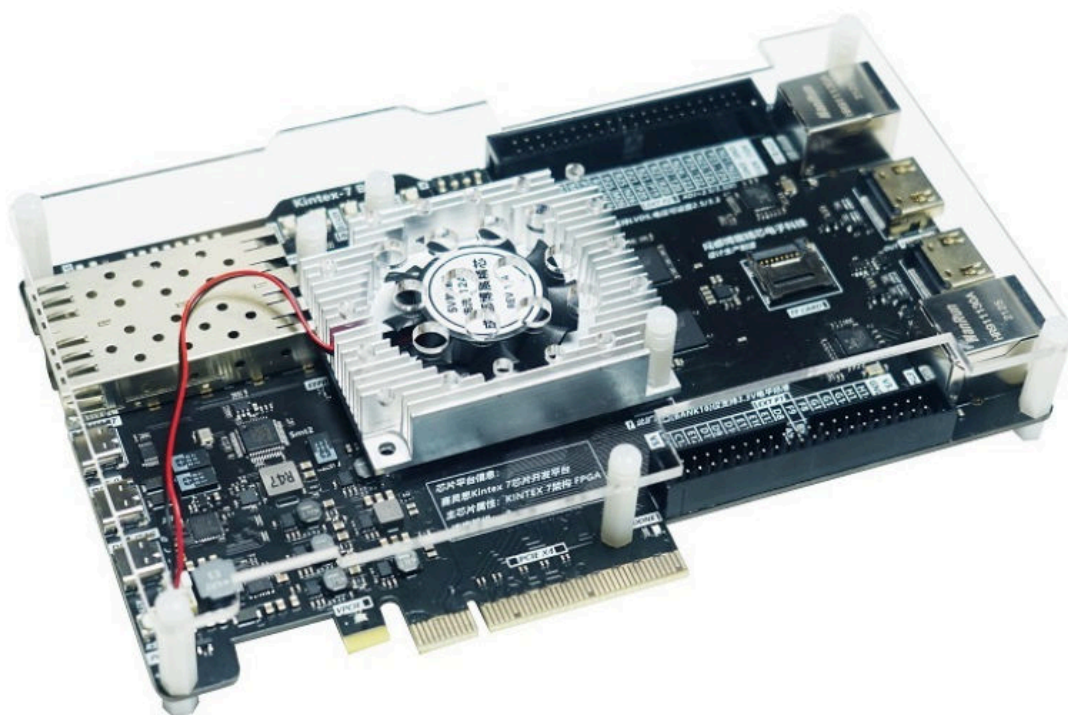


FPGA启动的MCS文件固化

----- 基于Kintex 7 base开发板



编写人：杨老师、吴老师

编写定稿日期：2020.06.26

目录

一、 文档实现功能介绍.....	3
二、 Kintex 7工程新建.....	3
三、 添加约束.....	3
四、 烧写FLASH.....	8
五、 上电启动.....	10

小熊猫内部教程，严禁外传

一、文档实现功能介绍

本文档实现如何将bit文件生成bin文件固化到FLASH然后上电启动。对于固化FLASH，我们工程需要在xdc添加约束，来告诉VIVADO软件我们FLASH基本参数信息。工程新建方法请参考文档(02_VIVADO简介及软件下新建FPGA工程教程)。

