目录

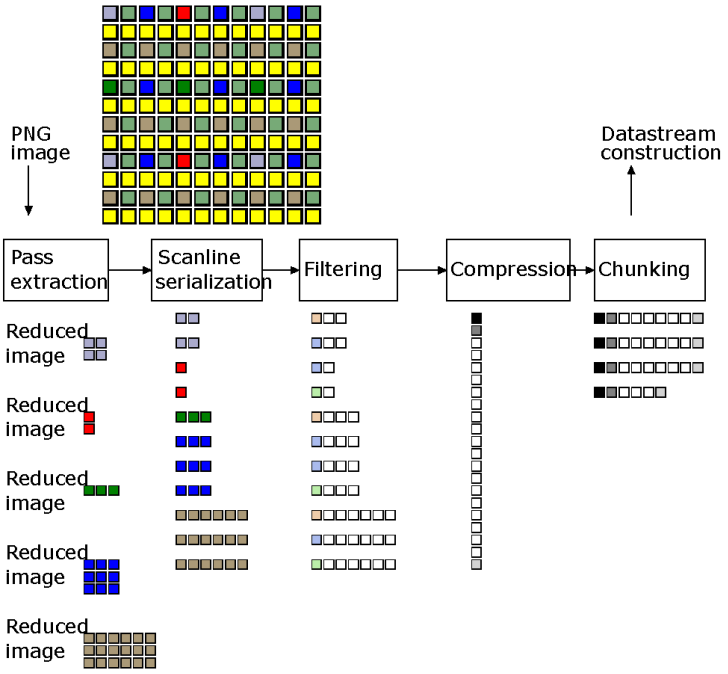
一、 背景介绍

PNG（Portable Network Graphics，便携式网络图形）是一种支持无损压缩的位图图形格式。与 GIF、JPEG 等其他常见的格式相比，PNG 具有无损压缩、支持多种颜色类型、支持透明特性、优化的网络传输显示等特性，因而成为了目前主流的图像格式之一。PNG 的 1.0 版本规范于 1996 年发布；现行版本是国际标准（ISO/IEC 15948:2003），并在 2003 年作为 W3C 建议发布。

根据我们目前能搜索到的资料，网络上对 PNG 的介绍几乎都停留在几个核心算法（如 deflate 算法）层面，而缺少完整应用实现的介绍。在软硬件参考实现方面，只有 libpng、lodepng 等开源的软件库，并且这些开源软件库往往需要支持 PNG 标准的绝大部分特性，同时提供一系列的应用程序编程接口（即往往是用作组件而不是一个应用），这不利于入门学习和实现。因此我们小组计划用硬件实现一个 PNG 编码器，并且给出其对应的参考软件（cmodel），意义如下：利于学习 PNG 的几个核心算法，学习 PNG 编码器的完整应用实现；从空白开始实现一个 PNG 硬件编码器，硬件编码器往往比软件编码器具有更好的性能，具有很强的实用性，并且与我们的专业和本课程都紧密相关；另外，以上的知识和流程也都可以类比到其他的音频、视频、图像的压缩、编码器实现过程中。因此，我们选择本题目，具有很好的理论意义和实践意义。

二、 原理

PNG 图像的压缩过程主要可以分为六个步骤，如下图所示：



1. Pass extraction：如果需要支持渐进式显示，图像像素将按照 Adam7 顺序被重新排列成几个更小的子图，称为 reduced images 或 passes；如果不需要支持渐进式显示，这一步被跳过。
2. Scanline serialization：由一行行的 scanline 形成图像像素数据流。其中，每一行 scanline 是由该行的各像素形成，每一个像素是由该像素的各通道或索引形成。
3. Filtering：分别对每行 scanline 进行滤波，以提高后续的压缩效率。每行 scanline 数据流由滤波后的值形成，并在每行 scanline 前添加一个字节来指示该行的滤波类型。
4. Compression：对滤波后的数据流（字节流）进行 deflate 压缩。deflate 压缩是 PNG 的核心算法，是同时使用了 LZ77 和 Huffman Coding 的一种无损压缩算法；其中 LZ77 是通过“滑动窗”使用已经出现过的相应匹配信息替换当前数据，Huffman Coding 则是将 LZ77 压缩后的符号结果进行编码。压缩后的数据流需要封装成符合 zlib 规范的数据流。
5. Chunking：压缩后的数据流和图像其他信息分别封装成不同的对应 chunk。
6. Datastream construction：chunk 和 PNG signature 组织成最终的 PNG 图像文件/数据流。最终的 PNG 图像文件/数据流由符合 PNG 规范的一个 PNG signature 和其后的一系列 chunk 组成。

三、 设计目标

我们小组计划做一个符合最新标准的简化版硬件 PNG 编码器，采用 verilog 语言实现。简化体现在我们仅支持固定配置，以降低设计复杂度和硬件实现代价。

预期支持的配置项如下表所示：

- 块类型（chunk type）仅支持最基本的三种：
  图像头（IHDR）、图像数据（IDAT）和图像尾（IEND）；
- 色彩类型（color type）仅支持包含透明通道的真彩色（RGBA）；
- 数据位宽（bit width）仅支持每通道 8bits；
- 不支持渐进式显示的隔行扫描（interlace=0）；
- 压缩时仅支持固定哈夫曼编码（BTYPE=1）。

配置变量	支持的配置项
chunk type	IHDR、IDAT、IEND
color type	6（RGBA）
bit width	8bits
interlace	0（no interlace）
BTYPE	1（compressed with fixed huffman codes）

硬件 PNG 编码器的输入为以行为单位连续输入的像素值（含 RGBA 四通道），编码器内部以行为单位进行滤波以及压缩，输出为符合标准的 PNG 码流。在权衡 IO 数和吞吐率后，我们将输入输出数据位宽均定为 32bits。考虑到压缩模块的时钟数是与输入数据相关的变量，行间流水时受限于最慢的一次压缩，我们在每行处理结束后再进行下一行的数据，以降低控制复杂度。

考虑图像中所包含的数据量十分巨大，为加速 debug 以及验证 PNG 编码器的功能，我们将在开源 PNG 编解码库 lodepng 的基础上搭建与硬件对应的简化版 PNG 编码器（cmodel），以检查滤波和压缩两核心模块以及最终输出的 PNG 码流。为进一步验证 PNG 码流的正确性与通用性，我们还将采用 python 中官方的 PNG 编解码库 pypng 进行解码，并与输入 RGBA 数据进行对比。

四、 参考文献

[1]"Portable Network Graphics (PNG) Specification (Second Edition)", ISO/IEC 15948:2003.

[2]"ZLIB Compressed Data Format Specification version 3.3", RFC 1950.

[3]"DEFLATE Compressed Data Format Specification version 1.3", RFC 1951.

[4]lodepng, https://github.com/lvandeve/lodepng.git.