XD-LAB-IMG-009

# Lab9：图像处理算法模块实验 1：边缘检测

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 |
| 1.1 | Joseph Xu | 使用 MATLAB 版本更新为 R2018b | 2018-12-20 |
| 1.2 | Joseph Xu | 文档图片微调 | 2019-2-14 |
| | | | |
| | | | |

# 审核记录

| 姓名 | 职务 | 签字 | 日期 |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# 目录

# 图目录

# 表目录

# 1. 实验简介

该实验通过 MATLAB 搭建一个边缘检测算法模型，并通过 MATLAB 的 HDL Coder 将模型自动生成为硬件模块，并在 SWORD4.0 上和视频接口集成，实现一个快速的边缘检测算法模块设计。

- **对于初学者，整个实验预计耗时 1.5 小时。**
- **对于熟练者，整个实验预计耗时 40 分钟。**

## 1.1 概述

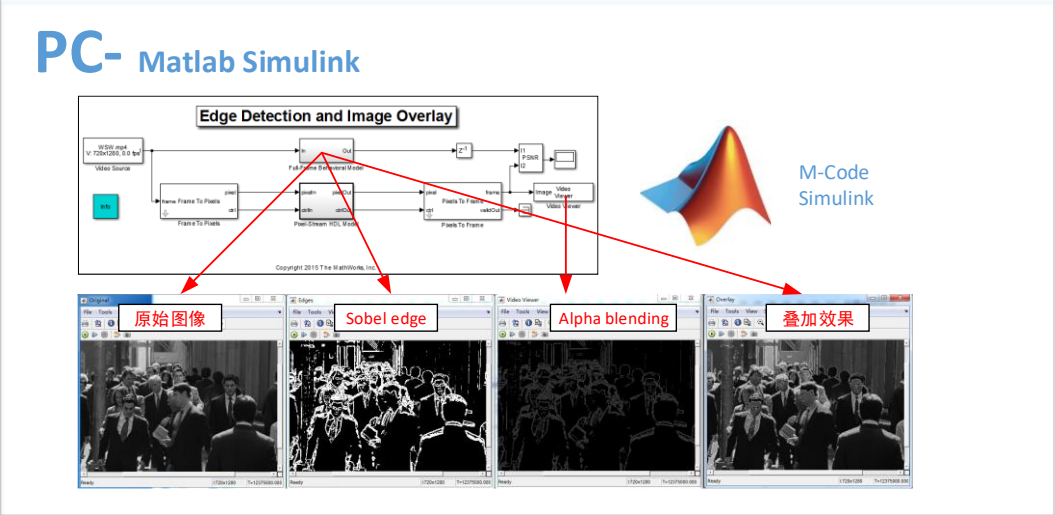



图 1-1 算法仿真流程图

## 1.2 实验目标

本实验的目标为 SWORD4.0 能够正常地在 HDMI 显示器上输出边缘检测的视频图像。

## 1.3 实验条件

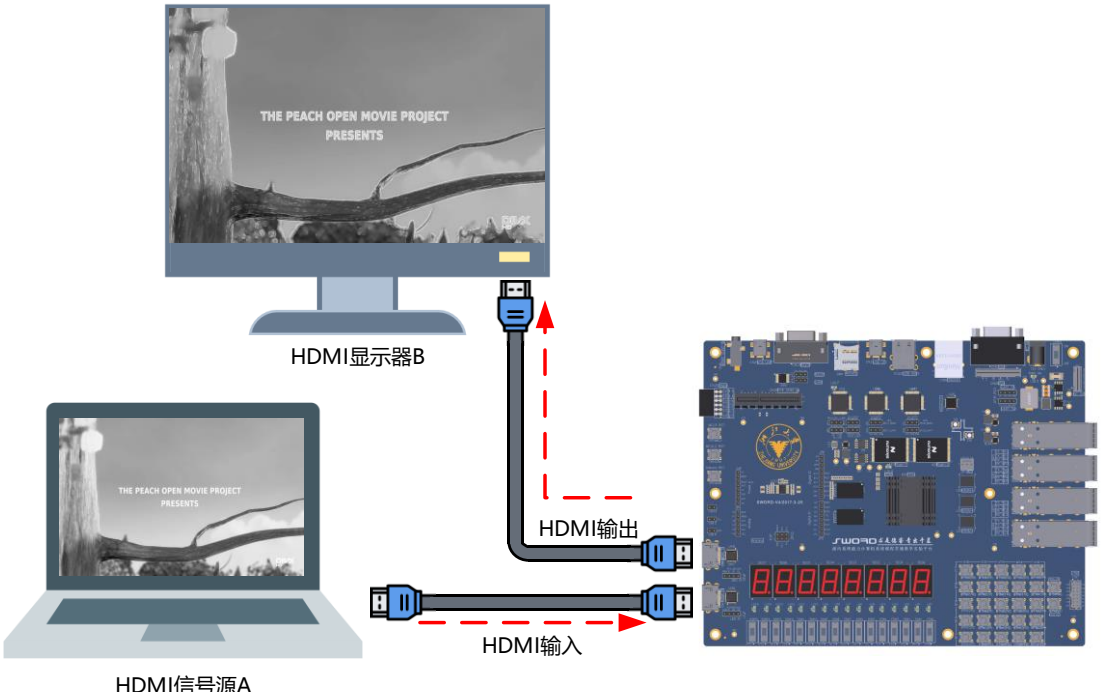
表 1-1 实验条件

| 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 |
| | MALTAB | 1 | 版本：R2018b |
| | 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：




说明：本实验中HDMI输入视频的分辨率和输出视频的分辨率相同

图 1-2 实验连接示意图

# 2. PARTA：算法仿真实验流程

本节将详细描述如何在 MATLAB 的环境下完成实验。请耐心阅读，仔细按照图示和文字说明进行操作。

## 2.1 操作步骤

1． 首先启动 MATLAB，本文选用版本为 Windows 版 R2018b。

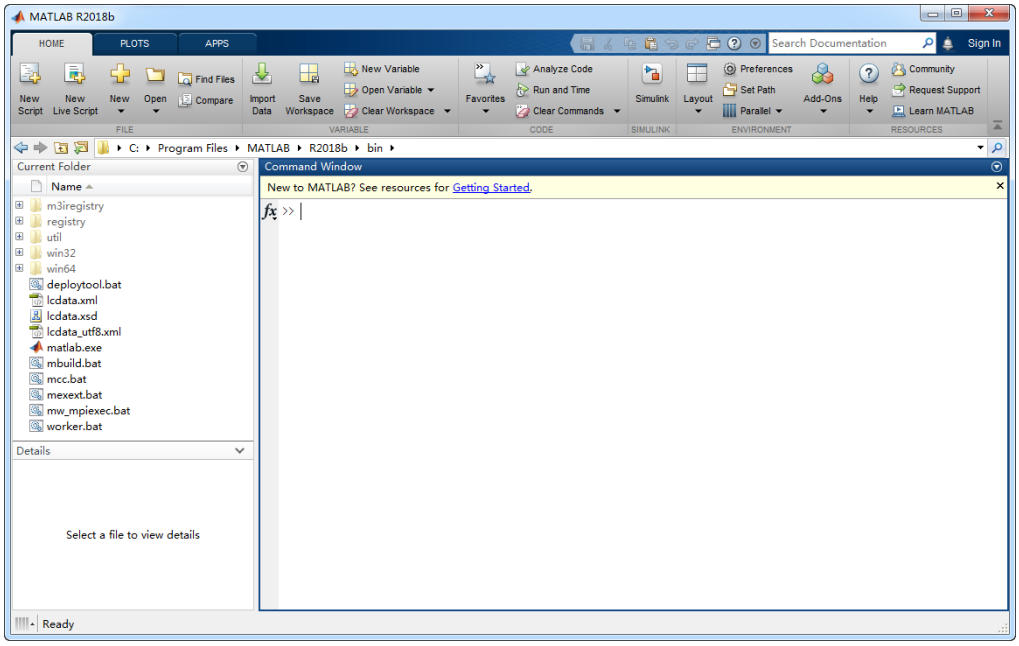



图 2-1 启动 MATLAB

接着我们要设定工作目录，即我们常用来作为存放自己的 MATLAB 文件和图片素材的目录。本文为 D:\ImageLabs\source\lab9\matlab（如果没有自己创建一个）：

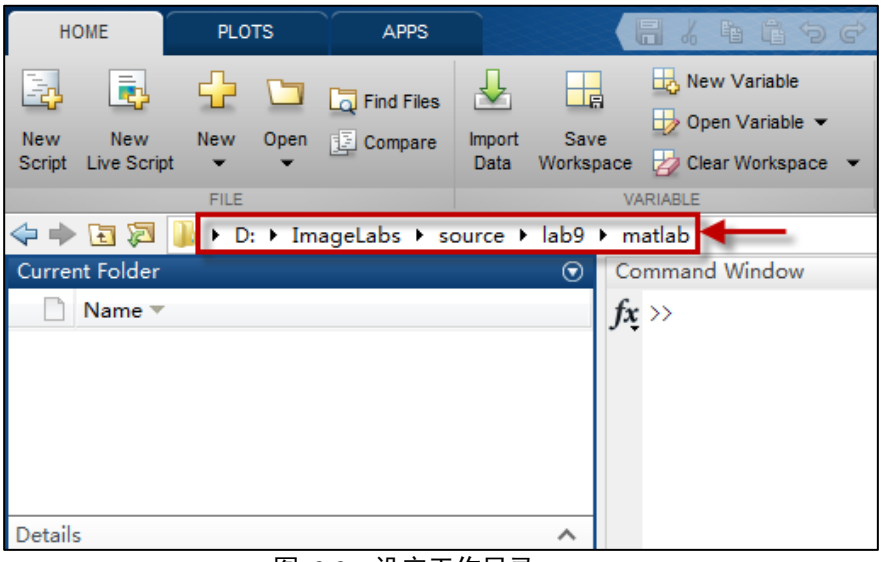



图 2-2 设定工作目录

然后我们将边缘检测的模型文件放进这个目录：


图 2-3 复制实验模型和视频素材

双击这个 slx 类型的文件，可以看到该模型文件如下图所示：

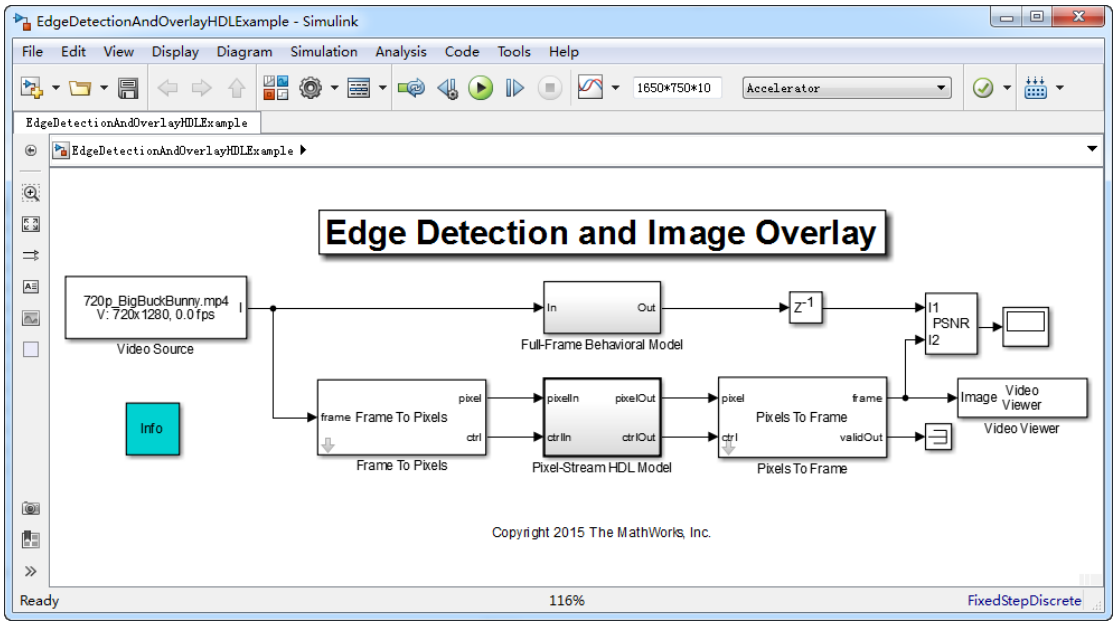



图 2-4 打开实验模型

从上图可以看出，该图像算法模型包括了一个视频输入源，其分辨率为 1280x720，帧速为 0.0fps，经过了 2 路处理，其中上面一路为基于完整帧（Full-Frame）的行为模型（即软件处理），下面一路为基于像素流（Pixel-Stream）的 HDL 模型（即硬件处理）。