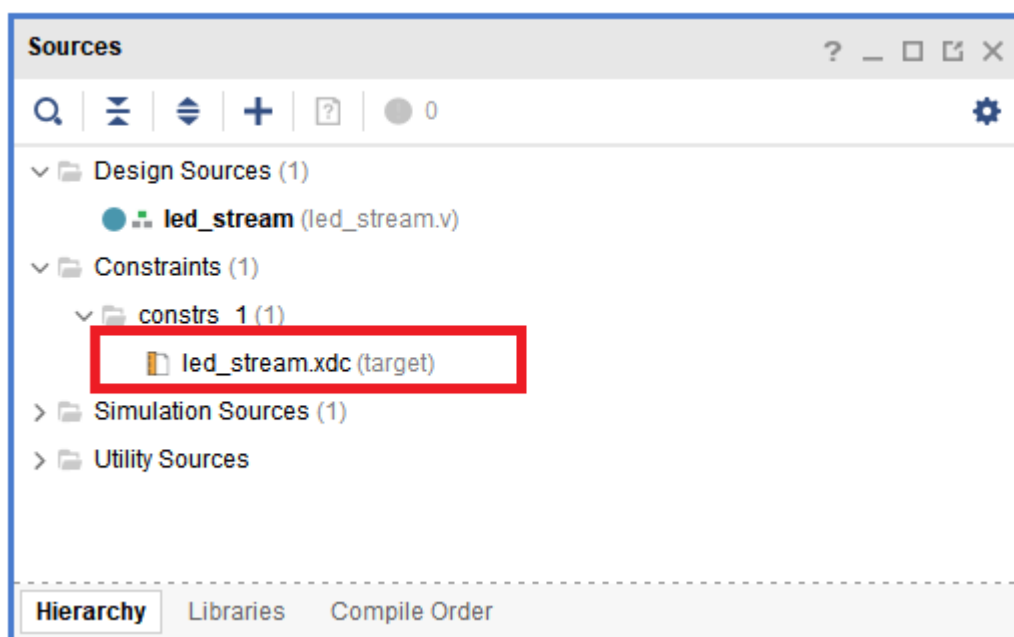
二、Kintex 7工程新建

本教程可用任意例程固化，我们实际演示采用第一个流水灯程序进行试验。我们直接打开第一个demo，在里面进行本教程实验即可。

起始页 (或file->Project-> New)	创建新工程(CreateNewProject)
向导起始页面	点击 Next->
ProjectName(工程名)	工程名: k701_led_stream 工程路径: (自己选择, 尽量不要有中文路径) 勾选CreateProjectSubdirectory点击Next->
AddSource(添加设计源文件)	点击 Next->
AddExsistingIP(添加已有的 IP)	点击 Next->
AddConstraints(添加已有约束文件)	点击 Next->
Default Part(默认配置, 芯片选型)	Family-> Kintex 7 Package-> FFG676 Speed-> -2 点击 Next-> 选择 XC7K325TFFG676-2
NewProjectSummary(新建工程概况)	确认工程信息, 选型等, 点击 Finish 完成

三、添加约束

我们打开k701_led_stream的约束文件：（关于约束文件，我们资料教程写完，后面看情况补充一个这些概念性的知识文档再介绍），这个约束文件是我们在led流水灯实验中，绑定管脚生成的xdc文件，xdc一般包含引脚约束（电平标准，引脚位置）等，以及时钟的约束，在线调试的一些信号约束，可以自己按照一定语法添加进去，也可以是软件自动添加（比如绑定管脚软件就会把信息写到这个文件）。



我们打开它看一下，可以看到，上面都是我们led和时钟，复位信号的引脚和电平约束。

```

1 set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
2 set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
3 set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
4 set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
5 set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
6 set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
7 set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
8 set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
9 set_property PACKAGE_PIN A23 [get_ports {led[0]}]
0 set_property PACKAGE_PIN A24 [get_ports {led[1]}]
1 set_property PACKAGE_PIN D23 [get_ports {led[2]}]
2 set_property PACKAGE_PIN C24 [get_ports {led[3]}]
3 set_property PACKAGE_PIN C26 [get_ports {led[4]}]
4 set_property PACKAGE_PIN D24 [get_ports {led[5]}]
5 set_property PACKAGE_PIN D25 [get_ports {led[6]}]
6 set_property PACKAGE_PIN E25 [get_ports {led[7]}]
7 set_property PACKAGE_PIN G22 [get_ports clk]
8 set_property IOSTANDARD LVCMOS33 [get_ports clk]
9 set_property IOSTANDARD LVCMOS33 [get_ports rst_n]
0 set_property PACKAGE_PIN D26 [get_ports rst_n]
1

```

我们在约束文件最下面添加我们的固化FLASH的约束语句：

```

set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]
set_property CONFIG_MODE SPIx4 [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 50 [current_design]
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.UNUSEDPIN Pullup [current_design]
set_property CFGBVS VCCO [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]

