

# WORD

XD-LAB-IMG-002

## Lab2: 像素处理实验 1: 灰度化和二值化

Joseph Xu

2019-2-14

## 修改记录

版本号.	作者	描述	修改日期
1.0	Joseph Xu	初稿	2018-4-1
1.2	Joseph Xu	图片微调	2019-2-14

## 审核记录

姓名	职务	签字	日期

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	1 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

## 目录

修改记录.....	1
审核记录.....	1
1. 实验简介.....	7
1.1 概述.....	7
1.2 实验目标.....	8
1.3 实验条件.....	8
1.4 实验原理.....	8
2. PARTA: 灰度化实验流程.....	12
2.1 操作步骤.....	12
3. PARTA: 灰度化实验结果.....	27
4. PARTB: 二值化实验流程.....	28
4.1 操作步骤.....	28
5. PARTB 二值化实验结果.....	40

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	2 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

## 图目录

图 1-1	彩色转灰度效果对比图.....	7
图 1-2	实验连接示意图.....	9
图 1-3	图像灰度化连接示意图.....	9
图 1-4	图像二值化连接示意图.....	9
图 1-5	Graying IP .....	9
图 1-6	Threshold IP .....	10
图 2-1	Vivado 下创建新工程.....	12
图 2-2	重命名实验目录.....	12
图 2-3	图像处理 IP 库文件存放位置.....	13
图 2-4	启动 Vivado.....	13
图 2-5	打开工程.....	14
图 2-6	设置: 添加 IP 库.....	14
图 2-7	选择图像 IP 库.....	15
图 2-8	图像处理 IP 库中的 IP 列表显示 .....	15
图 2-9	实验初始视图.....	16
图 2-10	添加灰度化 IP .....	16
图 2-11	断开 DVI2RGB IP 的 RGB 端口 .....	17
图 2-12	断开端口后视图.....	17
图 2-13	展开 DVI2RGB IP 的 RGB 端口 .....	18
图 2-14	RGB 端口展开后视图.....	18
图 2-15	连接端口.....	19
图 2-16	添加 Concat IP .....	19
图 2-17	添加 Concat IP 后的视图 .....	20
图 2-18	Concat IP 设置 .....	20
图 2-19	连接 Graying IP 后的视图 .....	21
图 2-20	端口连接检查.....	21
图 2-21	保存设计.....	22
图 2-22	创建实验顶层 Wrapper 文件 .....	22
图 2-23	自动更新顶层文件.....	22
图 2-24	Generate Bitstream.....	22
图 2-25	点击 Yes 确认生成 bit 文件.....	23

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	3 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

图 2-26	打开 Hardware Manager .....	23
图 2-27	硬件连接对应位置.....	24
图 2-28	实际硬件连接.....	24
图 2-29	Open target .....	25
图 2-30	Program Device .....	25
图 2-31	烧写目标器件.....	26
图 2-32	编程进度条.....	26
图 3-1	灰度化显示结果.....	27
图 4-1	回到 Vivado 主界面继续添加 IP .....	28
图 4-2	添加二值化 IP: Threshold .....	28
图 4-3	断开 RGB2DVI IP 的 vid_pVDE 端口 .....	29
图 4-4	断开数据流后的 IP 视图.....	29
图 4-5	Thresh IP 配置.....	30
图 4-6	Concat IP 设置 .....	30
图 4-7	端口连接检查.....	31
图 4-8	添加 Constant IP .....	31
图 4-9	第一个 Constant IP 设置 .....	32
图 4-10	第二个 Constant IP 设置 .....	32
图 4-11	两个 Constant IP 的连接 .....	33
图 4-12	保存设计.....	33
图 4-13	创建顶层文件.....	34
图 4-14	自动更新顶层文件.....	34
图 4-15	Generate Bitstream.....	34
图 4-16	点击 Yes 确认生成 bit 文件.....	35
图 4-17	忽略提示信息并点击 OK.....	35
图 4-18	打开 Hardware Manager .....	36
图 4-19	硬件连接对应位置.....	37
图 4-20	实际硬件连接.....	37
图 4-21	Open target .....	38
图 4-22	Program Device .....	38
图 4-23	烧写目标器件.....	38
图 4-24	编程进度条.....	39

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	4 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

图 16-1 二值化显示结果.....40

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	5 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

## 表目录

表 1-1	Graying IP 端口列表 .....	10
表 1-2	Threshold IP 端口列表 .....	10

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	6 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

## 1. 实验简介

该实验以实验 1 为基础，其中灰度化部分为按照一定的时钟频率将 24 位彩色像素数据转换为 8 位的灰阶数据；二值化部分为设定阈值后，以阈值为界限对每个像素进行比较，从而转换为单一的黑白图像。

- **对于初学者，整个实验预计耗时 1 小时。**
- **对于熟练者，整个实验预计耗时 20 分钟。**

### 1.1 概述

在摄影和计算机图像学等领域，灰度图指的是每个像素均用单一采样的明亮度来表示的图像，也就是说，灰度图中只包含亮度 (intensity) 信息，视觉上呈现不同程度的灰色效果。早期的黑白摄影照片即为灰度图的一种形式。到了彩色照片时期，每个像素采用了红绿蓝三原色 (简称 RGB) 来表示。但在图像处理上，一般仍然是先将彩色图片先转换成灰度图然后在进行处理，处理完成后，再由灰度图转换回彩色图。这样做的原因是因为一个彩色像素的数据宽度通常为 24 位 (RGB888) 或 16 位 (RGB565)，而一个灰度像素的数据宽度为 8 位，从数据处理带宽上来说，灰度图数据计算量能减少到彩色图数据计算量的 1/2 到 1/3。从彩色图像转换到灰度图像的操作称之为灰度化。

在灰度图的基础上，将像素的灰度按某个范围分割为黑 (最暗) 或白 (最亮) 的图像称为黑白图。将灰度图转换为黑白图的操作称之为二值化。

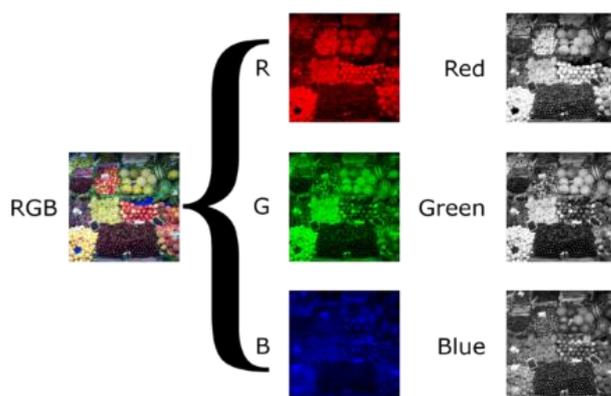


图 1-1 彩色转灰度效果对比图

<b>xingdeng</b>	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	7 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

## 1.2 实验目标

本实验的目标为 SWORD4.0 能够正常地在 HDMI 显示器上输出视频画面，其中：

- A. 1280x720@60Hz——灰度视频画面；
- B. 1280x720@60Hz——黑白二值视频画面；

## 1.3 实验条件

类别	名称	数量	说明
硬件	SWORD4.0	1	
	HDMI 信号源	1	如笔记本 HDMI 输出/台式计算机 HDMI 输出/带 HDMI 输出的视频机顶盒
	带 HDMI 接口的显示器	1	
	HDMI 视频线	2	
软件	Vivado Design Suite	1	版本：2014.4
	视频接口 IP 库	1	FPGA-Image-Library.zip*

\*注：FPGA-Image-Library 为戴天宇开发的一个开源图像处理 IP 库，该 IP 库遵循 LGPL，详情请见：<http://fil.dtysky.moe>

## 1.4 实验原理

该实验的连接方式如下图所示：

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	8 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

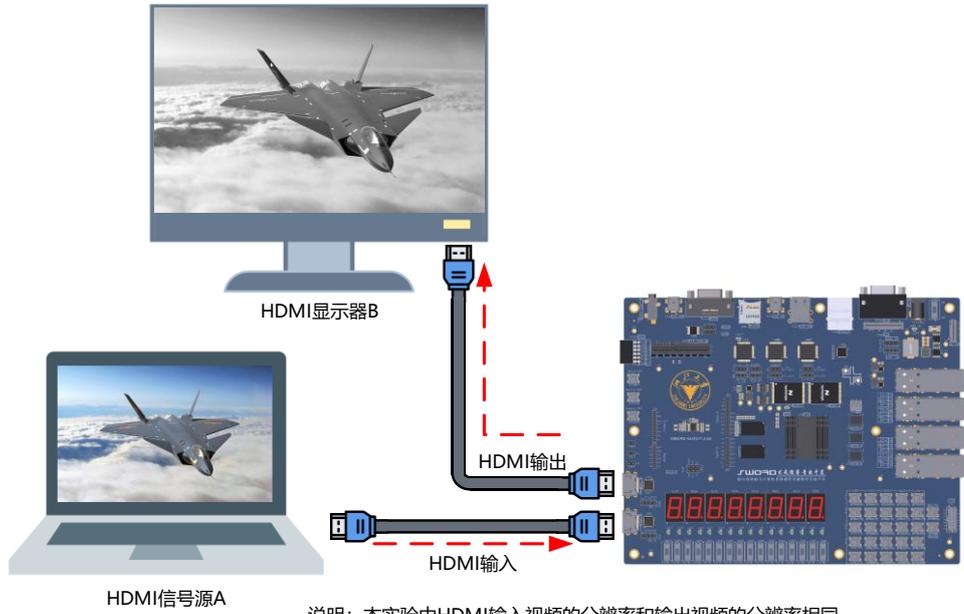


图 1-2 实验连接示意图

实验利用了 2 个 IP 来分别实现灰度化和二值化: Graying, Threshold。其中: 灰度化的实验 IP 连接示意图如下图所示:

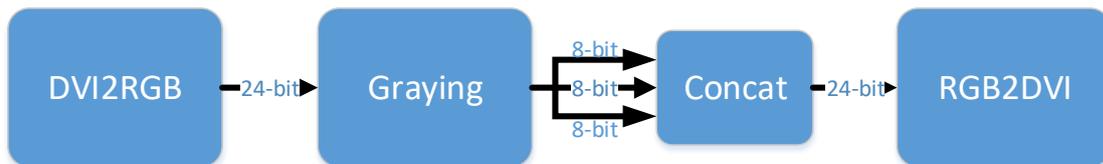


图 1-3 图像灰度化连接示意图

二值化的实验 IP 连接示意图如下图所示:



图 1-4 图像二值化连接示意图

Graying 这个 IP 的作用是将 24 位宽的 RGB 彩色数据转换为 8bit 的灰度数据。

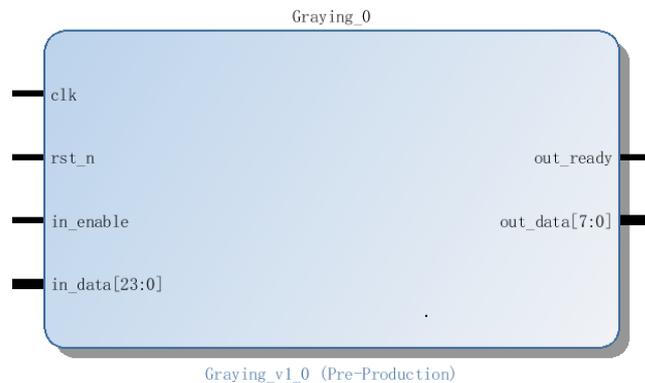


图 1-5 Graying IP

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	9 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

该 IP 的端口信号定义如下表所示:

表 1-1 Graying IP 端口列表

信号名	方向	宽度	含义
clk	输入	1	Clock.
rst_n	输入	1	复位, 低有效。
in_enable	输入	1	输入数据使能, 在流水线模式下, 它是另一个复位信号, 在请求响应模式下, 只有在它有效的时候 in_data 才会被真正地改变。
in_data	output	color_width - 1 : 0	输入数据, 必须和 in_enable 同步输入。
out_ready	output	1	输出数据有效, 在两种模式下, 这个信号都会在 out_data 可以被读取的时候有效。
out_data	输入	color_width - 1 : 0	输出数据, 将会和 out_ready 同步输出。

而 Threshold 这个 IP 的作用是将 8 位宽的图像数据按照设定的阈值转换为 0 或 1。

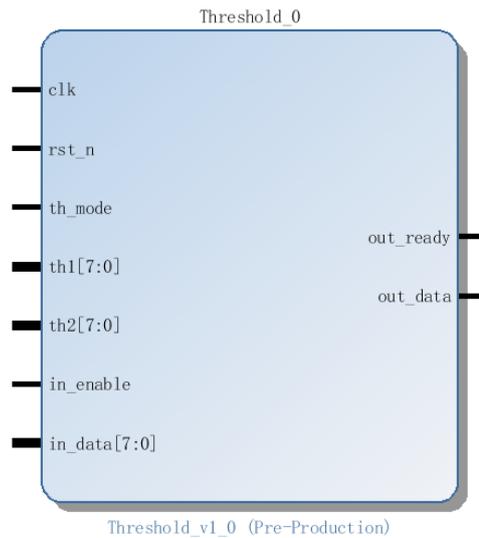


图 1-6 Threshold IP

该 IP 的端口信号定义如下表所示:

表 1-2 Threshold IP 端口列表

信号名	方向	宽度	含义
clk	输入	1	Clock.
rst_n	输入	1	复位, 低有效。
th_mode	input	0 为基本全局阈值化 1 为等高线阈值化	操作方法。
th1	input	color_width - 1 : 0	阈值 1, 用于两种模式。
th2	input	color_width - 1 : 0	阈值 2, 只能用于等高线阈值化模式。

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	10 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

信号名	方向	宽度	含义
in_enable	输入	1	输入数据使能，在流水线模式下，它是另一个复位信号，在请求响应模式下，只有在它有效的时候 in_data 才会被真正地改变。
in_data	输入	color_width - 1 : 0	输入数据，必须和 in_enable 同步输入。
out_ready	output	1	输出数据有效，在两种模式下，这个信号都会在 out_data 可以被读取的时候有效。
out_data	output	color_width - 1 : 0	输出数据，将会和 out_ready 同步输出。

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	11 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

## 2. PARTA: 灰度化实验流程

本章将详细描述如何在 Vivado 2014.4 的环境下完成实验。请耐心等待，仔细按照图示和文字说明进行操作。

### 2.1 操作步骤

1. 由于本实验是在实验 1 的基础上进行扩展，所以我们将之前的实验部分复制 1 份，具体做法为在 D:\ImageLabs 文件夹下，将鼠标左键选中 lab1，然后按住 Ctrl 键不放，并拖拽到空白处，这样得到一个 lab1 的副本，如下图所示：

