**2020年新工科联盟-Xilinx暑期学校团队项目设计文档**

**设计文稿提交格式**

**(Project Paper Submission Template)**

|  |  |
| --- | --- |
| **作品名称** | JPEG图像压缩 |
| **板卡型号** | ZedBoard |
| **所在班级** | B6 |
| **成员姓名、学号、学校** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | 组长 | 任静 | 西南科技大学 | 信息与通信工程 | 20BO186B | | 成员 | 李陈 | 西南科技大学 | 信息与通信工程 | 20BO186A | | 刘通 | 西南科技大学 | 控制科学与工程 | 20BO187C | | 李宇 | 西南科技大学 | 电子信息 | 20BO181A | |
| **Github链接** | *git地址：https://github.com/nili-hub/JPEG\_compression.git* |

**第一部分**

设计概述 /Design Introduction

（1.请概括地描述一下你的设计，可包括本设计目的、学习到的知识点、应用方向或者设想的应用场景等；2. 经组内成员讨论后以表格的形式描述项目中各成员在项目中发挥的作用或者贡献百分比；3.作品的展示照片）

随着图像质量及其对处理实时性要求的不断提高,软件方法实现图像压缩越来越难以满足实际需要,FPGA固有的并行处理能力及其在高速计算方面的优势使得它在图像压缩中的应用越来越广泛, 本项目设计为一个使用FPGA对JPEG图像进行压缩，用户可自行灵活定义压缩比。

|  |  |
| --- | --- |
| **成员** | **成员贡献** |
| 任静 | 负责PL部分HLS算法部分的资料搜集、算法仿真及优化，编写项目设计文档； |
| 李陈 | 主要负责PS与PL之间的图像数据通信，参与项目架构设计； |
| 刘通 | 负责算法部分的资料搜集、算法仿真、优化以及MATLAB部分的仿真； |
| 李宇 | 参与负责算法资料搜集；算法MATLAB仿真； |

**第二部分**

系统组成及功能说明 /System Construction & Function Description

（请对作品的1. 计划实现及已实现的功能；2. 项目系统框图；3. 使用的技术方向做说明）

本项目计划实现的功能有以下几点：

1.完成原始图像数据从PS到PL的传输；

2.完成JPEG图像的色彩空间转换；

3.完成JPEG图像的二维DCT变换；

4.完成JPEG图像的量化压缩；

5.完成压缩图像PL到PS的传输；

使用ZedBoard板卡搭建硬件平台；使用VIVADO HLS软件编写压缩算法并生成硬件IP；ZYNQ采用BRAM缓存图像数据；使用VIVADO SDK软件将图像数据通过AXI总线传输至PL部分进行压缩算法部分的处理。PL部分处理完成之后将压缩完成的图像数据传回PS部分。



**项目系统框图**

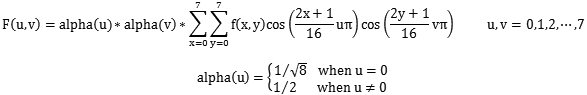
**算法原理简介：**

(1)颜色空间转换

JPEG图像使用的颜色空间是RGB颜色空间，在压缩算法中对RGB三个分量分别做压缩处理。对于压缩，首先做的事情就是把重要的信息和不重要的信息分开，YcbCr模型恰好能做到这一点。对于人眼来说，图像中明暗的变化更容易被感知到，这是由于人眼的构造引起的。视网膜上有两种感光细胞，能够感知亮度变化的视杆细胞，以及能够感知颜色的视锥细胞，由于视杆细胞在数量上远大于视锥细胞，所以我们更容易感知到明暗细节。因此需将RGB颜色空间转为YcbCr颜色空间。其转换公式如下：

(2)离散余弦变换

离散余弦变换DCT（Discrete Cosine Transform）是数码率压缩需要常用的一个变换编码方法。任何连续的实对称函数的付立叶变换中只含余弦项，因此余弦变换与付立叶变换一样有明确的物理意义。DCT是先将整体图像分成N\*N像素块，然后对N\*N像素块逐一进行DCT变换。由于大多数图像的高频分量较小，相应于图像高频分量的系数经常为零，加上人眼对高频成分的失真不太敏感，所以可用更粗的量化。因此，传送变换系数的数码率要大大小于传送图像像素所用的数码率。到达接收端后通过反离散余弦变换回到样值，虽然会有一定的失真，但人眼是可以接受的。二维余弦变换的公式如下：