```

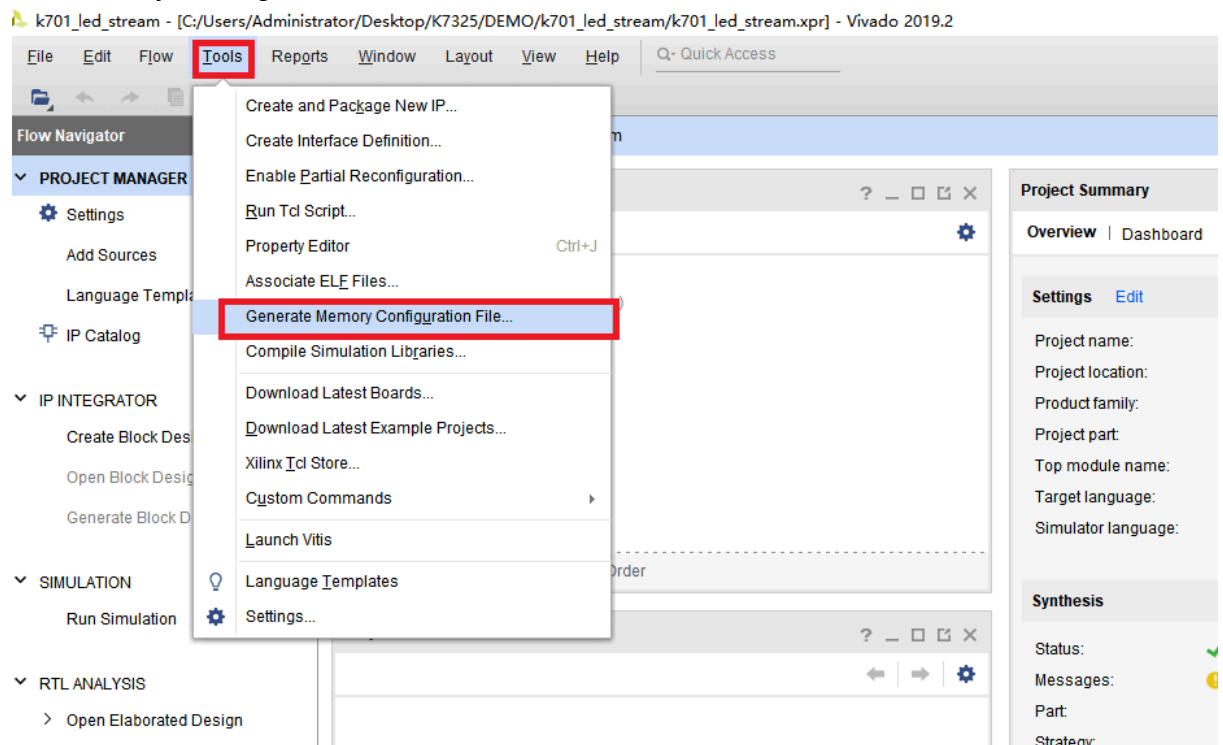
每一句的含义大家可以通过名字来简单理解一下，大概表示，FLASH位宽为4，使用

SPIx4模式，时钟速率50M，文件是压缩模式，不使用上拉电阻模式，电源VBUS为VCCO管脚提供，配置接口电平电压3.3等等。这些告诉软件，在生成配置文件和下载的时候，告诉软件基本的下载配置信息。

