

# Lab2 使用Sobel算子提取Lena的边缘

## 目标

在完成本实验后，您应当学会：

- 在Vitis HLS中创建使用AXI Stream接口的Sobel IP
- 在Vivado中构建包含DMA的IP集成
- 在PYNQ中学习DMA等接口的使用
- 在PYNQ中构建一个高效的Sobel图像处理应用

## 环境要求

- PYNQ-Z2远程实验室服务或物理板卡
- Vitis HLS
- Vivado

## 实验步骤

由于整体开发流程已经在Lab1中详细介绍，本部分仅重点介绍与Lab1中的区别之处，重复部分不再赘述

### 1. 在Vitis\_HLS中设计Sobel IP

1. 参考Lab1第一部分，创建项目，输入项目名称为sobel\_hls\_prj
2. 在Add/Remove Design Files 窗口中 Add Files 时选择以下文件
  1. sobel.h
  2. sobel.cpp
3. 在Add/Remove Testbench Files 窗口中 Add Files 时选择以下文件
  1. sobel\_test.cpp - TB
  2. data.txt - 原始图像
  3. dst\_1280x720 - 参考结果
  4. imgo.txt - TB输出的图像
4. 由于部分版本的Vitis HLS填写**相对路径**可能出现的问题，请将 sobel\_test.cpp 中第83行的 imgo.txt 的路径替换为您系统中的 imgo.txt 的**绝对路径**
5. 参考Lab1第一部分，依次进行C-Simulation, C-Synthesis, C/RTL Co-simulation和导出RTL
6. 解压 export.zip 所得到的文件夹即为下一步所用的IP

