

# Logos2 系列 FPGA 远程升级应用指南

(AN04002, V1.4)

(2024.11.26)

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	2020-11-26	初始版本
V1.1	2022-06-16	修改擦除 flash 超时计数器为 3 秒，并增加超时告警上报功能。 增加回读位流的 CRC 校验值寄存器，方便上位机查看。 更新升级流程，回读校验功能由默认校验改为可选。 修正偶现开关程序无法打开的问题。 更新寄存器地址及定义。 更新判断器件 ready 条件，由读 p_e_bit 改为读 wip_bit，增加适应性。 更新应用位流支持个数，由支持 1 个改为支持 1/2/3 个。 增加更新与启动位流流程分离功能，更新与启动可选择不同应用位流。 增加回读位流上传使能寄存器，回读位流默认传回上位机改为可选上传。 10、增加单独擦除/打开开关程序的功能。 11、增加 IPAL 模块支持数据位宽类型，默认 X8，并支持 X16/X32 两种位宽。
V1.2	2022-08-17	1、增加版本回退及看门狗设置说明。 2、增加生成合并位流时选择器件系列及位流类型的说明。
V1.3	2024-10-08	1、修正看门狗设置说明。 2、增加 CRC 计算公式说明。 3、支持器件列表增加 PG2L200H。
V1.4	2024-11-26	1、修复回读计算 CRC32 错误问题，更新代码。

## 术语与缩略语

Terms and Abbreviations 术语与缩略语	Full Spelling 英文全拼	Chinese Explanation 中文解释
CPLD	Complex Programmable Logic Device	复杂可编程逻辑器件
CRAM	Configuration Random Access Memory	配置存储器
JTAG	Joint Test Action Group	联合测试接口
CCS	Configuration Control System	配置控制系统
TAP	Test Access Port	测试访问端口
PDS	Pango Design Suite	紫光同创设计平台

## 目 录

文档版本修订记录 .....	1
术语与缩略语 .....	2
目 录 .....	3
图目录 .....	5
表目录 .....	6
1 概述 .....	7
1.1 介绍 .....	7
1.2 主要功能 .....	7
1.3 设计信息 .....	7
2 功能描述 .....	8
2.1 模块介绍 .....	8
2.1.1 串口模块(uart_top.v) .....	9
2.1.2 数据控制模块(data_ctrl.v) .....	9
2.1.3 spi flash 控制模块(spi_top.v) .....	10
2.1.4 IPAL 控制模块(ipal_ctrl.v) .....	10
2.1.5 顶层模块参数定义 .....	11
2.2 通用模块接口信号说明 .....	11
2.2.1 接口信号列表 .....	11
2.2.2 接口信号时序图 .....	13
3 参考设计 .....	16
3.1 寄存器说明 .....	16
3.1.1 读/写操作 .....	16
3.1.2 寄存器地址分配 .....	17
3.2 应用说明 .....	18
3.2.1 位流文件存储格式 .....	18
3.2.2 位流文件更新流程 .....	19
3.2.3 注意事项 .....	21
3.3 参考设计上板验证 .....	23
3.3.1 生成合并位流 .....	23
3.3.2 上板验证 .....	25
4 附录 .....	30
4.1 python 脚本 .....	30

4.2 python 脚本操作步骤 .....	30
免责声明 .....	31

应用示例 仅供参考

## 图目录

图 2-1 LOGOS2 系列 FPGA 器件远程升级应用示意图 .....	8
图 2-2 远程升级系统框图 .....	9
图 2-3 串口模块框图 .....	9
图 2-4 数据控制模块框图 .....	10
图 2-5 SPI FLASH 控制模块框图 .....	10
图 2-6 写位流时序图 .....	13
图 2-7 读位流时序图 .....	14
图 2-8 写开关时序图 .....	14
图 2-9 热启动时序图 .....	14
图 2-10 单独擦除开关时序图 .....	15
图 2-11 写位流数据时序图 .....	15
图 2-12 回读数据时序图 .....	16
图 3-1 位流文件在 FLASH 中的存储格式 .....	19
图 3-2 应用位流更新流程图 .....	21
图 3-3 版本回退及看门狗设置 .....	22
图 3-4 SFC 文件生成 .....	23
图 3-5 合并 1 个应用位流 .....	24
图 3-6 合并 2 个应用位流 .....	24
图 3-7 合并 3 个应用位流 .....	24
图 3-8 烧写合并位流文件 .....	25
图 3-9 读位流版本信息 .....	25
图 3-10 配置 CRC 寄存器 .....	26
图 3-11 发送写位流使能命令 .....	26
图 3-12 写应用位流完成 .....	27
图 3-13 读位流文件校验完成 .....	27
图 3-14 发送打开开关使能命令 .....	28
图 3-15 发送热启动使能命令 .....	28
图 3-16 热启动后读当前位流版本信息 .....	29

## 表目录

表 1-1 远程升级产品信息 .....	7
表 2-1 参数定义 .....	11
表 2-2 通用模块接口信号 .....	11
表 3-1 命令码 .....	16
表 3-2 寄存器地址分配表 .....	17
表 3-3 操作命令及返回值 .....	20

## 1 概述

### 1.1 介绍

本文档为深圳市紫光同创电子有限公司 Logos2 系列 FPGA 器件远程升级方案应用文档。本文档主要介绍了 Logos2 系列 FPGA 器件远程升级的主要功能、接口定义、接口时序、支持器件以及参考设计等。

### 1.2 主要功能

Logos2 系列 FPGA 器件远程升级的实质就是通过远程且不影响当前芯片工作状态的方式完成芯片的代码版本升级或回退，其功能包含利用串口在 IO 口状态保持不变的情况下，实现“flash 应用数据流升级、回读 flash 芯片中的应用位流、校验写入 flash 的应用位流是否正确”。

#### 注意

在芯片远程升级位流过程中，IO 口状态将保持不变（热启动 FPGA 除外）。

### 1.3 设计信息

表 1-1 远程升级产品信息

远程升级产品信息	
支持器件	PG2L200H PG2L100H PG2L50H PG2L25H
支持用户接口	串口
提供的设计文件	
远程升级参考设计	Verilog 文件 fdc 文件
开发工具支持	
设计工具	Pango Design Suite 2023.2-SP2

注：升级数据位流为简单 LED 闪灯示例。



## 2 功能描述

Logos2 系列 FPGA 器件支持 SPI Master 和 BPI Master 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 接口将位流编程到外部 Flash。

Logos2 系列 FPGA 器件通过同步字（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。用户可利用 JTAG，将合并位流(开关程序+跳转程序+黄金位流+应用位流)写入外部 flash 芯片，利用远程升级更新应用位流。Logos 2 系列 FPGA 器件支持 1~3 个应用位流，芯片上电后，配置控制系统（Configuration Control System, CCS）通过对应的开关程序的状态(打开或关闭)加载相应的位流文件。在更新应用位流的过程中，若出现掉电等异常情况，重新上电后，配置控制系统会自动加载黄金位流，确保 FPGA 有可用的版本，能正常启动。

Logos2 系列 FPGA 器件远程升级应用示意图如图 2-1 所示。

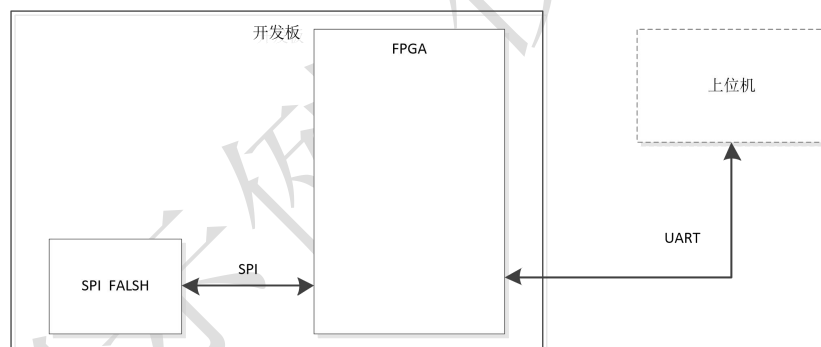


图 2-1 Logos2 系列 FPGA 器件远程升级应用示意图

### 2.1 模块介绍

Logos2 系列 FPGA 器件，远程升级系统框图如图 2-2 所示。右侧虚线框内的模块的主要功能为：与上位机通信、数据缓存、命令解析等。左侧实线框内为通用模块，主要控制读/写 flash 和热启动。若远程升级方案中通信接口换为以太网等其它接口，左侧实线框内模块可以通用<sup>1</sup>，右侧虚线框内模块需要根据通信协议适配修改。