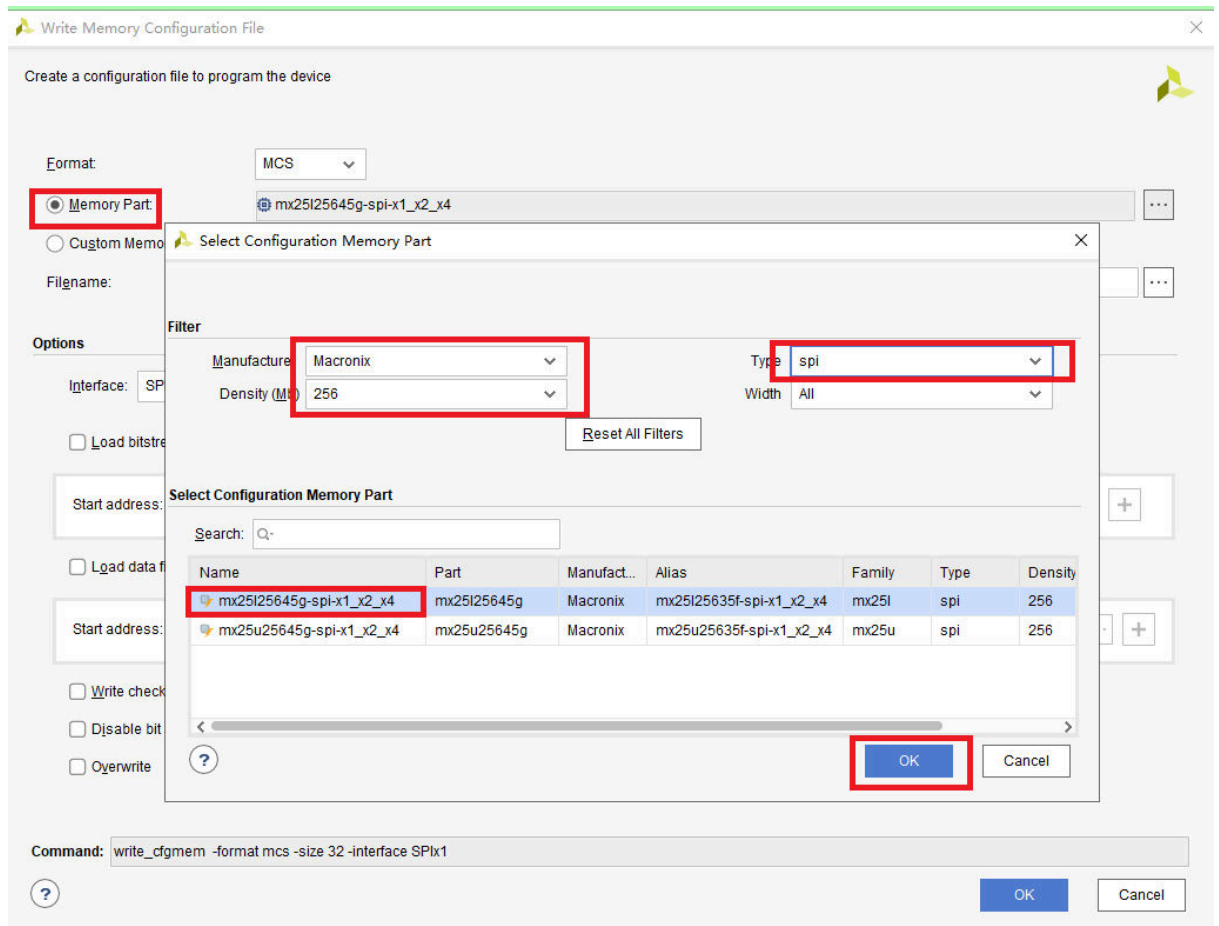
生成比特文件：

我们将功成综合，布局布线生成比特文件。这里不多讲这些过程，请参考《04_最简单的led流水灯》。

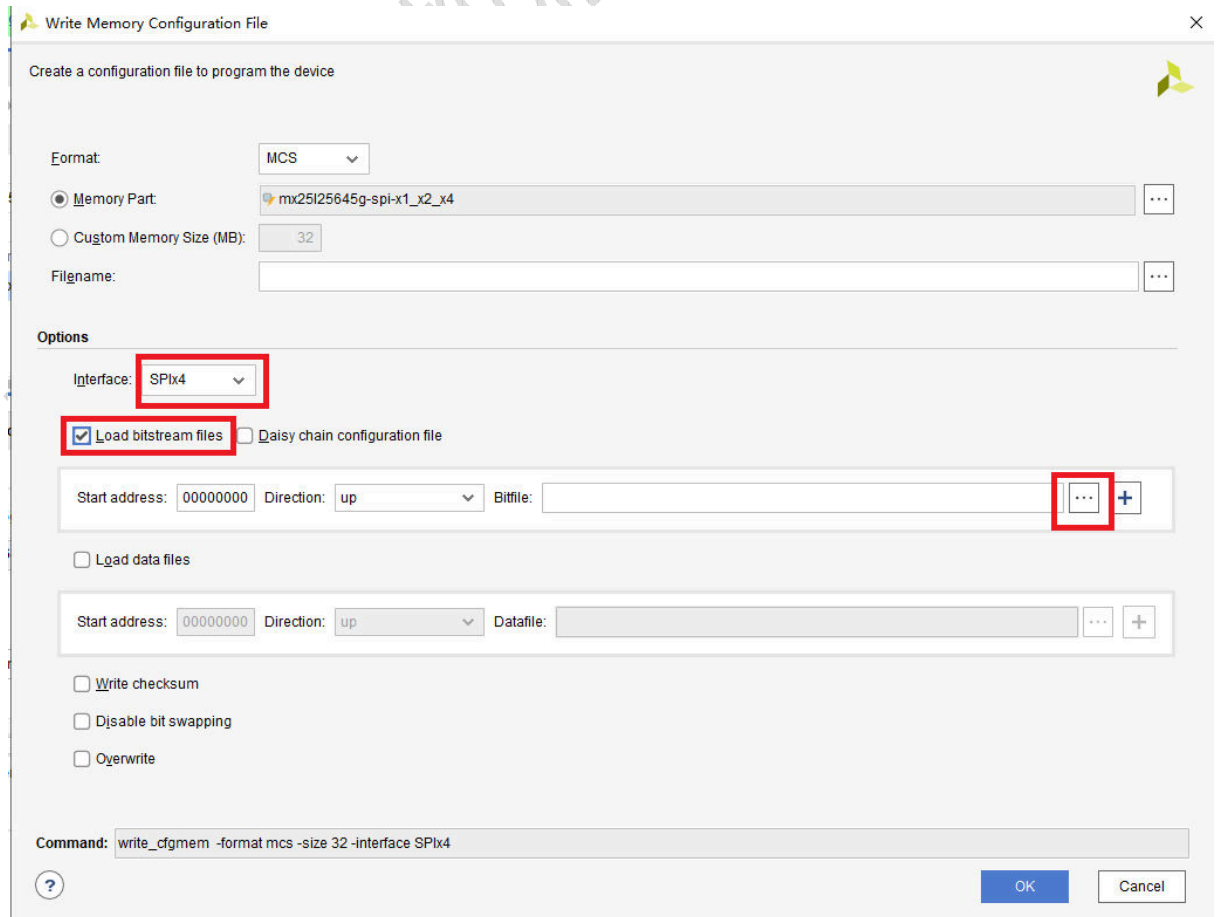
生成比特文件之后，我们开始生成配置FLASH的MCS文件，点击工具栏的Tools找到Generate Memory Configuration File进行创建文件：