下面我们先来看看 **Full-Frame Behavioral Model**，双击该模块，如下图所示：

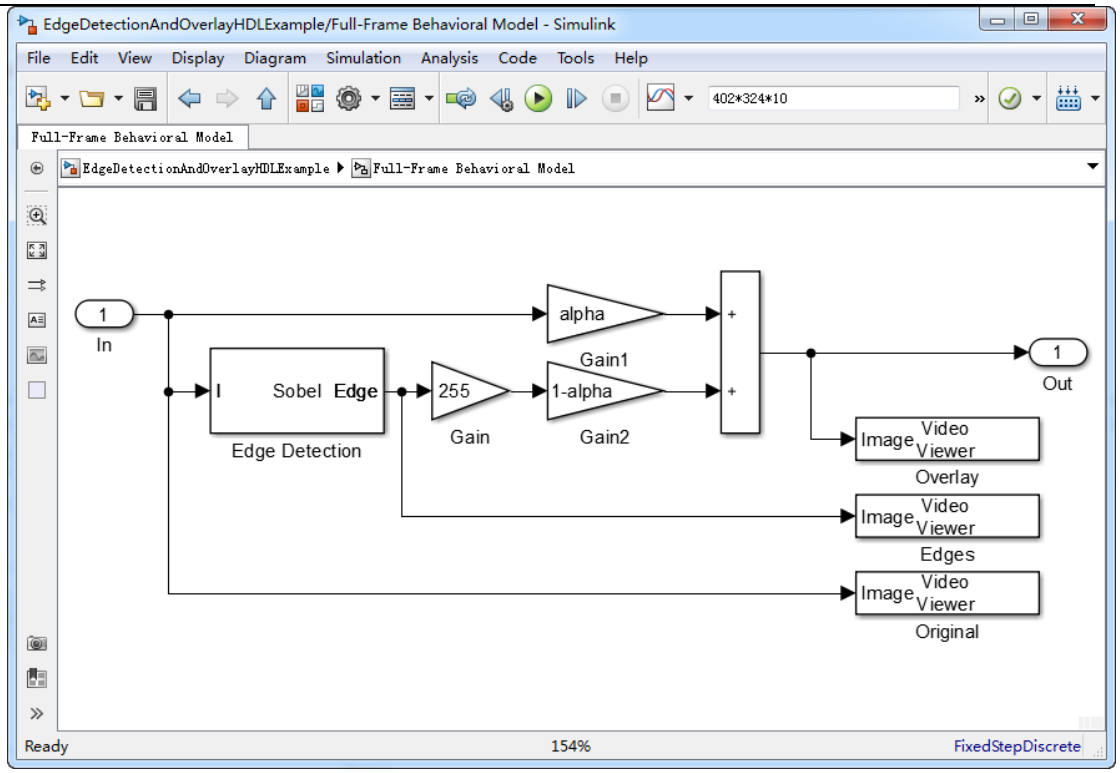



图 2-5 Full-Frame Behavioral Model 内部结构

该模型全部为 Simulink 内置的图像处理工具包（Image Processing Toolbox）所包含的模块（Block），其中有 1 个基于 Sobel 算子的边缘检测功能块，3 个增益模块（Gain），1 个输入模块，1 个输出模块，1 个加法模块和 3 个视频显示模块。该模型非常直观，其含义为接收输入视频，然后经过 Sobel 边缘检测和 Alpha（Alpha 即透明度）混合后输出，整个处理过程中，我们能够看到原始图像，边缘轮廓图像以及边缘轮廓叠加到原始图像后的层叠图像。

双击 Edge Detect 模块，如下图所示：

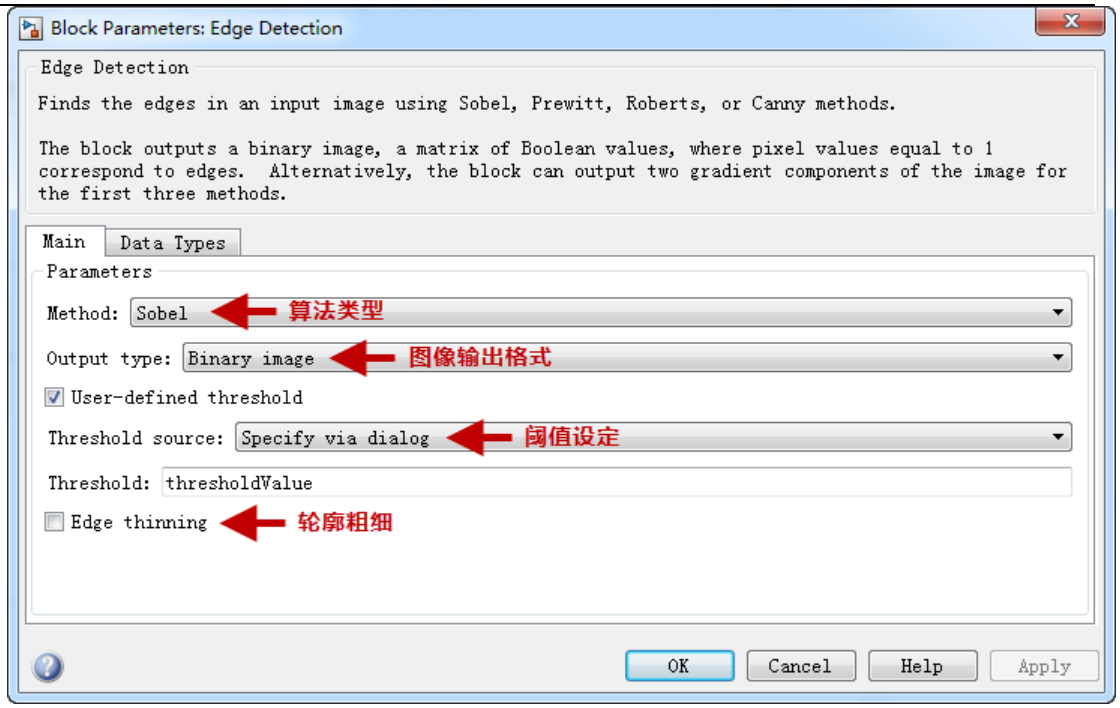



图 2-6 Edge Detect 模块参数设置

**提示：请自行修改上述参数设定，观察不同参数组合下的处理结果。**

设定好上述参数后，点击 OK 保存，并关闭对话框返回 **Full-Frame Behavioral Model。**

了解了基于完整帧的行为模型后，我们返回到最上层的算法模型视图，点击如下图所示的图标：

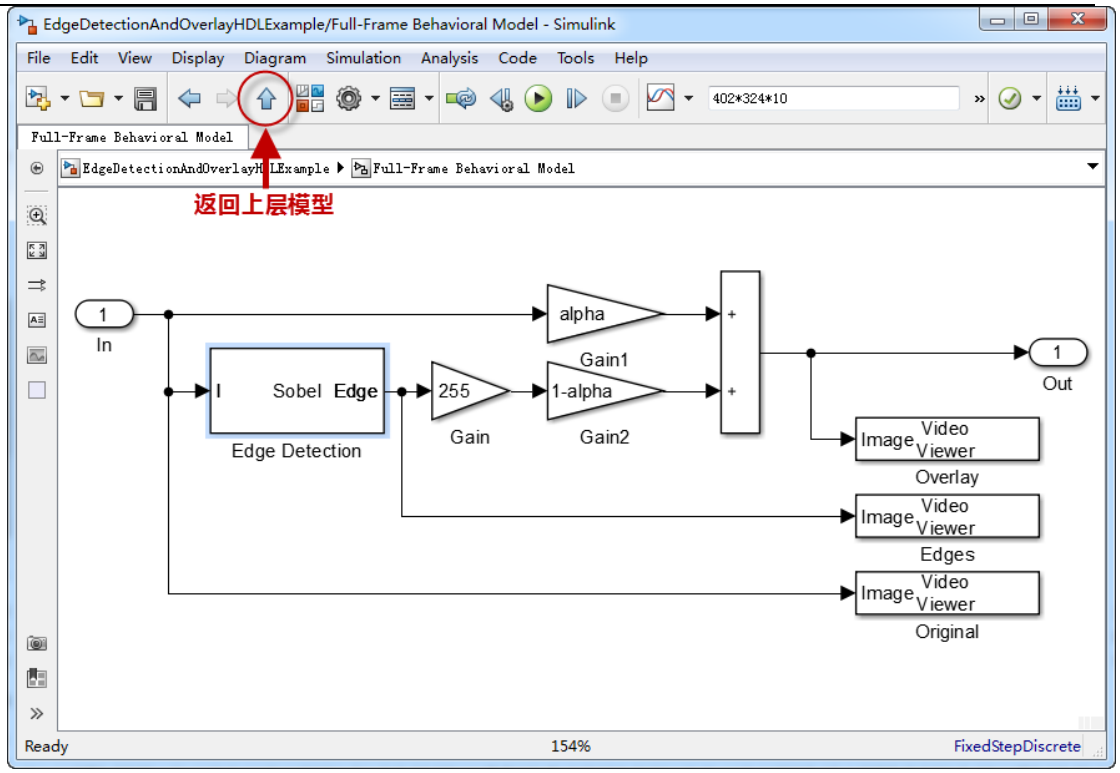



图 2-7 返回上层模型

在最上层点击运行按钮，开始算法模拟运行，如下图所示：

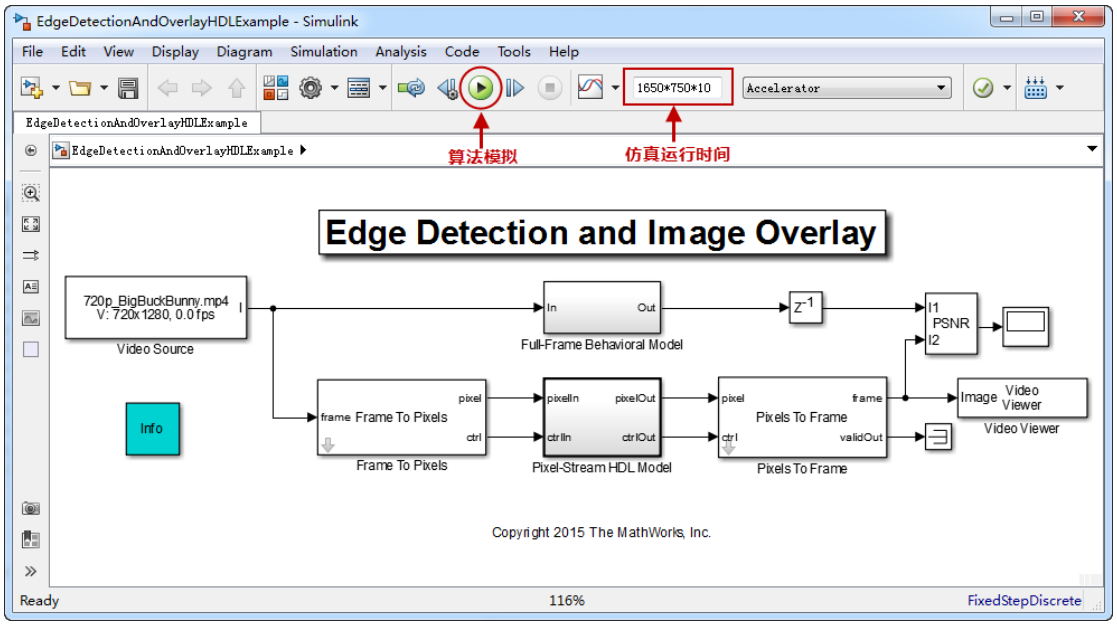



图 2-8 进行算法仿真

在该模型中，已经内置了算法模拟的运行时间，即 1650*750*10，即运行 10 帧图像的处理计算。运行后，我们能看到如下结果：

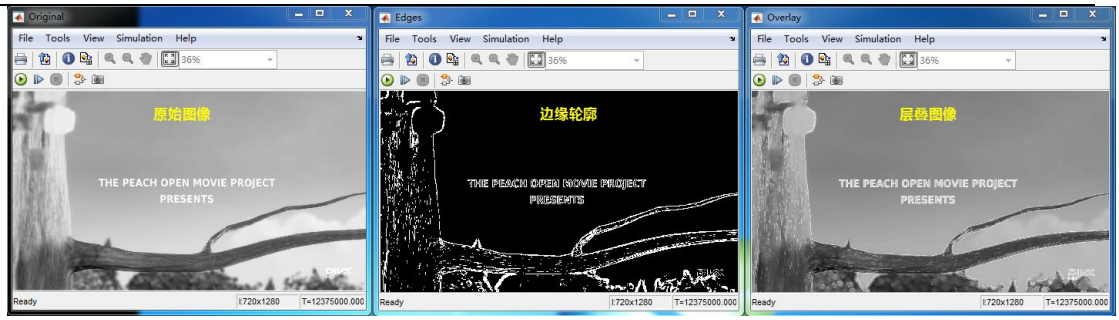



图 2-9 算法仿真结果

接着我们来看看基于像素流的模型，由于该模型后面要转为硬件处理，所以图像数据的处理方式从完整帧转为了像素流，亦即将一幅完整的图像按照一定数量的行像素进行采样-处理。双击 **Frame to Pixels** 模块，如下图所示：