上位机可通过串口读写远程升级相关配置寄存器和状态寄存器，下发各类控制命令。上

<sup>1</sup> 远程升级的流程和通用模块与前级模块数据交互方式不能改变。

上位机下发位流文件，串口模块(uart\_top.v)将串口协议数据转换后，送给后级数据控制模块(data\_ctrl.v)缓存。spi flash 控制模块(spi\_top.v)读取位流数据写入 flash 芯片，写入完成后发送完成指示信号给 IPAL 控制模块(ipal\_ctrl.v)。IPAL 控制模块将热启动命令写入 GTP\_IPAL\_E2 模块，进行热启动。重新启动后，配置控制系统（Configuration Control System, CCS）加载最新应用位流，完成远程升级。

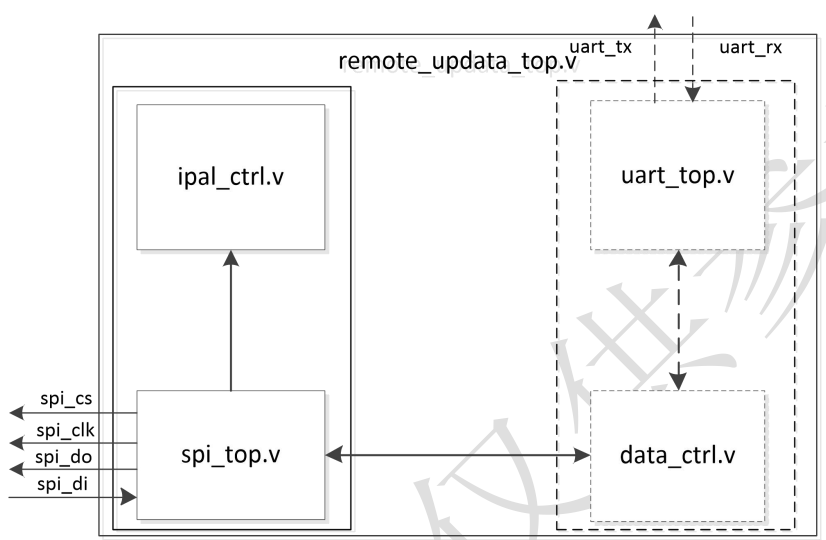


图 2-2 远程升级系统框图

### 2.1.1 串口模块(uart\_top.v)

串口模块主要负责与上位机之间的串口通信。默认波特率为 115200bps，不能动态修改。分频参数 CLK\_DIV 需要根据输入时钟频率修改，例如输入时钟 100Mhz 时为 12'd868，输入时钟 10Mhz 时为 12'd87。串口模块框图如图 2-3 所示。

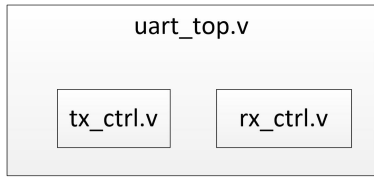


图 2-3 串口模块框图

### 2.1.2 数据控制模块(data\_ctrl.v)

数据控制模块主要有两个功能：解析上位机下发的命令及读写寄存器控制；缓存位流数据，并填充数据至 4KB 对齐，方便后续处理。

寄存器控制模块(uart\_reg\_ctrl.v)，主要接收上位机下发的命令，解析后下发给数据处理和 SPI 控制模块，及接收状态信息返回给上位机。

数据处理模块(data\_process.v)，主要接收位流数据，及根据状态机控制生成开关程序数据和填充数据写入 FIFO，供 SPI 控制模块读取。此模块的数据以 4KB 对齐，即生成的开关程序数据为 4KB，接收的位流数据不满 4KB 时需填充对齐。由于写 SPI FLASH 以页(256 字节)为单位，所以 256 字节为一个数据包写入 FIFO。

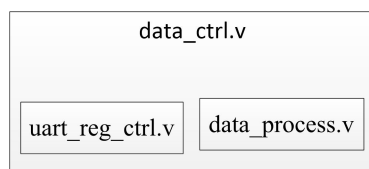


图 2-4 数据控制模块框图

### 2.1.3 spi flash 控制模块(spi\_top.v)

spi flash 控制模块完成 spi 总线控制，将位流数据写入外部 flash 芯片。spi\_driver.v 模块负责完成单条指令(读配置寄存器、擦除扇区、页编程等)的执行。spi\_top.v 模块将读/写位流命令分解成多个单条指令，写入 FIFO，再由 spi\_driver.v 模块依次执行。

#### 注意：

本案例验证所用的 spi flash 芯片为镁光的 N25Q032。具体指令可查看 N25Q032 的数据手册。若实际使用 spi flash 芯片指令与此芯片指令不兼容，则需要修改代码中的指令定义参数进行调整。

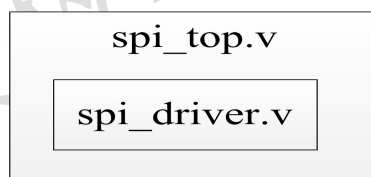


图 2-5 spi flash 控制模块框图

### 2.1.4 IPAL 控制模块(ipal\_ctrl.v)

IPAL 控制模块主要实现 Logos2 系列 FPGA 器件的热启动。位流数据写入完成后，IPAL 控制模块会收到一个指示信号，若 hotreset\_en 为 1，则立即热启动；若 hotreset\_en 为 0，则不会立即热启动，可以通过上位机手动将此寄存器写为 1，然后热启动。寄存器 hotreset\_en 的说明见后面的寄存器说明章节。热启动时，会根据 CRC 验证使能、验证结果、应用位流序号等，决定启动哪个位流。

### 2.1.5 顶层模块参数定义

为了适应不同设计需求，顶层模块中定义了一些参数。客户根据需求修改这些参数，若设置与实际设计不匹配，可能导致远程升级失败等问题。参数相关描述见 表 2-1。

表 2-1 参数定义

参数名称	描述
FPGA_VESION	FPGA 版本信息，用户可自定义，默认为年、月、日、时、分。
DEVICE	器件型号，用于判断位流占用 flash 空间大小。定义需与实际器件匹配，否则会导致功能异常。若后续推出新器件，用户可扩展。
USER_BITSTREAM_CNT	应用位流个数，可支持 1/2/3 个应用位流。定义需与实际情况匹配，否则会导致功能异常。
USER_BITSTREAM1_ADDR	应用位流 1 的起始地址，在黄金位流之后，且 4KB 对齐。
USER_BITSTREAM2_ADDR	应用位流 2 的起始地址，在应用位流 1 之后，且 4KB 对齐。
USER_BITSTREAM3_ADDR	应用位流 3 的起始地址，在应用位流 2 之后，且 4KB 对齐。

## 2.2 通用模块接口信号说明

### 2.2.1 接口信号列表

通用模块(spi\_top.v 和 ipal\_ctrl.v)与前级模块(data\_ctrl.v)的接口信号主要有使能控制信号、反馈指示信号、读/写数据交互信号。若使用其它接口与上位机通信，需要修改前级模块适配对应的通信协议。前级模块需要根据流程和反馈指示信号，对使能信号进行控制。写数据接口需要使用包长 256 字节的包 FIFO 存储数据，回读数据接口使用普通 FIFO 缓存数据。

表 2-2 通用模块接口信号

信号名称	方向	位宽	描述
时钟、复位信号(spi_top.v 和 ipal_ctrl.v)			
sys_clk	input	1	系统时钟。
sys_rst_n	input	1	系统复位，低有效。
外部管脚信号(spi_top.v)			
spi_cs	output	1	spi flash 芯片片选管脚。
spi_clk	output	1	spi flash 芯片时钟管脚。
spi_dq1	input	1	spi flash 芯片数据输入管脚。
spi_dq0	output	1	spi flash 芯片数据/指令输出管脚。
使能控制信号(spi_top.v)			
flash_wr_en	input	1	写位流数据使能，上升沿有效。
flash_rd_en	input	1	读位流数据使能，上升沿有效。
bitstream_wr_num	input	2	写位流序号，用于指定更新的应用位流。可支持 1/2/3 号应用位流，且不超过参数 USER_BITSTREAM_CNT。