经过余弦变换之后，左上角的直流分量保留了一个很大的数值，其他分量都接近于0，因此为后面进一步的压缩起到了充分的铺垫作用。

(3)数据量化

经过上一节介绍的离散余弦变换，图像数据虽然已经面目全非，但仍然是处于“可逆”的状态，也就是说我们还没有进入“有损”的那一步。总所周知，在数字系统中，量化是误差的主要来源，也正是量化，图像压缩的数据量得以减小。JPEG算法提供了两张标准量化系数矩阵，分别用于处理亮度数据Y和色差数据Cr以及Cb。经过量化之后，一大部分数据变为0，使得数据量大大减小。

在实际的算法实现中，图像经过颜色空间转换RGB-YCrCb、2-DCT变换、量化之后，需要再将图像数据复原。因此需要再进行2-IDCT变换，颜色空间转换YCrCb-RGB，使得图像能够复原。

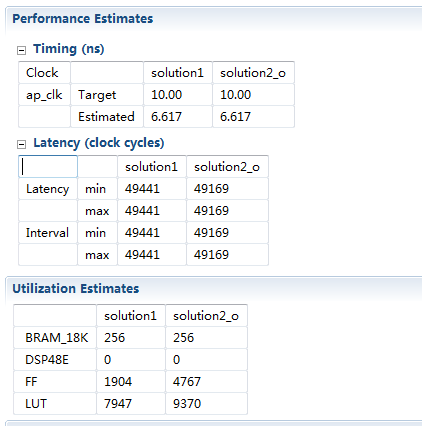
**第三部分**

完成情况及性能参数 /Final Design & Performance Parameters

（作品已实现的功能及性能指标）

**（1）优化过程**

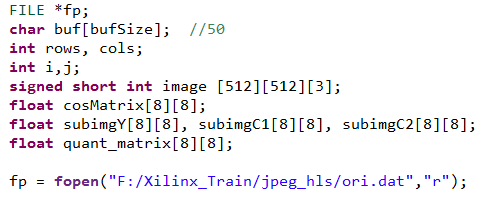
在HLS综合中，可以看到下图所示的综合报告，本项目采用的时钟为100MHZ。本项目应用了多种优化策略，包括PIPELINE、UNROLL、DATAFLOW等指令。solution2\_o是基于solution1使用PIPELINE指令优化后的综合报告。可以看出时延和初始间隔均有下降，而资源仍然充足。



**图3-1 HLS综合报告**

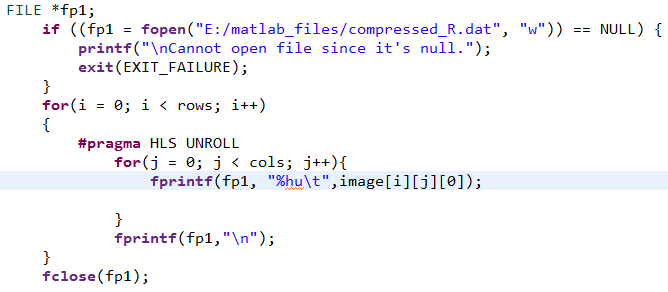
**（2）HLS仿真过程**

仿真图像源采用500\*303的JPEG图像，由PC写进ori.dat文件，下图3-2是HLS仿真文件读取图像数据的代码。由软件补足边界。



**图3-2 HLS仿真中读取图像数据**

图3-3是将图像数据存储的HLS代码贴图，将压缩后的图像数据分RGB三通道存储至文件。



**图3-3 HLS图像数据存储贴图**

图3-4是MATLAB将图像存储为.dat文件以及对HLS存储的图像数据进行复现。

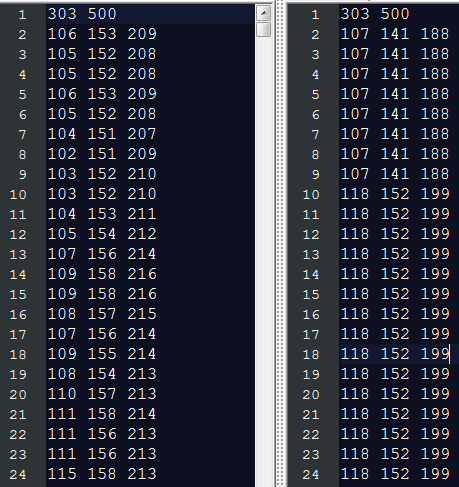
|  |  |
| --- | --- |
|  |  |
| **（a）写图像.dat文件** | **（b）图像复现** |
| **图3-4 MATLAB仿真贴图** | |