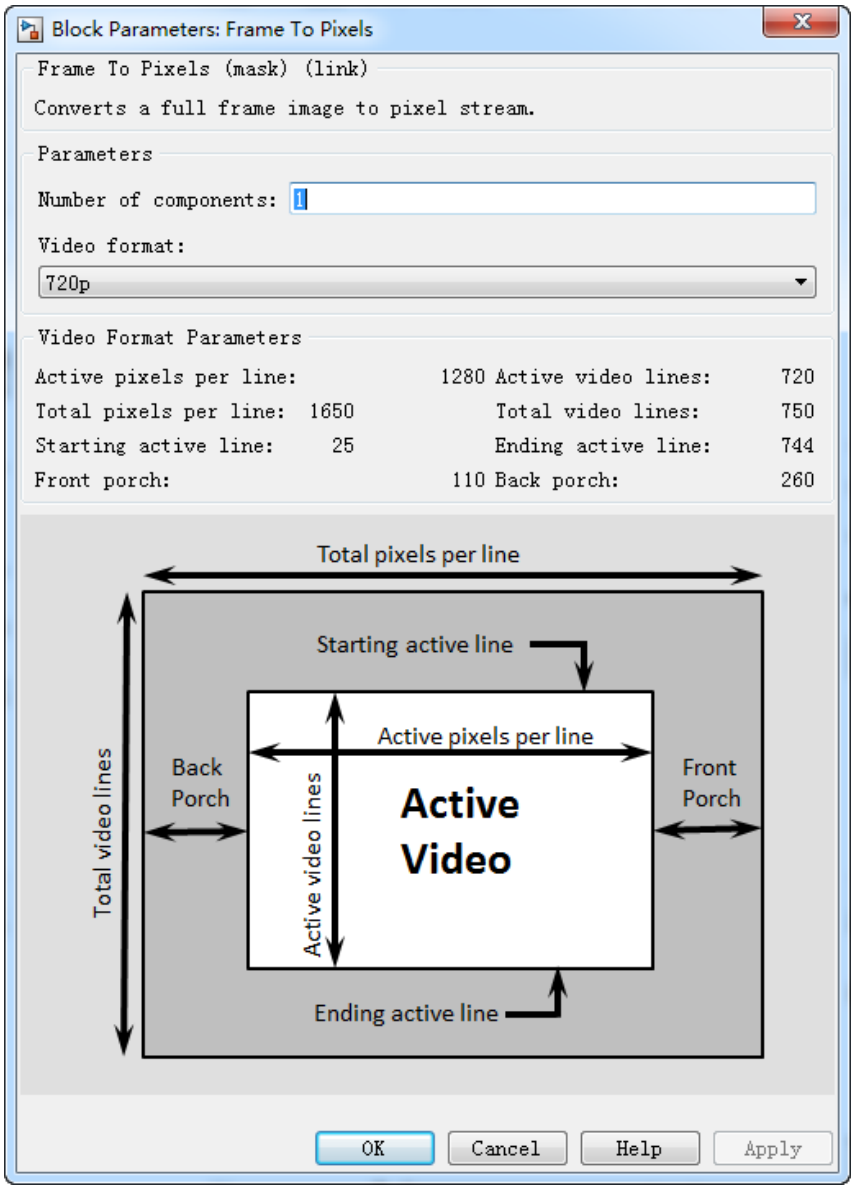



图 2-10 Frame to Pixels 模块参数

从该模块能很直观的看到一幅视频图像画面的各种说明，包括了：视频分辨率

格式，一行包含的像素值与 1 帧包含的行数，以及前沿（Front Porch），后延（Back Porch）等时序信息。

**注意：这里的设置需要和输入的视频或图像分辨率对应。**

接着双击 **Pixel-Stream HDL Model**，可以看到该模型包含了边缘检测，视频同步和图像叠加三个子模块，如下图所示：

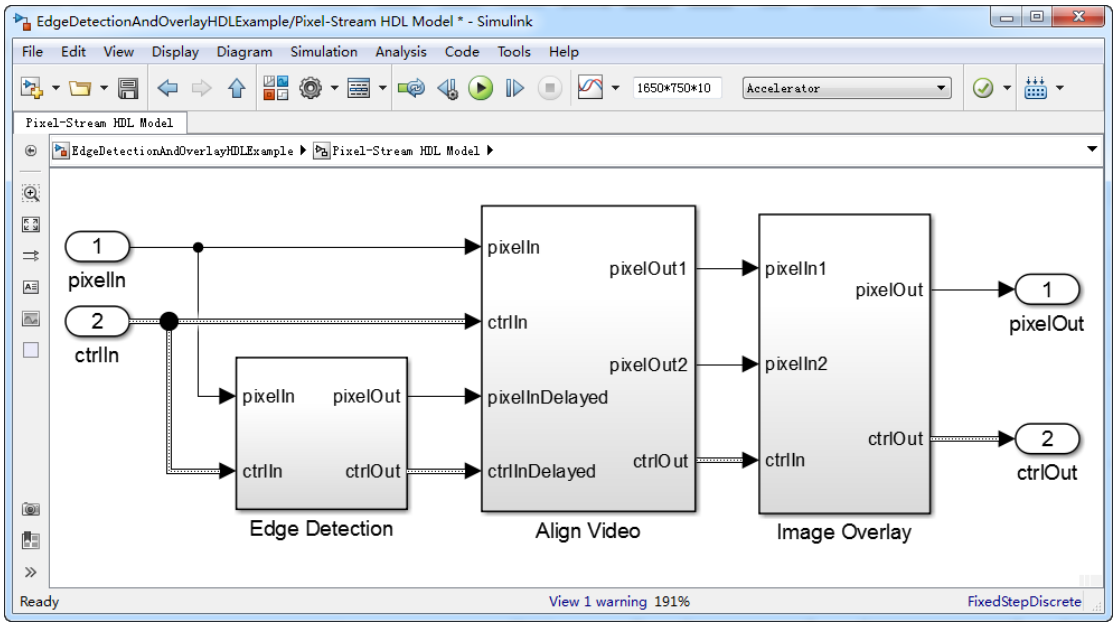



图 2-11 Pixel-Stream HDL Model 内部结构

双击 **Edge Detection** 模块查看更多细节，如下图所示：

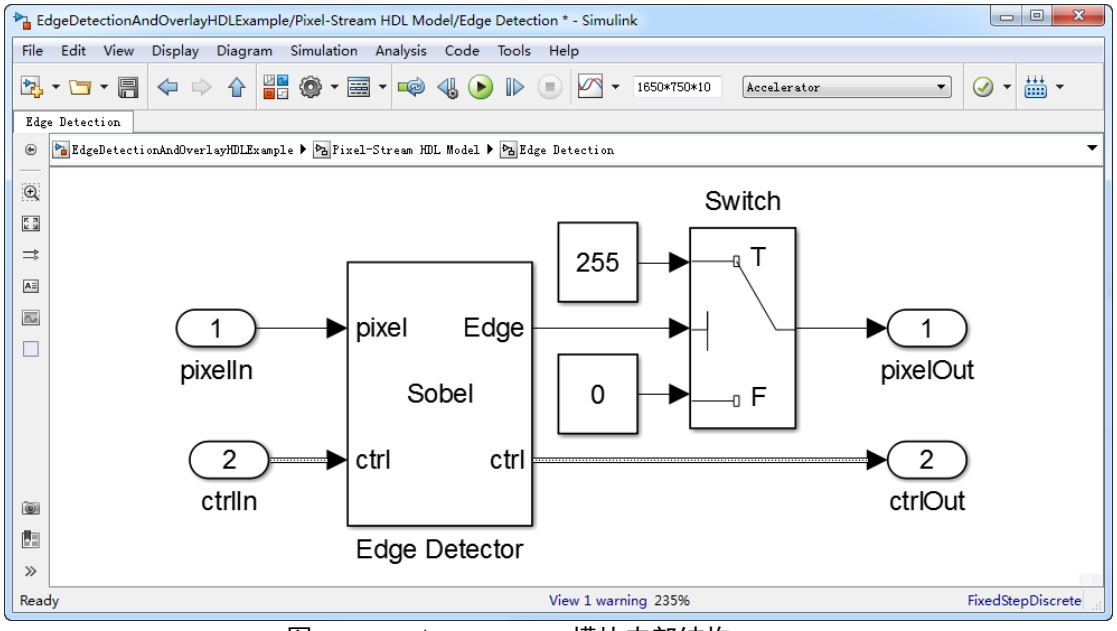



图 2-12 Edge Detection 模块内部结构

在这一层次上，可以看到边缘检测算子模型的最小模块，这里的各个模块都是由 Simulink 里的 Vision HDL Toolbox 所内置的功能块。

然后我们返回最上层的模型视图，开始验证该算法的硬件处理部分，在 MATLAB 的命令行窗口输入如下命令：

```
thresholdValue=8
alpha=0.5
```

上述参数设定了边缘检测模块的检测阈值以及图像叠加时的透明度。在本例中，检测阈值的范围为 0~256，超过该范围，系统会提示溢出（overflow）。同时，**阈值设定的越大，整体图像中检测到的边缘轮廓就越少。**

接着我们来生成 HDL 代码，鼠标右键单击 Pixel-Stream HDL Model，然后选择 HDL Code →Generate HDL for Subsystem，如下图所示：