信号名称	方向	位宽	描述
bitstream_rd_num	input	2	读位流序号，用于指定读取的应用位流。可支持 1/2/3 号应用位流，且不超过参数 USER_BITSTREAM_CNT。
bitstream_up2cpu_en	input	1	位流上传上位机使能，高有效。使能后，回读校验时上传位流。
crc_check_en	input	1	CRC32 校验使能，高有效。若不使能，则不进行回读校验。
clear_sw_en	input	1	单独擦除开关使能，上升沿有效。
bs_crc32_ok	input	2	[1]:为 1 则表示校验结果有效 [0]:校验结果，1'b0:校验正确，1'b1:校验错误。
write_sw_code_en	input	1	写开关使能，上升沿有效。
<b>反馈指示信号(spi_top.v)</b>			
bs_readback_crc	output	32	读位流校验 CRC 结果。
bs_readback_crc_valid	output	1	为 1 表示读位流校验 CRC 结果有效。
clear_sw_done	output	1	单独擦除开关完成指示，高有效。
clear_bs_done	output	1	擦除应用位流完成指示，高有效。
bitstream_wr_done	output	1	写位流文件完成，高有效。
bitstream_rd_done	output	1	读位流文件完成，高有效。
open_sw_code_done	output	1	写开关程序完成，高有效。
time_out_reg	output	1	擦除 flash 超时指示，高有效。
<b>回读数据接口(spi_top.v)</b>			
flash_rd_data	output	8	读位流数据。
flash_rd_valid	output	1	读位流数据有效。
flash_rd_data_fifo_afull	input	1	读位流数据缓存 FIFO 将满。
<b>写 flash 数据接口(前级数据缓存需要用包 FIFO) (spi_top.v)</b>			
bitstream_fifo_rd_req	output	1	写入 flash 位流文件缓存 FIFO 读请求。
bitstream_data	input	8	写入 flash 位流文件缓存 FIFO 读出数据。
bitstream_valid	input	1	写入 flash 位流文件缓存 FIFO 读出数据有效。
bitstream_eop	input	1	写入 flash 位流文件缓存 FIFO 数据包尾标识。每个数据包 256 字节(1 个 page)，方便后续处理。
bitstream_fifo_rd_rdy	input	1	写入 flash 位流文件缓存 FIFO 非空。
<b>Ipal 控制模块接口(ipal_ctrl.v)</b>			
hotreset_en	input	1	热启动使能，上升沿有效。
open_sw_num	input	2	热启动应用位流序号，可支持 1/2/3 号应用位流，且不超过参数 USER_BITSTREAM_CNT。
crc_check_en	input	1	CRC32 校验使能，高有效。若不使能，则不判断 CRC 校验结果。
bs_crc32_ok	input	2	[1]:为 1 则表示校验结果有效 [0]:校验结果，1'b0:校验正确，1'b1:校验错误。
open_sw_code_done	input	1	写开关程序完成，高有效。
ipal_busy	output	1	Ipal 忙指示信号，送到寄存器模块，防止本模块被综合工具优化。

除了列表中的信号，spi\_top.v 模块还有 flash\_flag\_status 等信号。为读/写 FLASH 芯片配置寄存器及状态指示信号，如需要使用，请参考 FLASH 芯片数据手册。若不使用，输入端

口接 0，输出端口悬空即可。

## 2.2.2 接口信号时序图

控制信号与指示信号主要跟远程升级的流程有关，回读数据接口和写数据接口主要涉及与前级模块的缓存 FIFO 数据交互。数据交互与控制流程的时序相对独立，所以分开说明。

结合 3.2.2 小节中位流文件更新流程，分步骤介绍相关信号时序图。

### 1. 写位流

如图 2-6 所示，使能写位流时，同时需要 bitstream\_wr\_num 有效。然后，等待擦除开关程序和应用位流完成。Spi\_top.v 模块返回擦除完成指示后，则可以开始发送应用位流数据(写 flash 数据接口)，接收数据完成后返回写位流完成指示。

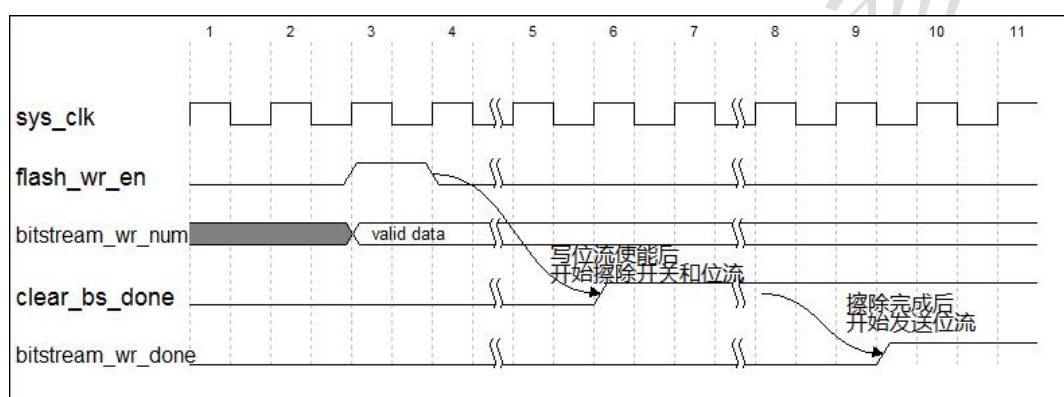


图 2-6 写位流时序图

### 2. 读位流

如图 2-7 所示，使能读位流时，同时需要 bitstream\_rd\_num 有效。读位流完成，Spi\_top.v 模块返回完成指示，同时回读的位流也完成了 CRC 计算并返回最终结果。前级模块收到 CRC 值后，根据校验使能配置(crc\_check\_en)，与上位机配置的 CRC 值进行比较，并得到校验结果(bs\_crc32\_ok)。若使能了上传位流(bitstream\_up2cpu\_en)，应用位流数据通过回读数据接口返回。

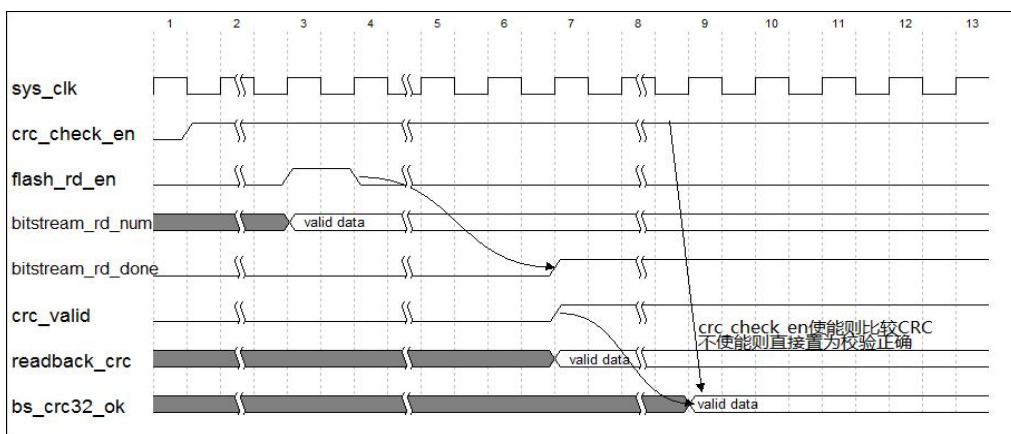


图 2-7 读位流时序图

### 3. 写开关

如图 2-8 所示，使能写开关，打开开关程序后返回完成指示。

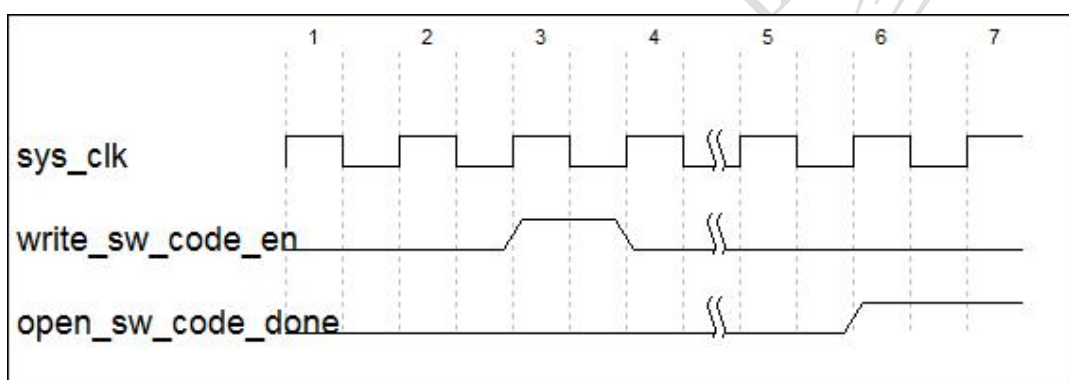


图 2-8 写开关时序图

### 4. 热启动

如图 2-9 所示，若使能了 CRC 校验(crc\_check\_en)，则会根据 CRC 校验结果(bs\_crc32\_ok)选择启动的位流。若不使能 CRC 校验，则只会选择启动 open\_sw\_num 对应的位流。前级模块写开关使能时，同时会配置 open\_sw\_num。待 spi\_top.v 模块返回打开开关完成指示后，才能发送热启动使能，否则不能启动成功。

