



FPGA实现图像拼接学习路线

学习要点提醒

- 开发流程规范
 - 先在Matlab或OpenCV中验证算法逻辑
 - 再进行RTL设计并在FPGA上部署
 - 遵循仿真→综合→实现→下载完整流程
- 资源与性能平衡
 - 根据硬件瓶颈计算数据吞吐量
 - 在资源占用与处理速度之间寻找最佳平衡点
- 代码设计规范
 - 信号传递方向、功能命名需要通俗易懂
 - 保证各模块功能独立
 - 顶层模块只调用子模块
- 参考
 - <https://bbs.huaweicloud.com/blogs/285121>
 - <https://zhuanlan.zhihu.com/p/697872687>
- 实施建议
 - 先简后繁，如先从两幅图像拼接入手，逐步扩展至多图场景
 - 优先确保功能正确，再优化实时性和资源效率

推荐工具与资源

- 一定要多用仿真!
 - 状态机绘制: <https://tool.chipdebug.com/fsm/>
 - ModelSim用于功能仿真验证
- 波形绘制: timegen3.2
- verilog语法: <https://www.runoob.com/w3cnote/verilog-tutorial.html>
- 框图绘制: draw.io
- 善用ai工具

拓展学习

- 附件论文
- OpenCV官网

实战项目阶段

- 模块化开发
 - 将系统划分为图像采集、预处理、特征匹配和融合输出、显示模块
 - 定义统一的数据接口标准
 - 使用AXI-Stream协议封装图像数据流
- 参考案例复现
 - 借助matlab OpenCV验证拼接算法正确性
 - FPGA图像拼接开源项目学习
- 框架设计搭建
 - 先完成摄像头-ddr-显示的数据流整体框架
 - 在框架上添加功能
 - tb随时仿真验证

基础准备阶段

- 数字电路与FPGA基础
 - 学习组合逻辑和时序电路的基本原理
 - 理解FPGA内部结构如LUT、BRAM和DSP单元
 - 准备开发环境: 1 在高云或sipeed官网上下载高云ide; 2 下载modelsim (推荐2020.4)
- 硬件描述语言 (HDL)
 - 掌握Verilog HDL或VHDL的基础语法结构 (可在HDLbits刷题)
 - 重点实践状态机设计方法
- 图像处理基础
 - 了解常见图像格式包括RGB、YUV和灰度图
 - 理解图像分辨率与像素排列方式
 - 参考学习《基于MATLAB与FPGA的图像处理教程》作者: 韩彬

核心技能进阶

- FPGA图像处理框架
 - 板卡例程学习 <https://github.com/sipeed/TangMega-138KPro-example>
 - 掌握视频和摄像头接口协议如HDMI、DVP
 - 学习摄像头寄存器参数及配置
 - 构建基于像素级处理的流水线数据流架构
 - 设计帧缓存系统并集成DDR控制器
- 图像拼接关键算法
 - 适用于FPGA的特征点提取算法如ORB
 - 透射变换原理和双线性插值
 - 图像融合策略如加权或多频段融合
- 设计技巧
 - tb测试文件编写
 - 掌握硬件计算技巧如采用定点数替代浮点数、移位操作等
 - 流水线设计
 - 深入理解时序, 根据时序报告优化时序
 - 片上debug分析关键信号