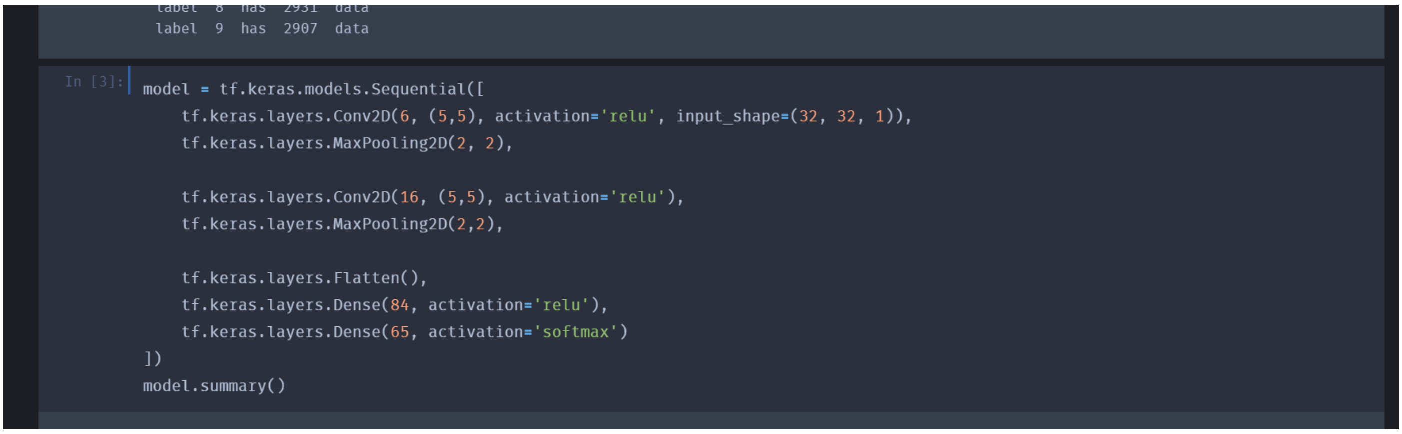
步骤 1 Tensoflow 搭建 Lenet 神经网络

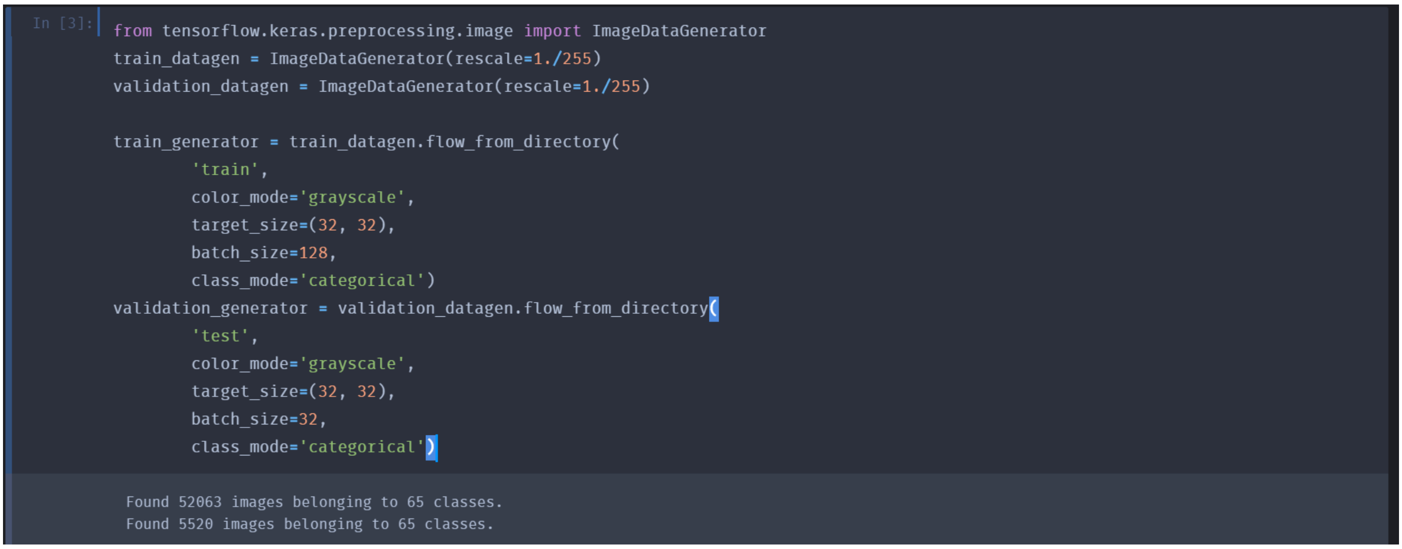
1 搭建网络



```
label  8  has  2931  data
label  9  has  2907  data
```

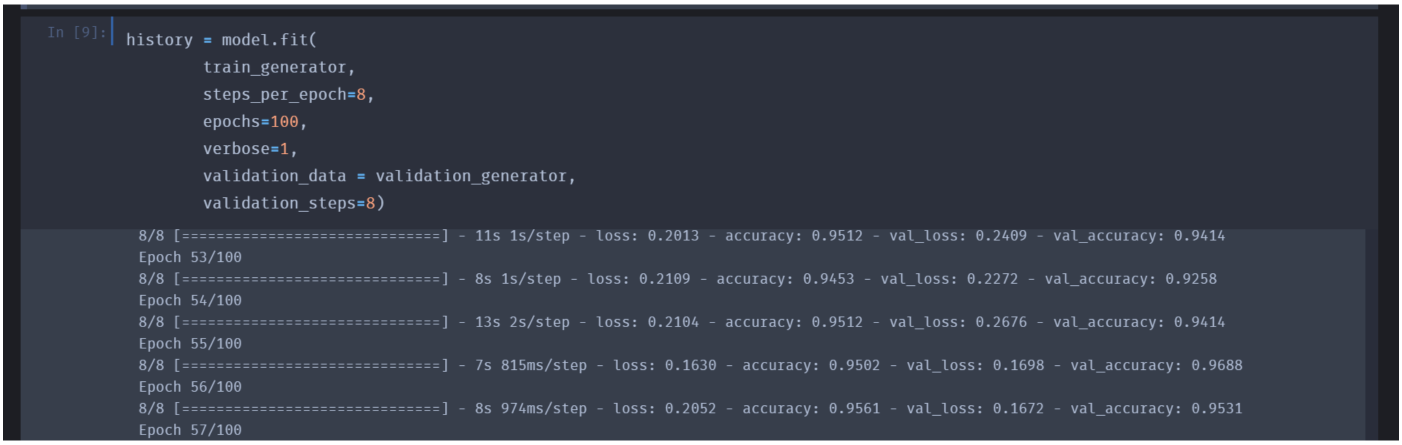
```
In [3]:  model = tf.keras.models.Sequential([
             tf.keras.layers.Conv2D(6, (5,5), activation='relu', input_shape=(32, 32, 1)),
             tf.keras.layers.MaxPooling2D(2, 2),

             tf.keras.layers.Conv2D(16, (5,5), activation='relu'),
             tf.keras.layers.MaxPooling2D(2,2),

             tf.keras.layers.Flatten(),
             tf.keras.layers.Dense(84, activation='relu'),
             tf.keras.layers.Dense(65, activation='softmax')
         ])
         model.summary()
```

2 载入数据集



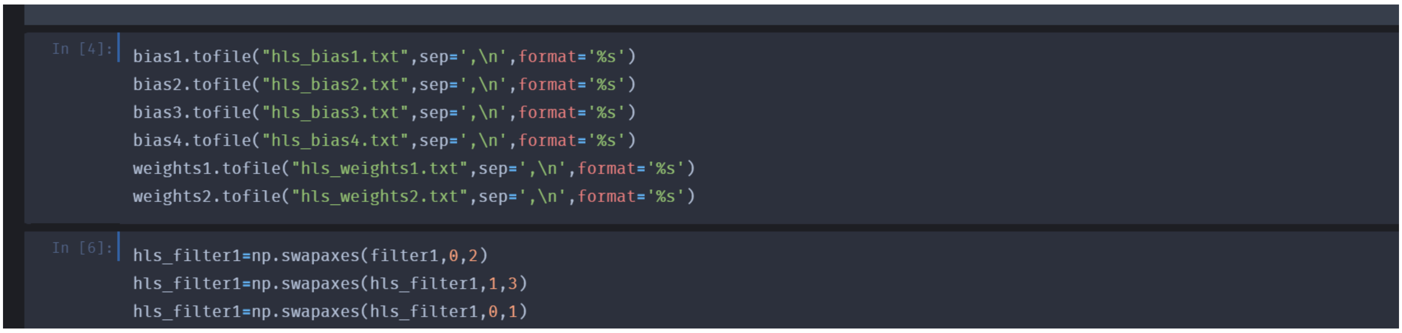
```
In [3]:  from tensorflow.keras.preprocessing.image import ImageDataGenerator
         train_datagen = ImageDataGenerator(rescale=1./255)
         validation_datagen = ImageDataGenerator(rescale=1./255)

         train_generator = train_datagen.flow_from_directory(
                 'train',
                 color_mode='grayscale',
                 target_size=(32, 32),
                 batch_size=128,
                 class_mode='categorical')
         validation_generator = validation_datagen.flow_from_directory(
                 'test',
                 color_mode='grayscale',
                 target_size=(32, 32),
                 batch_size=32,
                 class_mode='categorical')

           Found 52063 images belonging to 65 classes.
           Found 5520 images belonging to 65 classes.
```

3 开始训练



```
In [9]:  history = model.fit(
                 train_generator,
                 steps_per_epoch=8,
                 epochs=100,
                 verbose=1,
                 validation_data = validation_generator,
                 validation_steps=8)

           8/8 [==============================] - 11s 1s/step - loss: 0.2013 - accuracy: 0.9512 - val_loss: 0.2409 - val_accuracy: 0.9414
           Epoch 53/100
           8/8 [==============================] - 8s 1s/step - loss: 0.2109 - accuracy: 0.9453 - val_loss: 0.2272 - val_accuracy: 0.9258
           Epoch 54/100
           8/8 [==============================] - 13s 2s/step - loss: 0.2104 - accuracy: 0.9512 - val_loss: 0.2676 - val_accuracy: 0.9414
           Epoch 55/100
           8/8 [==============================] - 7s 815ms/step - loss: 0.1630 - accuracy: 0.9502 - val_loss: 0.1698 - val_accuracy: 0.9688
           Epoch 56/100
           8/8 [==============================] - 8s 974ms/step - loss: 0.2052 - accuracy: 0.9561 - val_loss: 0.1672 - val_accuracy: 0.9531
           Epoch 57/100
```