图3-5是将HLS仿真中的输出数据使用MATLAB画出的对比效果。在视觉感受上，可以看出压缩后的图像的高频分量被平滑，整张图显得不那么锐利；**在图像大小上，原图大小为49KB，压缩后的图像大小为10KB，压缩比达到1/5；**

|  |  |
| --- | --- |
|  |  |
| **(a)原始图像** | **(b)压缩图像** |
|  | |
| **(c)图像大小对比** | |

**图3-5 HLS仿真结果**

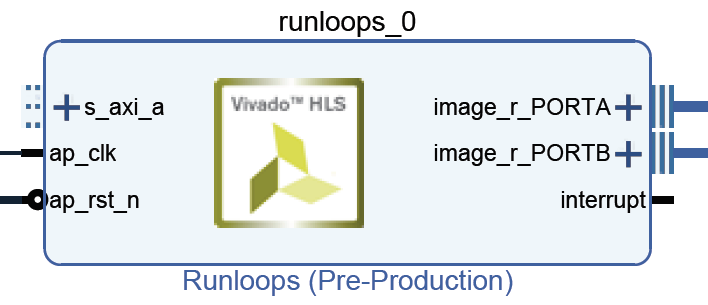
图3-6展示的是压缩前后部分图像数据的对比。能够发现图像数据均有所改变。



**图3-6 压缩前后图像数据对比**

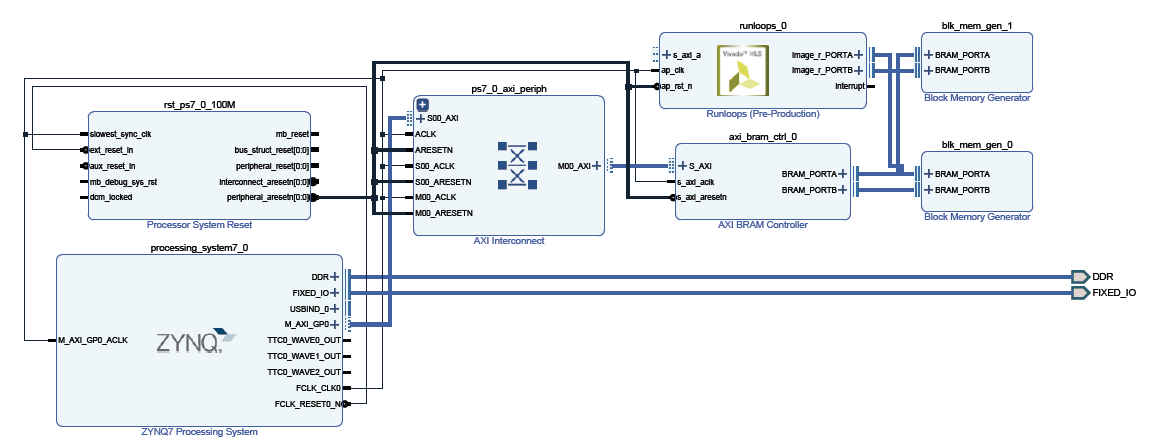
**（3） IP搭建过程**

在我们项目的设计中， PS与PL之间的通信采用BRAM缓存数据。图3-7是HLS算法IP，IP接口我们使用的是BRAM，这样能够和前级BRAM接口对应。图3-8是VIVADO中搭建的通信链路。包括有ZYNQ7处理器、处理器复位器、AXI Interconnect、AXI BRAM控制器、BRAM。