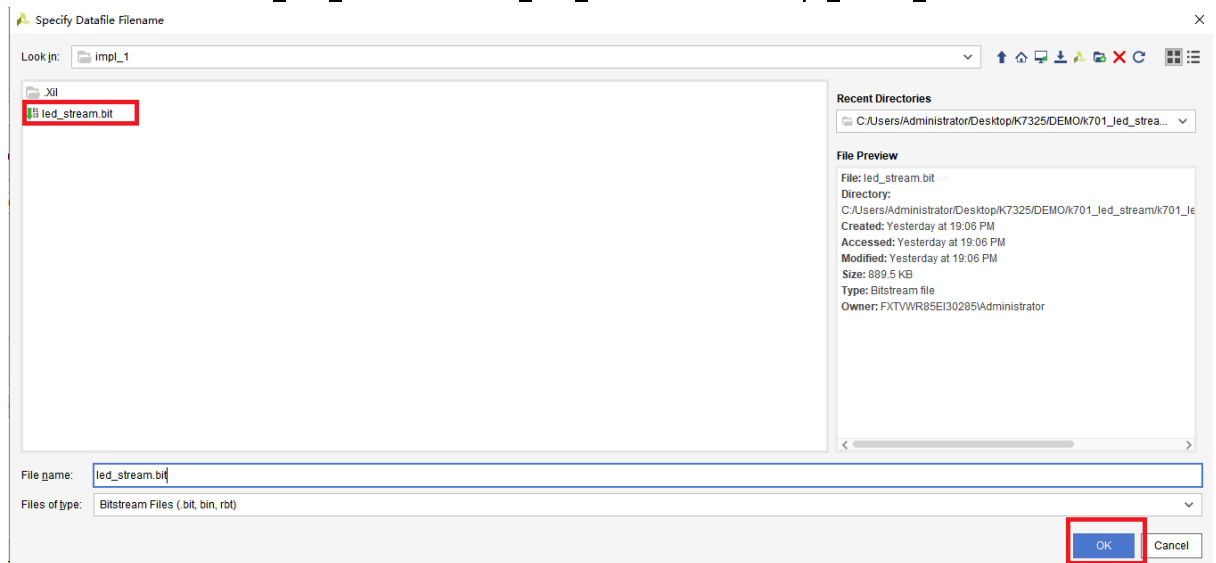
弹出的界面，我们如下配置，勾选Memory Part,点击右边的三个点 (...) 表示我们选择FLASH型号。我们这里写一下：选择厂家为Macronix，容量256 (M bit)，type选择spi，最后找到下面第一项，型号为mx25l256g-spi-x1_x2_x4即可。我们实际板子上用的就是这颗芯片，256Mbit容量实际为32M字节空间，一般情况下，足够大家做开发使用，容量也完全用不完。最后选完点击OK退出芯片选型。



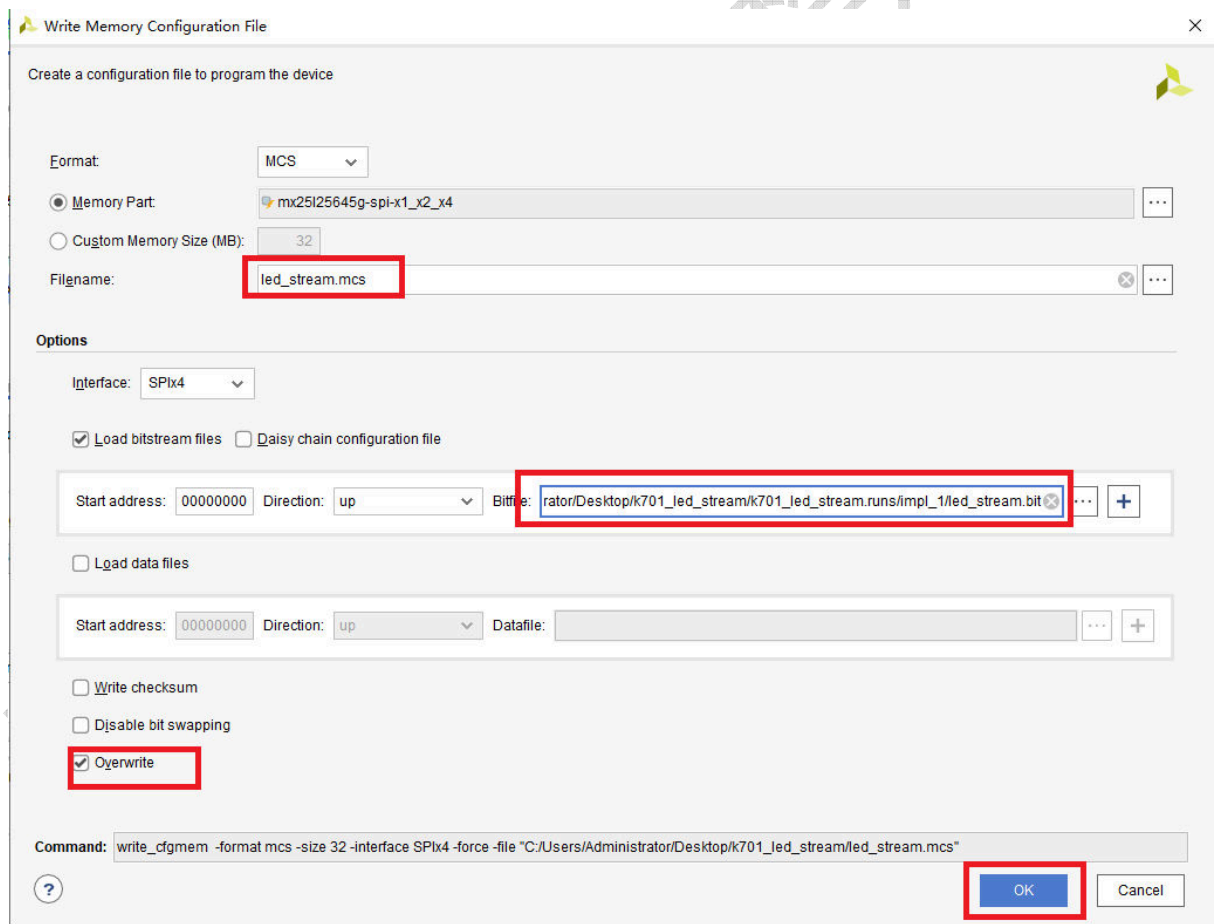
然后interface选择SPIx4,下面勾选Load bitstream files加载bit文件进来，点击右边的三个点 (...) 可以选择本工程生成的bit文件进来。