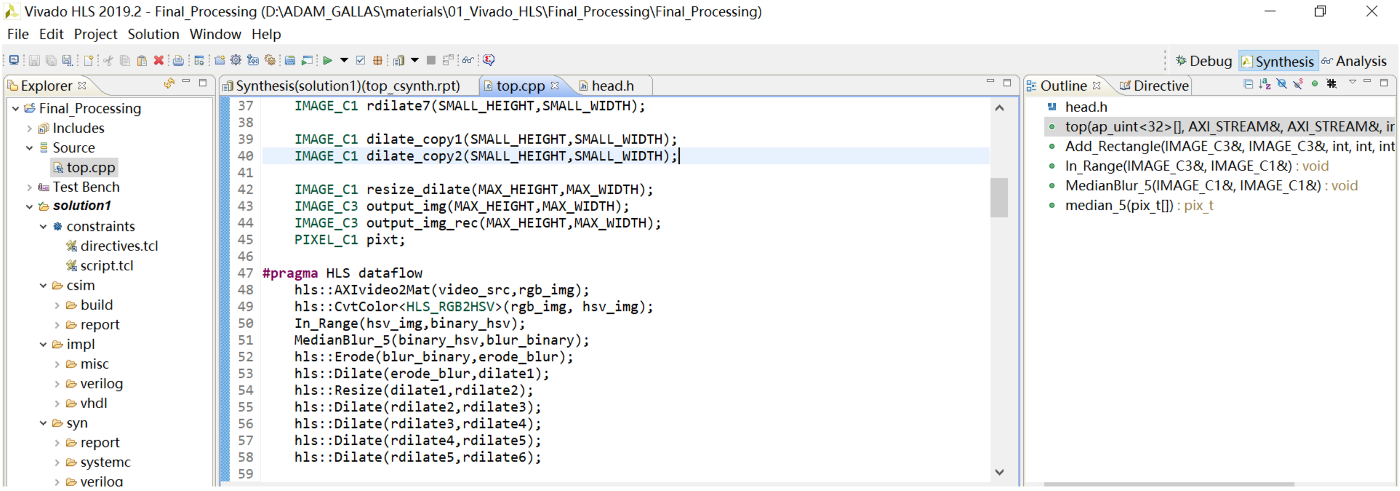
4 训练完成，导出模型



```
In [4]:  bias1.tofile("hls_bias1.txt",sep=',\n',format='%s')
         bias2.tofile("hls_bias2.txt",sep=',\n',format='%s')
         bias3.tofile("hls_bias3.txt",sep=',\n',format='%s')
         bias4.tofile("hls_bias4.txt",sep=',\n',format='%s')
         weights1.tofile("hls_weights1.txt",sep=',\n',format='%s')
         weights2.tofile("hls_weights2.txt",sep=',\n',format='%s')

In [6]:  hls_filter1=np.swapaxes(filter1,0,2)
         hls_filter1=np.swapaxes(hls_filter1,1,3)
         hls_filter1=np.swapaxes(hls_filter1,0,1)
```

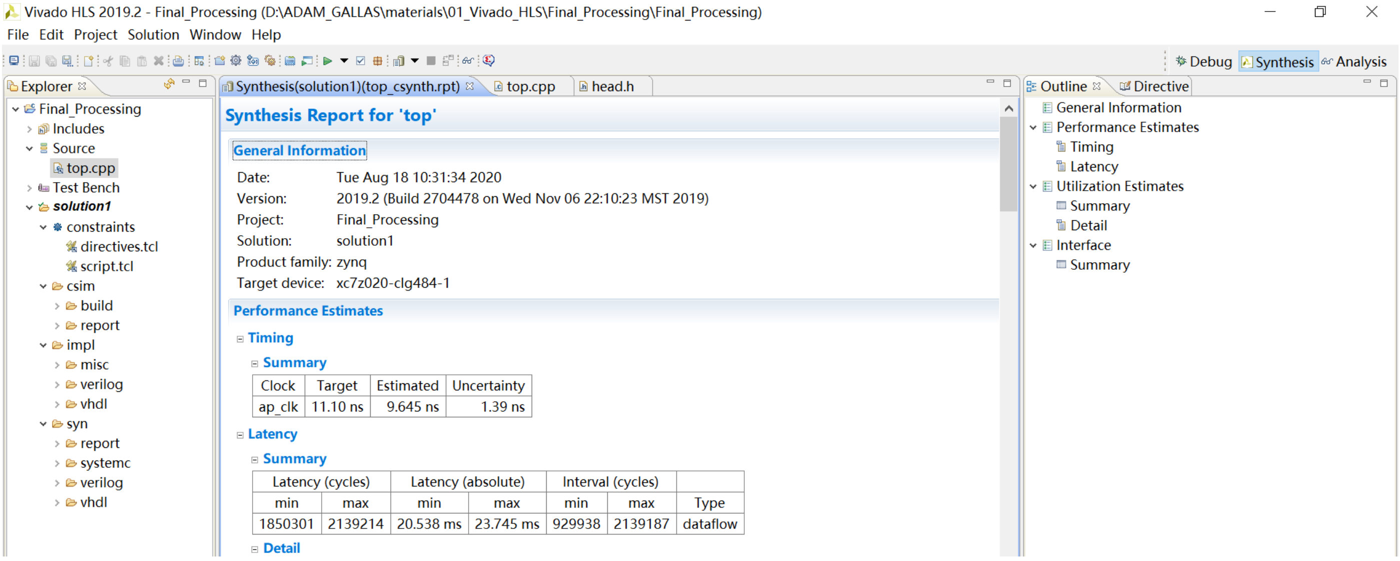
步骤 2 Vivado HLS 设计图像处理算法

1 设计算法


Vivado HLS 2019.2 - Final_Processing (D:\ADAM_GALLAS\materials\01_Vivado_HLS\Final_Processing\Final_Processing)
File Edit Project Solution Window Help
Debug
Synthesis
Analysis
Explorer
Final_Processing
Includes
Source
top.cpp
Test Bench
solution1
constraints
directives.tcl
script.tcl
csim
build
report
impl
misc
verilog
vhdl
syn
report
systemc
verilog
Synthesis(solution1)(top_csynth.rpt)
top.cpp
head.h
Outline
Directive
head.h
top(ap_uint<32>[], AXI_STREAM&, AXI_STREAM&, ir
Add_Rectangle(IMAGE_C3&, IMAGE_C3&, int, int, int
In_Range(IMAGE_C3&, IMAGE_C1&) : void
MedianBlur_5(IMAGE_C1&, IMAGE_C1&) : void
median_5(pix_t[]) : pix_t

```
    IMAGE_C1 rdilate7(SMALL_HEIGHT,SMALL_WIDTH);

    IMAGE_C1 dilate_copy1(SMALL_HEIGHT,SMALL_WIDTH);
    IMAGE_C1 dilate_copy2(SMALL_HEIGHT,SMALL_WIDTH);

    IMAGE_C1 resize_dilate(MAX_HEIGHT,MAX_WIDTH);
    IMAGE_C3 output_img(MAX_HEIGHT,MAX_WIDTH);
    IMAGE_C3 output_img_rec(MAX_HEIGHT,MAX_WIDTH);
    PIXEL_C1 pixt;

#pragma HLS dataflow
    hls::AXIvideo2Mat(video_src,rgb_img);
    hls::CvtColor<HLS_RGB2HSV>(rgb_img, hsv_img);
    In_Range(hsv_img,binary_hsv);
    MedianBlur_5(binary_hsv,blur_binary);
    hls::Erode(blur_binary,erode_blur);
    hls::Dilate(erode_blur,dilate1);
    hls::Resize(dilate1,rdilate2);
    hls::Dilate(rdilate2,rdilate3);
    hls::Dilate(rdilate3,rdilate4);
    hls::Dilate(rdilate4,rdilate5);
    hls::Dilate(rdilate5,rdilate6);
```

2 进行 C Simulation，Synthesis，C/RTL Cosimulation



Synthesis Report for 'top'

General Information

Date: Tue Aug 18 10:31:34 2020
Version: 2019.2 (Build 2704478 on Wed Nov 06 22:10:23 MST 2019)
Project: Final_Processing
Solution: solution1
Product family: zynq
Target device: xc7z020-clg484-1

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	11.10 ns	9.645 ns	1.39 ns

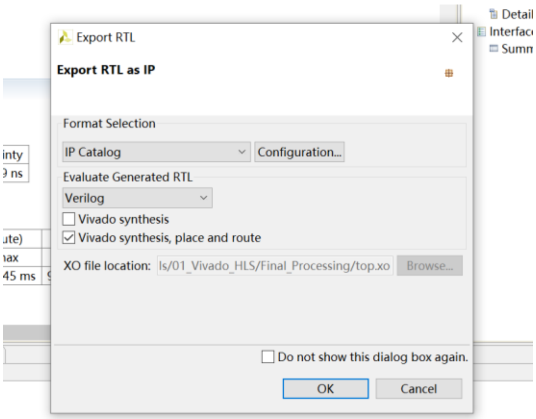
Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
1850301	2139214	20.538 ms	23.745 ms	929938	2139187	dataflow

Detail

3 导出 IP 核



IP Catalog

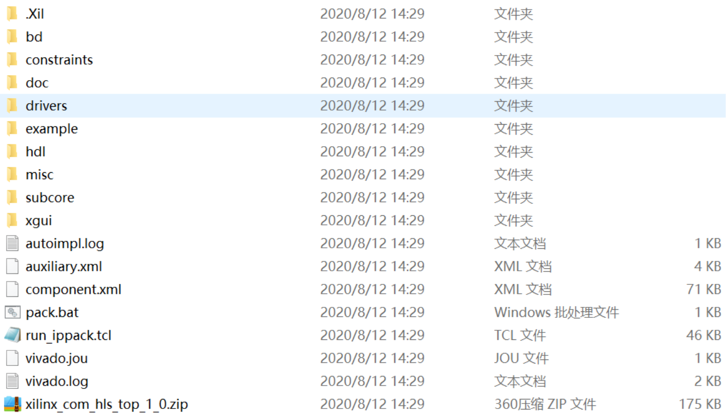
Verilog

☐ Vivado synthesis

☑ Vivado synthesis, place and route

XO file location: ls/01_Vivado_HLS/Final_Processing/top.xo

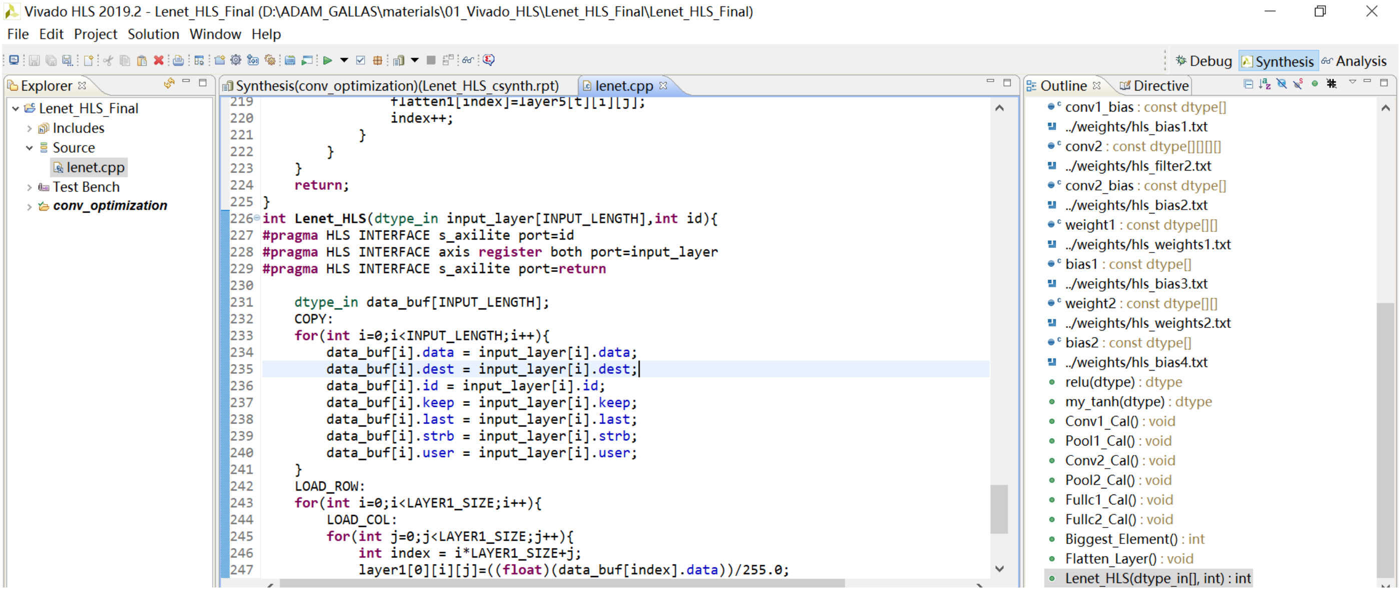
☐ Do not show this dialog box again.