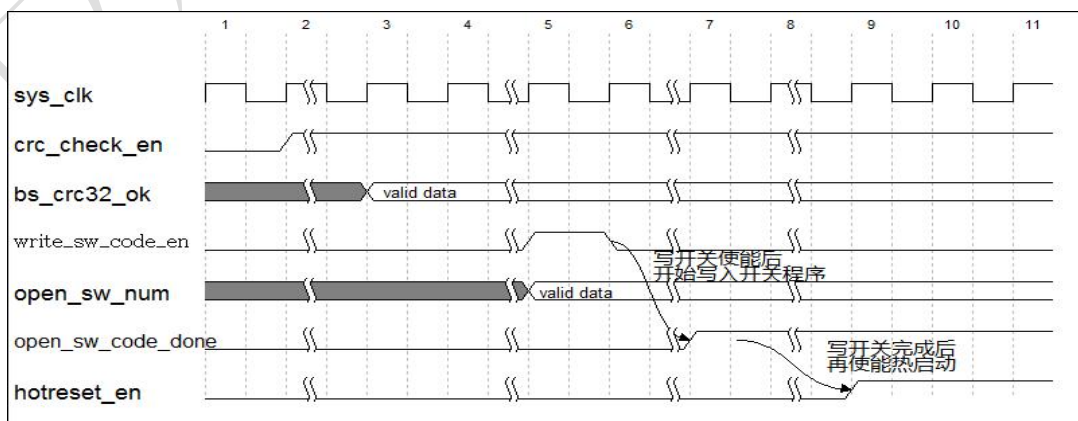


图 2-9 热启动时序图



若不更新，只切换启动的应用位流，则步骤如下：

5. 单独擦除开关

如图 2-10 所示，发送单独擦除开关使能，spi\_top.v 擦除完成后则会返回完成指示。

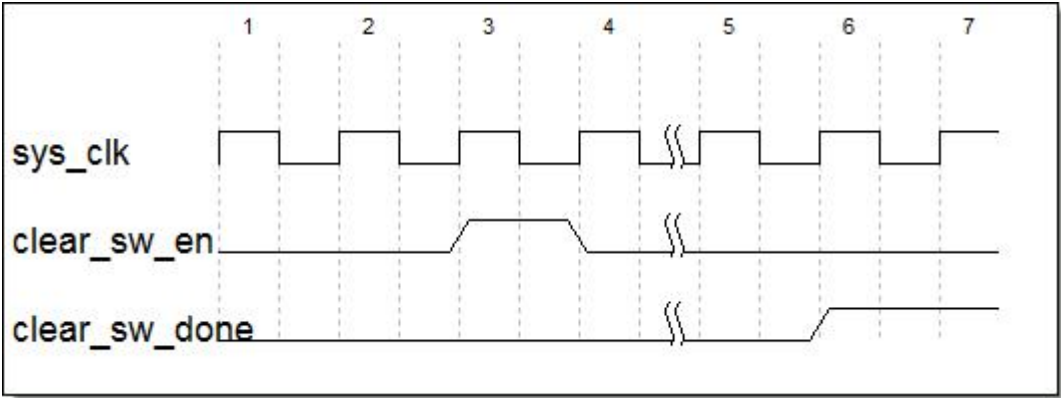


图 2-10 单独擦除开关时序图

6. 写开关

与步骤 3 相同。

7. 热启动

与步骤 4 相同。

写位流数据接口时序图如图 2-11 所示。

注意

前级 FIFO 需要满足两点条件：前级 FIFO 为包 FIFO，包长 256 字节；读请求信号与有效数据对齐，即读请求(bitstream\_fifo\_rd\_req)当拍出数据，无延时。

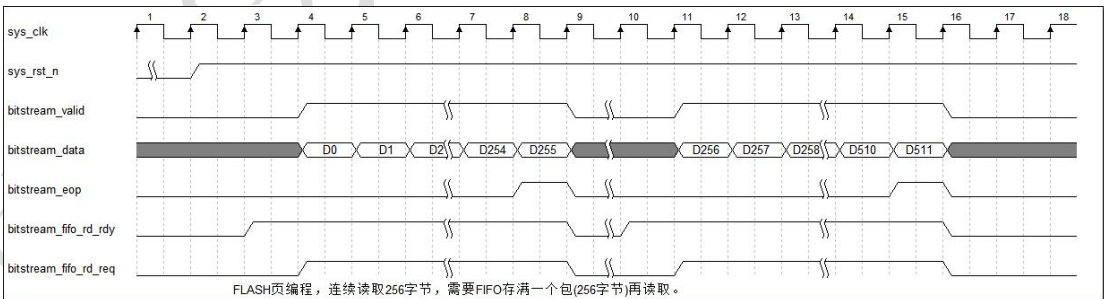


图 2-11 写位流数据时序图

回读位流数据的时序图如图 2-12 所示。

根据前级模块反压控制读取 FLASH 中的位流数据。为了方便控制并兼顾效率，每次读一页(256 字节)数据。每次读出的数据不管前级 FIFO 是否反压都会连续送出。若读完数据还有反压，则暂停读取，所以前级 FIFO 在满之前至少预留 256 字节的空间开



始反压。

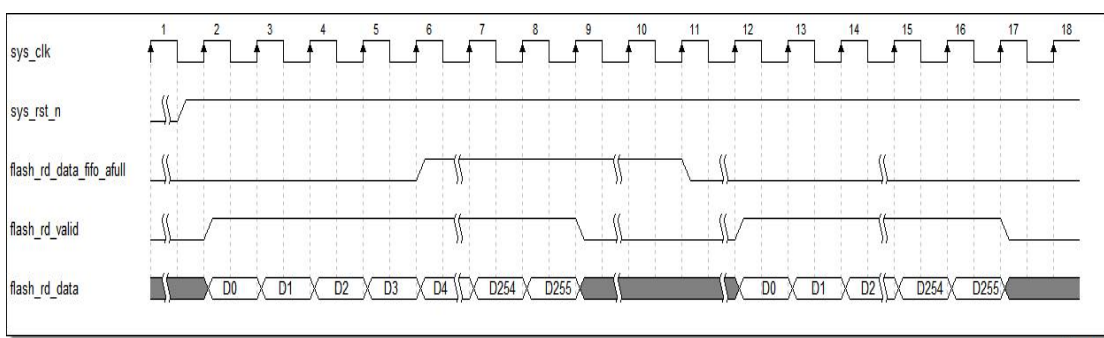


图 2-12 回读数据时序图

### 3 参考设计

#### 3.1 寄存器说明<sup>2</sup>

本参考设计选用紫光同创公司的开发板 P04I50RD01\_A0 为主控设备，在 PC 上以串口助手工具 UartAssist 作为上位机，完成远程升级功能的调试。由于上位机与开发板之间用串口通信的，没有区分寄存器地址与数据，所以需要自定义一套命令码。命令码如表 3-1 所示：

表 3-1 命令码

上位机下发命令码	上传上位机命令码	写数据流完成命令码	读地址命令码	写地址命令码
32'he7e7_e7e7	8'h55	32'h7e7e_7e7e	{1'b1,addr[6:0]}	{1'b0,addr[6:0]}

##### 3.1.1 读/写操作

读/写操作都是上位机下发的命令，读/写时都是上位机下发 32'he7e7\_e7e7+地址命令码。地址命令码为一个字节，低 7 位为地址，以最高位区分读/写，1'b1 表示读，1'b0 表示写。示例：读寄存器 1，32'he7e7\_e7e7+8'h81；写寄存器 1，32'he7e7\_e7e7+8'h01+写入的数据(1 字节)。

读寄存器命令下发，FPGA 会返回对应的寄存器值。擦除完成、写位流文件完成和 CRC 校验完成的信息，不用读操作，FPGA 会主动上传，格式相同。返回数据为 1 至多个字节，根据寄存器而定。写数据位流命令在后面更新流程的章节介绍。