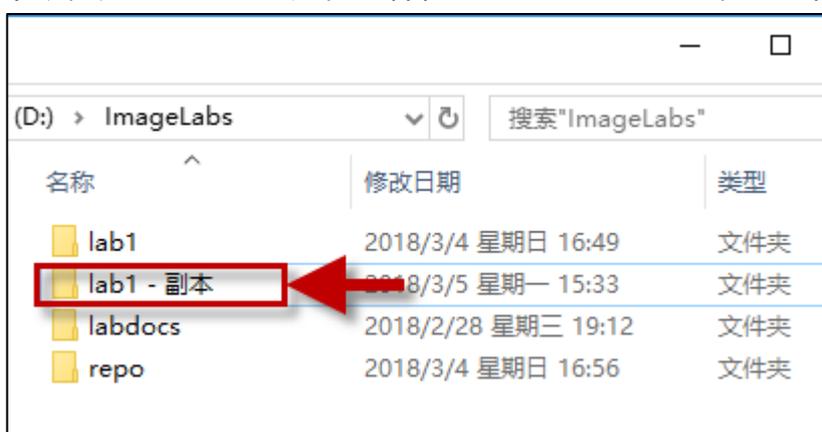


图 2-1 Vivado 下创建新工程

然后将 lab1 的副本重命名为 lab2，如下图所示，至此我们就可以在 lab2 文件夹里开始我们的实验内容：

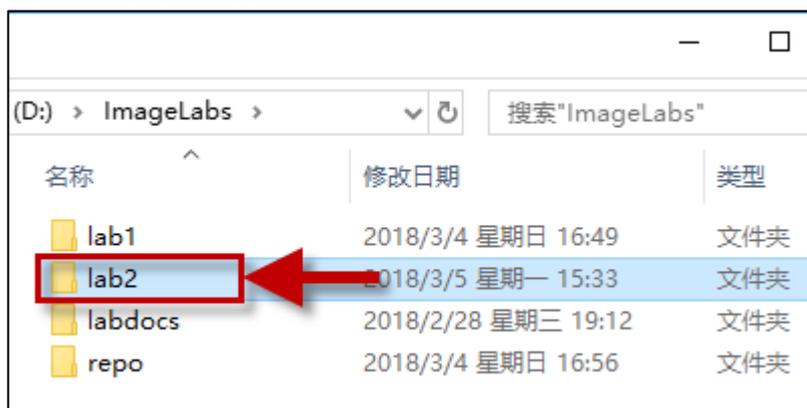


图 2-2 重命名实验目录

2. 将实验提供的开源 IP 库 FPGA-Image-Library.zip 解压到 D:\ImageLabs 下,和 repo, lab2 等文件夹同一级，如下图所示：

**提示：如果是初学者，请按照此路径存放，如果是 Vivado 的使用熟练者，可按照自己的习惯来选择 IP 库的存放位置。**

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	12 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

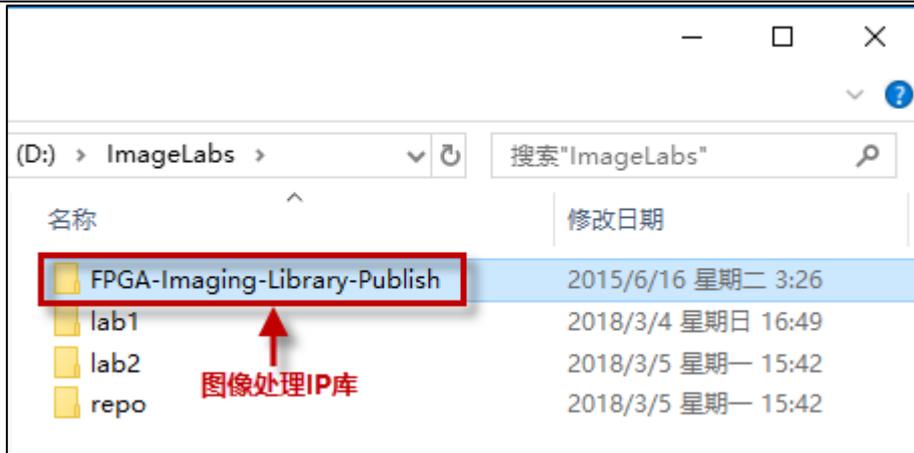


图 2-3 图像处理 IP 库文件存放位置

3 . 接着启动 Vivado 2014.4, 在启动界面选择 Open Project, 如下图所示:

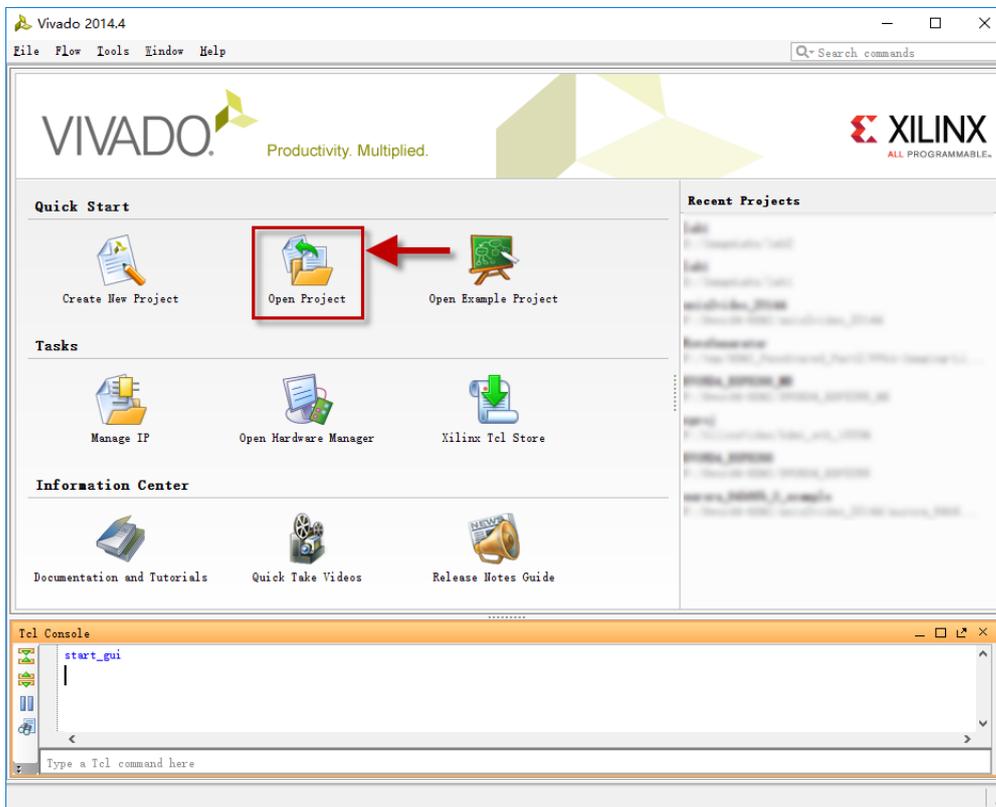


图 2-4 启动 Vivado

4 . 然后在选择对话框中, 找到之前的 lab2 (即 D:\ImageLabs\lab2), 然后选择 lab1.xpr 文件, 点击 OK, 打开工程, 整个过程如下图所示:

<b>xingdeng</b>	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	13 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

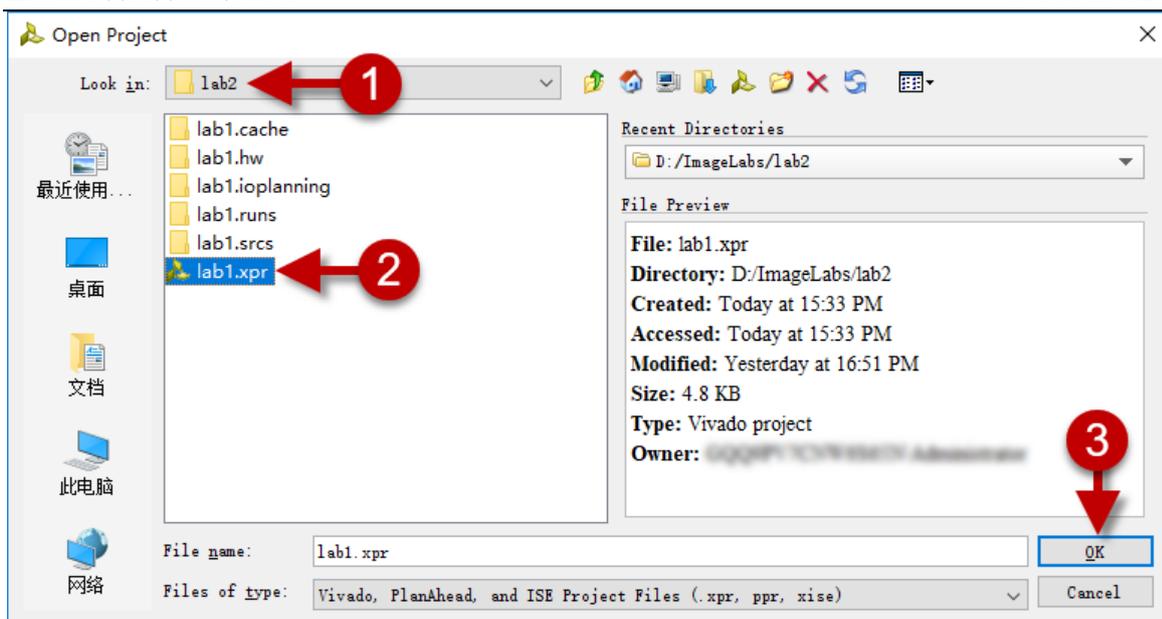


图 2-5 打开工程

5. 接着我们将 FPGA-Image-Library IP 库添加进来，在 Vivado 的主界面点击 Project Settings，接着在新开对话框中点击 IP，点击 Add Repository，过程如下图所示：

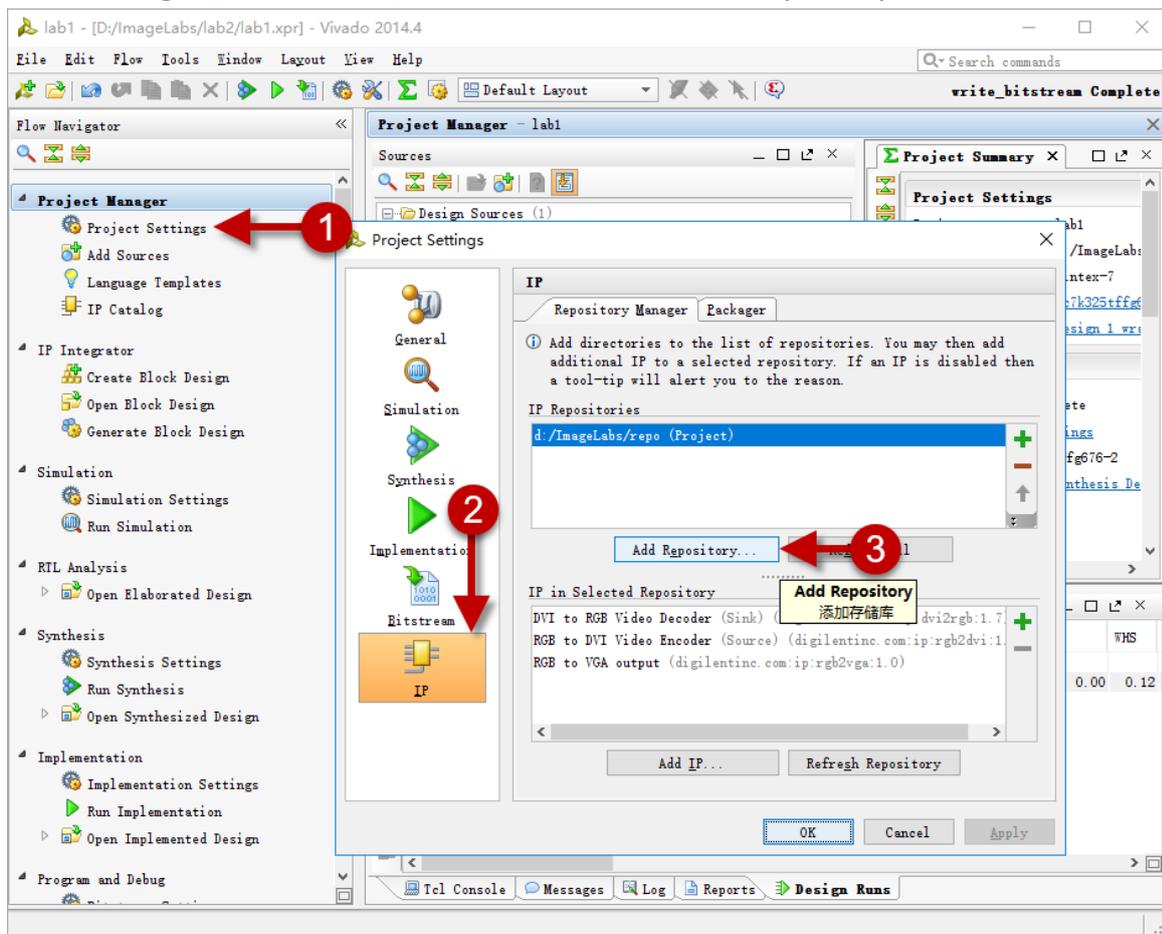


图 2-6 设置：添加 IP 库

6. 在对话框中找到 D:\ImageLabs\FPGA-Image-Library-Publish，然后点击 Select，

<b>xingdeng</b>	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	14 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

如下图所示:

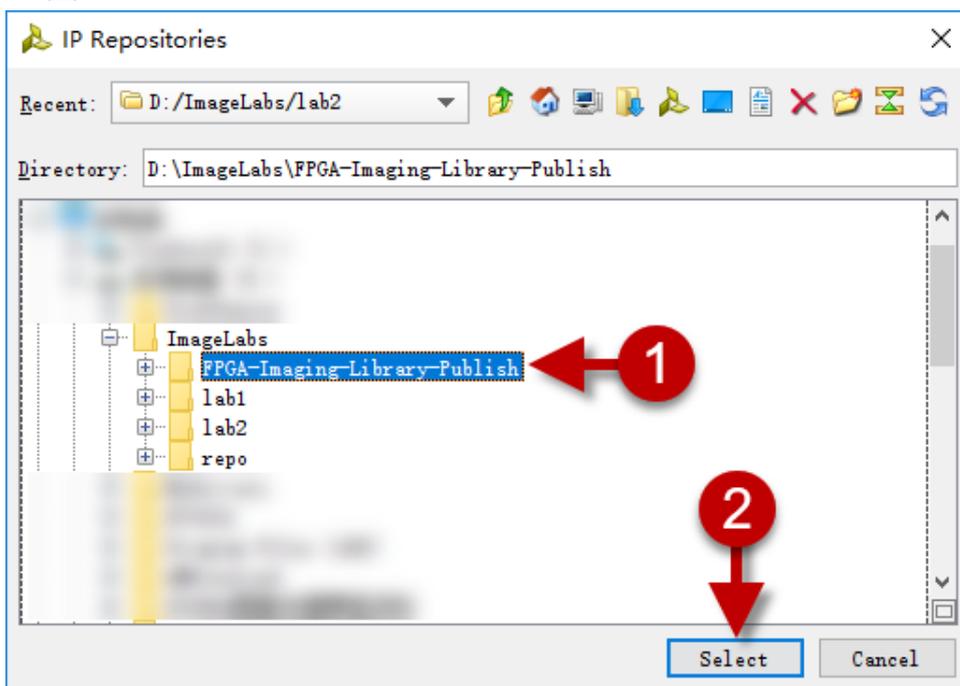


图 2-7 选择图像 IP 库

添加好 IP 库后, 能看到 Vivado 会自动扫描库中的 IP, 如果能看到如下图所示的一些 IP, 则表示 IP 库添加成功, 此时点击 OK 继续:

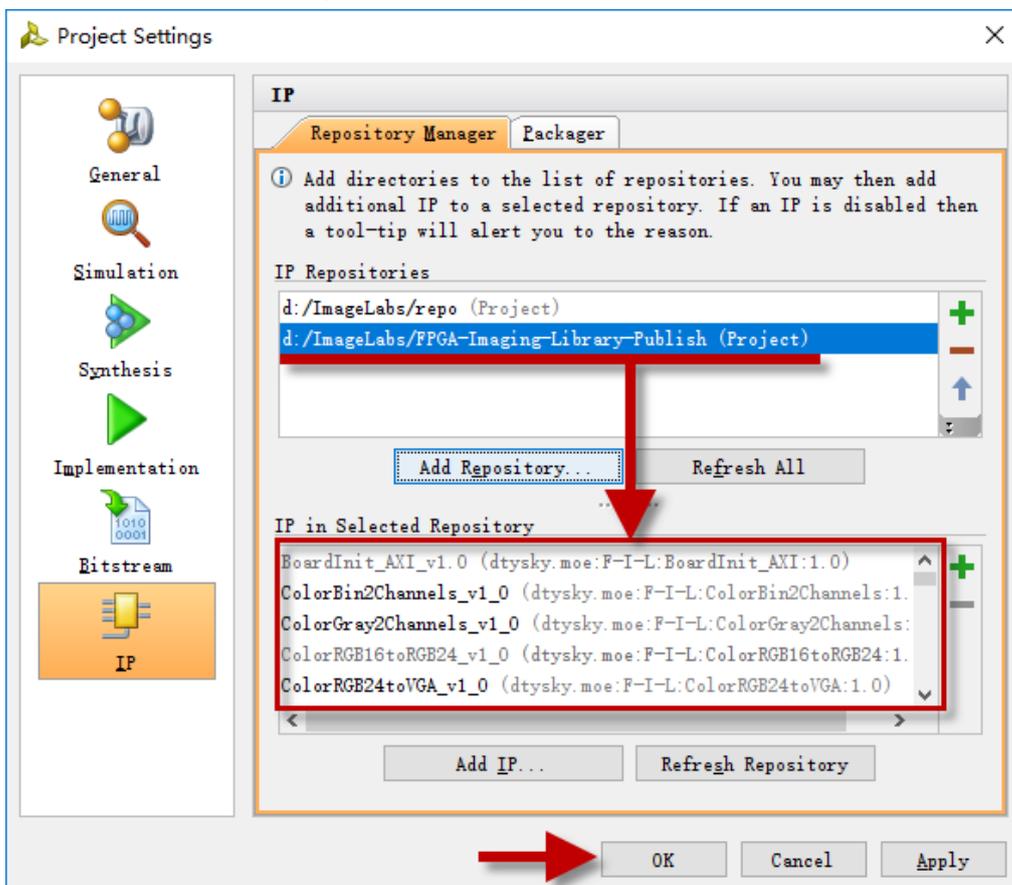


图 2-8 图像处理 IP 库中的 IP 列表显示

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	15 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

7. 回到 Vivado 的主界面后, 点击 Open Block Design, 这时会在主界面右边区域看到之前实验 1 的 IP 结构, 如下图所示:

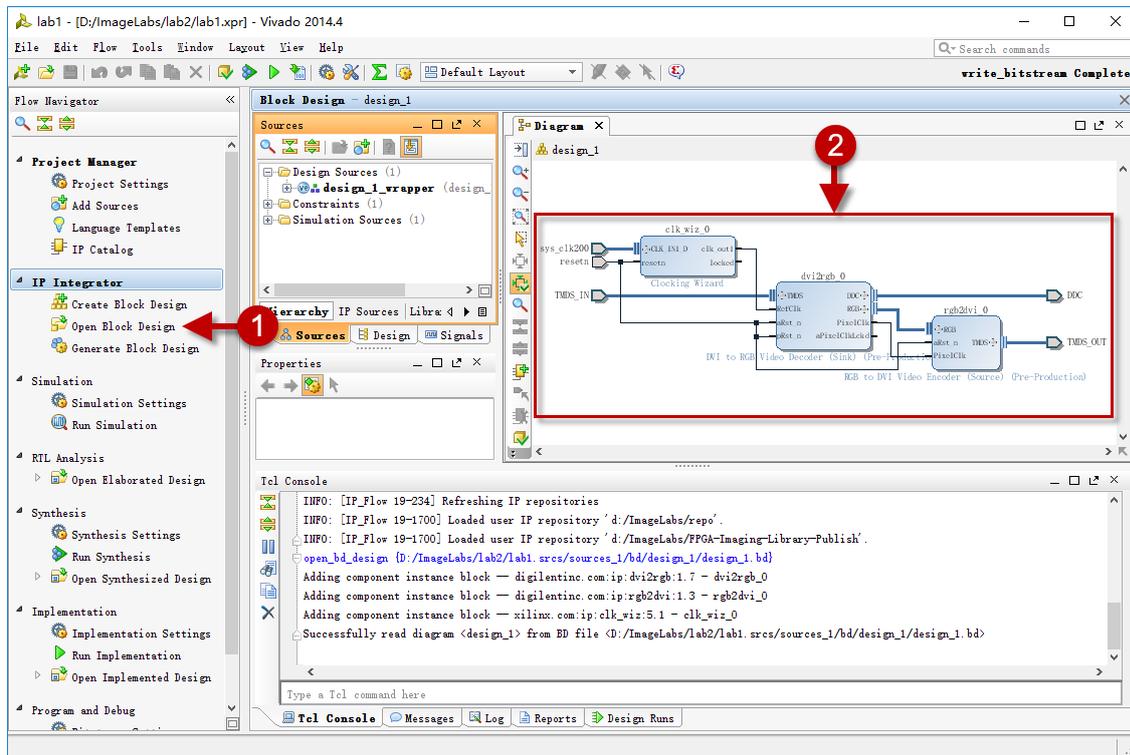


图 2-9 实验初始视图

8. 在此基础上, 我们开始添加 IP, 点击左边栏的 Add IP 图标, 然后在弹出的搜索框中, 输入 gray, 这时能看到搜索结果中有个 Graying 的 IP, 双击它进行添加, 整个过程如下图所示:

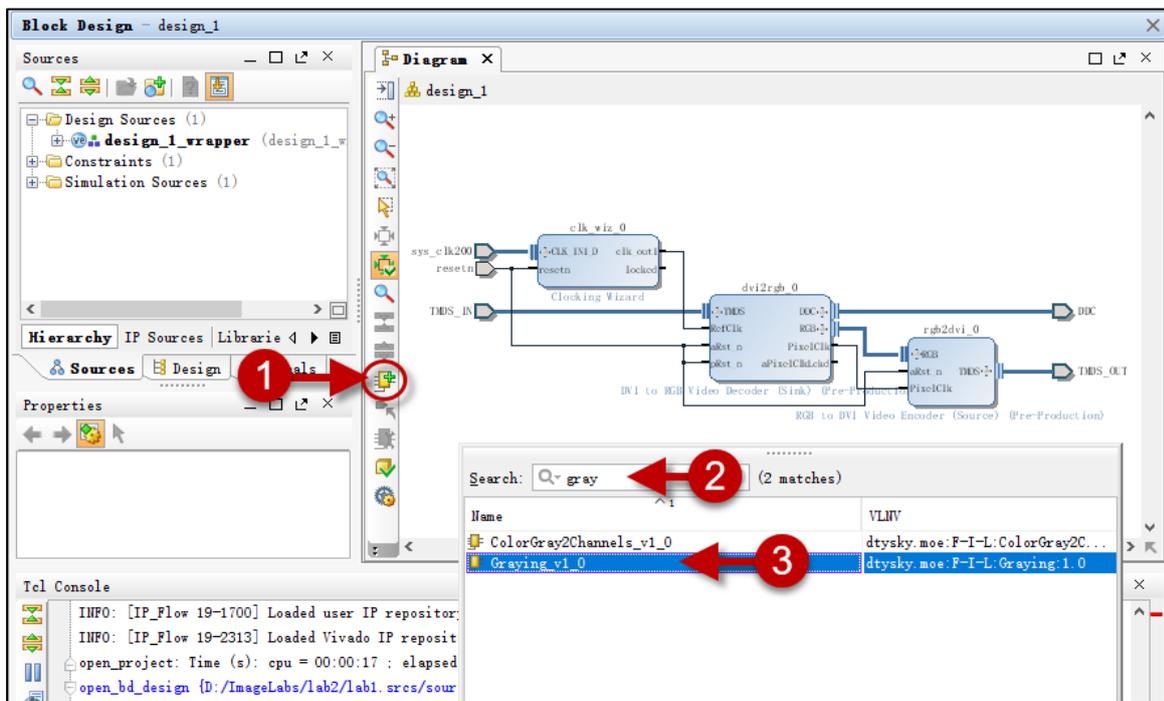


图 2-10 添加灰度化 IP

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	16 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

9. 由于我们要将灰度化 IP 接入到图像数据流中, 因此在模块化设计视图中, 我们需要现将之前的数据流先断开, 为此我们先用鼠标左键选中 DVI2RGB IP 的 RGB 端口, 此时会看到 RGB 信号高亮为浅黄色 (注意一定不要选中整个 IP), 然后鼠标右键单击, 在弹出菜单中选择 Disconnect Pin, 整个过程如下图所示:

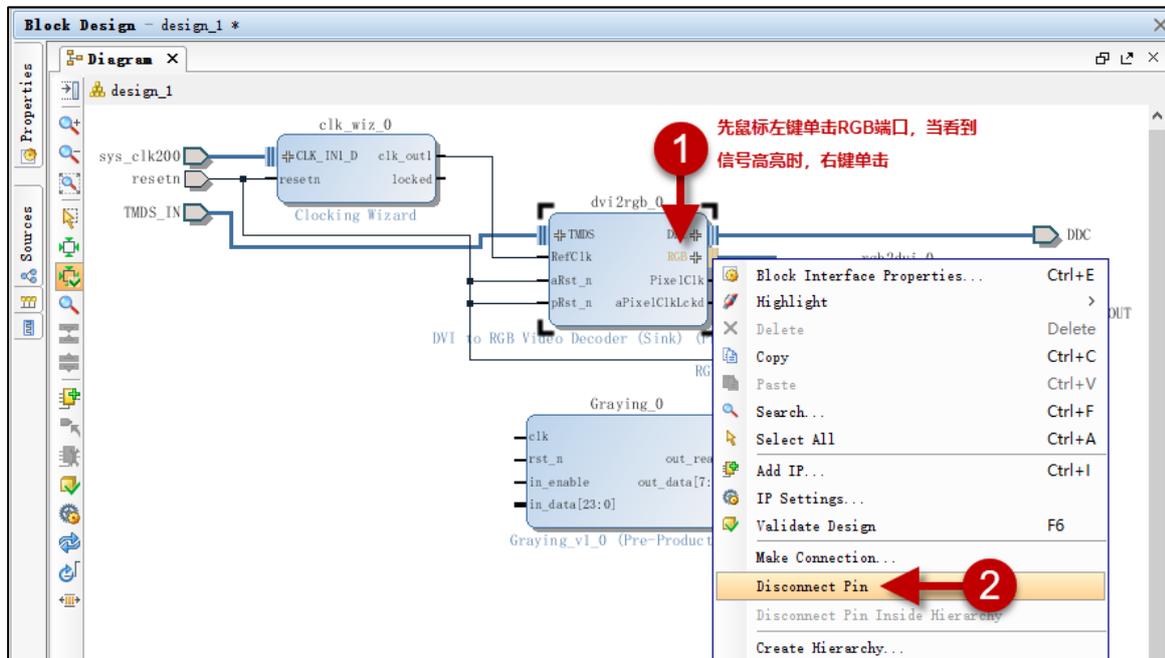


图 2-11 断开 DVI2RGB IP 的 RGB 端口

图像数据流断开后的 IP 视图如下图所示:

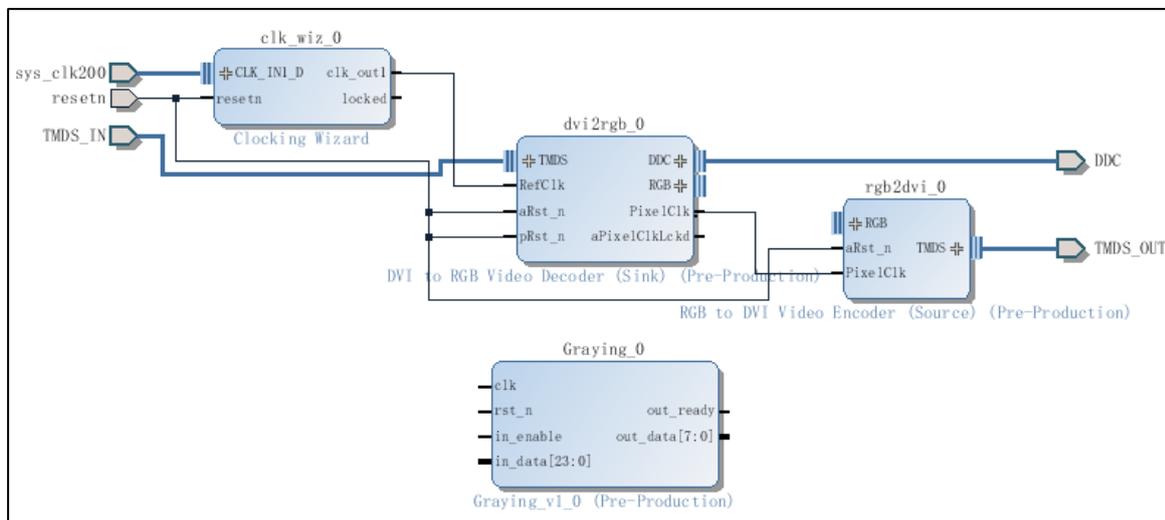


图 2-12 断开端口后视图

10. 接着我们开始将灰度化 IP 接入, 现将 DVI2RGB IP 的 RGB 端口展开, 将鼠标移动到 DVI2RGB 的 RGB 端口, 此时鼠标指针会变成一个  的符号, 如下图所示:

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	17 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