.Xil	2020/8/12 14:29	文件夹	
bd	2020/8/12 14:29	文件夹	
constraints	2020/8/12 14:29	文件夹	
doc	2020/8/12 14:29	文件夹	
drivers	2020/8/12 14:29	文件夹	
example	2020/8/12 14:29	文件夹	
hdl	2020/8/12 14:29	文件夹	
misc	2020/8/12 14:29	文件夹	
subcore	2020/8/12 14:29	文件夹	
xgui	2020/8/12 14:29	文件夹	
autoimpl.log	2020/8/12 14:29	文本文档	1 KB
auxiliary.xml	2020/8/12 14:29	XML 文档	4 KB
component.xml	2020/8/12 14:29	XML 文档	71 KB
pack.bat	2020/8/12 14:29	Windows 批处理文件	1 KB
run_ippack.tcl	2020/8/12 14:29	TCL 文件	46 KB
vivado.jou	2020/8/12 14:29	JOU 文件	1 KB
vivado.log	2020/8/12 14:29	文本文档	2 KB
xilinx_com_hls_top_1_0.zip	2020/8/12 14:29	360压缩 ZIP 文件	175 KB

步骤 3 Vivado HLS 设计神经网络推理计算程序

1 编写程序

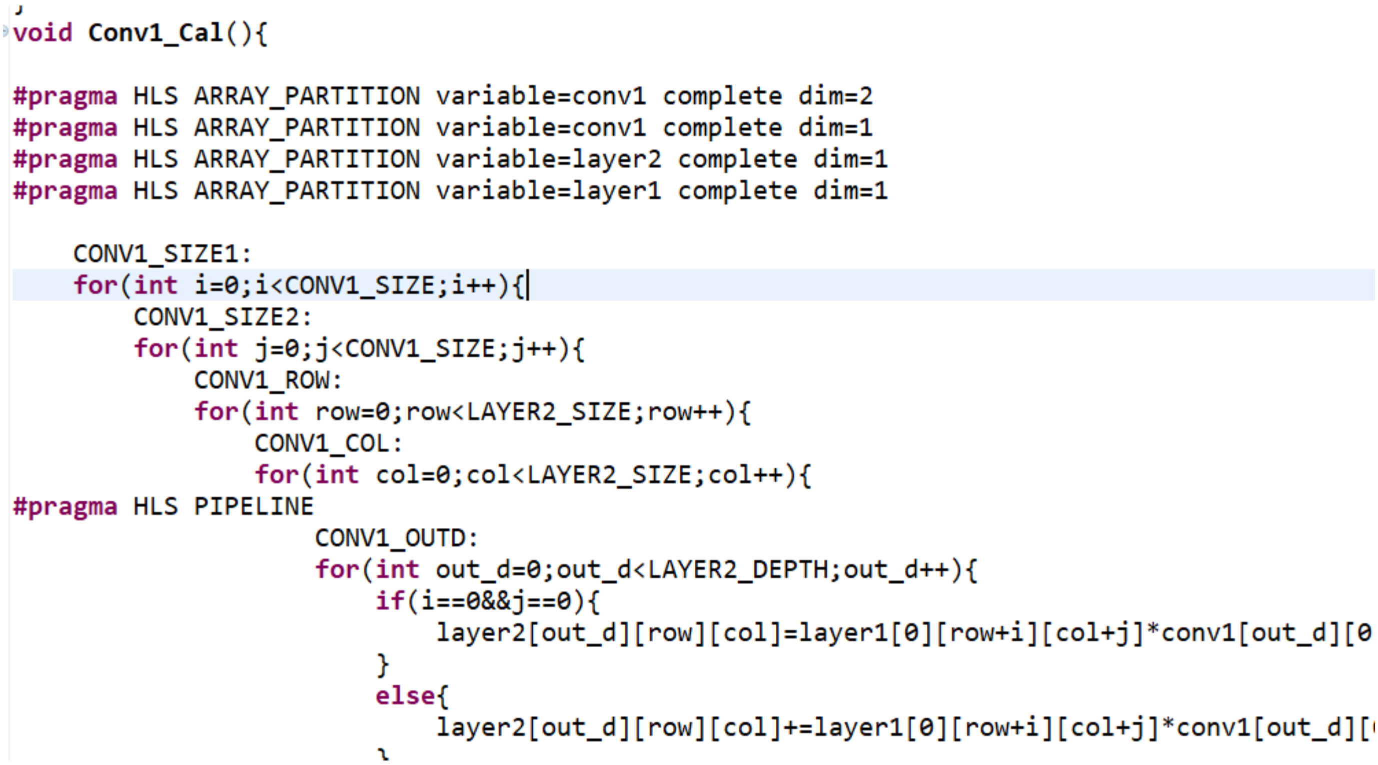


Vivado HLS 2019.2 - Lenet_HLS_Final (D:\ADAM_GALLAS\materials\01_Vivado_HLS\Lenet_HLS_Final\Lenet_HLS_Final)

```
                flatten1[index]=layer5[t][i][j];
                index++;
            }
        }
    }
    return;
}
int Lenet_HLS(dtype_in input_layer[INPUT_LENGTH],int id){
#pragma HLS INTERFACE s_axilite port=id
#pragma HLS INTERFACE axis register both port=input_layer
#pragma HLS INTERFACE s_axilite port=return

    dtype_in data_buf[INPUT_LENGTH];
    COPY:
    for(int i=0;i<INPUT_LENGTH;i++){
        data_buf[i].data = input_layer[i].data;
        data_buf[i].dest = input_layer[i].dest;
        data_buf[i].id = input_layer[i].id;
        data_buf[i].keep = input_layer[i].keep;
        data_buf[i].last = input_layer[i].last;
        data_buf[i].strb = input_layer[i].strb;
        data_buf[i].user = input_layer[i].user;
    }
    LOAD_ROW:
    for(int i=0;i<LAYER1_SIZE;i++){
        LOAD_COL:
        for(int j=0;j<LAYER1_SIZE;j++){
            int index = i*LAYER1_SIZE+j;
            layer1[0][i][j]=((float)(data_buf[index].data))/255.0;
```

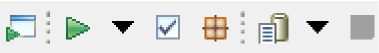
```
conv1_bias : const dtype[]
../weights/hls_bias1.txt
conv2 : const dtype[][][][]
../weights/hls_filter2.txt
conv2_bias : const dtype[]
../weights/hls_bias2.txt
weight1 : const dtype[][]
../weights/hls_weights1.txt
bias1 : const dtype[]
../weights/hls_bias3.txt
weight2 : const dtype[][]
../weights/hls_weights2.txt
bias2 : const dtype[]
../weights/hls_bias4.txt
relu(dtype) : dtype
my_tanh(dtype) : dtype
Conv1_Cal() : void
Pool1_Cal() : void
Conv2_Cal() : void
Pool2_Cal() : void
Fullc1_Cal() : void
Fullc2_Cal() : void
Biggest_Element() : int
Flatten_Layer() : void
Lenet_HLS(dtype_in[], int) : int
```

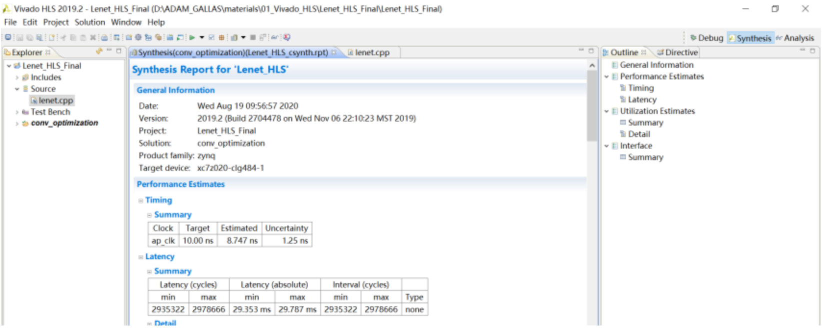
2 添加并行优化



```
}
void Conv1_Cal(){

#pragma HLS ARRAY_PARTITION variable=conv1 complete dim=2
#pragma HLS ARRAY_PARTITION variable=conv1 complete dim=1
#pragma HLS ARRAY_PARTITION variable=layer2 complete dim=1
#pragma HLS ARRAY_PARTITION variable=layer1 complete dim=1

    CONV1_SIZE1:
    for(int i=0;i<CONV1_SIZE;i++){
        CONV1_SIZE2:
        for(int j=0;j<CONV1_SIZE;j++){
            CONV1_ROW:
            for(int row=0;row<LAYER2_SIZE;row++){
                CONV1_COL:
                for(int col=0;col<LAYER2_SIZE;col++){
#pragma HLS PIPELINE
                    CONV1_OUTD:
                    for(int out_d=0;out_d<LAYER2_DEPTH;out_d++){
                        if(i==0&&j==0){
                            layer2[out_d][row][col]=layer1[0][row+i][col+j]*conv1[out_d][0
                        }
                        else{
                            layer2[out_d][row][col]+=layer1[0][row+i][col+j]*conv1[out_d][(
                        }
```

3 进行 C Simulation，Synthesis，C/RTL Cosimulation





Vivado HLS 2019.2 - Lenet_HLS_Final (D:\ADAM_GALLAS\materials\01_Vivado_HLS\Lenet_HLS_Final\Lenet_HLS_Final)

Synthesis Report for 'Lenet_HLS'

General Information

Date: Wed Aug 19 09:56:57 2020
Version: 2019.2 (Build 2704478 on Wed Nov 06 22:10:23 MST 2019)
Project: Lenet_HLS_Final
Solution: conv_optimization
Product family: zynq
Target device: xc7z020-clg484-1

Performance Estimates

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	8.747 ns	1.25 ns

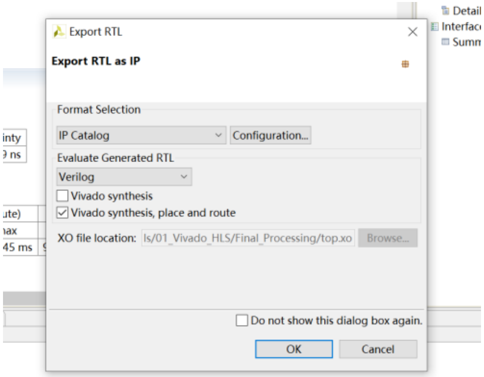
Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
2935322	2978666	29.353 ms	29.787 ms	2935322	2978666	none

Detail

4 导出 IP 核



Export RTL

Export RTL as IP

Format Selection

IP Catalog | Configuration...