<sup>2</sup> 本案例中上位机与 FPGA 之间通过串口通信，寄存器配置方式与命令格式均为适应串口定制。若用其它则需要调整。

### 3.1.2 寄存器地址分配

主要寄存器和命令，如表 3-2 所示。

表 3-2 寄存器地址分配表

名称	地址	读/写	功能描述
fpga_version	7'h0	R	版本信息，6 字节，自定义，例：48'h2020_0101_1230。
crc32_cfg	7'h1	W/R	上位机计算位流文件 CRC，然后配置到寄存器，4 字节。
test_reg	7'h2	W/R	测试寄存器
crc32_error_ind	7'h3	R	读位流 CRC 校验结果与 crc32_cfg 不相等标志。1'b1:不相等。
hotreset_en	7'h4	W/R	hotreset_en [0]: 热启动使能，高有效。若有效，更新位流完成后则热启动。
wr_bs_status	7'h5	R	wr_bs_status[0]: 擦除完成标识，高有效； wr_bs_status[1]: 单独擦除开关完成标识，高有效； wr_bs_status[2]: 单独打开完成标识，高有效； wr_bs_status[3]: 擦除超时，高有效； wr_bs_status[4]: 写位流文件完成标识，高有效。
crc_check_en	7'h6	W/R	crc_check_en[0]: CRC 校验使能，高有效。 1'b1:回读位流时进行 CRC 校验；1'b0:回读位流时不进行 CRC 校验。 注意：不使能时可以不要回读的步骤。
user_bitstream_cnt	7'h7	R	当前逻辑版本支持应用位流个数。由 USER_BITSTREAM_CNT 参数确定，有效参数值有 2'd1、2'd2、2'd3。
bs_readback_crc	7'h8	R	回读位流 CRC 校验结果，32 位。低地址对应低字节。
	7'h9		
	7'ha		
	7'hb		
bitstream_up2cpu_en	7'hc	W/R	bitstream_up2cpu_en [0]: 回读位流上传使能。 1'b1, 回读时，将位流传回上位机。 1'b0, 回读时，位流不传回上位机。
bitstream_num	7'hd	R	bitstream_num[1:0]:读应用位流号；bitstream_num[3:2]:写应用位流号。 有效值有 2'd1、2'd2、2'd3，且不超过 user_bitstream_cnt。
open_clear_sw_en	7'he	W/R	open_clear_sw_en [1:0]: 打开开关程序的位流序号，可选 1/2/3，且不超过 user_bitstream_cnt。 open_clear_sw_en [6]: 打开开关程序使能。 open_clear_sw_en [4]: 单独擦除开关程序使能。 打开开关时，配置 0x41/0x42/0x43。单独擦除开关时，配置 0x10。
wr_user_bs_en	7'h11	W	更新应用位流文件命令。wr_user_bs_en[1:0]对应 bitstream_wr_num，更新的应用位流序号，且不超过 user_bitstream_cnt。若 user_bitstream_cnt 为 2'd2，可更新 1 号或 2 号位流，写入 7'h11 或 7'h12。
	7'h12		
	7'h13		
rd_user_bs_en	7'h51	W	读应用位流文件命令。rd_user_bs_en [1:0]对应 bitstream_rd_num，读回的应用位流序号，且不超过 user_bitstream_cnt。
	7'h52		
	7'h53		

版本信息寄存器 fpga\_version 和 CRC 寄存器 crc32\_cfg 分别有 6 个字节和 4 个字节，读

这两个寄存器时会一次返回全部字节,写 `crc32_cfg` 时也是一次配置全部字节(`e7 e7 e7 e7 +01+xx xx xx xx`)。其它寄存器都只有一个字节,读/写只有一个字节内容。

返回数据格式为: `8'h55+地址+返回数据`。例:读版本信息收到数据 `55 00 20 20 01 01 12 30`,其中,55 为上传命令码,00 为地址,20 20 01 01 12 30 为版本信息寄存器的值。

## 3.2 应用说明

### 3.2.1 位流文件存储格式

位流文件在 flash 芯片内的存储格式如图 3-1 所示。合并位流文件在 flash 中是以 4KB(一个 Subsector)划分的。第一个 4KB 为开关程序,为 1023 个 `32'hffff_ffff+1` 个同步码(`32'h0133_2d94`)。第二个 4KB 是应用位流跳转程序,详细内容见《UG040005 Logos2 系列 FPGA 配置(configuration)用户指南》。每个应用位流都有一个开关程序和一个跳转程序。黄金位流必须在最后一个跳转程序之后,不满 4KB 的部分填充 `32'hffff_ffff`。应用位流在黄金位流之后,不满 4KB 的部分填充 `32'hffff_ffff`。黄金位流和应用位流的起始位置可在生成合并位流文件时设置,但黄金位流必须在应用位流前,且必须保证各位流之间不会出现地址重叠。

以 PG2L50H 器件包含 2 个应用的合并位流为例,说明各位流的起始地址如何计算:

8. PG2L50H 器件的位流大小为 2,113,704 字节,约 2064.164KB,由于在 FLASH 中存储是以 4KB 对齐的,所以占用 517 个 4KB。
9. 开关程序和跳转程序各占用 4KB,两个应用位流则总共需要 16KB 的空间。所以黄金位流的起始地址默认为 `0x4000(4x4KB,16KB)`。
10. 应用位流 1 的起始地址,默认为 `0x209000((4+517)x4KB,2084KB)`。
11. 应用位流 2 的起始地址,默认为 `0x40e000((4+517+517)x4KB,4152KB)`。
12. 各位流起始地址也可以设置为在默认地址之后 4KB 对齐的地址。如黄金位流起始地址设置为 `0x5000(20KB)`,则应用位流 1 的默认地址变为 `0x20a000((4+1+517)x4KB,2088KB)`。应用位流 2 同理。