### 2. 在Vivado中进行IP集成

#### 2.1 创建一个新Vivado项目

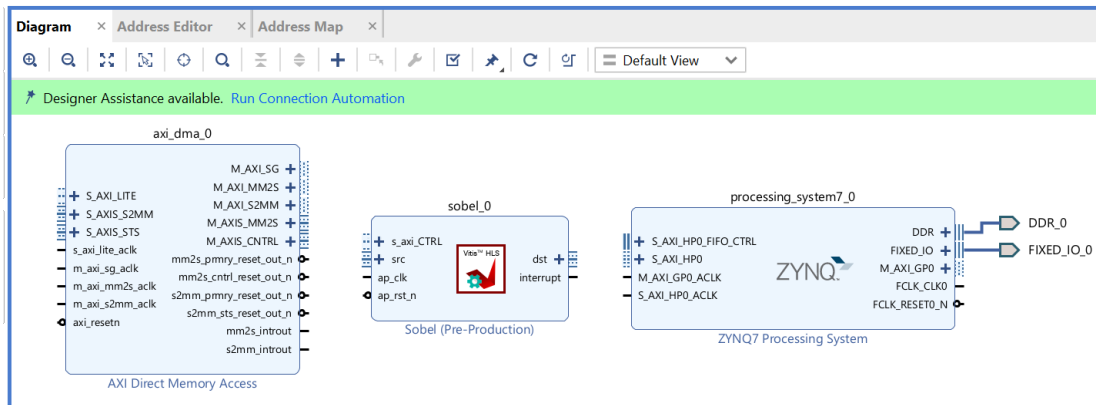
参考Lab1的2.1完成，输入项目名称为sobel\_vivado\_prj

## 2.2 导入IP

参考Lab1的2.2完成，选择文件夹时选择1.5中解压所得到的文件夹 /export

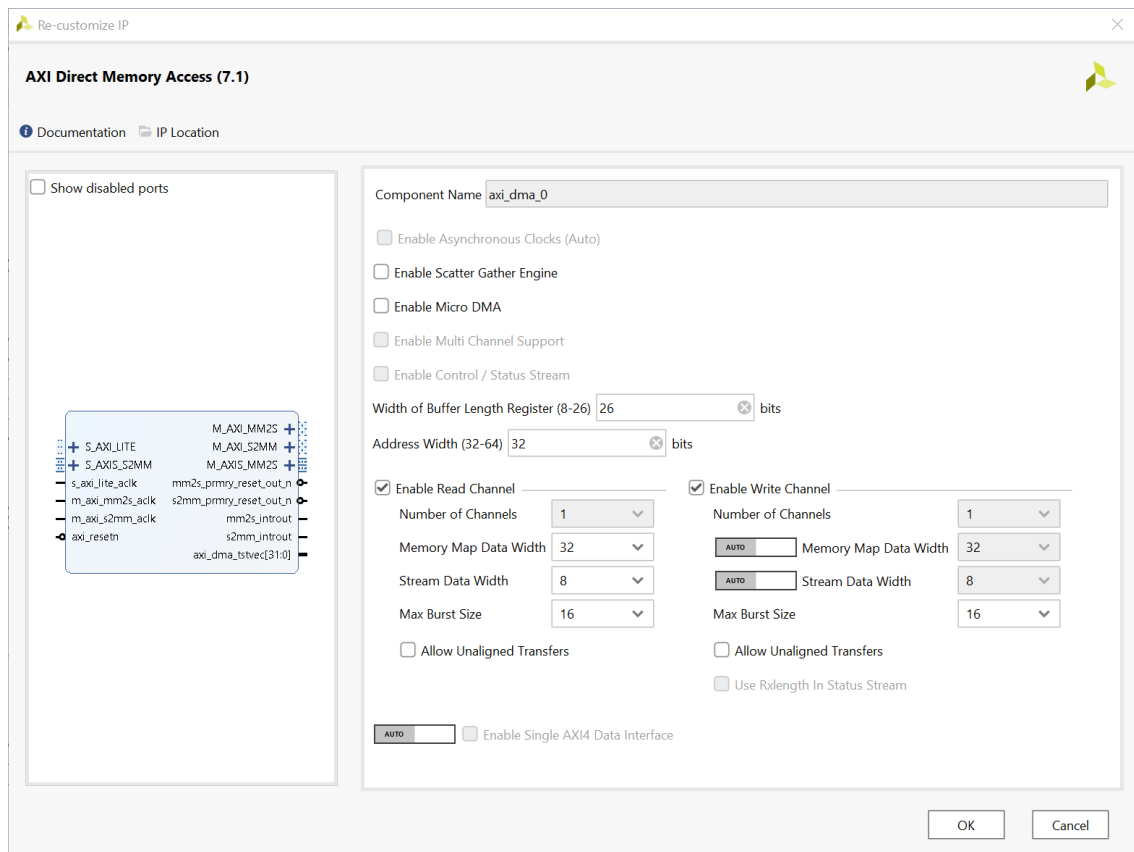
## 2.3 添加PS7, AXI DMA与Sobel IP

1. 参考Lab1的2.3.1-2.3.4的步骤，创建一个**Block Design**并添加**ZYNQ7 Processing System**
2. 参考Lab1的2.3.5，搜索并添加名为**Sobel**和**AXI Direct Memory Access**的两个IP到Block Design中



## 2.4 配置AXI DMA

1. 双击**AXI Direct Memory Access**模块进行IP的配置
2. 在弹出的**Re-customize IP**中取消勾选**Enable Scatter Gather Engine**
3. 将**Width of Buffer Length Register (8-26) bits**配置为26
4. 将左侧的**Stream Data Width**配置为8
5. 最后点击**OK**