Evaluate Generated RTL

Verilog

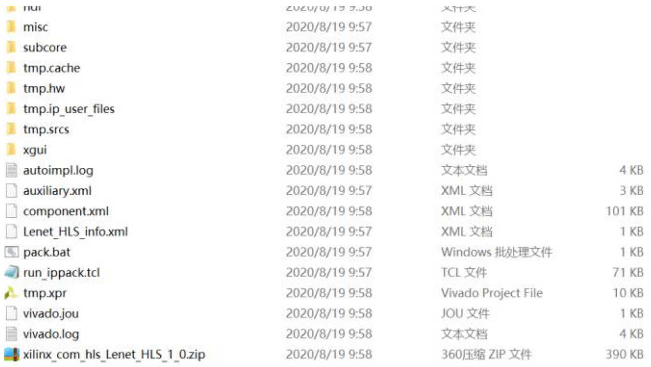
□ Vivado synthesis

☑ Vivado synthesis, place and route

XO file location: ls/01_Vivado_HLS/Final_Processing/top.xo Browse...

□ Do not show this dialog box again.

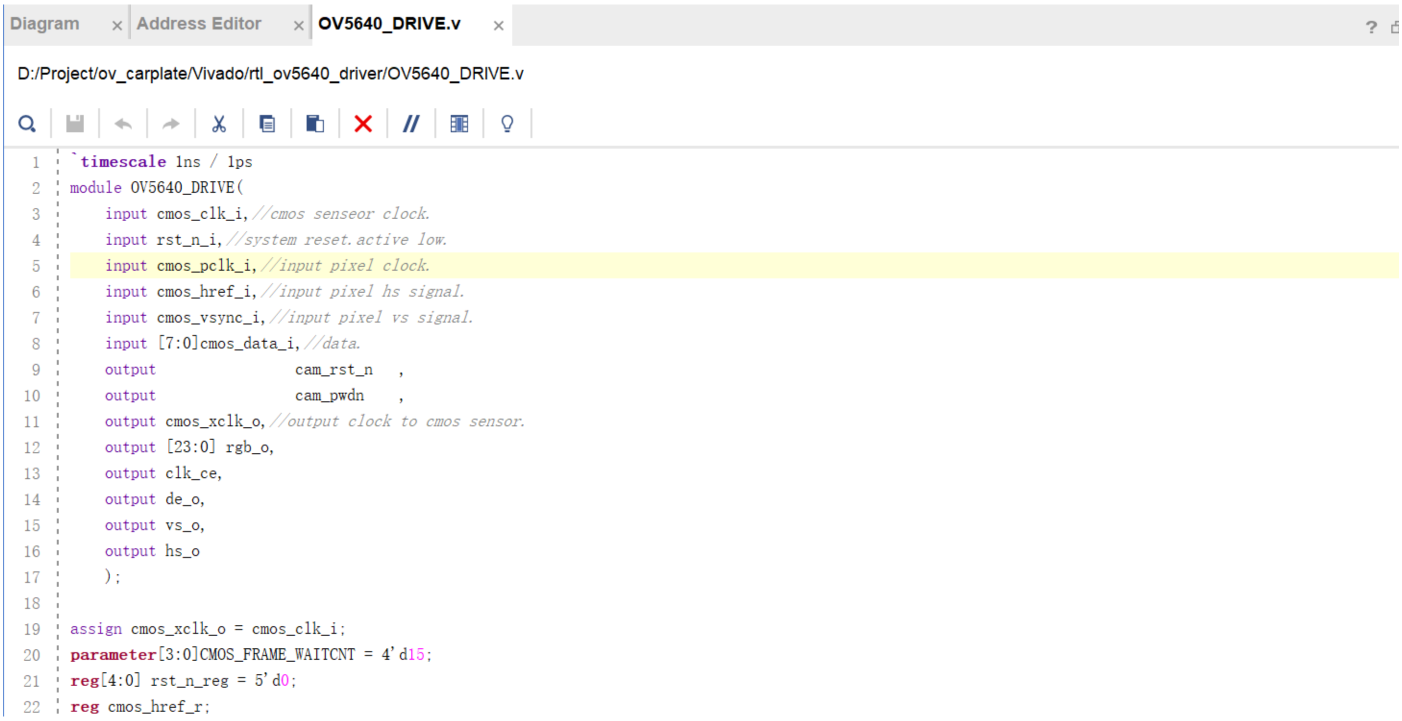
OK Cancel



hdl	2020/8/19 9:58	文件夹	
misc	2020/8/19 9:57	文件夹	
subcore	2020/8/19 9:57	文件夹	
tmp.cache	2020/8/19 9:58	文件夹	
tmp.hw	2020/8/19 9:58	文件夹	
tmp.ip_user_files	2020/8/19 9:58	文件夹	
tmp.srcs	2020/8/19 9:58	文件夹	
xgui	2020/8/19 9:58	文件夹	
autoimpl.log	2020/8/19 9:58	文本文档	4 KB
auxiliary.xml	2020/8/19 9:57	XML 文档	3 KB
component.xml	2020/8/19 9:58	XML 文档	101 KB
Lenet_HLS_info.xml	2020/8/19 9:57	XML 文档	1 KB
pack.bat	2020/8/19 9:57	Windows 批处理文件	1 KB
run_ippack.tcl	2020/8/19 9:57	TCL 文件	71 KB
tmp.xpr	2020/8/19 9:58	Vivado Project File	10 KB
vivado.jou	2020/8/19 9:58	JOU 文件	1 KB
vivado.log	2020/8/19 9:58	文本文档	4 KB
xilinx_com_hls_Lenet_HLS_1_0.zip	2020/8/19 9:58	360压缩 ZIP 文件	390 KB

步骤 4 Verilog 编写摄像头驱动程序

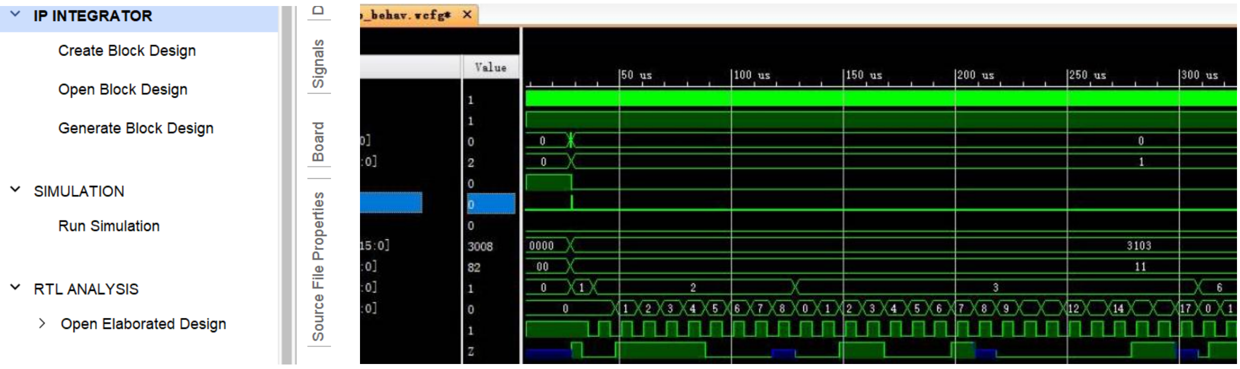
1 编写程序



D:/Project/ov_carplate/Vivado/rtl_ov5640_driver/OV5640_DRIVE.v

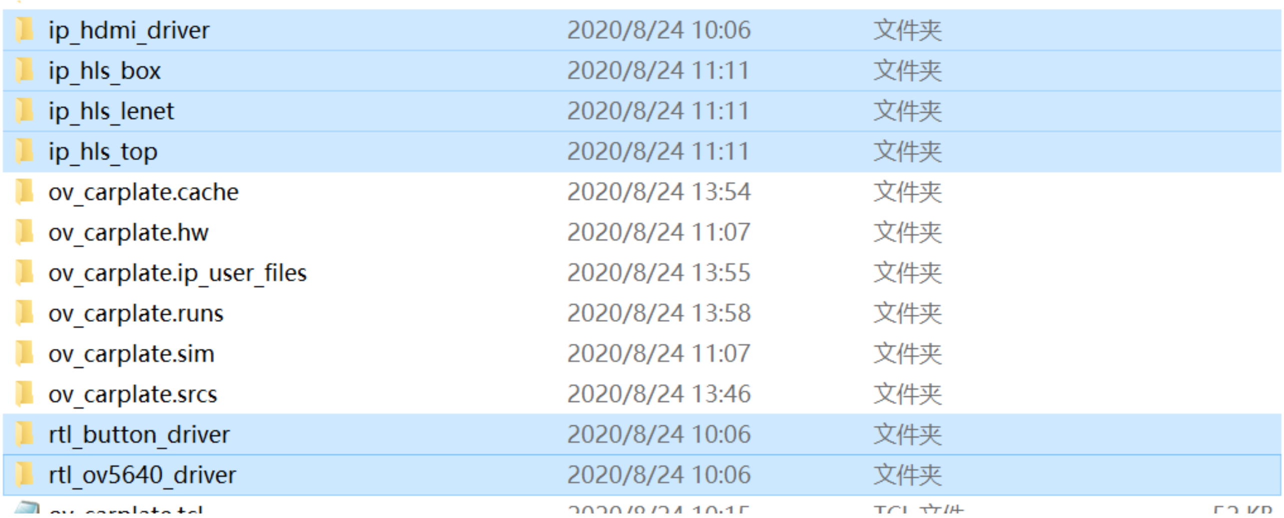
```
`timescale 1ns / 1ps
module OV5640_DRIVE(
    input cmos_clk_i, //cmos senseor clock.
    input rst_n_i, //system reset.active low.
    input cmos_pclk_i, //input pixel clock.
    input cmos_href_i, //input pixel hs signal.
    input cmos_vsync_i, //input pixel vs signal.
    input [7:0]cmos_data_i, //data.
    output                cam_rst_n    ,
    output                cam_pwdn     ,
    output cmos_xclk_o, //output clock to cmos sensor.
    output [23:0] rgb_o,
    output clk_ce,
    output de_o,
    output vs_o,
    output hs_o
    );

assign cmos_xclk_o = cmos_clk_i;
parameter[3:0]CMOS_FRAME_WAITCNT = 4'd15;
reg[4:0] rst_n_reg = 5'd0;
reg cmos_href_r;
```