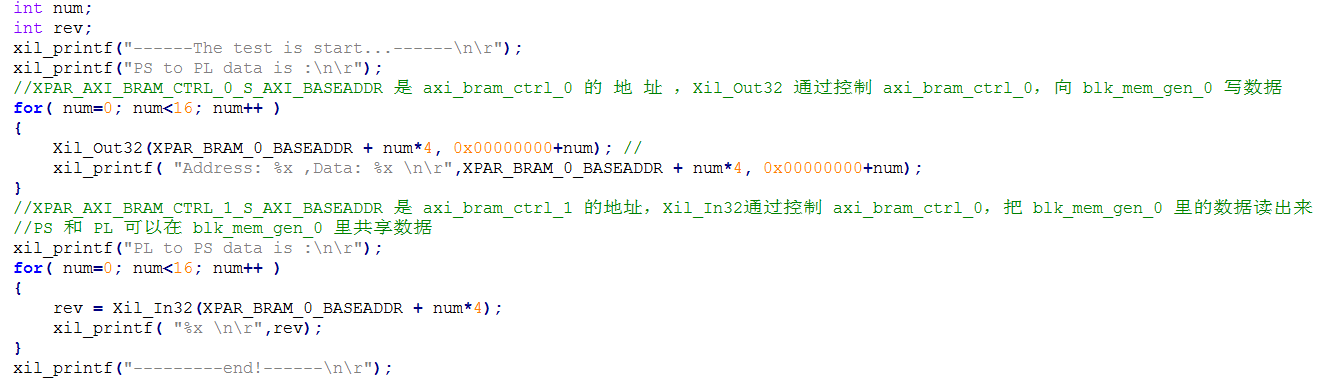
**图3-7 HLS算法IP**

图像数据从ZYNQ处理器送出、经AXI Interconnect、BRAM控制器进入BRAM，经Runloops核处理后由同样的路径送回PS。

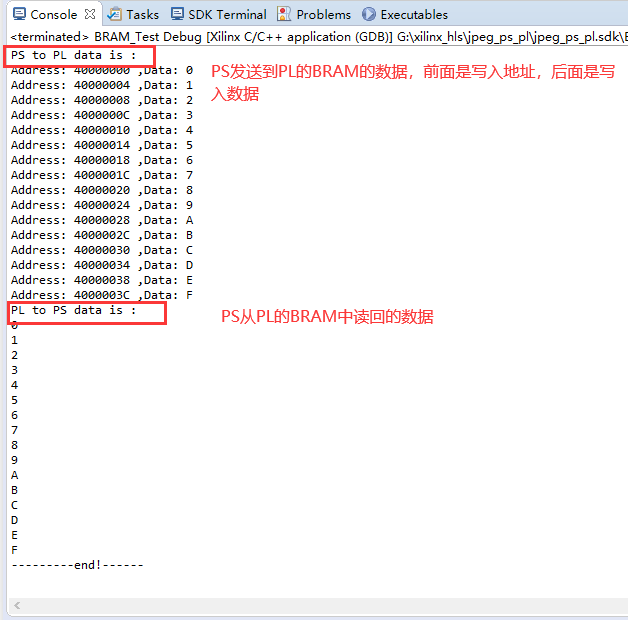


**图3-8 IP链路**

PS部分的开发由VIVADO SDK进行。图3-9是SDK中的数据传输控制。图3-10是SDK的数据测试截图。



**图3-9 SDK数据传输控制**



**图3-9 SDK数据传输测试**

**第四部分**

总结 /Conclusions

（谈一谈完成暑期学校课程后的收获与感想。请每位组员分开写。）

**姓名：任静 学号：20BO186B**

为期十余天的XILINX数据课程落下了帷幕，回想起来收益颇多。说起来我接触FPGA才一年多，这一年多的时间里接触过不少项目，但都是用VIVADO HDL工具进行的。HLS工具虽早有耳闻，但一直没有接触过，直到这次的暑假课程。

通过这次的暑假课程，我了解了VIVADO HLS工具的使用流程，掌握了HLS中对项目的各种优化指令。如PIPELINE、DATAFLOW、INLINE、UNROLL等指令。之前我一直想不通的是软件中的数据是怎么流向的。因为在硬件中，有时钟的驱动使得数据能够按照时序进入模块进行处理。那么在软件中对于一个一个的函数，数据是怎样一下就进到数据里的呢。后来我才想明白，原来数据在进入函数之前会有一个BUF用以缓存数据，使得数据能够与时钟对齐。