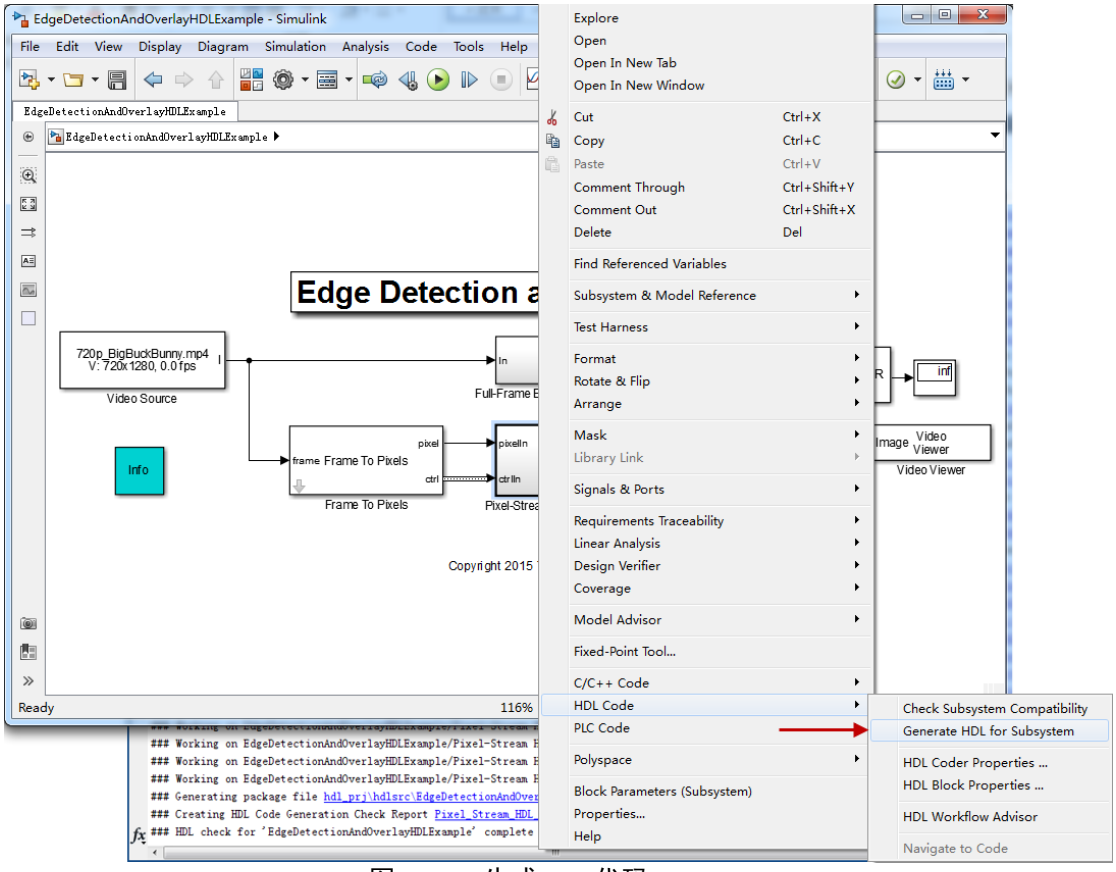



图 2-13 生成 HDL 代码

之后我们就能看到 MATLAB 开始生成 **Pixel-Stream HDL Model** 模型的 HDL 代码，如下图所示：

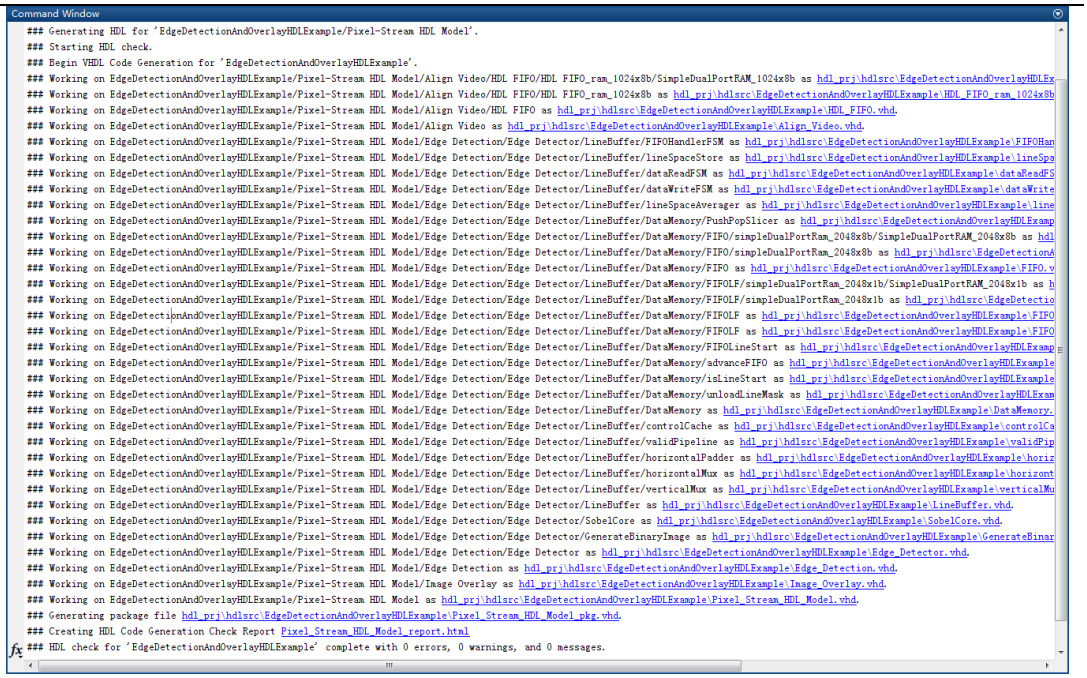



图 2-14 生成 HDL 代码后的信息输出

该 HDL 代码存放于当前工作目录下的 hdl_proj 目录，如下图所示：


图 2-15 生成的 HDL 代码目录位置

# 3. PARTB：算法模块硬件部署流程

## 3.1 操作步骤

前面我们通过 MATLAB 生成了 1 个边缘检测的算法模型的 HDL 代码，现在我们将该代码部署到 SWORD4.0 上。

1．首先启动 Vivado 2014.4，然后在主界面点击“Create New Project”，创建工程，如下图所示：


图 3-1 Vivado 下创建新工程

2． 在弹出的向导窗口点击 Next 继续，如下图所示：

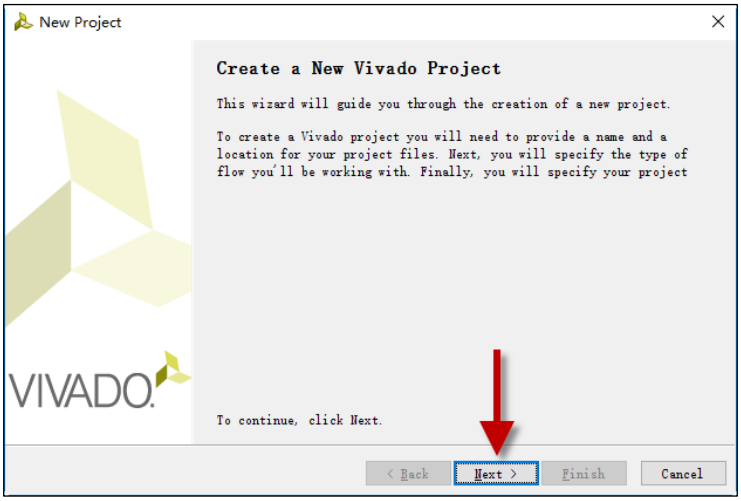



图 3-2 创建新工程向导窗口

3． 接着在窗口页面输入工程名，工程路径和相关选项，按如下信息填写（注意：为保证整个实验的流畅性，请严格按照以下信息填写）：

Project name: lab9

Project location: D:/ImageLabs

Create project subdirectory：勾选

**提示：如果本地没有 ImageLabs 这个目录，请自行创建一个**

填写完成后应如下图所示，点击 Next 继续；

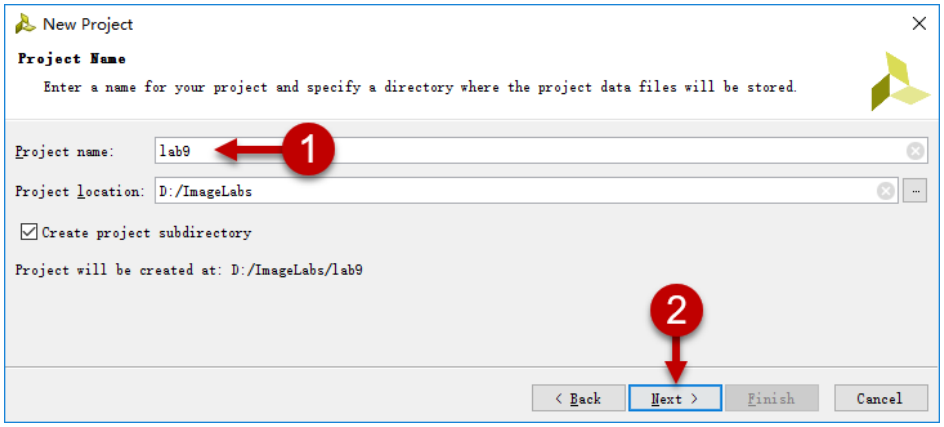



图 3-3 工程命名和保存路径

接着选择工程类型，选择 RTL Project，并勾选 Do not specify sources at this time，点击 Next 继续，如下图所示：

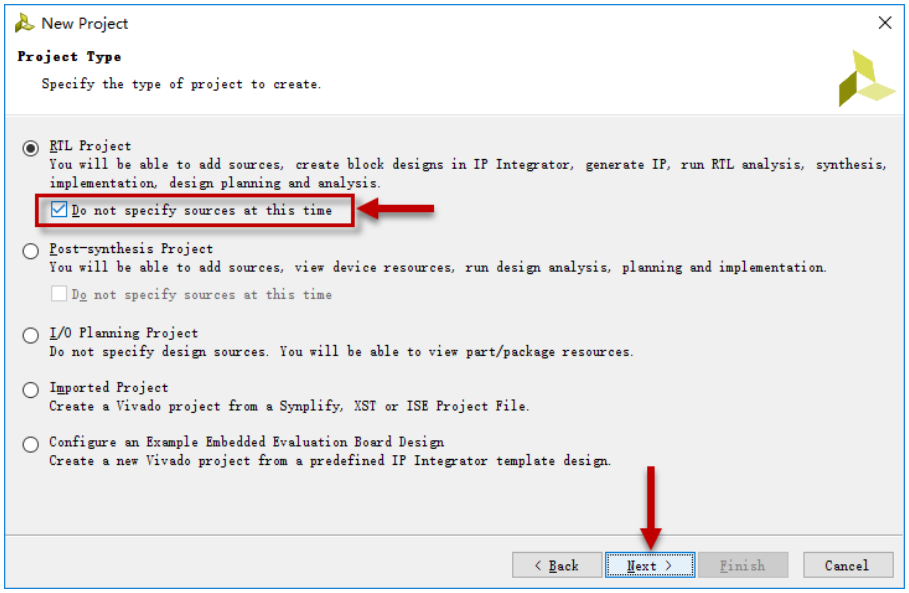



图 3-4 选择工程类型

2． 在 Default Part 页面按照如下信息选择目标器件：

Product category: General Purpose

Family: Kintex-7

Sub-Family: Kintex-7

Package: ffg676

Speed grade: -2

此时在器件列表中剩下的 3 个型号中选择 xc7k325tffg676-2 这个型号，然后点击 Next 继续，如下图所示：

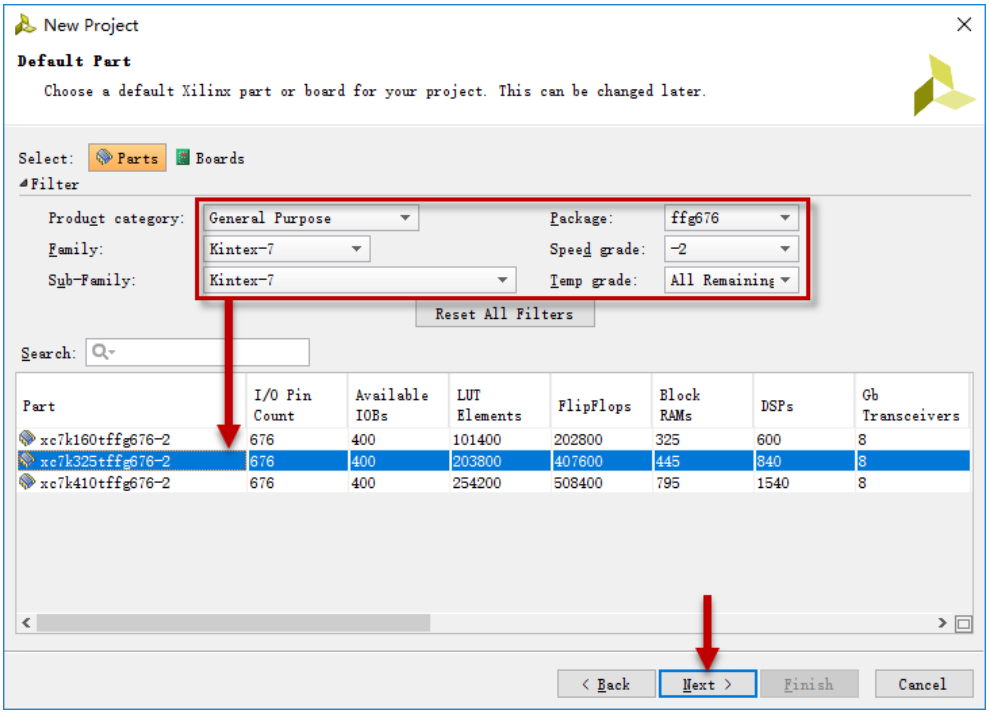



图 3-5 选择器件型号

在 New Project Summary 页面直接点击 Finish 完成新工程的创建，如下图所示：


图 3-6 点击 Finish 完成工程创建

接着要为新的工程添加一个 IP 库（repo），为此我们在 Vivado 主界面的左侧边栏点击 Project Settings，然后在弹出的设置窗口中选择 IP 项，接着点击 Add Repository，整个过程如下图所示：

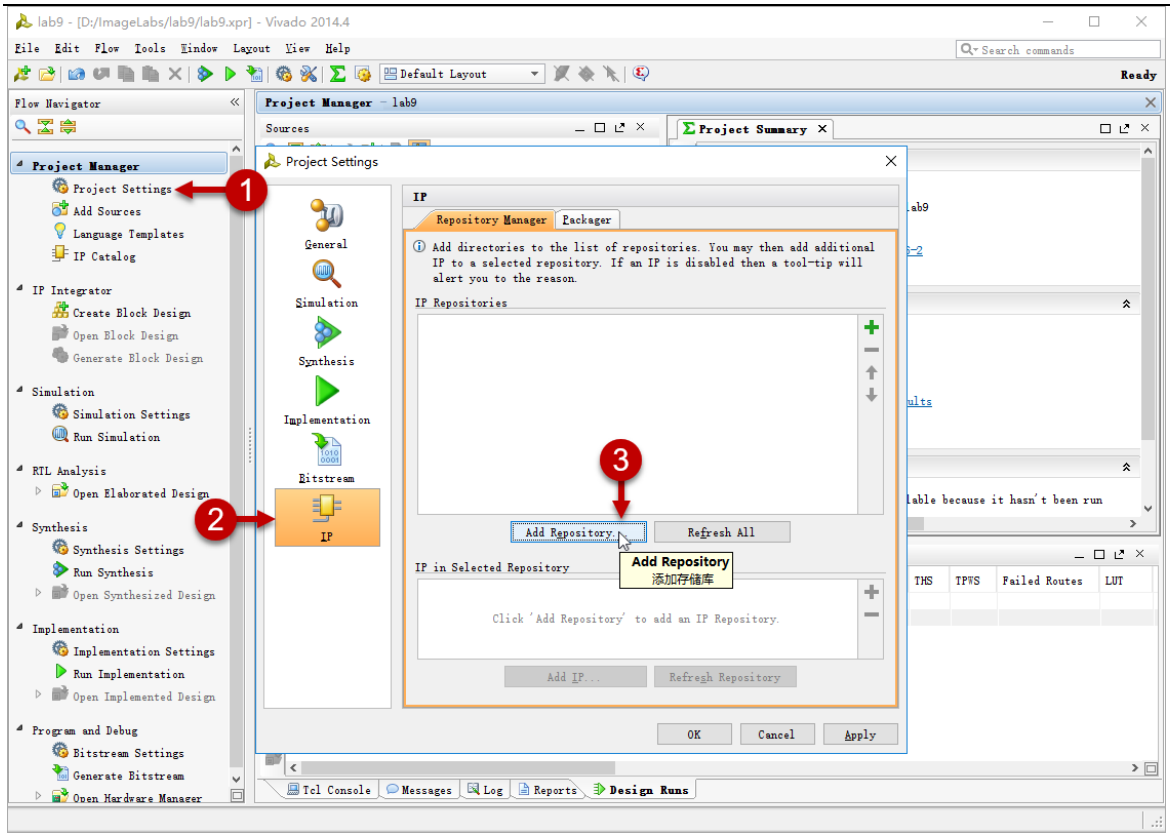



图 3-7 设置：添加 IP 库

在对话框中找到 D:\ImageLabs 目录，首先选择 FPGA-Image-Library-Pubulish，然后按住 Ctrl 键，同时选择 repo，点击 Select，整个过程如下图所示：