2 时序仿真



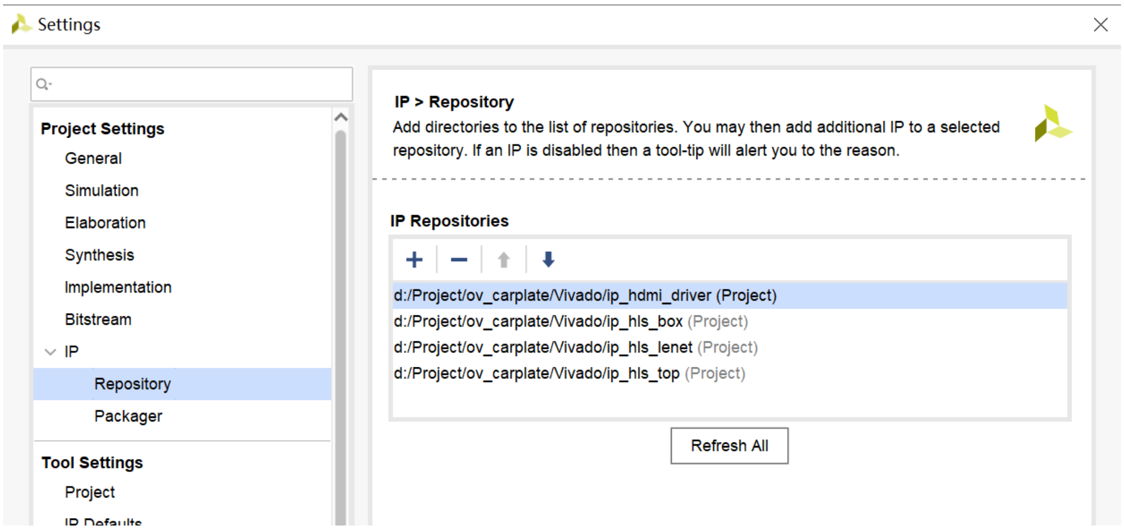
步骤 5 Vivado Block Design 设计

1 汇集 RTL 模块与 IP 核



ip_hdmi_driver	2020/8/24 10:06	文件夹	
ip_hls_box	2020/8/24 11:11	文件夹	
ip_hls_lenet	2020/8/24 11:11	文件夹	
ip_hls_top	2020/8/24 11:11	文件夹	
ov_carplate.cache	2020/8/24 13:54	文件夹	
ov_carplate.hw	2020/8/24 11:07	文件夹	
ov_carplate.ip_user_files	2020/8/24 13:55	文件夹	
ov_carplate.runs	2020/8/24 13:58	文件夹	
ov_carplate.sim	2020/8/24 11:07	文件夹	
ov_carplate.srcs	2020/8/24 13:46	文件夹	
rtl_button_driver	2020/8/24 10:06	文件夹	
rtl_ov5640_driver	2020/8/24 10:06	文件夹	
ov_carplate.tcl	2020/8/24 10:15	TCL 文件	53 KB

2 添加 IP 目录



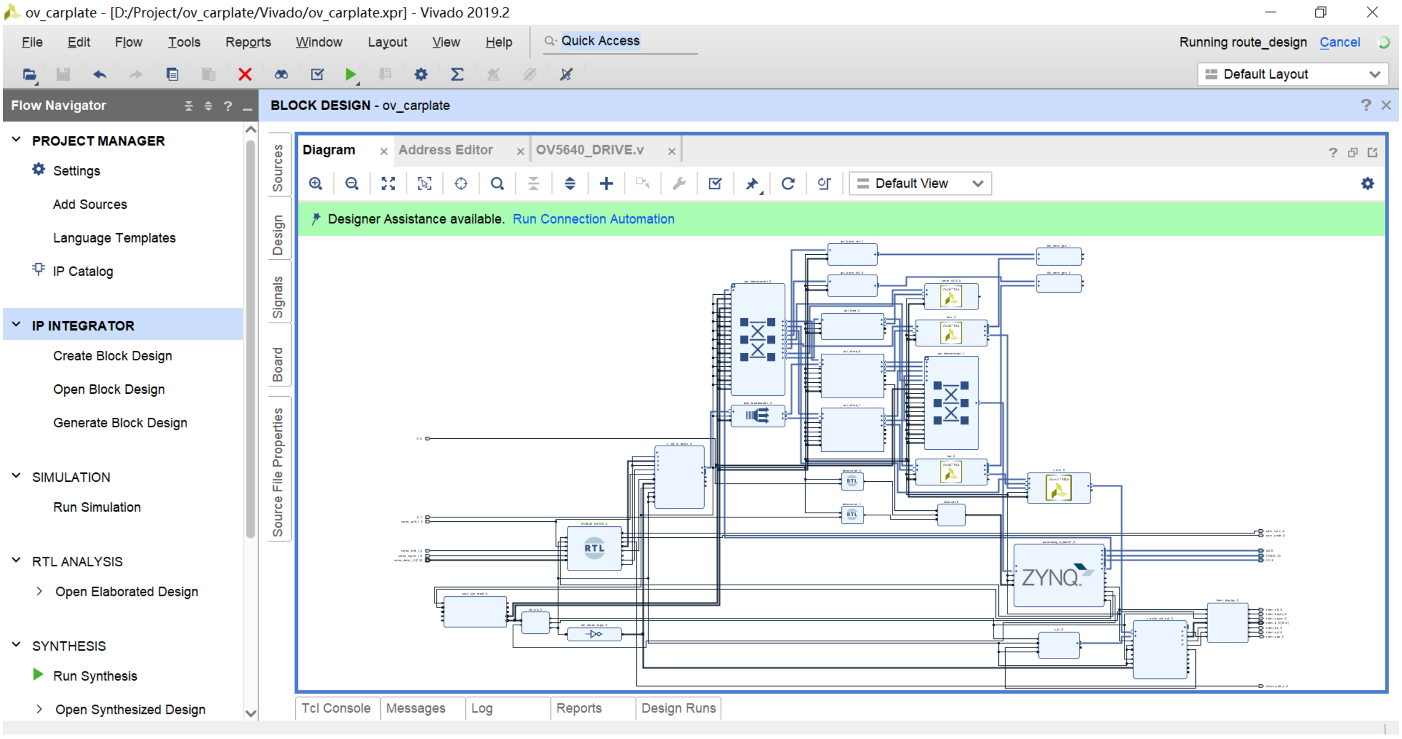
IP > Repository

Add directories to the list of repositories. You may then add additional IP to a selected repository. If an IP is disabled then a tool-tip will alert you to the reason.

IP Repositories

d:/Project/ov_carplate/Vivado/ip_hdmi_driver (Project)
d:/Project/ov_carplate/Vivado/ip_hls_box (Project)
d:/Project/ov_carplate/Vivado/ip_hls_lenet (Project)
d:/Project/ov_carplate/Vivado/ip_hls_top (Project)