当然在这次的项目中我们也遇到了很多的问题。比如如何实现图像数据传送到PL部分。起初我们想的是将图像数据存到SD卡中，经过PL压缩处理之后再将图像数据写回SD卡。这样似乎是可行的，但是问题马上就出现了。使用操作系统将图像数据写进SD卡的时候，JPEG图像数据会按照JPEG图像的存储格式进行存储。意思是SD卡的存储地址第一个数据并不是像素数据，而是用以描述JPEG图像的各种文件头。这样图像数据从SD卡中读出来就会很麻烦。即使读出来的话，后续写进去也要按照JPEG的存储格式进行写操作。这样就增加了很多工作量。因此我们想到使用PS传输图像数据。同样存在了一些问题。PS与PL之间进行数据传输有多种方法，可以通过FIFO，也可以通过BRAM。经过测试我们发现使用BRAM会使数据传输更加灵活。

这次项目的圆满完成离不开我队友的辛苦努力，大家从确立题目之后就开始进入紧张的开发过程。当然最后也要感谢屏幕上的老师们的辛苦讲解，老师们讲得非常的好。

**姓名：李陈 学号：20BO186A**

通过这次暑期学校的课程，熟悉了HLS工具的使用，了解了HLS基本操作流程，以及在HLS里面怎样优化自己的设计，学会了分析solution使用PIPELINE、ARRAY\_PARTITION、DATAFLOW、INLINE等不同优化策略对性能的影响，学会了如何在HLS中提升性能和平衡资源的各种技巧。还有对仿真结果的分析，对C仿真的分析以及C/RTL协同仿真的结果分析进而调整自己的设计。然后就是后期通过对深度学习加速器的一些学习，了解了一些开发技巧，如使用循环分块实现更大规模的卷积层处理，实现重复多次调用原有加速器，配合片下BRAM交互获取数据实现，以及帮助增加粗粒度流水优化，提升效率。还有通过课堂上的学习，了解了PYNQ框架，并且在PYNQ-Z2上运行了一些demo，并且自己也在上面使用python写了一些简单的功能帮助理解学习。了解了Vitis AI这门工具的使用，此前在参加Xilinx去年的XDF大会上，在Xilinx的一些开放实验室里面也体验过这门开发工具，感觉这门工具功能很强大，但是入门难度也有些大，这次又在暑期学校课堂上的学习对Vitis AI有了进一步的认识，现在有了HLS开发的基础，相信以后对Vitis的开发入门学习也会很有帮助。暑期学校中，老师对深度学习在PYNQ上的部署讲了很多，但是这方面的知识对没接触过这方面知识的我而言，理解起来会显得有些吃力，但是还是通过老师的讲解和课下查询的一些资料，对这方面有了一点初步了解，比如如何设计一个深度学习的加速器，后面有机会我还会进一步学习。

暑期学校后半阶段主要是组队完成项目，我们队选择的项目是jpeg图像压缩。我们队伍里的同学大多都是跟我一样，之前主要都是使用vivado用传统的HDL纯逻辑设计进行项目的开发，从刚开始接触HLS以及ZYNQ的开发，到这一项目的实战训练，对于我们而言是相当有挑战的。在这一过程中，我们也遇到了很多问题，我们队伍里的人有两个不在一起，但好在我们都是来自于同一个学校实验室的，在交流沟通上不存在多大的问题，大家都很努力，想尽力把项目做好，遇到问题时，大家会想办法一起讨论解决。虽然到项目提交日期，我们的项目还没调试到我们的理想状态，但是这个过程中我也收获了很多，如掌握了vicado HLS这门工具的使用，对ZYNQ的开发有了一个入门。也有了软硬协同开发的思想，如同发现了一个宝库，是懂得了可以将一部分硬件难以实现的开发转移到HLS里面开发，然后再将这部分开发部署到硬件中，还有在使用ZYNQ的开发时，就知道有一些工作可以放到PS里轻松实现，这样在以后的项目开发中，在设计方案时，我相信会给我提供出更多的选择，思维会更开阔。最后，非常感谢暑期学校的老师和同学，在群里和线下帮助解惑；还有感谢刘老师，每晚的助教会议上也会给我们答疑还有打气；最后还有我们努力队友们的互帮互助。

