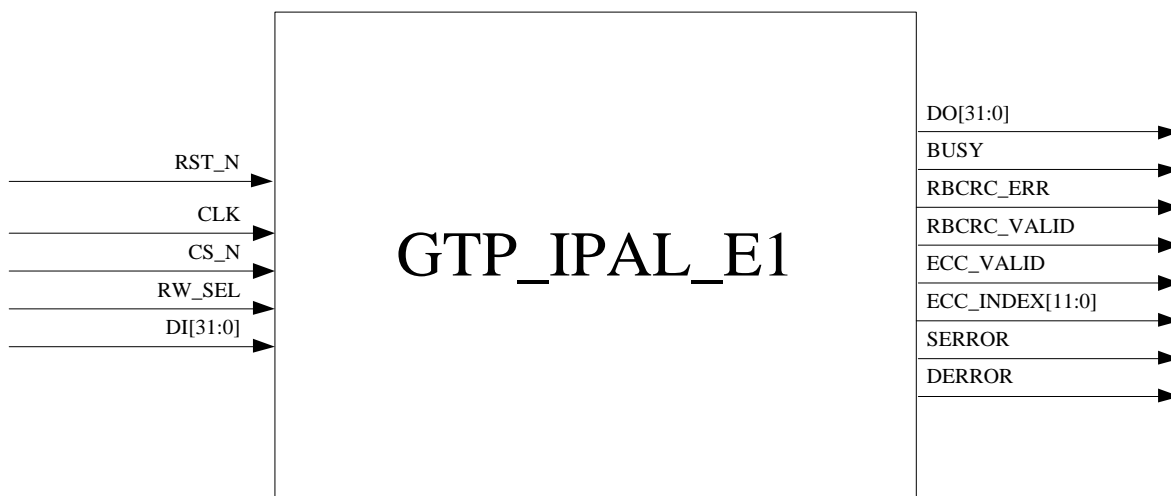


图 75 GTP\_IPAL\_E1 端口图

#### 1. 端口列表

表 56 GTP\_IPAL\_E1 端口列表

端口名称	I/O	位宽	描述
CLK	I	1	内部从并行接口时钟
RST_N	I	1	复位信号，低电平有效，仅用于仿真 <sup>(1)</sup>
内部从并行接口			
CS_N	I	1	片选信号，低电平有效
RW_SEL	I	1	读写选择：0写，1读；
DI	I	32	数据输入
DO	O	32	数据输出
BUSY	O	1	回读忙标志，1表示数据未准备好
回读 CRC 接口			
RBCRC_ERR	O	1	回读CRC错误标志，高有效。

RBCRC_VALID	O	1	回读CRC有效标志，高有效，持续1个时钟周期。
SEU 检测接口			
ECC_VALID	O	1	ECC有效标志，高有效。
ECC_INDEX	O	12	单比特错误的地址索引
SERROR	O	1	SEU检测的单比特错误标志
DERROR	O	1	SEU检测的双比特错误标志

注：（1）该信号仅用于仿真时使接口恢复初始状态，实际硬件上无此端口，对该端口操作即无必要也不会起任何作用。

2. 参数定义

表 57 GTP\_IPAL\_E2 参数列表

参数名称	参数类型	有效值	默认值	功能描述
IDCODE	Binary	0~32'hfffffff	32'haaaa5555	器件 IDCODE，仅用于仿真
DATA_WIDTH	String	"X8"、"X16"、"X32" <sup>(1)</sup>	"X8"	并行数据位宽
SIM_DEVICE	String	"PGL12G"、"PGL22G"、 "PGL25G"、"PGL50G"、 "PGL50H"	"PGL25G"	器件选择，仅用于仿真

注：（1）PGL25G、50G、50H只支持X8、X16位宽，不支持X32位宽。

3. 接口时序

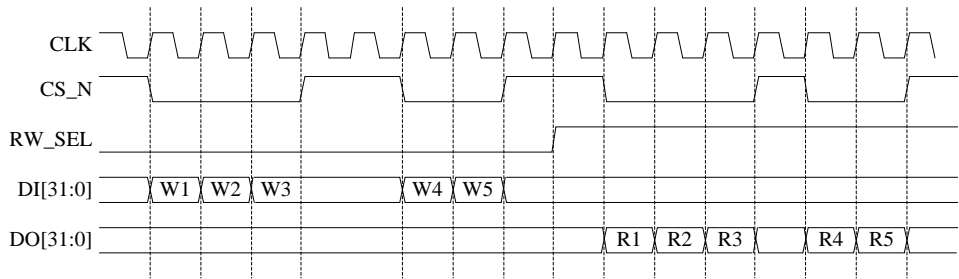


图 76 GTP\_IPAL\_E1 内部从并行接口读写接口时序

注:RW\_SEL信号跳变只能在CS\_N为高时进行。

4. 例化模板

```
GTP_IPAL_E1 #(
    .DATA_WIDTH("X8"),
    .IDCODE('b101010101010100101010101010101),
    .MEM_DEPTH(1616)
) <InstanceName> (
```

```

        .DO(),
        .ECC_INDEX(),
        .DI(),
        .BUSY(),
        .DERROR(),
        .ECC_VALID(),
        .RBCRC_ERR(),
        .RBCRC_VALID(),
        .SERROR(),
        .CLK(),
        .CS_N(),
        .RST_N(),
        .RW_SEL()
    );
    
```

## (二) UID 接口

### 1. 端口列表

表 58 GTP\_UDID 端口列表

端口名称	I/O	位宽	功能描述
DI	I	1	串行数据输入
DO	O	1	串行数据输出
SE	I	1	数据移位使能
LOAD	I	1	数据寄存器并行加载 UID CODE
CLK	I	1	时钟

### 2. 参数列表

表 59 GTP\_UDID 参数列表

参数名称	参数类型	有效值	默认值	功能描述
UDID_WIDTH	Integer	64	64	UID 长度，仅用于仿真 <sup>(1)</sup>
UDID_CODE	Binary	0~96'hffffffffffffffffffff	0	芯片身份标识码，仅用于仿真

注：（1）该接口模型通用于Pango Compact、Logos、Logos2系列器件，不同系列的UID长度不同，默认值统一为64，因此，为了保持仿真与上板一致，用于Logos器件时应保持为默认值64。

### 3. 接口时序

UID 长度为 64 位。



Timing diagram for the 64+ n Cycles mode. The diagram shows the relationship between CLK, LOAD, SE, DI, and DO signals over time. A horizontal bar at the top indicates a duration of "64+n Cycles". The DI signal is a sequence of data words: S1, S2, ..., S63, S64, S65, S66, ..., S(n+64). The DO signal is a sequence of data words: 63, 62, ..., 1, 0, S1, S2, ..., Sn. Vertical dashed lines mark the boundaries of the 64+n cycle period.

图 78 扩展 UID 位宽

```
GTP_UDID #(
    .UDID_CODE('b000000000000000000000000000000000000000000000000000000000000000')
) <InstanceName> (
    .DO(),
    .CLK(),
    .DI(),
    .LOAD(),
    .SE()
);
```



## 免责声明

### 版权声明

本文档版权归深圳市紫光同创电子有限公司所有，并保留一切权利。未经书面许可，任何公司和个人不得将此文档中的任何部分公开、转载或以其他方式披露、散发给第三方。否则，公司必将追究其法律责任。

### 免责声明

- 1、本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因本文档使用不当造成的直接或间接损失，本公司不承担任何法律责任。
- 2、本文档按现状提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。
- 3、公司保留任何时候在不事先声明的情况下对公司系列产品相关文档的修改权利。