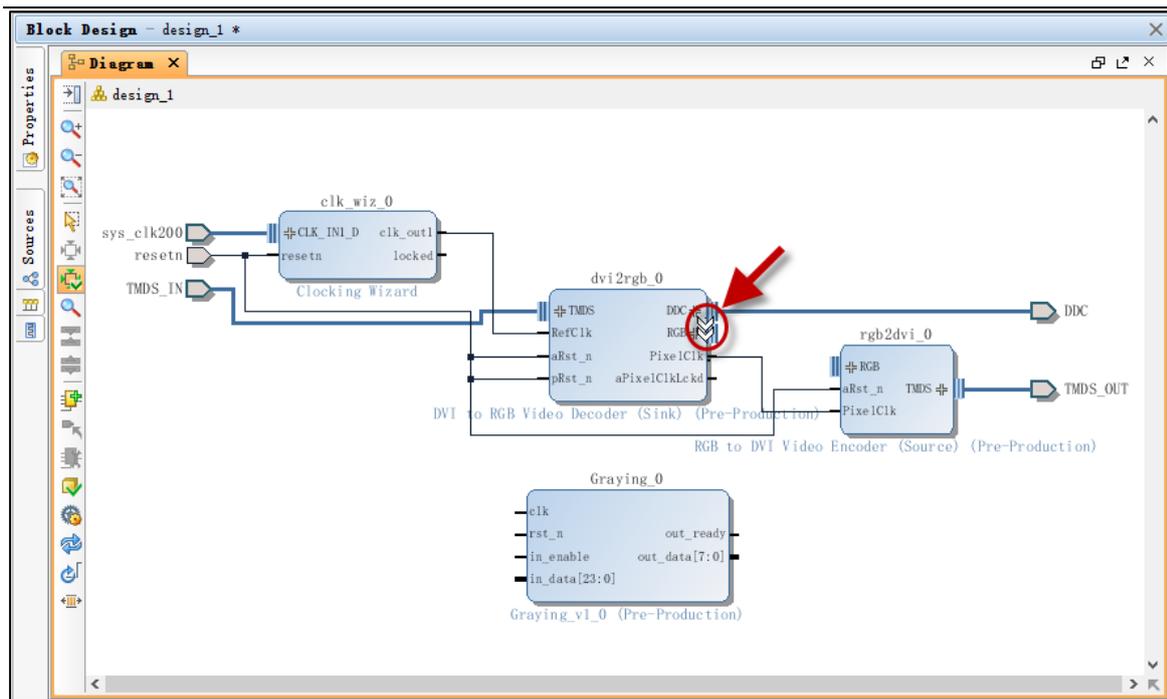


图 2-13 展开 DVI2RGB IP 的 RGB 端口

左键点击即可展开，如下图所示：

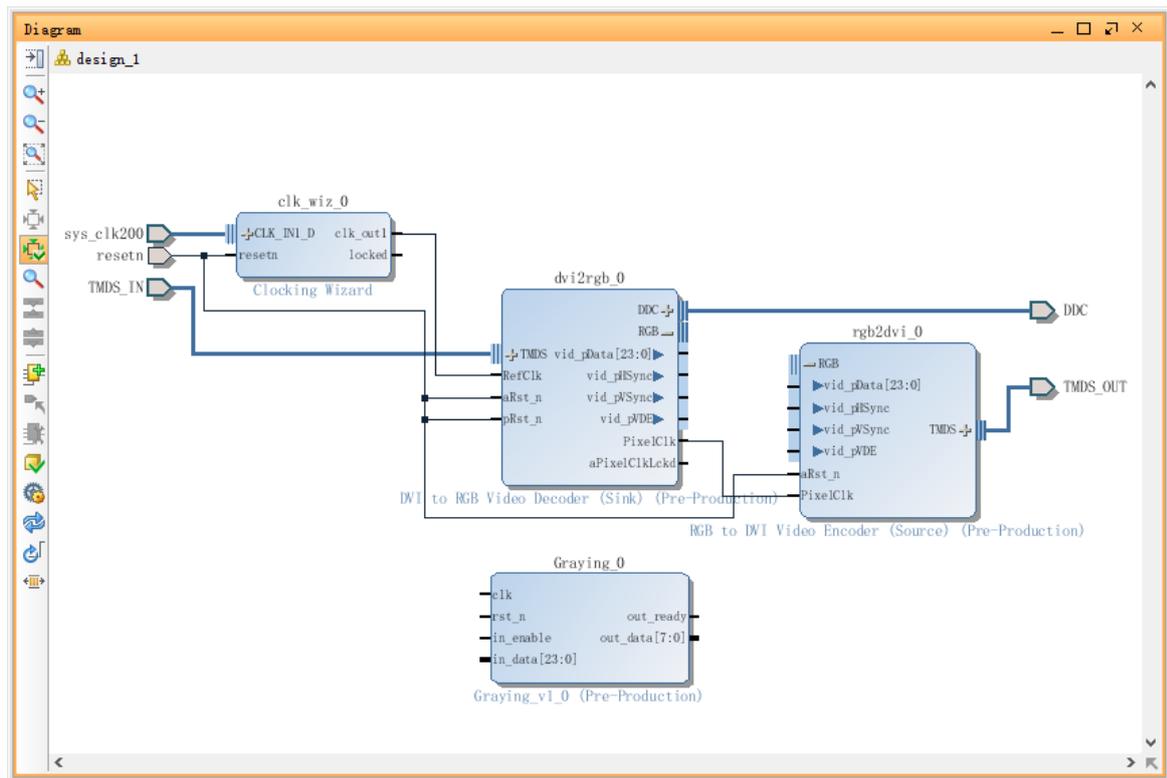


图 2-14 RGB 端口展开后视图

我们将按照如下方式先对 Graying IP 进行部分连接：

- Graying\_0:clk → dvi2rgb\_0:PixelClk
- Graying\_0:rst\_n → clk\_wiz\_0:resetn
- Graying\_0:in\_enable → dvi2rgb\_0:vid\_pVDE

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	18 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

连接的操作如下图所示，鼠标光标移动到对应端口，然后点击不放并拖拽到与之对应的连接端口上，如果符合连接规则，端口会自动显示绿色的勾。

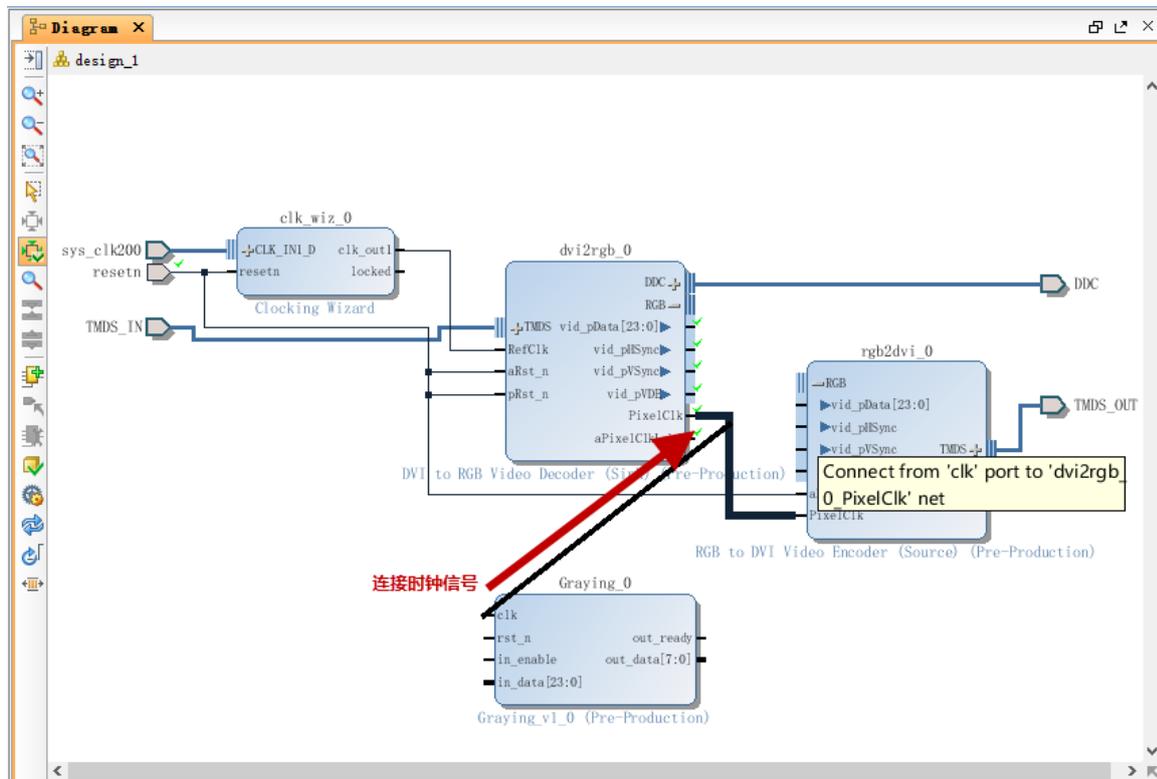


图 2-15 连接端口

11. 由于灰度化后的像素数据为 8 位宽,而显示输出端接收的数据宽度为 24 位宽,所以需要将 8 位转换成 24 位,我们直接调用一个 concat 的 IP 来实现转换,这个 IP 的作用是多端口的位宽拼接,点击 Add IP 图标,在弹出的搜索栏中输入 concat,然后在结果栏中双击 Concat 这个 IP 进行添加,整个过程如下图所示:

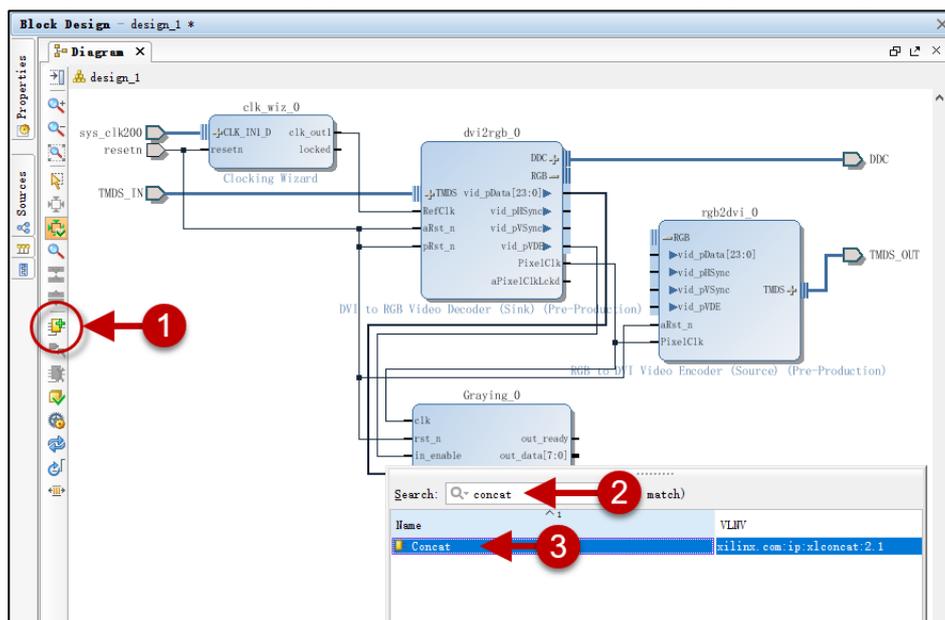


图 2-16 添加 Concat IP

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	19 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

添加后的 IP 视图如下图所示:

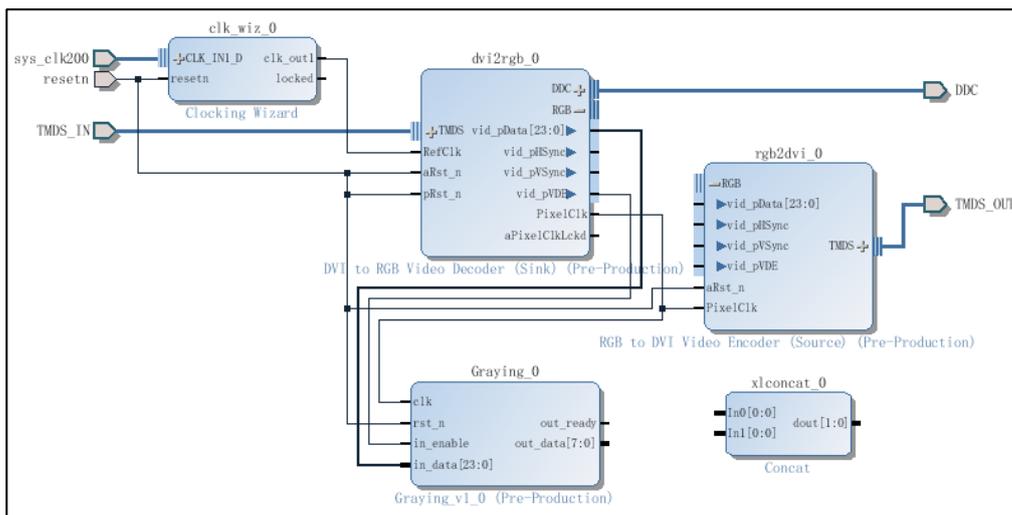


图 2-17 添加 Concat IP 后的视图

接着双击 xlconcat\_0 这个 IP 进行配置, 在配置对话框中, 进行如下设置:

- Number of Ports: 3
- Manual In0 Width: 8
- Manual In1 Width: 8
- Manual In2 Width: 8

完成设置后, 点击 OK 确认, 整个拖成如下图所示:

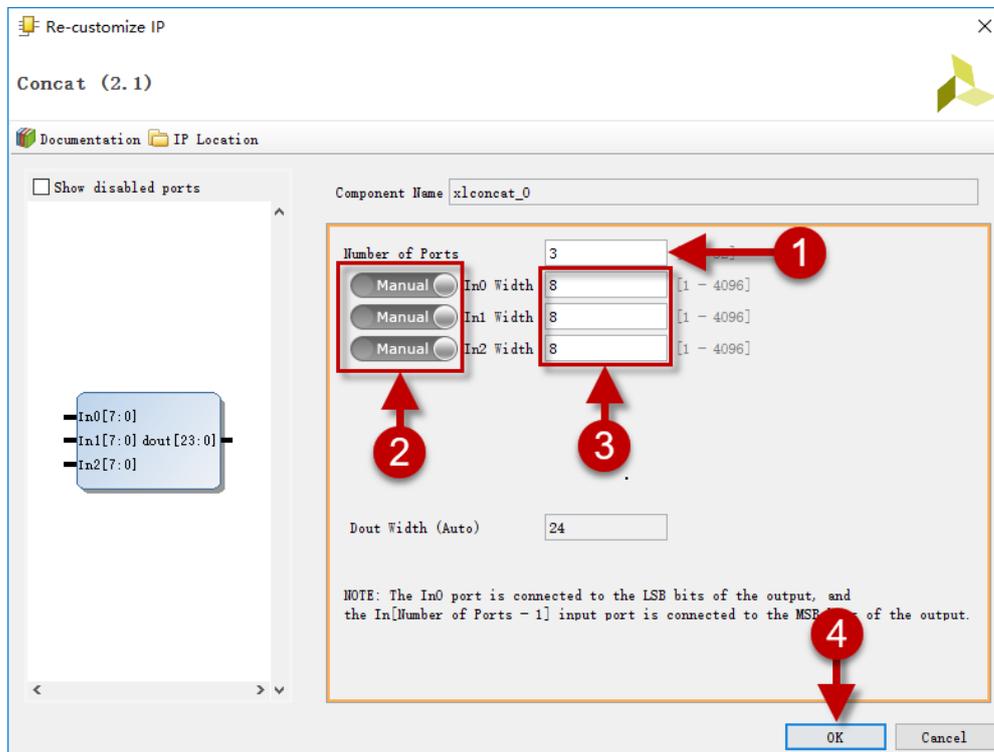


图 2-18 Concat IP 设置

12. 之后将 Graying\_0 和 xlconcat\_0 这两个 IP 进行连接, 连接方式如下:

Graying\_0:out\_ready → rgb2dvi\_0: vid\_pVDE

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	20 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

Graying\_0:out\_data[7:0] → xlconcat\_0:In0[7:0]  
 Graying\_0:out\_data[7:0] → xlconcat\_0:In1[7:0]  
 Graying\_0:out\_data[7:0] → xlconcat\_0:In2[7:0]  
 xlconcat\_0:dout[23:0] → rgb2dvi\_0:vid\_pData[23:0]  
 dvi2rgb\_0:vid\_pHSync → rgb2dvi\_0:vid\_pHSync  
 dvi2rgb\_0:vid\_pVSync → rgb2dvi\_0:vid\_pVSync

连接好后的 IP 视图如下图所示:

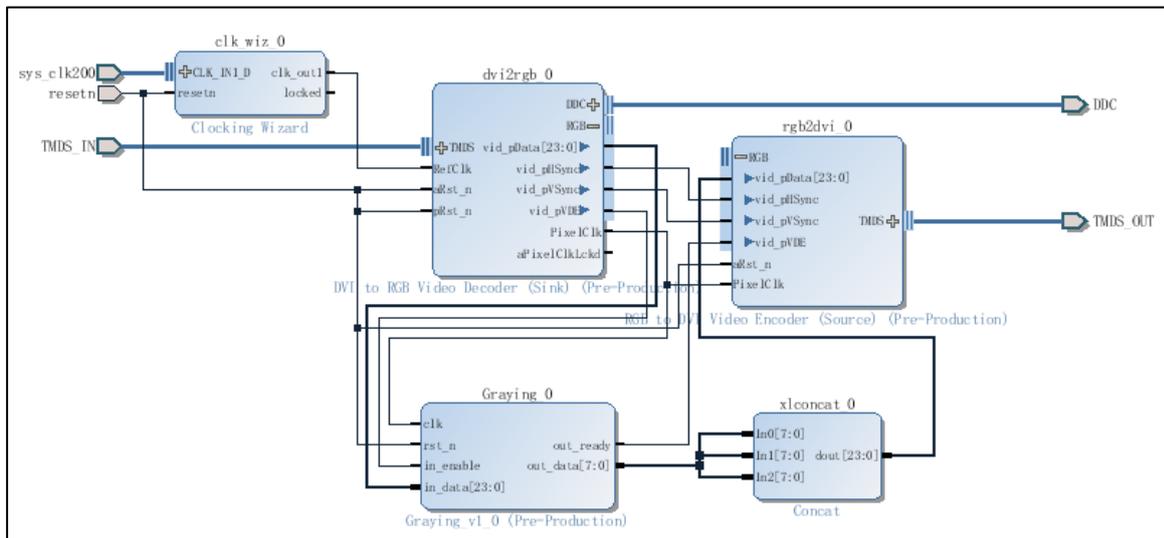


图 2-19 连接 Graying IP 后的视图

13. 请仔细检查各 IP 的端口连接是否正确, 为了方便核对, 本文特别准备了各种连接的高亮色图以示区别, 如下图所示:

**提示: 下图仅作为检查连接使用, 读者完全不必也按照图示颜色进行标注!!!**

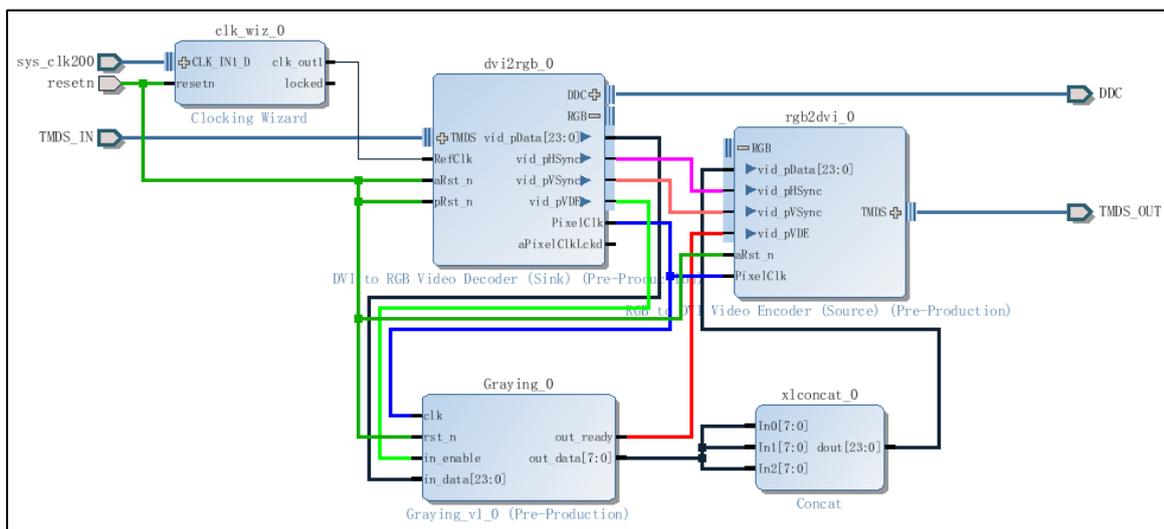


图 2-20 端口连接检查

14. 连接检查无误后, 即可保存 IP 模块化设计, 在 Vivado 主界面点击保存图标, 如下图所示:

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	21 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

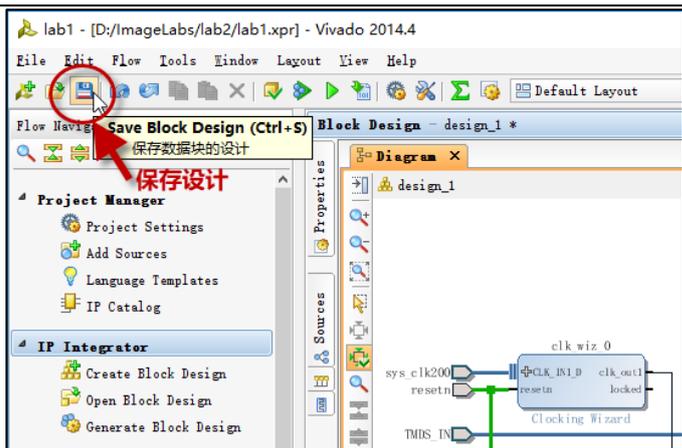


图 2-21 保存设计

接着在 Source 子窗口中展开 design\_1\_wrapper, 选中 design\_1.bd, 鼠标右键单击, 在弹出的菜单中选择 Create HDL Wrapper, 整个过程如下图所示:

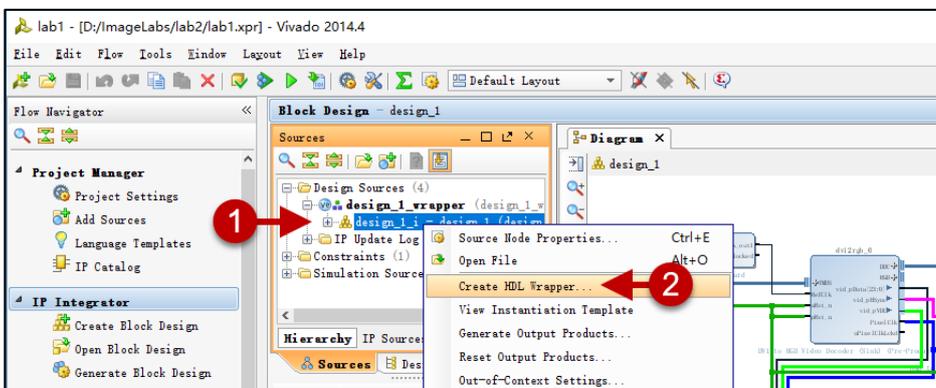


图 2-22 创建实验顶层 Wrapper 文件

接着在弹出的对话框中, 保持默认选项不变, 即选择 Let Vivado manage wrapper and auto-update, 然后点击 OK, 如下图所示:

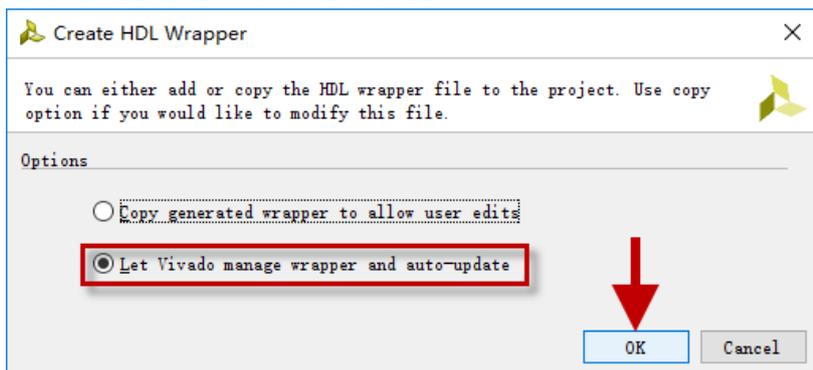


图 2-23 自动更新顶层文件

在 Vivado 主界面点击 Generate Bitstream, 生成 bit 文件, 如下图所示:

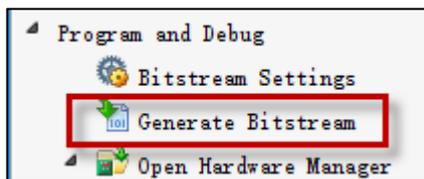


图 2-24 Generate Bitstream

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	22 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

在弹出的提示框中直接点 Yes 确认并继续，如下图所示：

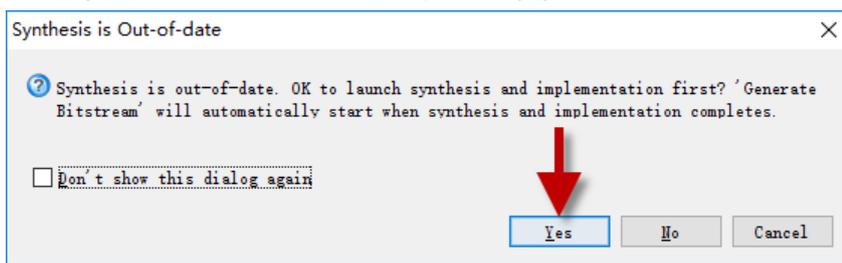


图 2-25 点击 Yes 确认生成 bit 文件

大约经过 10 分钟后，Vivado 会弹出 Bitstream Generation Completed 的提示框，表示 bit 文件完成，选择 Open Hardware Manager，然后点击 OK，如下图所示：

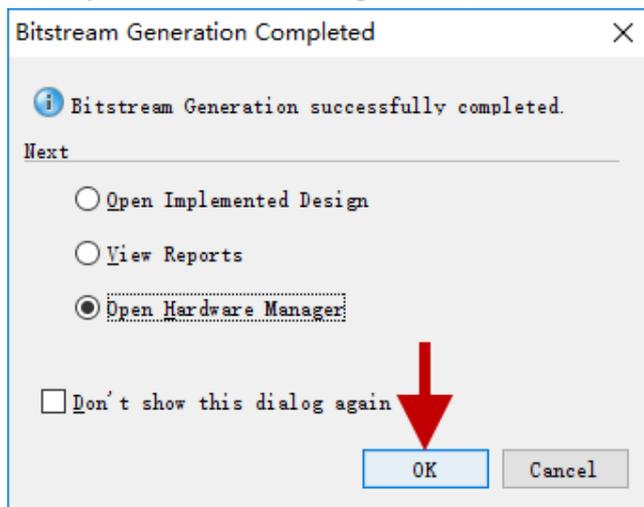


图 2-26 打开 Hardware Manager

接着我们需要对 SWORD4.0 硬件平台进行连接，根据下图示意依次进行如下操作：

- 1) 将电源线接上 SWORD4.0，注意此时 SWORD4.0 的开关不要打开；
- 2) 将下载器模块插到 SWORD4.0 的 CN7-JTAG 处，并将下载器的 USB 端口连到电脑；
- 3) 用一根 HDMI 线将 SWORD4.0 和 HDMI 信号源连接上；
- 4) 用一根 HDMI 线将 SWORD4.0 和 HDMI 显示器连接上；
- 5) 打开电源开关

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	23 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

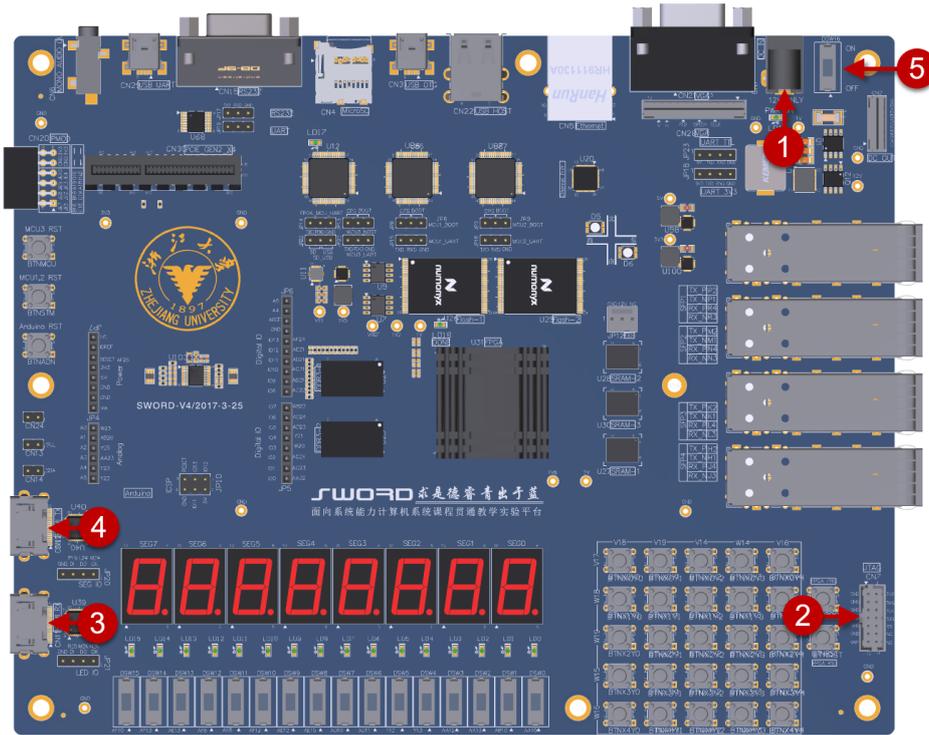


图 2-27 硬件连接对应位置

连接好后的效果如下图所示:



图 2-28 实际硬件连接

15. 接着在 Hardware Manager 界面下, 点击 Open target, 在随之弹出的菜单中选择 Auto Connect, 整个过程如下图所示:

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	24 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

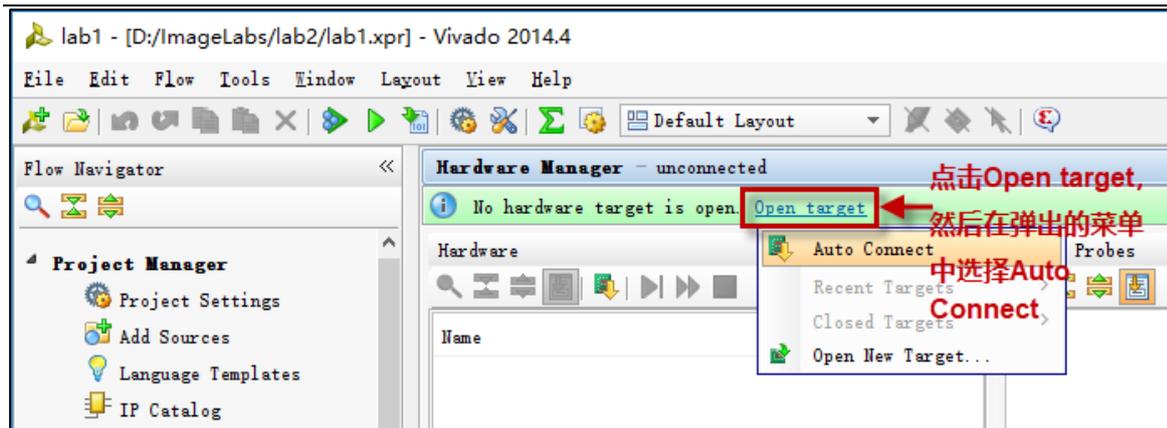


图 2-29 Open target

接着 Hardware Manager 会自动连接下载器并扫描 JTAG，一切正常的话，会显示出扫描到的目标器件：xc7k325t，鼠标右键单击目标器件，在弹出的窗口中选择 Program Device，整个过程如下图所示：

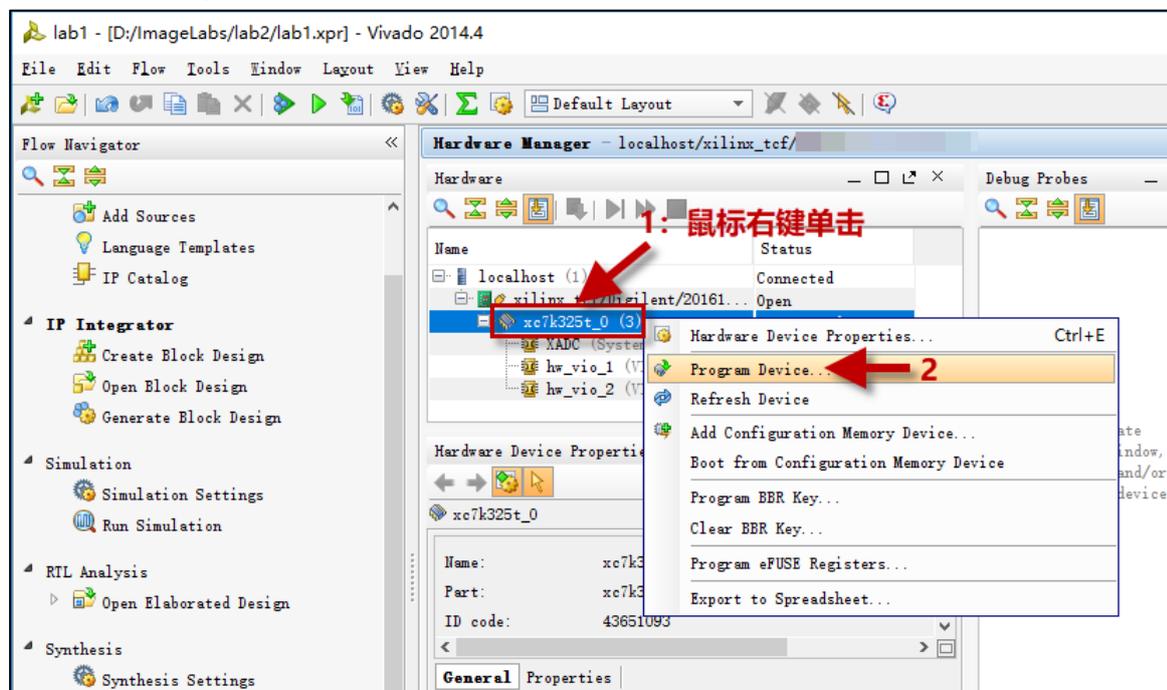


图 2-30 Program Device

16. 在弹出的对话框中，保持默认设置，直接点击 Program，如下图所示：

提示：如果 Debug probe file 这一栏有输入，可忽略之。

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	25 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