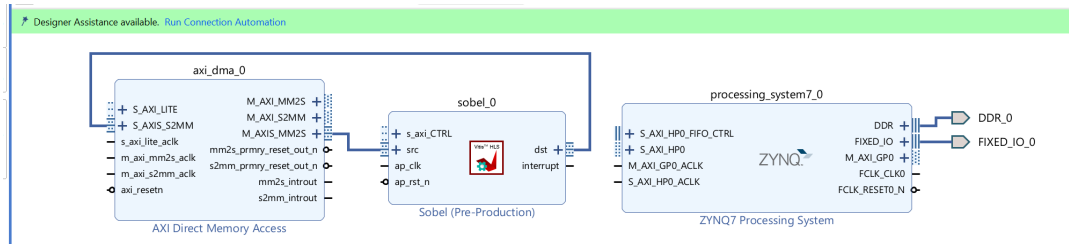


## 2.5 连线

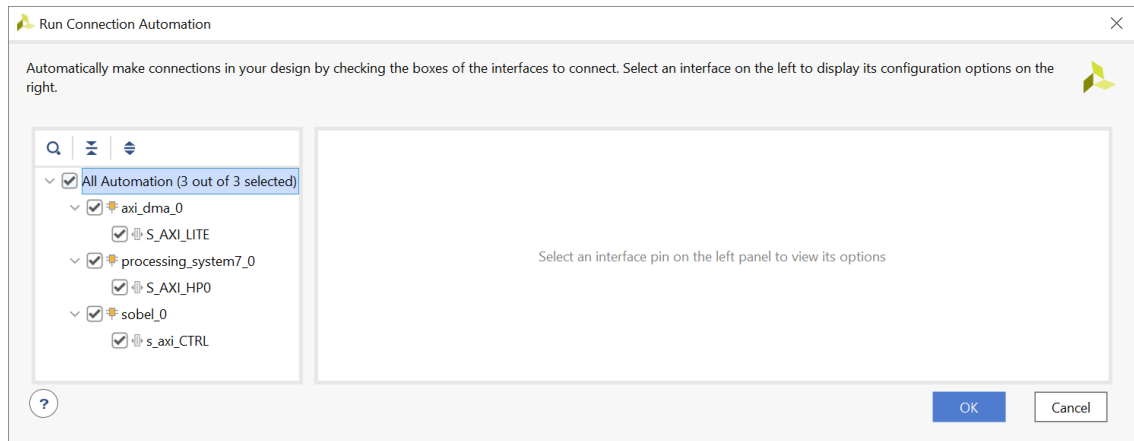
1. 将以下接口进行连接

1. **axi\_dma\_0::M\_AXIS\_MM2S** to **sobel\_0::src**

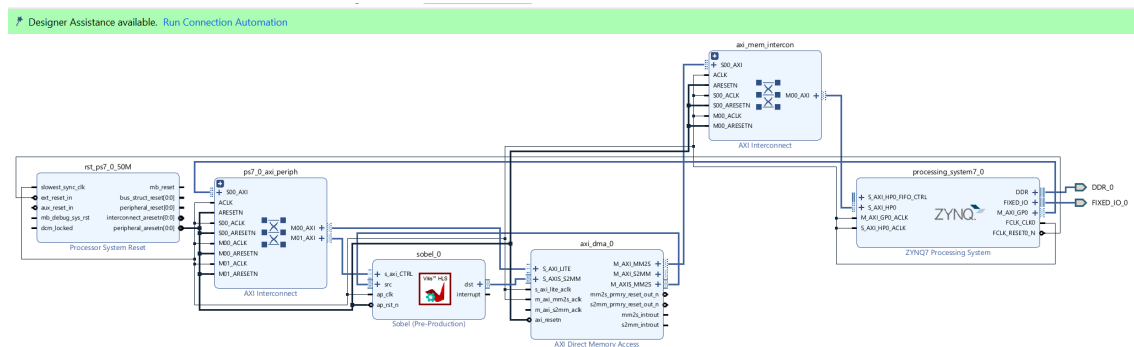
2. **axi\_dma\_0::S\_AXIS\_S2MM** to **sobel\_0::dst**



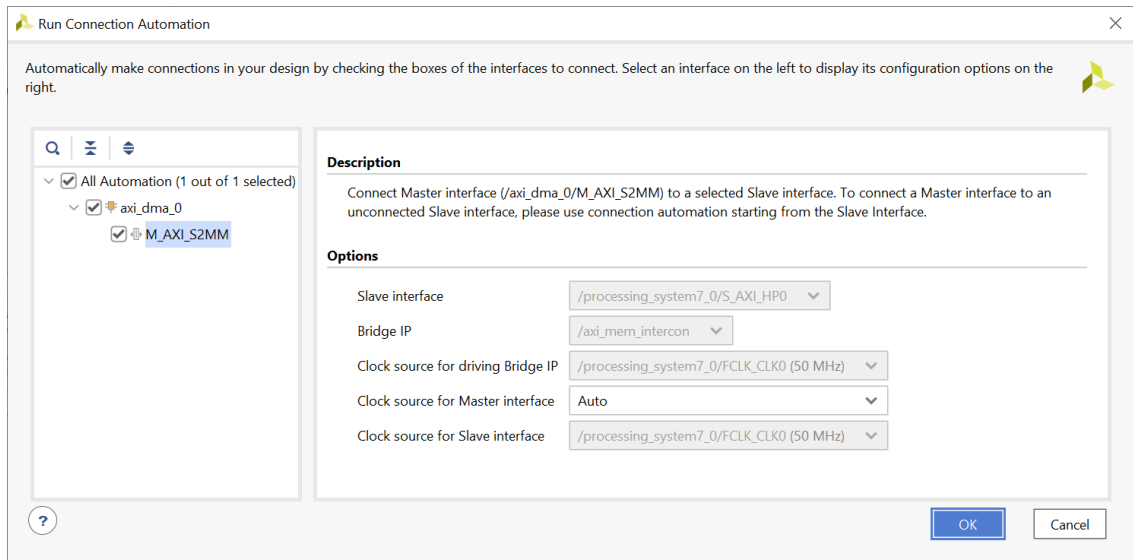
2. 点击上方的**Run Connection Automation**，弹出对应窗口，将左侧**All Automation** 选项勾选上，再点击**OK**