包含一个应用位流时,开关程序默认打开,即默认启动应用位流。包含多个应用位流时,默认打开开关程序 1,其它开关程序关闭,即默认启动第 1 个应用位流。

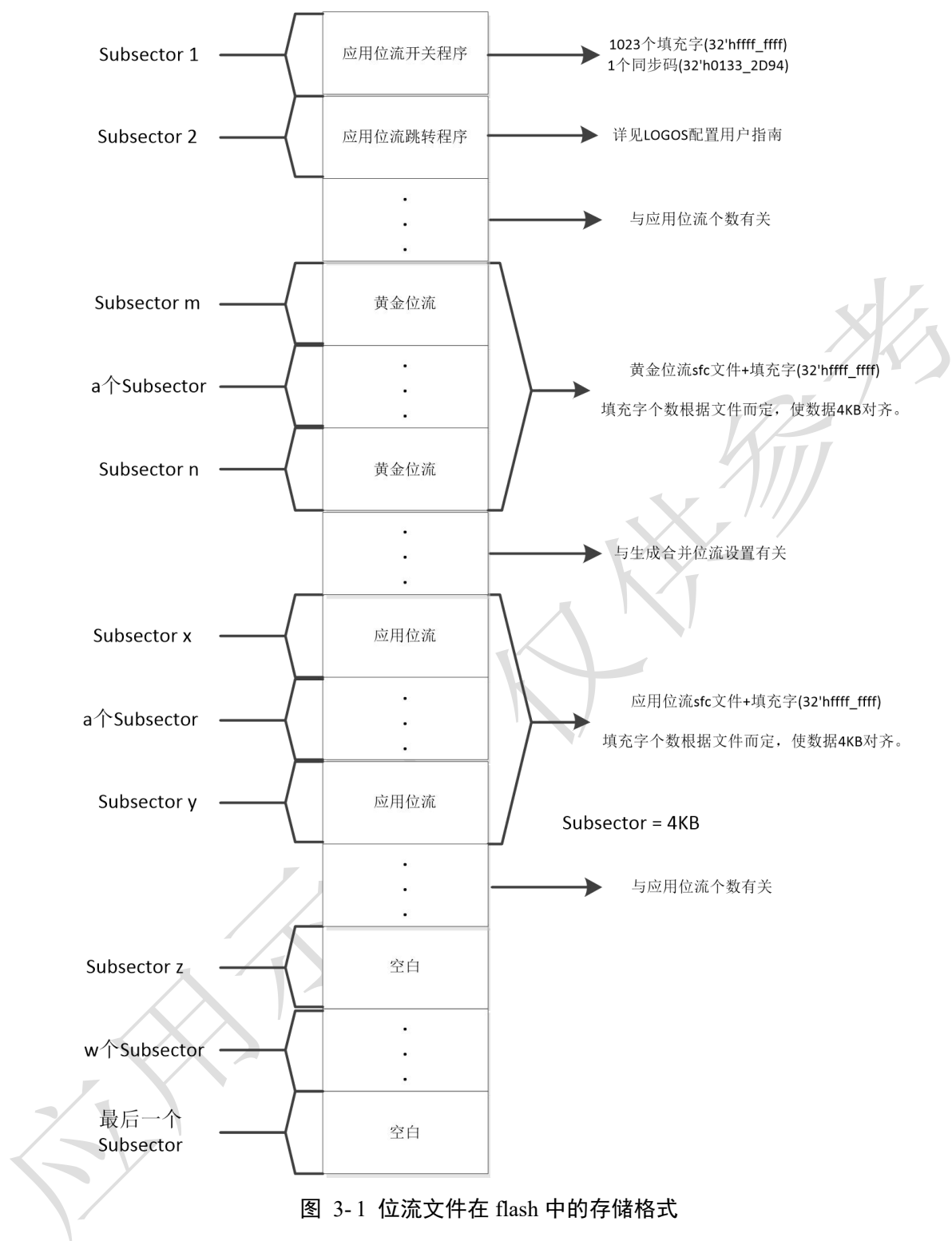


图 3-1 位流文件在 flash 中的存储格式

### 3.2.2 位流文件更新流程

远程升级时，更新位流的流程如图 3-2 所示。更新的过程，需要上位机与 FPGA 配合完成，其中[写位流使能](#)、[读位流使能](#)、[单独擦除开关使能](#)、[打开开关使能](#)、[热启动使能](#)由上位机下发命令，其它步骤由 FPGA 完成。升级过程中的命令及返回值如表 3-3 所示。

表 3-3 操作命令及返回值

名称	命令	返回完成标志
写位流 1 使能	e7 e7 e7 e7 11	55 05 01(擦除完成)
写位流 2 使能	e7 e7 e7 e7 12	55 05 10(写位流完成)
写位流 3 使能	e7 e7 e7 e7 13	55 05 08(擦除超时)
读位流 1 使能	e7 e7 e7 e7 51	55 03 01(校验错误) 55 03 00(校验正确)
读位流 2 使能	e7 e7 e7 e7 52	
读位流 3 使能	e7 e7 e7 e7 53	
单独擦除开关使能	e7 e7 e7 e7 0e 10	55 05 02(单独擦除完成) 55 05 08(擦除超时)
打开开关 1 使能	e7 e7 e7 e7 0e 41	55 05 04(打开开关完成)
打开开关 2 使能	e7 e7 e7 e7 0e 42	
打开开关 3 使能	e7 e7 e7 e7 0e 43	
热启动使能	e7 e7 e7 e7 04 01	无返回

升级的具体操作步骤如下：

1. 上位机下发写位流使能命令，等待 FPGA 擦除开关程序和应用位流。擦除完成 FPGA 发送给上位机完成标志( 55 05 01)。
2. 上位机收到擦除完成标志后，发送位流文件和位流结束标志(7e 7e 7e 7e)。上位机收到 (55 05 10)则表示写应用位流完成，可进行下一步操作。可以读位流进行校验，也可以重复前两步，再次写位流文件。
3. 上位机下发读位流使能，读取位流进行校验。校验完成后 FPGA 上报校验结果(55 03 01/00)。(55 03 01)表示校验结果为错误，(55 03 00)表示校验结果为正确。
4. 上位机发送打开开关使能命令。打开完成后，FPGA 发送给上位机完成标志(55 05 04)。
5. 上位机下发热启动使能，加载新的应用位流。若上步的校验结果为错误，则加载黄金位流。
6. 若不更新位流，只切换应用位流启动，其操作步骤如下：
7. 上位机下发单独擦除开关使能命令，擦除所有开关程序。擦除完成 FPGA 发送给上位机完成标志( 55 05 02)。
8. 上位机发送打开开关使能命令。打开完成后，FPGA 发送给上位机完成标志(55 05 04)。
9. 上位机下发热启动使能，加载新的应用位流。为了简化切换位流启动的流程，可关闭校验使能。

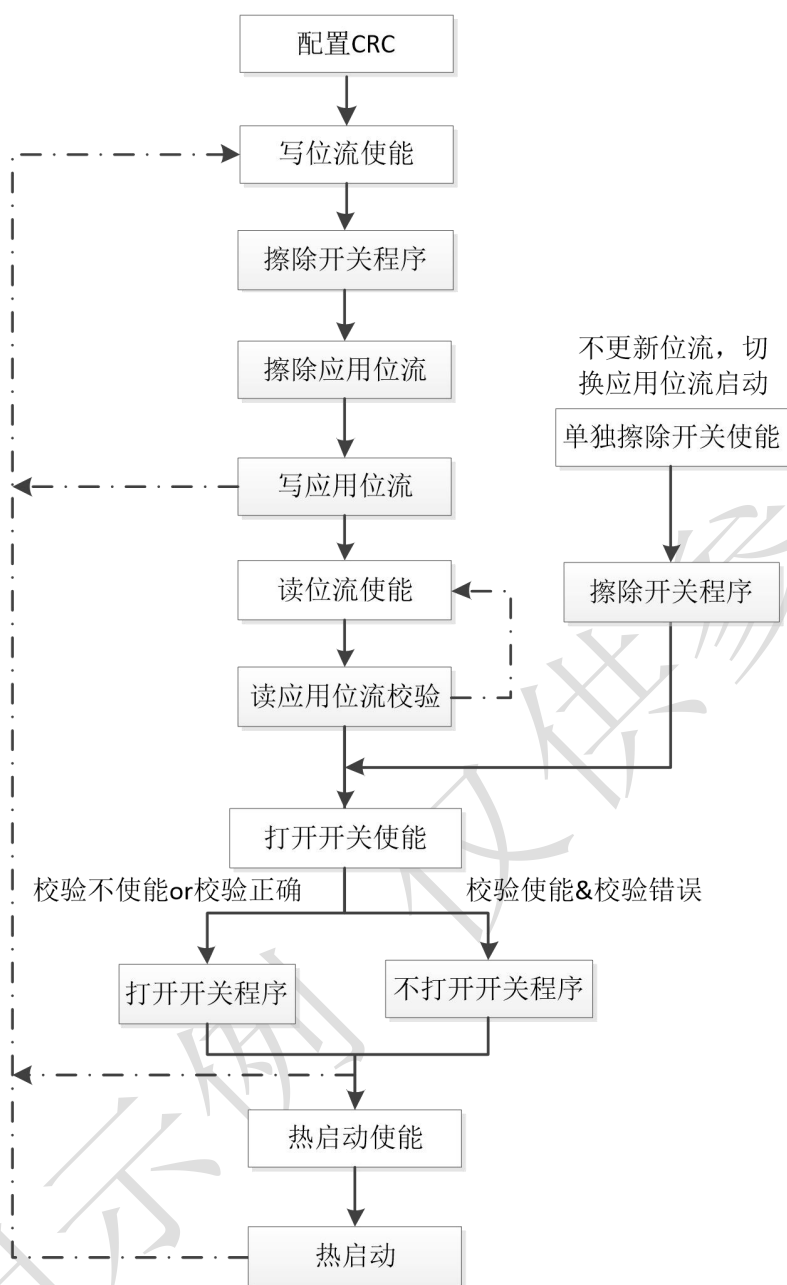


图 3-2 应用位流更新流程图

### 3.2.3 注意事项

1. 2020.2.SP2 之前的版本有去同步码，之后的版本没有去同步码。目前的代码是按没有去同步码的方式处理的，请生成合并位流时用 2020.2.SP2 或之后的版本。
2. 擦除位流所需时间与 flash 芯片性能及位流大小有关，详细说明见 flash 芯片数据手册。例如，镁光的 M25Q256 擦除 PG2L50H 位流所需时间大约为 10 秒。
3. DEMO 工程基于 P04I50RD01\_A0 开发板验证调试，SPI CLK 为 10Mhz。若在实际应用中出现读写错误，需要先确认是否 SPI CLK 频率过高。若 PCB 走线较近，为了减少升级时间，也可适当提高 SPI\_CLK 的时钟频率。
4. 在远程升级的过程中，每一步骤都必须完成后才能进行下一步操作，否则会出错，需