**姓名：刘通 学号：20BO187C**

很荣幸可以参加这次暑期学校的学习，也同样感谢各位老师用心的讲座与答疑。在暑期学校上课之前，自已从接触过Vivado HLS，而是一直用着Verilog DHL语言来设计一些项目，但是随着项目难度的提升，渐渐的发现硬件的基础描述语言实现一些复杂的算法与一些浮点数的运算设计会很是复杂。而这次假期学校所主讲的内容正是我所需要的知识。HLS可以轻松地解决常见的协议系统设计难题、简化FIFO和存储器访问。从两个矩阵的乘法的HLS设计优化的入门，再到 AI中卷积神经网络加速器的设计优化，这些都是一些对我十分有用并走在技术前端的知识。从暑期学校的这些天，我首先掌握的是Vivado HLS的设计流程，然后通过具体的实验（颜色空间转化、DCT变换等）对设计如何进行优化，以及各个优化指令（包括UNROLL、ARRAY\_RESHAPE、loop\_tripcount、 PIPELINE 、DataFlow 、Stream 、INLINE等）的作用进行了了解与测试。随后我们进行了更深一步的学习——设计一个AI加速器，在设计这个AI加速器中我们不仅用到了前两天学习的设计流程、项目指令优化的知识，还学习到了新的优化知识（量化，稀疏化和脉动矩阵）。量化矩阵可以以定点数代替浮点数使得计算效率更高，资源消耗少，节省存储资源占用，节省数据位宽。精度几乎无影响。稀疏化的目的是将很多权值参数接近于零的，对于这些小于一定预值的权重参数直接设置为零。等价于将原来的这些权值参数的神经元链接删除。使得计算减小。脉动矩阵使得输入的数据不再传递给多个PE,而是让数据在PE之间传递，达到计算的流水话架构。这些优化相比于前两天的指令优化显然是一种更深层次的优化。最后还有AI加速器获奖者的作品分享。让我知道到了一个好的AI加速应该怎样设计和优化。最后我们通过HLS设计出图像边缘检测实验的IP并在PYNQ上实现了硬件测试，实现了一个完整的项目。

项目收获与总结：在为期一个星期的项目设计中，首先我对JPEG编码有了很深的理解，JPEG编码过程主要包括;颜色空间转换（RGB2YUV）、8x8图像块、DCT变化、量化、zig\_zag矩阵扫描和编码（游程编码、DC/AC系数差分编码、huffman熵编码）几个步骤。其中最重要的就是2维DCT将我们图像数据的低频与高频数据分开，然后通过量化使矩阵的右下角的高频成分多数变为0,随后对数据进行Z子扫描后，进行编码达到更近一步扫描，我们首先从matlab仿真算法入手，然后对算法进行了C设计，然后将C改为更适合硬件的HLS语言。因为颜色空间转化，DCT变化涉及到浮点数计算，所以我们将浮点数使用AP\_fixed改为定点数计算。随后还对循环进行流水化优化、数据传输使用AXI接口、cos计算由Cordic变为LUT的方式对设计进行优化，通过这次项目让我更近一步的掌握了HLS的设计方法与优化。同时我会再接再励用HLS开发更多的项目。最后再次感谢各位老师与队友们的帮助！

**姓名：李宇 学号：20BO181A**

我在这一次的xilinx暑期培训中报的是B班。B班的内容对我来说是有一定难度的，但是这都是我意料到了的。通过这次的培训，让我了解到了HLS这个软件，知道了它在优化加速方面的强大功能。除此之外，我也对fpga在AI上的应用有了初步了解。通过跟着大家做项目的过程让我体会了一遍做项目的流程，发现了自己许多的知识盲区，比如：matlab的使用，FPGA模块之间的联调等等。因为我是即将进入研究生阶段学习的学生，所以这一次培训必然会给我三年的研究生生涯指明学习的方向。

在和别人的交流之中也了解了我们行业的发展方向以及别人的学习方法，通过与人交流也可以快速的促进我们的进步。