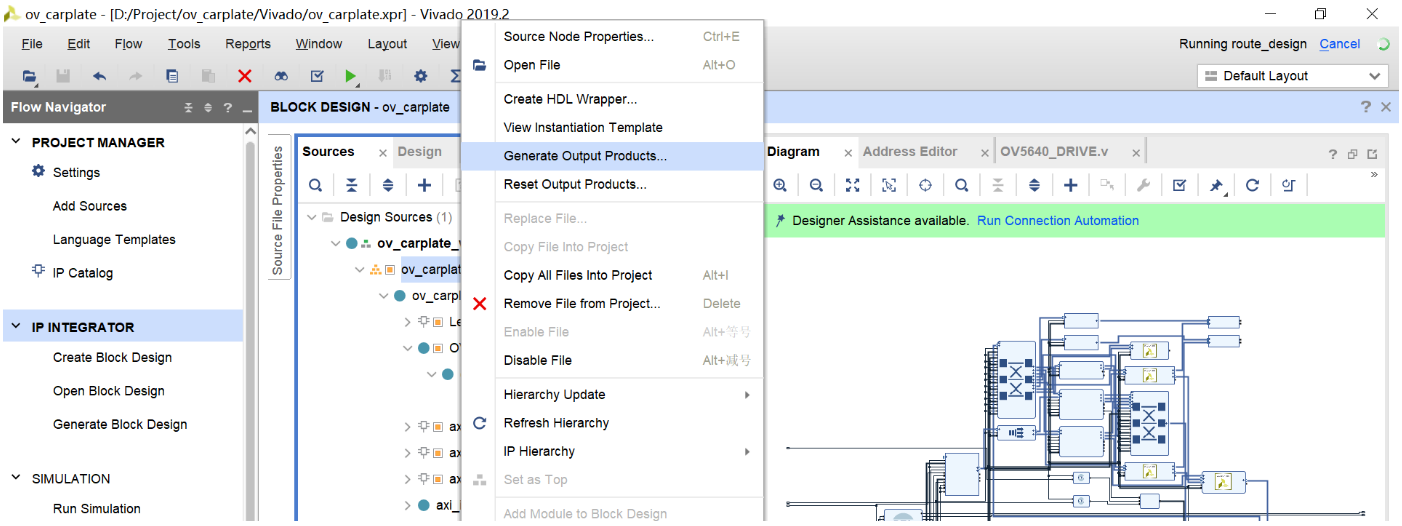
3 构建 Block Design



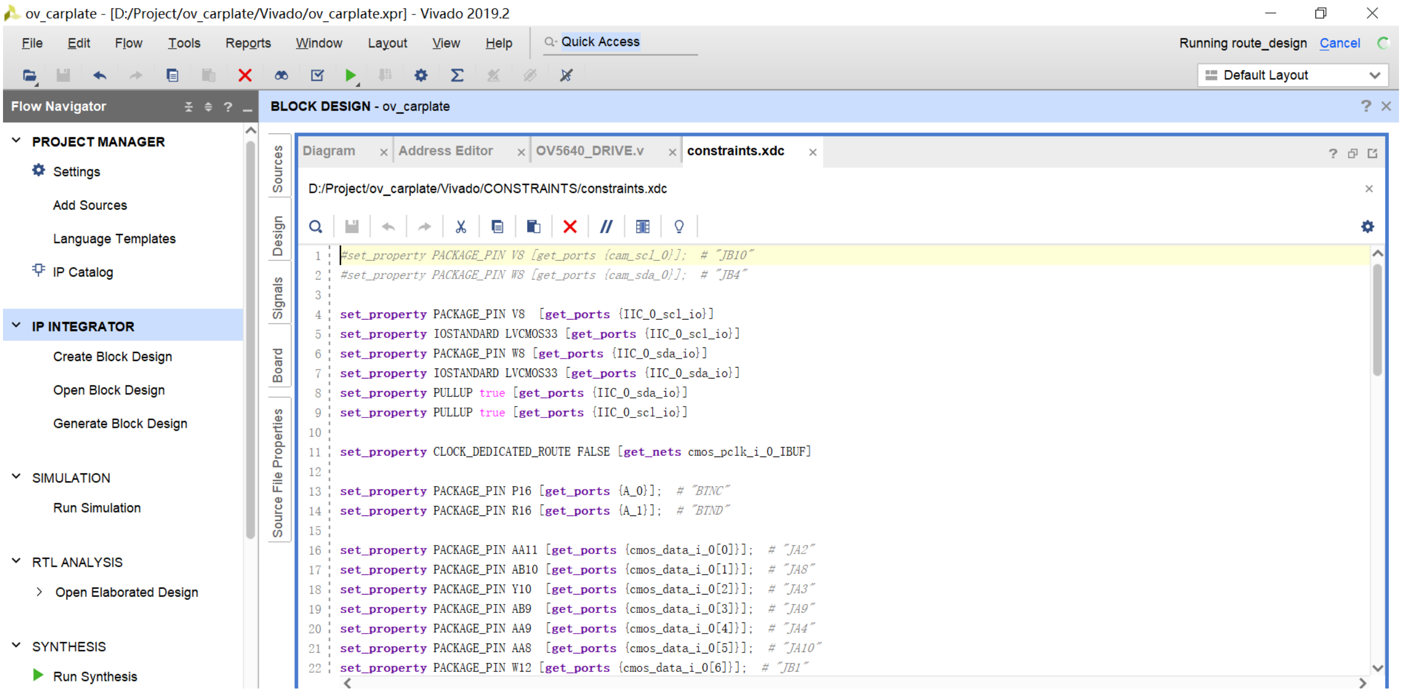
ov_carplate - [D:/Project/ov_carplate/Vivado/ov_carplate.xpr] - Vivado 2019.2

Designer Assistance available. Run Connection Automation

4 Generate Output Product，Create HDL Wrapper



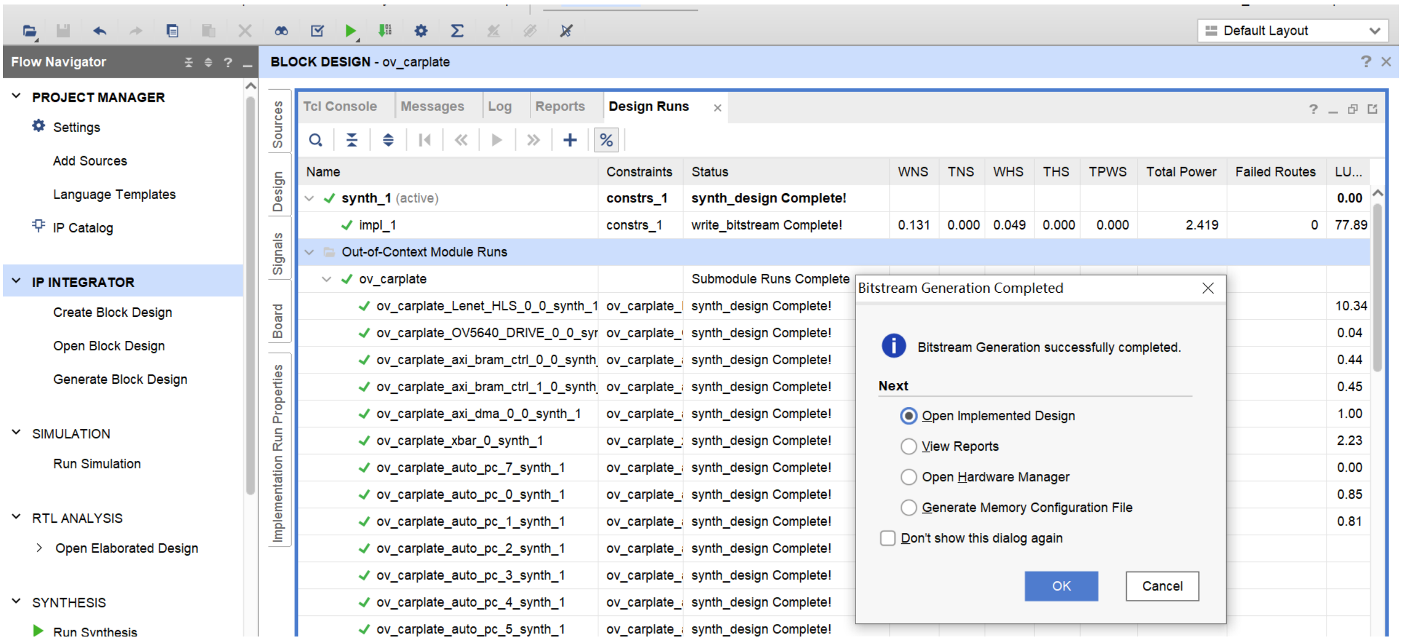
5 编写约束文件



D:/Project/ov_carplate/Vivado/CONSTRAINTS/constraints.xdc

```
#set_property PACKAGE_PIN V8 [get_ports {cam_scl_0}];  # "JB10"
#set_property PACKAGE_PIN W8 [get_ports {cam_sda_0}];  # "JB4"

set_property PACKAGE_PIN V8  [get_ports {IIC_0_scl_io}]
set_property IOSTANDARD LVCMOS33 [get_ports {IIC_0_scl_io}]
set_property PACKAGE_PIN W8 [get_ports {IIC_0_sda_io}]
set_property IOSTANDARD LVCMOS33 [get_ports {IIC_0_sda_io}]
set_property PULLUP true [get_ports {IIC_0_sda_io}]
set_property PULLUP true [get_ports {IIC_0_scl_io}]

set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets cmos_pclk_i_0_IBUF]

set_property PACKAGE_PIN P16 [get_ports {A_0}];  # "BTNC"
set_property PACKAGE_PIN R16 [get_ports {A_1}];  # "BTND"

set_property PACKAGE_PIN AA11 [get_ports {cmos_data_i_0[0]}];  # "JA2"
set_property PACKAGE_PIN AB10 [get_ports {cmos_data_i_0[1]}];  # "JA8"
set_property PACKAGE_PIN Y10  [get_ports {cmos_data_i_0[2]}];  # "JA3"
set_property PACKAGE_PIN AB9  [get_ports {cmos_data_i_0[3]}];  # "JA9"
set_property PACKAGE_PIN AA9  [get_ports {cmos_data_i_0[4]}];  # "JA4"
set_property PACKAGE_PIN AA8  [get_ports {cmos_data_i_0[5]}];  # "JA10"
set_property PACKAGE_PIN W12 [get_ports {cmos_data_i_0[6]}];  # "JB1"
```

6 Synthesis，Implementation，Generate Bitstream



Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LU...
✓ synth_1 (active)	constrs_1	synth_design Complete!								0.00
✓ impl_1	constrs_1	write_bitstream Complete!	0.131	0.000	0.049	0.000	0.000	2.419	0	77.89
Out-of-Context Module Runs										
✓ ov_carplate		Submodule Runs Complete								
✓ ov_carplate_Lenet_HLS_0_0_synth_1	ov_carplate_	synth_design Complete!								10.34
✓ ov_carplate_OV5640_DRIVE_0_0_syr	ov_carplate_	synth_design Complete!								0.04
✓ ov_carplate_axi_bram_ctrl_0_0_synth	ov_carplate_	synth_design Complete!								0.44
✓ ov_carplate_axi_bram_ctrl_1_0_synth	ov_carplate_	synth_design Complete!								0.45
✓ ov_carplate_axi_dma_0_0_synth_1	ov_carplate_	synth_design Complete!								1.00
✓ ov_carplate_xbar_0_synth_1	ov_carplate_	synth_design Complete!								2.23
✓ ov_carplate_auto_pc_7_synth_1	ov_carplate_	synth_design Complete!								0.00
✓ ov_carplate_auto_pc_0_synth_1	ov_carplate_	synth_design Complete!								0.85
✓ ov_carplate_auto_pc_1_synth_1	ov_carplate_	synth_design Complete!								0.81
✓ ov_carplate_auto_pc_2_synth_1	ov_carplate_	synth_design Complete!								
✓ ov_carplate_auto_pc_3_synth_1	ov_carplate_	synth_design Complete!								
✓ ov_carplate_auto_pc_4_synth_1	ov_carplate_	synth_design Complete!								
✓ ov_carplate_auto_pc_5_synth_1	ov_carplate_	synth_design Complete!								

Bitstream Generation Completed

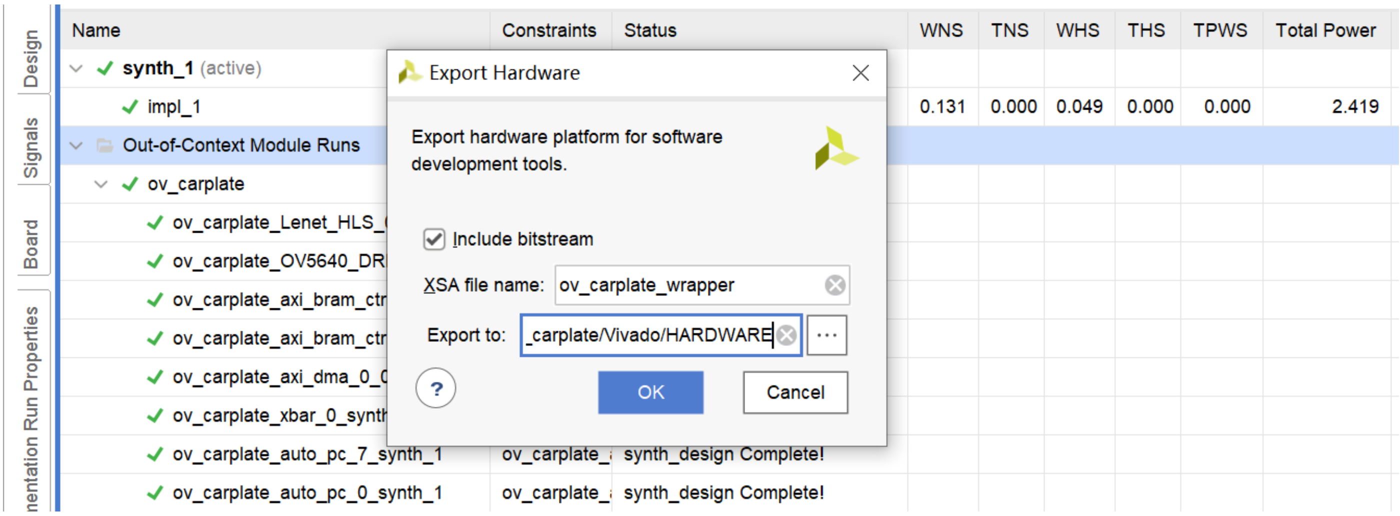
Bitstream Generation successfully completed.