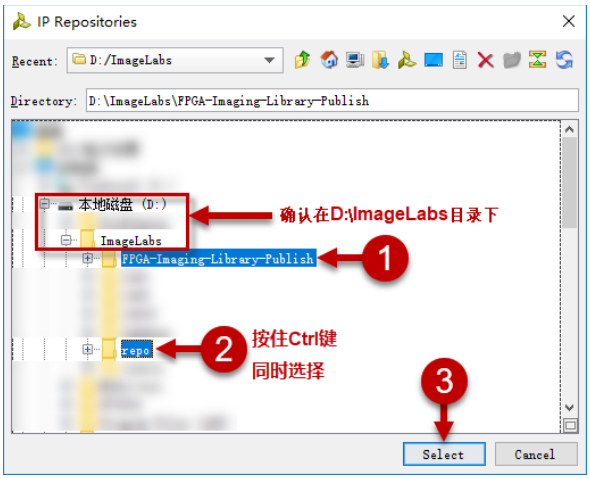



图 3-8 选择 IP 库的位置

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：

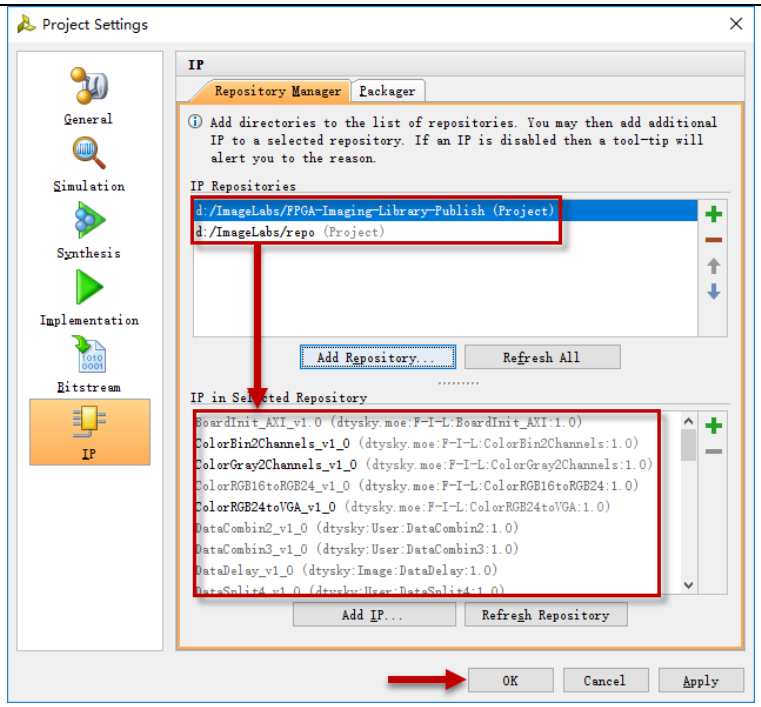



图 3-9 IP 库中的 IP 列表的显示

接着在 Vivado 主界面点击 Add Sources 图标，在弹出的窗口中选择 Add or create design sources，点击 Next 继续，过程如下图所示：

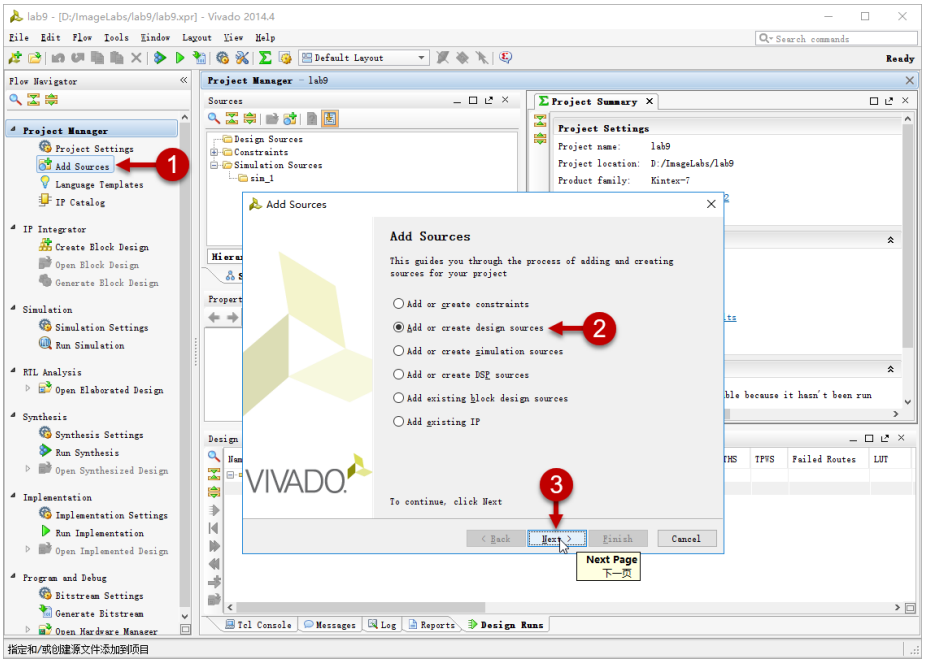



图 3-10 添加设计文件

在对话框中点击 Add Files 按钮，如下图所示：

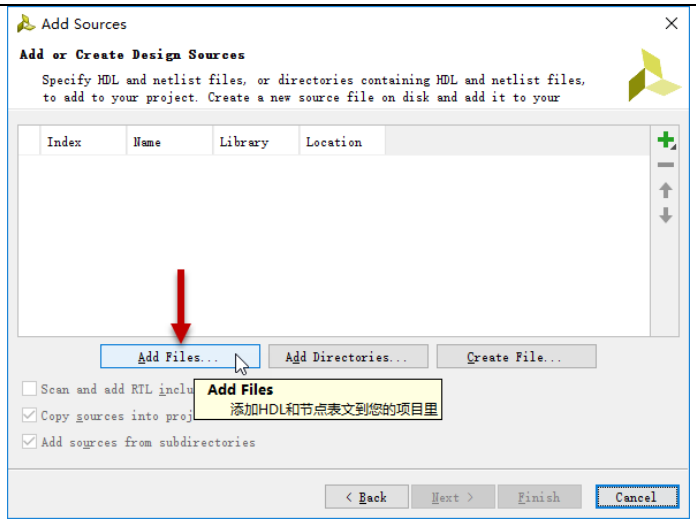



图 3-11 添加文件

在文件选择窗口，找到 D:\ImagesLabs\source\lab9 文件夹，将如图示的 3 个文件选中，直接点回车完成添加：

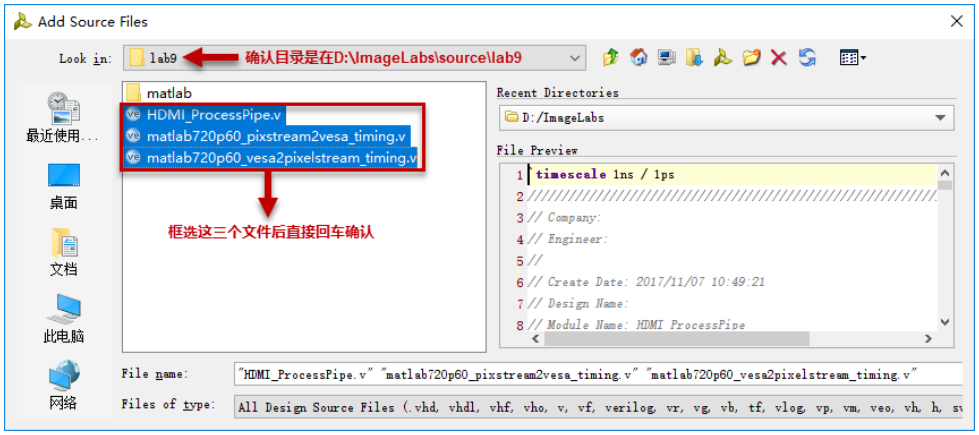



图 3-12 选择已有的设计文件

然后在文件添加窗口可以看到 3 个文件被添加，然后勾选 Copy sources into project，点击 Finish 完成文件添加，过程如下图所示：

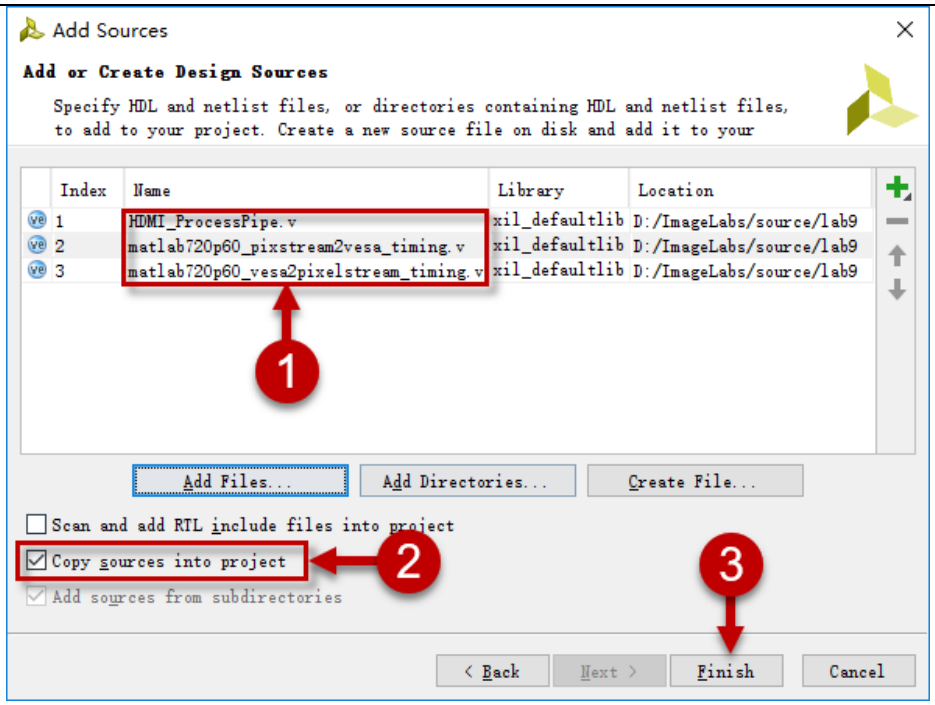



图 3-13 点击 Finish 确认添加文件

回到 Vivado 主界面，可以看到刚刚添加的是一个顶层设计文件，其中包含了一些 IP 和设计模块，有一些模块前面显示的是带问号的图标，表明该模块或 IP 未添加到工程中。下面我们就来补全，点击 Vivado 主界面的 IP Catalog，然后在弹出的搜索栏中，输入 dvi2rgb，在搜索结果会显示 DVI to RGB Video Decoder，过程如下图所示：