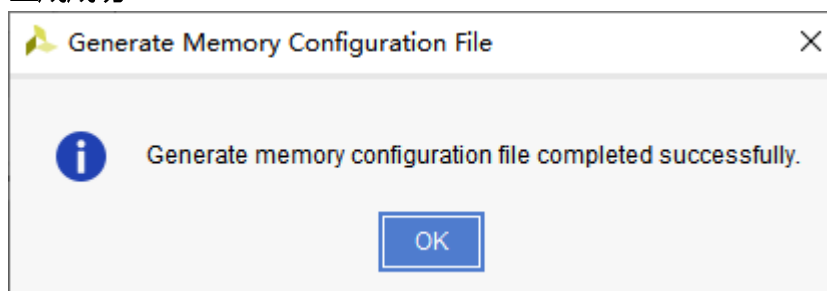
Bit文件位于: /k701_led_stream/k701_led_stream.runs/impl_1/led_stream.bit



加载进来bit文件之后记得把上面的Filename名字填上，表示生成的flash配置MCS文件名，然后点击OK:



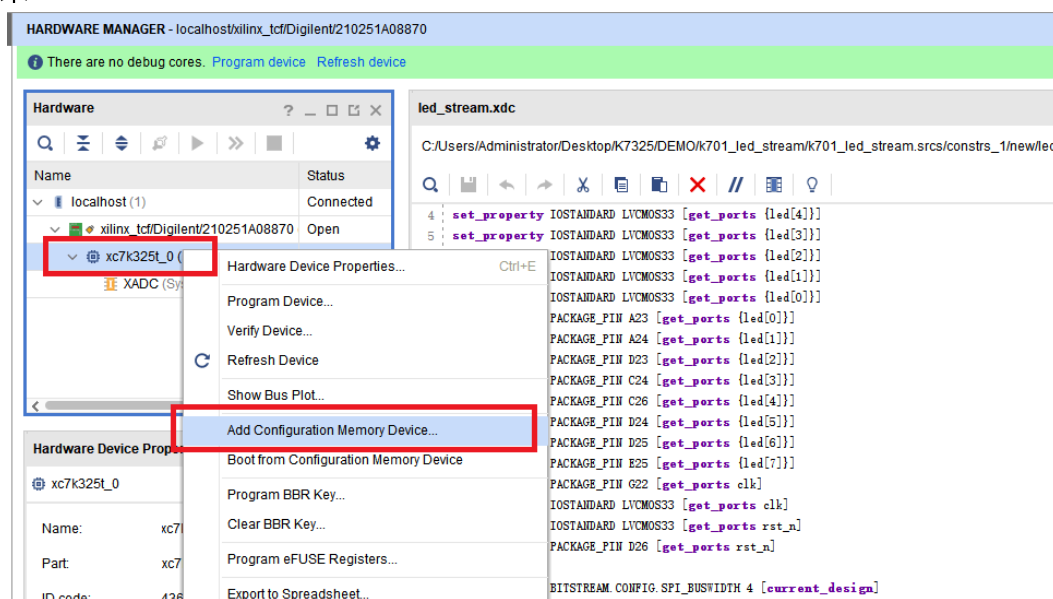
生成成功:



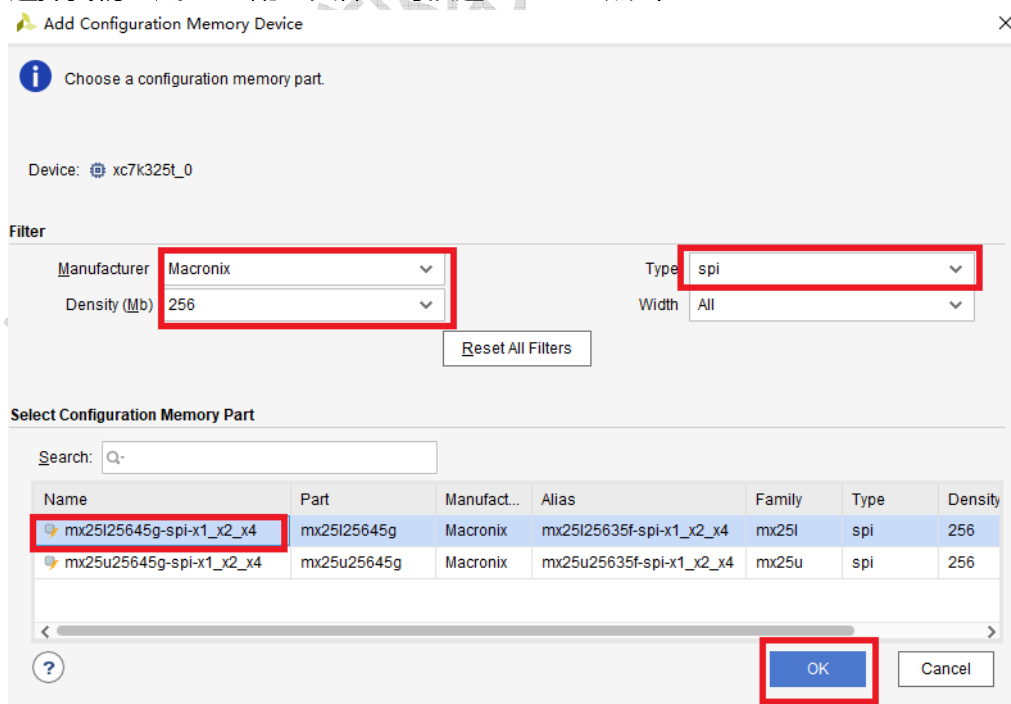
如果生成失败，根据弹出的提示框检查，以及检查我们的xdc文件，然后重新尝试。记住，VIVADO选择bit文件路径要确定选的是不是我们对应工程的bit文件，选到其他文件去了就不能用。