Next

- ● Open Implemented Design
- ○ View Reports
- ○ Open Hardware Manager
- ○ Generate Memory Configuration File

□ Don't show this dialog again

7 Export Hardware



Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power
✓ synth_1 (active)								
✓ impl_1			0.131	0.000	0.049	0.000	0.000	2.419
Out-of-Context Module Runs								
✓ ov_carplate								
✓ ov_carplate_Lenet_HLS_(								
✓ ov_carplate_OV5640_DR								
✓ ov_carplate_axi_bram_ctr								
✓ ov_carplate_axi_bram_ctr								
✓ ov_carplate_axi_dma_0_0								
✓ ov_carplate_xbar_0_synth								
✓ ov_carplate_auto_pc_7_synth_1	ov_carplate_	synth_design Complete!						
✓ ov_carplate_auto_pc_0_synth_1	ov_carplate_	synth_design Complete!						

Export Hardware

Export hardware platform for software development tools.

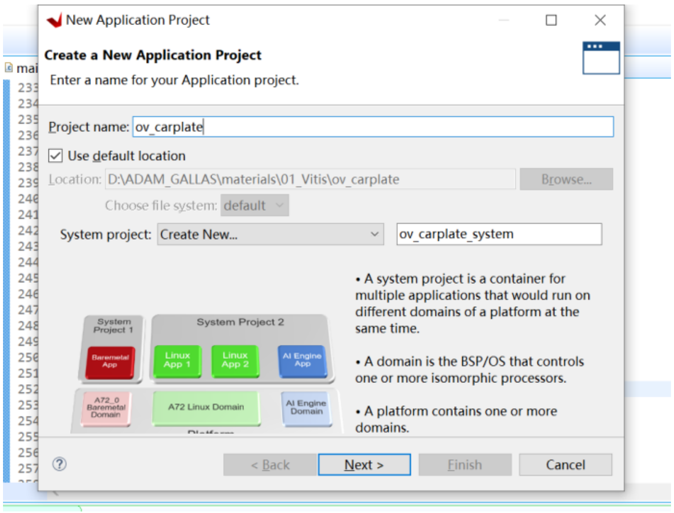
☑ Include bitstream

XSA file name: ov_carplate_wrapper

Export to: _carplate/Vivado/HARDWARE

步骤 6 使用 Vitis 进行 PS 端与 IP 配置

1 新建工程



New Application Project

Create a New Application Project

Enter a name for your Application project.

Project name: ov_carplate

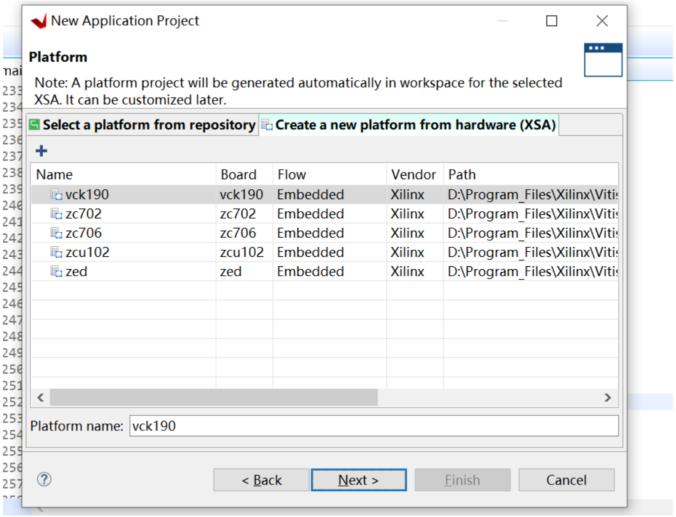
☑ Use default location

Location: D:\ADAM_GALLAS\materials\01_Vitis\ov_carplate

Choose file system: default

System project: Create New... ov_carplate_system

- A system project is a container for multiple applications that would run on different domains of a platform at the same time.
- A domain is the BSP/OS that controls one or more isomorphic processors.
- A platform contains one or more domains.



New Application Project

Platform

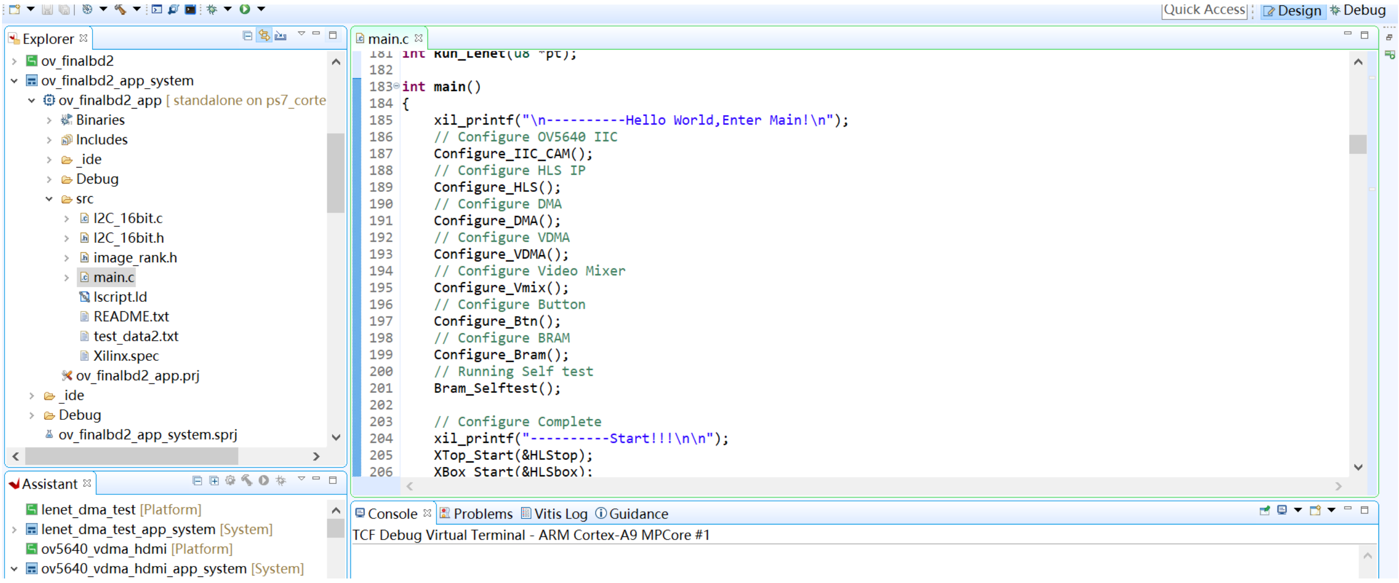
Note: A platform project will be generated automatically in workspace for the selected XSA. It can be customized later.

Select a platform from repository | Create a new platform from hardware (XSA)

Name	Board	Flow	Vendor	Path
vck190	vck190	Embedded	Xilinx	D:\Program_Files\Xilinx\Viti
zc702	zc702	Embedded	Xilinx	D:\Program_Files\Xilinx\Viti
zc706	zc706	Embedded	Xilinx	D:\Program_Files\Xilinx\Viti
zcu102	zcu102	Embedded	Xilinx	D:\Program_Files\Xilinx\Viti
zed	zed	Embedded	Xilinx	D:\Program_Files\Xilinx\Viti

Platform name: vck190

2 初始化 IP



Explorer

- ov_finalbd2
- ov_finalbd2_app_system
  - ov_finalbd2_app [standalone on ps7_corte
    - Binaries
    - Includes
    - _ide
    - Debug
    - src
      - I2C_16bit.c
      - I2C_16bit.h
      - image_rank.h
      - main.c
      - lscript.ld
      - README.txt
      - test_data2.txt
      - Xilinx.spec
    - ov_finalbd2_app.prj
  - _ide
  - Debug
  - ov_finalbd2_app_system.sprj

main.c

```
int Run_Lenet(u8 *pt);

int main()
{
    xil_printf("\n----------Hello World,Enter Main!\n");
    // Configure OV5640 IIC
    Configure_IIC_CAM();
    // Configure HLS IP
    Configure_HLS();
    // Configure DMA
    Configure_DMA();
    // Configure VDMA
    Configure_VDMA();
    // Configure Video Mixer
    Configure_Vmix();
    // Configure Button
    Configure_Btn();
    // Configure BRAM
    Configure_Bram();
    // Running Self test
    Bram_Selftest();

    // Configure Complete
    xil_printf("----------Start!!!\n\n");
    XTop_Start(&HLStop);
    XBox_Start(&HLSbox);
```

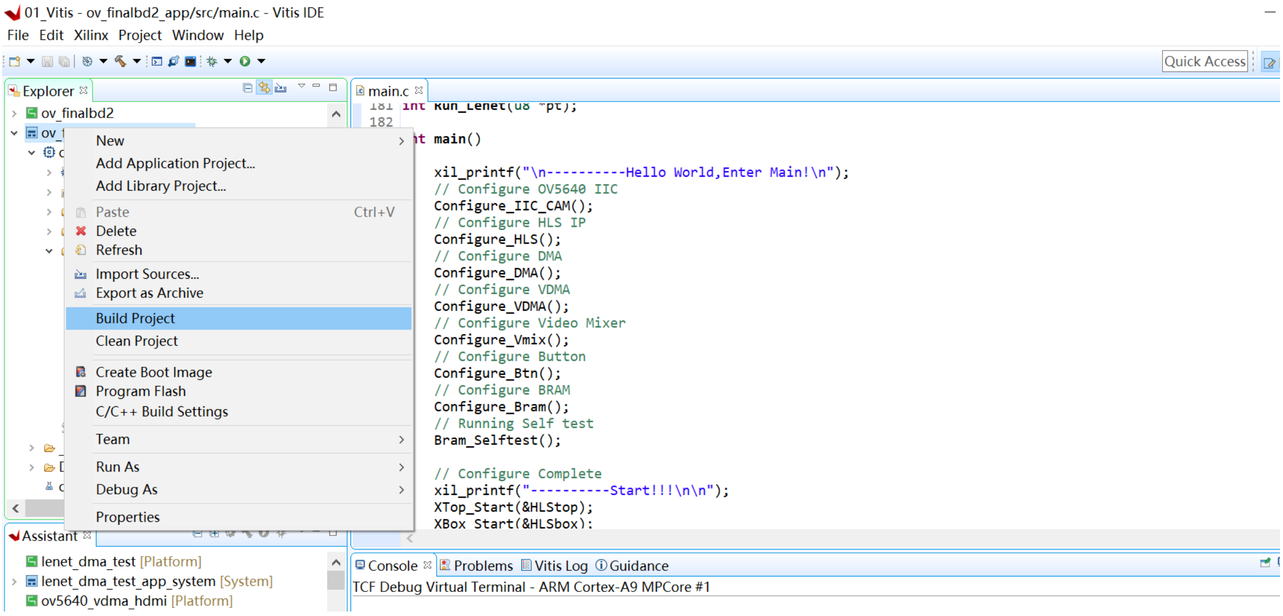
Assistant

- lenet_dma_test [Platform]
- lenet_dma_test_app_system [System]
- ov5640_vdma_hdmi [Platform]
- ov5640_vdma_hdmi_app_system [System]