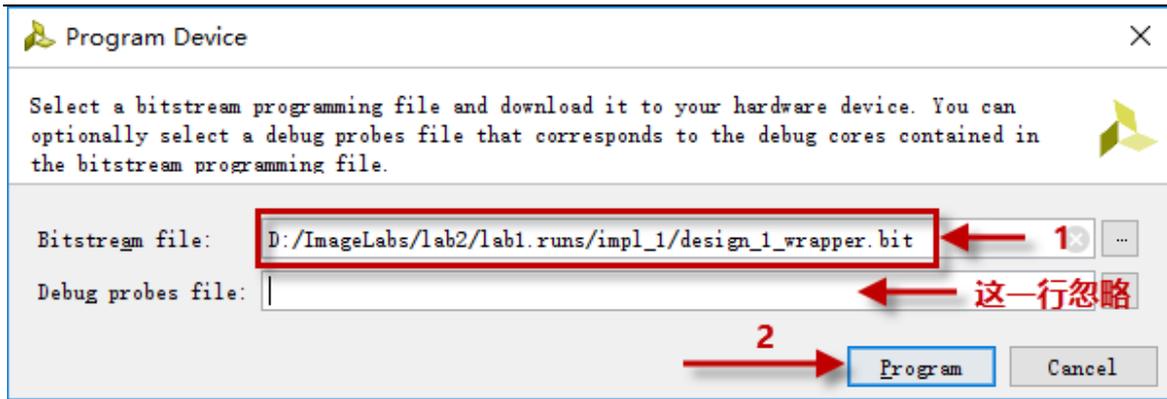


图 2-31 烧写目标器件

随着如下图所示进度条显示 100%，即表示目标器件烧写完毕。即可进入实验现象观察阶段。

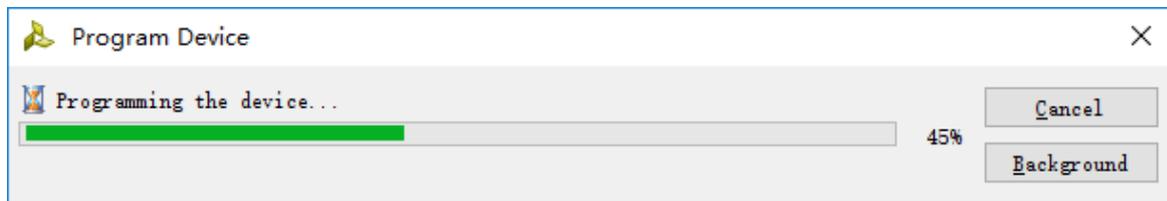


图 2-32 编程进度条

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	26 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

### 3. PARTA: 灰度化实验结果

首先我们让 HDMI 信号源显示一幅素材图片 (位于同文件夹下的 J20.png), 接着我们将连接 HDMI 输入端口的 HDMI 线在信号源端重新插拔一次, 以便让信号源设备重新检测 (Detect) 一下接收设备, 一切正常的话, 我们即可在 HDMI 显示器上看到显示画面。



图 3-1 灰度化显示结果

<b>XINGDENG</b>	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	27 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

## 4. PARTB: 二值化实验流程

### 4.1 操作步骤

1. 完成了灰度化部分的实验后，我们继续进行二值化部分的时候，回到 Vivado 的主界面，点击 Open Block Design 图标，然后在打开的 IP 设计视图的左边栏，点击 Add IP 的图标，过程如下图所示：

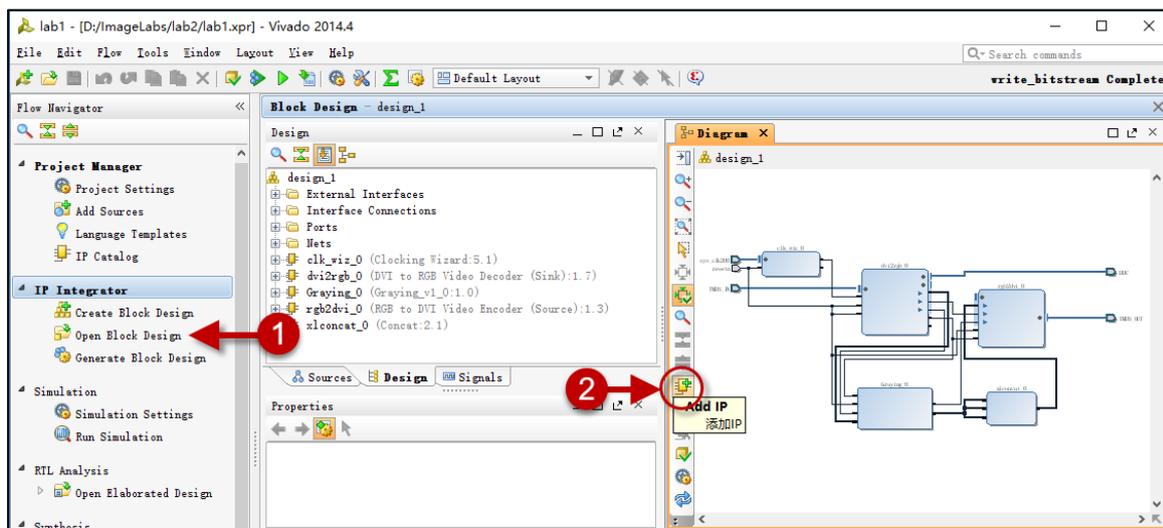


图 4-1 回到 Vivado 主界面继续添加 IP

在搜索栏中输入 Thresh，在搜索结果中双击 Threshold\_v1\_0，添加二值化 IP，过程如下图所示：

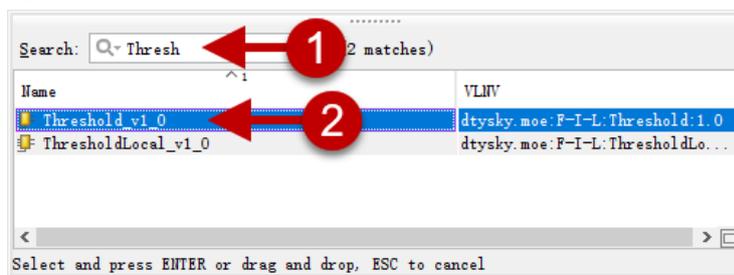


图 4-2 添加二值化 IP: Threshold

2. 添加 Thresh IP 后，和前面部分的实验一样，为了将这个 IP 接入数据流，我们需要断开之前的图像数据流，为此，鼠标选中 RGB2DVI IP 的 vid\_pVDE，当该端口高亮为浅黄色之后，鼠标右键单击，在弹出的菜单中选择 Disconnect Pin，过程如下图所示：

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	28 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

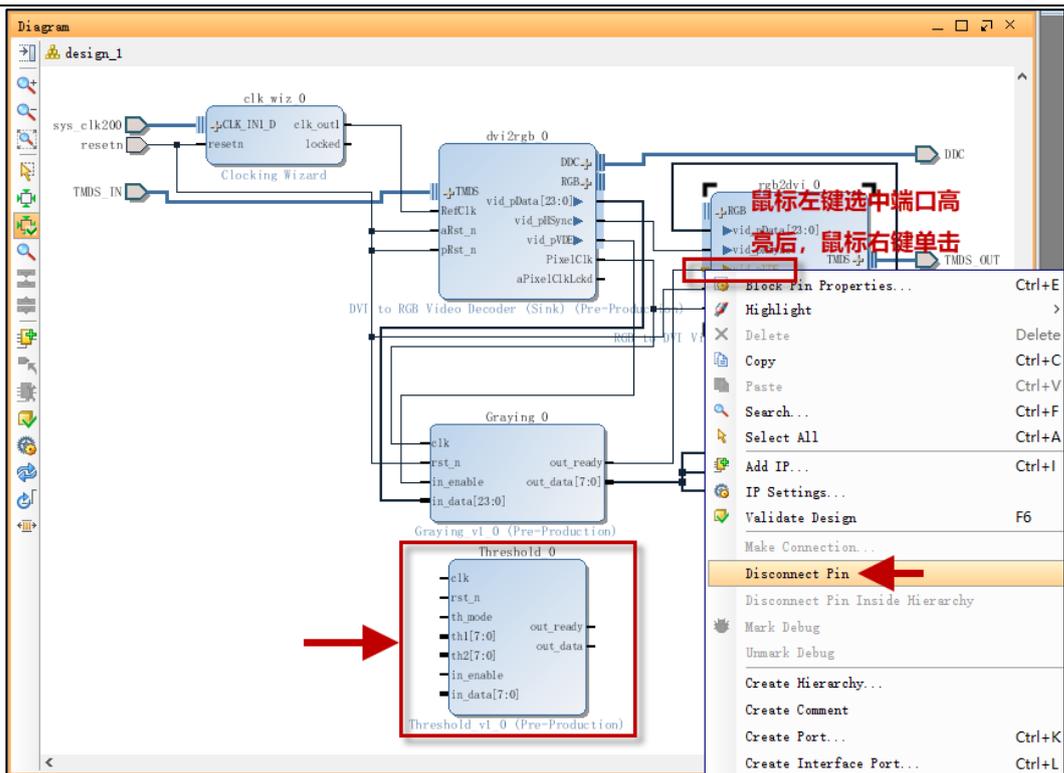


图 4-3 断开 RGB2DVI IP 的 vid\_pVDE 端口

- 按同样的方式, 断开 RGB2DVI IP 的 vid\_pData[23:0]端口和 Concat IP 的 In0, In1, In2 端口, 断开后的效果如下图所示:

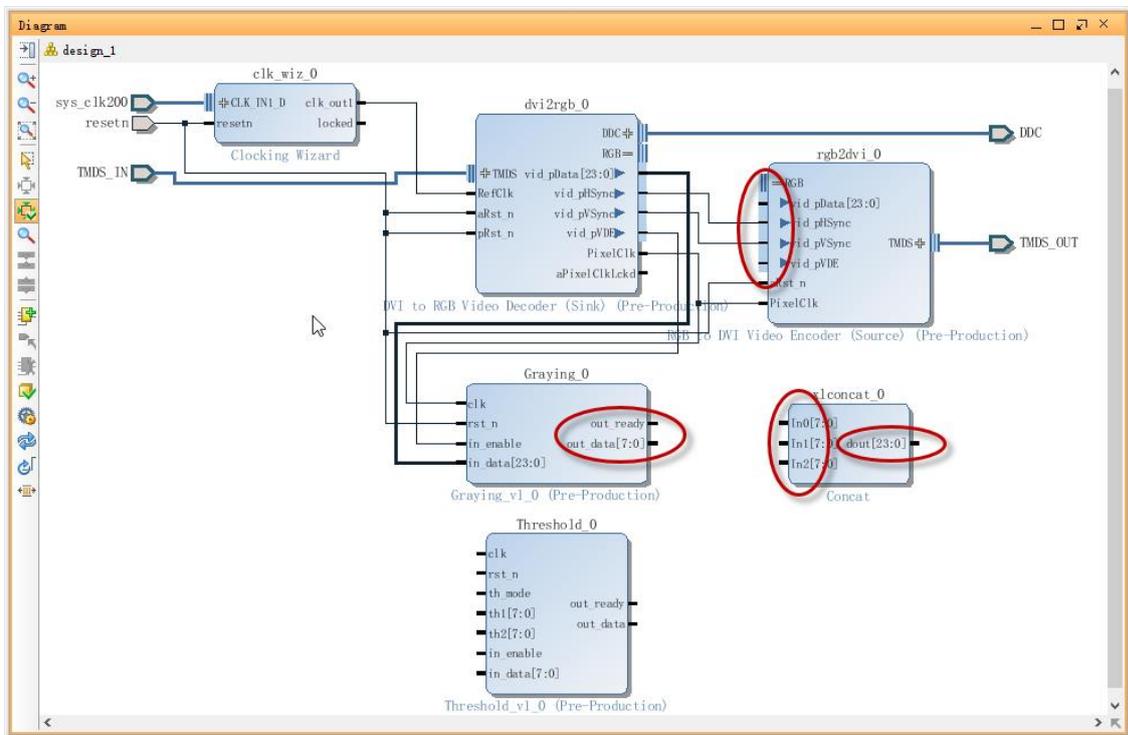


图 4-4 断开数据流后的 IP 视图

- 双击 Thresh IP 进行配置, 在配置窗口中, 将 Color Width 设置为 8, 点击 OK 确认, 如下图所示:

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	29 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

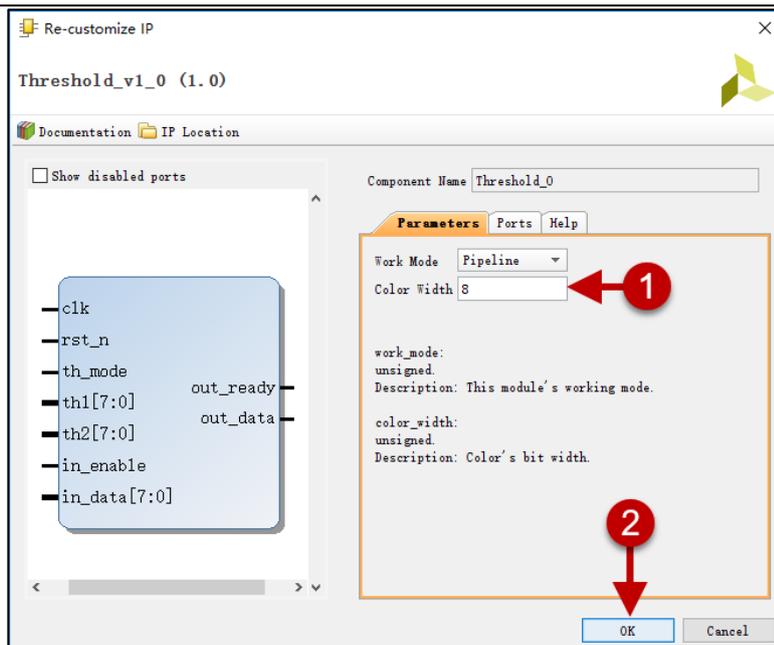


图 4-5 Thresh IP 配置

5. 双击 xlconcat\_0 IP 进行配置，在配置窗口中，将 Number of Ports 改为 24，然后将下面的每个输入端口模式都改为 Manual，Width 设置为 1，完成设置后点 OK 继续，如下图所示