四、烧写FLASH

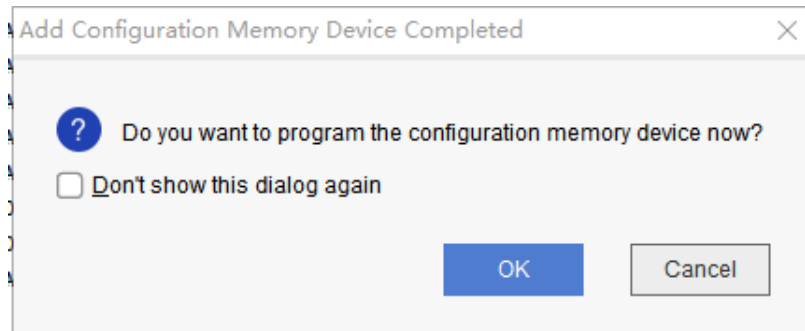
上一节生成了烧写FLASH的MCS配置文件，我们接着开始烧写。我们打开硬件管理器，开发板上电并且连接开发板找到芯片，然后右键选择Add configuration device添加FLASH设备进来：



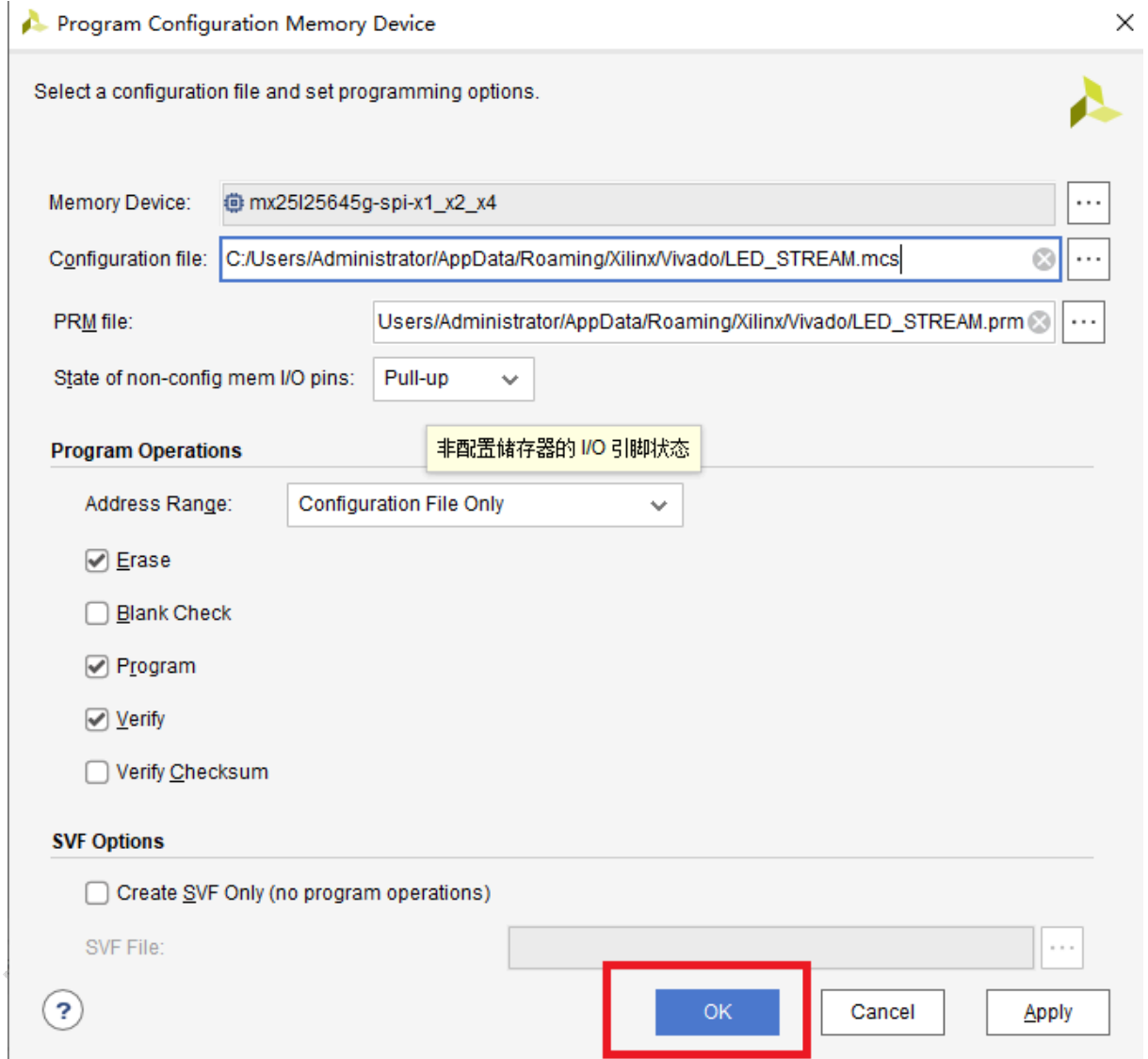
选择我们生成MCS配置文件的时候选的FLASH点击OK:



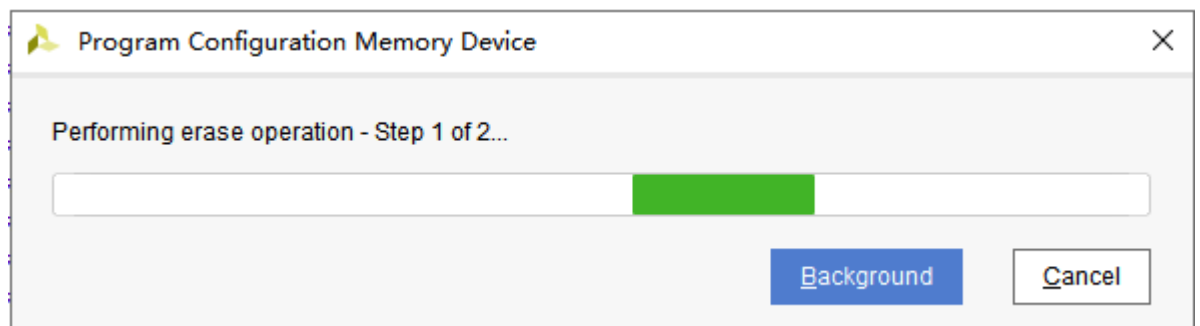
然后我们看到硬件管理下面出现了这个FLASH,并且弹出对话框问是否现在烧写;



上面的对话框直接点击OK，弹出烧写界面直接点击OK即可开始烧写：



等待烧写完成。烧写分为几个过程，首先擦除，然后写入，最后校验：



烧写结束，关掉开关，我们设置一下拨码开关。如何设置请看电路板背面即可。这里

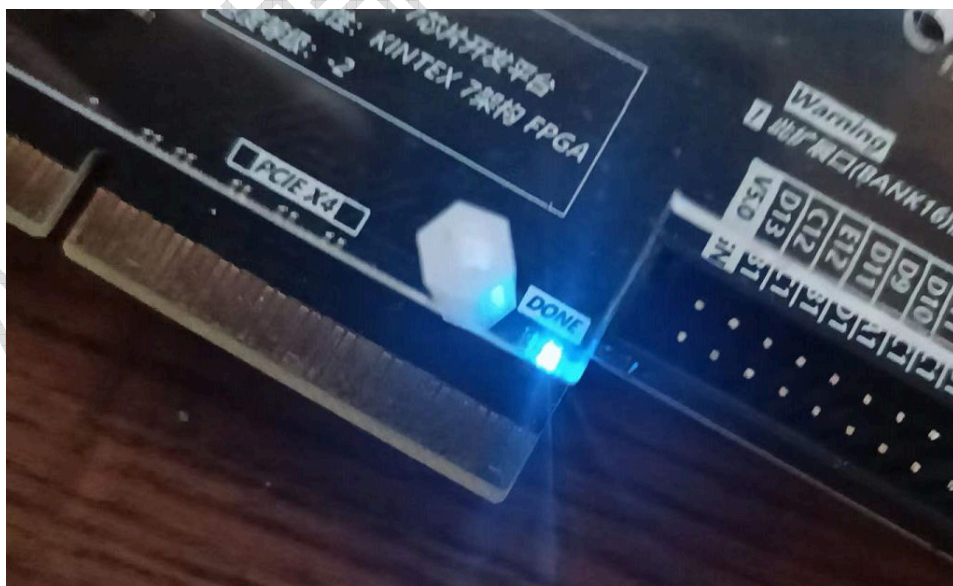
我们贴一下图：



设置2-3段ON，第四段OFF表示商店从FLASH加载。

五、上电启动

最后我们重新打开电源开关上电，可以看到四个LED已经开始流动闪烁。并且，PCIE靠近扩展IO的一个DONE信号LED亮起，表示已经完成配置。DONE信号就是FPGA提供的一个上电配置完成信号，当JTAG正在配置，或者上电没有正确配置，DONE信号LED不会亮起，或者FLASH没有正确的可以加载启动的文件，也不会亮起，当正确加载启动，灯亮，表明开始工作。



本教程到此结束。大家可以使用我们的其他例程作本实验，把生成的bit文件固化到FLASH运行。对于大的工程，大家烧写到FLASH，功耗大，连接电脑可能导致供电不足，或者直接不能正常加载，大家需要额外将DC 5.0或者 UART口连接一个5V/2A以上的充电器供电即可。