Console | Problems | Vitis Log | Guidance

TCF Debug Virtual Terminal - ARM Cortex-A9 MPCore #1

3 Build Project，进行初步测试

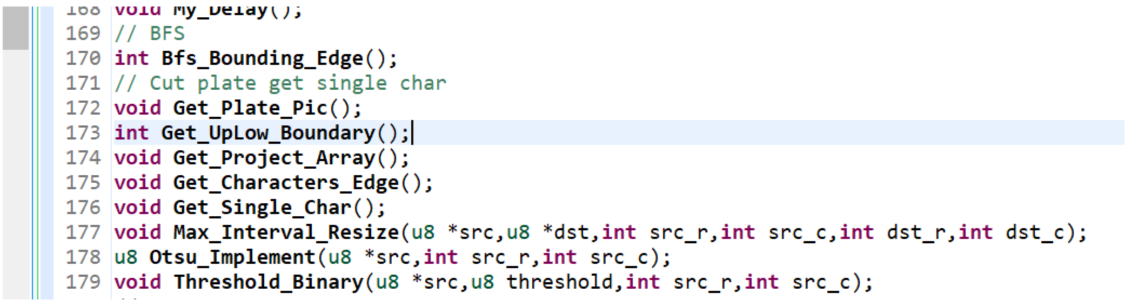


```
int Run_Lenet(u8 *pt);

t main()

    xil_printf("\n---------Hello World,Enter Main!\n");
    // Configure OV5640 IIC
    Configure_IIC_CAM();
    // Configure HLS IP
    Configure_HLS();
    // Configure DMA
    Configure_DMA();
    // Configure VDMA
    Configure_VDMA();
    // Configure Video Mixer
    Configure_Vmix();
    // Configure Button
    Configure_Btn();
    // Configure BRAM
    Configure_Bram();
    // Running Self test
    Bram_Selftest();

    // Configure Complete
    xil_printf("---------Start!!!\n\n");
    XTop_Start(&HLStop);
    XBox_Start(&HLSbox);
```

TCF Debug Virtual Terminal - ARM Cortex-A9 MPCore #1

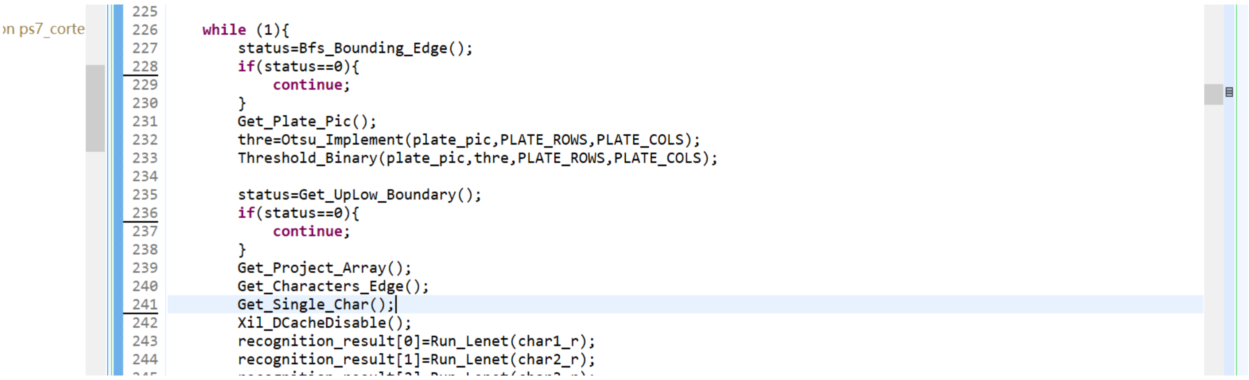
步骤 7 PS 端图像处理算法设计

1 编写子过程函数



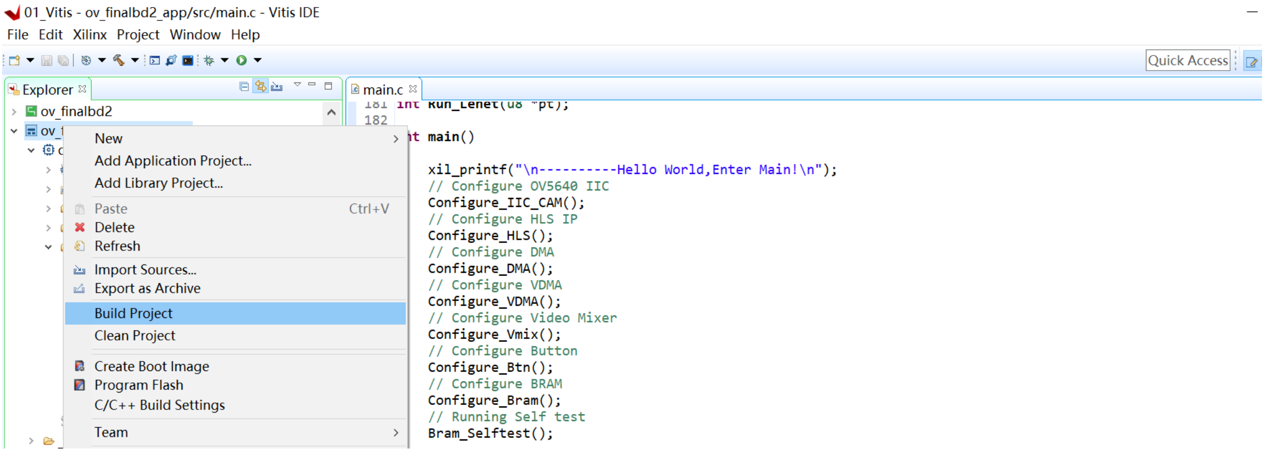
```
void My_Delay();
// BFS
int Bfs_Bounding_Edge();
// Cut plate get single char
void Get_Plate_Pic();
int Get_UpLow_Boundary();
void Get_Project_Array();
void Get_Characters_Edge();
void Get_Single_Char();
void Max_Interval_Resize(u8 *src,u8 *dst,int src_r,int src_c,int dst_r,int dst_c);
u8 Otsu_Implement(u8 *src,int src_r,int src_c);
void Threshold_Binary(u8 *src,u8 threshold,int src_r,int src_c);
```

2 进一步整合拼接



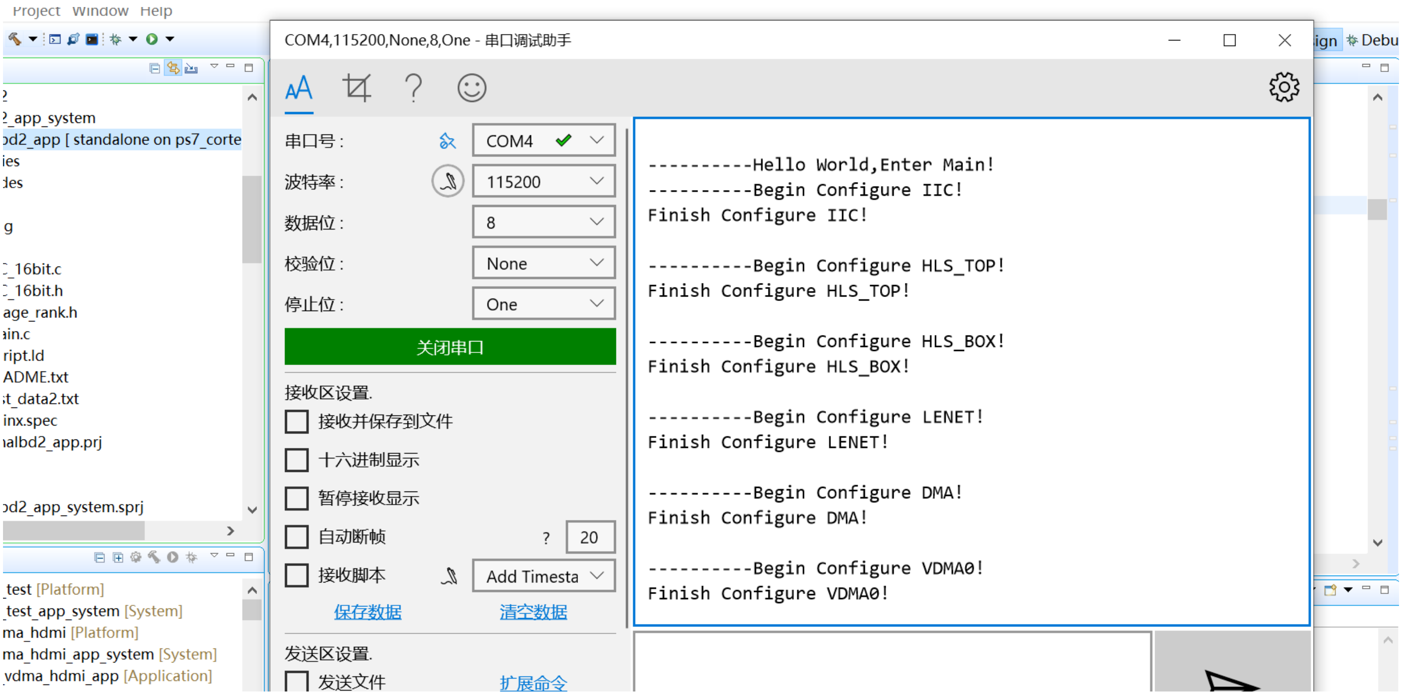
```

    while (1){
        status=Bfs_Bounding_Edge();
        if(status==0){
            continue;
        }
        Get_Plate_Pic();
        thre=Otsu_Implement(plate_pic,PLATE_ROWS,PLATE_COLS);
        Threshold_Binary(plate_pic,thre,PLATE_ROWS,PLATE_COLS);

        status=Get_UpLow_Boundary();
        if(status==0){
            continue;
        }
        Get_Project_Array();
        Get_Characters_Edge();
        Get_Single_Char();
        Xil_DCacheDisable();
        recognition_result[0]=Run_Lenet(char1_r);
        recognition_result[1]=Run_Lenet(char2_r);
```

3 Build Project



```
int Run_Lenet(u8 *pt);

t main()

    xil_printf("\n---------Hello World,Enter Main!\n");
    // Configure OV5640 IIC
    Configure_IIC_CAM();
    // Configure HLS IP
    Configure_HLS();
    // Configure DMA
    Configure_DMA();
    // Configure VDMA
    Configure_VDMA();
    // Configure Video Mixer
    Configure_Vmix();
    // Configure Button
    Configure_Btn();
    // Configure BRAM
    Configure_Bram();
    // Running Self test
    Bram_Selftest();
```

4 连接串口，上板验证测试



```
----------Hello World,Enter Main!
----------Begin Configure IIC!
Finish Configure IIC!

----------Begin Configure HLS_TOP!
Finish Configure HLS_TOP!

----------Begin Configure HLS_BOX!
Finish Configure HLS_BOX!

----------Begin Configure LENET!
Finish Configure LENET!

----------Begin Configure DMA!
Finish Configure DMA!

----------Begin Configure VDMA0!
Finish Configure VDMA0!
```