要重新复位或上电恢复。

5. 某些情况下，如串口助手或上位机系统原因，直接读取较大的 sfc 文件，会出现数据丢失或发送数据量限制。则需要利用脚本转换位流数据再下发。脚本内容及操作方法见附录。
6. 参数定义必须与实际设计匹配。如参数“DEVICE”定义与实际器件不一致，则会导致代码内判断位流大小错误，状态机卡死，无法完成升级功能。其它参数同理。
7. 在发送写位流使能、读位流使能、打开开关使能命令时，位流序号不能超过合并位流中的位流个数。例：参数定义 USER\_BITSTREAM\_CNT=2'd2，则不能发送“写位流 3 使能”、“读位流 3 使能”、“打开开关 3 使能”命令。
8. 上位机计算应用位流 CRC32 值时，请确认位宽、比特序等参数及计算公式与代码中一致，否则会验证结果错误。DEMO 工程中，计算 CRC 的参数如下：初始值为 32'hffff\_ffff，比特序为“NORMAL”，CRC 类型为 CRC\_32(以太网校验)，输出数据位宽为 8 位。
9. 在生成应用位流前，需要使能版本回退和看门狗，并设置看门狗初始值。看门狗初始值根据位流加载时间(具体计算参见配置 UG)确定。需要保证能加载完成，否则会功能异常；也不宜设置太大，避免回退等待时间太久。例如：加载需要 30 秒，看门狗可设置为 35 秒。看门狗计数器时钟为配置时钟，每个时钟周期计数器减 1。设置界面如下图 3-3 所示。

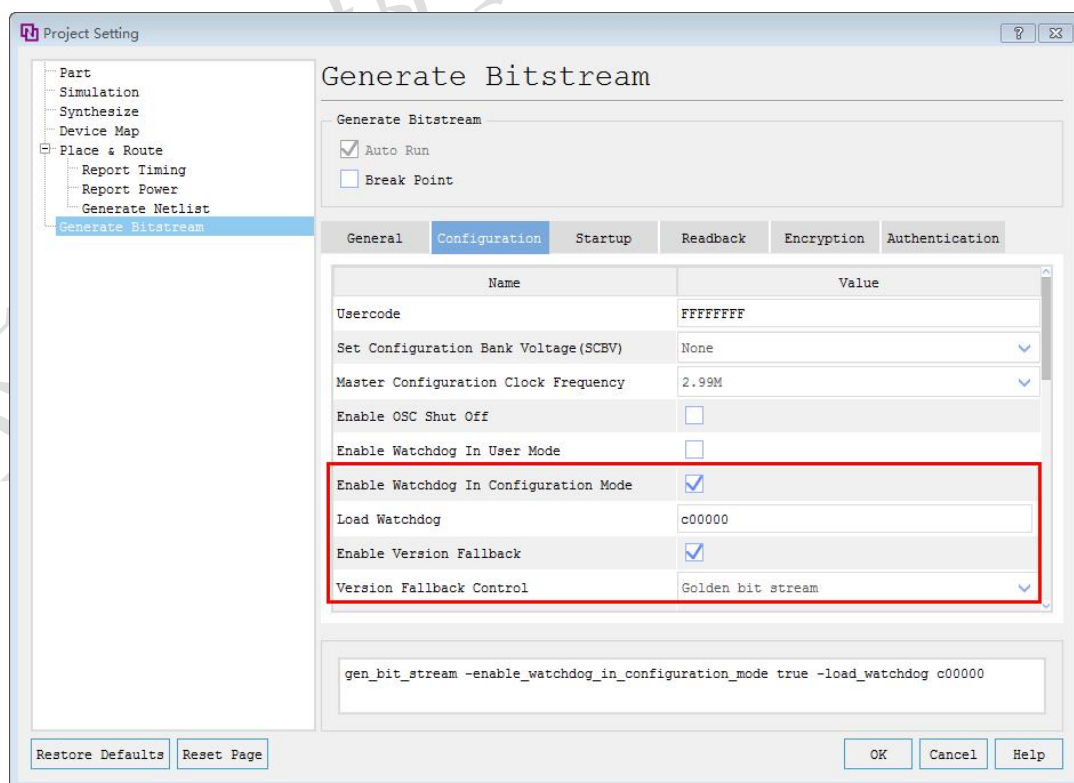


图 3-3 版本回退及看门狗设置

10. 若增加新器件，需要修改 spi\_top.v 模块中，参数“DEVICE”对应代码。例如：增加 PG2L200H，则增加一行代码：“PG2L200H” : subsector\_num <= 16'd2244 ;。
- PG2L200H 非压缩位流大小为 9188552 字节，8973.1953125KB，是 4KB 的 2243.298828125 倍，向上取整即为 2244。

### 3.3 参考设计上板验证

#### 3.3.1 生成合并位流

- 通过 PDS-Configuration 的 Generate Flash Programming File 插件，将黄金数据流和应用数据流分别转换成 sfc 文件，如图 3-4 所示。转换文件时需选择对应的 flash 芯片厂家及型号，可选型号见第三步红框内的下拉列表。

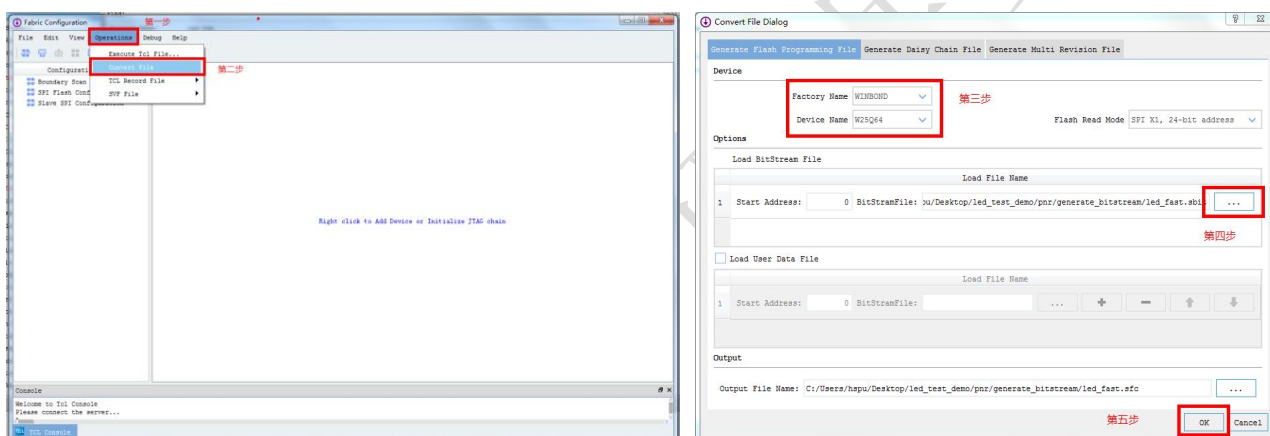


图 3-4 SFC 文件生成

- 在通过 PDS-Configuration 的 Generate Multi Revision File 插件，将黄金数据流和应用数据流整合为一个合并的 sfc 文件。

如图 3-5 所示，左边为合并 1 个应用位流的默认设置。右边的远程升级模块顶层参数定义，需正确设置应用位流 1 的起始地址，不关心其它应用位流。生成合并位流时，同时需要注意正确选择器件系列(Logos2)和位流类型(SPI Upgrade Data Stream)。



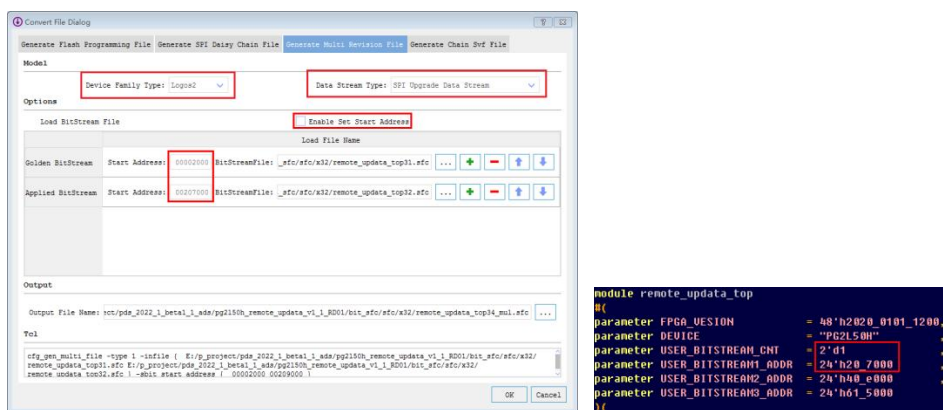


图 3-5 合并 1 个应用位流

如图 3-6 所示，左边为合并 2 个应用位流的默认设置。右边的远程升级模块顶层参数定义，需正确设置应用位流 1 和应用位流 2 的起始地址，不关心应用位流 3。