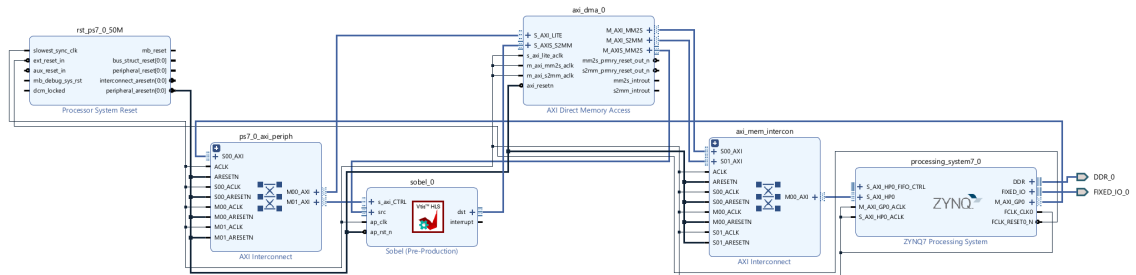
3. 可得到如下连接，同时可以发现窗口上方仍然有**Run Connection Automation**



4. 再次点击**Run Connection Automation**，弹出对应窗口，将左侧**All Automation** 选项勾选上，再点击**OK**



5. 待自动连接完毕，可以得到如下的Block Design



6. 参考Lab1的2.3.8-2.3.11，完成Validation和Create HDL Wrapper等操作

## 2.5 综合与导出比特流

1. 参考Lab1的2.4的操作，完成综合与生成比特流的操作。

## 3. 构建PYNQ设计

### 3.1 提取bit与hwh文件

1. 将\sobel\_vivado\_prj\sobel\_vivado\_prj.runs\impl\_1目录下的design\_1\_wrapper.bit文件复制到自己的文件夹中保存，并重命名为sobel.bit
2. 将\sobel\_vivado\_prj\sobel\_vivado\_prj.gen\sources\_1\bd\design\_1\hw\_handoff目录下的design\_1.hwh文件复制到自己的文件夹中保存，并重命名为sobel.hwh

### 3.2 访问Jupyter

1. 请先完成PYNQ远程实验室的账号注册与Jupyter访问
2. 登录Jupyter界面，点击界面右上方的upload按钮，将 /jupyter 目录下的以下文件上传到开发板上
  - Lab2\_sobel\_axis.ipynb - Notebook文件
  - lena.jpg - 待处理的图片
3. 在Lab2\_sobel\_axis.ipynb的目录下新建 /overlay\_axis 文件夹，将3.1中得到的sobel.bit与sobel.hwh文件传入到该目录下
  - 如果你在前面操作中导出失败了，你也可以先使用提供的 /overlay\_axis 目录下的sobel.bit与sobel.hwh文件上传，以完成余下上板实验

### 3.3 部署与运行Overlay

1. 在Jupyter Notebook中进入到Lab2\_sobel\_axis.ipynb页面，Kernel自动加载完成显示为Python3字样
2. 点击窗口上侧的Run按钮，Jupyter Notebook会执行当前Cell，同时自动切换到下一个Cell
3. 完成按照顺序依次点击Run至结束即可，各代码块的含义在Jupyter Notebook中已经标注，请阅读Jupyter Notebook中的信息继续完成实验。