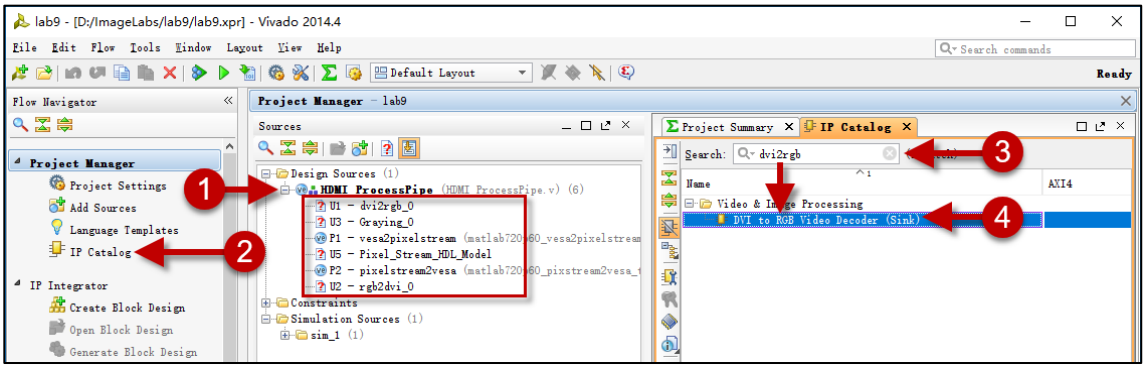



图 3-14 从 IP 库中添加 IP

双击这个 IP，Vivado 会弹出该 IP 的配置对话框，按照如下图所示进行配置，并点击 OK 完成：

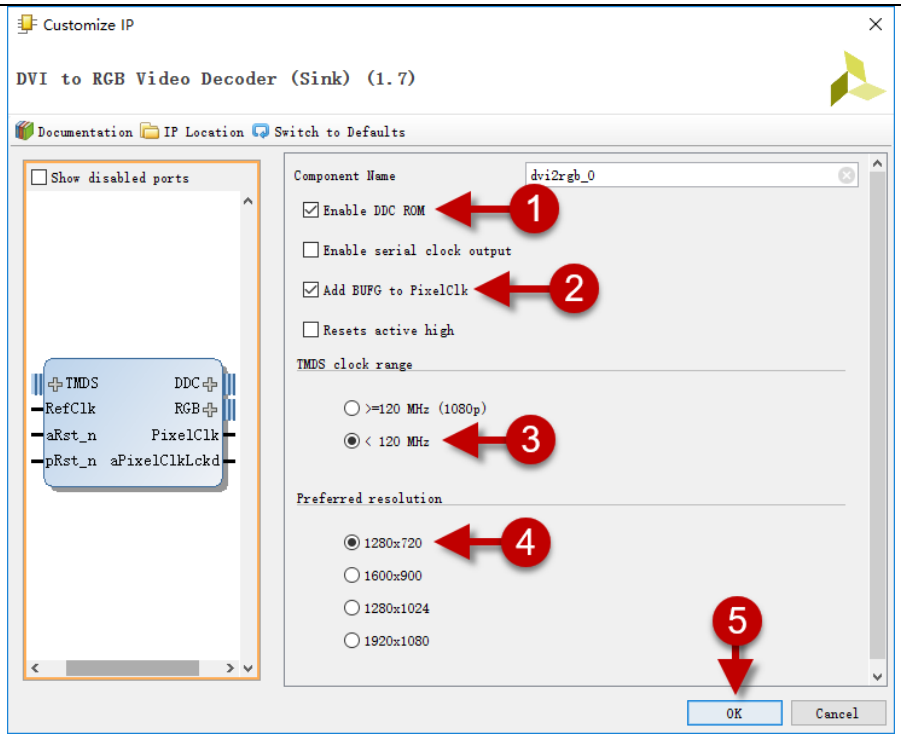



图 3-15 配置 DVI2RGB IP

接着会弹出一个 IP 生成文件的窗口，点击 Generate 继续，如下图所示：

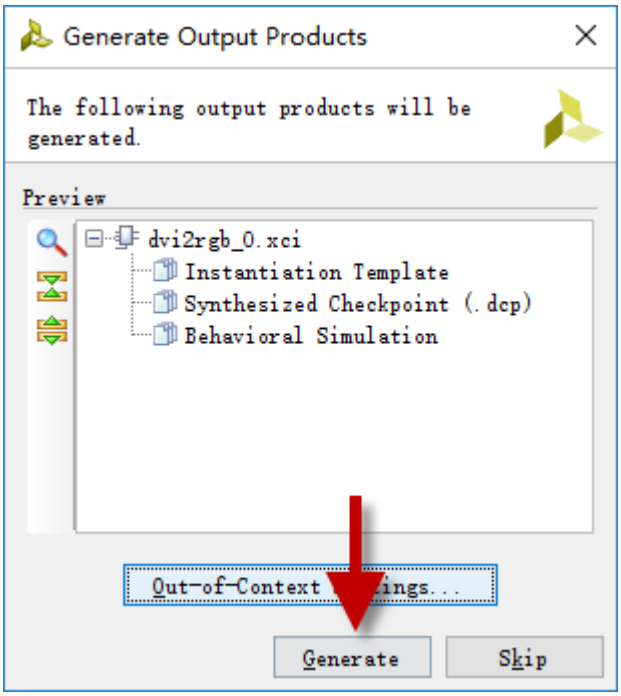



图 3-16 点击 Generate 生成 DVI2RGB IP

在随之弹出的提示窗口，点击 OK，如下图所示：

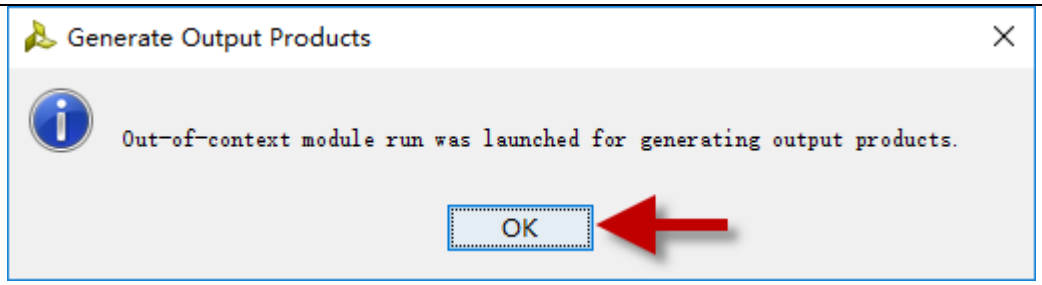



图 3-17 点击 OK 确认 IP 相关文件已生成

接着按同样的方法，在 IP Catalog 的搜索栏输入 rgb2dvi，并双击搜索结果 RGB to DVI Video Encoder，进行配置，过程如下图所示：

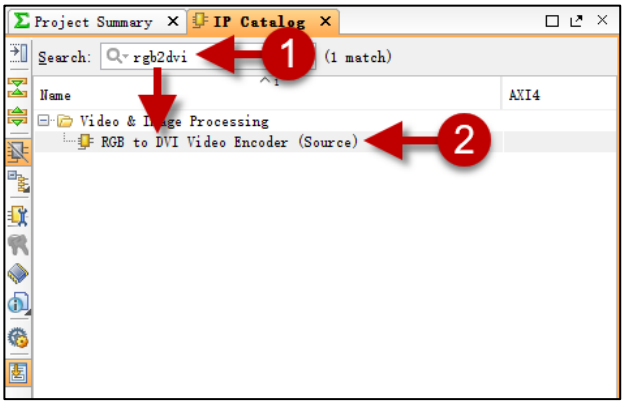



图 3-18 从 IP 库中搜索 RGB2DVI IP

在配置窗口中，按如下图示进行配置，点击 OK 完成：

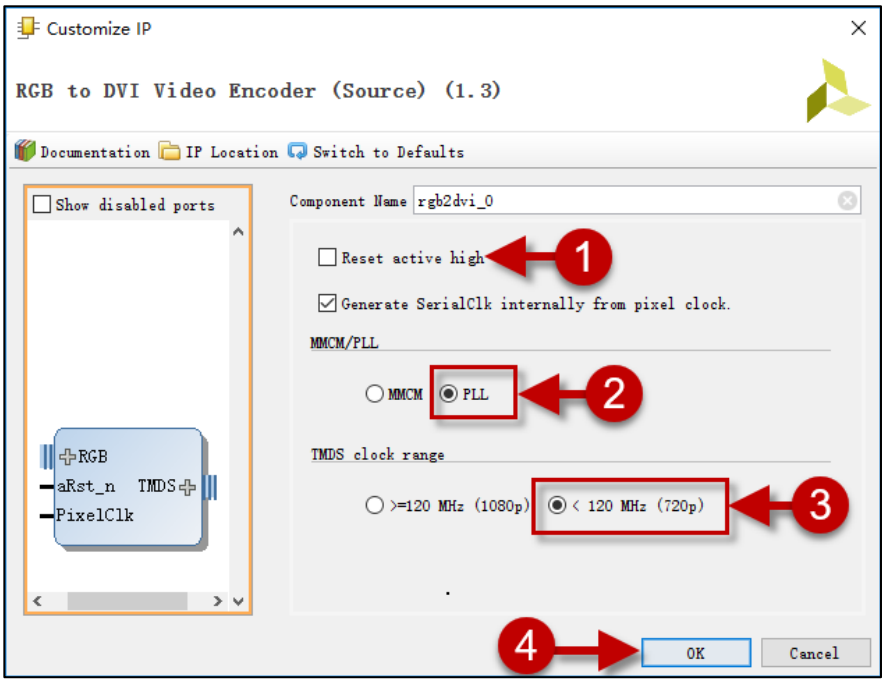



图 3-19 配置 RGB2DVI IP

接着会弹出一个 IP 生成文件的窗口，点击 Generate 继续，如下图所示：

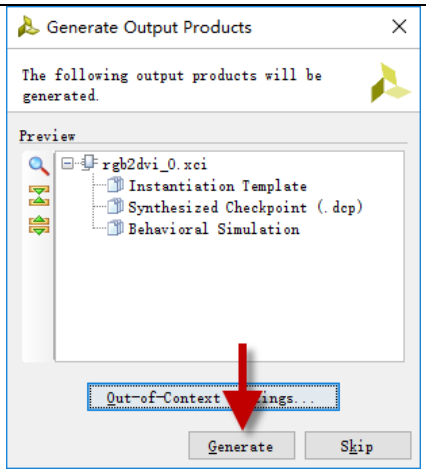



图 3-20 点击 Generate 生成 RGB2DVI IP

在随之弹出的提示窗口，点击 OK，如下图所示：

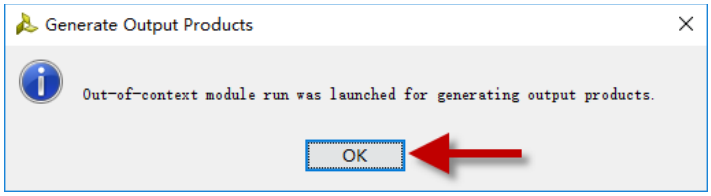



图 3-21 点击 OK 确认 IP 相关文件已生成

在 IP Catalog 的搜索栏输入 Graying，并双击搜索结果，过程如下图所示：

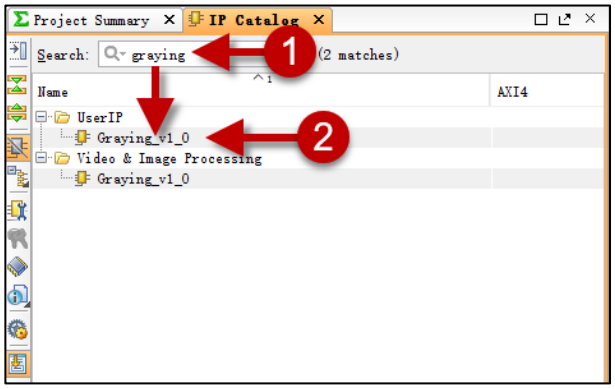



图 3-22 从 IP 库中搜索灰度化 IP

在配置对话框中，按如下图示进行配置，点击 OK 完成：

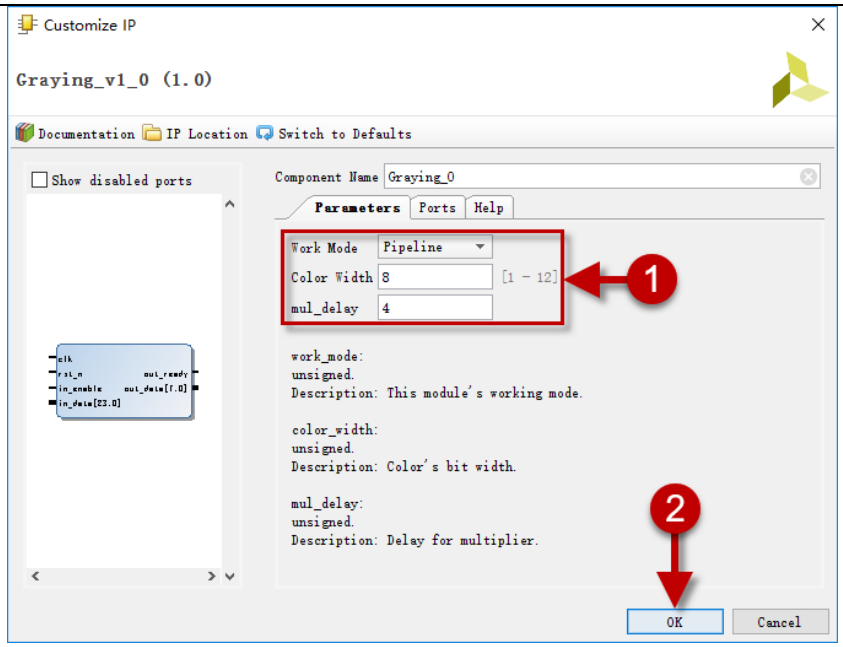



图 3-23 配置 Graying IP

接着会弹出一个 IP 生成文件的窗口，点击 Generate 继续，如下图所示：

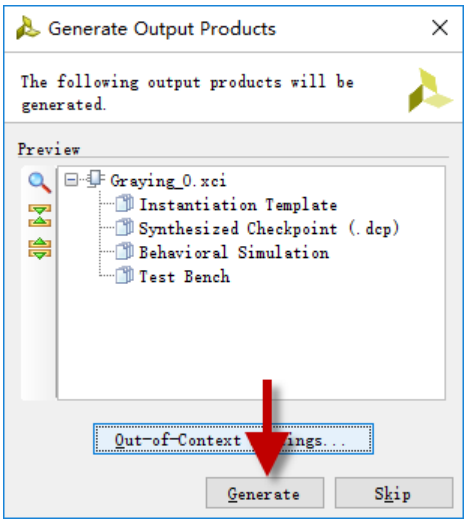



图 3-24 点击 Generate 生成 Graying IP

至此，我们已经完成了整个设计的大部分文件导入或添加，但还有一个模块是空着的：Pixel_Stream_HDL_Model，如下图所示：

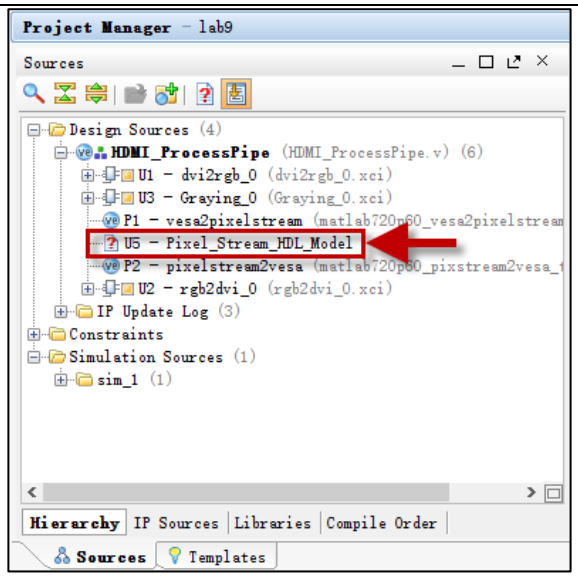



图 3-25 最后添加 MATLAB 生成的文件

下面我们就将之前在 MATLAB 中仿真的算法模型生成的 HDL 模块代码添加进来，在 Vivado 主界面点击 Add Sources 图标，在弹出的窗口中选择 Add or create design sources，点击 Next 继续，过程如下图所示：