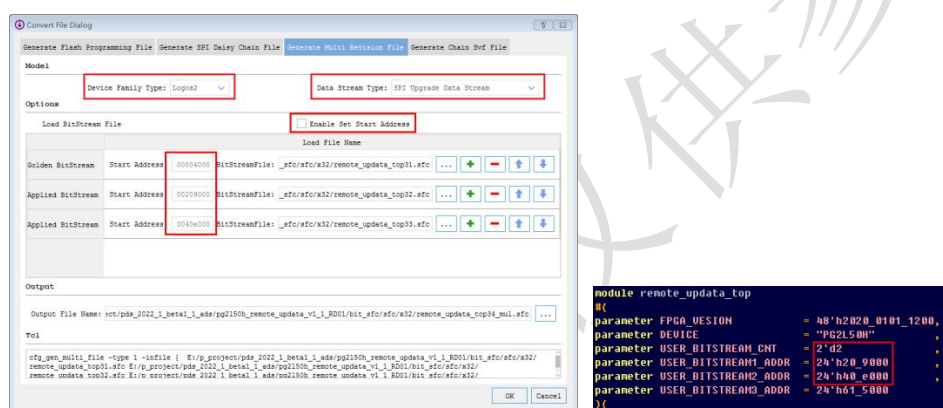


图 3-6 合并 2 个应用位流

如图 3-7 所示，左边为合并 3 个应用位流的默认设置。右边的远程升级模块顶层参数定义，需正确设置所有 3 应用位流的起始地址。

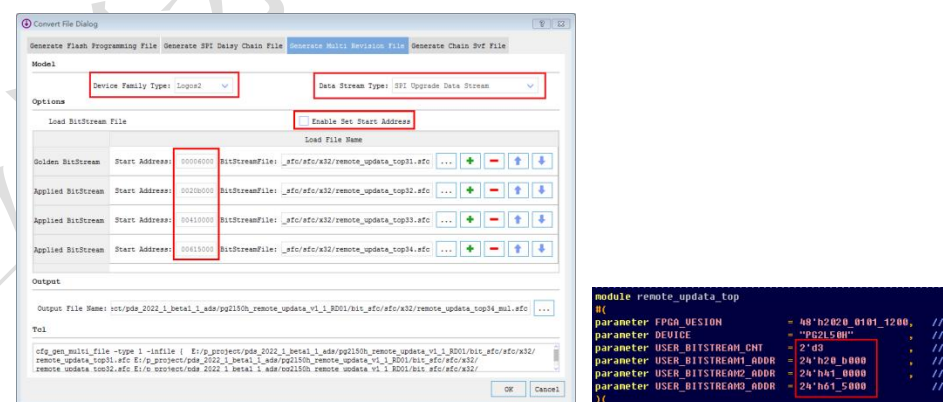


图 3-7 合并 3 个应用位流

若设置位流起始地址，需勾选使能，具体设置方法参考 3.2.1 小节说明。

► 通过 PDS-Configuration，将合并位流文件烧写到 flash 芯片，如图 3-8 所示。

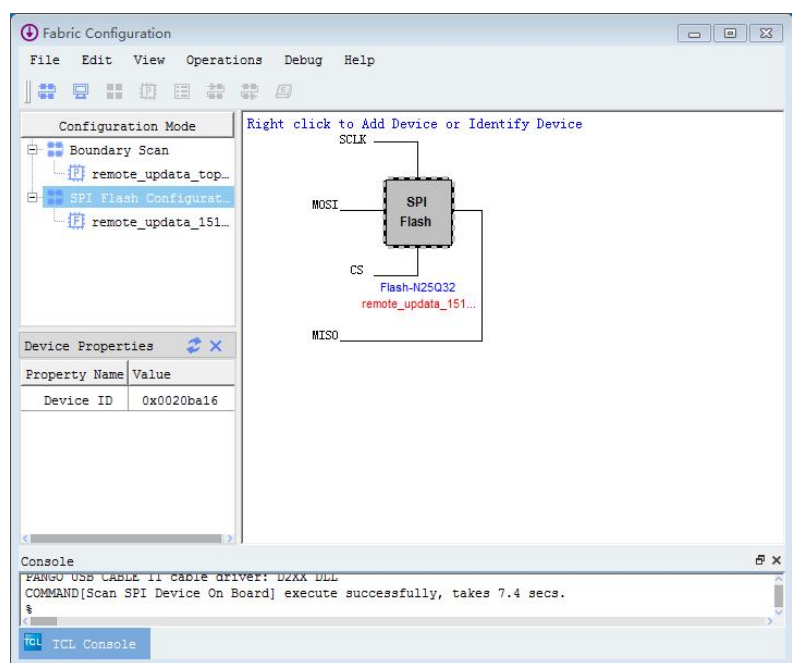


图 3-8 烧写合并位流文件

### 3.3.2 上板验证

以下实例，合并位流中包含 2 个应用位流，更新应用位流 1。

➤ 在上位机发送读版本信息命令，如图 3-9 所示。



图 3-9 读位流版本信息

当前的位流文件版本时间为：2022-05-31，14:45。该步骤非远程升级必要，主要方便确认是否更换成功。

➤ 在上位机配置 CRC 寄存器，如图 3-10 所示。



图 3-10 配置 CRC 寄存器

CRC 寄存器地址为 01，配置时地址后接 CRC 值。在读位流文件进行校验之前，需要上位机计算应用位流的 CRC32 结果，并配置到 CRC 寄存器。

➤ 在上位机发送写应用位流使能命令，如图 3-11 所示。



图 3-11 发送写位流使能命令

发送完写应用位流使能命令后，需要等待擦除完成，收到完成指示(55 05 01)后才能继续下发位流数据。

➤ 在上位机发送应用位流数据，如图 3-12 所示。



图 3-12 写应用位流完成

应用位流发送完后，需要发送结束标志码(7e 7e 7e 7e)，收到(55 05 10)，则表示写应用位流完成。

- 在上位机读位流文件进行校验，如图 3-13 所示。



图 3-13 读位流文件校验完成

读位流文件使能后，FPGA 会将读出的位流文件数据返回给上位机，同时计算 CRC32。读完应用位流后，会将计算的 CRC32 与上一步骤中配置到 CRC 寄存器的值比较，并将比较结果(55 03 01/00)返回上位机。

- 在上位机发送打开开关命令，如图 3-14 所示。



图 3-14 发送打开开关使能命令

在热启动前，需先打开开关程序。打开完成后，会返回完成指示(55 05 04)到上位机。

➤ 在上位机发送热启动命令，如图 3-15 所示。



图 3-15 发送热启动使能命令

不管校验结果是否正确，均可热启动。若校验结果不正确，可以重复以上步骤，重新写应用位流再读出校验，直至校验结果正确后热启动。

➤ 热启动后读当前版本信息，如图 3-16 所示。



图 3-16 热启动后读当前位流版本信息

若热启动前的校验结果正确(55 03 00)，热启动后加载之前更新的应用位流，否则加载黄金位流。

## 4 附录

### 4.1 python 脚本

内容如下:

```
fp = open(r'D: \asd_py\sfc\remote_updata_top.sfc', 'rb')
data = fp.read()
fp.close()
fp1 = open(r'D: \asd_py\sfc\remote_updata_top.txt', 'w')
idx = 0
for tmp in data:
    fp1.write("{:0>2s} ".format(hex(tmp)[2:]))
    idx += 1
    if idx == 16:
        fp1.write("\n")
        idx = 0
fp1.close()
```

运行脚本前, 需要修改文件路径和文件名。

### 4.2 python 脚本操作步骤

11. 将生成的 sbit 文件, 先在 PDS→Configuration→Operations→Convert File 中转换成 sfc 文件。
12. 将 sfc 文件放入对应的文件夹, 并修改脚本中的文件路径和名称。
13. 在脚本所在文件夹, 运行脚本完成转换。按住 shift 键+点击鼠标右键, 弹出菜单中选择“在此处打开命令窗口(W)”, 然后输入“python asd.py”(asd.py 为脚本文件名), 回车即可。



## 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。

4、本文档披露的信息用于帮助用户解决应用方面的问题，但不保证文档没有瑕疵；如果用户没有按照本文档提供的方法操作，由此引起的功能异常和性能衰退，公司不承担任何责任，也无法承认是产品的问题。同时，本文档提供的方案仅是其中一种可行性方案，不能保证涵盖所有应用场景，如果用户按照本文档的描述操作存在功能异常或者性能衰退，公司也不能保证是产品的问题。