

双目视觉实时处理系统

简介

本说明旨在指导如何构建一个双目视觉实时处理系统。

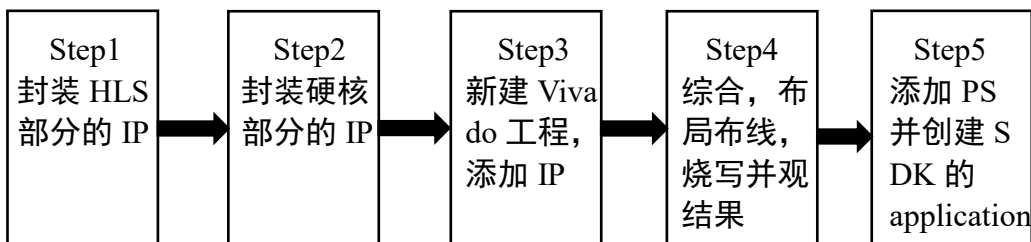
目标

- 利用 HLS 封装算法部分 IP
- 利用 Vivado 实现摄像头配置，VGA 等模块并构建数据通路
- 利用 SDK 实现 PS-PL 通信

过程

根据此说明中提供的步骤构建处理系统，主要包括以下 5 个步骤：封装 HLS 部分的 IP，封装硬核部分 IP，新建 vivado 工程添加上面封装的 IP，烧录并观察结果，配置 PS 和 SDK application

总体流程



封装 HLS 部分的 IP

Step1

- 1-1. 创建一个新的 HLS 工程，目标器件为 xc7z010clg400-1
 - 1-1-1. 创建一个新的工程命名为 pre.prj
 - 1-1-2. 将工程源码中的 pre.cpp 和 pre.h 添加到工程中，不需要添加 testbench
 - 1-1-3. 在 solution configuration 中将时钟频率设置为 8
 - 1-1-4. 在 device selection dialog 界面选择 zynq xc7z010clg400-1
 - 1-1-5. 点击 finish 完成创建
- 1-2. C 综合
 - 1-2-1. 选择 solution > Run C synthesis > active solution
 - 1-2-2. 综合结束后可以展开 explorer 中的 solution1 来查看综合报告
- 1-3. 导出 RTL 并封装 IP
 - 1-3-1. 选择 solution > export RTL 选择 VHDL，点击 OK
 - 1-3-2. 在 explorer 展开 solution1 > impl > ip，可以看到 IP 被封装到了一个 zip 压缩包中，至此，可以在以后的工程中添加此 IP 了

- 1-4. 再创建一个新的 HLS 工程，目标器件为 xc7z010clg400-1
 - 1-4-1. 创建一个新的 HLS 工程命名为 stereoBM.prj
 - 1-4-2. 将工程源码中的 stereoBM.cpp 和 stereo.h 添加到工程中，在添加 testbench 时添加 stereo_test.cpp
 - 1-4-3. 在 solution configuration 中将时钟频率设置为 8
 - 1-4-4. 在 device selection dialog 界面选择 zynq xc7z010clg400-1
 - 1-4-5. 点击 finish 完成创建
- 1-5. C 仿真
 - 1-5-1. 选择 project > Run C simulation 进行 C 仿真
- 1-6. C 综合
 - 1-6-1. 选择 solution > Run C synthesis > active solution
 - 1-6-2. 综合结束后可以展开 explorer 中的 solution1 来查看综合报告
- 1-7. C/RTL Co-Simulation
 - 1-7-1. 选择 solution > Run C/RTL Co-Simulation，选择 VHDL
 - 1-7-2. 点击 OK，完成后可以观察 Cosimulation 报告
- 1-8. 导出 RTL 并封装 IP
 - 1-8-1. 选择 solution > export RTL 选择 VHDL，点击 OK
 - 1-8-2. 在 explorer 展开 solution1 > impl > ip，可以看到 IP 被封装到了一个 zip 压缩包中，至此，可以在以后的工程中添加此 IP 了

封装硬核部分 IP

Step2

- 2-1. 打开 Vivado2016.4，新建一个工程并命名为 buzzer
- 2-2. 添加源文件
 - 2-2-1. 将硬核部分源码中的 buzzer.v 文件添加到 source 中
- 2-3. 封装 IP
 - 2-3-1. 将整个工程导出为 IP
- 2-4. 打开 Vivado2016.4 > IP manager，按照上面的流程分别创建 camera_capture_320, C MOS_Capture, HD_div_4_with_clk_sync, i2c_com, power_on_delay, reg_config, system_ctrl, vga_out, VGA_out_choose, x 这 10 个 IP

新建 Vivado 工程，添加 IP

Step3

- 3-1. 打开 Vivado2016.4 新建一个工程，命名为 camera_zybo
- 3-2. 在 Project Settings 中的 IP 中添加上述所有 IP 的路径
- 3-2. 打开 block design 添加上面写好的所有 IP，并且按照压缩包中的 design.pdf 连线

综合，布局布线，烧写并观察结果

Step4

- 4-1. 点击 Flow Navigator > Program and Debug > Generate Bitstream，如果有对话框弹出询问是否保存，选择保存，忽略警告
- 4-2. 点击 Flow Navigator > Program and Debug > Open Hardware Manager，连接板卡并烧录
- 4-3. 连接显示器，观察实验结果，下面两幅图是左右摄像头拍摄到的图像，上面是视差图

添加 PS 模块并创建 SDK application

Step5

5-1. 创建 select_3 IP

5-1-1. 打开 Vivado2016.4, 打开 IP manager, 创建一个新的 select_3 IP

5-1-2. 点击 Tools > Create and Package IP, 选择 AXI IP 并编辑

5-1-3. 创建完成后在 source 中打开 select3_v1_0.v, 将里面的内容全部用软核部分中的源文件 select3_v1_0.v 代替, 将 select_3_v1_0_S00_AXI.v 中的内容也用软核部分源文件 select_3_v1_0_S00_AXI.v 的内容代替

5-1-4. 点击 Flow Navigator > Package IP, 合并更改, 点击 Package 进行封装

5-2. 向上面的工程中添加 select_3 IP 以及 PS

5-2-1. 打开上面的工程, 在 Project Settings 中将刚刚封装好的 IP 的路径添加到 IP repository 中

5-2-2. 打开 block design, 添加 select_3 IP 以及 zynq PS

5-2-3. 根据实验指导书中指导配置 PS

5-2-4. 在 block design 顶部出现 run connection automation, 点击, 选择 select_3 进行自动连线

5-3. 在 SDK 中创建 application

5-3-1. 点击 Generate Bitstream

5-3-2. 生成 bit 文件成功后点击 File > Export > Export Hardware, 选择 include bitstream

5-3-3. 点击 File > Launch SDK

5-3-4. 新建空白 application, 在 src 中添加软核部分源代码中的 display_distance.c

5-3-5. 编译通过后 program FPGA 并 run as debugger

5-3-6. 打开 terminal 观察结果