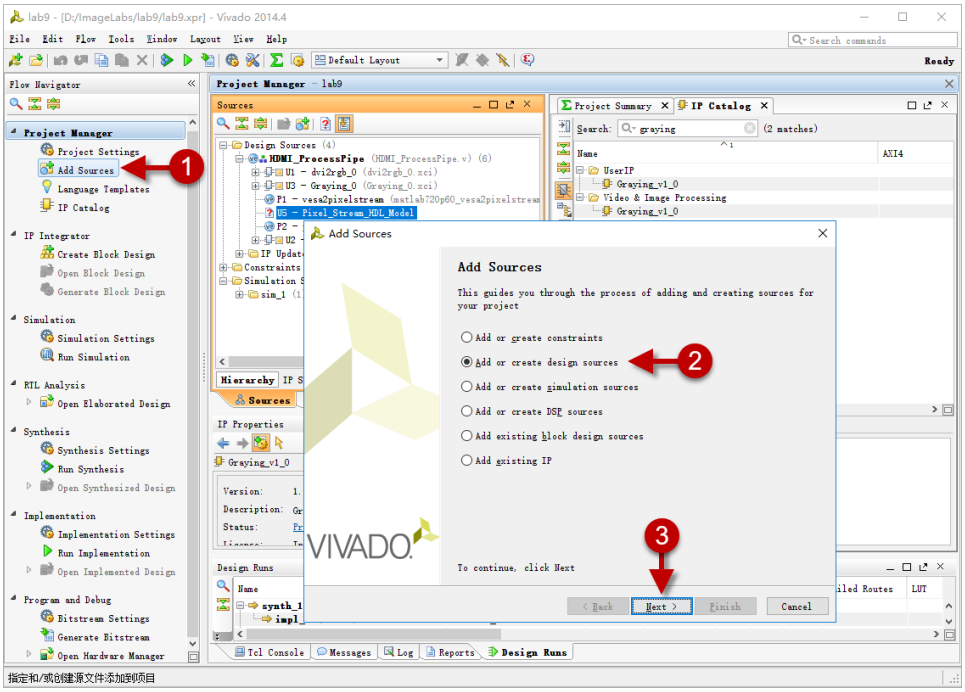



图 3-26 添加设计文件

在对话框中点击 Add Files 按钮，如下图所示：

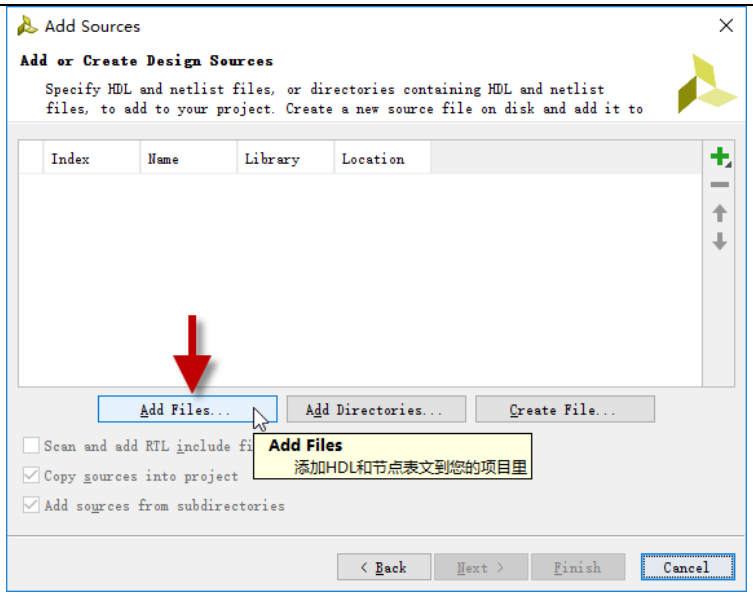



图 3-27 添加文件

在文件选择窗口，找到 MATLAB 生成的 HDL 代码所在的文件夹，即：D:\ImagesLabs\source\matlab\lab9\hdl_proj\hdlsrc\EdgeDetectionAndOverlayHDLExample

将如图示的文件选中，直接点回车完成添加：

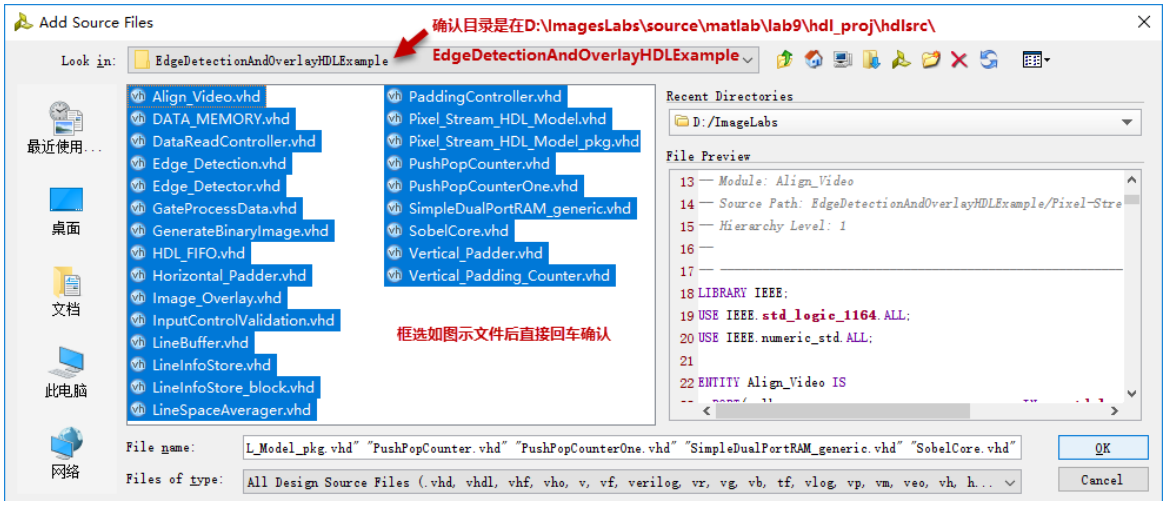



图 3-28 添加生成的全部 HDL 文件

在文件添加窗口，能看到添加的文件，这里要核对一下一共有 24 个文件添加进来了，然后勾选 Copy sources into project，之后点击 Finish 完成文件的添加，过程如下图所示：

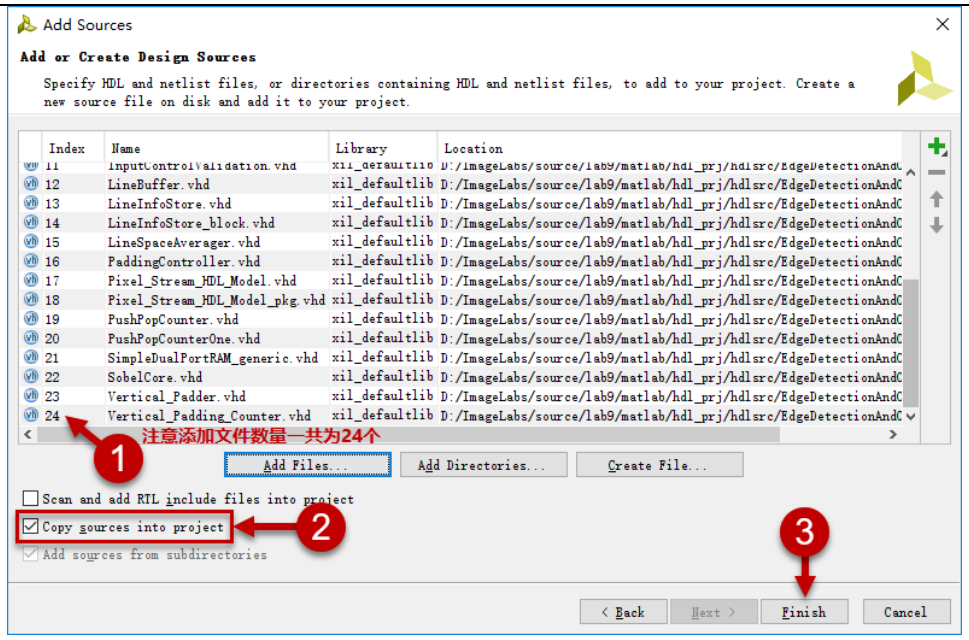



图 3-29 确认添加文件的数量

文件添加后，在 Vivado 主界面的 Source 窗口能看到 Pixel_Stream_HDL_Model 也被添加进来了，如下图所示：

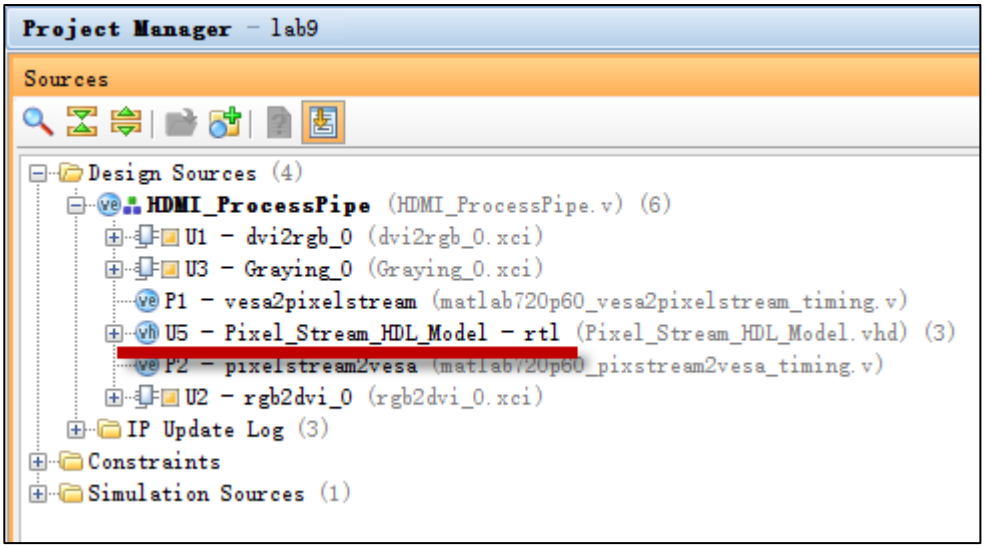



图 3-30 添加全部文件后的代码视图

接着我们要添加约束文件，在 Vivado 主界面点击 Add Sources 图标，在弹出的窗口中选择 Add or create constrains，点击 Next 继续，过程如下图所示：

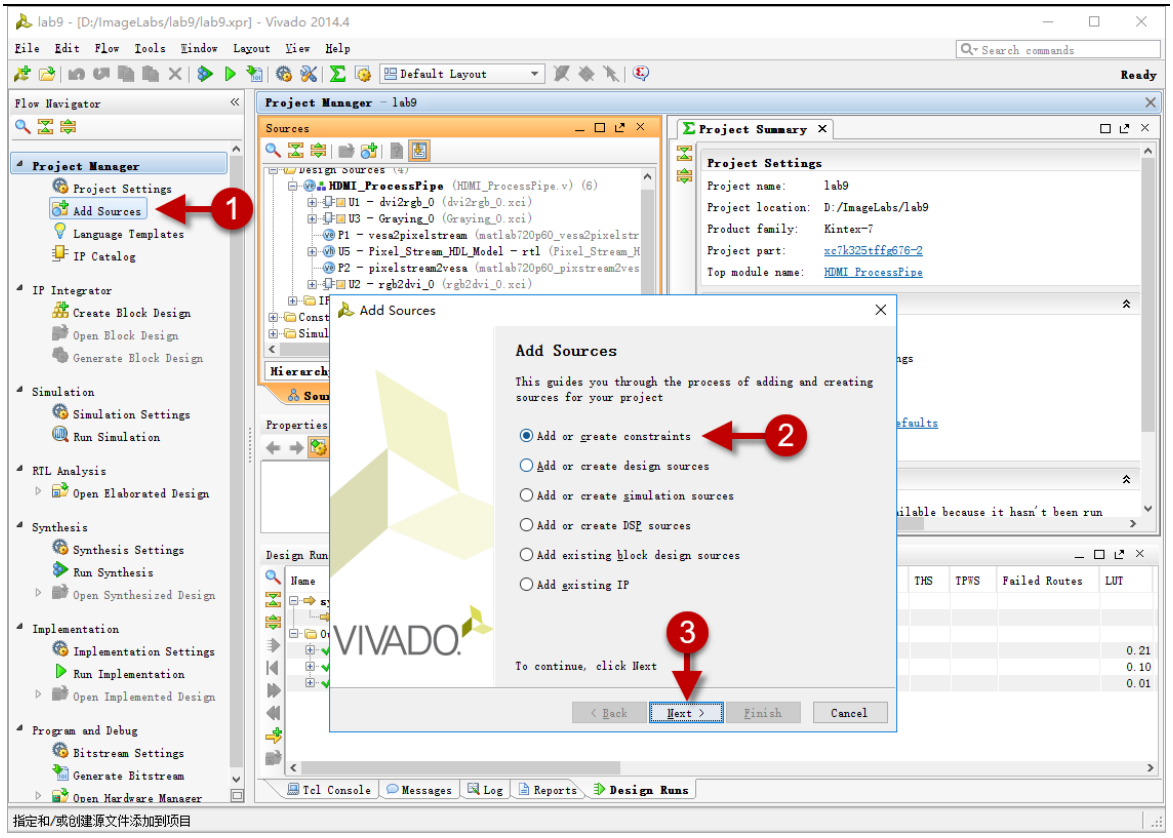



图 3-31 添加约束文件

在文件选择窗口，找到约束所在的文件夹，即：

D:\ImagesLabs\source\lab9\

将如图示的文件选中，直接点回车完成添加：

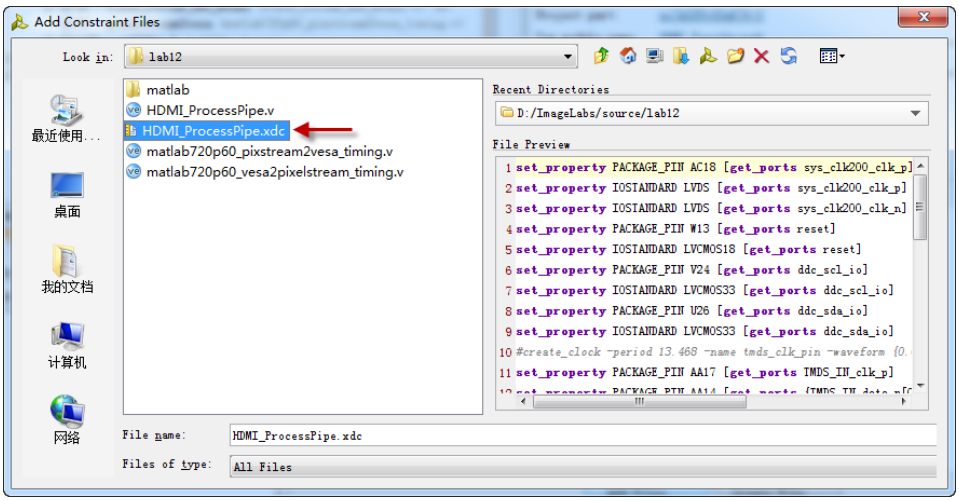



图 3-32 选择正确的 XDC 文件

在文件添加窗口，检查添加的文件名和文件路径，无误后，勾选 Copy constrains files into project，然后点击 Finish 完成添加，过程如下图所示：