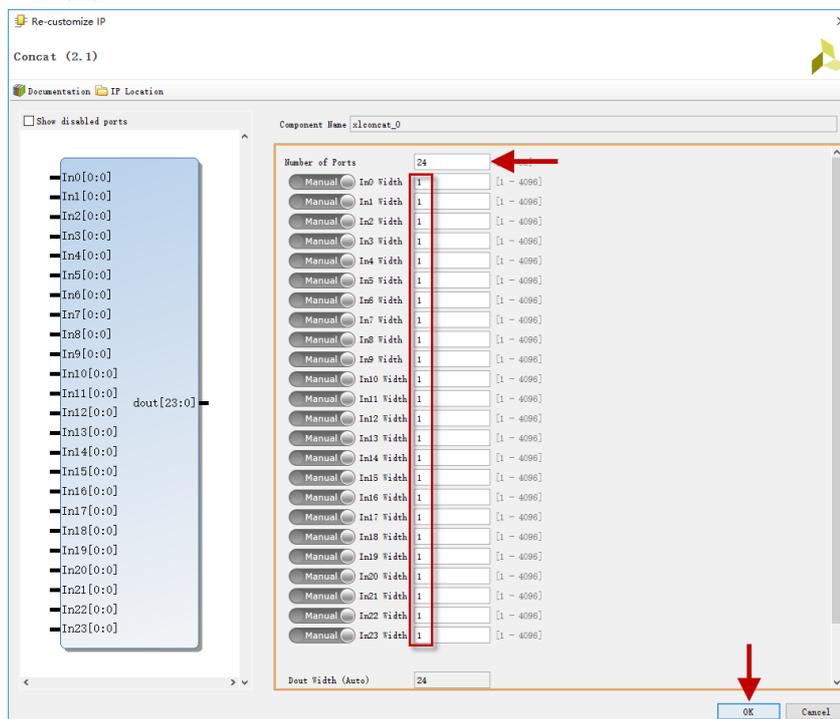


图 4-6 Concat IP 设置

6. 完成对 Thresh 和 xlconcat 这两个 IP 的配置后，进行 IP 的连接，连接后的效果如下图所示，请仔细检查各 IP 的端口连接是否正确，为了方便核对，下图各种连接的高亮色图以示区别：

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	30 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

**提示：下图仅作为检查连接使用，读者完全不必也按照图示颜色进行标注!!!**

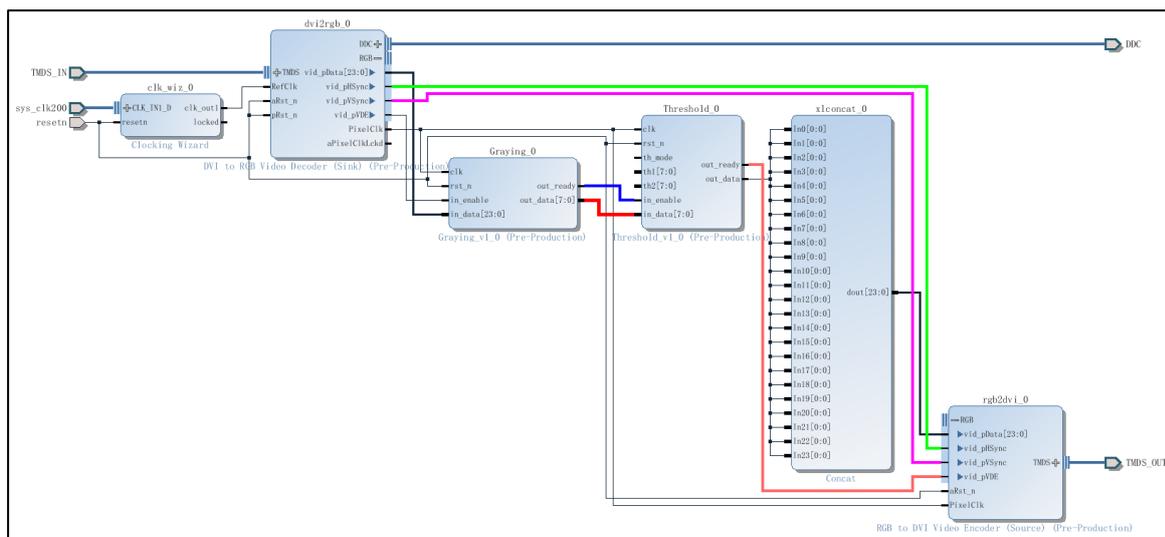


图 4-7 端口连接检查

7. 注意到上图中 Thresh IP 还有 3 个端口未连接，分别是 th\_mode，th1 和 th2。根据 IP 的端口定义，th\_mode 为阈值模式 (0 为全局模式，1 为等高线模式)，th1 为全局阈值输入，th2 为等高线阈值输入。本实验中我们使用的是全局模式。因此我们添加 2 个 constant IP 来作为输入，连接 th\_mode 和 th1。
8. 点击 IP 视图左边栏的 Add IP 图标，在弹出的搜索栏中输入 constant，在搜索结果中**双击 Constant 两次**，以添加 2 个这样的 IP，如下图所示：

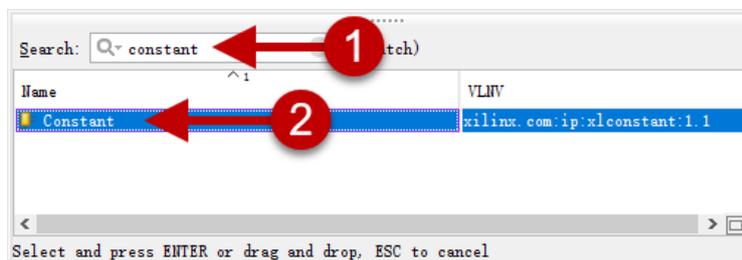


图 4-8 添加 Constant IP

9. 添加好后，在 IP 设计视图中双击 xlconstant\_0 这个 IP，这个 IP 是作为阈值模式的输入。在配置对话框中将 Const Width 设置为 1，将 Const Val 设置为 0，完成设置后，点击 OK 继续，如下图所示；

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	31 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

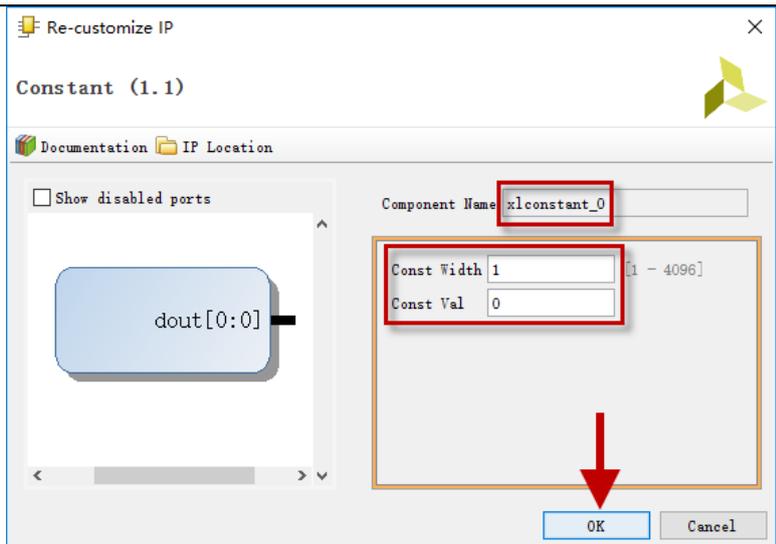


图 4-9 第一个 Constant IP 设置

- 10. 接着在 IP 设计视图中双击 xlconstant\_1 这个 IP,这个 IP 是作为全局阈值的输入。在配置对话框中将 Const Width 设置为 8, 将 Const Val 设置为 200, 完成设置后, 点击 OK 继续, 如下图所示;

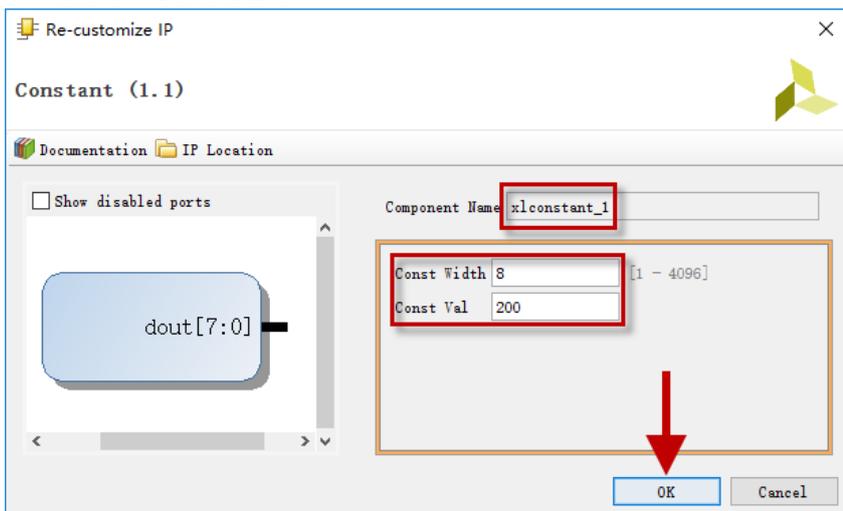


图 4-10 第二个 Constant IP 设置

完成设置后, 将 xlconstant\_0 和 xlconstant\_1 两个 IP 连接到 Thresh IP, 连接方式如下图所示:

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	32 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

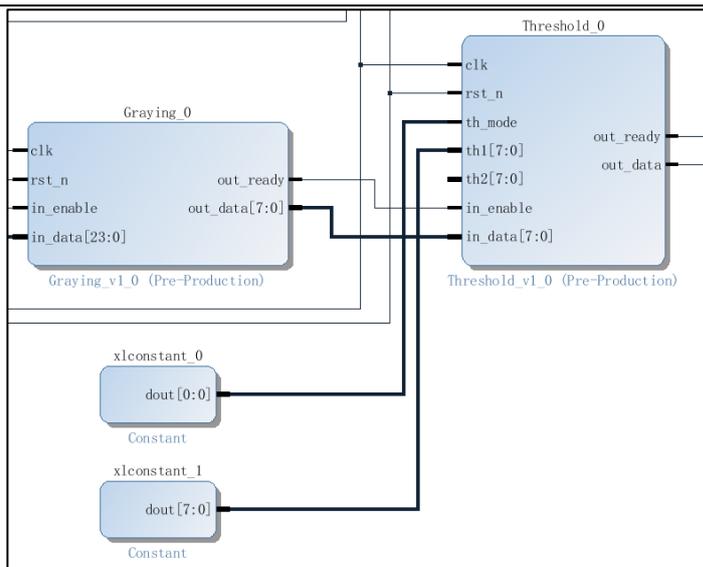


图 4-11 两个 Constant IP 的连接

11. 连接检查无误后，即可保存 IP 模块化设计，在 Vivado 主界面点击保存图标，如下图所示：

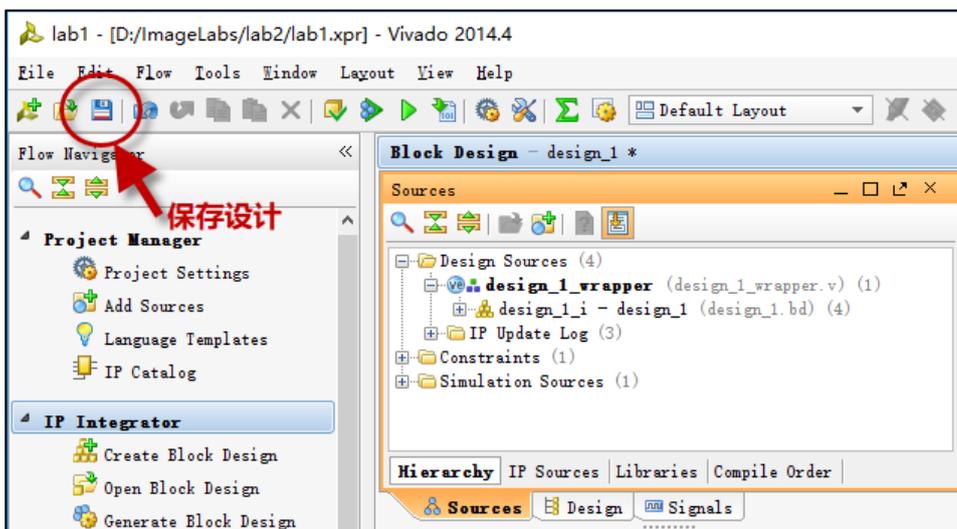


图 4-12 保存设计

12. 接着在 Source 子窗口中展开 design\_1\_wrapper，选中 design\_1.bd，鼠标右键单击，在弹出的菜单中选择 Create HDL Wrapper，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	33 of 40
	作者	修改日期	公开	
Joseph Xu	2019/2/14			

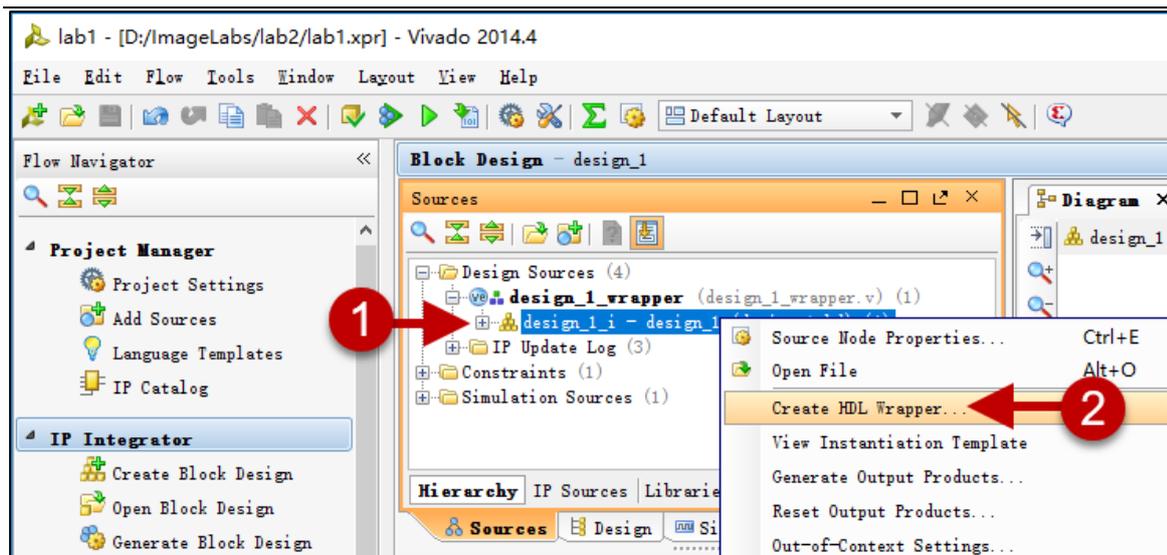


图 4-13 创建顶层文件

13. 接着在弹出的对话框中,保持默认选项不变,即选择 Let Vivado manage wrapper and auto-update, 然后点击 OK, 如下图所示:

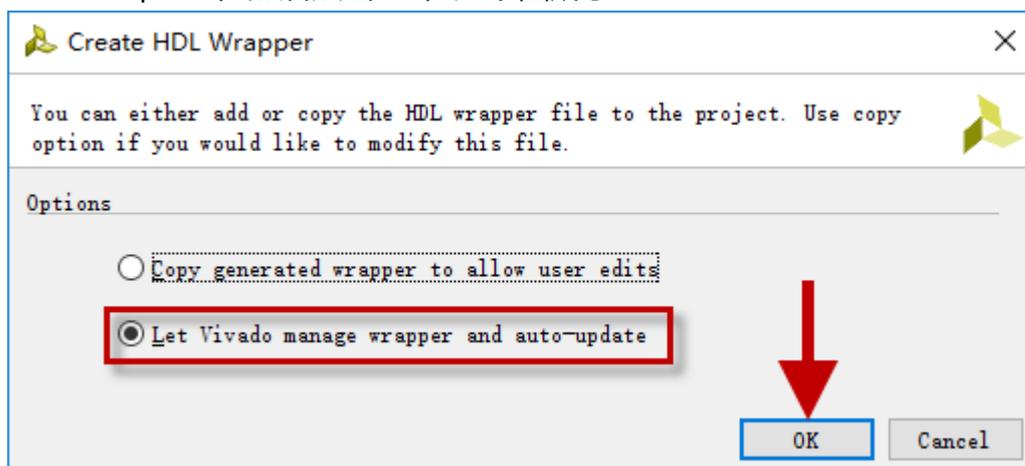


图 4-14 自动更新顶层文件

14. 在 Vivado 主界面点击 Generate Bitstream, 生成 bit 文件, 如下图所示:

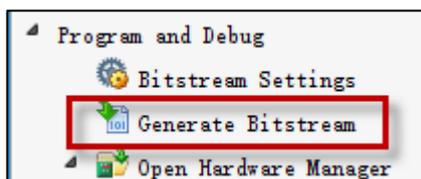


图 4-15 Generate Bitstream

在弹出的提示框中直接点 Yes 确认并继续, 如下图所示:

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	34 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

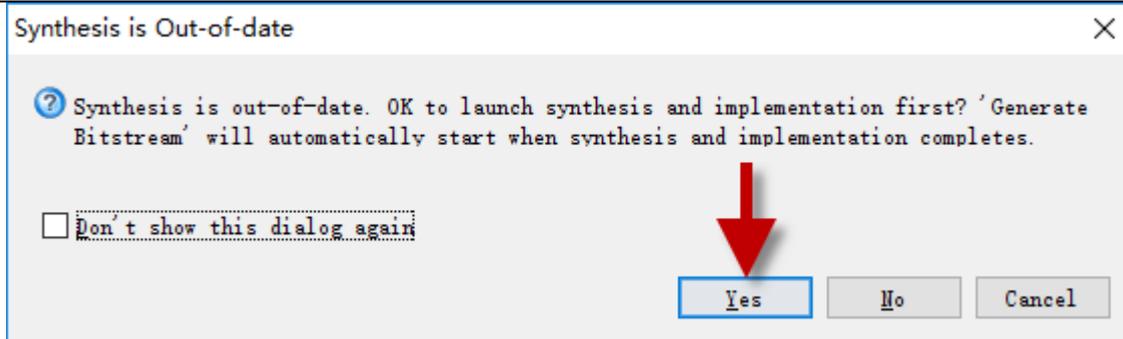


图 4-16 点击 Yes 确认生成 bit 文件

之后会有一个提示框，提示 th2 未连接，忽略这条信息，直接点 OK 继续，如下图所示：

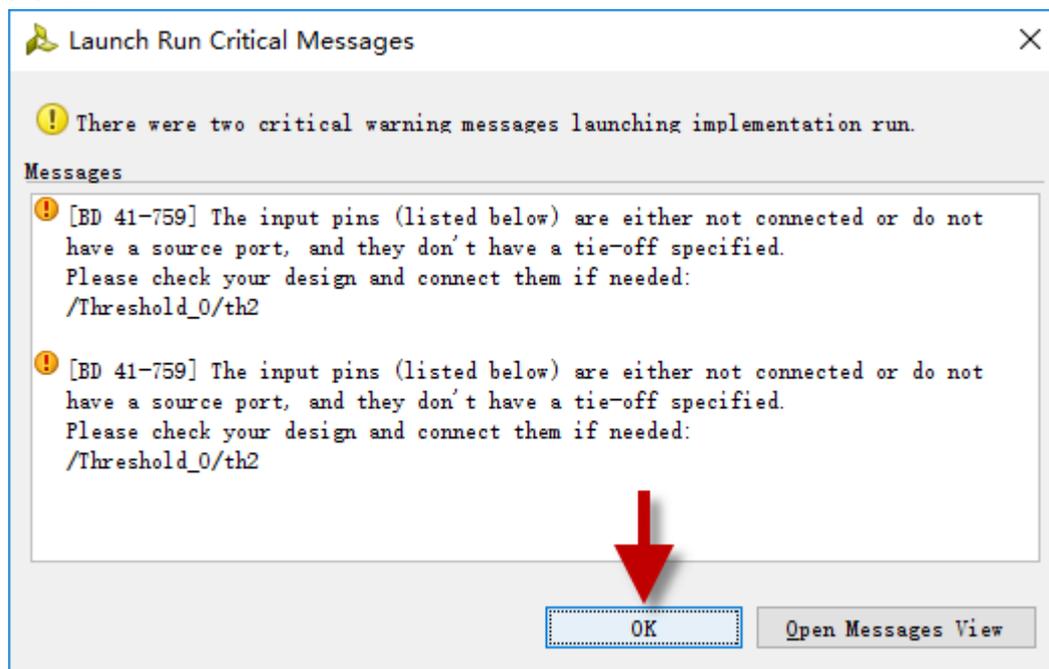


图 4-17 忽略提示信息并点击 OK

大约经过 10 分钟后，Vivado 会弹出 Bitstream Generation Completed 的提示框，表示 bit 文件完成，选择 Open Hardware Manager，然后点击 OK，如下图所示：

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	35 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

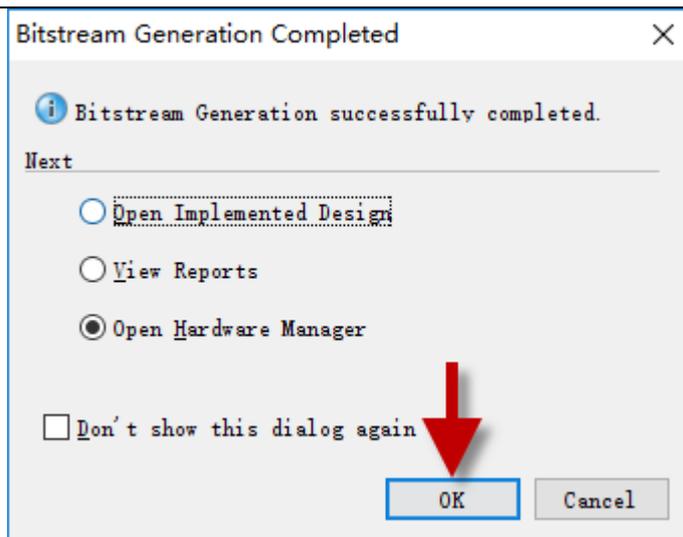


图 4-18 打开 Hardware Manager

15. 接着我们需要对 SWORD4.0 硬件平台进行连接，根据下图示意依次进行如下操作：

- 1) 将电源线接上 SWORD4.0，注意此时 SWORD4.0 的开关不要打开；
- 2) 将下载器模块插到 SWORD4.0 的 CN7-JTAG 处，并将下载器的 USB 端口连接到电脑；
- 3) 用一根 HDMI 线将 SWORD4.0 和 HDMI 信号源连接上；
- 4) 用一根 HDMI 线将 SWORD4.0 和 HDMI 显示器连接上；
- 5) 打开电源开关

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	36 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

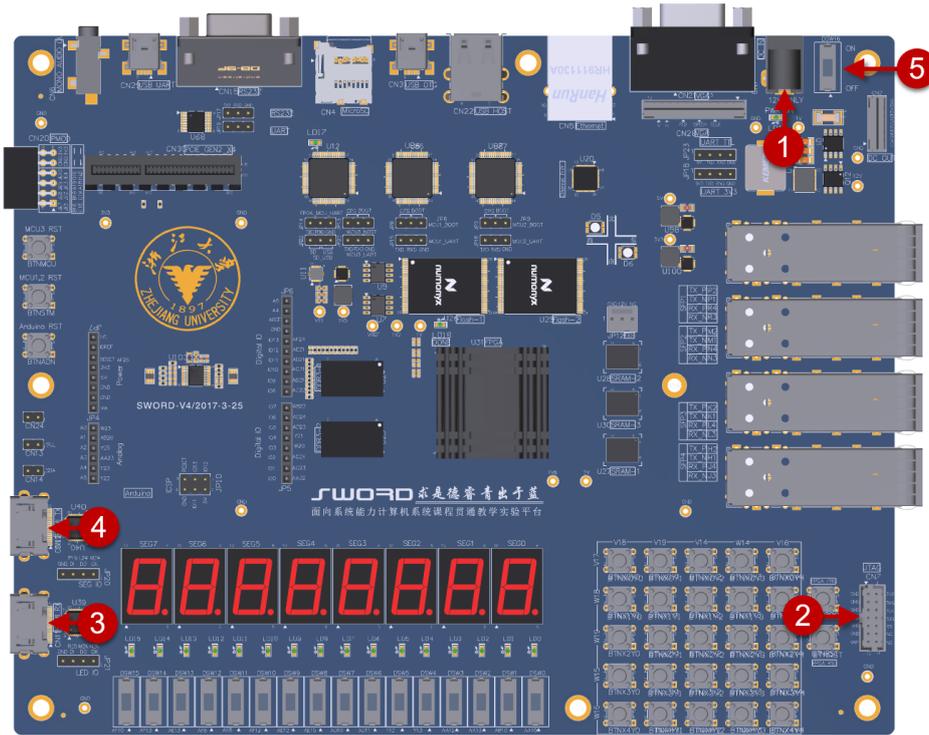


图 4-19 硬件连接对应位置

连接好后的效果如下图所示：



图 4-20 实际硬件连接

接着在 Hardware Manager 界面下，点击 Open target，在随之弹出的菜单中选择 Auto Connect，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	37 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

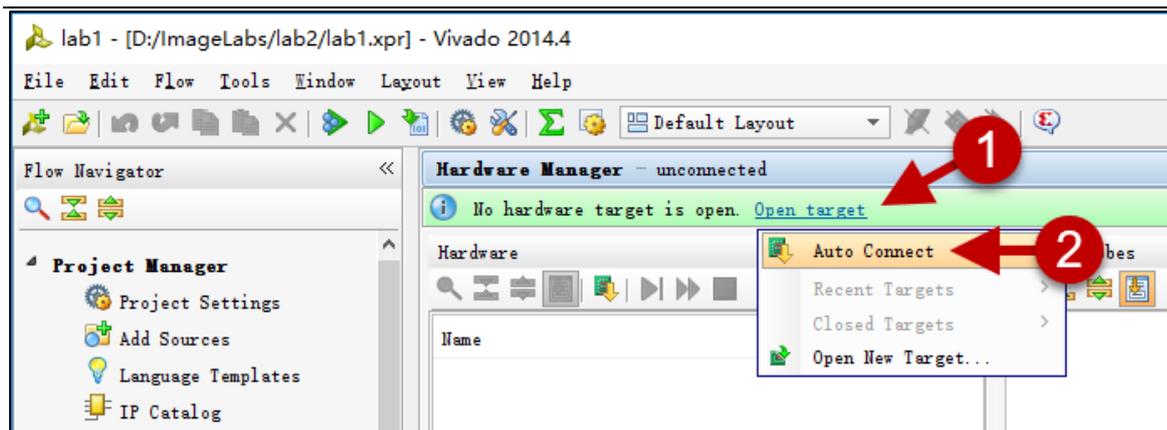


图 4-21 Open target

接着 Hardware Manager 会自动连接下载器并扫描 JTAG，一切正常的话，会显示出扫描到的目标器件：xc7k325t，鼠标右键单击目标器件，在弹出的窗口中选择 Program Device，整个过程如下图所示：

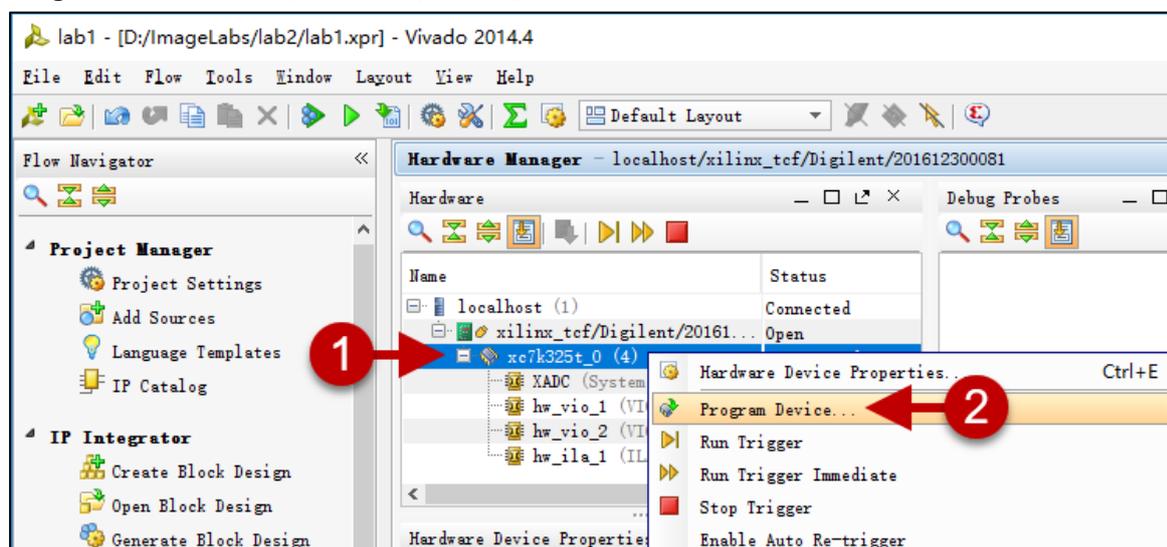


图 4-22 Program Device

在弹出的对话框中，保持默认设置，直接点击 Program，如下图所示：

提示：如果 Debug probe file 这一栏有输入，可忽略之。

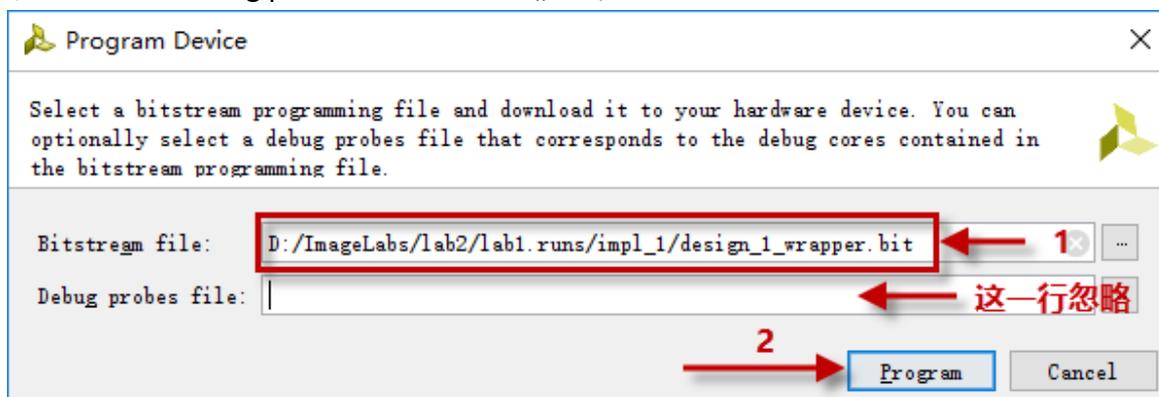


图 4-23 烧写目标器件

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	38 of 40
	作者	修改日期		
	Joseph Xu	2019/2/14	公开	

随着如下图所示进度条显示 100%，即表示目标器件烧写完毕。即可进入实验现象观察阶段。

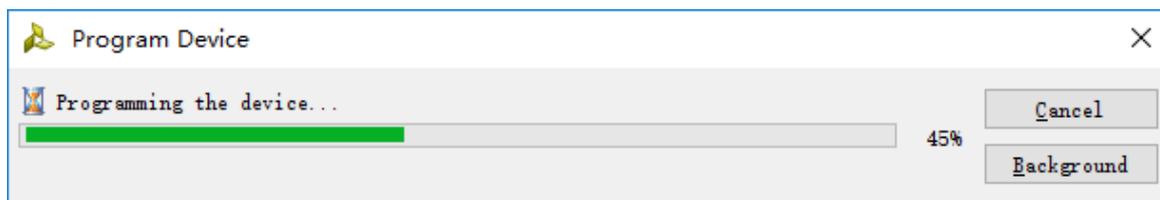


图 4-24 编程进度条

	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	39 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		

## 5. PARTB 二值化实验结果

首先我们让 HDMI 信号源显示一幅素材图片 (位于同文件夹下的 J20.png), 接着我们将连接 HDMI 输入端口的 HDMI 线在信号源端重新插拔一次, 以便让信号源设备重新检测 (Detect) 一下接收设备, 一切正常的话, 我们即可在 HDMI 显示器上看到显示画面。



图 5-1 二值化显示结果

<b>XINGDENG</b>	标题	文档编号	版本	页
	Lab2: 像素处理实验 1	XD-LAB-IMG-002	1.2	40 of 40
	作者	修改日期	公开	
	Joseph Xu	2019/2/14		