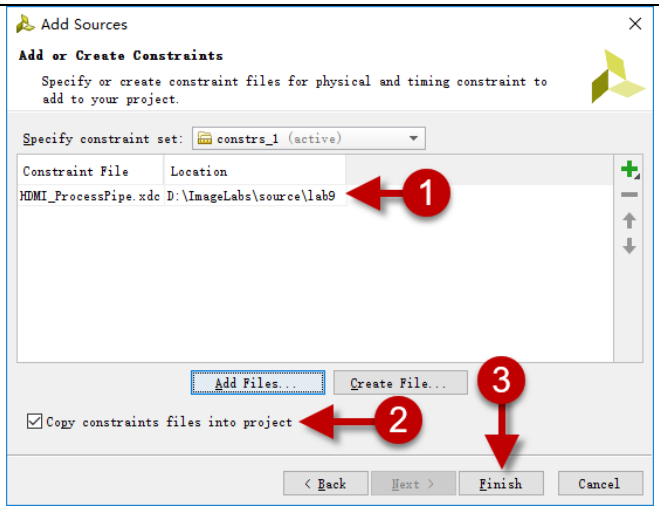



图 3-33 确认文件名和路径正确后点击 Finish 确认

至此，边缘检测算法模块的 HDL 代码已经部署完毕，在 Vivado 主界面点击 Generate Bitstream，并在随后弹出的提示对话框中点击 Yes 继续，整个过程如下图所示：

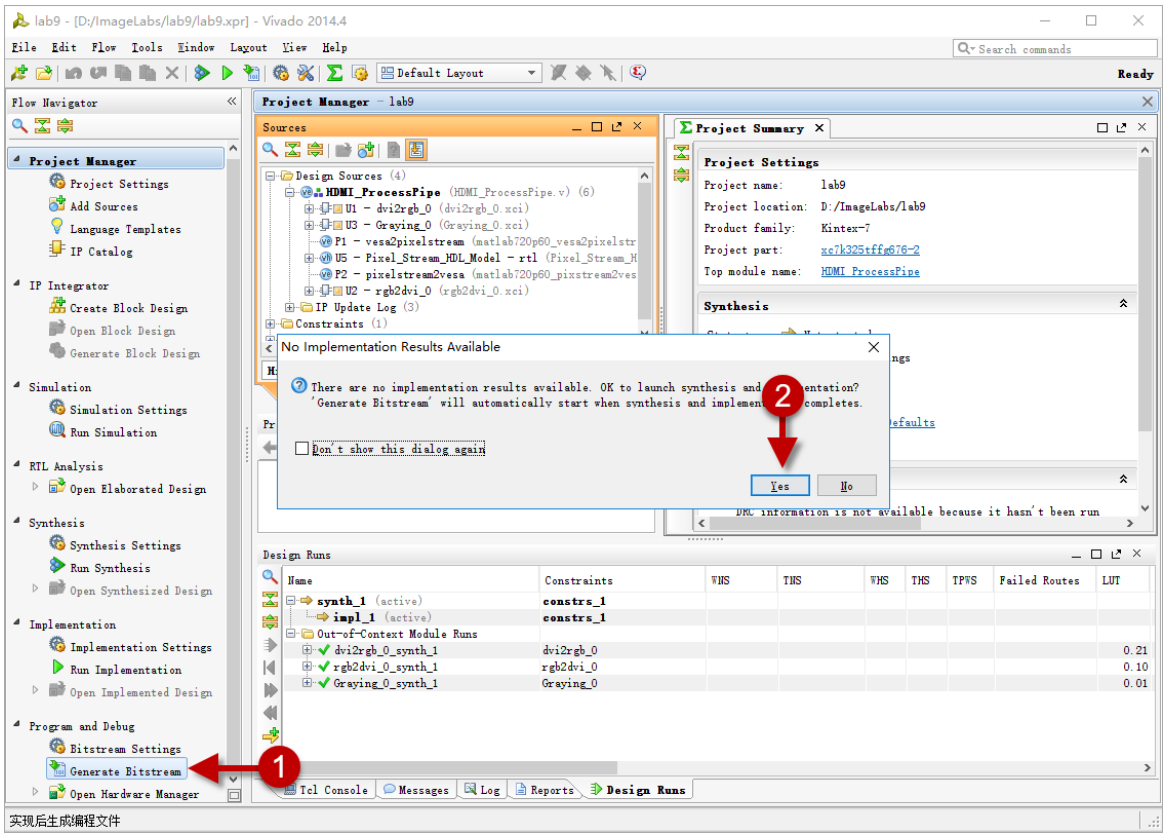



图 3-34 生成 Bitstream

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：

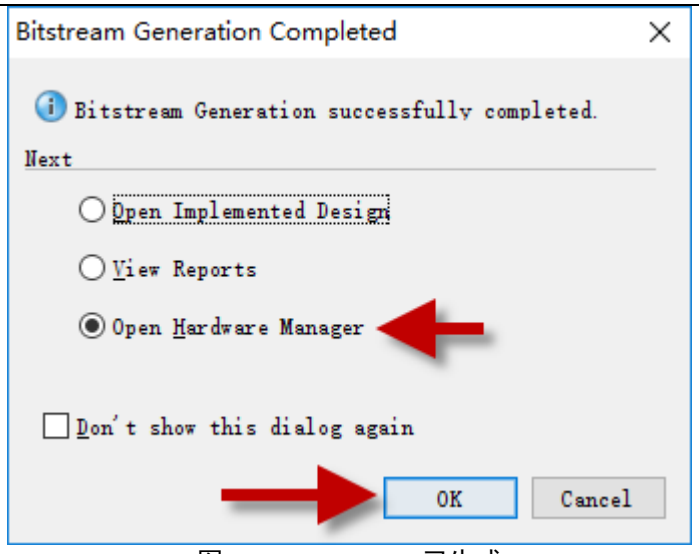



图 3-35 Bitstream 已生成

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

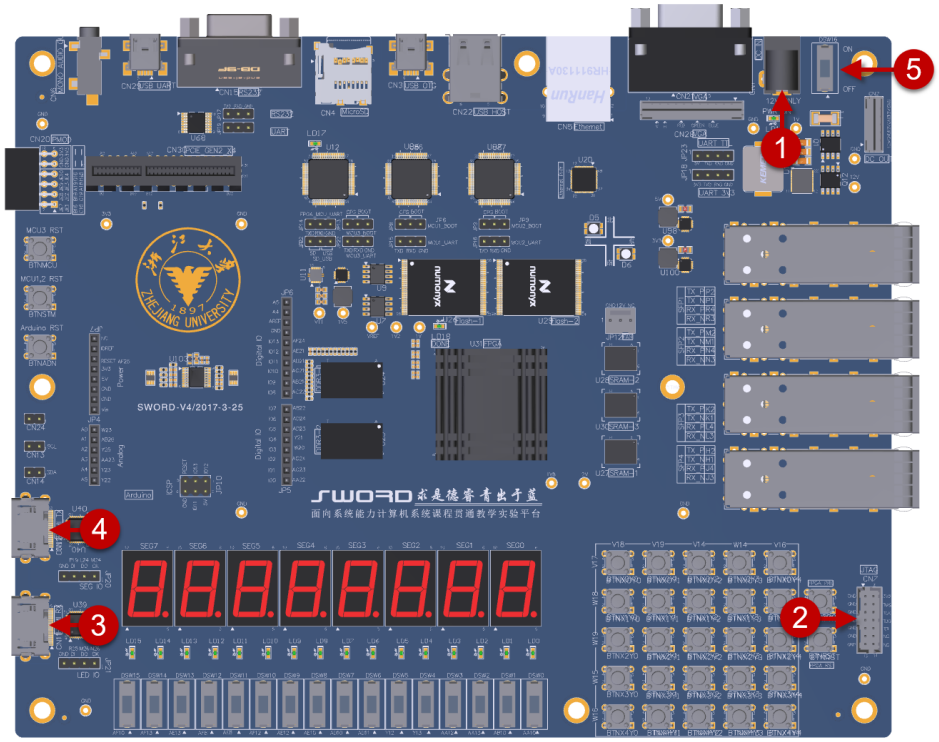



图 3-36 硬件连接对应位置

连接好后的效果如下图所示：

图 3-37 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：

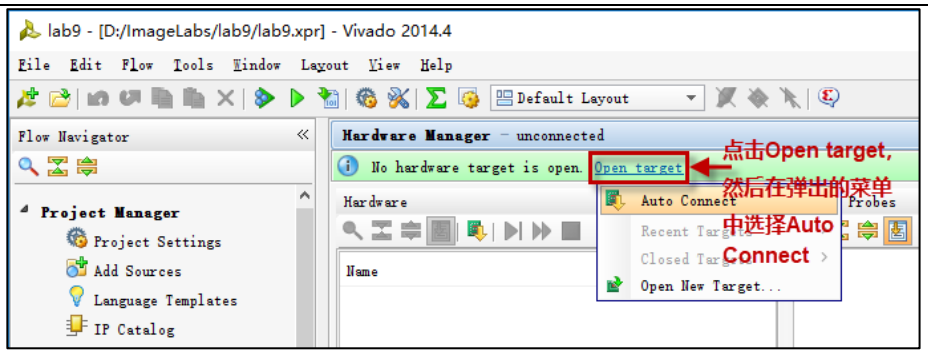



图 3-38 在 Hardware Manager 下 Open target

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：

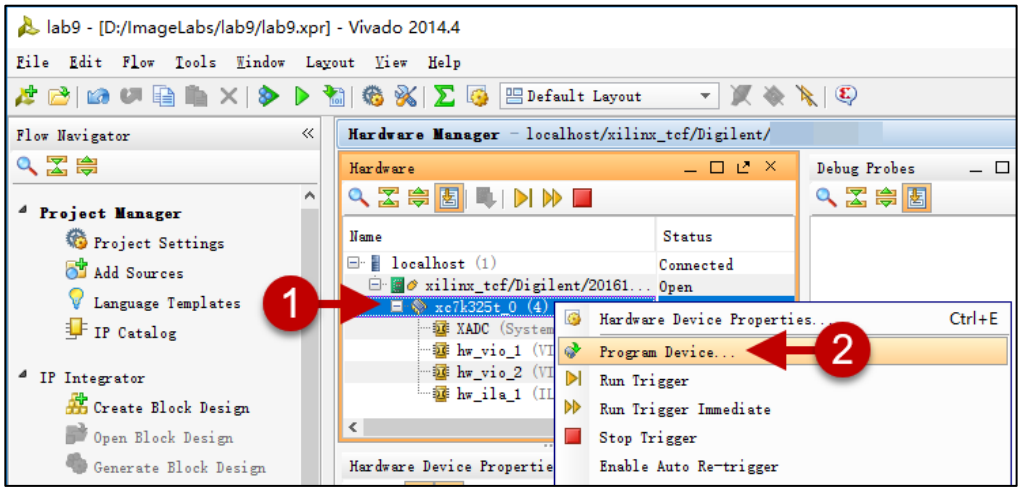



图 3-39 选择目标器件先下载 Bitstream

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。

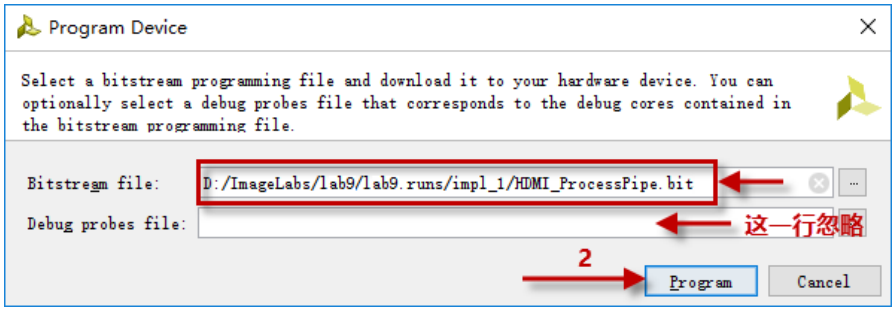



图 3-40 确认文件无误后点击 Program 下载

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。

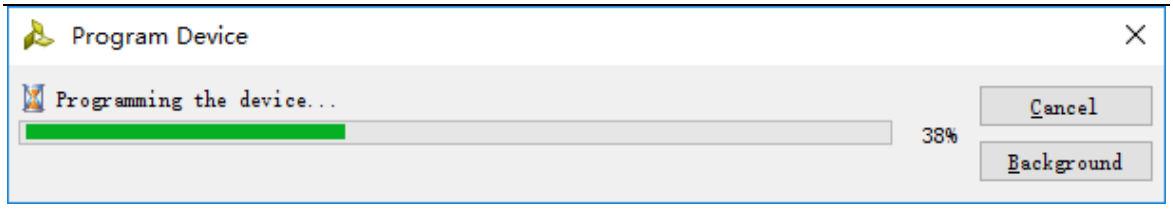



图 3-41 下载进度条显示

# 4. 实验结果

此时我们可以将连接 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 4-1 显示器上显示的结果画面